

IMPERIAL

IMPERIAL COLLEGE LONDON
DEPARTMENT OF DESIGN ENGINEERING

On Distributed Ledger Technologies: Designing Decentralised and Fair Algorithmic Applications

Aida Maria Manzano Kharman

Supervised by:
Professor Robert Noel Shorten
Dr. Pietro Ferraro

June 2, 2025

Submitted in part fulfilment of the requirements for the degree of Doctor of Philosophy
in Design Engineering of Imperial College London and the Diploma of Imperial College
London.

Declaration of Originality

I, Aida Maria Manzano Kharman, hereby declare that this thesis and the work reported herein was composed by and originated entirely from me. Information derived from the published and unpublished work of others has been acknowledged in the text and references are given in the list of sources.

Copyright

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-Non Commercial 4.0 International Licence (CC BY-NC).

Under this licence, you may copy and redistribute the material in any medium or format. You may also create and distribute modified versions of the work. This is on the condition that: you credit the author and do not use it, or any derivative works, for a commercial purpose.

When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes.

Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

Abstract

This work focuses on the study of distributed ledger applications, presenting proposals of fair and decentralised applications to counter scenarios in which centralisation of wealth and power are the norm. The first contribution is a novel architecture for a decentralised data market, in which participants crowd-source data and receive a fair share of the reward. The market is shown to be resilient against a number of adversarial behaviours.

Subsequently, an algorithm to prove one's location is presented. This algorithm is a key component necessary to the functioning of the data market. In contrast to prior approaches, the design does not require assumptions of honest participation, nor dependence on an external ground truth to identify malicious actors. It is fully peer-to-peer, robust in highly adversarial settings, and compatible with privacy-preserving techniques. The security and reliability of the algorithm are evaluated empirically and characterised mathematically. The protocol is then generalised into a consensus mechanism applicable beyond location verification. An extended mathematical model is developed for this case, and its performance under varying operational conditions is systematically characterised.

Finally, a study of governance vulnerabilities in Distributed Ledger Technologies is presented. This work provides a taxonomy of formalised properties necessary for good governance, solutions to implement them and an evaluation of how the absence of these cause severe vulnerabilities. The analysis is then extended to realm of Decentralised Autonomous Organisations (DAOs), which are a class of applications implemented on Distributed Ledger Technologies. The findings anticipated several governance exploits that later materialised, incurring losses in the scale of millions for multiple DAOs.

Overall, this thesis aims to contribute to the technological development of distributed ledger applications with the goal of furthering social good. This work presents architectures, algorithms, and governance properties that prioritise fairness, decentralisation, and resilience.

Acknowledgements

I would like to thank my family for their unwavering love and support. To my parents, my partner and extended family that have loved me, held me and encouraged me throughout this joyous journey. I am grateful to my supervisors for their kind and compassionate mentorship, for the trust they gave me, and the fulfilling academic discussions at the whiteboard. To all my colleagues and co-authors, for the enriching collaborations, all of whom I learned a lot from. I deeply thank IOTA Foundation for funding this research. I would also like to thank my close friendships for their warmth and kindness, to all the incredible people I have crossed paths with during my time in London who have made every day a gift. Finally, I would like to thank my handball families at Imperial and Chelsea, that brought me endless joy and a happy place to play. To the coaches and the support team I have had, that believed in me, and granted me the gift of strength and movement, and confidence to believe in myself.

Dedication

For my family.

Contents

1	Introduction	11
1.1	Background	12
1.2	Challenges	15
1.2.1	Threats to System Integrity: The Adversarial Landscape	15
1.2.2	Fundamental Design Dilemmas: The "No Free Lunch" Principle	17
1.2.3	The Challenge of Auditability and Transparency	17
1.2.4	The Imperative of Trust Assumptions	18
1.3	Research Objectives	19
1.4	Contributions	19
1.4.1	Publications	20
1.5	Structure	20
2	Preliminaries	21
2.1	The Crypto in Crypto(currencies)	21
2.1.1	Basics: What Is Encryption?	22
2.1.2	Key Generation	22
2.1.3	Symmetric Encryption	25
2.1.4	Public-Key Cryptography	28
2.1.5	Hash Functions	32
2.1.6	Zero Knowledge Proofs	33
2.1.7	Homomorphic Encryption	36
2.1.8	Post Quantum Cryptography	37
2.2	DLTs: Architectures and Flavours	39
2.2.1	Architectures	40
2.2.2	Consensus Algorithms	41
2.2.3	Transaction Record Model	42
3	A proposal for a decentralised data market	44
3.1	Contributions Disclosure	44
3.2	Motivation	44
3.3	Research Scope	47
3.4	Related Work	48
3.5	Architecture	49

3.6	Components	52
3.7	Algorithms	53
3.7.1	Verification Algorithm	53
3.7.2	Voting Algorithm	54
3.7.3	Data Consensus	59
3.7.4	Access Control Mechanism	61
3.8	Robustness of data market in adversarial environments	64
3.8.1	Simulation Setup	66
3.8.2	Evaluation of Results	67
3.9	Conclusions	70
3.10	Critical evaluation of this work	70
4	Proof of Position Algorithms	72
4.1	Contributions Disclosure	72
4.2	Motivation	73
4.3	Related Work	75
4.4	Components	77
4.5	Overview of T-PoP	78
4.6	TPoP Protocol	79
4.6.1	Tree Building	80
4.6.2	Checks Algorithm	81
4.6.3	Verification	82
4.7	Preliminary Theoretical Analysis	84
4.8	Agent-Based Simulations	86
4.9	Preliminary Results	88
4.10	Comparison of Graphical Probability Model and Simulated Environment . .	91
4.11	Conclusion	92
5	Proof of Position Algorithms: Extended Model	93
5.1	Contributions Disclosure	93
5.2	Introduction	93
5.3	Mathematical Modelling of T-PoP	94
5.3.1	Agent States	95
5.3.2	Condition 1: Child to Parent Approval	100
5.3.3	Condition 2: Tree Has Sufficient Nodes	108
5.3.4	Condition 3: All Nodes Are Unique	108
5.4	T-PoP's Complexity	112
5.5	Detecting Platoon Attacks	114
5.6	Agent Based Simulations	116
5.7	Performance of TPoP	124
5.8	Implementation Considerations	128
5.9	Limitations and Future Work	129
5.10	Conclusion	129

6	Tree Consensus Algorithms	131
6.1	Contributions Disclosure	131
6.2	Motivation	131
6.3	Related Work	132
6.4	Mathematical Model	134
6.5	Parameter Selection of TCA	139
6.6	Evaluation	140
7	Distributed Ledger Technologies: Challenges in Governance	144
7.1	Contributions Disclosure	144
7.2	Motivation	144
7.3	Related Work	146
7.4	Methodology	147
7.5	Governance Components	149
7.6	Vulnerabilities	150
7.6.1	Exploits leveraging public votes	150
7.6.2	Exploit leveraging non-binding elections	152
7.6.3	Exploits leveraging purchasable voting power	153
7.6.4	Other vulnerabilities	153
7.7	Governance Properties	154
7.7.1	Participation	154
7.7.2	Incentives	156
7.7.3	Voting Mechanism	157
7.7.4	Security	158
7.8	Evaluation	161
7.8.1	Participation	163
7.8.2	Incentives	163
7.8.3	Voting Mechanism	164
7.8.4	Security	164
7.8.5	Missing properties	165
7.9	Conclusion	166
8	DAO Governance: Systemic Failures and a \$163 Million Heist	168
8.1	Contributions Disclosure	168
8.2	Motivation	168
8.3	DAO Governance: From Promise to Peril	169
8.4	Necessity of Secrecy	171
8.5	DAO Voting: Survival of the Richest	171
8.5.1	Off-Chain Voting and Hybrid Alternatives	172
8.6	Current Solutions: Insecure and Insufficient	173
8.7	Aftermath: the DAO Casualties	176
9	Conclusion	179

Bibliography	181
A An example of an MEV election	217
B Instances of Fraud in Web3 Applications	220

List of Figures

2.1	Public Key Cryptography, by David Gothberg - Public Domain, source: Wikimedia.	23
2.2	Symmetric Key Cryptography, by Michel Bakni - CC BY-SA 4.0, source: Wikimedia.	23
3.1	Data Market Architecture. Credit for the images is given in ¹	50
3.2	Example of an election performed with C-MEV with three winners chosen and three rounds.	59
3.3	Access Control Mechanism: Agents in coalition C_{q_1} must compute the Shapley value, $\psi(C_{q_2})$, of the new incoming data, X_{t+1} , using X_{t+1} and $v_{C_{q_1}}(\cdot)$. Once the agents of C_{q_2} receive their Shapley value, they must complete an amount of proof-of-work that is inversely proportional to their Shapley value. This work is to compute the Shapley value of $\psi(X_{t+2})$ and so on. Once a coalition completes this work they may enter the market.	62
3.4	Correlation matrix of the different contaminants with the Air Quality Index.	63
3.5	SHAP values of each of the data points per contaminant.	64
3.6	Mean of the absolute SHAP values, aggregated per contaminant.	65
3.7	Characterisation of data consensus algorithms' behaviour under different degrees of coordinated data poisoning attacks.	66
3.8	Breakdown analysis of data consensus algorithms, with a coordinated data poisoning attack.	67
3.9	Characterisation of breakdown of C-MEV combined with median-of-means algorithm	69
4.1	An example of the BBC using ACLED data to report on protests in Iran, supplementing the data with landmark cross-referencing for location verification. The article in question is [228]. Reproduced under fair dealing for the purpose of review, © BBC.	75
4.2	Overview of the Tree Proof of Position steps.	79
4.3	Tree building examples. Agent a_i constructs a commitment to their alleged position $\hat{p}_i$. The panel on the top right shows the construction of a tree for $d = 1$ and $w_1 = 4$, while the panel on the bottom right shows the construction of a tree for $d = 2, w_1 = 2, w_2 = 2$	81
4.4	Example of T-PoP algorithm with $d = 2, w_1 = 2, w_2 = 2$	83

4.5	Graphical Probability Model of T-PoP with parameters $h = 2, w_1 = 2, w_2 = 2$. The red lines indicate that those variables influence the output of a specific node.	85
4.6	T-PoP performance for the graphical probability model (depicted in Figure 4.5). The panels in the left column show the reliability rate (True Positives), while the panels in the right column show the security rate (True Negatives). The first row is associated with model parameters, $\theta_1 \{t = 1, h = 1, w_1 = 6\}$, while the second row is associated with model parameters, $\theta_2 = \{t = 1, h = 2, w_1 = 2, w_2 = 2\}$	87
4.7	T-PoP performance for the agent-based simulator model. The panels in the left column show the reliability rate (True Positives), while the panels in the right column show the security rate (True Negatives). The first row is associated with model parameters $\theta_1 = \{t = 1, h = 1, w_1 = 6\}$, while the second row is associated with model parameters $\theta_2 = \{t = 1, h = 2, w_1 = 2, w_2 = 2\}$. Notice the close similarity to Figure 4.6.	88
4.8	Jensen-Shannon divergence (JSD) between the agent-based simulator reliability TP_s , and the probability graphical model reliability TP_m (left column), and between the agent-based simulator security TN_s , and the probability graphical model security TN_m (right column). Results are shown for $\theta_1 = \{t = 1, h = 1, w_1 = 6\}$ (top row) and $\theta_2 = \{t = 1, h = 2, w_1 = 2, w_2 = 2\}$ (bottom row).	89
5.1	Venn diagram representing possible agent state combinations.	96
5.2	Possible scenarios in which a dishonest agent may appear in a prover tree.	99
5.3	Process of parent to child approval	100
5.4	Visual representation of a tree architecture with dimensions $\theta = \{t = 1, h = 1, w_1 = 6\}$	104
5.5	Theoretical performance of TPoP with $\theta_1 = \{t = 1, h = 1, w_1 = 6\}$	105
5.6	Visual representation of a tree architecture with dimensions $\theta_2 = \{t = 1, h = 2, w_1 = 2, w_2 = 2\}$	105
5.7	Theoretical performance of TPoP with $\theta_2 = \{t = 1, h = 2, w_1 = 2, w_2 = 2\}$	106
5.8	Expected number of edges in the tree.	107
5.9	Number of edges in agent's trees after running T-PoP.	109
5.10	A) A tree of height $h = 2$ and branching factor $w = 2$, with agents and levels labelled. B) Two separated scenarios visualized, when agents a_2 and a_3 selected at random in the field of view of a_1 are $\delta < r$ nearby (top) versus $\delta > r$ far apart (bottom).	111

5.11	Validation of theoretical approximation for probability of a randomly sampled tree ($h = w = 2$) containing 7 distinct agents. The results show the probability as a function of the number of agents N uniformly distributed independently and at random over a unit square plane. The red curve shows the probability calculated through a numerical experiment, where a tree is generated following the tree-building algorithm 2, starting with a randomly selected prover. The dashed curve shows the value of the theoretical approximation denoted $P(L_0, L_1, L_2)$ calculated by equations (5.25)-(5.28). . .	113
5.12	Expected number of edges in a platoon tree of height 1 and branching factor 6, with threshold = 100%.	116
5.13	Expected number of edges in a platoon tree of height 2 and branching factor 2, with threshold = 100%.	117
5.14	Low density performance of TPoP with a tree of height 2 and branching factor 2, and a threshold = 100%.	120
5.15	Low density performance of TPoP with a tree of height 1 and branching factor 6, and a threshold = 100%.	121
5.16	Low density performance of TPoP with a tree of height 2 and branching factor 2, and a threshold = 40%.	122
5.17	Low density performance of TPoP with a tree of height 1 and branching factor 6, and a threshold = 40%.	123
5.18	High density performance of TPoP with a tree of height 2 and branching factor 2, and a threshold = 100%.	124
5.19	High density performance of TPoP with a tree of height 1 and branching factor 6, and a threshold = 100%.	125
5.20	High density performance of TPoP with a tree of height 2 and branching factor 2, and a threshold = 40%.	126
5.21	High density performance of TPoP with a tree of height 1 and branching factor 6, and a threshold = 40%.	127
6.1	Probability of the Tree Consensus Algorithm accepting a tree, for various tree dimensions. The branching factor is kept constant across depth levels. Convention follows that apex means that the probability of accepting a claim, p_a , is 1, and the probability of rejecting, p_r , and ignoring the claim, p_i , is 0. The left corner vertex indicates $p_r = 1$, and right corner vertex means $p_i = 1$	141
6.2	Expected number of approvals for a given tree, for various heights and branching factors. The branching factor is kept constant across depth levels.	142
7.1	PRISMA flow diagram for systematic review of literature conducted.	148
8.1	Vocdoni's Zero Knowledge Rollup Proposal. Source: Vocdoni.	175
8.2	Vocdoni's Anonymous Voting Schema. Source: Vocdoni.	176

A.1	A prize wheel for sampling an outcome ordering $o \in \mathcal{O}$ from a probability measure π .	218
-----	---	-----

List of Tables

2.1	XOR operation	25
2.2	AND operation	26
2.3	Current quantum physical computation devices.	39
3.1	The lower-carnality ordering set when $A = \{a_i, a_j, a_k\}$ and one winner is selected. Agents in the same brackets are given the same rank in an ordering.	58
4.1	A truth table showing confirmation (1) or rejection (0) of a parent's (a_i) position by a witness (a_j), depending on the honesty (h) and coercion (c) states of each agent. Notice that the relationship between a_i and a_j is symmetrical.	86
4.2	Jensen-Shannon divergence (JSD) between the normalized reliability (TP) and security (TN) maps for two sets of model parameters.	91
5.1	Parent to child witness selections.	101
5.2	Approvals of a child to a parent.	103
5.3	Optimal points to maximise honest to honest agent approvals.	116
6.1	Values assigned to each possible agent response.	135
6.2	Probability of each agent's possible response to varying claims.	135
6.3	Possible agent responses for a single parent with two children.	136
7.1	Governance properties satisfied in varying cryptocurrencies.	162
7.2	Voting thresholds for proposal approval across various cryptocurrencies.	164
8.1	Current DAO governance solutions.	176
8.2	DAO hacks and hostile takeovers exploiting governance models.	178
A.1	The lower-cardinality ordering set when $A = \{a_i, a_j, a_k\}$ and $K = 1$. Agents in the same brackets are given the same rank in an ordering.	217

Chapter 1

Introduction

The collection of work carried out in this thesis focuses on designing new Distributed Ledger Technology (DLT) applications. I explore building algorithms, structures and architectures that leverage Distributed Ledger Technologies, and examine in these proposals the following questions: are these systems resilient to attack vectors? Are they *truly* decentralised, fair and secure? Do they safeguard their users' privacy? Are they auditable and verifiable? In all proposed systems, the motivation I pursue is to further what I believe was the original mission of DLTs: to add positive value to society and further its uses for social good. To answer these questions I first explore formalised notions of fairness, decentralisation, verifiability and so on. In cases where there is no consensus on the accepted formal notion in the academic field, I either survey the options available, or construct bespoke formalisations specific to the context of this work. I then evaluate the proposed algorithm or architecture against these notions, and stress-test it to find where it breaks and where its vulnerabilities lie.

In some cases, it is not possible to find a solution that satisfies all desirable criteria, and so I explore where compromises can be made. At the conclusion of this journey, I make three main contributions to the field: i) a proposed architecture to share data (or any digital asset) that is fair, decentralised and resilient to a number of attack vectors; ii) a consensus algorithm to increase trust amongst mutually distrusting agents, that is decentralised, privacy-preserving and does not rely on assumptions of honest behaviour; and ii) a comprehensive study of the governance related vulnerabilities that afflict DLTs and by extension DAOs (Decentralised Autonomous Organisations). In the latter, I also provide a framework for practitioners where solutions to these vulnerabilities are outlined.

These projects are closely linked and co-dependent to achieve my motivation of designing fair, decentralised and resilient systems. My research began with creating an architecture for a decentralised data market, and in doing so, I found it necessary to develop a method to determine whether the data being transacted was real or artificial. Solutions exist to determine this, but are de-facto centralised at best. Designing a decentralised data market that relies on a centralised verification mechanism seemed to defeat the purpose of

a decentralised architecture. Thus I embarked on the project of developing, testing and characterising a peer-to-peer algorithm to probabilistically determine whether a data-point is real. Later I realised that this algorithm may be used beyond this specific application, to verify whether wider class of generalised statements are true or false. I proceeded to adapt the mathematical modelling of the algorithm to fit this generalisation and test it for various operating conditions. Finally, I explored governance related vulnerabilities of DLT architectures and applications built on them. The proposed data market architecture is implemented on a DLT, and therefore it is only as decentralised and secure as the DLT protocol itself. In earlier work I had studied vulnerabilities of electronic voting protocols, and learned that it is possible to coerce and bribe participants if certain properties are absent in a voting protocol. This inspired my research of DLT governance, where I uncover that these necessary properties are absent in all DLT platforms I review, which as a consequence can enable voter coercion, bribing, centralisation of power, plutocracies and game-theoretic incentives for to undermine governance protocols for individual gain. No matter how secure, decentralised, fair, or private a protocol is, if it relies on a DLT that fails to provide these properties, the application cannot uphold them either. Hence the motivation to pursue this line of research.

The work presented in this thesis aims to advance the state of the art in decentralised systems and contribute to grounding the design of Web3 applications in rigorous, scientifically verifiable methods. During this research, I encountered many projects in the ecosystem which make strong claims about decentralisation, fairness, or security without adequate scientific justification. Such claims, while compelling to non-expert users, can obscure critical assumptions or limitations. This thesis also invites reflection on the assumptions underpinning decentralised architectures. For example, if a protocol is, in theory, decentralised, and its security relies on access to a centralised ground truth to detect malicious behaviour, this may undermine the very decentralisation it seeks to achieve. In such cases, the trade-offs in performance, complexity, and scalability of decentralised systems may not be justified, particularly if a central authority is assumed trustworthy by design.

1.1 Background

A Distributed Ledger is a decentralised, peer-to-peer record of information [424]. It is maintained by a network of nodes (computers) where information is added following certain rules [527]. Given it is decentralised, nodes must agree on what information to add. This decision is achieved following a *consensus mechanism*: a set of instructions that nodes follow to agree on what information to add to the record [358]. Why are DLTs of importance? Since their emergence, a vast array of projects have been built on DLT architectures, most notoriously cryptocurrencies¹.

Whilst some earlier work laid the foundations [518], [143], [116], [40] it is Satoshi Nakamoto's²

¹A digital currency, also treated as an asset, that is transacted on a DLT peer-to-peer network, and records of these transactions are maintained on said ledger [14].

²The author's true identity is unknown, and it is unclear if this name is perhaps a pseudonym. It could

landmark paper *A Peer-to-Peer Electronic Cash System* [374] proposes the architecture of the first cryptocurrency: Bitcoin.

Context is critical to understand the importance of Nakamoto’s work: Bitcoin is proposed in the wake of the Great Recession of 2008. It is conceived as a reactionary and idealistic attempt to redistribute power and decentralise decision-making of financial service providers. Satoshi states in 2009: *“The root problem with conventional currency is all the trust required to make it work. The central bank must be trusted not to debase the currency, but the history of fiat currencies is full of breaches of that trust. Banks must be trusted to hold our money and transfer it electronically, but they lend it in waves of credit bubbles with barely a fraction in reserve. We have to trust them with our privacy, trust them not to let identity thieves drain our accounts. Their massive overhead costs make micropayments impossible.”* [373].

With this work, Satoshi Nakamoto is widely regarded to have kickstarted the Web3 revolution. Web3 is a loosely defined term that commonly refers to the next iteration of the World Wide Web. Compared to Web 2 where platforms are centralised and users interact with them, Web3 champions shared user ownership, decentralisation and distributed ledger technologies [403]. A key characteristic of Web3 services is that users are also service providers, as well as owners of the service itself.

Some examples of services built on and enabled by DLTs include cryptocurrencies, cryptocurrency exchanges³, data markets (or other digital assets marketplaces such as Non Fungible Token (NFTs) marketplaces), Decentralised Autonomous Organisations (DAOs)⁴, voting platforms and cross-ledger bridges⁵ amongst others.

The DLT ecosystem is vibrant and proliferating, and the number of projects in the space have been steadily increasing since 2017, with more than 10,000 coins and tokens in circulation as of June 2025⁶. The impact of the industry is not only marked by number of projects in it, but also by their worth. At time of writing (June 2025) the total market capitalisation for cryptocurrencies is reported at \$3.31 trillion⁷, and the total treasury of DAOs is \$21.4 billion [157]. This growth has been marked by a high rate of technological innovation, particularly in the field of applied cryptography. A noteworthy shift has also been observed in the participation of these ecosystems. They have become more mainstream and accessible, beyond the highly technical and specialised user. Early adopters were often developers, self-described members of the cypherpunk⁸ community, and the maintainers and service providers of these technologies (and operating perhaps on the margins of le-

be you!

³Platforms where users may exchange one cryptocurrency for another [541].

⁴An online co-operative built on DLT architectures where participants hold stake in the organisation and have voting rights on how the organisation is run and what projects are implemented [371].

⁵A technology that connects one DLT network to another [542].

⁶Source: Statista.

⁷Source: CoinMarketCap.

⁸A movement originating in the 1980s where participants advocate for the use of cryptography and the right to privacy. Participants often ascribe to libertarian ideology, favour decentralisation and champion freedom from centralised authority and surveillance [263].

gality). Nowadays, non-expert users are able to transact and interact in these systems, without a requirement of understanding technical minutiae of said protocols. What may be interpreted as a strong indicator that DLTs are here to stay is institutional adoption and regulation efforts. Major financial institutions now develop and adopt crypto-asset services for their clients, governments have developed regulation [249, 551] and some even piloting their own form of digital currencies, albeit institutionalised and centralised [33].

Examples include El Salvador’s⁹ recognition of Bitcoin as legal tender[19], the EU’s Markets in Crypto-Assets Regulation (MiCA) [551], and various central bank digital currency (CBDC) pilots across the world [33]. Political and social organisations have also begun adopting DAO governance tools—such as the Catalanian Esquerra Republicana party¹⁰ and New Belarus’ experimentation with blockchain voting platforms¹¹. Established banks such as JP Morgan Chase¹², Goldman Sachs¹³, BBVA¹⁴ and Santander¹⁵ have launched crypto-asset trading and/or custody services. Even highly publicised figures, such as the current U.S. President Donald Trump, have issued blockchain-based digital assets [181]. While these developments are not without controversy—particularly regarding volatility, security, and governance—it is increasingly clear that digital assets and decentralised systems will remain a significant part of the financial and technological landscape.

Although this increase in adoption has heralded innovation and a number of projects driven to further social good, not all has been positive. The darker side of the crypto boom reveals a high incidence of scams, heists, hacks, theft and other criminal activity such as money laundering and terrorism funding [474]. A list of such examples is available in Appendix B. The scale and impact of these attacks is sobering, with the total amount of funds lost being in the magnitude order of billions [112, p. 80], [558]. A major concern arising in these decentralised systems is who is liable when the system fails, and who should compensate and protect the affected victims. Centralised institutions facilitate traceability of liability. And yet, the inception of Bitcoin was rooted in deep mistrust of centralised financial institutions.¹⁶ This sentiment emerged as consequence of their engaging in excessive risk and borrowing, inadequate financial regulation and unethical behaviour post and during the 2008 recession [190]. The general public lacked the means to easily audit and verify these institutions’ decisions, and this argument often cited by advocates of DLTs as a reason to champion decentralised systems instead. Hence the Web3 revolution, the appeal of decentralisation: everything is auditable, decentralised, verifiable and therefore more trustworthy— *allegedly*. It is hard to empirically measure trust in centralised institutions like governments, but recent developments like the 2017 Cambridge Analytica scandal [75], the 2020 January 6th Capitol Attacks [141] in the United States and the 2024 Romanian

⁹El Salvador also created the world’s first Bitcoin-based local economy, also dubbed as the ‘Bitcoin Beach’.

¹⁰Source: EuopaPress.

¹¹Source: Vocdoni.

¹²Source: JP Morgan.

¹³Source: Goldman Sachs.

¹⁴Source: BBVA.

¹⁵Source: Santander.

¹⁶Ironically, many of them are now adopting distributed ledger technologies.

presidential election annulment [409] evidence a pattern of unrest. These incidents all have in common the attempt to influence election results by spreading misinformation and/or targetting and manipulating the electorate. In an era where generating false information is easy and cheap, how can we distinguish information that looks real, from information that *is* real?

This is the heart of my motivation for this PhD. I came to the conclusion that the answer to that question is built on two tenets: consensus and verifiability. A Russian proverb: ‘*trust but verify*’ eloquently captures the importance of verifiability. Verifiability is at the heart of what we distinguish as truthful and trustworthy, from what we do not [301]. Even if individually we do not always exercise our power to verify, knowing we can, makes us trust [314]. The second necessary ingredient is consensus. This seems especially important when verification is hard, inaccessible, costly or time consuming. An example: if a highly unlikely event is reported, a reader is more likely to trust this event is true if a) a number of other sources report the same [238] and b) if the sources provide information that is verifiable for the reader [532]. The first of these highlights the importance of consensus, the agreement of the majority. The *argumentum ad populum* fallacy is the implication that if many people believe something to be true, then it must be so. Whilst this is evidently untrue, it is not far off from how we operate in society in our day to day lives. We trust sources and information built on a web of trust, cultivated through consensus [122], but hinging on our power to audit through verifiability [396]. It is these two concepts: consensus and verifiability, that have driven my research here presented.

1.2 Challenges

DLTs promise decentralisation, transparency and fairness. The system operates through consensus, and does not have a single point of failure. Users share responsibility in maintaining the network and providing service, and the ownership of the resource transacted on the network is also shared. This feature is novel and unique in the digital space, and whilst having heralded a paradigm shift and hosts of innovation, it is not free of challenges. The distributed nature of these architectures introduces a complex landscape of potential vulnerabilities and design dilemmas [237]. In this section, I provide a high-level overview of these challenges, grouping them by category. Whilst this is a non-comprehensive list, it serves as a starting point for the reader to gain a bird’s-eye view perspective of the current landscape of difficulties. In subsequent chapters, the application specific challenges will be explored and outlined in greater depth.

1.2.1 Threats to System Integrity: The Adversarial Landscape

Distributed Ledger Technologies are an implementation of a decentralised computer network. As such, they are vulnerable to the known possible attacks that any decentralised network can face, but they also present new challenges, unique to DLTs.

Let us begin by exploring the threat landscape of decentralised networks. Some of these

threats attempt to attack honest nodes in the network, for example: eclipse attacks [242], man-in-the-middle attacks [176], replay attacks [405] and traffic analysis [418]. Others attempt to undermine the network’s integrity by rendering the network incapable of providing its purposed service. Examples include denial of service attacks [514], network partitioning [286] and routing attacks. Finally, other attacks are mounted with the intention to gain unfair advantages in the network or obtain undue power, such as Sybil attacks [166], data poisoning [554], exploiting bugs or zero-day vulnerabilities [312] or injecting malware into the network protocol.

DLTs also introduce new, often more complex, attack vectors due to their specific architectural design [558]. Some of these also aim to partition the network and compromise its integrity. Ultimately, DLTs must reflect one version of the truth only, one sequence of valid transactions, and some attacks aim to re-write this accepted sequence, or partition the network such that other nodes are unsure which is correct. Some examples include forking attacks [38] (splitting the network), spamming attacks [553]—in which users send high volumes of bogus or empty transactions to slow down the network and time until consensus is reached—and double spend attacks [381]. The latter also giving attackers unfair advantages by allowing them to spend the same funds twice. Forking can sometimes occur as a consequence of legitimate disagreements in the DLT as to which version of the truth is accepted.

It is interesting to observe a nuance in these attacks: some may argue that they are not vulnerabilities but rather design choices. This ideology is encapsulated by the ‘*code is law*’ belief. Its proponents argue that if a distributed ledger protocol’s code permits a particular behaviour—even if such behaviour has adverse consequences for other participants or the system at large—then exploiting these affordances is not inherently malicious [350]. Rather, it is seen as operating within the bounds of the system’s rules, since, in this view, the code itself constitutes the ultimate authority. Such exploits, or design choices (depending on who you ask) also include front running [145]—where participants reorder transactions to extract financial advantage—or 51% attacks [394], in which an entity controlling a majority of network power can rewrite transaction history, or forcibly censor others, are considered technically legitimate, even if they threaten the foundational premise of decentralisation. In the context of DLT governance, this concept has been used to justify vulnerabilities such as heists and forcible take-overs, and exploiting smart contract security vulnerabilities (which often go undetected until funds have been siphoned). Finally, there are a separate class of attacks which directly prey on the DLT users. These are not the ones in which users may stretch the protocol rules to their own personal gain, but rather those in which the DLT project itself is a scam and purposely designed to be taken advantage of: not by the users but by the creators of the project. In this category the reader can find all class of nefarious schemes: rug-pulls, Ponzi schemes, unaudited projects positing as decentralised but in fact being nothing more than a centralised database with no real value or codebase: the so called ‘pump-and-dumps’, or simply put: a fraud [474].

Whilst some of these threats are mitigated by leveraging standard security primitives,

others require bespoke solutions specific to the DLT application. It is natural to assume that exploitable design choices should be minimised, however, doing so without sacrificing other desirable properties is not always straightforward. This matter gives rise to a new class of challenges: the design dilemmas DLTs face.

1.2.2 Fundamental Design Dilemmas: The "No Free Lunch" Principle

There are properties that are necessary to protect DLT applications from attack, such as auditability, coercion-resistance, immutability, security and verifiability. Unfortunately, it has been identified that frequently, some desirable properties come at the expense of others. A key challenge therefore is to find a compromise without incurring major costs in the application's performance and security. A well known 'trilemma' in the DLT community is the one of security, decentralisation and scalability [372]. Decentralisation is a key requirement of DLTs. Indeed, delivering this property was the driving purpose of the creators of Distributed Ledger Technologies. Forgoing centralisation however, comes at the cost of speed. Transaction processing is slower, as consensus is necessary to ensure only one correct version of the ledger remains. As DLTs invest greater research effort in scalability, or in other words, speeding up their technology and increasing the rate of service provided, whilst maintaining a permissionless and open ledger, a cost is often paid on the security front. Security audits are necessary to avoid threats such as smart contract vulnerabilities, incentives to undermine the protocol and a whole host of attacks possible described in the previous section. This is one example of desirable properties that are in conflict with each other, but we uncover more in Chapter 7. The challenge of finding an appropriate compromise is an ongoing one, but in this work, one of the aims is to shed light on an important nuance: some properties are mutually exclusive or at a tension with one another by design, others need not be. This nuance is further explored in the context of governance in both Chapters 7 and 8.

1.2.3 The Challenge of Auditability and Transparency

Another challenge that has emerged in the space is that of auditability. This may be rather perplexing for the reader, as one may assume that the open-source nature of DLTs would facilitate audits and traceability when exploits happen. In practise, a number of heists, smart contract exploits and design flaws in DLT protocols have been mounted, but finding culprits is not a straightforward task. Exploits and threats may go unnoticed until the damage has been done, in which case recovering stolen funds can be very difficult [54, 455, 499, 62]. Technologies such as cryptocurrency tumblers, also known as mixers, enable wrongdoers to anonymise transactions of their stolen funds by mixing them with other honest transactions [497]. Although many of these platforms have faced legal repercussions in some countries [270, 481], they are still available to use in others.

In reality, malicious actors need not rely on these techniques to do away with stolen or questionably obtained funds. They may split the funds into a large number of wallet addresses and send many transactions of smaller amounts, mixed with transactions of un-

suspicious funds, effectively creating a highly complex web of transactions making traceability markedly hard [499]. Even in the case of specialised task-forces which may succeed in finding the wallet addresses holding stolen funds, (which are often law enforcement or specialised private investigation labs), obtaining the stolen funds is another separate problem. Without a wallet’s key, it is impossible to obtain the funds held in it [550]. Unless the key is held by a decentralised exchange, or leaked, or seized through law enforcement searches (this is assumed it is found or held in some cyber/physical trace), the funds cannot be extracted due to the cryptography in place that secures it. Usually the method to identify culprits lies in the exit point from the digital space into the ‘real world’: the moment in which they exchange the stolen cryptocurrency for a tangible asset [508, p. 16]. This transaction is more likely to leave a trace involving information such as names or bank accounts or other personally identifiable data. However, assuming malicious actors already operate on the margins of legality, it is not unfathomable that these funds may simply be laundered, or cashed out for other kinds of illicit services that accept payment in cryptocurrencies with no questions asked [124].

1.2.4 The Imperative of Trust Assumptions

Selecting appropriate trust assumptions was another significant challenge encountered during this work. This challenge is fundamental to the design of DLT architectures. DLTs were born out of a deep distrust of centralised institutions: they aim to minimise centralising trust, but they do not eliminate trust altogether. More precisely, permissionless DLT applications such as Bitcoin, aim to operate without requiring trust between participants for the system to function. This design choice does not imply that trust is eliminated, rather, it is shifted onto a different component of the DLT. Much like a debt, at some point the price must be paid, and in the case of Bitcoin, the consequence is considering proof-of-work (computational power) a proxy for trust. In this case, the price paid is: speed, energy footprint and in practise, centralisation of transaction approval rate [293]. This assumption implies that the agent with greatest computational power should be trusted to dictate what is the next valid transaction. In practise this means that the agent with the greatest computational power is the most likely one to append the next successful transaction. But in reality, there is no correlation between computational power and behaving honestly. Indeed, it has been evidenced that agents with high computational power can still misbehave and re-order transactions for their own personal gain, even if this is at the expense of other participants in the network [145].

Researchers have found that in reality, trust assumptions, like other design choices and properties in a DLT, are not binary but rather a spectrum [415]. Selecting an appropriate compromise and making assumptions that are grounded in reality are open and ongoing challenges. How does one ensure that chosen trust assumptions do not de-facto re-introduce centralisation into the system? One must also be wary of making unrealistic assumptions that will not hold in reality, as these may severely impact the security and reliability of the proposed solution [37]. It is not possible to avoid trust assumptions altogether, and these need not be unrealistically adversarial either. However, it is imperative that the safety of

a protocol does not hinge on a trust assumption that in practise will not hold.

1.3 Research Objectives

1. Designing decentralised systems that are reliable and resilient to attack without introducing unrealistic assumptions of good behaviour or de-facto re-introducing centralisation.
2. Creating peer-to-peer protocols that allow for the fair exchange of value for reward.
3. Uncovering the vulnerabilities of DLT apps and proposing frameworks and to evaluate and mitigate them.
4. Providing decentralised solutions that are compatible with protecting the users' privacy whilst ensuring they also cannot exploit the solution for personal and unfair gain.

1.4 Contributions

1. A decentralised, peer-to-peer architecture to transact digital assets that is fair, efficient and resilient against double selling, Sybil attacks, data poisoning and wormhole attacks¹⁷.
2. A class of verification algorithms that is peer-to-peer, privacy preserving, and that may also be used as a consensus algorithms. They do not rely on assumptions of honest behaviour to operate, and provide mathematical guarantees of security and reliability that can be tailored to the user's needs.
3. A comprehensive mathematical model of said class of algorithms, both for a location verification use case and for the generalisable class of consensus algorithms.
4. Open-source implementations of the algorithm used as a location verification algorithm, then the generalised version that can be used as a consensus mechanism, and the mathematical modelling, and the agent-based simulator for both.
5. A privacy preserving implementation of the location verification algorithm, using fully homomorphic encryption.
6. A comprehensive study of governance related vulnerabilities of both DAOs and DLTs.
7. A survey of necessary properties to prevent said vulnerabilities, including formalisations of said properties and possible implementations that deliver them, and a study of which of them are currently mutually exclusive.

¹⁷A wormhole attack involves an agent claiming to be in one location, but in reality being elsewhere [248].

1.4.1 Publications

Journal Publications

A. Manzano Kharman, C. Jursitzky, Q. Zhou, P. Ferraro, J. Marecek, P. Pinson, and R. Shorten, “An adversarially robust data-market for spatial, crowd-sourced data,” *ACM Distributed Ledger Technologies: Research and Practice*, Just Accepted, Nov. 2024, doi:[10.1145/3703464](https://doi.org/10.1145/3703464). [280]

A. Manzano Kharman, P. Ferraro, H. Hamedmoghadam, and R. Shorten, “Tree Proof-of-Position Algorithms,” *IEEE Internet of Things Journal*, vol. 12, no. 11, pp. 16393–16409, Jun. 2025, doi:[10.1109/JIOT.2025.3532161](https://doi.org/10.1109/JIOT.2025.3532161). [344]

A. Manzano Kharman and W. Sanders, “Vulnerabilities that arise from poor governance in Distributed Ledger Technologies,” *ACM Digital Government: Research and Practice*, Just Accepted, May 2025, doi:[10.1145/3731566](https://doi.org/10.1145/3731566). [343]

Conference Publications

A. M. Kharman, P. Ferraro, A. Quinn and R. Shorten, "Robust Decentralised Proof-of-Position Algorithms for Smart City Applications," 2023 62nd IEEE Conference on Decision and Control (CDC), Singapore, Singapore, 2023, pp. 112-119, doi:[10.1109/CDC49753.2023.10384054](https://doi.org/10.1109/CDC49753.2023.10384054). [281]

A. M. Kharman and B. Smyth, "DAO Governance Protocols and their Vulnerabilities," 2024 6th Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS), Berlin, Germany, 2024, pp. 1-2, doi:[10.1109/BRAINS63024.2024.10732536](https://doi.org/10.1109/BRAINS63024.2024.10732536). [276]

A. Gangemi and A. M. Kharman, "Private Dispute Resolution on Ethereum," 2024 IEEE International Conference on Blockchain (Blockchain), Copenhagen, Denmark, 2024, pp. 183-192, doi:[10.1109/Blockchain62396.2024.00032](https://doi.org/10.1109/Blockchain62396.2024.00032). [206]

1.5 Structure

The thesis is structured as follows: Chapter 2 presents preliminaries to better understand the security underpinnings and various architectures of DLTs. Chapter 3 outlines a proposal for a decentralised data market and Chapters 4 and 5 introduce a peer-to-peer algorithm to prove one’s position, necessary for the functioning of the data market described in Chapter 3. Chapter 6 later extends this algorithm to a generalised consensus algorithm, and Chapters 7 and 8 uncover governance vulnerabilities of DLTs and DAOs respectively. Finally, Chapter 9 concludes with a manifesto of the motivation and wider context of the work here presented.

Chapter 2

Preliminaries

Abstract

This thesis contributes to improving fairness in decentralised applications by building on Distributed Ledger Technologies (DLTs). To support this, the chapter introduces the essential background needed to understand DLTs at a high level. It outlines the cryptographic foundations that provide security, immutability, and consensus, and reviews the main design choices of ledger systems, including their architectures, consensus algorithms, and transaction record models.

2.1 The Crypto in Crypto(currencies)

Before surveying DLT architectures and implementation details, it is important to understand the cryptographic primitives that enable DLTs to work. Indeed, cryptocurrencies get the first part of their name from the field of cryptography, which underpins their functioning and delivers security, immutability and consensus. The goal of this section is to make the crypto in cryptocurrencies accessible to a reader without a formal mathematical background. This section is by no means a comprehensive overview of all cryptographic protocols, or an in depth explanation of how each presented protocol operates. A number of thorough resources exist for the curious reader to follow up any that may be of interest. Rather, this section is intended to give reader enough understanding such that in subsequent chapters, when reference to encryption protocols and other cryptographic tools is made (explicitly or implicitly), it can be understood what these are, what purpose they serve and why they are necessary.

Before proceeding, I would like to acknowledge and extend my gratitude to the following academics that have let me join their courses remotely or have made their teaching material publicly available online: Professor Christoph Paar’s course on Introduction to Cryptography¹ [391], Professor Michael Huth’s course on Cryptography Engineering², Professor

¹Course available [online](#).

²This course was completed through Imperial College and not publicly available to the best of my knowledge.

Bill Buchannan’s online resources [99], and Professors Dan Boneh, Shafi Goldwasser, and Dawn Song’s Zero Knowledge Proofs Course [89]. Their resources, along with foundational educational material such as [352, 460, 461, 273] have proved invaluable. The following section is entirely indebted to them.

2.1.1 Basics: What Is Encryption?

Cryptography is the science of sharing communication privately, in the presence of adversaries [273, p. 7]. Not all information should be public: cryptography enables selective disclosure. In the most simple scenario, a user A wishes to send another user B, a message, but would like this message to remain secret, and only reveal its contents to its intended recipient. Cryptographers usually call the sender Alice, the receiver Bob and the message, m .³ The un-encrypted message is called a plaintext, the encrypted message a ciphertext, and key is a value used to perform encryption and decryption. Let us adopt this convention for the rest of this work.

If Alice and Bob are mutually trusting and face no threat of eavesdropping or attack, they may simply exchange messages freely without any concern. Unfortunately, reality is rarely so kind, as communication channels are often vulnerable to interception, manipulation, or surveillance by adversaries. Further, Alice may not trust Bob, and she may not have guarantees that the person she is communicating with, is indeed Bob. To make the messages secret, Alice performs an encryption operation, and Bob will undo this operation through decryption. Both these operations require a key. In some cases, the same key is used to perform the encryption and the decryption. This class of encryption algorithms are called symmetric-key encryption algorithms, as seen in Figure 2.2 [352, p. 191]. In others, the key used to encrypt is different to the key used to decrypt the message. These are the public-key cryptography algorithms, as shown in Figure 2.1 [352, p. 283].

Very early versions of cryptography relied on the encryption algorithm being secret to ensure communications remained secure. This is no longer the case in modern cryptography. Kerckhoffs’s principle states that a cryptographic algorithm should remain secure even if everything about the algorithm itself, except the key⁴, is public knowledge [391, p. 7].

2.1.2 Key Generation

To encrypt any message, it is first necessary to possess a key. It is vital that this key remains secret to ensure communications are secure. Let us imagine Alice encrypts a message with the most powerful and secure encryption scheme known to man. If an attacker obtains her key, it does not matter how secure her encryption algorithm is, her communications *will* be compromised. One way an adversary could compromise communications is by simply guessing her key. For example, they may try all possible values the key may take. This is called the brute force attack, and it is avoided by ensuring the possible number of values the key can take is so large that this attack is computationally infeasible [391, p. 7].

³For a fascinating biography of Alice and Bob’s (fictional!) lives, the reader may refer to this [speech](#).

⁴In the case of public-key cryptography, the *private* key must remain secret, not the public key.

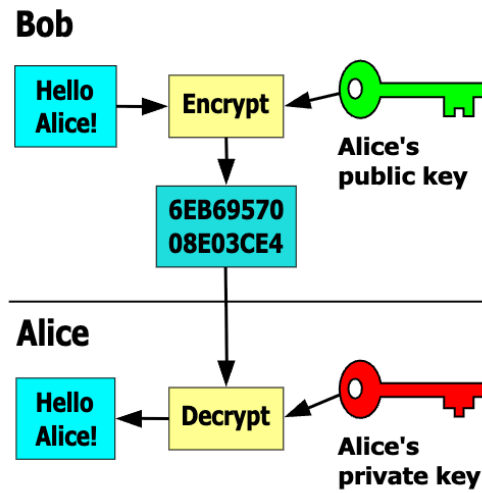


Figure 2.1: Public Key Cryptography, by David Gothberg - Public Domain, [source: WikiMedia](#).

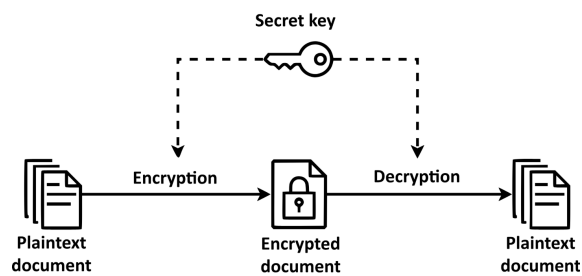


Figure 2.2: Symmetric Key Cryptography, by Michel Bakni - CC BY-SA 4.0, [source: WikiMedia](#).

To ensure the selected key cannot be guessed, the key should be entirely unpredictable [352, p. 169]. Random number generators (RNGs) are leveraged to create a string of numbers that is unpredictable, and that can therefore be used as a key [250]. True randomness is not guessable, and thus the highest standard to aim for when choosing secret keys. True Random Number Generators (TRNGs) generate numbers by measuring random physical processes [16]. For example, some hardware RNGs measure electrical noise in resistors or semiconductors (Intel’s RDRAND and RDSEED) [135, 453], others have historically used radioactive decay to generate random bits (HotBits) [524]. Alternative examples include generating random numbers from radio static or from quantum optical noise (Quantis RNG) [231].

Another class of RNGs are Pseudo-Random Number Generators (PRNGs). Unlike True RNGs, their output may *look* random, but is not [50]. In fact, researchers discovered that as long as this random looking output is not actually distinguishable from true randomness, it provides the same level of security as a truly random key. PRNGs generate a string of numbers following a deterministic algorithm that depends on the starting point, called a *seed*. If the seed and the algorithm are known, the output can be replicated, so it is important that the seed remains private. However, this does not exclude them from being useable in cryptographic applications. Seeds are often generated by measuring high entropy environment variables, which is another way of saying difficult to predict environmental variables. Cryptographically Secure Pseudo-Random Number Generators (CSPRNGs) are a class of PRNGs that meet certain security standards. These standards vary in their precise definitions, but in essence they capture the three following properties:

1. The outputs of the cryptographic pseudo-random number generator (CPRNG) are indistinguishable from random outputs to any efficient adversary [544].
2. They provide backwards security (backtracking resistance), meaning it is not possible for an adversary to recover previous states or outputs if the CPRNG is compromised in its current state. Barak and Halevi, and the National Institute of Standards and Technology (NIST) offer definitions for this property in [47] and [50] respectively.
3. The CPRNG is unpredictable. In other words, there is no polynomial-time algorithm that can predict future outputs non-negligibly better than a random guess. This definition is strengthened by also requiring prediction resistance: future outputs remain unpredictable even if a prior internal state of the CPRNG is compromised [47], [50].

These are theoretical frameworks that give us a means to test a pseudo random number generator to check if it is cryptographically secure. However, for the practical user that just wants to generate secure keys, how should they go about doing so? Typically, cryptographic libraries available in modern programming languages provide access to secure random number generators maintained by the operating system. These generators collect unpredictable data, known as *entropy*, from various physical and software sources within the computer, such as mouse movements, keystrokes or timing other hardware operations.

Plaintext	Key	Ciphertext
0	0	0
0	1	1
1	0	1
1	1	0

Table 2.1: XOR operation

This entropy is then used to produce random numbers suitable for generating cryptographic keys [501].⁵ By using these well-established tools, the generated keys will have sufficient entropy to make them computationally infeasible for an attacker to guess or compute.

In summary: predictable keys are dangerous. They allow the attacker to compromise communications without attacking the encryption algorithm itself. Random keys is the gold standard. In the absence of true randomness, you may still generate secure keys as long as they meet certain security standards.

In reality, although practical and theoretical frameworks exist to manage key generation, there is an important human factor in key generation and management. Some DLT projects are compromised because people are not always very good custodians of their private keys, or use guessable keys [254, 226]. This is especially relevant in scenarios where custodial wallets are used. The latter refers to an alternative method of key storage in which a third party is trusted to manage the private keys to your funds, which are often used in exchanges. Many projects have been subject to a varying attacks involving key exploits: phishing attacks to access passwords of wallets⁶, injection of malicious code to harvest private keys⁷, or attacks leveraging the fact that predictable keys were used⁸[515].

In subsequent chapters, encryption schemes are necessary to secure communication, and it is assumed that participants are able to securely generate and manage their own keys.

2.1.3 Symmetric Encryption

As described earlier, symmetric encryption algorithms use the same key to encrypt and decrypt a message. The simplest form of encryption algorithms are **stream ciphers**. Each bit in the message is encrypted with a bit from the key using an encryption operation. Generally, for a message $m = m_1m_2m_3...m_n$ and a key k , where $k = k_1k_2...k_n$, first we ensure the key and the message are of equal length [273, p. 194]. The encryption operation $E(m, k)$ can then be applied to each element in the message, such that the ciphertext $c = c_1c_2...c_n$, where $c_i = E(m_i, k_i) \forall i \in n$. A simple example for an encryption operation is the XOR operator, shown in Table 2.1. Whilst this encryption mechanism is incredibly simple, it offers highly desirable properties that combined with certain modifications, make

⁵For example, Cloudflare—one of the largest service providers in web infrastructure, security, and content delivery—uses randomness from a wall of lava lamps in their headquarters to generate their random numbers [320].

⁶CoinCheck Phishing Attack.

⁷IOTA Trinity Wallet Hack.

⁸Password Cracker Attack.

Plaintext	Key	Ciphertext
0	0	0
0	1	0
1	0	0
1	1	1

Table 2.2: AND operation

it unbreakable.⁹ The first important property is that the ciphertext output is statistically balanced, meaning that if we assume the key bits are truly random (i.e.: could be 0 or 1 with equal likelihood), then each ciphertext bit has a 50% chance of being 0 or 1—making it impossible to infer any information about the plaintext from the ciphertext alone [461, p. 164].

The AND, OR, NAND or other logic operators do not offer the same guarantees. Let us inspect the AND gate for example, shown in Table 2.2. An attacker can infer information by inspecting the ciphertext because the ciphertext output of 0 is more likely to appear than the output of 1. There is a conditional probability dependency: if the ciphertext is 1, the plaintext input must have been 1, and in fact, the key input must also have been a 1. Similarly, if the ciphertext output is 0, it is more likely that the plaintext input was 0 as opposed to 1.¹⁰ Attackers can easily exploit this property to derive the plaintext, and even the key [460, p. 41].

Another important property of the XOR function is that it is invertible. Applying it to the ciphertext with the same key will yield the original plaintext. A number of stream cipher algorithms exist, which vary in precise implementation details, but mostly use the XOR function as the encryption mechanism.

In practice, stream cipher implementations use a simple encryption operation, typically the XOR function, and focus primarily on how to securely generate the pseudorandom bitstream (the key) that is combined with the plaintext [352, p. 191]. It is of utmost importance that the key is generated from at least a Cryptographically Secure Pseudo-Random Number Generator, or at best a True Random Number Generator, and that it is of the same length as the message for this cipher to be secure [391, p. 40].

There are certain appeals of using stream ciphers: they have lower complexity requirements and are thus easier to implement, making them an attractive option for hardware constrained applications [391, p. 49]. They are also fast and provided care is taken with key generation, they are very secure [273, p. 32]. Their drawbacks are the impractical key length, and the constraints they introduce with key management and generation [460, p. 141]. They are found in wallet applications of DLTs, especially for secure key storage, and in general, are used alongside with asymmetric encryption to securely exchange

⁹If the key used is kept secret, is truly random, never reused and the same length as the message, it has been shown that the XOR cipher (in this case called the One Time Pad) is unbreakable [449].

¹⁰That is, assuming the distribution of plaintexts is uniform, which may not be the case for natural language.

messages.¹¹

Another class of symmetric encryption algorithms are **block ciphers**. These also perform the encryption and decryption operation with the same key, but perform the encryption on a sequence of bits at a time, as opposed to bit by bit [460, p. 123]. These are more practical than stream ciphers due to reduced key lengths. This class of algorithms has been well established since approximately the 1970s, and extensively studied [391, pp. 56–57]. They are built on implementing theoretical notions outlined by Claude Shannon on *confusion* and *diffusion*. Confusion is the property that makes the relationship between the ciphertext and the key obscured [448]. The property of diffusion means the impact that an element in the plaintext has on the ciphertext outcome is spread across the ciphertext [448]. In other words, changing one input in the plaintext should affect more than one output on the ciphertext, and ideally most or all of them. These algorithms often operate by performing a number of rounds, where each round acts on a block of data. Where one round may provide diffusion, or another may provide confusion, chaining multiple of these rounds subsequently, delivers security by achieving both properties.

Most practical block cipher designs follow one of two main structural paradigms: Feistel networks [273, p. 211] or substitution–permutation networks (SPNs) [273, p. 204]. In addition, a large family of modern ciphers, including block and stream ciphers, are built using the Addition–Rotation–XOR (ARX) design style [78].

Feistel networks are a type of block cipher design where the input block is split into two halves. In each round, one half is processed through a function (called a round function) that combines the data with part of the encryption key (also called a subkey)¹², typically using operations like substitution and XOR. The result is then combined with the other half, and the two halves swap places before the next round [461, p. 244]. This structure is easy to reverse for decryption, using the same steps with the subkey order reversed. Well-known examples include DES, Blowfish, and Camellia [391, p. 81].

Substitution–Permutation Networks (SPNs), used in ciphers like AES, process the entire data block at once in each round [273, p. 204]. Each round includes a substitution step (replacing small parts of data using fixed lookup tables called S-boxes) followed by a permutation step (reordering bits or bytes to spread the effect of substitutions). The substitution rounds deliver confusion because it breaks linearity patterns between the plaintext and ciphertext, and the permutation rounds shuffle bits or shift rows and columns of the input. These rounds are designed to deliver diffusion by ensuring that a single input bit change affects many output bits over multiple rounds [404]. SPNs are efficient on modern hardware and offer strong diffusion and nonlinearity [391, p. 115].

¹¹For example, Trezor hardware wallets use ChaCha20-Poly1305 to protect both user communication and sensitive key material. [Source: Trezor Wallet Documentation](#).

¹²Subkeys for each round are derived from the main encryption key through a process called the key schedule. This is a function that takes a main key, and from it will generate a series of keys, typically smaller in size than the main key, that are each applied to their corresponding round. To decrypt, the order of subkeys is simply reversed and applied to the same Feistel function rounds, that are also reversed in order.

ARX-based designs rely exclusively on three simple operations: modular addition, bitwise rotation, and XOR operation. Unlike SPNs, they do not use S-boxes, which makes them easier to implement securely in software, especially on constrained or low-power devices [78]. ARX operations execute in constant time on modern CPUs, giving them inherent resistance to timing attacks¹³ [78]. Block cipher examples include Speck (a Feistel–ARX design) [57] and Threefish (an SPN–ARX design) [188], while ChaCha [78] is an ARX-based stream cipher. These designs have become popular in lightweight cryptography due to their efficiency and side-channel attack robustness [57, 188].

Symmetric encryption schemes are presented because although they are not explicitly discussed in the novel contributions of this thesis, along with asymmetric encryption, provide the basis of secure communication between nodes. For any decentralised system where nodes are communicating with one another in an adversarial environment and are mutually distrusting, encryption is necessary to deliver properties of message integrity, confidentiality and authenticity of the information being exchanged. Usually, key exchange between nodes is performed through asymmetric encryption protocols, which are outlined in the next section, and once a secure handshake has been established, message exchanges proceed through symmetric encryption [161]. Whilst this is not explicitly stated in subsequent chapters, it is assumed that nodes securely communicate through standard protocols. This is relevant in every contribution of this work: in data market communication exchanges, in consensus algorithms exchange of messages amongst participants, and in DLT and DAO governance for users to express votes. Without encryption, necessary properties in these applications proposals fall through.

2.1.4 Public-Key Cryptography

Public-key cryptography, also known as asymmetric encryption, operates differently to symmetric encryption. For each user, both of these keys are linked: there is a value in Alice’s public key that is derived from her private key, and the same holds for Bob [461, p. 202]. Due to the mathematical construction that these keys are built from, it is not possible to derive the private key from the public key value related to it [391, p. 206].

There are multiple functionalities that can be achieved with public key cryptography: key exchange, confidentiality and digital signatures [391, p. 169]. In this work, the one of most relevance is digital signatures. These enable Alice to send a message to Bob, and for Bob to assert that the message is indeed from Alice. If the reader may recall, in symmetric encryption, both Alice and Bob share the same key, so it is not possible to prove which of the two actually constructed the message. Some situations require that if Alice creates a message, she cannot later deny it was hers. This property is called non-repudiation [391, p. 151]. In DLTs this is critical: if Alice sends a message in which she states she wishes to transfer some funds to Bob, she should not be able to later reclaim the funds by arguing she did not send that message. Digital signatures deliver this property of non-

¹³This is an attack where the attacker is able to obtain information by analysing how long it takes for the computer or microcontroller device to perform certain operations [291].

repudiation, and also provide integrity (Alice’s message was not modified in transit) and message authentication (the message is truly from Alice) [460, pp. 221–224].

Key exchange protocols enable users to establish a common key despite communicating over an insecure channel [460, p. 219], and confidentiality pertains to ensuring messages remain secret and only disclosing them to their intended recipient [352, p. 12]. We have already seen algorithms for confidentiality in symmetric encryption, and whilst key exchange protocols will not be further discussed, the reader can find resources on the topic in [162], [391, p. 242], [460, pp. 219–231] and [273, pp. 359–372]. The latter three also contain extensive explanation of public-key encryption schemes covering the topic of confidentiality.

The way digital signatures are used in DLT transactions is as follows: Alice wishes to send funds to Bob. These funds are locked to a certain address, which is a value that is derived from a user’s public keys. To spend these funds, Alice must prove she has the secret key that corresponds to the address (constructed from the public key) that these funds are locked to. To prove this without revealing the private key, she must generate a valid digital signature. When she creates the transaction to send funds to Bob, everyone can check that her signature maps to the address (public key) of the spendable funds [374]. Thanks to cryptographic constructions, her signature will only be valid if she truly holds the private key that corresponds to that public key.

There are numerous digital signature algorithms available [352, p. 425]. DLTs mostly use the Elliptic Curve Digital Signature Algorithm (ECDSA). These deliver same levels of security with shorter key lengths compared to their DSA and RSA digital signature counterparts [391, p. 288]. ECDSA also offers smaller signature sizes compared to RSA, and although RSA’s verification operation is performed faster, its signing is slower.¹⁴ Hence in DLT applications, elliptic curve based digital signature algorithms are the most prevalent.

The following section provides a summary of how digital signatures are constructed from elliptic curve cryptography, without assuming significant prior knowledge. If the reader is already experienced with elliptic curve cryptography, they may wish to skip this section.

The Discrete Logarithm Problem

The security of public-key encryption algorithms is underpinned by the difficulty of efficiently solving a mathematical problem.¹⁵ The difficult-to-solve problem that secures elliptic curve cryptography algorithms is a variant of the **discrete logarithm problem** (DLP) [460, p. 23]. The discrete logarithm problem was first applied in the Diffie–Hellman key exchange protocol [162], and is described in detail in [352, p. 103] and [391, p. 252]. In very simple terms, the problem states that it is hard to find an exponent x , where y and g are given, such that $g^x = y$. In other words, computing the *discrete* logarithm $x = \log_g y$

¹⁴Source: SSL.

¹⁵The two most widely studied problems are the integer factorisation problem and the discrete logarithm problem. The former underpins RSA [352, p. 89], while the latter is fundamental to systems such as Diffie–Hellman and elliptic curve cryptography [352, p. 113].

is computationally difficult.¹⁶ What does ‘hard’ mean? It means there is no computationally efficient algorithm that exists to solve this problem.¹⁷ That is, no algorithm with polynomial-time complexity is known. Evidently, this problem is trivial to solve for small numbers, but not so when g is the generator of a finite cyclic group (with a multiplicative operator), $\mathbb{Z}_p^*$, of order p , where p is a large prime. For an overview of what a group, finite cyclic group, group operator and generator is, the reader may find these definitions in [352, p. 75]. In this case, finding an integer $x \in \{1, 2, \dots, p-1\}$ such that $g^x \equiv y \pmod{p}$, when g , p , and y are known, is computationally difficult. Computing $y = g^x \pmod{p}$ is easy, but recovering x from g and y is believed to be computationally hard. This asymmetry is the basis for the security, as it acts as a one way function [391, p. 217].

Elliptic Curve Cryptography

Having discussed the discrete logarithm problem, we can proceed to introduce the notion of elliptic curves. These are curves in the form of $y^2 = x^3 + ax + b$ [460, p. 23]. The (x, y) points that satisfy the curve equation form a group. We can define a group operation on the group points derived from the elliptic curve, which is point addition. Then from this operation, we can construct the Elliptic Curve Discrete Logarithm Problem (ECDLP), invented by Koblitz [290] and Miller [356] independently. This problem asks, given a (generating) point P on a curve, and another point T , such that $T = d \circ P$, meaning that the group operation (denoted $\circ$) is performed d times on point P , finding the integer d is hard. $1 \leq d \leq \#E$ where $\#E$ is the number of points on the curve [391, p. 281]. The group operation is adding the points in the elliptic curve, where this addition operation is slightly peculiar, although it has a geometric interpretation that is explained in [391, p. 281]. The order of the group (i.e. the size, which is $\#E$ in this case) must be known to set up the Elliptic Curve Discrete Logarithm Problem. Although finding the exact number of points on the elliptic curve is difficult, there are theorems that exist that provide a bound to this number [240].

In practical applications of distributed ledgers, numerous digital signature algorithms are used that are built on Koblitz elliptic curve types. ECDSA (Elliptic Curve Digital Signature Algorithm) is used in Bitcoin¹⁸ and previously in Ethereum [347]. It is standardised by NIST for use over elliptic curve groups $E(\mathbb{F}_p)$, most commonly with NIST P-curves such as P-256 [376]. Other curves, such as `secp256k1` defined by the Standards for Efficient Cryptography Group, are also widely used in practice, for example in Bitcoin. The signing process involves the signer choosing a random number that is only used once (called a nonce) [460, p. 143]. The nonce is a different parameter to the private key. This scheme is vulnerable to nonce reuse; if the same nonce is used for two different messages, the private key of the signer can be recovered [376].

EdDSA (Edwards-curve Digital Signature Algorithm), is another scheme that improves

¹⁶Not to be confused with the *continuous* logarithm problem, that is efficiently solvable with approximations and numerical methods.

¹⁷That is significantly better than brute force.

¹⁸Source: [Learn Me A Bitcoin](#).

upon ECDSA by using deterministic nonces and a different curve. The most prominent instantiation, `Ed25519`, is based on a twisted Edwards curve defined over the finite field $\mathbb{F}_{2^{255}-19}$. The signing process is deterministic: the nonce is derived from the message and private key using a cryptographic hash function¹⁹ [266] (typically SHA-512). This mitigates vulnerabilities related to poor randomness in nonce generation. EdDSA signatures are faster to compute and verify than ECDSA and are more resistant to side-channel attacks [266]. Example applications using them include Cardano²⁰ and Zcash²¹.

Another alternative is Schnorr signatures which are defined over the same elliptic curve group as ECDSA but use a simpler design [460, p. 228]. They are based on linear equations whereas ECDSA and EdDSA have a more complex structure involving modular inversion and non-linear steps. They are non-malleable²² by design and are provably secure in the random oracle model [105] under the discrete logarithm assumption.²³ By contrast, ECDSA’s security proofs rely on multiple interdependent assumptions, and without adequate protections, ECDSA signatures can be malleable [97]. Schnorr’s useful advantage is that it enables signature aggregation and multi-signature schemes, by which multiple parties can jointly produce a single compact signature [461, p. 338]. This property is not available in ECDSA or EdDSA. As of Bitcoin’s Taproot upgrade (BIP-340), Schnorr is officially supported, replacing ECDSA in certain contexts.²⁴

BLS signatures, named after their inventors Boneh, Lynn, and Shacham, are constructed using bilinear pairings [353, p. 3] over elliptic curve groups—most commonly the BLS12-381 curve [88]. They support efficient signature aggregation, allowing multiple signatures on different messages from different signers to be combined into a single compact signature. This enables not only threshold signatures—where multiple parties collaboratively generate a valid signature without reconstructing the private key—but also batch verification, which verifies multiple signatures simultaneously rather than individually. BLS signatures are shorter than many alternative schemes, but they require pairing operations, which are computationally intensive and depend on larger curve parameters. This makes them less suitable for resource-constrained environments. Nonetheless, they are used in projects such

¹⁹See Subsection 2.1.5 for an overview on hash functions.

²⁰Source: Cardano.

²¹Source: Zcash Improvement Protocol.

²²The property of malleability in the context of cryptography refers to: the ability to transform a ciphertext into another ciphertext, that when decrypted, results in a plaintext that is related to the original, unmodified one. Generally, it is undesirable that encryption schemes deliver malleable ciphertexts, as this can threaten message integrity, although as we will see later, some encryption schemes are malleable by design.

²³The sentence *provably secure in the random oracle model* is a mouthful. What does that mean? A scheme is provably secure if you can show that breaking it would require solving a well-known problem that is believed to be hard—like integer factorisation. Expanding this to provably secure *in the random oracle model*: the random oracle model is a mathematical model that assumes the existence of an “oracle”—in other words, a function you can query that returns a truly random output for each unique input, and always gives the same output if you repeat that input. If such an oracle exists and can be queried, then the scheme can be proven secure within this hypothetical model. This sounds abstract, and it is: in practice, hash functions are used in place of the ideal random oracle. While hash functions are not truly random oracles, they are treated as such when applying the security proof.

²⁴Source: Bitcoin’s Taproot upgrade (BIP-340).

as Ethereum 2.0²⁵, Dfinity²⁶, Skale²⁷, and the Chia Network²⁸.

A Remark on Digital Signature Algorithms in DLTs

There is a subtlety that must be highlighted regarding the use of digital signatures. A digital signature scheme allows a verifier to check that a signature was generated using the private key corresponding to a given public key. It does not, however, verify the actual identity of the signer. For example, Alice may receive a valid signature, but this only proves that the signer knows the private key corresponding to a certain public key—it does not guarantee that Bob was the one who signed. Without additional safeguards, a malicious agent could mount a man-in-the-middle attack by intercepting communications and substituting Bob’s legitimate public key with their own [391, p. 290], thereby deceiving Alice into accepting forged signatures. To prevent this, a certificate authority (CA) is employed, which binds a public key to a user’s identity by issuing a digital certificate. Therefore, if Bob provides a valid signature along with his certificate issued by a trusted CA, Alice can verify that (i) the message was signed using the private key corresponding to Bob’s certified public key, and (ii) this public key is indeed bound to Bob’s identity.

Blockchain protocols do not require certificates to send and receive transactions. The burden of verifying who funds are sent to, rests on the user. Bitcoin introduced support for an interactive payment protocol, outlined in BIP-70, in which a payee could send a signed message that is a payment request. This signed message contains the payee’s certificate, which asserts to the payer that the user is who they claim to be. However, this protocol saw little adoption due to its reliance on external dependencies like OpenSSL and limited wallet support compared to simpler alternatives such as BIP-21.²⁹ Current models embrace a rationale of ‘payment identifiers, not person identifiers’³⁰, and identity verification is offloaded to other external solutions unless it is a permissioned DLT. This lack of identity verification may be considered a design flaw, or a design principle. Given the prevalence of illicit purchases that cryptocurrencies have been used for [124], it is not unfathomable that it is, in fact, a design choice.

2.1.5 Hash Functions

Hash functions are another fundamental cryptographic tool used in Distributed Ledger Technologies. They are deterministic³¹ functions mapping from arbitrary-length input strings to fixed-length output: $H : \{0, 1\}^* \rightarrow \{0, 1\}^n$ [352, p. 321].

These properties include: *pre-image resistance*, *second pre-image resistance*, and *collision resistance* [352, p. 323]. Let us denote x as the input of the hash function, and y as the output of the hash function, such that $y = H(x)$. Pre-image resistance states that

²⁵Source: [Ethereum Foundation](#).

²⁶Source: [Dfinity GitHub Repository](#).

²⁷Source: [Skale documentation](#).

²⁸Source: [Chia Network GitHub Repository](#).

²⁹Source: [Bitcoin Optech](#).

³⁰Source: [BIP-21](#).

³¹Meaning the same input always results in the same output.

given y , it is computationally infeasible to find any x such that $H(x) = y$. In other words, this property means given an output, it is hard to find the input that produced it. Second pre-image resistance is defined as: given x , it is hard to find $x' \neq x$ such that $H(x') = H(x)$. Collision resistance means it is computationally infeasible to find any distinct pair of inputs x, x' such that $H(x) = H(x')$. There is a distinction between second-preimage resistance and collision resistance: the former requires the adversary to find another valid message, x' , such that it will hash to the same result of $H(x)$. Collision resistance asks the adversary to find *any* pair of valid messages that have the same hash output. Also following from Shannon’s concept of *diffusion*, hash functions should exhibit an ‘avalanche effect’ [185, p. 22]. This is an informal term to denote the property that a small change in the input should significantly affect the output. These properties make hash functions what is informally referred to as one-way functions, meaning it is hard to invert the output to find the original input, but computing the output for a given input is easy.

These properties make hash functions a very useful tool in distributed ledger applications. Arguably, their most famous use is in underpinning the consensus algorithm of many blockchains such as Bitcoin. This consensus algorithm is commonly referred to as ‘proof-of-work’. During this process, a node in blockchain performs an inverting operation of finding a valid input that hashes to a valid range within the provided output. Further detail on consensus algorithms is provided in Section 2.2.2. Hash functions are also used to derive address values for wallets from public keys, and to sign messages in digital signature algorithms. Furthermore, because they map data inputs of arbitrarily large size to a fixed, smaller size output, they are useful to efficiently summarise large amounts of transaction data. Combined with their previously mentioned properties, it is also possible to check that this summarised data has not been modified or tampered with, thus providing data integrity.

An important feature of hash functions is that the difficulty to invert them remains high in quantum computation models [23]. Further discussion on quantum computing and its implications for Distributed Ledger Technologies is provided in Section 2.1.8. In it, Grover’s algorithm is discussed, that provides a quadratic speed-up for unstructured search problems such as hash inversion [230]. For now, the reader should note that: if quantum computation is a generalised and widespread reality, hash functions will, in theory, be easier to solve than in classical computation models, but not trivially easy to solve.

2.1.6 Zero Knowledge Proofs

Zero-knowledge proofs (ZKPs) are cryptographic protocols first introduced in 1985 by Goldwasser, Micali, and Rackoff [220], and have since found powerful applications in Distributed Ledger Technologies. These kinds of protocols involve an exchange between two parties: the prover and the verifier. The prover has a secret, and they wish to demonstrate to the verifier that they are in possession of this information, without having to disclose it to the verifier. Through this protocol, the verifier only learns whether the prover is telling

the truth.

At a high level, a zero-knowledge proof involves a structured exchange of information between the prover and the verifier, often in the form of a multi-step interaction or a single proof message, depending on the type of protocol. The prover begins with a secret—such as a password, a solution to a mathematical problem, or the preimage of a hash—and a corresponding statement they wish to prove true. The verifier, meanwhile, knows the statement but not the secret. The goal is for the prover to demonstrate that the statement is valid because they know the secret, without revealing any information about the secret itself [460, p. 371]. This is achieved by encoding the statement and secret into a mathematical form, and then generating a proof that satisfies three key properties: *completeness* (a truthful prover can convince the verifier), *soundness* (a dishonest prover cannot cheat), and *zero-knowledge* (no information about the secret leaks) [168]. Depending on the protocol, this interaction may involve challenges, commitments, and responses, or it may be condensed into a single succinct proof that can be verified efficiently.

Since their inception, they received little practical uptake until what Eli Ben Sasson terms: ‘*the Cambrian explosion of zero-knowledge proofs*’³² in 2018. Currently, they are heralded as the solution to the scalability concerns of DLTs, whilst also offering privacy by design [322]. They do so by enabling DLT transactions to be grouped and processed ‘off chain’, simply meaning, not on the blockchain network [491]. A user then constructs a zero-knowledge proof attesting that a large batch of transactions was processed correctly according to the protocol rules. This compact proof, which is significantly smaller than the data it represents, is then posted to the blockchain as a single transaction [168].

Zero-knowledge proof (ZKP) systems have become central to the scalability and privacy layers of modern Distributed Ledger Technologies, with multiple constructions optimised for different trade-offs in proof size, verification efficiency, cryptographic assumptions, and trust requirements [482]. Among the most widely studied and deployed are Bulletproofs, zk-SNARKs (zero-knowledge Succinct Non-interactive Argument of Knowledge) and zk-STARKs (zero-knowledge Scalable Transparent Argument of Knowledge). Each offers distinct performance characteristics and thus have varying adoption across DLT ecosystems.

Bulletproofs are non-interactive³³ zero-knowledge arguments of knowledge constructed from the discrete logarithm assumption in prime-order groups (typically instantiated over elliptic curves). They use an inner-product argument and are made non-interactive via the Fiat–Shamir heuristic [101, 189]. In an inner-product argument, the prover can demonstrate that they possess two secret vectors that combine to a claimed result. The result is disclosed, while the vectors remain hidden [101]. Bulletproofs are transparent—that is, they require no trusted setup—and yield range proofs of logarithmic size, which makes them highly efficient for confidential range proofs. However, their verification remains linear in

³²Professor Eli Ben Sasson’s lecture on zk-STARKs (a class of zero knowledge proof protocols) p.7.

³³An interactive proof protocol between a prover and a verifier can be made non-interactive if the verifier’s messages are nothing more than random challenges that the prover must answer. This transformation can be achieved using the Fiat–Shamir heuristic, which replaces the verifier’s random challenges with the output of a cryptographic hash function [189].

the circuit size, making them less competitive than succinct proof systems for large arithmetic circuits [101, 68]. Bulletproofs have seen real-world deployment in privacy-focused cryptocurrencies, most prominently in Monero, where their adoption in 2018 reduced average transaction sizes by over 80% and significantly lowered fees.³⁴

Generic zk-SNARKs (e.g., PLONK [202], Marlin [120], Sonic [338]) achieve succinct proofs and extremely fast verification, with sizes often in the order of a few hundred bytes [168]. These constructions rely on pairing-friendly³⁵ elliptic curves and their security is underpinned by the difficulty of knowledge-of-exponent assumption [83]. Many SNARKs still require a trusted setup [229, 338, 120], though newer universal and updatable ceremonies (e.g., PLONK [202]) mitigate the per-circuit setup burden. Their high efficiency explains widespread adoption in Ethereum-based implementations such as zkSync³⁶ and Polygon zkEVM³⁷, where succinctness directly reduces on-chain verification costs and enables economically viable scaling.

zk-STARKs have a proof size larger than zk-SNARKs (typically tens to hundreds of kilobytes vs. hundreds of bytes for SNARKs), but still significantly smaller than the raw computation being proven. The verification time for the statements scales quasi-linearly with the logarithm of the computation size, making verification efficient even for large statements [67, 482]. Their construction is based on probabilistic checks over polynomial commitments, and interactive oracle proofs (IOPs)³⁸ [67]. They avoid trusted setup by relying only on publicly verifiable randomness. The security of zk-STARKs relies on the hardness of collision-resistant hash functions (a very conservative and well-studied assumption), unlike SNARKs which rely on elliptic-curve pairings and knowledge-of-exponent assumptions. In other words, if an adversary could find collisions or otherwise break the binding property of the hash commitments, they might be able to forge a proof of an incorrect computation. In summary, their advantages include their properties of being i) transparent (no trusted setup requirement); ii) plausibly post-quantum secure (since their security is based on hash functions) and iii) scalable to very large computations. However, their main drawback is the larger proof size and verification times compared to the extremely succinct SNARK proofs.

When considered together, these proof systems illustrate the trade-offs between efficiency, transparency, and security assumptions. Bulletproofs and STARKs avoid trusted setups entirely, but at the cost of larger verification overhead or proof sizes [389]. SNARKs achieve significantly more succinctness and increased verifier efficiency, but require a trusted setup and their security relies on the hardness of pairing-based cryptographic assumptions. The latter security assumption is less rigorously scrutinised and established in practice than classical assumptions such as discrete logarithms [363, p. 13]. In cryptocurrencies, zero knowledge proofs are state of the art to deliver primarily scalability, and confidentiality as

³⁴Source: Monero.

³⁵See [353, p. 3] for an outline on pairing-based cryptography.

³⁶Source: Ethereum.

³⁷Source: Polygon.

³⁸For a clear, jargon-free explanation of what an IOP is, the reader may refer to the following [source](#).

a consequence. They are frequently implemented in ‘layer 2’³⁹ Some examples include Monero using Bulletproofs, Ethereum using zk-SNARKs and zk-STARKs, ZeroCash, Polygon Network, Mina Protocol and Aztec using zk-SNARKs [389, 482].

2.1.7 Homomorphic Encryption

Homomorphic encryption schemes are a class of public key encryption schemes that offer malleability by design [3]. This property means that a ciphertext can be transformed into another ciphertext, that when decrypted is related to the original plaintext. Generally this property is undesirable, however it can be useful in some scenarios, as it enables performing computations over encrypted data, and decrypting only the result. One relevant application is in the field of electronic voting [71]. This class of schemes allow encrypted votes to be tallied whilst they remain encrypted. The resulting ciphertext, when decrypted, reveals the sum of each plaintext vote.

For a given additive homomorphic encryption algorithm E , where the plaintext $m \in \mathbb{Z}_n$ there exists a ciphertext operation $\circ$ such that $E(m_1) \circ E(m_2) = E(m_1 + m_2 \pmod{n})$. Similarly, for a multiplicative homomorphic encryption scheme, there exists a ciphertext operation $\star$ such that $E(m_1) \star E(m_2) = E(m_1 \cdot m_2 \pmod{n})$.

Some schemes are partially homomorphic, meaning only a certain operation is supported under encryption. This can be either addition or multiplication of ciphertexts (modulo the scheme’s underlying group or ring parameter). Examples include Paillier [392], ElGamal [177] and Benaloh [71] cryptosystems. Another class of schemes are called somewhat homomorphic schemes: these enable both addition and multiplication operations, but limit the number of operations that can be applied. One example is the BGN encryption scheme [87], and earlier examples by Gentry [211]. Fully homomorphic schemes support both addition and multiplication operations on encrypted data, without a limit on the number of computations performed. The cost they pay is the computational overhead [3]. These often require an operation known as *bootstrapping* after a certain number of operations performed on the ciphertext. As the number of computation operations performed increases, the amount of noise in the ciphertext also increases. At a certain point, the noise can be so high that decryption of the ciphertext would not yield the original plaintext with sufficiently high probability. The operation of *bootstrapping* reduces the noise in the ciphertext by homomorphically evaluating the decryption operation using an encryption of the secret key [459], and was devised by Gentry [211]. Generally, it is undesirable to apply this operation frequently, as it is a computationally expensive operation. Examples of FHE schemes include the BGV [93], GSW [212] and the CKKS [119] schemes.

Although not frequently used in DLTs, they are of use in this work, as later seen in Chapters 4 and 5. Their advantage is preserving confidentiality whilst allowing for a computation to be performed. This property is especially useful when it is undesirable to trust a third

³⁹In the DLT community, the core protocol functionality is commonly referred to as ‘layer 1’, and solutions built on top of this protocol (i.e.: querying the layer 1 logic and interoperating with it) are referred to as ‘layer 2’ solutions [446].

part with decryption of sensitive values to perform a computation over them. In the DLT space, Zama is an example of a commercial implementation of a FHE scheme (namely using Torus Fully Homomorphic encryption [121], which is a variant of the GSW scheme using ring algebraic structures). Their aim is to provide confidentiality over smart contract computations with their technology.

2.1.8 Post Quantum Cryptography

Thus far, all the cryptographic schemes presented rely on the assumption that solving a certain problem is hard. Quantum computational models can challenge some of these assumptions. If quantum computers become practical, algorithms exist to efficiently solve certain mathematical problems that currently underpin the security of most public key encryption schemes [73]. This would have a significant impact beyond the scope of DLTs alone, but we focus on this field only, as it is the subject of study in this work.

Quantum models of computation differ to current computational models. Classical computation models information as bits: the smallest, most basic unit of information from which the rest of computation is developed. A bit is in fact an abstraction of a physical process: computation operations are implemented in physical devices, they are not merely abstract instructions. In practise, the transistor is the physical device that allows current to flow through it, and the bit is the abstraction used to reflect the information encoded in this physical process: a high voltage level can be encoded as a 1, and a low voltage as a 0 [533]. Researchers have found that alternative physical processes can also be used,⁴⁰ and the information abstraction for these processes is called a qubit. The relevant difference to note between a bit and a qubit is the following: a bit encodes a binary value, whereas a qubit encodes a vector measuring the ‘state’ of the physical process [217]. Depending on the physical quantum implementation, these values can be any binary representation. For example: low or high energy state or horizontal or vertical light wave direction [217]. But any time before the physical element is observed, it could be in any of those states. To represent this, a qubit is defined as a vector with two important components: phase and amplitude [244, p. 26]. The magnitude of the qubit vector component represents the probability that when observed, it will be equal to that value. The congruency between a qubit and a bit is that a qubit, when observed, will also collapse to a binary value [244, p. 34]. Like classical logic gates, quantum gates allow for these qubits to be manipulated, and thus program logic can be constructed from them.

Peter Schor devised an algorithm in this quantum model of computation that is able to efficiently⁴¹ find the prime factors of an algorithm. This directly challenges the assumption that solving the discrete logarithm is hard, and therefore rendering cryptographic schemes that rely on it insecure [452]. Precisely, this affects public-key encryption algorithms that leverage the discrete logarithm problem, such as the Diffie-Hellman key exchange and RSA encryption [77]. Lev Grover devised an algorithm that provides a quadratic speed-up to

⁴⁰Some based on superconductors, some based on trapped ion experiments, others on optical photons [382].

⁴¹Meaning it runs in polynomial time.

the problem of finding a collision in a hash function [230]. Both of these algorithms are instances of solving the period-finding problem. That is, finding the phase (or period) of a function $f(x)$ such that $f(x) = f(x + r)$ for some minimal r .

Does this mean confidentiality, security and privacy are doomed? No. There exist other cryptographic algorithms that rely on different mathematical problems being hard to solve (like lattice-based cryptography) in order to provide security. The NIST is providing standardisation for which ones should be adopted [364], and the research community in cryptography and cryptanalysis continues to propose algorithms [92, 199, 187], audit their security, and in some instances break [110, 79] and discard them.

Quantum computation, like classical computing, relies on two components: hardware and software. Building a quantum computer that can run quantum algorithms is non-trivial. Large multinationals are leading this research effort, summarised in Table 2.3. Regardless of improved hardware capabilities, the reality is that currently, quantum computation devices are unable to achieve a large number of bits and require a high number of redundancy for error correction [182]. Current factorisation efforts remain far from being competitive with the key lengths employed in practice for public-key cryptography, with IBM having successfully factored 15 into its prime factors of 5 and 3 [513], whereas standard RSA keys is between 2048 to 4096 bits long [49]. Other results have been heralded as successes for larger primes, but the work in [232] uncovers mathematical ‘tricks’ used in the choice of primes to facilitate this factoring, and tendencies in the field to exaggerate the practical usefulness of these achievements.

How quantum computing will affect DLTs is largely unclear. Opinions vary in the academic field. Some highlight the reality of quantum factorisation and the practicality of the quantum threat, whilst other academics exercise increased caution. For example, Bernstein argues that, among known quantum collision algorithms, none matches the cost-efficiency of classical methods [76]. The authors of [234] further constrain possible quantum gains under realistic device limitations, and [477] explore limits in analogous settings (Grover oracle models) that challenge broad assumptions about quantum speed-ups. However, it is sound to state that quantum computing will have a significant impact on the components of DLTs that rely on discrete logarithm based public key cryptography. Namely, DLTs frequently use elliptic curve digital signatures for transaction validation, which are not quantum safe, and should be replaced with standardised alternatives. Hash functions are unlikely to suffer significantly in the quantum computation model, and symmetric key encryption remains safe (provided adequate key management—although this is a requirement irrespective of quantum computing being a widespread reality or not). A potential effect quantum computing could have is in the strategy of miners [432]. Sattath argues that a new danger introduced by quantum mining is that miners need not commit to a fixed number of search iterations beforehand—they can stop and measure mid-computation upon observing competitor blocks—effectively salvaging partial progress rather than discarding all prior work. This flexibility undermines the assumption that most non-winning mining effort is wasted, and can lead to significantly more forks, challenging Bitcoin’s consensus

stability.

Company	Model	Technology Type	Qubits	Year Launched
IBM	System One	Superconducting transmon	20	2019
	System Two [205]	Superconducting transmon	133	2023
Google	Sycamore [32]	Superconducting transmon	53	2019
Quantinuum	System Model H1-1	Trapped-ion (QCCD)	12–20	2020–2025
	System Model H2 [156]	Trapped-ion (QCCD)	56	2023–2024
IonQ	Harmony, Aria, Forte [256]	Trapped-ion	11–36	2020–2022
Rigetti	Ankaa-3 [422]	Superconducting transmon	84	2024
	Novera [421]	Superconducting transmon	9	2023
Microsoft	Majorana 1 [10]	Topological (Majorana qubits)	8	2025

Table 2.3: Current quantum physical computation devices.

2.2 DLTs: Architectures and Flavours

Having gained an overview of the techniques that underpin DLTs’ functioning, we may proceed to discuss how DLTs vary in practical implementations. The reader may recall that from the previous chapter that a DLT is a decentralised, online record of transactions [424]. This record, also known as ledger, is maintained and populated by a network of nodes (i.e.: computers) [484]. Nodes follow a set of rules to decide what information to include in the ledger, and to agree on what version of the record of transactions is accepted [323].

DLTs must offer two minimum requirements to be fit for purpose: they must be decentralised—lacking a single point of control or failure—and they must record only one immutable, non-conflicting version of the truth. A number of projects have emerged that prove there are many ways in which these core functionalities can be delivered. They come in different flavours: they may vary the node network architecture, the set of rules they follow to come to an agreement on what is the accepted set of transactions, or the basis on which nodes are allowed to participate in the protocol. These are some of the main differences that DLT architectures may present, if we only consider the baseline protocol, but we may also consider the interoperability and compatibility with other solutions built on or relying on the DLT architecture itself. In the DLT community, the core protocol functionality is commonly referred to as ‘*layer 1*’, and solutions built on top of this protocol (i.e.: querying the layer 1 logic and interoperating with it) are referred to as ‘*layer 2*’ solutions [446]. As such, there exists a rich design space, and a wide range of architectural choices that can be explored, both for layer 1 and layer 2 parameter choices.

Remark: throughout the rest of this work, the terms DLTs and cryptocurrencies such as Bitcoin or Ethereum will be used interchangeably. However, for the purpose of precision, it should be clarified that cryptocurrencies are considered a set of DLT implementations.

2.2.1 Architectures

Blockchain

The first architecture for a decentralised ledger technology that was proposed is Bitcoin, which is a blockchain implementation [374]. Blockchain, as the name indicates, is a single sequence of ‘blocks’. These ‘blocks’ are packets of information, which contain an accepted ordering of valid transactions. Each transaction is a record of a user sending funds from their address to another user’s address. A user’s address, is often also called a wallet [81]. This can be a bit of a misnomer, because there are no coins that are stored in this address, and an address is not a memory allocation or a database [485]. Rather, an address is a value that is derived from a user’s public key [485].⁴² This address value tells other participants in the network where they can send funds to, but it does not certify ownership of the funds by itself. Ownership, or rather the right to spend, is only proven when users sign a transaction with their valid private key. In other words: there are funds that can only be spent by whoever provides a valid cryptographic signature matching the address value.

Directed Acyclic Graphs

This DLT architecture stores blocks (transactions) in a graph data structure. The graph contains no cycles, and is directed towards the genesis block, or in other words, the first block of the ledger [401]. The importance of the graph being acyclic stems from the need to prevent circular or self-referential validation of blocks. The direction of the graph provides an order on the history of the ledger, displaying blocks in the temporal sequence in which they were created. Compared to blockchains, DAGs can offer higher throughput and lower latency, since multiple transactions can be appended concurrently rather than waiting to be assembled into sequential blocks [439]. However, this parallelism also introduces challenges: consensus mechanisms for DAGs are often more complex, and without careful design they may struggle with issues of security, double-spending prevention, or ensuring a consistent global order of transactions [12].

Hashgraph

A hashgraph is another architecture for distributed ledgers, which, unlike blockchain, does not arrange transactions into a single linear chain. Instead, it uses a directed acyclic graph where participants share information through a protocol called ‘gossip about gossip’ [45]. In this process, nodes exchange signed messages that contain both transactions and references to previous communications. From this, every participant can reconstruct the full history of who has talked to whom, and in what order. This enables the network to

⁴²The exact procedure is described in multiple sources, one example being [Elliptic](#). In brief, a Bitcoin address is obtained by applying a series of hash functions to the public key (itself derived from the user’s private key), followed by the addition of a checksum—a short block of data used to detect errors in transcription or transmission [391, p. 144].

reach consensus using a technique known as virtual voting, rather than requiring energy-intensive proof-of-work or a leader to propose blocks [527]. Unlike DAGs such as IOTA’s Tangle, Hashgraph does not rely on a central coordinator to finalise transactions. An additional feature of Hashgraph is transaction ordering fairness. Consensus timestamps are assigned to transactions based on when they were actually received by the network, so that earlier transactions appear before later ones, rather than leaving the order to the discretion of miners or validators. The main drawback of Hashgraph is that it is permissioned and patented, which restricts openness and adoption compared to public blockchains [12].

2.2.2 Consensus Algorithms

Consensus algorithms are the set of rules that participants in a DLT follow in order to come to an agreement on what is the accepted record of transactions on the ledger [527]. There are a wide range of consensus mechanisms in the DLT space, and a number of publications have reviewed their functioning and comparisons, see for example [528, 328, 527, 252]. Whilst the number of existing mechanisms is high, many of these are variants or hybrid implementations of the two most frequently used which are proof-of-work and proof-of-stake [252]. As such, we focus on these.

Proof-of-Work

In proof-of-work (PoW) consensus protocols, miners are required to find a nonce (number only used once) such that when combined with the block’s contents and hashed (typically using SHA-256 in Bitcoin), the resulting hash is less than a predetermined target value [374]. This target is a difficulty threshold: the lower its value, the smaller the solution space for valid nonces. In other words, finding ν such that $\nu \in \mathbb{N}$ satisfies the inequality $H(\text{block data} \parallel \nu) < T$, where T is the current *target* value set by the network’s difficulty adjustment algorithm. The difficulty of finding a valid nonce secures the network by making it costly to rewrite history; once a block is accepted, altering its data would require re-computing the PoW for it and all subsequent blocks. Verifying a hash is trivial compared to generating it, thus allowing all other participants to efficiently verify the PoW was performed correctly.

Proof-of-Stake

In proof-of-stake (PoS) consensus protocols, the right to propose and validate new blocks is not determined by solving computational puzzles, but instead by the amount of cryptocurrency that a participant has “staked,” i.e., locked up as collateral [427]. Formally, let $V = \{v_1, v_2, \dots, v_n\}$ be the set of validators, where each validator v_i has an associated stake $s_i \in \mathbb{R}_{\geq 0}$. The total stake is:

$$S = \sum_{j=1}^n s_j.$$

The probability of validator v_i being selected to propose the next block is then:

$$P(v_i) \propto \frac{s_i}{S}.$$

This probabilistic leader selection replaces the brute-force search of PoW with a mechanism that is economically costly to subvert: altering the ledger would require controlling a majority of the total stake rather than a majority of computational power [527]. Once a block is proposed, other validators confirm it by signing with their private keys, and the block is added to the chain if it collects sufficient attestations [528]. Whilst PoS avoids the energy costs of PoW [552], it raises concerns about economic centralisation: participants with greater stake have proportionally higher influence and reward, potentially reinforcing wealth concentration. Current PoS ledgers attempt to address these issues by allowing smaller participants to pool their stake with larger ones, so that rewards are shared. Another concern of the PoS system is the “nothing-at-stake” problem: the concern that validators could sign multiple competing chains at no cost [427, 528]. In practice, PoS protocols discourage this behaviour by taking away part of a validator’s deposit if they attempt to support conflicting histories [318], and by enforcing rules that make the agreed chain final once enough participants have confirmed it.

2.2.3 Transaction Record Model

If funds are not stored in addresses, how does blockchain keep track of funds, and who can spend them? Bitcoin’s answer to this concern is a model called the unspent transaction output model, abbreviated as UTXO, whilst other cryptocurrencies such as Ethereum use an alternative solution: the account-based model.

UTXO Model

The UTXO system is conceptually similar to spending cash. For example, if Alice pays one pound to Bob with a five-pound note, Alice hands over the note, Bob keeps one pound, and Alice receives four pounds in change. In Bitcoin, if Alice sends funds to Bob, the transaction consumes one of Alice’s unspent outputs and creates two new outputs: one paying Bob the intended amount, and another returning the remaining balance to Alice (the “change”) [158]. Each transaction output can be spent only once, so the global set of UTXOs can be represented as a directed acyclic graph of transactions, where edges connect outputs to the inputs that consume them. Unspent outputs remain available for future transactions, while spent ones are permanently invalid. An attempt to reuse a spent output constitutes a *double spend*, and would cause a ‘fork’ in the network [447]. In other words: two conflicting versions of the ledger. The consensus algorithm used ensures that only one version prevails, and thus a double spend would eventually be orphaned, meaning no new blocks are attached to it [374].

Account-Based Model

The account-based model instead resembles how balances are tracked in a bank ledger. Rather than consuming and creating discrete outputs, each participant has an account with a balance that is updated directly when transactions occur [537]. For example, if Alice sends one pound to Bob, Alice’s account balance is reduced by one and Bob’s is increased by one. This model is used by Ethereum [537] and many other smart contract platforms, as it simplifies state management and makes it easier to implement programmable logic

such as contracts that hold and transfer funds [98]. However, because balances are mutable rather than discrete, preventing double spending requires careful consensus on the exact sequence of transactions [98]. In other words, the network must agree on a single global order of transactions to ensure that two conflicting updates (e.g., Alice trying to spend the same funds twice) cannot both be applied.

Chapter 3

A proposal for a decentralised data market

Abstract

This chapter presents an architecture for a decentralised data market. It is proposed for applications in which agents are incentivised to collaborate to crowd-source their data. The architecture is designed to reward data that furthers the market’s collective goal, and distributes reward fairly to all those that contribute with their data. Simulations show that the architecture is resilient to *Sybil*, *wormhole*, and *data poisoning* attacks. In order to evaluate the resilience of the architecture, the breakdown points for various adversarial threat models are characterised, using as context an automotive use case.

3.1 Contributions Disclosure

The work here presented was published in [280], which was authored by myself in collaboration with Christian Jursitzky, Quan Zhou, Pietro Ferraro, Jakub Marecek, Pierre Pinson and Robert Shorten. Everything presented in this chapter has been written by myself. I co-collaborated with Quan Zhou on the Maximum Entropy Voting contribution. Pietro Ferraro led the modification of the Maximum Entropy Voting algorithm (C-MEV), and the simulations of the data consensus algorithms along with Christian Jursitzky (Figures 3.7, 3.8, 3.9). Everything else is my own work: the simulations, text and ideas, with guidance and collaboration of my supervisors.

3.2 Motivation

This project constituted my initial research endeavour and was the primary motivation for undertaking this doctoral study. Its inception was motivated by an observed emergence of case studies where the value exploited from data was not recompensed to the original data producers.

Since the advent of the Web 2 era, the internet has shifted from a static information-delivery model to a participatory, user-driven paradigm [85]. By the early 2010s, a shift towards data-driven business models was noticeably taking place [562]. Users made their data available to large platform providers—perhaps inadvertently—in exchange for services (for example, web browsing) [164]. In recent years, the fairness and ethics of this exchange has come into question [478], [307], [125], [235], [302]. Leveraging user data to maximise advertising efficiency seems in principle, harmless. However, this business model has come under scrutiny following a number of scandals concerning the use of participants’ data for unethical purposes [75], [531]. Of concern is that participants were unaware of what their collected data was being used for and not informed of opt-out possibilities [435]. Whilst users do receive a service from these providers (often for free or a minimal fee), they relinquish their data to the provider, and it has been argued that this exchange of value may not always be symmetrical, fair or ethical [337], [271]. The prevailing ownership model is highly centralised: a many-to-one structure in which millions of individuals concede their data to a small number of dominant platforms, which then monetise it [456].

When I began this research (October 2021 – May 2022), due diligence indicated that while alternative models for data ownership were under early exploration, none had yet succeeded in delivering a decentralised architecture for data as a tradable asset. Decentralised ownership structures themselves are not novel, and were successfully being incorporated into the digital space since the Web3 movement emerged [163]. However, none had successfully addressed this matter for data as the transacted good. For me, ‘success’ required meeting three criteria: the solution is i) truly decentralised¹, ii) resilient to attack vectors and iii) fairly exchanges value between data producers and buyers.

Reflecting on this work three years later, the scandals and unethical practises that personally motivated me to pursue this research remain pressing [547]², [265]. The Facebook-Cambridge Analytica scandal was an eye-opening event. In 2018, a former employee³ reported that the consulting company Cambridge Analytica had collected data of over 80 million Facebook users to build a psychological profile of them [540, p. 7]. This information was used for a targetted advertising campaign to influence the 2016 US election, and the information on users was collected without consent [183]. In fact, Cambridge Analytica had a worldwide portfolio of elections they had been hired to influence [246, p. 15] and

¹To clarify, when referring to the solution as ‘truly decentralised,’ this is not to claim that the entire solution is fully decentralised in every aspect. Rather, that the specific components I designate as decentralised truly operates without reliance on a centralised authority.

²Of relevance from this source are the following quotes: *"The Lavender software analyzes information collected on most of the 2.3 million residents of the Gaza Strip through a system of mass surveillance, then assesses and ranks the likelihood that each particular person is active in the military wing of Hamas or PIJ. According to sources, the machine gives almost every single person in Gaza a rating from 1 to 100, expressing how likely it is that they are a militant. Lavender learns to identify characteristics of known Hamas and PIJ operatives, whose information was fed to the machine as training data, and then to locate these same characteristics — also called “features” — among the general population, the sources explained. “The more information, and the more variety, the better,” the commander writes. “Visual information, cellular information, social media connections, battlefield information, **phone contacts**, **photos**.”"*

³This former employee is often regarded as a whistleblower. Their intentions remain questionable—the former employee then founded their own similar company shortly after breaking the information [255], using the same collected dataset [292].

even in the aftermath of the company’s dissolution, multiple other similar companies have been founded by former executives of the consultancy [114, p. 8]. More recently, in 2024, the Romanian election was annulled following the de-classification of documents by the Romanian intelligence services that *"showed evidence of voting manipulation through social media platforms, illegal campaign financing on TikTok, cyber-attacks orchestrated by external forces and suspected Russian interference"* [468], [409]. These examples underscore the dangers of granting large multinationals unaudited access to personal data. Other concerning applications include using vast amounts of collected data for state surveillance or military operations, which often involve state collaborations with private, multinational companies [504], [472], [227], [113].

Beyond political misuse, questions of fairness in the economic value exchange persist. In 2021, Apple introduced its App Tracking Transparency framework [27], requiring applications to request explicit user consent for cross-platform tracking and sharing of their data with third parties for advertising purposes. This single opt-out mechanism caused the share prices of major platforms such as Snapchat, Facebook, Pinterest, and Google to plummet, [148] losing an estimated \$142 billion in market value [233]. This led me to question whether the worth of the service received from these platforms is as valuable as the cost of the data provided to them for free.

There are also ethical concerns surrounding the “invisible” labour behind the data economy. Data workers perform the essential, yet undervalued tasks of content moderation, data cleaning and data labelling. Service providers already relied on their work for training their machine learning models or delivering safe content on their platforms [355]. Businesses outsource this task to workers in Africa, Latin America and Southeast Asia, where workers receive compensation in the ranges of 1 to 3 dollars an hour [399]. Numerous reports have highlighted the emotional damage to workers from exposure to violent content, frequently reporting trauma and inadequate support [209], as well as pressure to meet unrealistic productivity targets for inhumane compensation [413]. The report in [179] is a sobering read. In 2022, a year after completing this research project, OpenAI released ChatGPT: a Large Language Model with a chat interface. This event ensued the subsequent boom of generative Large Language Models, as well as a larger class of generative models. This technological arms race came with a hidden human cost: the number of large multinationals with a demand for data workers continued to increase. Without dismissing the importance of scientific innovation in new model architectures, these models are akin to an engine without fuel if no quality data is available to train them on. The exploitative practise of data work has been vehicular to enable this boom, and it is a stark example in asymmetric value exchange. The prevalence of unethical practises to obtain data is not confined to this case only. Reports of companies using datasets to train their models without compensating the producers of said data have continued to emerge: examples include using pirated books [509], [289] and scraping Getty images [245].

None of this sat particularly well with me, so I thought: why don’t we design an architecture to challenge this status quo? What if we could bring the cooperative structure of

simultaneously owning and producing into this industry? The Web3 movement has shown that users can successfully co-opt what was a previously centralised service and allowed them to both own and produce it, even in domains as entrenched as money. Who thought a network of people could organise and decide to side-step governments and banks to create, run and own their own currency? Thus, I decided to examine how this could be done with data. Can we own, create, and transact our own data? This architecture is my attempt to do so.

However, applying decentralisation to data presents unique challenges, as data cannot be treated in the same way as other goods where democratisation has succeeded. Unlike physical goods—or even digital currencies—it is both possible and often desirable to generate new data on demand, and to sell the same dataset to multiple buyers. These characteristics mean that a direct analogy to cryptocurrencies is inadequate. Addressing these challenges is central to the architecture proposed in the work that follows.

3.3 Research Scope

The challenges described in the above section call for a novel infrastructures to track and trade data ownership. These infrastructures can be classified as being either centralised, in which an oracle looks after security and management issues, or decentralised in which trust and security is encoded as part of the data market peer-to-peer (P2P) protocol. The design of such markets is not new and there have been numerous attempts to design marketplaces to enable the exchange of data for money [473, 46, 412, 222]. This, however, is an extremely challenging endeavour. Data cannot be treated like a conventional commodity due to certain properties it possesses. It is easily replicable; its value is time-dependant and intrinsically combinatorial; and dependent on who has access to the data set. It is also difficult for companies to know the value of the data set a priori, and verifying its authenticity is challenging [9]. Our particular interest is in developing a data market design that is hybrid in nature; *hybrid* in the sense that some components of the market are provided by trusted infrastructure⁴, but where the essential components of the market place, governing ownership, trust, data veracity, etc., are all designed in a decentralised manner. This data market is designed for crowd-sourced sensing applications with a view to: (i) achieving a verifiable exchange of data ownership between sellers and buyers; (ii) given an oversupply of data, ensuring that participating agents receive a fair amount of writing access rights to the market; (iii) automatically selecting data points, from all those available, to add the most value to the data market’s collective goal; and (iv) being robust in adversarial environments by providing protection against Sybil, wormhole and data

⁴In our case two components are provided by centralised infrastructure: identity and voting. Identity can be replaced by decentralised solutions, but paying a price in complexity. We consider this choice of centralisation as justified given that the precise identity itself is not critical to the functioning of the data market, what matters is that the agent is real. Furthermore, in this automotive industry case study, all participants can check that the identity is valid by simply checking the agent has a license plate. Thus, this centralised verifier is publicly verifiable. Regarding the voting, it is required that some agent or group of agents run a voting scheme. However, elections can be delivered with end-to-end verifiability guarantees, allowing all participants to check the correctness of the election, and the election can be run by a number of agents, thereby making it possible to distribute responsibility amongst mutually distrusting agents.

poisoning attacks (defined in Section 3.8). Further, we characterise its breakdown points for various adversarial threat models in an automotive use case. Finally, to illustrate the potential utility of the data market, we consider a specific use case in a smart mobility environment that, we hope, might represent a first step towards more general architectures. In this setting, drivers of vehicles within a smart city wish to monetise the data harvested from their cars’ sensors [475]. This use case, while simplifying several aspects, still captures many pertinent aspects of more general data market design: for example, detection of fake data; certification of data-quality; resistance to adversarial attacks.

To summarize, the contributions of this chapter are:

- An architecture proposal for a data market. The architecture is resilient against a number of attack vector and has guarantees of fairness. It is also compatible with privacy preserving techniques.
- A novel proof-of-work mechanism that is adaptive, useful for the functioning of the market and fair by design.
- An alternative formulation for the Maximum Entropy Voting (MEV) algorithm, called Combination-MEV (C-MEV) and a novel application of this algorithm in the context of automotive data use cases.

3.4 Related Work

Numerous authors have proposed centralized [9, 317, 349, 414, 357, 410, 150] and decentralized [412, 253, 496, 46, 222, 467, 390] data markets, recently. Industry-specific data markets, e.g., in power systems, agricultural or IoT sensor data [221, 5], are also proliferating. Some recent decentralised implementations include Catena-X [367], Ocean Protocol [197], Data Broker DAO [512]. Of the decentralised data market proposals, most use Blockchain architectures as a basis for their work. However, these proposals often fail to address blockchain design flaws that rapidly become exposed in the context of data markets [412, 253, 496]. For example, Proof-of-Work (PoW) based blockchains reward miners proportionally to their computational power. Aside from the widely discussed issue of energy wastage [552], blockchain based systems also typically use commission based rewards to guide the interaction between users of the network, and blockchain miners. Such a miner-user interaction mechanism is not suitable in the context of data markets, effectively prioritising wealthier users’ access to the data market. In addition, miners with greater computational power are more likely to earn the right to append a block, and thus earn the commission. This reward can then be invested in more computational power, leading to a positive feedback loop where more powerful miners become more and more likely to write blocks and earn more commissions. Similarly, the wealthier agents are the ones more likely to receive service for transactions of higher monetary value. This could cause traditional PoW-based blockchains to centralise over time [63]. It is worth noting that centralised solutions to data markets already exist, such as [65], which namely focus on implementing methods to share and copy data, and to protect certain rights to it, such

as read rights. Furthermore, none of these implementations consider the security concern of selling artificially generated data. They do ensure that the posted metadata matches the sold data, but there is no method to ensure that the data that is sold is of good quality, sound and real. Instead, they trust that the seller is honest, or incorporate the assumption that a reputation mechanism exists to filter agents based on trustworthiness. Our work addresses these security concerns by proposing an architecture that is resilient to data poisoning (i.e., injection of fake data streams) and wormhole attacks (claiming data has been collected from one location, when in fact it has been collected from elsewhere).

Another possible categorisation of prior work relates to the trust assumptions made in the system design. The work in [414] assumes that upon being shared, the data is reported truthfully and fully. In practice, this assumption rarely holds. This assumption is justified in their work by relying on a third party auditor, which the authors of [496] also utilise. However, introducing an auditor simply shifts the trust assumption to their honest behaviour and forgoes decentralisation. In [9], it is identified that the buyer may not be honest in their valuation of data. They propose an algorithmic solution that prices data by observing the gain in prediction accuracy that it yields to the buyer. However, this comes at the cost of privacy for the buyer: they must reveal their predictive task. In practice, many companies would not reveal this Intellectual Property, especially when it is the core of their business model. The same concern applies to the proposal in [327]. The work of [375] is an example of a publicly verifiable decentralised market. Their system allows for its users to audit transactions without compromising privacy. Unfortunately, their ledger is designed for the transaction of a finite asset: creating or destroying the asset will fail to pass the auditing checks. For the transaction of money this is appropriate: it should not be possible to create or destroy wealth in the ledger (aside from public issuance and withdrawal transactions). However, for data this does not hold. Users should be able to honestly create assets by acquiring and declaring new datasets they wish to sell. Furthermore, their cryptographic scheme is built to transfer ownership of a single value through Pedersen commitments. This is not suitable for transferring a dataset, which would likely contain a large number of data points.

The data market architecture presented in this chapter deviates from those traditionally considered in the literature (such as those mentioned above) in a number of key aspects. First, rather than being centralised or decentralised, this architecture is hybrid in nature. Secondly, in this data market, the trust assumptions are embedded in mechanisms that can be verified by all agents using the market place.

3.5 Architecture

In this section, a high-level overview of the proposed architecture is presented. The reader should recall that the data market is designed to operate in crowd-sourced environments, where the following conditions prevail:

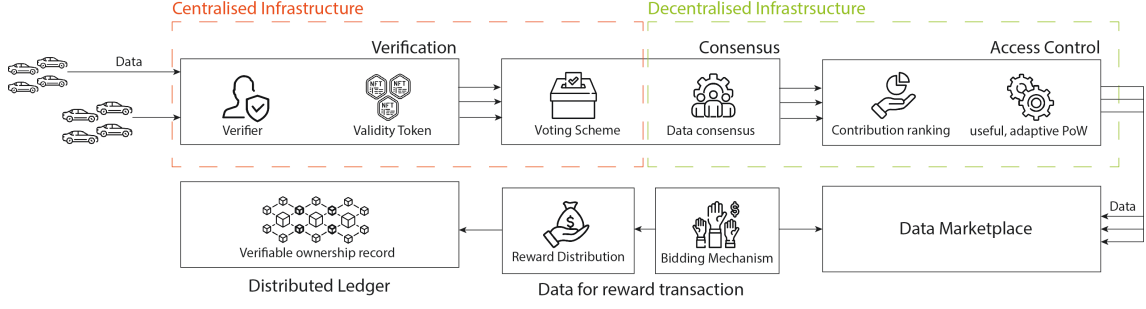


Figure 3.1: Data Market Architecture. Credit for the images is given in ⁵.

1. For each potential data point that can be made available in the data market there is an over-supply of measurements, as is the case, for example, in many crowd-sourcing applications.
2. Competing sellers are interested in aggregating (crowd-sourcing) data points from the market to fulfil a specific purpose. For example, in applications in which sensors measure environmental quantities (rainfall, pollution levels), data may be aggregated to increase robustness, or simply to provide a macroscopic view of the measurements.
3. Buyers agree to only purchase data from the market, and that each data point in the market has a unique identifier so that replicated data made available on secondary markets can be easily detected by data purchasers. This may seem like a strong assumption. Why would buyers wish to purchase data from this market? Because our proposal offers increased data quality, reliability and trustworthiness. The protocol is resilient against injections of fake data, so buyers have security guarantees: they will not be acquiring bogus data.
4. There is a mechanism that can verify the geographical location of an agent at time of data collection with a certain degree of confidence. Several works have been carried out that corroborate that this is a reasonable assumption to make [332, 560, 91, 538, 86] and later we provide one such technique in [282].

In many mobility applications, where cars are used as sensing devices, many, if not all of the above conditions prevail. To illustrate this, consider applications where cars seek to monetise information associated with their geographical location. First, agents present a valid proof of their identity and location, as well as demonstrating that their information is timely and relevant. Agents that succeed receive a validity token that allows them to form groups with other agents in their proximity, which we call spatial coalitions. Each agent measures data points from a location quadrant. These data points are then aggregated by an elected committee of agents from a spatial coalition of that location quadrant. The

⁵In order of appearance: Icons made by Freepik, Pixel perfect, juicy_fish, srip, Talha Dogar and Triangle Squad from www.flaticon.com.

spatial coalition provides an objective function, that is used to determine the utility of said data. It is then possible to calculate how valuable data points are with respect to said objective function. This is done by computing the Shapley value [450] of the data points. The Shapley value is used to measure the marginal contribution of data point towards maximising the objective function, and is shown to satisfy notions of fairness. The higher the Shapley value, the more valuable that data point is. Each agent receives a Shapley value for their data point provided. This determines the amount of proof-of-work they must compute to sell their data. The greater the Shapley value, the less work they must perform. Indeed, this work consists of computing the Shapley value of a new set of incoming data points. This feature ensures that spam attacks are costly because for every new data point an agent wishes to sell, they must perform a new a proof-of-work. Furthermore, the work agents perform is useful for the functioning of the data market.

An architecture that implements such a system is depicted graphically in Figure 3.1. We proceed with a brief description of the functional components of the data market.

Verification: Agents' position are verified by a proof of position mechanism [282] that ensures they provide a valid position and identity. This component ensures that spam attacks are expensive. All agents in the market can verify the validity of a position proof and valid identity because this information is publicly available.

Voting Scheme: Agents belonging to a spatial coalition (a group of agents in close proximity to one another) agree on what data is worthy of being trusted and sold on the data market. Agents express their preferences for whom they trust to compute the most accepted value of a data point in a given location. This is carried out through a voting scheme.

Data Consensus: Once a committee of agents is elected (in the step above), it must come to a consensus as to what is the agreed upon dataset associated to a given location. This is computed by the group following an algorithm that aggregates the coalition's data. This component ensures that at a specific location the dataset does not contain data points from faulty sensors or from malicious agents (i.e., agents can decide to eliminate outliers).

Access Control Mechanism: This mechanism used to decide how data should be prioritised to enter the data market. This mechanism has two steps: firstly, all data points are assigned a value, which determines the priority they receive to enter the data market; and secondly, proportionally to this priority, the agent owning that data point must perform an adaptive, useful proof-of-work to sell their data.

Data Market: The content of the datasets is not public. Buyers purchase the right to access the datasets and perform analytics on them. To provide sufficient information for the buyers, each dataset has metadata associated with it that provides a description of the dataset. The sellers also provide the provenance of the data along with a valid proof of position to verify this, and the objective function that their dataset maximises and its value. Buyers can access and browse the market and place bids for specific datasets

in exchange for monetary compensation. Buyers may wish to purchase: (i) access (but not ownership) to the entire dataset; (ii) access (but not ownership) to only part of the dataset; (iii) ownership of the dataset or of part of it, which would give them the rights to redistribute or perform further analytics on said dataset. Each right has a corresponding price that would be decided between the sellers and the buyers.

Bidding Mechanism: The mechanism matches buyers to the sellers of data. This component determines the price-per-right. At this stage, a spatial coalition formed of multiple agents is considered to be one seller. Interested buyers have their own value function, which they may or may not choose to disclose. This function is different to the one publicly disclosed by the sellers in the data market. Buyers can make an informed choice with the public value function to determine if the data sold is useful to their predictive task. Successful sales will be recorded in an immutable ownership record that is public, such that all participants of the market can see which agents have rightful access to a dataset.

Purchasing Datasets: Once the steps above have been completed and the metadata of data have been posted publicly on the market, the sellers chose which buyer or buyers they will sell their dataset to. Finally, once a transaction is successful, the reward of the sale is distributed amongst the participants of the spatial coalition that generated the dataset according to a given reward distribution function.

Reward Distribution: Once a transaction is successful, then the reward of the sale is distributed amongst the participants of the spatial coalition that generated the sold dataset. This is to ensure that all agents participating in the crowd-sourcing of the dataset receive adequate compensation for it.

Distributed Ledger: Successful transactions are recorded on a distributed ledger to provide a decentralised, immutable record of ownership. The ledger determines which agents have access to particular data, and what access rights they are allowed. Access and ownership to data is managed and exchanged through the use of smart contracts [56].

3.6 Components

In this section we provide definitions of the components of, or interacting in the data market architecture.

Agent: An agent is defined as $a_i \in A$ where A is the set of all agents competing to complete the marketplace algorithm to become sellers. The index $i \in N$, where $N = |A|$ and N is the total number of agents in the data market at a given time, t .

Data Point: A data point is defined as $x \in X$ where x denotes the data point of agent and X is the space of all possible measurements. When relevant, we denote x_i as the data point belonging to agent a_i , and we also denote x_t as the data point collected at time t .

Location quadrant: The set of all possible agent locations is defined as $\mathcal{L}$. The location quadrant q , is an element of $\mathcal{L}$, where $q \in \mathcal{L}$.

Buyer: A buyer is denoted b , where $b \in B$ and B is the set of agents looking to purchase ownership (or any other rights) to the datasets available for sale.

Spatial Coalition: A spatial coalition, C_q , is a group of agents in the same location quadrant, q .

Crowdsourced Dataset: Agents in a spatial coalition, C_q , aggregate their data points to provide an agreed upon data point, X_t , at time, t .

Objective Function: An objective function, $v_{C_q}(\cdot)$, maps the aggregate data points of a spatial coalition to a utility.

Shapley Value: The Shapley value, $\psi(C_q)$, is a mechanism to distribute reward amongst a coalition, C_q , by considering how valuable the contributions of each agent in the coalition are, with respect to an objective function, $v_{C_q}(\cdot)$. The Shapley value is the unique reward allocation that satisfies all the properties of the Shapley fairness criteria [450].

3.7 Algorithms

This section describes the algorithms used in the data market. Namely, the proposed market is orchestrated using three algorithms: one for agent verification; one to implement a reputation-based voting system and an access control algorithm. We proceed by describing their technical operation.

3.7.1 Verification Algorithm

This algorithm is run by a central authority to provide agents with a seal of credibility for the data they wish to provide. The purpose of this algorithm is to prevent Sybil attacks. Agents can verify that a commitment⁶ is well formed and that the proof of position algorithm outcome is correct. However, they must trust the central authority to check that the ID provided by the agent is valid.⁷ For example in the context of mobility applications, vehicle license plates can be used as a proxy for a valid identity.

The validity of the data submission must be verified before the data reaches the data market, to avoid retroactive correction of poor quality data. This is done through the Verify algorithm, defined below. The algorithm relies on the following functions:

Commitment: An agent commits to their data point by generating a cryptographic commitment that is hiding and binding, such that the data point cannot be changed once the commitment is provided. A commitment to a data point x_i , location quadrant q , and ID i , of an agent a_i , at time t , is defined as $c \leftarrow \text{Commitment}(a_i, x_i, q, t)$

Proof of ID: Let the Proof of ID be an algorithm, IDProof, that verifies the valid identity of an agent a_i , with ID i . In the context of the application presented, this identification

⁶A cryptographic commitment is a primitive that allows an agent to commit to a value such that it is hidden and cannot be changed later [147].

⁷Some decentralised proofs of identity have emerged such as the Proof of Humanity project [137].

will be the license plate of a vehicle. The algorithm will return a boolean, α , that will be *True* if the agent has presented a valid license plate and *False* otherwise. Then IDProof is defined as the following: $\alpha \leftarrow \text{IDProof}(i)$. It is executed by a central authority that can verify the validity of an agent's identity.

Proof of Position: The proof of position algorithm, PoP, is called by an agent a_i , with ID i . The algorithm takes as inputs the agent's commitment, c , and their location quadrant, q . Let us define PoP as the following algorithm:

$\beta \leftarrow \text{PoP}(q, c)$

where the output will be a boolean β that will be *True* if the position q matches the agent's true location and *False* otherwise.

Algorithm 1 $\text{VERIFY}(a_i, x_i, q, t)$

```

1:  $c \leftarrow \text{Commitment}(a_i, x_i, q, t)$ 
2:  $\alpha \leftarrow \text{IDProof}(i)$ 
3:  $\beta \leftarrow \text{PoP}(q, c)$ 
4: if  $t$  is not expired then
5:    $\gamma \leftarrow \text{True}$ 
6: else
7:    $\gamma \leftarrow \text{False}$ 
8: end if
9: if  $\alpha = \text{True} \wedge \beta = \text{True} \wedge \gamma = \text{True}$  then
10:  return True
11: else
12:  return False
13: end if

```

3.7.2 Voting Algorithm

This algorithm is used to elect a set of agents who will be tasked with aggregating data from a given location. The voting algorithm we propose is a reputation-based Maximum Entropy Voting (MEV) variant. Social choice theory offers a wide catalogue of voting mechanisms, where each deliver some desirable properties whilst compromising on others. The data market architecture is compatible with any choice, although we favour probabilistic⁸ or cardinal⁹ voting schemes, in light of Arrow's findings. Arrow famously found in [30] that no deterministic, ordinal scheme can satisfy three seemingly straightforward properties:

Non-Dictatorial: There exists no single voter with power to determine the outcome of the voting scheme.

Independence of Irrelevant Alternatives: The output of the voting system for candidate A and candidate B should depend only on how the voters ordered candidate A and candidate B, and not on how they ordered other candidates.

⁸Deterministic voting schemes will always return the same output if run many times on the same ballots (input). Probabilistic schemes may return a different result if they are run multiple times on the same set of ballots, because the output is selected according to a probability distribution.

⁹Ordinal voting schemes only consider what ranking candidates are preferred in, but not by how much one candidate is preferred over another. Cardinal voting schemes do consider this magnitude.

Pareto property: If all voters prefer candidate A to candidate B, then the voting system should output candidate A over candidate B.

Probabilistic schemes sidestep this impossibility result, however Gibbard [213] and McLennan [348] studied them and found that if we consider another reasonable property which is that the voting scheme is resistant to tactical voting, i.e., it satisfies *strategy-proofness*, then the voting scheme must be the Random Dictator¹⁰. Whilst this scheme in theory offers desirable properties, in practise, if a malicious agent is sampled as the winner, the consequences would be dire in our application. This malicious agent could aggregate bogus data and inject it into the market. Maximum Entropy Voting is able to offer Arrow’s properties, and delivers a weaker notion of strategy-proofness [334], which makes it an attractive choice for our application. It also mitigates against randomly selecting a potentially extreme or malicious candidate as the winner. Maximum Entropy Voting, by design, chooses the candidate ordering with the highest amount of entropy, or in other words, the least amount of ‘surprise’. This most moderate choice is subject to the constraint of satisfying *Representative Probability*, which requires that the probability of the outcome placing candidate A above candidate B should be the same as the fraction of the voters preferring A to B.

To summarise: MEV probabilistically chooses the most moderate, least surprising candidate choice, and this probability matches the preference distribution of the electorate. MEV unfortunately pays a price in computational complexity, which motivated our adaptation of the scheme to optimise its speed for a smaller number of candidates. We call this adaptation Combination-MEV, and it is outlined later in this section.

Finally, before proceeding with outlining how MEV is used in our proposal, we introduce one more concept: reputation. Let us recall that this data market’s practical application is for a crowd-sourcing environment. In our case, we propose the example of vehicles measuring environmental data. MEV is the algorithm by which vehicles delegate trust amongst each other to aggregate data from the same location. This example differs from a political election where the candidates are known to the voters. We must find a meaningful way to rank agents (vehicles) to choose the most trustworthy ones, in a scenario where it is unlikely that vehicles know much about each other. In this case, reputation seemed a reasonable choice to measure the utility of trustworthiness. We therefore use a variation of MEV that takes into consideration the reputation of each agent.

Reputation-based Maximum Entropy Voting

We now proceed with introducing the necessary components of the voting algorithm utilised.

Reputation: Let reputation be the metric representing trustworthiness that is assigned to an agent. Formally an agent a_i assigns a score of trustworthiness to an agent a_j . This score is denoted as $r_{i \rightarrow j}$. We assume agents assign reputation following a rational strategy

¹⁰This scheme selects a ballot at random to determine the election winner.

or an agreed upon utility function.¹¹

Representative Probability:

This property states that the probability of the election outcome resulting in candidate A being placed above candidate B should be the same as the proportion of the voters preferring A to B.

Agent Ordering: An agent ordering, o , is defined as a permutation of candidates in [334]. To reduce computational complexity, we suggest computing o by selecting a subset of candidates as the preferred group, such that the order of preferred and non-preferred candidates does not matter.

Ordering Set: The ordering set $\mathcal{O}$ is the set of all possible candidate orderings, such that $o \in \mathcal{O}$.

Probability Measure of Ordering Set: The (discrete) probability measure, $\pi : \mathcal{O} \rightarrow \mathbb{R}_{\geq 0}$ gives a probability of each ordering $o \in \mathcal{O}$ being selected as the outcome ordering, o^* . The probability measure π , is the one with maximal entropy whilst also adhering to the *Representative Probability* property.

Algorithm Procedure

To combine maximum entropy voting and reputation, the agent begins by constructing a pairwise preference matrix $S(a_i) \in \mathbb{R}^{N \times N}$, from the reputation scores it assigns to each candidate.

$$S(a_1) = \begin{array}{c|ccc} & a_i & a_j & a_k \\ \hline a_i & 0 & 1 & 1 \\ a_j & 0 & 0 & 1/2 \\ a_k & 0 & 1/2 & 0 \end{array} \quad (3.1)$$

This matrix is the agent's vote. The entry of a pairwise preference matrix is indexed by every two agents of A , and its values are defined as:

$$S(a_i)_{j,k} = \begin{cases} 1 & \text{if } a_i \text{ prefers } a_j \text{ and } j \neq k \\ 0.5 & \text{if } a_i \text{ prefers both equally and } j \neq k, \\ 0 & \text{if } a_i \text{ prefers } a_k \text{ or } j = k \end{cases} \quad (3.2)$$

for $a_j, a_k \in A$. Agent a_j is preferred to a_k if, for example, $\frac{1+|x_i| \cdot r_{i \rightarrow j}}{1+|x_i - x_j|} > \frac{1+|x_i| \cdot r_{i \rightarrow k}}{1+|x_i - x_k|}$ and both agents are equally preferred if the two values are equal. All agents that can vote, do so by constructing their pairwise preference matrix $S(a_i)$.

Next, an election administrator (an agent tasked with running the election)¹² computes

¹¹A possible way to allocate reputation could be trusting data measurements based on the age of the vehicle or manufacturer. Other examples include: [39] and [335].

¹²Namely, the election administrator is tasked with aggregating all the voters' votes and solving the optimisation problem of MEV to find the winning candidates.

the average pairwise preference matrices over all agents, $S(A)$, where:

$$S(A) := \frac{1}{N} \sum_{a_i \in A} S(a_i). \quad (3.3)$$

$S(A)$ is therefore the aggregated pairwise preference of all agents. Its entries, $S(A)_{j,k}$, display the proportion of agents that prefer candidate a_j over agent a_k . Then, the election administrator must find the candidate ordering, o , that has the highest entropy whilst respecting the preference distribution of the voters, reflected in $S(A)$. This ordering is found by solving these constraints as an optimisation problem.

Optimisation Problem Setup

The original MEV [334] runs an optimisation over all candidate orderings to maximise entropy, subject to the *Representative Probability* property. This strongly defines the computational complexity of the problem because the number of orderings is the factorial of the number of candidates. As a variant of MEV, we consider agent combinations, instead of permutations for the ordering set $\mathcal{O}$, such that A is divided into a preferred group $\mathcal{P}$, of cardinality K , and non-preferred group $\mathcal{NP}$. Then K is the number of winners needed and $N = |A|$. Hence, the cardinality of the ordering set decreases from $N!$ to $\frac{N!}{K!(N-K)!}$. For small K , this leads to a drastic reduction of the computational complexity. We call this variant of the original MEV, C-MEV (Combination MEV).

To begin, for each ordering $o \in \mathcal{O}$, the administrator can define a pairwise preference matrix $S(o)$ (in the same way as 3.2), whose entries are computed as follows:

$$S(o)_{j,k} = \begin{cases} 1 & \text{if } a_j \text{ is placed over } a_k \\ 0.5 & \text{if both are in the same group and } j \neq k \\ 0 & \text{if } a_k \text{ is placed over } a_j \text{ or } j = k \end{cases} \quad (3.4)$$

for $a_j, a_k \in A$.

Now let the administrator define an **unknown** probability measure $\pi : \mathcal{O} \rightarrow \mathbb{R}_{\geq 0}$. $\pi(o)$, $o \in \mathcal{O}$, which gives the probability of o being chosen as the outcome ordering. Then, they construct a theoretical preference matrix $S(\pi)$ as follows:

$$S(\pi) := \sum_{o \in \mathcal{O}} \pi(o) \cdot S(o). \quad (3.5)$$

The entry $S(\pi)_{j,k}$ states the probability of the outcome ordering placing a_j over a_k under probability measure π . The definition of *Representative Probability* simply requests that $S(\pi) = S(A)$. With this, the administrator can now proceed to formulate the optimisation problem to solve and thus find the ideal candidate ordering.

Recall that the entropy of π measures the uncertainty of choosing a possible candidate ordering, o , from $\mathcal{O}$. The entropy of π is $-\sum_{o \in \mathcal{O}} \pi(o) \log \pi(o)$ [225], and the uniform

$\mathcal{O}$	o_1	o_2	o_3
Preferred $\mathcal{P}$	(a_i)	(a_j)	(a_k)
Non-Preferred $\mathcal{NP}$	(a_j, a_k)	(a_i, a_k)	(a_i, a_j)

Table 3.1: The lower-carnality ordering set when $A = \{a_i, a_j, a_k\}$ and one winner is selected. Agents in the same brackets are given the same rank in an ordering.

distribution is the one with the maximum amount of entropy. Since the aim is to maximise entropy, the election administrator can compute the election outcome by solving the following optimisation problem:

$$\begin{aligned}
\pi^* = \max_{\pi} & - \sum_{o \in \mathcal{O}} \pi(o) \log \pi(o) \\
\text{s.t.} & \sum_{o \in \mathcal{O}} \pi(o) \cdot S(o) = S(A) \\
& \sum_{o \in \mathcal{O}} \pi(o) = 1 \\
& \pi(o) \geq 0 \quad \forall o \in \mathcal{O}
\end{aligned} \tag{3.6}$$

The solution π^* , returns the probability of selecting each ordering $o \in \mathcal{O}$. The result of the C-MEV election computed by randomly sampling an ordering, o^* , from the probability distribution π^* .

For example: let us consider an election where one winner is selected out of candidates $A = \{a_1, a_2, a_3\}$. If the probability distribution π^* , computed following 3.6, states that: $\pi^*(o_1) = 0.5$, $\pi^*(o_2) = 0.3$ and $\pi^*(o_3) = 0.2$, where each ordering is defined in Table 3.1, then the election will return agent a_1 as the winning candidate with the highest probability. In other words: C-MEV will return the most moderate candidate, whilst ensuring that the outcome adheres to the *Representative Probability* property.

Remark. *Quan Zhou proposes a methodology to solve this optimisation problem in a computationally more efficient manner, which can be found in the Appendix of the first version of this work uploaded on ArXiv, on pages 24-26 [279].*

Scaling C-MEV for More Candidates

In this scheme, K , the number of winners chosen, needs to be kept small to maintain the computational complexity of C-MEV low. However, in order to increase K , we can perform C-MEV iteratively, J many times. This can be done as follows:

- Solve C-MEV and find the preferred group $\mathcal{P} = \mathcal{P}_1$, of cardinality K .
- Obtain $A_1 = A \setminus \mathcal{P}_1$.
- From A_1 find the new preferred group $\mathcal{P}_2$, of cardinality K .

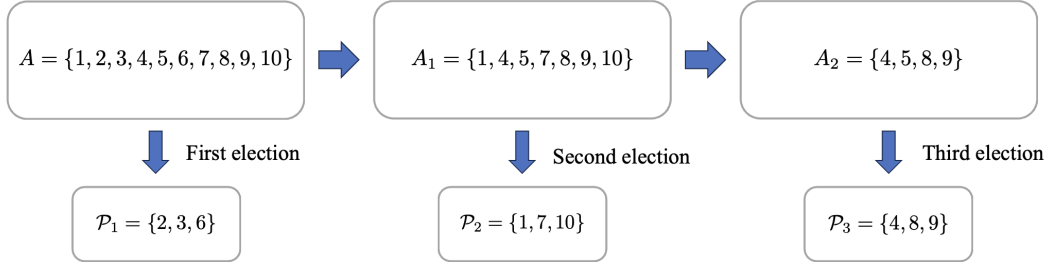


Figure 3.2: Example of an election performed with C-MEV with three winners chosen and three rounds.

- Repeat the above three steps J times.

Figure 3.2 depicts a simple example of this process, with three rounds ($J = 3$) and three chosen winners ($K = 3$). This iterative version of C-MEV allows us to obtain J groups of K many preferred candidates. Notice that this is a decrease in computational complexity with respect to solving C-MEV for groups of $K \cdot J$ candidates. To show that, define $\mathbb{C}(i, j)$ as the complexity to solve C-MEV for j groups of i candidates. Then, the complexity to solve C-MEV for a group of $K \cdot J$ candidates is $\mathbb{C}(K \cdot J, 1) = \frac{N!}{(K \cdot J)!(N - K \cdot J)!}$, whereas the computational complexity to solve C-MEV, J times for groups of K candidates is $\mathbb{C}(K, J) = \sum_{i=0}^J \frac{(N - K \cdot i)!}{K!(N - K \cdot (i + 1))!}$. Then it can be shown that

$$\mathbb{C}(K, J) = \sum_{i=0}^J \frac{(N - K \cdot i)!}{K!(N - K \cdot (i + 1))!} < J \frac{N!}{K!(N - K)!} < \frac{N!}{(K \cdot J)!(N - K \cdot J)!} = \mathbb{C}(K \cdot J, 1)$$

which proves that iterative C-MEV leads to a decrease in computational complexity.

For a step by step summarised election process, the reader may refer to Appendix A. In it, the election steps are clearly outlined in order, for ease of exposition.

3.7.3 Data Consensus

Once a set of agents have been elected to aggregate data from a given location quadrant, they then proceed with computing this aggregation. Typical methods to aggregate data include computing the mean, median or mode. Recall we assume there is an oversupply of data, thus we consider that an aggregation of this data is necessary. In other scenarios where this oversupply is not the case, this stage can be simply replaced with the entrusted agents posting the final dataset.

The purpose of this step is to filter out data points that have been submitted either by malicious agents or agents with faulty sensors. The underlying assumption is that the majority of the agents participating in the data collection are honest and will not report fraudulent or fake data. However, we test this method with large shares of dishonest agents to measure its resilience in adversarial environments. It is also assumed that agents in the same geographic area measure similar results within margins of measurement precision.

Therefore, by aggregating agents' measurements, the aim of the data consensus stage is to resolve conflicts in diverging results. Moreover, it should be possible to compute this aggregation in a privacy-preserving manner: this prevents agents from stealing valuable data points from each other.

The mode is unsuitable for continuous data where every value could be unique. Under light-tailed conditions¹³, the mean can be considered a more faithful aggregator than the median, as it integrates contributions from every data point rather than relying solely on order statistics. However, the mean is very sensitive to outliers. The median is a more robust estimator, but harder to compute privately than the mean [172]. We choose the median-of-means as a suitable compromise: it retains the robustness to outliers characteristic of the median, while still aggregating contributions from all data samples like the mean. The median-of-means is computed as follows: given n independent samples $x_1, \dots, x_n$, we first partition them into k groups of (approximately) equal size, each containing $m \approx n/k$ samples. For each group G_j , we compute its empirical mean:

$$\hat{\mu}_j = \frac{1}{m} \sum_{i \in G_j} x_i, \quad j = 1, \dots, k.$$

The median-of-means estimator is then defined as the median of these group means:

$$\hat{\mu}_{\text{MoM}} = \text{median}(\hat{\mu}_1, \hat{\mu}_2, \dots, \hat{\mu}_k)$$

Robustness of Estimators

The robustness of an estimator can be formally quantified through its *breakdown point*. For a sample of size n and an estimator T , the finite-sample breakdown point (FSBP) is defined as:

$$\varepsilon_n^*(T) = \frac{m^*}{n} \tag{3.7}$$

where m^* is the largest number of sample points that can be replaced by arbitrary values without driving the estimator to $\pm\infty$. The *asymptotic breakdown point* (ABP) is given by:

$$\varepsilon^*(T) = \lim_{n \rightarrow \infty} \varepsilon_n^*(T) \tag{3.8}$$

The Mean

The sample mean has an FSBP of $\varepsilon_n^*(\bar{X}) = 1/n$, since replacing a single observation by an extreme outlier suffices to move the mean arbitrarily far. Consequently, the ABP is

$$\varepsilon^*(\bar{X}) = \lim_{n \rightarrow \infty} \frac{1}{n} = 0$$

indicating that the mean is highly non-robust.

¹³Meaning that extreme values are less likely. This is the expected in our case. Unless adversaries are including outlier data, we expect data from the same location and time to be largely similar and have a low number of outliers.

The Median

In contrast, the sample median requires contamination of approximately half the data to break down. Its FSBP is:

$$\varepsilon_n^*(\text{Med}) = \frac{\lceil n/2 \rceil}{n}$$

and its ABP is:

$$\varepsilon^*(\text{Med}) = \frac{1}{2}$$

which is the maximal possible breakdown point for any estimator of location [165].

The Median of Means

The median-of-means (MOM) estimator divides the data into k groups, computes the mean within each group, and then takes the median of these group means. To corrupt the median-of-means, at least half of the groups must be contaminated, and a group can be corrupted by altering a single observation within it. Thus the FSBP is approximately:

$$\varepsilon_n^*(\text{MOM}) = \frac{\lceil k/2 \rceil}{n}$$

while the ABP satisfies:

$$\varepsilon^*(\text{MOM}) = \lim_{n \rightarrow \infty} \frac{\lceil k/2 \rceil}{n} = 0$$

Hence, while median-of-means has greater finite-sample robustness than the mean, it does not retain a positive asymptotic breakdown point.

3.7.4 Access Control Mechanism

We propose a novel access control mechanism to regulate entry to sell data in the market. Let us recall first that a spatial coalition, C_q , is formed by agents within the same location quadrant, q , where $C_q = \{a_1, a_2, a_3, \dots\}$. Each agent $a_i \in C_q$ measures a data point x_i at a given time t . Agents in C_q then agree on an objective function, $v_{C_q}(\cdot)$ which determines the utility of their dataset. A committee of agents is elected from agents in C_q using Combination-Maximum Entropy Voting, following section 3.7.2. This committee aggregates data points provided by agents in C_q using an aggregation algorithm outlined in 3.7.3. The resulting aggregated data point is denoted as X_t , where $X_t = \text{median-of-means}(x_1, x_2, x_3, \dots)$. It is then possible to calculate how valuable data points are with respect to said objective function. This is done by computing the Shapley value, $\psi(X_t)$. The Shapley value is used to measure the marginal contribution of data point towards maximising the objective function, and is shown to satisfy notions of fairness [450]. Agents in C_q must complete a proof-of-work that is inversely proportional to the Shapley value they received for their data point contribution. The more valuable their data, the less work they must perform.¹⁴ Indeed, this assigned proof-of-work consists of computing the Shapley value of the next incoming data point(s) wishing to enter the data market, $\psi(X_{t+1})$. This work enables the functioning of the market. More specifically, we

¹⁴In this context, more work would involve computing the Shapley value of a greater number of data points.

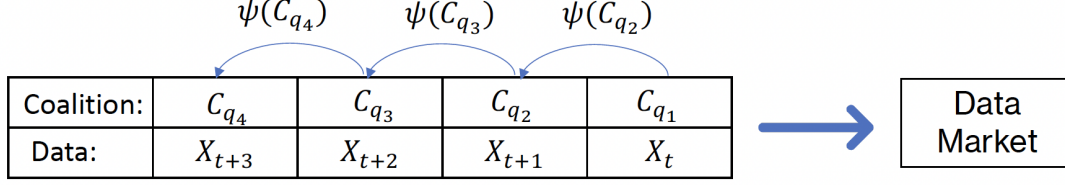


Figure 3.3: Access Control Mechanism: Agents in coalition C_{q_1} must compute the Shapley value, $\psi(C_{q_2})$, of the new incoming data, X_{t+1} , using X_{t+1} and $v_{C_{q_1}}(\cdot)$. Once the agents of C_{q_2} receive their Shapley value, they must complete an amount of proof-of-work that is inversely proportional to their Shapley value. This work is to compute the Shapley value of $\psi(X_{t+2})$ and so on. Once a coalition completes this work they may enter the market.

make use of smart contracts¹⁵ to automate this process. In our context, a smart contract will be executed by an agent a_i to compute the Shapley value of agent a_j 's data. The outputs will be the Shapley value of agent a_j 's data and a new smart contract for agent a_i , recording proof of a_i 's completed work.

A contextual example

We illustrate an example of crowd-sourcing pollution measurements with real data. We showcase how the Shapley value would be used to rank the data points in terms of value, and allocate a proportional proof-of-work correspondingly.

We use the data on pollution levels of a range of different contaminants taken from a number of cities in India. The data has been made publicly available by the Central Pollution Control Board.¹⁶ The cleaned and processed data was accessed from ¹⁷. We illustrate an example wherein a buyer is interested in purchasing data on pollution levels of different contaminants in order to predict the Air Quality Index (AQI) of a given location. We generate a linear regression model to predict AQI, which has been previously done in [470], although other options for models to predict AQI have been explored in alternative works such as [20] and [300]. This model will serve as the objective function.

A description of how AQI is calculated can be found in ¹⁸. Following from this calculation, it is reasonable to observe how the variables PM2.5 (Particulate Matter 2.5-micrometer in $\mu g/m^3$) and PM10 (Particulate Matter 10-micrometer in $\mu g/m^3$) are highly correlated with AQI. This is corroborated by the results in Figure 3.4. We include them as well as NO, NO2, NOx, NH3, CO, SO2 and O3 as training features for the linear regression model.

Agents collecting measurements of different pollutants have their data points evaluated by a preceding set of agents that must calculate some proof-of-work. This proof-of-work is computing the Shapley value of a data point. To do so, they utilise the given objective

¹⁵A smart contract is a program that will automatically execute a protocol once certain conditions are met. It does not require intermediaries and allows for the automation of certain tasks [123][487].

¹⁶Source: Central Pollution Control Board.

¹⁷<https://www.kaggle.com/datasets/rohanrao/air-quality-data-in-india>

¹⁸Source: Central Pollution Control Board.

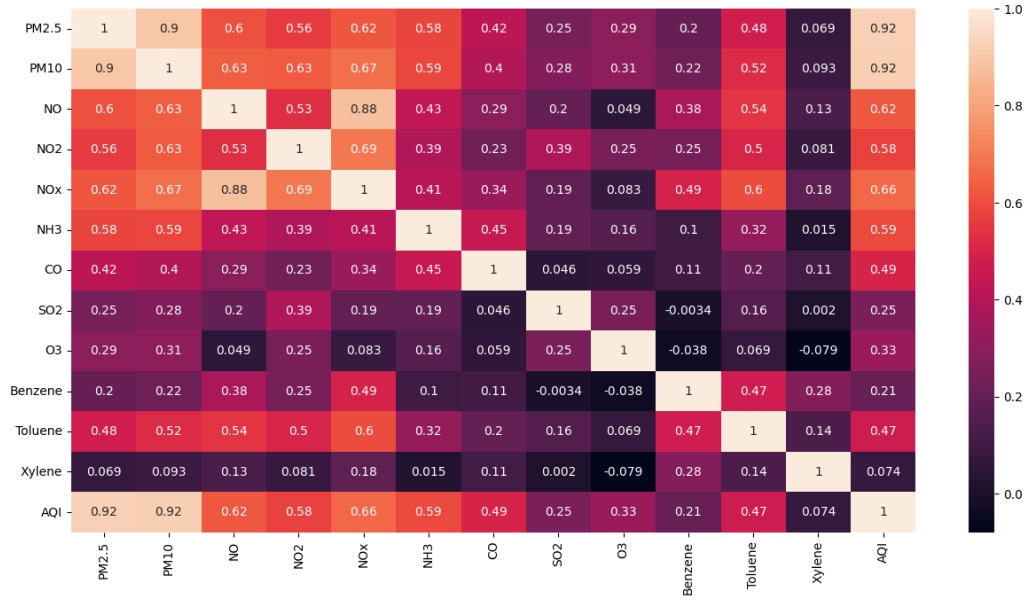


Figure 3.4: Correlation matrix of the different contaminants with the Air Quality Index.

function, which in this case is the linear regression model, and the agent's data.

We show the results of calculating the Shapley value of individual data points within a given dataset in Figure 3.5. We simulate this using the SHAP library, presented in [331]. Following from the SHAP documentation: *"Features pushing the prediction higher are shown in red, those pushing the prediction lower are in blue"* [330]. By visual inspection of Figure 3.5 we can conclude that data points measuring high PM2.5 and PM10 concentrations increase the predicted AQI the most.

Consequently, the agents having provided those data points would receive the highest Shapley values, and thus have to perform less proof-of-work. In this context, that would mean computing the Shapley value of a smaller number of new incoming data points.

We also show the mean absolute value of the SHAP values for each pollution contaminant in Figure 3.6. From this we can deduce that any measurement belonging to the highest SHAP value classes will be deemed more worthy and thus the agent submitting it will have to perform less proof of work to sell it. Every spatial coalition C_q would have their own combined Shapley value, which is the total of the Shapley values of each data sample in that coalition's dataset.

Remark. *The reader may rightly question the privacy risks of an agent accessing another one's dataset to compute the Shapley value. What is to incentivise them to compute the Shapley value honestly, and what is to prevent them from stealing or duplicating another agent's data if they realise it has a high Shapley value?*

To address the first concern, the computation of the Shapley value is automated through the use of smart contracts. These can be inspected by anyone to ensure their correct operation.

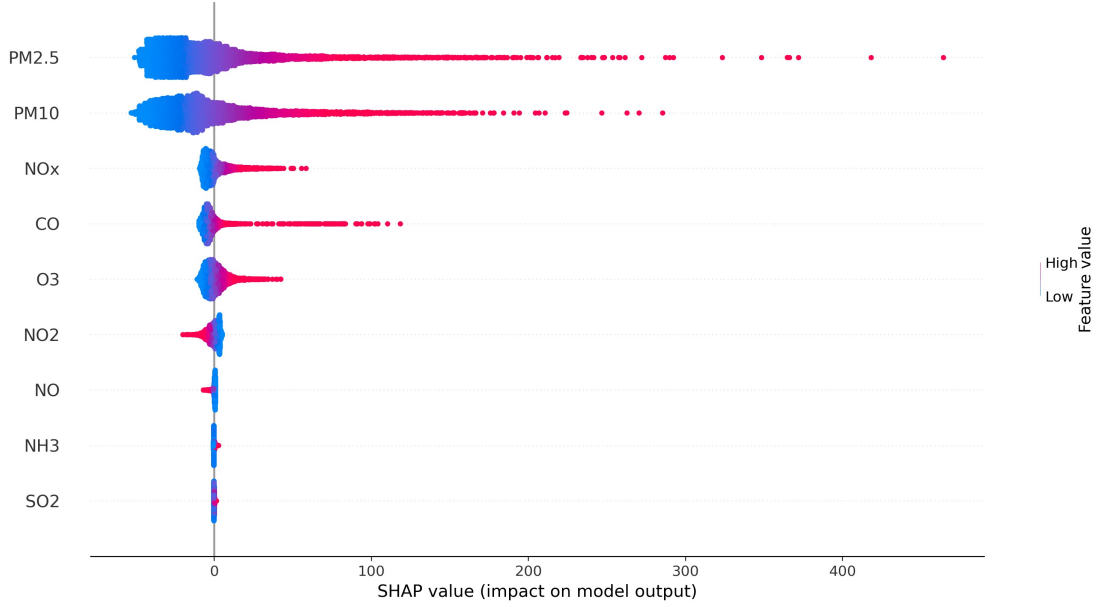


Figure 3.5: SHAP values of each of the data points per contaminant.

Furthermore, [329] and [555] propose ways to compute Shapley values under encryption. This allows the computation of this value without having to reveal the dataset to the agent carrying out said computation.

Secondly, in the market there is no protection against agents duplicating data, but they cannot monetise this copied data unless they go through the verification, consensus and then access control stages again. Because we are in an environment with an oversupply of data and that is crowd-sourced, the data is unlikely to be highly sensitive and thus the incentive to go through these steps is very small. In addition to this, if data duplication is indeed deemed a pertinent issue for the context of a different data market implementation, the Shapley value calculation can be replaced by the Shapley Robust algorithm proposed in [9], for which we have written a Python implementation using the SHAP explainer Python library. This is an adaptation of the Shapley value calculation that will penalise highly similar data, thereby penalising attackers wishing to monetise duplicated data. This implementation can be found in <https://github.com/aidamanzano/DataMarket>.

3.8 Robustness of data market in adversarial environments

In an adversarial, decentralised environment, one must take into account the possibility of attacks on the system. We proceed to describe their nature and how these are mitigated by the algorithmic components of the data market architecture.

Sybil Attacks are a type of attack in which an attacker creates a large number of pseudonymous identities which they use to exert power in and influence over the network [166]. Sybil attacks are mitigated in the verification stage, as agents must present a valid proof of identity. This proof is granted to them through a centralised authority but

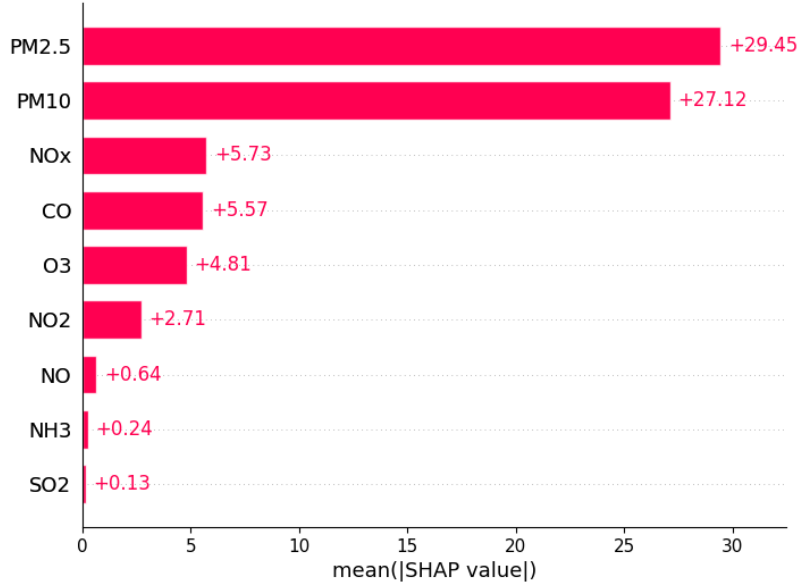


Figure 3.6: Mean of the absolute SHAP values, aggregated per contaminant.

all other agents can verify that it exists and therefore that it must be valid. In the context of smart mobility applications, generating multiple identities is made expensive because agents must provide a valid vehicle license plate to enter the market and sell data.

Wormhole attacks involve a user maliciously reporting they are in a location that is not the one they are truly in [247]. An attack can be mounted by a series of malicious actors claiming to measure data from a location they are not truly in, and wishing to monetise this fraudulent data. To mitigate against this attack, agents must present a valid proof of position in the verification stage. This proof is assumed to be correct and sound, and by definition, agents are only able to present one valid proof.

Data poisoning is an attack where malicious agents collude to report fake data in order to influence the agreed upon state of a system [554]. Malicious agents wishing to report fake data must influence enough agents in their spacial coalition to ensure that sufficient agents in the data consensus mechanism get elected to compute a fake data point. Probabilistic voting schemes make the cost of this coercion significantly high. Furthermore, to sell the uploaded data point, the agent must first perform a useful proof-of-work that is proportional to how valuable the data point is deemed. The less useful the data point the more work the agent must complete to sell it. Selling spam data will therefore be very time consuming for an attacker.

Remark. *The reader may also question the resilience of this protocol to inaccurate data measurements. Indeed, some agents may have faulty sensors and collect inaccurate but not malicious data. This may include not only the value of the data but the geolocation of the data as well. This scenario is also considered in our protocol. There are three possible ways in which inaccurate data may arise: the inaccurate data is either vastly divergent from the*

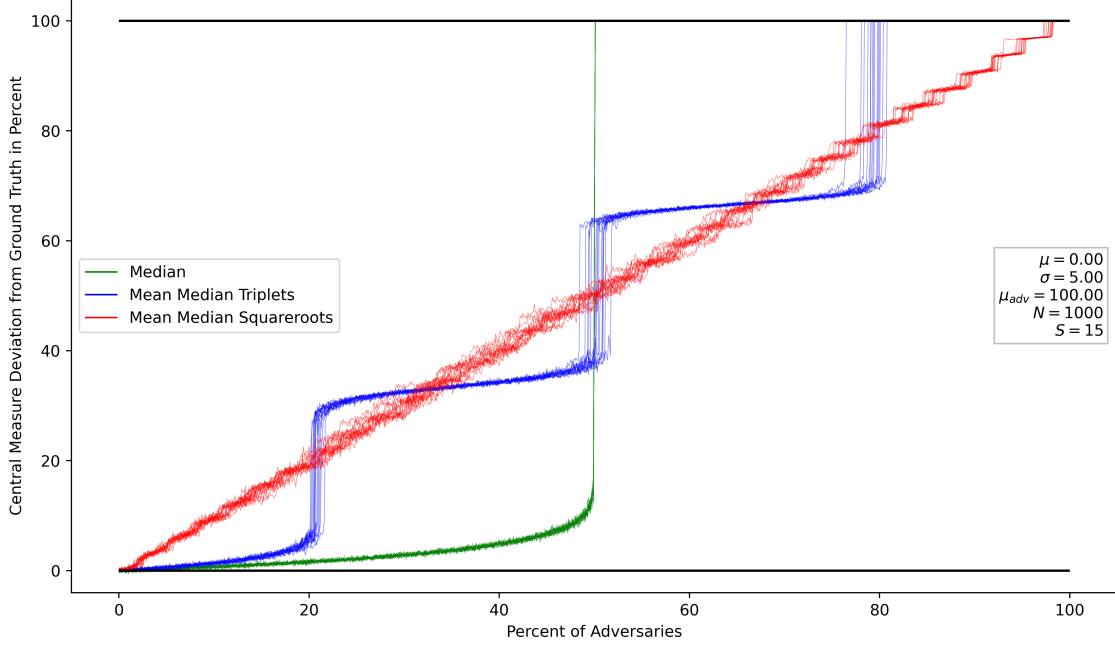


Figure 3.7: Characterisation of data consensus algorithms’ behaviour under different degrees of coordinated data poisoning attacks.

accepted value, or it is similar to the accepted value but slightly noisy, or reported from a wrong location. In the first case, the data point will be filtered out in the voting and the data consensus protocol. In the second case, the data point is sufficiently similar such that it will not affect the final dataset sold, and in the final case, the proof of position protocol prevents this from happening.

As the Sybil and the wormhole attack are not possible due to the design of the data market, we proceed by showcasing results that illustrate the robustness of the data market with respect to the data poisoning attack.

3.8.1 Simulation Setup

We simulate a set of N agents, each tasked with measuring the same physical phenomenon. Honest agents produce samples from a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$, where μ represents the true underlying value, while malicious agents collude by always reporting a fixed adversarial value μ_{adv} to mimic a data poisoning attack. Each agent is also assigned a reputation: with probability $1/2$ the reputation equals 1, otherwise it is drawn from $\mathcal{N}(\mu_{rep}, \sigma_{rep}^2)$ with $\mu_{rep} \gg 1$, in order to model a population of low-reputation agents attempting to sell inaccurate data. To aggregate the reports we employ the median-of-means estimator, comparing two grouping strategies: fixed group size $G = 3$ (triplets), and group size $G = \sqrt{N}$. For each configuration we perform S Monte Carlo trials, varying both the total number of agents N and the size of the adversarial coalition.

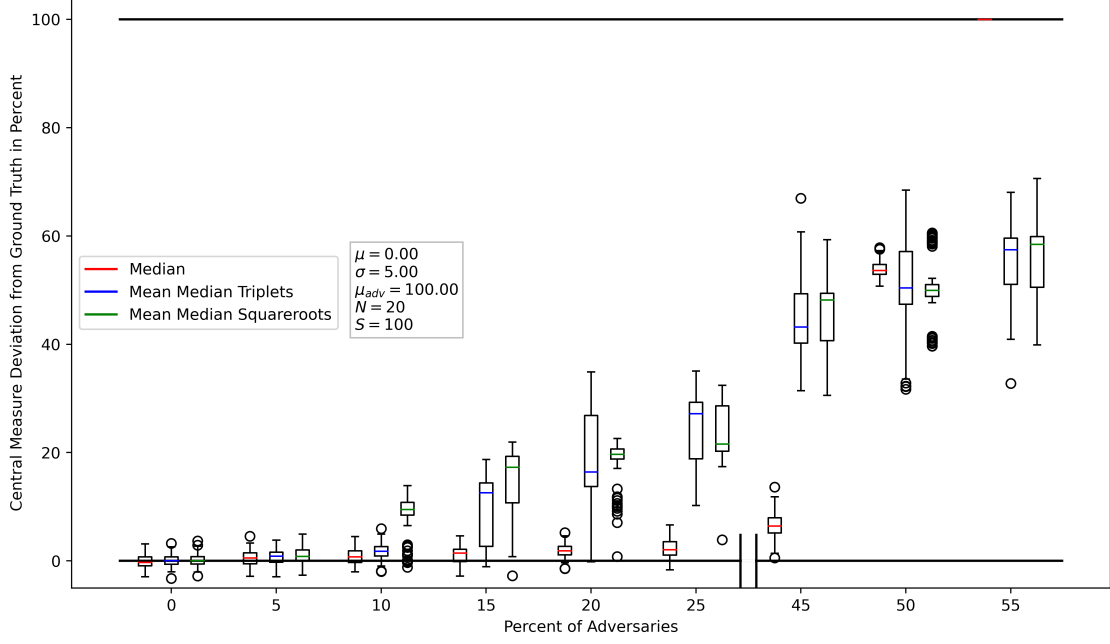


Figure 3.8: Breakdown analysis of data consensus algorithms, with a coordinated data poisoning attack.

3.8.2 Evaluation of Results

Figure 3.7 shows simulations of the median and median-of-means algorithm described in Section 3.7.3.¹⁹ The median-of-means algorithm can have a variable number of agents in each group, G , and there are a total of N agents. We measure the performance of the median-of-means algorithm with $G = 3$ (i.e., 3 agents per group) in blue, and $G = \sqrt{N}$ agents per group, in red. We perform 25 simulations with 1000 agents per simulation ($S = 25$ and $N = 1000$).

As expected, it can be observed that when the percent of adversaries in the network increases, the deviation of the reported data point from the ground truth increases. For the median algorithm, this deviation is significant when the percent of adversaries in the network is 50%, which is consistent with its finite-sample breakdown point of $1/2$. For the median-of-means algorithm with $G = 3$ (3 agents per group), there are three different percentages at which a significant deviation from the ground truth can be observed. These steps coincide with the expected number of malicious agents in the coalition increasing by one agent each time. For example, when the percentage of adversaries is below 20%, the expected number of malicious agents in the coalition is less than one, meaning that it is most likely that all agents within the sampled triplet groups will be honest. However, for a percent of adversaries between 20% and 50%, it is most likely that at least one out of three is malicious, and so on.

For the $G = \sqrt{N}$ case, the linear deviation is likely explained by the expected number of corrupted groups. For an adversarial fraction, p , the probability that all agents in a group

¹⁹In both Figures 3.8 and 3.7, the median-of-means algorithm is labelled as Mean Median.

are honest is:

$$\Pr[\text{all honest}] = (1 - p)^{\sqrt{N}}.$$

Therefore, the probability that a group is corrupted (i.e. contains at least one adversary) is the complement event:

$$\Pr[\text{corrupted}] = 1 - (1 - p)^{\sqrt{N}}$$

We know that each group is of size $\sqrt{N}$. If there are N agents in total, then there are $k = \frac{N}{\sqrt{N}}$ many groups, which is simply $\sqrt{N}$. Therefore, the expected number of corrupted groups is:

$$k \cdot \Pr[\text{corrupted}] = \sqrt{N} \cdot 1 - (1 - p)^{\sqrt{N}}$$

If we linearly approximate $1 - (1 - p)^{\sqrt{N}}$ using the binomial approximation and retaining only the linear term in p (equivalently, applying a first-order Taylor expansion around $p = 0$) we obtain that $1 - (1 - p)^{\sqrt{N}} \approx p\sqrt{N}$.²⁰ Thus:

$$\mathbb{E}[\#\text{corrupted groups}] = k \cdot p\sqrt{N} = \sqrt{N} \cdot p\sqrt{N} = pN$$

This approximation holds for small p , which explains the linear relationship between deviation and an adversarial share of $p \approx [0, 0.2]$. For a larger p , the aforementioned approximation should not be used, as the amount of error from that approximation would increase significantly. However, it can be observed that as p increases, the relationship still appears to be linear, but more noisy. I am not sure what causes this behaviour. I have considered using a different approximation for the expression $1 - (1 - p)^{\sqrt{N}}$ that is suitable for larger p , such as a first-order logarithmic Taylor series approximation. This would yield an approximation of $1 - (1 - p)^{\sqrt{N}} \approx 1 - e^{-p\sqrt{N}}$. However, this approximation would indicate an exponential increase in the probability of a group being corrupted after $p \geq 0.2$. Observing Figure 3.7, the deviation scales linearly, not exponentially, even for larger p . This is likely because although the rate at which the expected number of corrupted groups increases may be exponential, that does not necessarily mean that the absolute deviation in the measured data point needs to be exponential too. Nonetheless, the results show that the more adversaries, the higher the number of corrupted groups in the median-of-means algorithm, which in turn, increases the deviation. Further work should uncover the relationship between deviation increase in the measurement, and the expected number of corrupted groups.

It can be observed that in both cases, the median-of-mean algorithm is contaminated with a lower share of adversaries than the median. So the median-of-means algorithm is not contamination-robust in the breakdown sense. Rather, its strength is variance reduction in the presence of a high share of adversaries.

We then investigate the behaviour of the algorithm when smaller numbers of agents are present (which also represents more realistic scenarios), the same simulation was carried

²⁰We approximate $(1 - p)^{\sqrt{N}}$ using the binomial approximation, which yields $1 - p\sqrt{N}$. Therefore, substituting back into the expression, $1 - (1 - p\sqrt{N})$ yields $p\sqrt{N}$.

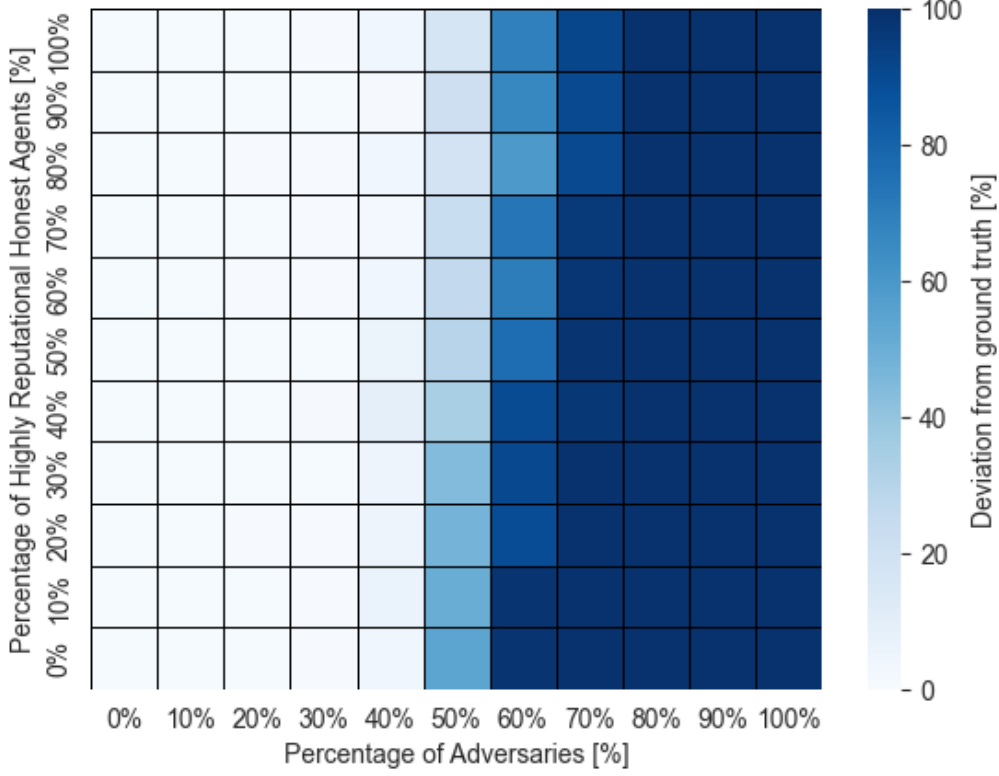


Figure 3.9: Characterisation of breakdown of C-MEV combined with median-of-means algorithm

out but with $N = 20$ agents, as depicted in Figure 3.8. The number of Monte-Carlo simulations is increased to $S = 100$ to provide meaningful statistical results. From the results shown in Figure 3.8, it can be observed that the algorithms act similarly in the range of 0 to 15% of adversaries, although the median outperforms the other algorithms in the interval of 15% to 50% of adversaries. This is expected, as the median is the most robust estimator, however, it breaks down at the 50% share of adversaries (as mentioned earlier, due to its finite-sample breakdown point). The deviation amount relative to the adversarial share is generally consistent with the results in Figure 3.7, reaffirming that the median is more robust in less adversarial scenarios, but less so when the adversarial share is higher than $1/2$. The median-of-mean algorithm offers proportionality in deviation given a share of adversaries, avoiding the exponential shoot in deviation in a highly adversarial scenario.

Finally, Figure 3.9 represents how much the reported data deviates from the ground truth, when combining the reputation-based C-MEV voting scheme and the median-of-means algorithm. The y-axis represents the percentage of highly reputational agents within the set of honest agents, and the x-axis represents the percentage of adversaries. The number of Monte-Carlo simulations is set to $S = 100$. We sample the reputation for the high-reputation honest actors, from a Gaussian distribution $\mathcal{N}(\mu_{rep}, \sigma_{rep}^2)$ where $\mu_{rep} = 100$ and $\sigma_{rep} = 30$. In this simulation, the voting scheme (C-MEV) outputs 5 groups with 3 agents in each group ($J = 5, K = 3$), out of a set of $N = 30$ agents. The chosen agents

then aggregate data points to provide the agreed upon dataset of a given location, using the median-of-means algorithm.

Visual inspection of the proposed results lead to two main observations: (i) combining C-MEV and median-of-means algorithm has a higher breakdown point than data consensus alone; (ii) an increased share of highly reputational agents among the honest ones leads to an increase of the breakdown point. This is because, the higher the reputation of a honest agent, the more likely they are to be selected, thus decreasing the chances of malicious agents being elected. The benefit of including C-MEV is especially evident when the share of adversaries is below $1/2$, leading to a drastic decrease in deviation from ground truth. We can observe from Figures 3.7 (and 3.8) that without C-MEV, the deviation was approximately linearly proportional to the share of adversaries, but including C-MEV, this deviation drops to 0 when the percent of adversaries is between 0 and less than 50%.

In summary, it can be concluded that combining the voting algorithm and the median-of-means aggregation method, the system offers strong protection against *data poisoning* attacks. To succeed in mounting a *data poisoning* attack, the malicious coalition must control from more than half of the network, assuming a functional reputation system exists.

3.9 Conclusions

We have presented a fair, decentralised data market architecture that is robust to a number of attacks. The novelty of this work includes: ranking data in terms of how valuable it is using the Shapley value, and using it to proportionally adapt the proof-of-work each agent must complete. Furthermore, the proof-of-work is itself useful and necessary for the functioning of the market, and thus not wasteful. We also utilise a voting scheme that satisfies desirable properties of fairness, and introduce an optimisation to make its computational complexity significantly lower for the context of this work. We evaluate the resilience of the median-of-means algorithm combined with the voting mechanism against *data poisoning* attacks. Our simulations show a drastic increase in robustness when including the voting mechanism, especially for lower adversarial shares. They also show an increase in robustness in highly adversarial scenarios compared to using the median alone.

3.10 Critical evaluation of this work

Thus far, I have discussed the technical limitations of this work and what it does and does not deliver. I want to take this section to critically evaluate the success of this work, and whether it achieved what I was motivated for it to achieve. My motivation to pursue it was as described above, to democratise data ownership and reward. To challenge the status quo of some people doing the ugly, hard and un-glamorous work of collecting and cleaning data, and getting underpaid for it, or simply this data being conceded for free, unknowingly scraped by some large multinational without fair recompense. I perhaps rather naively thought that the reason why this is the current state of affairs was due to

technical limitations. At the time of writing, there was no decentralised data market design I could find that solves the technical challenges to enable decentralised transaction of data and fair reward to the user. And whilst this contribution is novel, and it addresses technical barriers to enable decentralised data transacting, I believe the cause of this problem is not due to technical limitations.

The question is much more simple: why would a large multinational pay for data when they can just scrape it for free? The incentive for the data provider to use this is very clear: getting a fair reward for their data. But for the company? Why would these monopolies or oligopolies that own the field, have unmeasurable amount of financial power ever bother to act more ethically and source data fairly? The answer to that question I could not resolve with a technical solution.

To give this work a fair shot, a number of decentralised data markets have recently emerged in the time after this work was developed, and some even crediting our contributions [80, 43, 44, 175]. These are more extensively discussed in Section 3.4. There have also been some data consortiums born out of regulatory needs such as [94], [171], [131]. This is perhaps a signalling that there *is* a desire for people to reclaim ownership and fair reward of their data.

There is a more evident incentive for a company to use this kind of structure when they require some guarantee that the data purchased is actually *real*. Also, another incentive stems from companies being subject to further regulation and needing to prove what they do and do not do with acquired data [520]. Whilst this is encouraging, the reality is that these unethical practices and monopolies have not yet been challenged by decentralised data architectures, and further, perhaps non-technical research may be necessary to address this problem.

Chapter 4

Proof of Position Algorithms

Abstract

This chapter describes a proposal for a class of decentralised proof of position algorithms called Tree-Proof-of-Position (T-PoP). T-PoP algorithms rely on a network of interconnected devices to establish how likely it is that an agent is in their claimed position. The algorithms are able to operate under highly adversarial conditions, with probabilistic guarantees of security and reliability. We do not assume that agents are honest for the algorithms to function, and we consider multiple ways in which agents may misbehave. Most of the current proof of location proposals are centralised, thus forgoing verifiability and privacy for the users. Decentralised solutions also exist, but they suffer from drawbacks that make them unsuitable for realistic, adversarial use cases. In this chapter, a preliminary theoretical model for T-PoP and its security properties is presented, and is validated through a large number of Monte-Carlo simulations. We provide a case study of two specific instances of T-PoP which we test, and analyse their security and reliability properties under a range of adversarial conditions.

4.1 Contributions Disclosure

The work presented in this chapter was published in [281] and further developed in [344]. For the first publication, my specific contributions are as follows: I conceived and implemented the algorithm, developed the agent-based simulator, defined the agent behaviours, co-developed the graphical probability model, and conducted a comprehensive survey of the related literature. I am first author on all sections, with the exception of the Jensen–Shannon Divergence evaluation section. For the second publication, I authored every section except Conditions 2 and 3 (pages 10 and 11). I developed the equations and coded the simulators of the entire work, with guidance and collaboration of my supervisors.

4.2 Motivation

In the previous chapter, a proposal for a decentralised data market was described. This architecture relies on a functional proof of position mechanism, which is described in Chapter 3, Section 3.7.1. This component of the data market is included to prevent agents from falsifying the provenance of their data. A proof of position mechanism would determine if the agent is truly where they claim to be, although it would not prevent an agent with a valid location proof from submitting data from elsewhere. However, other components in the data market are in place to hinder this possible attack, namely, the consensus and access control mechanisms. Having explored existing solutions for proof of position mechanisms, I found the options unsuitable for this particular use case for varying reasons. In some cases, the trust assumptions of honesty were unrealistic, in others, the solution was centralised. Hence the motivation to develop a proof of position algorithm that is decentralised, does not assume honest behaviour of participants, is compatible with privacy preserving techniques and can be adapted to provide desired levels of security and reliability.

Aside from being a necessary component of the aforementioned data market, there are a number of commercial applications that rely on providing location-based services. Some examples include Waze [21], Uber [140], Google Reviews and streaming services such as Netflix [325]. In these instances, it is of critical importance that the claimed location of their users can be ascertained. Otherwise, participants may lie to gain undue advantages. For example, participants in Waze have been known to falsely report high levels of congestion to divert traffic away from their neighbourhood [458]. Similarly, an anecdotal example in [53] shows how these location-based services may be targetted: in this example, an artist wheeled a cart of 99 phones which faked a traffic jam in Google maps. Another perhaps more serious application of location proofs is reporting on conflict. In these scenarios, the spread of misinformation and deep-fakes has become a growing concern, especially with the recent rise of generative AI [557], [534] and social media [549], [368].¹

Freedom of press and access to information is a human right, as stated in Article 19 of the Universal Declaration of Human Rights [506]. How we choose to believe information I theorise is built on: evidence and consensus. Evidence requires verifiability, and not just for a select few. During conflict, access to first-hand sources of information is limited [428], and collecting it is dangerous. Governments may (and do) restrict freedom of press [24], and some even going as far as carrying out targetted killings of journalists [406], [51]. It is in these contexts where data verification is difficult because it is not readily accessible to the average citizen, and events are particularly target to misinformation and censorship from powerful actors. Nevertheless, is in these scenarios where information verification is most crucial. This context therefore provides three critical design criteria: consensus and verifiability are minimum, but often disregarded is decentralisation. Web3 advocates herald

¹Concerning examples include and are not limited to sharing images claiming they are from a conflict when they are in fact from a video game [129], or another conflict that already happened years ago [431], [479].

decentralisation as the solution to all woes [536]. Others (and I) disagree: decentralisation is slow, clunky and introduces complexity [208], [126]. Not all applications are in need of decentralisation, but for reporting in conflict, it is critical. There is a direct conflict of interest with central authorities, especially if they are a belligerent party [2]. In this context, I believe the price paid for decentralisation is worth the benefit, which is reduced susceptibility to censorship.

In cases like this where official sources may not always be trustworthy, the remaining alternative is the accounts of individuals reporting on the ground. How can citizens verify the veracity of these sources? How may they assert these accounts originate from where they claim to be in? One evident method to verify one's location is using Global Positioning System (GPS), however, GPS signals can be easily spoofed and jammed [18], [260]. The Global Positioning System is also owned by the United States Space Force, and its civilian use, although widespread, is subject to government restrictions [288]. In the context of conflicts, GPS has also been targetted to undermine war opponents [393]. Thus, while GPS remains the most prevalent tool for location verification, its vulnerabilities and geopolitical dependencies highlight the need for alternative or complementary proof-of-location mechanisms.

It is also worth noting that databases exist collecting worldwide data on political conflict and violence events. Two notable examples include the ACLED [411] and the UCDP GED [483] databases. Their data is collected by a network of internationally based researchers, who analyse first and second hand sources to populate the datasets. Both projects have clear documentation on what constitutes an event, the inclusion and exclusion criteria and the nature of the sources.² Both datasets have also been used extensively by multiple UN bodies, governments across the world, news outlets, policy-making bodies and NGOs. They acknowledge and clearly disclose their limitations, which is a testament to their scientific integrity. The ACLED for example, has a public table of retracted events³, and the UCDP GED acknowledges that unlike the ACLED, they do not encode non-violent events (such as protests or troop movements) [174]. In both cases, geo-locating data is highlighted as a challenge. For example, the work in [174] that compares both datasets states that: *"fighting often takes place in rural areas, away from human habitation, and it is usually impossible to get precise geo-coordinates for a location out in the bush unless there happens to be a geographical landmark in that place"*. Figure 4.1 shows an example of the British Broadcasting Company (BBC) using ACLED's data, where reporters manually perform cross-referencing with physical landmarks to verify the data provenance. The documentation of the ACLED also outlines how each data point has a location code which: *"dictates the level of certainty of the location recorded for the event. The higher the number, the lower the precision"* [6]. Given the challenges reporters faced with verifying location provenance, I set to answer the following question: what if we could offer provable guarantees of location provenance *before* using conflict data? This is the purpose of the proposed proof of position algorithm: to verify one's location before trusting their ac-

²ACLED methodology, UCDP GED methodology.

³ACLED Deleted events.

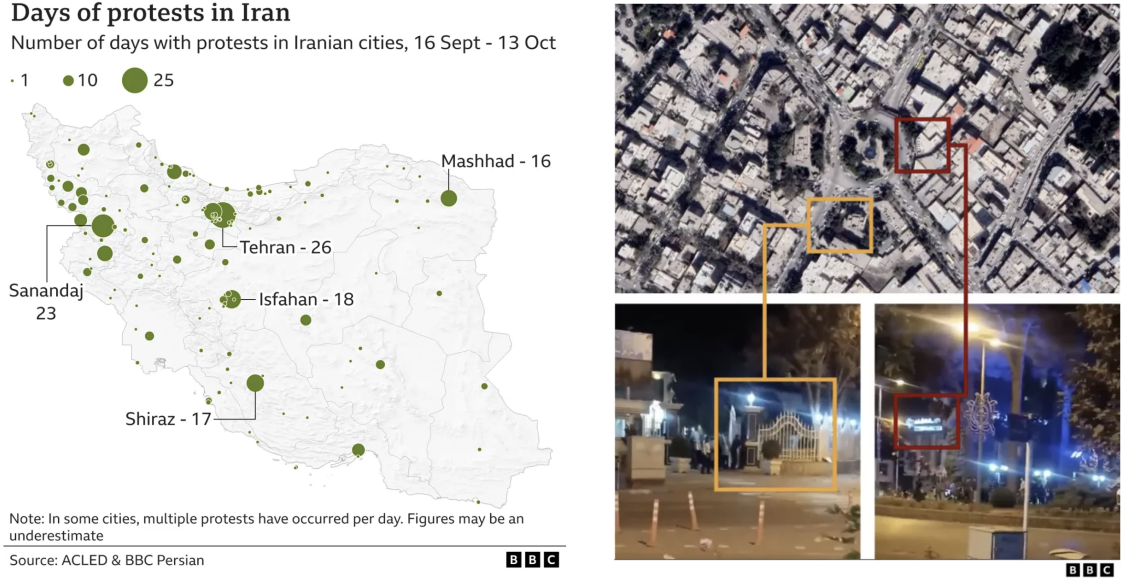


Figure 4.1: An example of the BBC using ACLED data to report on protests in Iran, supplementing the data with landmark cross-referencing for location verification. The article in question is [228]. Reproduced under fair dealing for the purpose of review, © BBC.

count. Our motivation is for the algorithm to operate in a decentralised manner, however, decentralisation alone does not suffice. Information on an individual’s position is sensitive. Revealing it may incur safety risks to the user. Thus, the proof of position solution must also be privacy preserving. Namely, one should be able to provide a verifiable proof of their position, without having to publicly reveal their position to unwanted third parties. In addition, for any solution to be applicable to real-world use cases, it must be designed to withstand adversarial environments and account for individuals that lie and misbehave. In the context of proof of position, there are many ways an agent could lie: they may lie about their own position, but also about the position of others. With these considerations in mind, the objective in this chapter is to provide a proof of position algorithm that is decentralised; private; and robust to adversarial attacks.

4.3 Related Work

An early example of a decentralised proof of location scheme was presented in APPLAUS [561]. The APPLAUS scheme makes a number of valuable contributions, namely it looks to address collusion attacks using graph clustering and computing a ‘betweenness’ metric. Collusion attacks are especially dangerous. It is difficult to discern between a number of honest nodes that are in agreement, and a number of dishonest nodes colluding to lie. In [559], nodes in the graph that are weakly connected are considered less trustworthy. They also present a weight function that decays with time, and compute the trustworthiness of a node by calculating a node’s ratio of approvals to neighbours. These contributions serve as a starting point to the work here presented. However, in their work, users must register their public and private keys with a trusted Certificate Authority. In our work we do not

assume that a centralized authority can be trusted, and indeed we argue that introducing this assumption makes a system de-facto centralised and poses security and privacy risks.

SHARP [556], presents a private proximity test that does not require a user to reveal their actual location to a server, and furthermore, they present a secure handshake method wherein users do not need to have a pre-shared secret. A notable contribution is that a witness may only extract the session key if they are indeed in the vicinity of the prover, thus making it very difficult for a prover to claim to be in a false location. Another important contribution is that indeed, the centralised server is only used to relay messages and does not store private and sensitive user information. The security metric in this work is to ensure that the location tags are unforgeable, thus implying that the protocol is robust towards location cheating. However, non-transferability and non-repudiation must be considered too. A user in a given location can generate a valid proof, but without the aforementioned properties, they could relay this valid proof to a malicious agent that is not in the same location as them. Hence, the work in SHARP, [556], fails to consider that lying about your location is not the only way that a malicious agent can cheat.

Vouch +, presented in [86] is a decentralised approach to prove location, with a focus on addressing high speed and platooning scenarios. The major disadvantage of the work presented is that its security relies on picking a proof provider that is honest. This assumption, in our opinion, is too strong. We aim to develop a protocol wherein the prover could lie, and the system would still have a probabilistic guarantee of detecting this.

The protocol presented in SPARSE, [384], does not allow the prover to pick their own witnesses, making collusion significantly harder. They also do not implement a distance bounding protocol, meaning that the speed of the location proof generation is independent of the user's private key and the cost of implementing their protocol is reduced. Furthermore, SPARSE does address necessary security concerns, and achieves integrity, unforgeability and very importantly: non-transferability. However, similarly to the work of Boeira et al. [86], the prover is assumed to be a trusted entity which supposedly does not publish users' identity and data.

The protocol of Alhamleh and AlQahtani [13] also presents a similar concept to that of APPLAUS, [559], with the session score metric, wherein instead of considering a node's weight, they measure the percentage of valid data points in a location proof. Drawbacks of the work presented include a failure to consider the threat of having a physically compromised infrastructure. This is comprised of a number of access points that provide location data. Should these be attacked, the prover can no longer generate a location proof. Furthermore, there is little discussion on how the pre-shared secret between the access point and the prover is in fact exchanged.

The work of Javali et al. [264] proposes a fast location proof generation protocol, with a robust security analysis of the system. However, similar to [13], they do not consider that an access point may be compromised and collude with malicious actors to generate a fake location proof.

In [11], the Akand et al. present a formal model and security definition using game-based security proofs. They prove their proof of location system satisfies their desired security properties, and consider the scenario where the infrastructure issuing location proofs is tampered.

The work of Amoretti et al. [22] further develops that of Zhu [561] and applies their contributions to a blockchain context. Namely, they utilise the ‘betweenness’ metric to measure a node’s trustworthiness, and observe that a group of collaboratively lying cars may suspiciously, consistently discard proof requests of other honest neighbours in their vicinity. In [90] the authors propose a proof of position protocol using blockchain-agnostic smart contracts, and the work in [539] also suggest blockchain as a platform for the location proofs generated, but introduce the use of Zero-Knowledge Proofs to provide hierarchical privacy, without significantly compromising the performance of the proof generation. A number of other works also propose solutions leveraging blockchain technology, see for example [383] and [517]. However, blockchain is an architecture that tends towards centralisation. This effect has been noticeably observed in Bitcoin [63]. Our protocol does not rely on blockchain, thus avoiding the drawbacks inherent to its design.

4.4 Components

Let us now proceed by introducing the components necessary to understand the functioning of the T-PoP algorithm.

Agents: Any participant in the proof-of-position protocol is considered an agent, a_i , where $a_i \in A$, $|A| = N$ and N is the total number of agents operating in the protocol. All agents are in a given *state*, that determines if they are honest, and coerced, or not. For brevity, we omit the agent’s index where possible.

Environment: We assume that agents willing to participate in the protocol are situated in an environment, E , where $E \subseteq \mathbb{R}^2$ is compact. We note that this does not represent a limitation of the protocol and that any space with a suitable distance metric can be employed for the T-PoP protocol.

Position: All agents have a *real* position, p , and a *claimed* position, $\hat{p}$. Each agent can claim to be in a position different to their real one (i.e.: lie about their position) by committing to $\hat{p}$ such that $\hat{p} \neq p$.

$$\hat{p} \neq p \vee \hat{p} = p \tag{4.1}$$

In other words, the *claimed* position may or may not be equal to the *real* position. We say that $\hat{p}, p \in E$.

Agent states: All agents are in a given *state*, which is comprised of three attributes:

honesty, coercion and position.

- An agent is considered *honest* if they claim to be in their real position. i.e.: $\hat{p} = p$, and *dishonest* otherwise, where $\hat{p} \neq p$.
- All agents are in a real position. *Honest* agents will claim to be in their real position, and *dishonest* agents will claim to be elsewhere.
- An agent is considered *coerced* if they attest to seeing a *dishonest* agent in their claimed (fake) position. In other words: a *coerced* agent will attest to seeing agents that are not actually in its vicinity, if the latter are dishonest. *Non-coerced* agents will only attest to seeing⁴ other nearby agents in their *real* position.

Every $a_i \in A$ is either *honest* or *dishonest*. The set of honest agents is denoted by $H \subseteq A$, and the set of dishonest agents is denoted by $\overline{H}$. Also, every agent $a_i \in A$ is either *coerced* or *non-coerced*. The set of coerced agents is denoted by $C \subseteq A$, and the set of non-coerced agents by $\overline{C}$.

Prover: An agent initiating the proof-of-position protocol, wishing to prove that they are in their *claimed* position, is called a *prover*.

Range of sight: Agents have a fixed, positive range of sight, r_i with $r_i \geq 0$.

Field of view: An agent a_i 's field of view, $\overline{D}_i$, is defined as the closed disk area with centre $\hat{p}_i = (x_i, y_i)$ and radius r_i . We say agent a_j with position $\hat{p}_j = (x_j, y_j)$, is in a_i 's field of view if the position of a_j falls inside $\overline{D}_i$.

Witness: Provided a_j is in a_i 's field of view, a_j may be named as a *witness* of a_i , depending on the *states* of both agents.

Approvals: If a witness a_j attests to seeing the agent that named it, a_i , the witness is said to *approve* a_i . Approvals are always computed from both the agents' *claimed* position.

4.5 Overview of T-PoP

In this section we outline a brief, high-level description of the protocol we propose. It is structured around three stages: **Commitment**, **Tree Building**, and **Verification**. These stages are outlined in Figure 4.2. The protocol is initiated by an agent wishing to prove they are in a claimed position. In the initial stage, **Commitment**, each agent constructs a cryptographic commitment of their claimed position, using a cryptographic

⁴Seeing in this case implies a distance based verification, for example receiving a Bluetooth signal within a physical distance range.

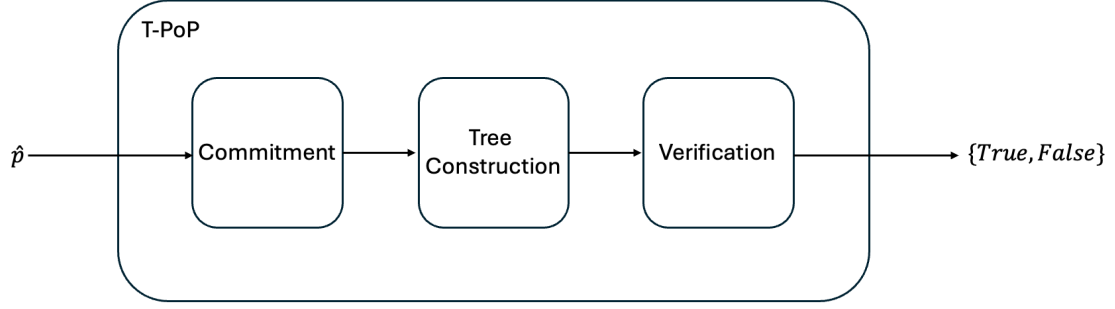


Figure 4.2: Overview of the Tree Proof of Position steps.

commitment scheme⁵. This commitment ensures the position claim cannot be changed *a posteriori*, i.e., it is binding, and is also hiding, meaning the position need not be publicly revealed. Upon constructing the commitment, each agent then uploads their commitment onto a decentralised network. This can be a transaction in a DLT or uploaded to the Inter-Planetary File System, a peer-to-peer file sharing network [408]. In this manner, immutability and verifiability are assured.

Subsequently, the **Tree Building** stage follows. It may only be computed by an agent that has already committed to a position in the **Commit** stage. Said agent is considered the prover. The prover constructs a tree by naming other nearby agents, which are considered the prover’s witnesses. Each node in the tree is an agent, where the prover is the root of the tree, and the witnesses are the leaves. The tree’s dimensions are predetermined *a priori*, namely, the height and the branching factor. The protocol imposes strict constraints to prevent naming the same witness twice in the same tree, aiming to mitigate deceitful practices (e.g., agents cyclically validating each other).

The final stage of the protocol is the **Verification** stage. In it, the prover’s tree of witnesses is assessed to verify the prover’s position. The witnesses may attest to seeing the agent that named them, or not. If a witness does so, this is considered an approval. If sufficient approvals in the tree are amassed, the prover’s position claim is considered valid, provided other necessary criteria are also met in the tree.

Remark: The physical nature of the location verification is only relevant in the application of smart mobility proposed for the data market application in Chapter 3. T-PoP, however, works effectively with any statement that can be logically verified. In fact, logical verification is the only necessary requirement for our algorithm to function.

4.6 TPoP Protocol

In this section we provide an overview of T-PoP from an algorithmic perspective and we introduce some notation that is used in the remainder of the paper.

⁵Many options are available: KZG commitment schemes [272], Pedersen commitments, Merkle trees or Hashing [147]. Some schemes also offer the desirable property of being additively and/or multiplicatively homomorphic [200]. The choice of commitment scheme will be driven by application-specific needs.

Accordingly, there are three main algorithms that T-PoP calls:

- *Tree Building*: the process of building the tree of witnesses.
- *Checks*: the process of verifying that the tree that has been built satisfies certain criteria, listed below.
- *Verification*: the process of assessing the tree and deciding whether or not to confirm the prover's position.

4.6.1 Tree Building

A *prover's* aim is to amass sufficient *approvals* to support their position claim. Indeed, the prover provides a tree data structure that summarises their received approvals and the dependencies between them. The number of nodes and edges present in the tree are evaluated to determine the likelihood of the prover's claim being true. We proceed by outlining the algorithm the prover follows when constructing the tree structure of approvals.

Terminology: Each node in the tree represents an agent. The root of the tree is the prover. In a pair of nodes connected by a directed edge, the node where the edge originates is the *parent*, and the node where the edge terminates is the *child*. Each edge represents an *approval*. Namely, an edge from parent a_i to child a_j encodes a_j 's approval of a_i . The tree must be of a specific dimension. Precisely, the variables are:

- **height**: the size of the longest path from the root to the leaves of the tree, h .
- **branching factor**: the number of children, w_d , that a parent must have at a given depth level, d . For brevity, if the branching factor is constant across levels, we omit the index and simply denote it as w .

Algorithm 2 Tree Building

Require: Prover a_i , height h , branching factors $w_1, \dots, w_d$

- 1: Initialise a_i as the root of the tree G and as a parent of depth level 1.
 - 2: **for** $d = 1, \dots, h - 1$ **do**
 - 3: **for all** parents p at depth level d **do**
 - 4: p names w_{d+1} children from its set of neighbours.
 - 5: All named children are added as nodes of G at depth level $d + 1$, with p as their parent node.
 - 6: **end for**
 - 7: **end for**
 - 8: **return** G
-

The number of witnesses per depth level, n_d , can therefore be computed recursively:

$$n_d = w_d \cdot n_{d-1} \forall d \in \{1, \dots, h - 1\} \quad (4.2a)$$

$$n_0 = 1 \quad (4.2b)$$

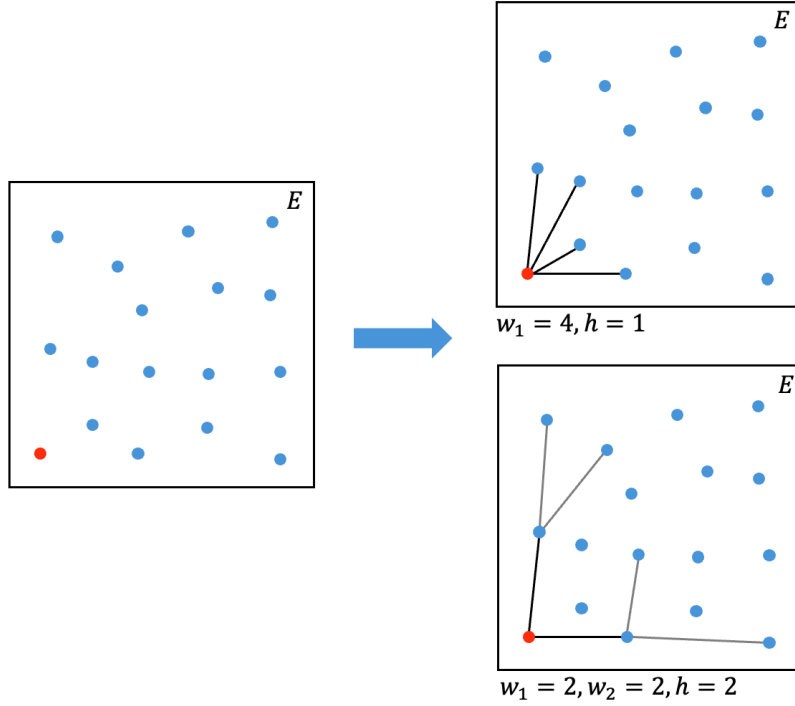


Figure 4.3: Tree building examples. Agent a_i constructs a commitment to their alleged position $\hat{p}_i$. The panel on the top right shows the construction of a tree for $d = 1$ and $w_1 = 4$, while the panel on the bottom right shows the construction of a tree for $d = 2$, $w_1 = 2$, $w_2 = 2$.

4.6.2 Checks Algorithm

This algorithm is called within the **Verification** algorithm, and its purpose is to assess, for each parent's sub-tree, whether it satisfies certain criteria. If so, the parent node is considered honest and therefore remains in the prover's tree. Otherwise, it and its branches are pruned from the prover's tree. The criteria being checked for in each sub-tree are defined in algorithm 3, and are the following:

1. each child *approves* their parent (Line 1),
2. the parent has named sufficient children (Line 2),
3. no child has been already named in the tree by another parent (Line 3),
4. no parent names the same child twice (Line 4).

Indeed, criterion 3 and 4 can be summarised by ensuring each agent in the tree is unique. The threshold parameter, t , is a percentage that determines the minimum number of children that a parent must have. A parent remains in the tree *iff* $|parent's\ children| \geq w_d \cdot t$.

Algorithm 3 Checks

Require: Child a_j , Parent a_i , Named Agents A , Branching Factor w_d , threshold t

```
1: if  $a_j$  approves  $a_i$ 
2:   and  $\#[a_i\text{'s children}] \geq w_d \cdot t$ 
3:   and  $a_i$  not in  $A$ 
4:   and  $\#set([a_i\text{'s children}]) = \#[a_i\text{'s children}]$  then
5:     Add  $a_j$  to  $A$ 
6:     return True
7:   else
8:     return False
9:   end if
```

4.6.3 Verification

Starting at the lowest depth level, for each parent in that level, **Verification** calls the **Checks** algorithm for all of its children. If successful, the parent and its branch remains in the tree, otherwise, it is pruned. It then moves up a level, and performs the same operations, only considering the non-pruned nodes. The algorithm terminates at the root of the tree, where it evaluates if the tree has sufficient nodes after pruning. If so, the *prover* is considered honest, otherwise it is considered dishonest. Using the prover's tree and the **Checks** function, **Verification** calls **Checks** in a reverse breadth-first-search manner.

It can therefore be observed that T-PoP has the following operating conditions that can be varied: the threshold t , and the tree dimensions (which are the height h and branching factor vector $(w_1, \dots, w_{h-1})$). We denote these operating conditions as θ , where $\theta = \{t, h, (w_1, \dots, w_{h-1})\}$.

Example

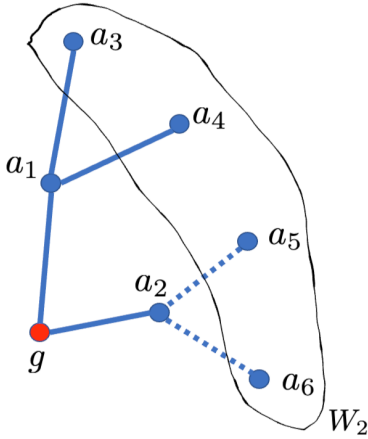
For clarity, let us illustrate a practical example. Consider the T-PoP example in Figure 4.4, in which $\theta = \{t = 0.5, h = 2, w_1 = 2, w_2 = 2\}$, and so $n_1 = 2$ and $n_2 = 4$. Solid arrows mean that a witness approves their parent and dotted lines mean that a witness does not approve their parent. Agents a_5 and a_6 are dishonest agents, so that their committed positions, $\hat{p}_5$ and $\hat{p}_6$, are different from their true positions. However, agent a_2 does not know this, it saw the agents next to it and it picked a_5 and a_6 as witnesses. So, a_5 and a_6 do not confirm that a_2 is a neighbour of theirs, whereas a_3 and a_4 confirm that a_1 is a neighbour of theirs.

As per line 15 of the **Verification** algorithm 4, agent a_1 has enough confirmed witnesses ($\text{Parent Approvals}(a_1) = 2 \geq t \times w_2 = 0.5 \times 2$) and stays in the tree, while agent a_2 does not have enough confirmed witnesses ($\text{Parent Approvals}(a_2) = 0 < 0.5 \times 2$), and so a_2 is removed from the tree. However, since the total number of confirmed witnesses at level 2 is $D_2 = 2 \geq t \times n_2 = 0.5 \times 4$, T-PoP does not terminate for g (line 21 of **Verification** algorithm). It continues by moving up to depth level 1 and evaluating the **Checks** algorithm 3 again. At this point, a_2 has been removed but a_1 confirms that g is its neighbour. Therefore, the final output of T-PoP is that g 's position claim is *True*,

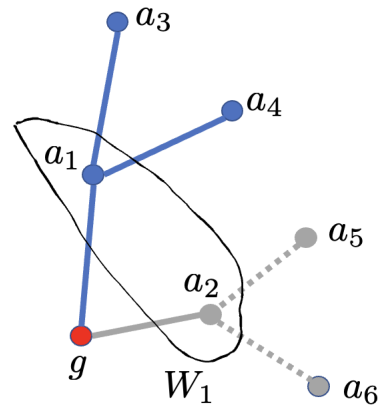
Algorithm 4 Verification

Require: Tree G , threshold t , Height h , Branching Factor $w_1, \dots, w_d$, threshold t

```
1: Initialise Named Agents to  $A = []$ 
2: Initialise  $G$ 's nodes to not pruned
3: Initialise Depth Level Approvals to  $D_1, \dots, D_{d-1}$ 
4: for  $d = h - 1, \dots, 1$  do
5:   Depth Level Approval Counter:  $D_d = 0$ 
6:   for each parent in at depth level  $d$  do
7:     Parent Approvals = 0
8:     for each parent's child do
9:       if child is not pruned then
10:        if Checks(child, parent,  $A$ ,  $w_d$ ,  $t$ ) is True then
11:          Parent Approvals + = 1
12:        end if
13:      end if
14:    end for
15:    if Parent Approvals <  $w_d \cdot t$  then
16:      parent is pruned
17:    else
18:       $D_d$  + = 1
19:    end if
20:  end for
21:  if  $D_d < w_d \cdot t$  then
22:    return False
23:  else
24:    return True
25:  end if
26: end for
```



(a) T-PoP begins by evaluating the leaves of the tree, which are the witnesses in W_2 . In this case, agents a_5 and a_6 do not confirm that they see agent a_2 , even though a_2 is an honest agent. This leads to agent a_2 being eliminated from the tree.



(b) The algorithm proceeds by moving up one level, and now evaluates the witnesses in W_1 . Agent a_2 has been eliminated by the tree (shown in grey) and so only agent a_1 is left.

Figure 4.4: Example of T-PoP algorithm with $d = 2$, $w_1 = 2$, $w_2 = 2$.

because $D_1 = 1 \geq t \times n_1 = 0.5 \times 2$. As can be seen in the example above, t is critical in determining the output of T-PoP. For instance, if $t = 1$, then $D_2 = 2 < t \times n_2 = 1 \times 4 = 4$, causing T-PoP to terminate at the depth level 2 and returning an output of *False* for g .

4.7 Preliminary Theoretical Analysis

In this section and what remains of this chapter, the first iteration of the mathematical modelling of T-PoP is presented. This includes definitions of the evaluation metrics for T-PoP's performance, a mathematical model describing its functioning, and an agent-based simulator used to study T-PoP's behaviour under varying conditions, such as adversarial presence and population density. This work was extensively developed in later work, and thus will be presented separately in the subsequent chapter.

The motivation of developing a mathematical model for T-PoP is to give its users probabilistic guarantees of its performance. In this manner, the user may input their expected operating conditions of their environment into the model, and receive a probabilistic guarantee of the algorithm's security and reliability. To begin constructing the mathematical model, we first assume that each agent has a probability of being honest and of being coerced. Because T-PoP is a probabilistic algorithm, the purpose of this mathematical model is to model the probability that any given tree with agents of different states will be considered truthful or not by T-PoP. We assume that the honesty and coercion states of each agent are independently and identically distributed (iid) Bernoulli trials, $\mathcal{B}(\cdot)$. Formally, for each agent, we define two independent random variables, $h \sim \mathcal{B}(p_h)$ and $c \sim \mathcal{B}(p_c)$, where $p_h \in [0, 1]$ and $p_c \in [0, 1]$ are the probabilities of any agent being honest and coerced, respectively (and it follows that $1 - p_h$ and $1 - p_c$ are the probabilities of an agent being respectively dishonest and non-coerced). Depending on the outcome of these trials for a witness at depth level d , it will then deterministically confirm that the witness at level $d - 1$, which named them, is its neighbour or not (note that agents might be lying about whether another agent is their *true* neighbour or not). The outcome of this interaction is summarized in the truth table 4.1. This behaviour is entirely dependent on the honesty and coercion state of each agent, provided both agents are within each other's range of sight. If agent, a_i , verifies agent a_j 's position, the outcome is 1, and 0 otherwise. Figure 4.5 depicts the graphical probability model constructed for a tree where $h = 2, w_1 = 2, w_2 = 2$. This Figure shows for each agent, denoted P if it is the prover, or w_i if its a witness, what variables influence the probability of a prover receiving enough approvals. These variables are the states that each agent can take, which are its honesty and coercion. The Figure illustrates the relationship between each of these variables across numerous agents in a tree.

In this model, we assume that the density of agents in E is very high. This means that provers, which are constructing their tree following algorithm 2, are always able to find enough witnesses at each level and that each witness is always unique. This assumption may not be satisfied in practice, since agents may be isolated and therefore not have enough witnesses around them. In the subsequent chapter we consider the scenarios where these

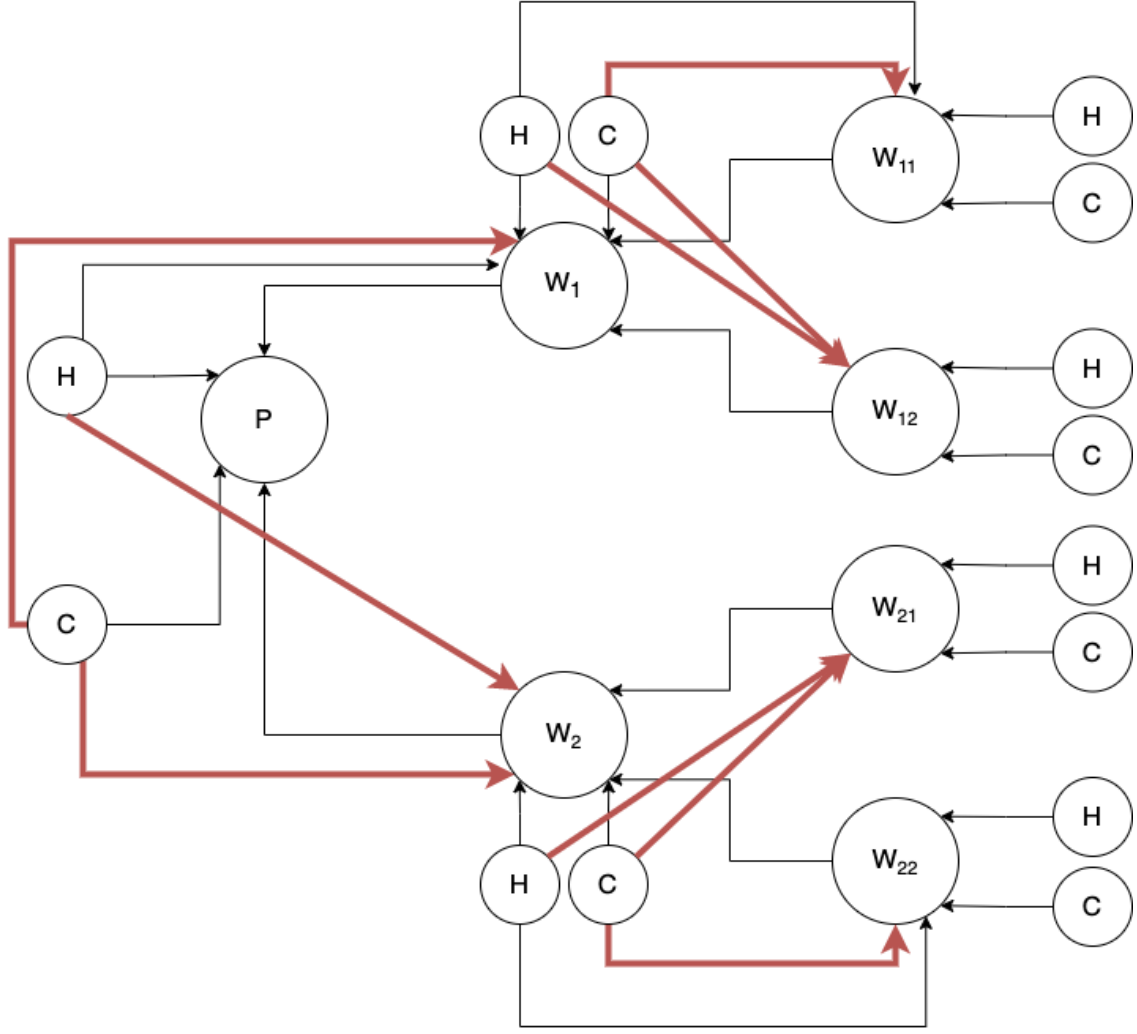


Figure 4.5: Graphical Probability Model of T-PoP with parameters $h = 2, w_1 = 2, w_2 = 2$. The red lines indicate that those variables influence the output of a specific node.

conditions may not be satisfied and include them into our extended model. Although it may arise that an honest agent is simply unable to find enough unique witnesses, we argue that if an agent does not have sufficient witnesses, it is reasonable to consider them less trustworthy, despite them being honest. It is of importance to remember that the honesty and coercion states of each agent are unknown and unobservable in practise, both to the algorithm and to all other agents in the environment. Indeed, that is the purpose of T-PoP: to determine who is honest and who is not. Once the tree has been created, the **Verification** step can be used to provide the outcome of the algorithm, which can be either 0 (if the algorithm deems the prover dishonest) or 1 (if the algorithm deems the prover honest). Given a prover, g (the root of the tree), we define a random variable, $C(g) \in \{0, 1\}$, whose outcome depends on the ensemble of independent and identically distributed random variables, h, c , in its constructed tree, and on T-PoP parameters, $\theta \equiv \{t, h, (w_1, \dots, w_d)\}$. In order to analyse T-PoP's performance, we consider two metrics: reliability and security. We define them as follows:

Definition 4.7.1. True Negatives or Security, TN , is a conditional probability quan-

$a_i \backslash a_j$	h and $\bar{c}$	h and c	$\bar{h}$ and c	$\bar{h}$ and $\bar{c}$
h and c	1	1	1	1
h and $\bar{c}$	1	1	0	0
$\bar{h}$ and c	1	0	1	0
$\bar{h}$ and $\bar{c}$	1	0	0	1

Table 4.1: A truth table showing confirmation (1) or rejection (0) of a parent’s (a_i) position by a witness (a_j), depending on the honesty (h) and coercion (c) states of each agent. Notice that the relationship between a_i and a_j is symmetrical.

ifying the ability of the algorithm to detect dishonest agents, $\bar{H}$. Specifically, it is the true-negative conditional probability, which, under stationarity assumptions, is independent of $i \in \{1, \dots, |A|\}$:

$$TN \equiv \Pr[TPoP(a) = False | a \in \bar{H}]$$

Definition 4.7.2. True Positives or Reliability, TP , is a conditional probability quantifying the ability for the algorithm to detect honest agents, H . Specifically, it is the true-positive conditional probability. Once again, under stationarity assumptions:

$$TP \equiv \Pr[TPoP(a) = True | a \in H]$$

In Figure 4.6, we display empirically evaluated TN and TP for two sets of parameters, respectively: $\theta_1 = \{t = 1, h = 1, w_1 = 6\}$ and $\theta_2 = \{t = 1, h = 2, w_1 = 2, w_2 = 2\}$, in each case varying p_h and p_c in their ranges $[0, 1]$, with steps of 0.02. To emphasize the functional dependence of these probabilistic performance metrics on the honesty and coercion probabilities of the iid agents, we denote these metrics by $TP(p_h, p_c)$ and $TN(p_h, p_c)$, respectively. These are evaluated empirically via extensive Monte Carlo simulations of the graphical model. Specifically, we simulated 5000 trees for each of the two parameter settings above.

4.8 Agent-Based Simulations

In this section, we present an agent-based simulator, coded in Python, to replicate a more realistic scenario for T-PoP and to validate the graphical theoretical model that we presented in the previous Section. Each agent has a number of varying attributes such as their range of sight, position, velocity, unique identifier and whether they are honest or dishonest, and coerced or not. Depending on the latter variables, each agent will commit to their true position or a fake one, and will add agents to their set of neighbours as outlined in Table 4.1. We then create an environment with a fixed density of agents in it, and place

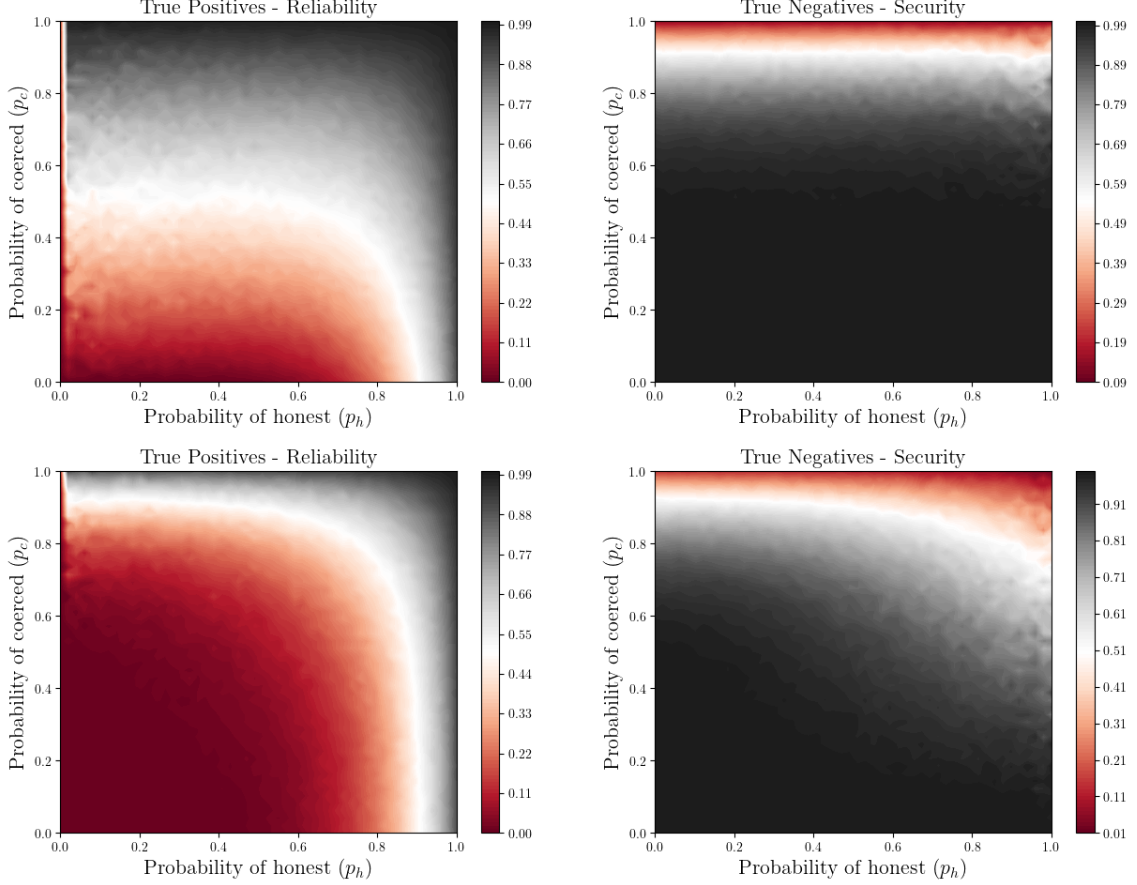


Figure 4.6: T-PoP performance for the **graphical probability model** (depicted in Figure 4.5). The panels in the left column show the reliability rate (True Positives), while the panels in the right column show the security rate (True Negatives). The first row is associated with model parameters, $\theta_1 = \{t = 1, h = 1, w_1 = 6\}$, while the second row is associated with model parameters, $\theta_2 = \{t = 1, h = 2, w_1 = 2, w_2 = 2\}$.

these randomly and uniformly across the environment. We allow them to move according to their velocity vector, within the bounds of the environment. Each time the agents move, all agents construct a new set of neighbours and discard the previous one. Next, each agent wishing to claim their position runs T-PoP; namely, they run the **Tree Construction** and the **Verification** algorithms. Our simulator can be found in this [GitHub Repository](#).

Other key variables are the threshold, depth and number of witnesses used. A greater threshold increases security, but also reduces reliability. Increasing the number of witnesses increased both security and reliability. However, this may not be a suitable measure for sparser scenarios, or cases where agents are moving at high speed, potentially incurring a communication overhead. We advocate for the users to select the appropriate threshold, depth and number of witnesses based on the individual needs of their own application. Lowering the threshold can lower security, but provides more flexibility in the system. The user can then select an appropriate number of witnesses based on the expected density of their network, and use the depth parameter to find an appropriate trade-off between security, reliability, communication overhead and flexibility.

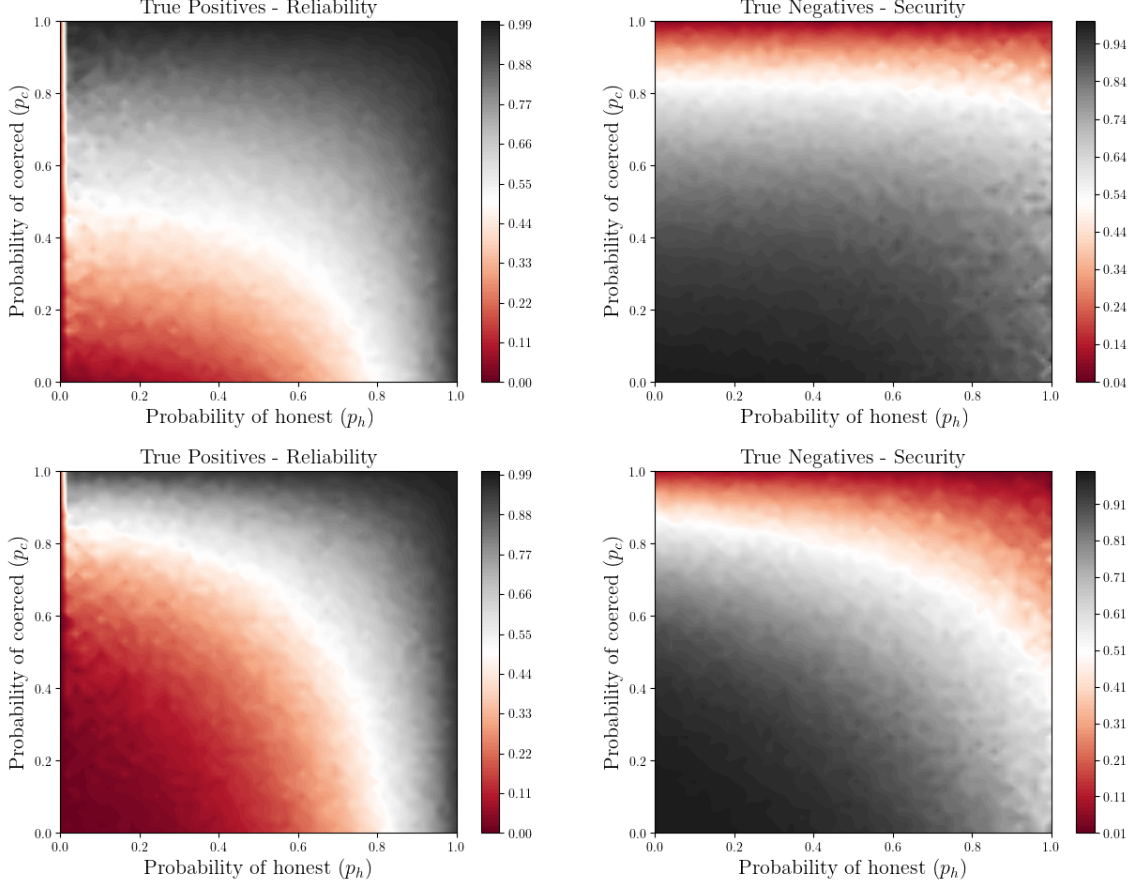


Figure 4.7: T-PoP performance for the **agent-based simulator** model. The panels in the left column show the reliability rate (True Positives), while the panels in the right column show the security rate (True Negatives). The first row is associated with model parameters $\theta_1 = \{t = 1, h = 1, w_1 = 6\}$, while the second row is associated with model parameters $\theta_2 = \{t = 1, h = 2, w_1 = 2, w_2 = 2\}$. Notice the close similarity to Figure 4.6.

These simulations reveal that generally, T-PoP is very secure although less reliable. This behaviour can be tuned according to the operating parameters, to allow the algorithm to still consider agents with less approvals honest, thus allowing for more resilience in case of errors such as randomly selecting the same witness twice in the tree.

4.9 Preliminary Results

In this section, the aim is to outline the preliminary results on the performance of T-PoP for a given choice of operating conditions, as well as to validate the results from the graphical probability model (shown in Figure 4.5), with the objective of creating an analytical framework for analysis of the T-PoP class of algorithms.

The simulations have been set up as follows: we considered each possible combination of p_h and p_c in the ranges $[0, 1]$, with steps of 0.02. For each combination we ran 50 Monte Carlo simulations and we computed empirical estimates of the values of reliability $TP(p_h, p_c)$ and security $TN(p_h, p_c)$. Simulations are set up in such a way that on average each agent has 50 neighbours within their field of view. While this number might appear very high, the

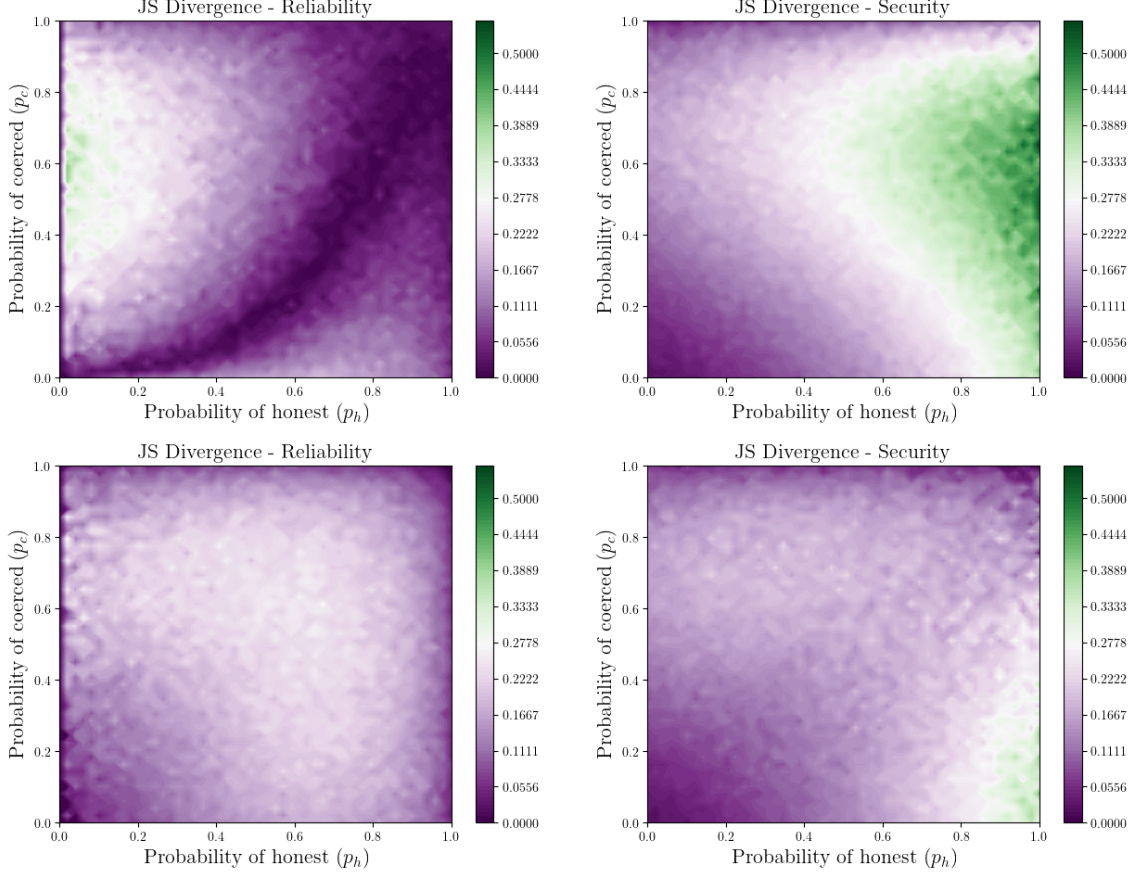


Figure 4.8: Jensen-Shannon divergence (JSD) between the agent-based simulator reliability TP_s , and the probability graphical model reliability TP_m (left column), and between the agent-based simulator security TN_s , and the probability graphical model security TN_m (right column). Results are shown for $\theta_1 = \{t = 1, h = 1, w_1 = 6\}$ (top row) and $\theta_2 = \{t = 1, h = 2, w_1 = 2, w_2 = 2\}$ (bottom row).

purpose was to ensure the results were not divergent from the graphical probability model due to a lack of sufficient neighbours. Moreover, real-life situations with high density of pedestrians (e.g., the underground during peak hours) would map well into this scenario. We ran these simulations for each set of parameters, $\theta_1 = \{t = 1, h = 1, w_1 = 6\}$ and $\theta_2 = \{t = 1, h = 2, w_1 = 2, w_2 = 2\}$. The results are shown in Figure 4.7.

T-PoP with θ_1 yields better performance overall (as both reliability $TP(p_h, p_c)$ and security $TN(p_h, p_c)$ rates are higher for each choice of p_h and p_c). The simulations with θ_2 show that decreasing the number of witnesses by a third and increasing the height by 1 allows us to achieve similar results. This is useful because—while the total number of nodes in each prover’s tree is the same for both scenarios—a tree of depth 2 with 2 witnesses per parent places a smaller communication overhead on the prover, because it only needs to name and communicate with 2 witnesses, as opposed to 6.

Overall, in high density scenarios, the results of both simulations show that if $p_h > 0.9$ and $p_c < 0.2$ T-PoP is capable of achieving security $> 0.85\%$ and reliability $> 0.9\%$ for θ_1 . For the same p_h and p_c the algorithm achieves rates of security $> 0.7\%$ and reliability $> 0.9\%$

for θ_2 . For lower proportions of honest agents and higher proportions of coerced agents (i.e. in the presence of many colluding, dishonest and coerced agents), the performance of T-PoP degrades. This is to be expected in a decentralised system such as T-PoP, since it is in theory impossible to distinguish between a group of honest agents verifying each other and a group of dishonest and coerced agents collaborating to verify each other in a fraudulent manner. Accordingly, we can observe across all figures that—even when the percentage of honest agents is low—the security remains high at the expense of reliability. We observe that whilst, indeed, T-PoP can detect true negatives (i.e. be secure) in highly (and perhaps even unrealistically highly) adversarial environments, the drawback is that it penalises honest agents too harshly (i.e. is unreliable). This is a consequence of the collaborative nature of the algorithm. When the number of honest agents in the system is low (i.e. $p_h \downarrow 0$), they will with high probability (w.h.p.) be misclassified as dishonest because they will select dishonest witnesses w.h.p. In general, for both scenarios it can be observed that T-PoP is a highly secure algorithm, although less reliable. This makes it suitable for scenarios where a False Positive is dangerous or highly undesirable. It is worth noting that these results hold for a maximum level threshold of 1, and in scenarios where a higher reliability is important, the threshold can be relaxed or the tree dimensions can be decreased.

The density of agents in the environment vastly affected the performance of T-PoP. This was especially noticeable when the average number of agents per range of sight in the environment was lower than or equal to the total number of nodes of the tree being constructed, which greatly increased the number of False Negatives, thus making T-PoP unsuitable for low density environments (for these specific tree dimensions). This is because when the density is lower, the probability of selecting the same agent twice, or of not having sufficient neighbours to construct the tree, greatly increases.

Validation of the graphical model

For validation of the graphical probability model, we make use of the Jensen-Shannon Divergence (JSD) [351] to quantify how close the probability distributions obtained through the agent-based model (i.e. the T-PoP implementation) and the graphical model are. This metric is used to measure how similar are probability distributions to one another. In what follows, we refer to the values of reliability, TP , and security, TN , obtained from the simulated agent-based model as TP_s and TN_s , and the ones obtained from the graphical model as TP_m and TN_m . We compute two JSD-based metrics: (i) the (p_h, p_c) -indexed (i.e. pointwise) JSD map between TP_m and TP_s , and between TN_m and TN_s , respectively; and (ii) the global JSD between the normalized TP_m and TP_s maps, and the normalized TN_m and TN_s maps, respectively. By “normalized”, we mean that each of these positive maps is divided by its element sum, yielding a probability mass function. In case (ii) we can therefore condense into a single number the relative performances displayed in the figures (TP and TN , respectively) for the simulated T-PoP system (s) and its graphical model (m).

The results for the point-wise evaluation ((i) above) of the JSD are shown in Figure 4.8, while the global evaluation ((ii) above) is summarised in table 4.2. Note that $0 \leq \text{JSD} \leq 1$, with lower values achieved when probabilities are close in value. From our results, it can be observed that there is agreement between the behaviour of the simulated system and the graphical model. It is clear that—at least for high density scenarios—the behaviour of the graphical model closely mirrors that of the implemented T-PoP system. Nevertheless, the pointwise JSD results reveal increased discrepancies in the security (S) metric when $p_h \uparrow 1$ (i.e. for high proportions of honest users).

Parameters	$JSD(TP_m, TP_s)$	$JSD(TN_m, TN_s)$
θ_1	0.139	0.095
θ_2	0.174	0.059

Table 4.2: Jensen-Shannon divergence (JSD) between the normalized reliability (TP) and security (TN) maps for two sets of model parameters.

4.10 Comparison of Graphical Probability Model and Simulated Environment

In the graphical model, we consider only the agent’s coercion and honesty attributes when it comes to witness approval, whereas in the simulated environment the approval is dependent also on the agents’ range of sights, positions and velocity. We hypothesize that the attributes with most importance that influence the outcome of T-PoP are the agent’s honesty and coercion state, thus these are the ones we considered in the graphical probability model. In the graphical model, agents do not move and they do not build the tree for T-PoP according to other agents that are within their range of sight. Rather, agents build their trees assuming that other agents are always present, and approvals are made following Table 4.1 only. Upon building a more realistic simulation scenario, where we introduce variables that add noise and uncertainty to the behaviour defined in Table 4.1, we observe that, at least for a high density scenario, T-PoP behaves in a quantitatively similar way to the one characterised by our graphical model. This validates the hypothesis that, indeed, the most important attributes for the performance of T-PoP are those of honesty and coercion, in high density scenarios. However, there are other factors that can influence the outcome of T-PoP which are not yet considered in our model. For example, depending on the density, the probability of having enough neighbours of sampling unique neighbours may vary. These conditions should be further explored and developed into our model to produce a more realistic, generalisable model. Indeed this is precisely what the following chapter explores.

4.11 Conclusion

We presented a proof of position class of algorithms that are fully decentralised. They can be run by any agent participating in the network and they do not assume trust in a central authority, nor do they rely on physical infrastructure. We also considered a range of attack vectors by allowing agents not only to lie about their own position, but also about others' positions. Our algorithm can also be computed in a privacy-preserving manner, as there is no need for the true location of an agent to be revealed to the network. It suffices to disclose that neighbouring agents are within each other's proximity. We developed a graphical probability model for this class of proof of position algorithms, and statistically validated the model via comparative analysis of its agent-based simulated behaviour. In future work, we use the theoretical model to predict the performance of T-PoP as a function of its operating conditions, θ . Specifically, we will be interested in characterising the effect of the height of the tree (h), threshold (t) and number of witnesses (w_d) on the security and reliability of the T-PoP class of algorithms. Developing such a framework can allow users to select the optimal operating conditions of the algorithm to meet their needs, based on their expected density, fault tolerance and proportion of honest and non-coerced agents in their system. Finally, what remains unaddressed but is later answered in the following chapter is understanding T-PoP's resilience to a more complex adversarial scenario. In its current form, it is not possible for T-PoP to distinguish between a tree of honest agents approving each other, and a malicious tree of dishonest and coerced agents approving each other, both with enough approvals and sufficient, unique neighbours named. We call this attack a malicious platoon, and subsequent development of the current mathematical framework aims to probabilistically detect the presence of these platoons.

Chapter 5

Proof of Position Algorithms: Extended Model

Abstract

This work presents an extended mathematical model for the previously outlined class of algorithms: Tree Proof of Position. An optimised and extended agent-based simulator is constructed, and we perform a larger number of simulations for a wider range of operating conditions of the algorithm. This open-source code-base also includes a privacy-preserving implementation of the algorithm using homomorphic encryption. This enables participants to obtain a proof that their location claim is valid, without having to publicly disclose their location. Finally, a model to probabilistically detect a specific class of adversarial attacks is developed. This attack is a collaborative cluster of dishonest agents that collude to verify each other's location continuously in time. This behaviour is in principle indistinguishable from honest agents verifying each other, but we propose a probabilistic method to detect it.

5.1 Contributions Disclosure

This chapter described the research outputs that led to the work published in [344]. For this publication, I did not develop the model for Conditions 2 and 3. I was responsible for all other aspects of the work.

5.2 Introduction

This chapter outlines a further development of the work presented in the previous chapter. This includes an extended, more comprehensive mathematical model of the T-PoP algorithm, improvements to the efficiency of the agent-based simulator, along with an increased number of simulations for a wider range of operating conditions, and a mathematical model to detect malicious agents collaborating together to defraud the algorithm.

In line with the application proposed in Chapter 3, we consider a context of a smart city, where vehicles use sensors to measure data of interest. Before aggregating and selling this crowd-sourced data, they must ascertain that they are indeed in their reported location. This is the purpose of T-PoP. In subsequent sections we discuss the constructed agent-based simulator. For this application, we consider each agent to represent a vehicle, although there is no physical requirement that dictates that an agent should be a vehicle. Agents could be represented by phones, IoT devices or any other proxy for a physical person with a minimum capacity of computation to perform the algorithm.

5.3 Mathematical Modelling of T-PoP

In the previous chapter, a preliminary mathematical model of T-PoP is introduced, which is a graphical probability model, shown in Figure 4.5. Although the model is shown to be faithful to T-PoP's behaviour in a simulated environment of agents, this model is constructed for two specific tree dimensions. It is of great interest to extend this model to any generalised tree architecture and environment density, which is precisely what is achieved in this chapter. This mathematical model can be used to select the most suitable operating conditions (i.e.: the tree dimensions) of T-PoP to achieve desired performance levels. These dimensions are the height, h , branching factor, w , and threshold, t . Recall these are denoted as $\theta = (h, w, t)$.

As discussed in Section 4.6 of Chapter 4, the heart of T-PoP is the **Checks** algorithm, which is recursively called in the **Verification** algorithm. The **Checks** algorithm assesses whether the criteria necessary to confirm an agent's position claim are satisfied. These criteria are the following:

1. each child *approves* their parent,
2. each parent has named sufficient children,
3. all agents are unique in the tree.

We hypothesize that it suffices to compute the probability that each of the criterion above are met to determine the probability that an agent will receive a valid proof of position. We proceed by computing the individual probabilities of each criterion and combine them into one model. Consequently, the probability that an agent a_i with state s_u will receive a proof of position through T-PoP with operating conditions θ , is equal to the product of the probability of each criterion above being satisfied:

$$P(\text{TPoP}(a) = \text{True} | s_u, \theta) = \prod_{j=1}^3 P(\text{Criteria } j = \text{True} | s_u, \theta)$$

Finally, in order to analyse T-PoP's performance, we consider the two metrics of security and reliability, previously defined in 4.7.1 and 4.7.2 respectively.

In what follows, a mathematical expression for $P(\text{Criteria } j = \text{True} | s_u, \theta), j \in \{1, 2, 3\}$

is derived, and subsequently it is demonstrated that the model faithfully represents T-PoP's behaviour. This is done by conducting extensive agent-based simulations, and we find that the results obtained are in complete agreement with those of the mathematical model provided. The rationale behind the analysis is to understand the performance of the algorithm when multiple agents collude together to try to fake their own position (dishonest agents) or attest that they see another agent in their fake position (coerced agents). First let us begin by formalising the agent states mathematically. This allows us to construct a probability model to determine the likelihood that a prover will obtain sufficient approvals in their tree.

5.3.1 Agent States

All agents have three attributes: honesty, α , coercion, β and a claimed position, γ . These attributes are binary, taking either a **True** or **False** value. We say an agent a_i has an honesty attribute α_i , coercion attribute β_i and claimed position γ_i , where $a_i \in A$, recalling that A is the set of all agents.

- *Honesty*: An agent a_i is either *honest*, $\alpha_i = h$ or *dishonest*, $\alpha_i = \bar{h}$. An *honest* agent claims to be in their *real* position, whereas a *dishonest* agent claims to be in a position that is different to their *real* position. We call the set of *honest* agents $H = \{a_i \in A : \hat{p}_i = p_i\}$ and the set of *dishonest* agents $\bar{H} = \{a_i \in A : \hat{p}_i \neq p_i\}$.
- *Coercion*: An agent a_i is either *coerced*, $\gamma_i = c$ or *non-coerced*, $\gamma_i = \bar{c}$. When approving other agents, a *coerced* agent will approve other agents in their *claimed* position (regardless of whether this *claimed* position is equal to their *real* position or not). A *non-coerced* agent will only approve agents in their *real* position. We call the set of *coerced* agents $C = \{a_i \in A : \beta_i = c\}$ and the set of *non-agents* $\bar{C} = \{a_i \in A : \beta_i = \bar{c}\}$.
- *Position*: An agent a_i claims to be in a position. If the agent is *honest*, their *claimed* position is equal to their *real* position p . As such, we say, $\gamma_i = t$. If the agent is *dishonest*, their *claimed* position $\hat{p}$ is different to their *real* position, p , and therefore, $\gamma_i = \bar{t}$. We call the set of agents that claim to be in their real position $T = \{a_i \in A : \gamma_i = t\}$ and the set of agents that claim to be elsewhere $\bar{T} = \{a_i \in A : \gamma_i = \bar{t}\}$.

All three attributes of an agent a_i are denoted as follows:

$$\alpha_i \in \{h, \bar{h}\} \tag{5.1a}$$

$$\beta_i \in \{c, \bar{c}\} \tag{5.1b}$$

$$\gamma_i \in \{t, \bar{t}\} \tag{5.1c}$$

We depict the agents having each attribute (honesty, coercion and position) as a set in Figure 5.1. Each numbered region in Figure 5.1, u , where $u \in \{1, 2, 3, 4, 5, 6\}$ corresponds

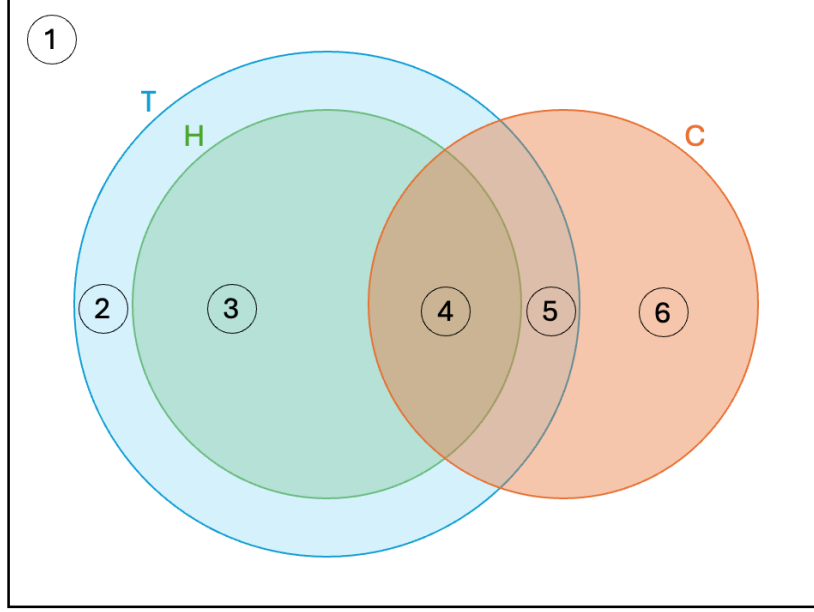


Figure 5.1: Venn diagram representing possible agent state combinations.

to a possible set of attribute combinations, S_u , that an agent may possess.¹ For example, if $a_i \in S_2$ then a_i belongs to the set of all agents that are dishonest, $\overline{H}$, the set of all agents that are not coerced, $\overline{C}$, and the set of all agents that claim to be in their true position, T .

For ease of exposition, we define the *state* of an agent a_i , $x(a_i)$, to be the set of attributes an agent has:

$$x(a_i) = \{\alpha_i, \beta_i, \gamma_i\} \quad (5.2)$$

where $x(a_i)$ may be any of the following permutations:

$$s_1 = \{\overline{h}, \overline{c}, \overline{t}\} \quad (5.3a)$$

$$s_2 = \{\overline{h}, \overline{c}, t\} \quad (5.3b)$$

$$s_3 = \{h, \overline{c}, t\} \quad (5.3c)$$

$$s_4 = \{h, c, t\} \quad (5.3d)$$

$$s_5 = \{\overline{h}, c, \overline{t}\} \quad (5.3e)$$

$$s_6 = \{\overline{h}, c, t\} \quad (5.3f)$$

In short, we say that an agent a_i has a state $x(a_i)$, and that it belongs to the set of agents

¹There are three possible attributes, that can each take a true or false value, meaning there should be a total of 2^3 possible states. However, in effect there are only 6, because the set of honest agents is a strict subset of the agents claiming to be in a true position. By definition there are no honest agents that claim to be in false positions.

S_u , ($a_i \in S_u$), where $i \in |A|$, $u \in \{1, 2, 3, 4, 5, 6\}$ and

$$S_1 = \{\overline{H} \cap \overline{C} \cap \overline{T}\} \quad (5.4a)$$

$$S_2 = \{\overline{H} \cap \overline{C} \cap T\} \quad (5.4b)$$

$$S_3 = \{H \cap \overline{C} \cap T\} \quad (5.4c)$$

$$S_4 = \{H \cap C \cap T\} \quad (5.4d)$$

$$S_5 = \{\overline{H} \cap C \cap \overline{T}\} \quad (5.4e)$$

$$S_6 = \{\overline{H} \cap C \cap T\} \quad (5.4f)$$

$$P(a_i \in S_u) = P(x(a_i) = s_u) \quad (5.5a)$$

$$P(x(a_i) = s_u) = P(\alpha_i)P(\beta_i)P(\gamma_i|\alpha_i) \quad (5.5b)$$

Formally, we assume that variables α and β are binary random variables, independent of each other. These attributes represent the probability of an agent being honest and coerced respectively. We can therefore define the probability of an agent being honest (i.e.: $P(\alpha_i = h)$) as p_h , and being coerced (i.e.: $P(\beta_i = c)$) as p_c , where $p_h, p_c \in [0, 1]$. Similarly, $1 - p_h$ and $1 - p_c$ are the probabilities of an agent being dishonest and non-coerced respectively. Formally:

$$p_h = P(x(a_i) = s_u, \forall u \in \{3, 4\}) \quad (5.6a)$$

$$p_c = P(x(a_i) = s_u, \forall u \in \{4, 5, 6\}) \quad (5.6b)$$

where $p_h, p_c \in [0, 1]$.

$$p_h = P(\alpha_i = h) \quad (5.7a)$$

$$p_c = P(\beta_i = c) \quad (5.7b)$$

$$1 - p_h = P(\alpha_i = \overline{h}) \quad (5.7c)$$

$$1 - p_c = P(\beta_i = \overline{c}) \quad (5.7d)$$

Next, we wish to compute the following probabilities: $P(\alpha_i = h)$, $P(\beta_i = c)$ and $P(\gamma_i = t)$ where:

$$P(\alpha_i = h) = 1 - P(\alpha_i = \overline{h})$$

$$P(\beta_i = c) = 1 - P(\beta_i = \overline{c})$$

$$P(\gamma_i = t) = 1 - P(\gamma_i = \bar{t})$$

However, attributes γ and α are not independent. By definition:

$$P(\gamma_i = t | \alpha_i = h) = 1 \quad (5.8a)$$

$$P(\gamma_i = \bar{t} | \alpha_i = h) = 0 \quad (5.8b)$$

In other words: an honest agent will always claim to be in their real position. Therefore the probability that they claim to be in a fake position is always zero. If $P(\alpha_i = \bar{h})$, finding $P(\gamma_i = t)$ is slightly more convoluted. This is because when a dishonest agent is sampled in a tree, one cannot, in theory, distinguish if its claimed position is its real position or not. Indeed, that is the purpose of T-PoP.

Remark: When a dishonest agent is named in a tree, it could be that the dishonest agent is being observed in their real position, or that it has somehow tricked other nearby agents into pretending it is in their vicinity, when in fact it is elsewhere (by sending a fake signal or through any other possible attack vector). Because of this, we consider dishonest agents to effectively exist in two places: their real position, and their claimed position, which is fake.

To determine $P(\gamma_i = t)$, we must therefore consider the two possible cases in which a dishonest agent may appear in a tree: as a parent or as a child. This is depicted in Figure 5.2.

Let us begin with the first case, shown in Figure 5.2a. If the dishonest agent a_i , is the parent in the tree:

$$P(\gamma_i = t | \alpha_i = \bar{h}) = 0 \quad (5.9a)$$

$$P(\gamma_i = \bar{t} | \alpha_i = \bar{h}) = 1 \quad (5.9b)$$

This is because the dishonest agent would never construct a tree for their real position: doing so would expose their lie. In the second case, as depicted in Figure 5.2b, if the dishonest agent is a child in the tree, then $P(\gamma_i = t) = 1$ if their parent is *non-coerced*, and $P(\gamma_i = t) = 0$ if their parent is *coerced*. Formally, for a dishonest child a_i , with a parent a_j :

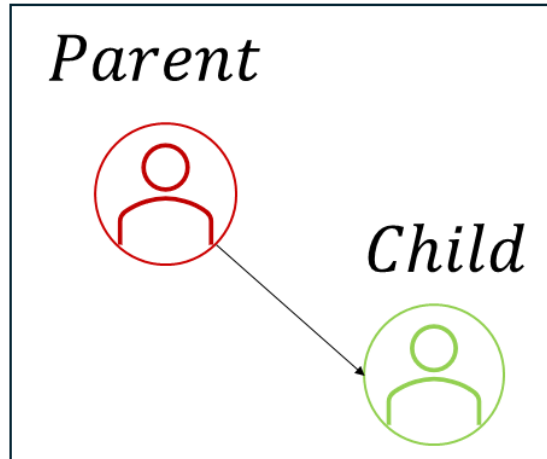
$$P(\gamma_i = t | \alpha_i = \bar{h}, \beta_j = c) = 0 \quad (5.10a)$$

$$P(\gamma_i = \bar{t} | \alpha_i = \bar{h}, \beta_j = c) = 1 \quad (5.10b)$$

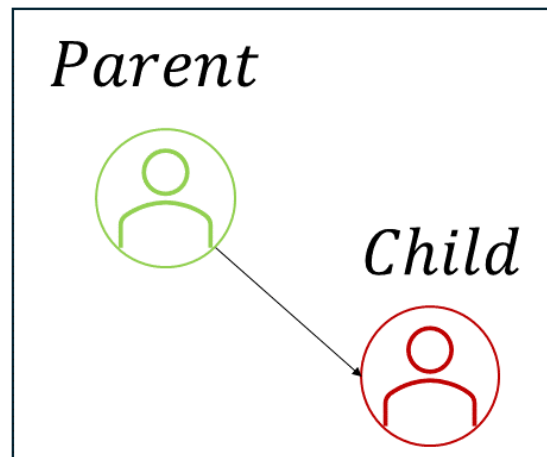
$$P(\gamma_i = t | \alpha_i = \bar{h}, \beta_j = \bar{c}) = 1 \quad (5.10c)$$

$$P(\gamma_i = \bar{t} | \alpha_i = \bar{h}, \beta_j = \bar{c}) = 0 \quad (5.10d)$$

In other words, equation 5.10a is equivalent to the following statement: the probability that a coerced parent picks a dishonest child in their fake position is 0. To summarise: the agent's attributes of honesty, α , and coercion, β , are independent of one another. However,



(a) Dishonest parent



(b) Dishonest child

Figure 5.2: Possible scenarios in which a dishonest agent may appear in a prover tree.

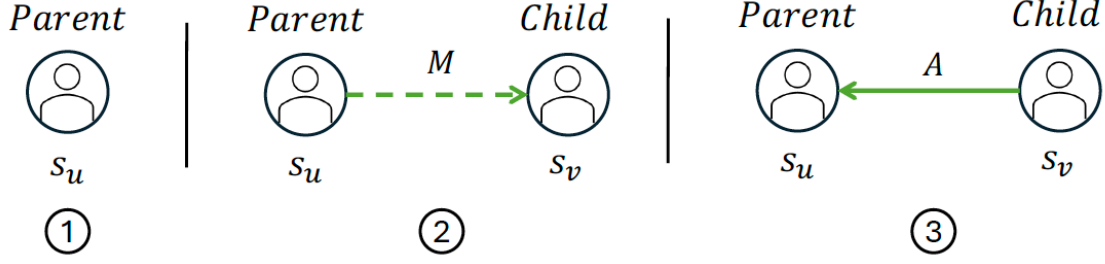


Figure 5.3: Process of parent to child approval

the agent's attribute of claimed position, γ , is by definition dependent on its honesty, α , and if the agent is a child, it is also dependent on the coercion attribute, β , of its parent.

Remark: Recall that a *coerced* agent is one that covers up for other agents that lie about their position. This means that if they happen to name a dishonest agent as their child, they will always 'see' them in their fake position. The reader may consider the state of coercion as an agent being bribed, attacked or threatened to co-operate with another agent that is lying about their own position.

5.3.2 Condition 1: Child to Parent Approval

Let us proceed by formalising the most crucial criterion: whether a child will approve the parent that named them. There are three stages in the process of an approval, as depicted in Figure 5.3.

Initial Parent State

First, consider the probability of the parent being in state s_u in equation 5.5. The vector in equation 5.11 summarises the probability of the parent being in any of the initial states.

$$I_1 = [P(s_1), P(s_2), P(s_3), P(s_4), P(s_5), P(s_6)]^T \quad (5.11)$$

where $P(s_u)$ is short for $P(\text{parent} \in S_u)$ with $\sum_{k=1}^6 P(s_k) = 1$.

Picking a Child, Given a Parent

In the next step, we determine the probability of picking a child in a given state, provided the parent is in another state. Let us denote the child as agent $\mathbf{c}$ and the parent as agent $\mathbf{p}$. We then summarise the probability that a parent $\mathbf{p}$ in a state s_u will select a child $\mathbf{c}$ in state s_v in Table 5.1.

Certain parent to child combinations are not possible by definition. This is shown by the entries of the Table 5.1 that are 0. For example, a *non-coerced* parent will **never** name a *dishonest* child in their *fake* position. They only 'see' agents in their real position. We

Parent \ Child	$\overline{H} \cap \overline{C} \cap \overline{T}$	$\overline{H} \cap \overline{C} \cap T$	$H \cap \overline{C} \cap T$	$H \cap C \cap T$	$\overline{H} \cap C \cap \overline{T}$	$\overline{H} \cap C \cap T$
$\overline{H} \cap \overline{C} \cap \overline{T}$	0	$m_{1,2}$	$m_{1,3}$	$m_{1,4}$	0	$m_{1,5}$
$\overline{H} \cap \overline{C} \cap T$	0	0	0	0	0	0
$H \cap \overline{C} \cap T$	0	$m_{3,2}$	$m_{3,3}$	$m_{3,4}$	0	$m_{3,6}$
$H \cap C \cap T$	$m_{4,1}$	0	$m_{4,3}$	$m_{4,4}$	$m_{4,5}$	0
$\overline{H} \cap C \cap \overline{T}$	$m_{5,1}$	0	$m_{5,3}$	$m_{5,4}$	$m_{5,5}$	0
$\overline{H} \cap C \cap T$	0	0	0	0	0	0

Table 5.1: Parent to child witness selections.

summarise the information in Table 5.1 as a matrix:

$$M = \begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} & m_{1,4} & m_{1,5} & m_{1,6} \\ m_{2,1} & m_{2,2} & m_{2,3} & m_{2,4} & m_{2,5} & m_{2,6} \\ m_{3,1} & m_{3,2} & m_{3,3} & m_{3,4} & m_{3,5} & m_{3,6} \\ m_{4,1} & m_{4,2} & m_{4,3} & m_{4,4} & m_{4,5} & m_{4,6} \\ m_{5,1} & m_{5,2} & m_{5,3} & m_{5,4} & m_{5,5} & m_{5,6} \\ m_{6,1} & m_{6,2} & m_{6,3} & m_{6,4} & m_{6,5} & m_{6,6} \end{bmatrix} \quad (5.12)$$

where the entry $m_{u,v}$ is defined as:

$$m_{u,v} = P[x(\mathbf{c}) = s_v | (x(\mathbf{p}) = s_u)] \quad (5.13)$$

and

$$x(\mathbf{c}) = s_v = \{\alpha_{\mathbf{c}}, \beta_{\mathbf{c}}, \gamma_{\mathbf{c}}\}$$

$$x(\mathbf{p}) = s_u = \{\alpha_{\mathbf{p}}, \beta_{\mathbf{p}}, \gamma_{\mathbf{p}}\}$$

To compute $m_{u,v}$ it suffices to evaluate:

$$m_{u,v} = P(\alpha_{\mathbf{c}})P(\beta_{\mathbf{c}})P(\gamma_{\mathbf{c}}) \quad (5.14)$$

$P(\alpha_{\mathbf{c}})$ and $P(\beta_{\mathbf{c}})$ are known through equations 5.7a and 5.7b respectively. To evaluate $P(\gamma_{\mathbf{c}})$ we begin with the prover agent, which has no parents by definition. In this case, $P(\gamma_{\mathbf{p}}) = P(\gamma_{\mathbf{p}} | \alpha_{\mathbf{p}})$, and we simply refer to equations 5.8 or 5.9 and pick the correct case. We then consider that parent's child, and to compute $P(\gamma_{\mathbf{c}})$, we evaluate

$$P(\gamma_{\mathbf{c}}) = P(\gamma_{\mathbf{c}} | \alpha_{\mathbf{c}})P(\beta_{\mathbf{p}})$$

using equations 5.8 or 5.10. If the child is honest, we use equations 5.8, otherwise we use

5.10. If the tree architecture has more than one depth level, the child agent then becomes the parent of the next depth level, and the same process is repeated for its children until the leaves of the tree are reached. It can be observed that $P(\gamma_{\mathbf{c}})$ is a binary coefficient in equation 5.14. We evaluate this coefficient for every possible parent to child state combinations in equations 5.15.

$$P[\text{child} \in \overline{H} \cap T | \text{parent} \in H \cap C \cap T] = 0 \quad (5.15a)$$

$$P[\text{child} \in \overline{H} \cap \overline{T} | \text{parent} \in H \cap C \cap T] = 1 \quad (5.15b)$$

$$P[\text{child} \in \overline{H} \cap T | \text{parent} \in H \cap \overline{C} \cap T] = 1 \quad (5.15c)$$

$$P[\text{child} \in \overline{H} \cap \overline{T} | \text{parent} \in H \cap \overline{C} \cap T] = 0 \quad (5.15d)$$

$$P[\text{child} \in \overline{H} \cap T | \text{parent} \in \overline{H} \cap C \cap T] = 0 \quad (5.15e)$$

$$P[\text{child} \in \overline{H} \cap \overline{T} | \text{parent} \in \overline{H} \cap C \cap T] = 1 \quad (5.15f)$$

$$P[\text{child} \in \overline{H} \cap T | \text{parent} \in \overline{H} \cap C \cap \overline{T}] = 0 \quad (5.15g)$$

$$P[\text{child} \in \overline{H} \cap \overline{T} | \text{parent} \in \overline{H} \cap C \cap \overline{T}] = 1 \quad (5.15h)$$

$$P[\text{child} \in \overline{H} \cap T | \text{parent} \in \overline{H} \cap \overline{C} \cap T] = 0 \quad (5.15i)$$

$$P[\text{child} \in \overline{H} \cap \overline{T} | \text{parent} \in \overline{H} \cap \overline{C} \cap T] = 0 \quad (5.15j)$$

$$P[\text{child} \in \overline{H} \cap T | \text{parent} \in \overline{H} \cap \overline{C} \cap \overline{T}] = 1 \quad (5.15k)$$

$$P[\text{child} \in \overline{H} \cap \overline{T} | \text{parent} \in \overline{H} \cap \overline{C} \cap \overline{T}] = 0 \quad (5.15l)$$

Remark: We adopt a different notation in equations 5.15 to aid with readability. For example, equation 5.15a is equivalent to expressing said probability as:

$$P[\gamma_{\mathbf{c}} = t, \alpha_{\mathbf{c}} = \overline{h} | \beta_{\mathbf{p}} = c] = 0$$

This expression states that: a *coerced* parent will never pick a *dishonest* child in its *real* position. In other words: a coerced parent **always** picks a dishonest child in their *fake* position. Note that only the coercion attribute of the parent affects the result of equations 5.15. We include all three attributes to ease exposition, to show all possible state combinations.

Child to parent approval

The final stage shown in Figure 5.3 is finding the probability that a child approve their parent. This probability is also entirely dependent on the state of both the parent and the child, and is summarised in Table 5.2. For example, the entry corresponding to row 4 and column 1 states: the probability that an honest, coerced child in their true position *approves* a dishonest, non-coerced parent in their fake position is always 1.

Parent Child	$\overline{H} \cap \overline{C} \cap \overline{T}$	$\overline{H} \cap \overline{C} \cap T$	$H \cap \overline{C} \cap T$	$H \cap C \cap T$	$\overline{H} \cap C \cap \overline{T}$	$\overline{H} \cap C \cap T$
$\overline{H} \cap \overline{C} \cap \overline{T}$	0	0	0	1	0	0
$\overline{H} \cap \overline{C} \cap T$	0	0	0	0	0	0
$H \cap \overline{C} \cap T$	0	0	1	1	0	0
$H \cap C \cap T$	1	0	1	1	1	0
$\overline{H} \cap C \cap \overline{T}$	0	0	0	1	1	0
$\overline{H} \cap C \cap T$	0	0	0	0	0	0

Table 5.2: Approvals of a child to a parent.

This table can be expressed as a matrix:

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (5.16)$$

Thus, all components necessary to characterise whether a parent will receive an approval from their child have been mathematically formalised. We define $\mathcal{P}_{u,d}$ as the probability that a prover $\mathbf{p}$, with state $x(\mathbf{p}) = s_u$ will receive an approval from a witness at depth level d of the witness tree. This probability $\mathcal{P}_{u,d}$ can be expressed as follows:

$$\mathcal{P}_{u,d} = e_u^T (M \odot A)^d \mathbf{1} \quad (5.17)$$

where e_u is the u -th element of the canonical base of $\mathbb{R}^6$ and $\odot$ represents the element-wise matrix multiplication (Hadamard product) and $\mathbf{1} = [1, 1, 1, 1, 1, 1]^T$. Then, the probability that a prover $\mathbf{p}$, with state $x(\mathbf{p}) = s_u$ will have enough approvals is equivalent to the probability that, for each level d of the tree, $D_d \geq t \cdot n_d$, where D_d is the number of approvals at a given depth level d . The number of approvals D_d can be modelled as a Binomial distribution with parameter $\mathcal{P}_{u,d}$. Therefore, we can formalize the probability of satisfying criterion 1 as

$$P(\text{Criteria 1} = \text{True} | s_u, \theta) = P(D_d \geq t \cdot n_d, \forall d \in \{1, \dots, h\})$$

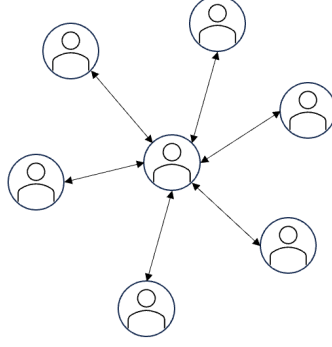


Figure 5.4: Visual representation of a tree architecture with dimensions $\theta = \{t = 1, h = 1, w_1 = 6\}$.

Which in turn can be expressed as:

$$\prod_{d=1}^h \sum_{k=\lceil tn_d \rceil}^{n_d} \binom{n_d}{k} \mathcal{P}_{u,d}^k (1 - \mathcal{P}_{u,d})^{n_d-k}. \quad (5.18)$$

Equation 5.18 allows us to quickly simulate the behaviour of T-PoP for different values of θ (under the assumption that $P(\text{Criteria 2} = \text{True} | s_u, \theta) = 1$ and $P(\text{Criteria 3} = \text{True} | s_u, \theta) = 1$, which holds true for large agent densities). As an example, consider Figures 5.5 and 5.7 which show, for values of p_h and p_c ranging in the interval $[0, 1]$, the performance of the T-PoP for two set of parameters, respectively: $\theta_1 = \{t = 1, h = 1, w_1 = 6\}$ and $\theta_2 = \{t = 1, h = 2, w_1 = 2, w_2 = 2\}$. Both tree architectures are depicted in Figures 5.4 and 5.6 respectively. Later, in Section 5.6 we show that the theoretical behaviour closely matches the behaviour of a detailed agent-based simulator of TPoP.

Furthermore, we can define the probability that a child will confirm that they see the parent that selected them (i.e., that an edge exists between a parent and child). This probability can be expressed as follows:

$$I_{d+1} = I_d(M \odot A) \quad (5.19)$$

where I_d is the probability vector of a parent being in any given state, at depth level d of the tree. Using the initial starting state in equation 5.11, we can then compute the expected number of edges, $\mathbb{E}(e)$, in a tree of any given height, h , and branching factor, w_d :

$$\vec{v} = \sum_{d=1}^h \prod_{i=1}^d w_i (I_0(M \odot A)^d) \quad (5.20)$$

$$\mathbb{E}(e) = \mathbf{1}^\top \vec{v} \quad (5.21)$$

Using equations 5.18 and 5.20, we can compute the expected number of edges in a prover's tree as the values of p_h and p_c range from 0 to 1. Figure 5.8 shows the results of equation 5.21.

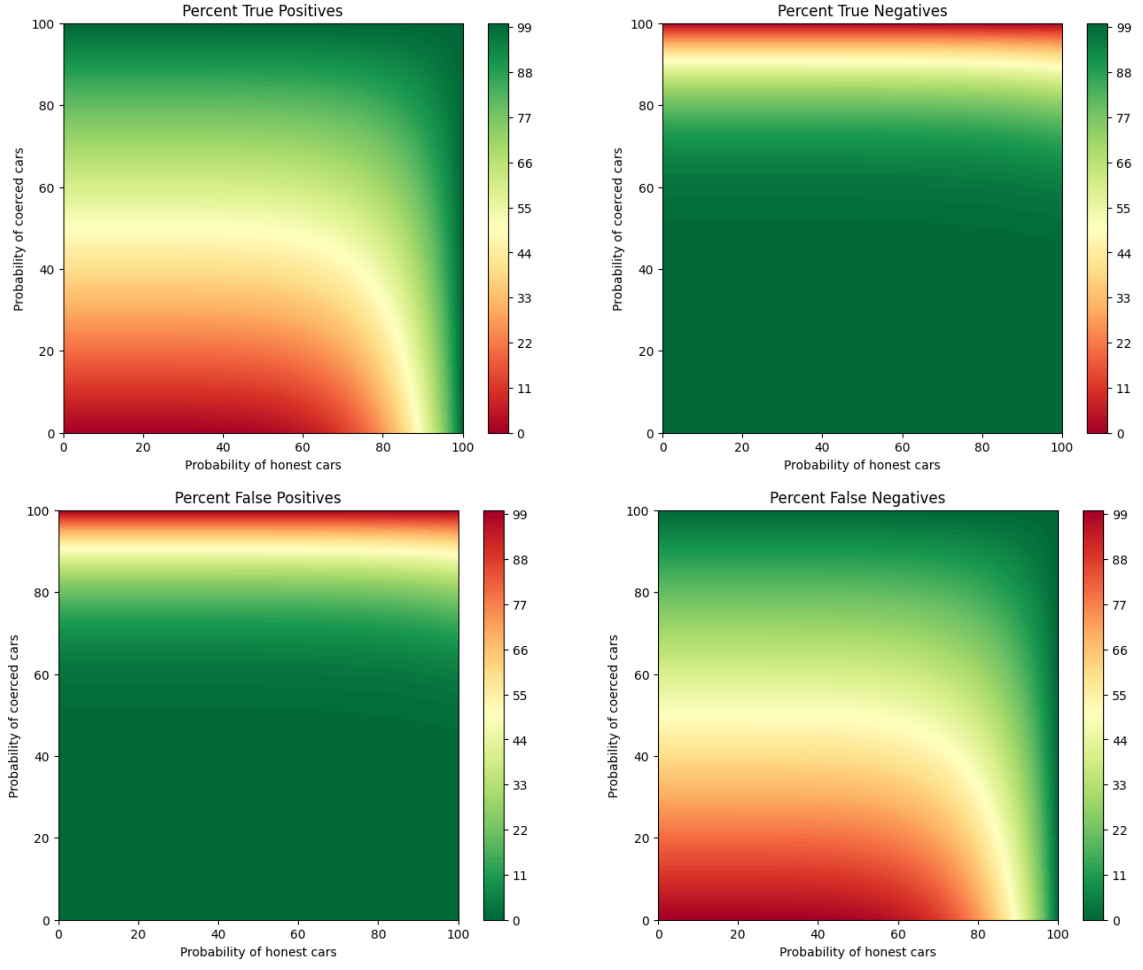


Figure 5.5: Theoretical performance of TPoP with $\theta_1 = \{t = 1, h = 1, w_1 = 6\}$.

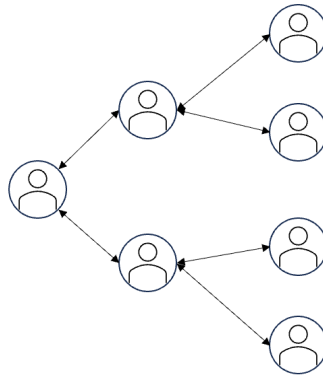


Figure 5.6: Visual representation of a tree architecture with dimensions $\theta_2 = \{t = 1, h = 2, w_1 = 2, w_2 = 2\}$.

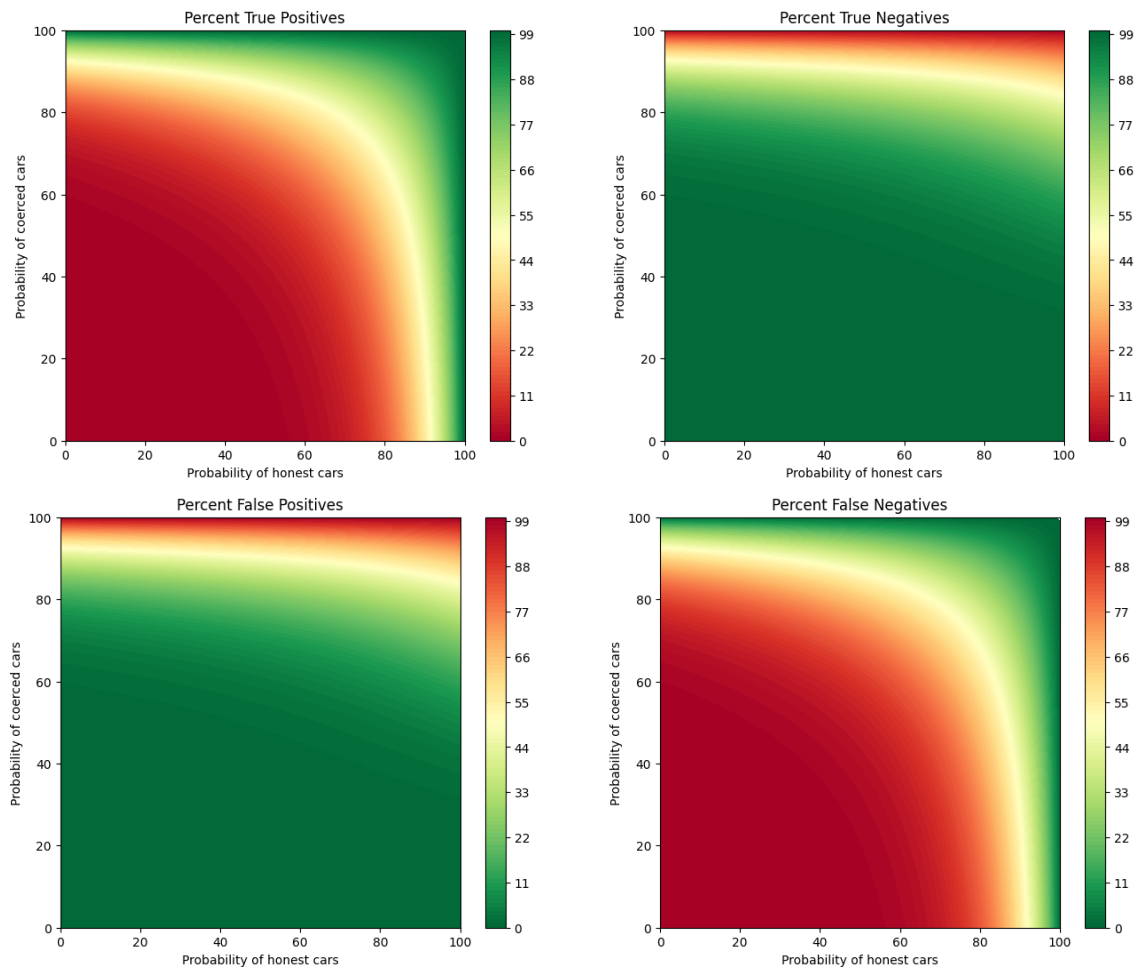
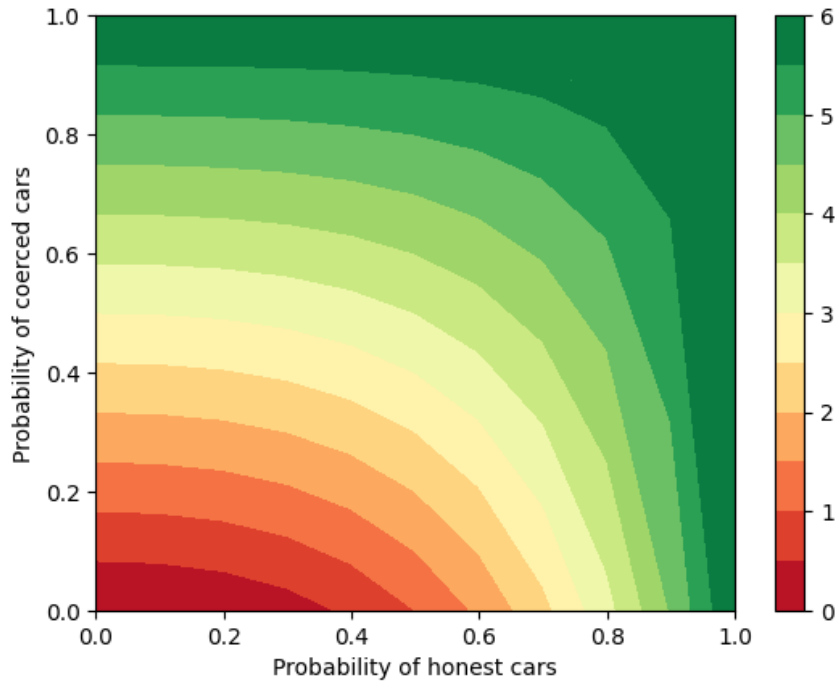
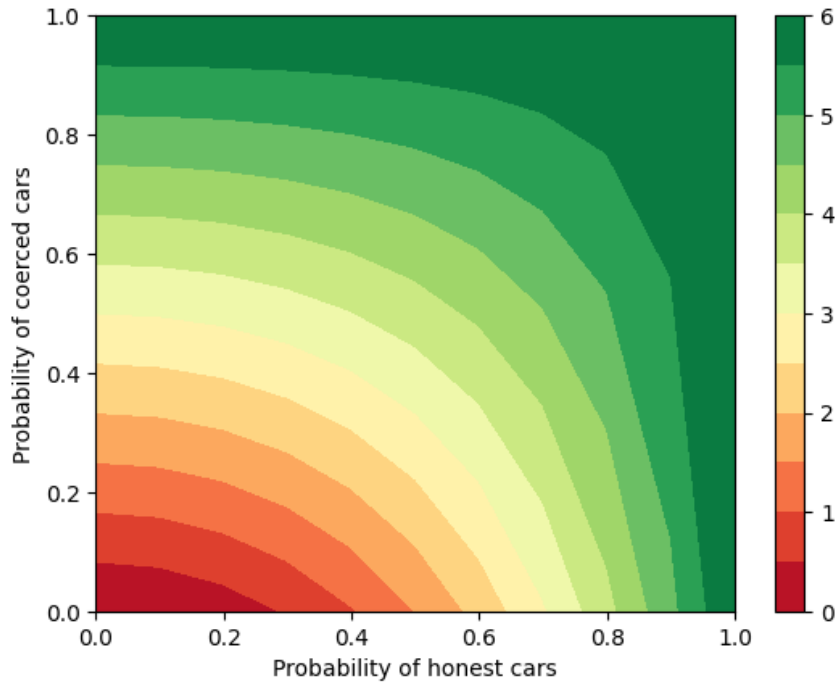


Figure 5.7: Theoretical performance of TPoP with $\theta_2 = \{t = 1, h = 2, w_1 = 2, w_2 = 2\}$.



(a) Tree of height 2 and branching factor 2, with threshold = 100%.



(b) Tree of height 1 and branching factor 6, with threshold = 100%.

Figure 5.8: Expected number of edges in the tree.

We then conduct 423,500 Monte-Carlo agent-based simulations, shown in Figure 5.9, that count the number of edges in each agent's tree after running T-PoP. In them, we create a uniform grid of 5 by 5 square units, and a total of 3500 agents, meaning the density is 140 agents per unit square. Each agent is set to have a range of sight equivalent to the size of the unit square, meaning there are 140 agents within their range of sight. If the density of agents is not sufficiently high, an agent will not have enough edges in their tree because it does not have sufficient agents in its vicinity to name as children. We therefore purposefully select a high value of agents, to ensure that the probability of a tree not having sufficient edges is not due to a lack of nearby agents being available, but rather because the agents do not approve each other. This is precisely what we are modelling in equation 5.21. These simulations are conducted with a threshold 100%. The results obtained are in complete agreement with the mathematical model presented in Figure 5.8, thereby demonstrating its validity. The code and the simulation data are available in our [GitHub](#) repository.

5.3.3 Condition 2: Tree Has Sufficient Nodes

The second condition is modelled by considering the following assumptions:

- We assume a constant average density of agents μ ;
- We assume that agents are scattered uniformly across the environment, independently from one another and at random.

Accordingly, let N_a be the amount of neighbours of agent a . This random variable can be approximated by a Poisson Point Process of parameter $\lambda = \mu \cdot \pi r^2$, where r is the range of sight of the agent. It follows then that the probability that there are at least n_d neighbours around agent a can be immediately computed as

$$P(N_a > n_d) = 1 - P(N_a \leq n_d) = 1 - \sum_{k=0}^{n_d-1} e^{-\lambda} \frac{\lambda^k}{k!} \quad (5.22)$$

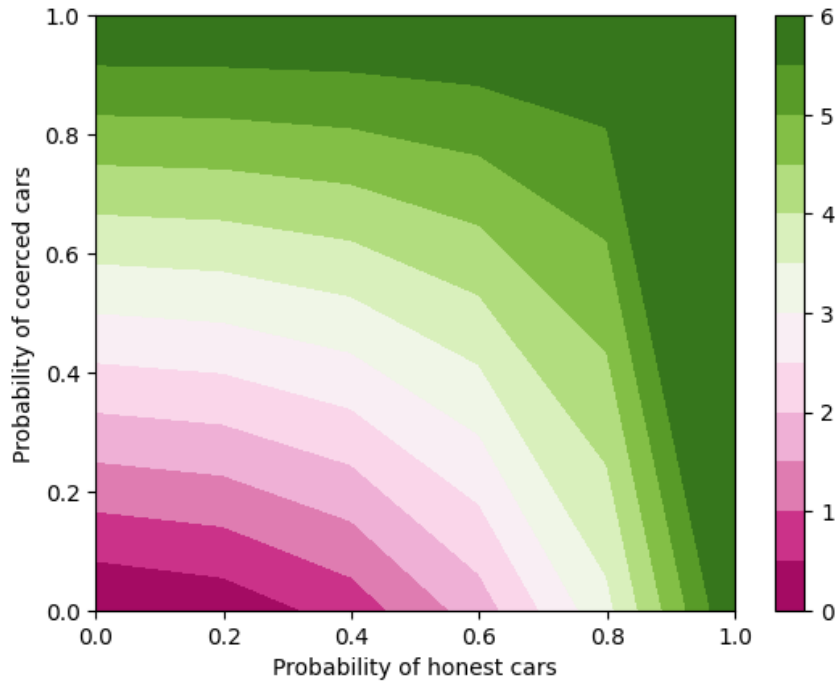
This means, that for TPoP with a certain θ the probability of satisfying condition 2, $P(\text{Criteria 2} = \text{True} | s_u, \theta)$, for all s_u , can be expressed as

$$\prod_{i=0}^{h-1} \prod_{j=1}^{n_i} \left(1 - \sum_{k=0}^{n_{i+1}-1} e^{-\lambda} \frac{\lambda^k}{k!} \right), \quad (5.23)$$

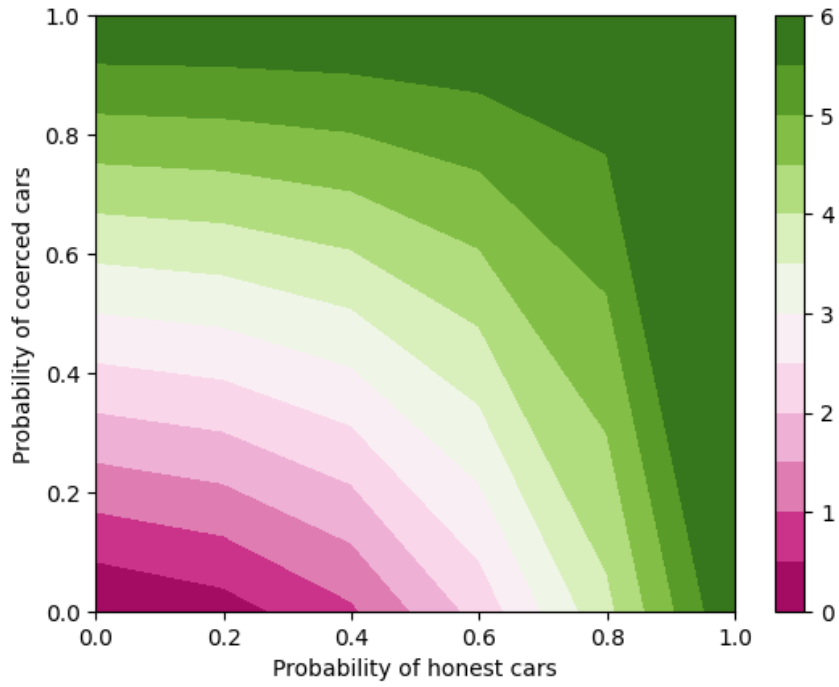
which is simply the product of the probabilities that each parent in the tree will have enough neighbours.

5.3.4 Condition 3: All Nodes Are Unique

In this subsection we analyse the probability of constructing a tree with distinct nodes given a certain density of agents. Specifically we focus on the two structures that are



(a) Tree of height 2 and branching factor 2, with threshold = 100%.



(b) Tree of height 1 and branching factor 6, with threshold = 100%.

Figure 5.9: Number of edges in agent's trees after running T-PoP.

presented in Section 5.6:

- TPoP with $\theta_1 = (t = 1, h = 1, w_1 = 6)$,
- TPoP with $\theta_2 = (t = 1, h = 2, w_1 = 2, w_2 = 2)$.

For brevity, we denote $P(\text{Criteria 3} = \text{True}|s_u, \theta)$ as $p(\theta)$ (as the probability of condition 3 being true does not depend on the state of the prover). In the remainder of this section we make the assumption that each agent selects witnesses among its neighbours without replacement. In other words, we assume an agent can select another agent as witness, and not eliminate it from its subsequent choice of agents to choose from. Although it may be an unrealistic assumption (as in practice each prover would select distinct witnesses and they would not deliberately try to hinder their own proof), it allows us to provide a lower bound on the probability of selecting a tree made of unique nodes.

Let us now assume that a population of N agents are distributed uniformly at random on a plane of size $l \times l$, with every agent's range of sight set to constant r . We can view agents as nodes in a graph, where there exists a link between two nodes if and only if the distance between corresponding agents is less than or equal to r . This graph structure is called a Random Geometric Graph and it has a known mean node-degree $\langle k \rangle \simeq \pi N r^2 / l^2$ [398]. The mean node-degree $\langle k \rangle$ of the graph is equivalent to the expected number of neighbours in the population of agents. For our calculations we make use of a function $\kappa_{n,y}$ defined as:

$$\kappa_{n,y} = \frac{\binom{\langle k \rangle - n}{y}}{\langle k \rangle^y / y}, \quad (5.24)$$

representing the probability of picking $n + y$ unique agents from a set of $\langle k \rangle$ neighbours, given that n options are picked beforehand (eliminated) and y new agents are to be picked. For the case of parameter θ_1 , $p(\theta_1)$ is straightforward to compute and it is equal to $\kappa_{0,6}$.

For the tree with parameter θ_2 , finding the probability that all nodes are unique is more complex. Let us consider a tree started by a prover a_1 at the root, with a tree height of $h = 2$, and branching factor of $w = 2$: in this case, each parent in the tree chooses two of its neighbours at random (with replacement) as its children. We begin by computing the probability that the nodes at level 1 will not choose themselves. Let a_2 and a_3 (at depth level 1 in the tree) be the children of a_1 (which is at level 0) in the tree, selected at random within a_1 's field view; $a_2, a_3 \in \overline{D}_1$. Agents $a_4, \dots, a_7$ are the leaves of the tree (forming level 2); see Fig. 5.10A. We denote the probability of 7 distinct agents forming the three levels of the tree (L_0, L_1, L_2) as $P(L_0, L_1, L_2)$, which can be written as:

$$P(L_0, L_1, L_2) = P(L_0)P(L_1|L_0)P(L_2|L_0, L_1), \quad (5.25)$$

with $P(L_0) = 1$, $P(L_1|L_0)$ denoting the probability that $a_1 \neq a_2 \neq a_3$, and $P(L_2|L_0, L_1)$ being the probability that $a_1, \dots, a_7$ are different agents given that a_1, a_2, a_3 are distinct.

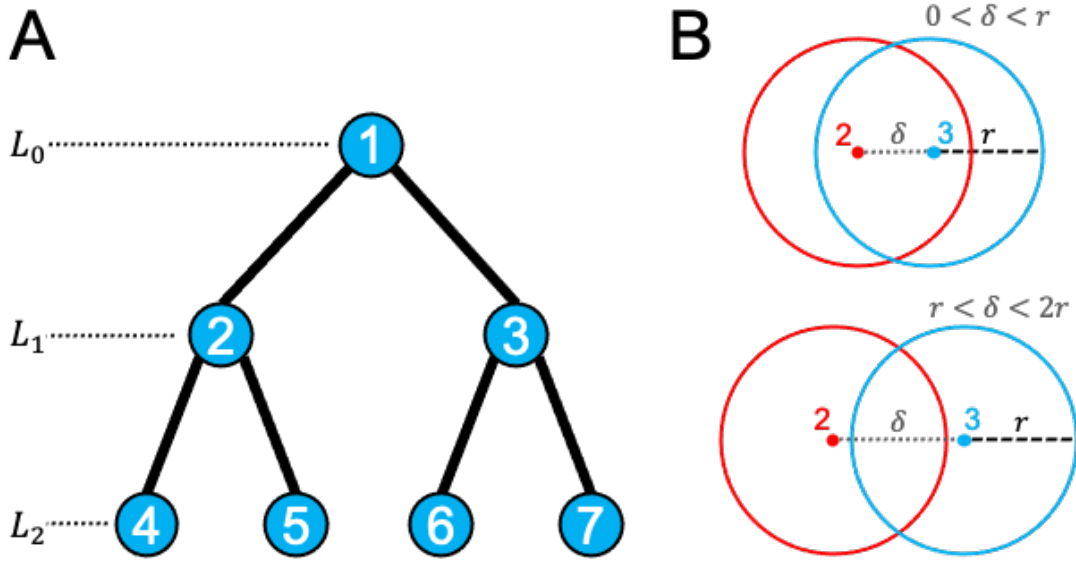


Figure 5.10: A) A tree of height $h = 2$ and branching factor $w = 2$, with agents and levels labelled. B) Two separated scenarios visualized, when agents a_2 and a_3 selected at random in the field of view of a_1 are $\delta < r$ nearby (top) versus $\delta > r$ far apart (bottom).

The probability of having $a_1 \neq a_2 \neq a_3$ can be calculated in a straightforward manner, since a_1 has an expected number of $\langle k \rangle$ neighbours with none of them eliminated as options. Thus, using equation (5.24), we obtain the probability of picking two different agents at random from a set of size $\langle k \rangle$, i.e., $P(L_1|L_0) = \kappa_{0,2}$. Now, with a_2 and a_3 picked at random within the field of view of a_1 , which is a disk of radius r , we use the probability density function for the distance between the two agents $\delta = \|p_2 - p_3\|_2$. This is simply the probability density function for the distance between two random points in a disk:

$$f(\delta) = \frac{4\delta}{\pi r^2} \cos^{-1}\left(\frac{\delta}{2r}\right) - \frac{2\delta^2}{\pi r^3} \sqrt{1 - \frac{\delta^2}{4r^2}}. \quad (5.26)$$

If $\delta < r$ then a_1 is a neighbour of a_2 and vice versa. Thus for each agent, one neighbour out of the expected number of neighbours is eliminated (in order to have distinct agents in the tree). See Figure 5.10B (top panel). Also, for all possible values of δ , i.e., $0 < \delta < 2r$, the probability of selecting a common neighbour by either a_1 or a_2 equals the fraction of their field of view that is overlapping. This can be calculated by $\mathcal{A}_\delta$, a function of δ , defined as below:

$$\mathcal{A}_\delta = \left[2r^2 \cos^{-1}\left(\frac{\delta}{2r}\right) - \frac{\delta}{2} \sqrt{4r^2 - \delta^2} \right] / \pi r^2, \quad (5.27)$$

where the numerator of the right hand side is the overlapping area of two circles with radius r with centres δ far apart, and the denominator is the area of the circle with radius r .

Without loss of generality, assume that a_2 picks two random neighbours a_4 and a_5 as its children (starting with $a_1 \neq a_2 \neq a_3$). The prover agent a_1 is certainly a neighbour of a_2 , and if $\delta < r$ ($\delta > r$) then a_3 is (is not) a neighbour of a_2 , and $a_1, \dots, a_5$ are distinct with

probability κ_2 (κ_1). Also, with probability $\mathcal{A}_\delta$, a_4 is in the intersection between the fields of view of a_2 and a_3 , thus reducing the number neighbours which a_3 can pick from (for its children to be distinct from the agents already appearing in the tree). With probability $\mathcal{A}_\delta^2$, both a_4 and a_5 are neighbours of a_3 and reducing a_3 's options by two, and with probability $\mathcal{A}_\delta(1 - \mathcal{A}_\delta)$, a_4 is a neighbour of a_3 and a_5 is not. Using the above logic we can approximate the probability of having 7 distinct agents in the tree, given that $a_1 \neq a_2 \neq a_3$, i.e.:

$$\begin{aligned}
P(L_2|L_0, L_1) = & \int_0^r f(\delta) \kappa_{2,2} \\
& \cdot [(1 - \mathcal{A}_\delta)^2 \kappa_{2,2} + 2(1 - \mathcal{A}_\delta) \mathcal{A}_\delta \kappa_{3,2} + \mathcal{A}_\delta^2 \kappa_{4,2}] d\delta \\
& + \int_r^{2r} f(\delta) \kappa_{1,2} \\
& \cdot [(1 - \mathcal{A}_\delta)^2 \kappa_{1,2} + 2(1 - \mathcal{A}_\delta) \mathcal{A}_\delta \kappa_{2,2} + \mathcal{A}_\delta^2 \kappa_{3,2}] d\delta.
\end{aligned} \tag{5.28}$$

Finally, inserting equation (5.28) into equation (5.25), we have the theoretical approximation for the probability of the tree being comprised of strictly distinct agents, i.e., $p(\theta_2) = P(L_0, L_1, L_2) = \kappa_{0,2} \cdot P(L_2|L_0, L_1)$. This value of $P(L_0, L_1, L_2)$, given a constant range of sight r , is only dependent on the density of agents N/l^2 and can be calculated numerically.

The theoretical approximation of $P(L_0, L_1, L_2)$ can be validated via a numerical experiments. To do so, we uniformly scatter N agents on a unit 2-dimensional square, independently and at random. A tree is then sampled, by randomly drawing a prover from the population, which then draws two agents at random (with replacement) within its field of view (here, r is set to 0.1) these drawn agents then repeat the same process again, until the desired height of the tree is reached. We vary N from 200 to 10,000, and for each N sample 10,000 trees at random and count those made up of 7 distinct agents. Figure 5.11, compares the fraction of sampled trees with all distinct agents (red curve) with the theoretical approximation given by equations (5.25)-(5.28). As demonstrated by the results, the approximation performs well, converging fast to the numerical observation as the density of agents increases.

As a final remark we note that it is very difficult to obtain a general expression to compute the probability of finding distinct agents in a tree with for any θ . It is however quite straightforward to approximate $p(\theta)$, for a given $\langle k \rangle$ and a given r , by making use of Monte-Carlo simulations.

5.4 T-PoP's Complexity

T-PoP is comprised of two algorithms: **Verification** and **Checks**. These are defined in Algorithms 4 and 3 accordingly. **Verification** is a (reverse) breadth-first-search algorithm, meaning its runtime in the worst case scenario is $O(|V| + |E|)$, where V is the number of nodes and E the number of edges in the tree data structure [134]. The **Verification**

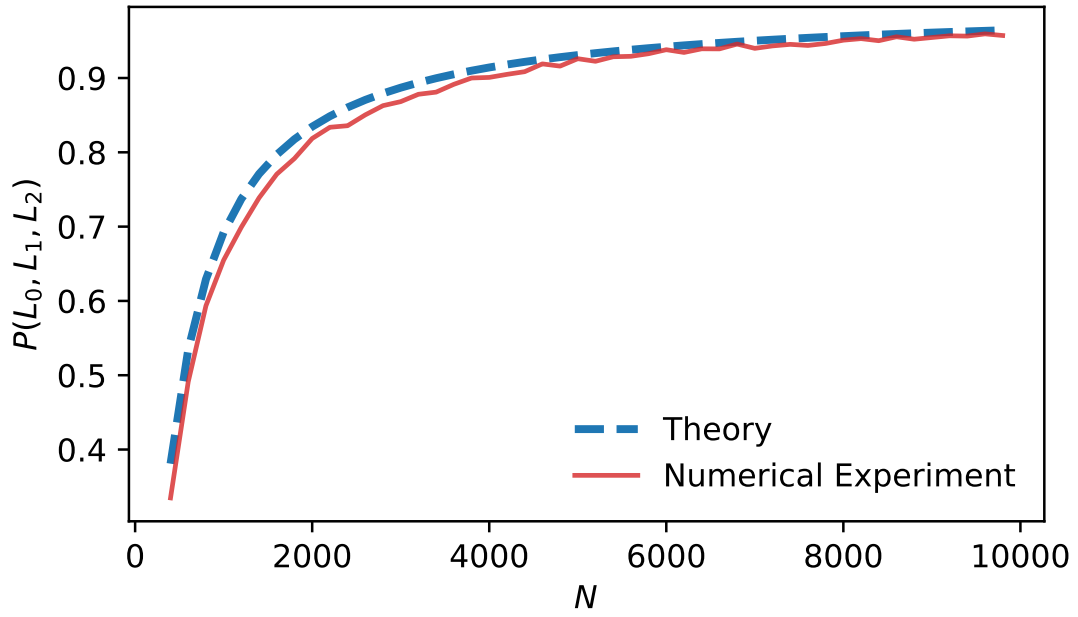


Figure 5.11: Validation of theoretical approximation for probability of a randomly sampled tree ($h = w = 2$) containing 7 distinct agents. The results show the probability as a function of the number of agents N uniformly distributed independently and at random over a unit square plane. The red curve shows the probability calculated through a numerical experiment, where a tree is generated following the tree-building algorithm 2, starting with a randomly selected prover. The dashed curve shows the value of the theoretical approximation denoted $P(L_0, L_1, L_2)$ calculated by equations (5.25)-(5.28).

algorithm calls the **Checks** algorithm at each node, and the **Checks** algorithm has a time complexity of $O(|V|)$. So T-PoP's time complexity is at worst $O(|V|^2 + |E||V|)$ for the prover. However, it is $O(|V|)$ for each verifier. Since there is only one prover, and more verifiers, T-PoP is a lightweight algorithm to run for most of its participants. We note that all necessary computations are done locally by agents, both provers and verifiers, and that there is no multi-party computation involved. Each agent performs the necessary verifications and then shares these with other relevant agents. Thus, there is no single point of failure for the algorithm. In terms of communication overhead, the minimum total number of message exchanges in a tree is $2 * V$. This is the case if the tree only has one depth level: the parent sends a query to their child, and the child answers if they approve the parent or not. If the depth is increased, the child has to respond to their parent but also name their own children, query them, and return their response to the parent. However, if communication overhead is a concern, the tree dimension can be changed to reduce complexity without sacrificing the number of agents in the tree by simply reducing the depth level of the tree.

5.5 Detecting Platoon Attacks

Given T-PoP's collaborative nature, it is susceptible to a platooning attack. In it, one *dishonest* agent coerces a subset of agents to *approve* their (fake) *claimed* position. This cluster of agents cruise together and because the *dishonest* agent has sufficient approvals, T-PoP will consistently fail in detecting that this claim is dishonest. To address this attack vector we propose a methodology to probabilistically determine if a tree is comprised of a platoon, or if it originates from a general adversarial scenario. We stress, for the latter case, dishonest and coerced agents may still appear in the tree; we simply do not assume that **all** the members in the tree are cooperating. The methodology is as follows: first, we assume that an estimate can be made regarding the percentage of honest and coerced agents in the network. In other words, that p_h and p_c are known or can be estimated, and that therefore $1 - p_h$ and $1 - p_c$ can be computed. We consider this assumption to be reasonable, as most Byzantine Fault Tolerant systems also make assumptions regarding the number of honest agents available in the network [306], [109]. A tree is then constructed, with a certain number of agents and approvals in it. TPoP's purpose is to determine if the prover is honest about their position claim. In theory, the state of each agent in that tree is unknown (i.e.: if they are honest or dishonest and coerced or non-coerced). We constructed a model in Section 5.3 that characterises the probability that TPoP will consider a prover truthful. Following the exact same steps from Section 5.3, we construct a similar model that also characterises the probability that a prover is honest, but when considering the possible states that the agents can be in, we only include those that can participate in a platoon. The reader can think of this modification as shrinking the possible state space of the agents to only the malicious/cooperative ones. Both models characterise the probability that TPoP considers a prover truthful, but they differ in answering the following: *for a given tree, what possible agents may this tree have originated from?* Using both models, one can evaluate the expected number of approvals in a tree for each adversarial scenario.

Then, given a tree with a certain number of approvals, it is possible to determine which scenario that number of approvals is more likely to have originated from.

Probabilistic Detection Model

By definition, the leader of a platoon (and therefore the prover in the tree) is dishonest. This means that a platoon is always initiated by an *dishonest* agent in their *claimed* position, and not their *real* position. We can therefore restrict the state space of the probability vector defined in equation 5.11 to the following:

$$\tilde{I}_1 = [P(s_1), P(s_4), P(s_5)] \quad (5.29a)$$

$$P(s_4) = 0 \quad (5.29b)$$

Recall that this vector represents the probability of the prover agent being in a certain state.

Remark: We include state s_4 in the set of possible agent states, since the agent is *coerced*, so it may be part of the set of agents approving the platoon *prover*, however, it may never be the *prover*, since it is *honest*. We then modify the matrix 5.12 to become:

$$\tilde{M} = \begin{bmatrix} m_{1,1} & m_{1,4} & m_{1,5} \\ m_{4,1} & m_{4,4} & m_{4,5} \\ m_{5,1} & m_{5,4} & m_{5,5} \end{bmatrix} \quad (5.30)$$

where the entries of $\tilde{M}$ are computed following equation 5.14, but we only consider parent to child combinations where the agents are in states $\{s_1, s_4, s_5\}$, so $u, v \in \{1, 4, 5\}$. Similarly, to create the platoon approvals matrix, $\tilde{A}$, we sample the entries $a_{i,j}$, of A that correspond to the approvals of all the possible agent pair combinations in states $\{s_1, s_4, s_5\}$.

$$\tilde{A} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (5.31)$$

With these components, we can proceed to compute the expected number of edges in a platoon tree by substituting $\tilde{I}_d$, $\tilde{M}$ and $\tilde{A}$ into the corresponding terms in equation 5.20 and 5.21. We then show the model's behaviour in Figures 5.12 and 5.13. To summarise, the steps above are identical to those in section 5.3, but we subsample the agent state space to only include the agents that could participate in a platoon. This is equivalent to assuming no honest, non-coerced agents appeared in the tree, because this is in fact the definition of a platoon.

We highlight that to detect honest trees, it does not suffice with simply counting the number of edges. These edges must be connecting nodes that are honest, i.e., agents in states s_3

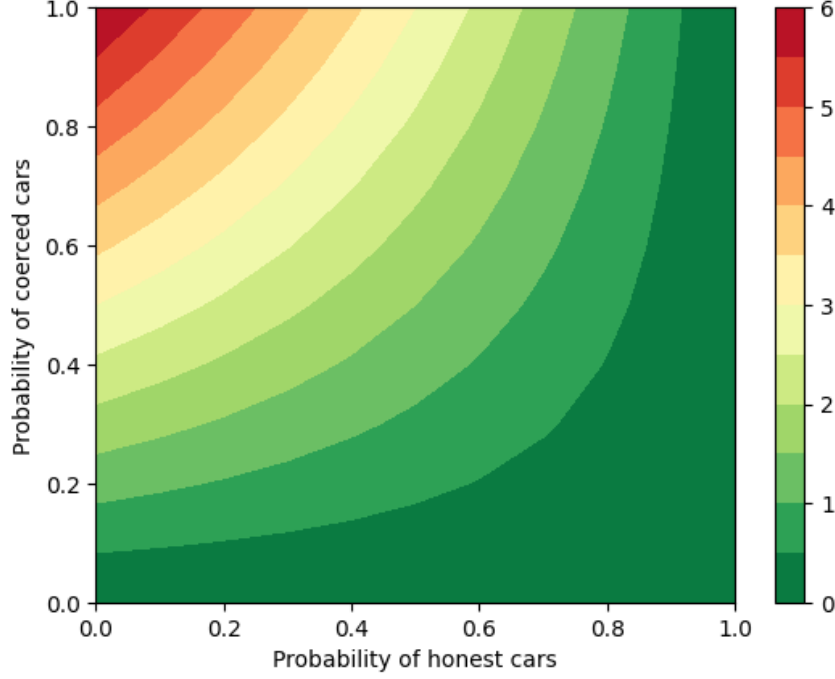


Figure 5.12: Expected number of edges in a platoon tree of height 1 and branching factor 6, with threshold = 100%.

and s_4 . The challenge is that the state of a node is not known or observable when a tree is assessed. However, we may find the optimal conditions such that the probability of an edge connection between agents in states s_3 and s_4 are maximised. The elements of the matrix M that encode child to parent approvals that are honest are entries $m_{3,3}, m_{3,4}, m_{4,3}, m_{4,4}$. The optimal values of p_h and p_c that maximise the probability of these edges existing is denoted in Table 5.3.

$m_{i,j}$	Parent	Child	Optimal Point
$m_{3,3}$	s_3	s_3	$P(H) = 1, P(C) = 0$
$m_{3,4}$	s_3	s_4	$P(H) = 1, P(C) = 0.5$
$m_{4,3}$	s_4	s_3	$P(H) = 1, P(C) = 0.5$
$m_{4,4}$	s_4	s_4	$P(H) = 1, P(C) = 1$

Table 5.3: Optimal points to maximise honest to honest agent approvals.

5.6 Agent Based Simulations

We simulate the performance of T-PoP in both high density and low density cases, for varying operating parameters and ranging values of honest and coerced agents. Our purpose is to determine the security and reliability guarantees that T-PoP provides in both scenarios.

We construct an agent-based simulator, coded in Python, which is available in our [GitHub](#)

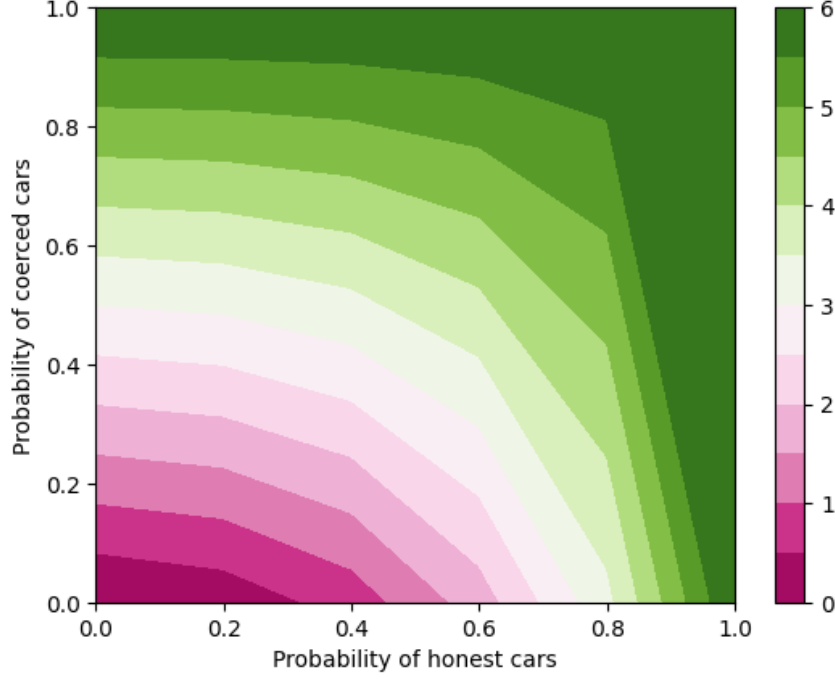


Figure 5.13: Expected number of edges in a platoon tree of height 2 and branching factor 2, with threshold = 100%.

Repository. In it, we create an environment of a given size, and instantiate a fixed number of agents with a fixed value for their range of sight. Agents are assigned an honesty, α_i , and a coercion, β_i , attribute independently and at random, following equations 5.7a and 5.7b. We repeat the simulations for values of p_h and p_c belonging to the interval $[0, 1]$. All agents are given a real position, within the bounds of the defined environment, and dishonest agents also have a claimed position that is different to their real position. These positions are uniformly and randomly distributed across the environment. We first simulate a low-density scenario, as shown in Figures 5.14, 5.15, 5.16 and 5.17. We initialise an environment of size 10 by 10 arbitrary square units, and instantiate 350 agents, such that each has a range of sight of 1 unit. Thus, agents have an average of 3.5 agents within their field of view. The environment size is arbitrary, our aim is to simulate a scenario where agents wishing to construct a tree have less available neighbours than the minimum required for the construction of the tree. In this case, agents see on average 3 but need at least 6 for a tree. This scenario is highly unrealistic, but serves to test T-PoP's performance in worst possible scenarios. Each agent constructs a tree, and then runs the T-PoP algorithms to obtain a proof-of-position. These simulations are carried out across the ranging values of p_h and p_c , starting from 0 to 100% in intervals of 10%. At each combination of p_h and p_c , 5 Monte Carlo simulations are run for every agent. In total, that is 211,750 T-PoP simulations.² We test the performance of T-PoP for the following operating conditions:

²There are 121 possible combinations of p_h and p_c when varying the values at intervals of 10%, 350 agents and 5 simulations per possible combination of p_h and p_c yields $5 * 121 * 350$.

$$\theta_1 = (t = 1, w_1 = 2, w_2 = 2, h = 2)$$

$$\theta_2 = (t = 1, w_1 = 6, h = 1)$$

$$\theta_3 = (t = 0.4, w_1 = 2, w_2 = 2, h = 2)$$

$$\theta_4 = (t = 0.4, w_1 = 6, h = 1)$$

We then consider a high-density scenario, as shown in Figures 5.18, 5.19, 5.20 and 5.21. We run these simulations with the density at which the probability of selecting all unique agents in the tree tends to 1. From Figure 5.11, this density is found to be 50 agents per unit squared, when each agent has a range of sight of 1 square unit. In these simulations we create a grid of 5 by 5 arbitrary square units, with 1250 agents in total, each with a range of sight of 1 square unit. The choice of environment size is also arbitrary, we simply choose parameters to ensure the average number of neighbours in each agent's range of sight is 50. We again run the simulations for each possible combination of p_h and p_c values, at intervals of 10%. In each combination of p_h and p_c values, we run T-PoP for each of the 1250 agents and repeat this process 5 times. In total, the number of simulations is 756,250. We do so for the same operating conditions previously outlined.

In order to assess the performance of TPoP for each scenario we use an empirical approximation of the True Negatives (security or TN) and of the True Positives (reliability or TP). For completeness we also consider the False Negatives (FN) and the False Positives (FP). Each classification instance is described as below:

- A true positive (TP) instance means TPoP correctly labelled an agent as honest.
- A true negative (TN) means TPoP correctly labelled an agent as dishonest.
- A false positive (FP) means TPoP incorrectly classified a dishonest agent as honest.
- A false negative (FN) means TPoP incorrectly classified an honest agent as dishonest.

$$TP\% = \frac{TP}{TP + FN} \times 100 \quad (5.32a)$$

$$FN\% = 100 - (TP\%) \quad (5.32b)$$

$$TN\% = \frac{TN}{TN + FP} \times 100 \quad (5.32c)$$

$$FP\% = 100 - (TN\%) \quad (5.32d)$$

Our assessment of performance is based on the measurement of four indicators in a simulated environment of spatially distributed agents: sensitivity and specificity, and the complement of each. Sensitivity (also known as recall) measures the proportion of honest agents correctly identified by the algorithm. This metric is the $TP\%$. Specificity measures

the rate of dishonest agents correctly classified as such. This measurement is $TN\%$. We also measure the complement of each of these two metrics to provide a complete picture of the algorithm’s performance. The False Negatives rate ($1 - \text{Recall}$), measures the proportion of actual honest agents that were missed by TPoP; this metric is less relevant in security critical contexts. It is more important to ensure that the False Positives rate ($1 - \text{Specificity}$) is minimised. False Positives represent the dishonest agents (with falsified positions) that were able to fool the algorithm and remain undetected. However, it is still important to measure the False Negatives rate to ensure that TPoP is not simply classifying all agents as dishonest to maximise security. We present all four indicators to ensure users can understand the trade-offs between them. Different contexts will warrant placing greater importance on different indicators. Greater reliability needs will necessitate higher recall/sensitivity rates, whereas increased security will translate to a need for higher specificity (and therefore a lower false positive rate). Notice that we consider sensitivity and specificity over precision because the correctness of TPoP’s classification is more important than how consistent these classifications are. A high precision can still mask TPoP’s inability to detect dishonest agents, since we assume in most real-world scenarios the percent of dishonest agents will be significantly lower than the honest ones.³

Remark: the results of the simulations illustrated in Figures 5.14 to 5.21 present artifacts for p_h belonging to $[0, 0.1]$ and $[0.9, 0.1]$. This is due to the fact that in those ranges there are either no honest agents and no coerced agents (due to the resolution of 10%). We note however that the results, aside from these artifacts, are consistent with the theoretical results presented in Figures 5.8 and 5.9.

³For example, many Byzantine Fault Tolerant systems rely on the assumption that the number of honest agents in the network is 3 times greater than the number of dishonest agents. [306], [109].

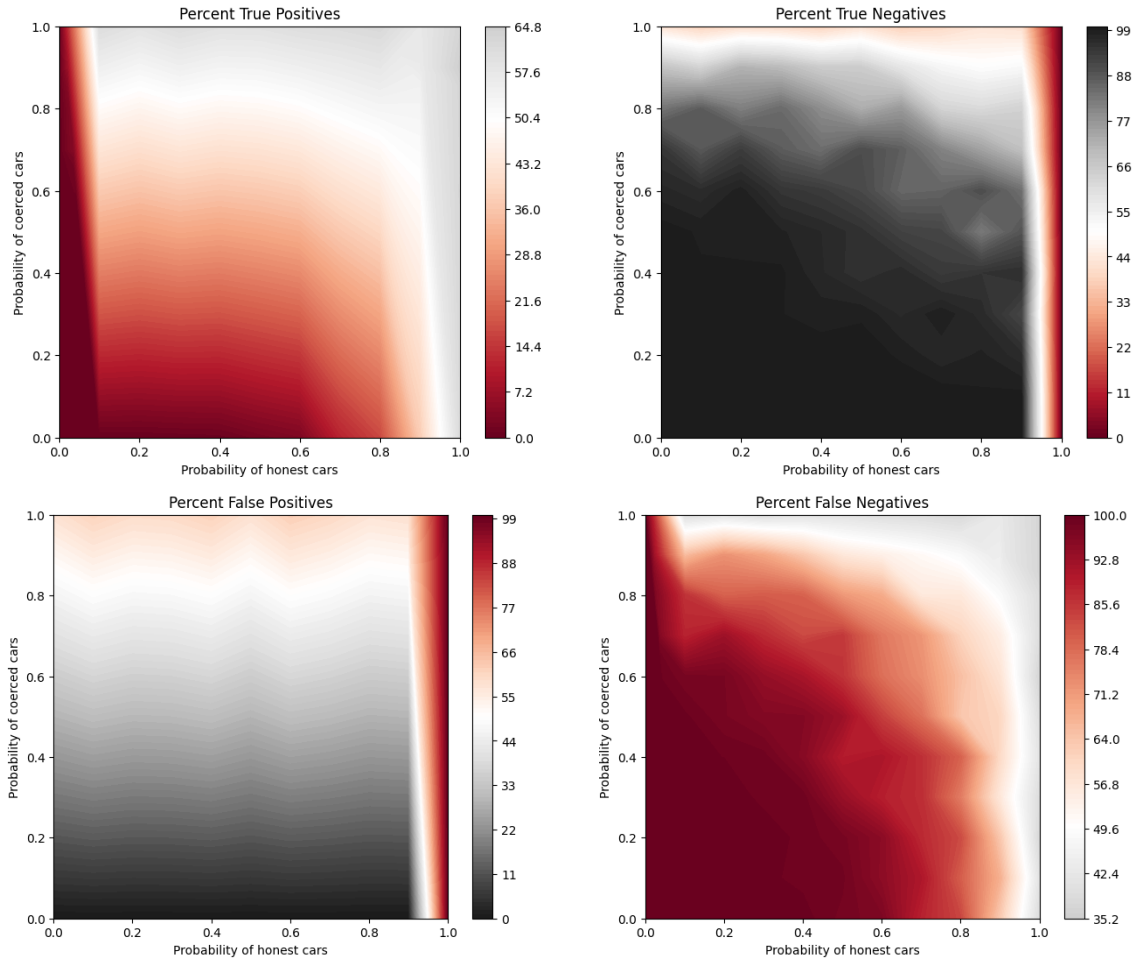


Figure 5.14: Low density performance of TPop with a tree of height 2 and branching factor 2, and a threshold = 100%.

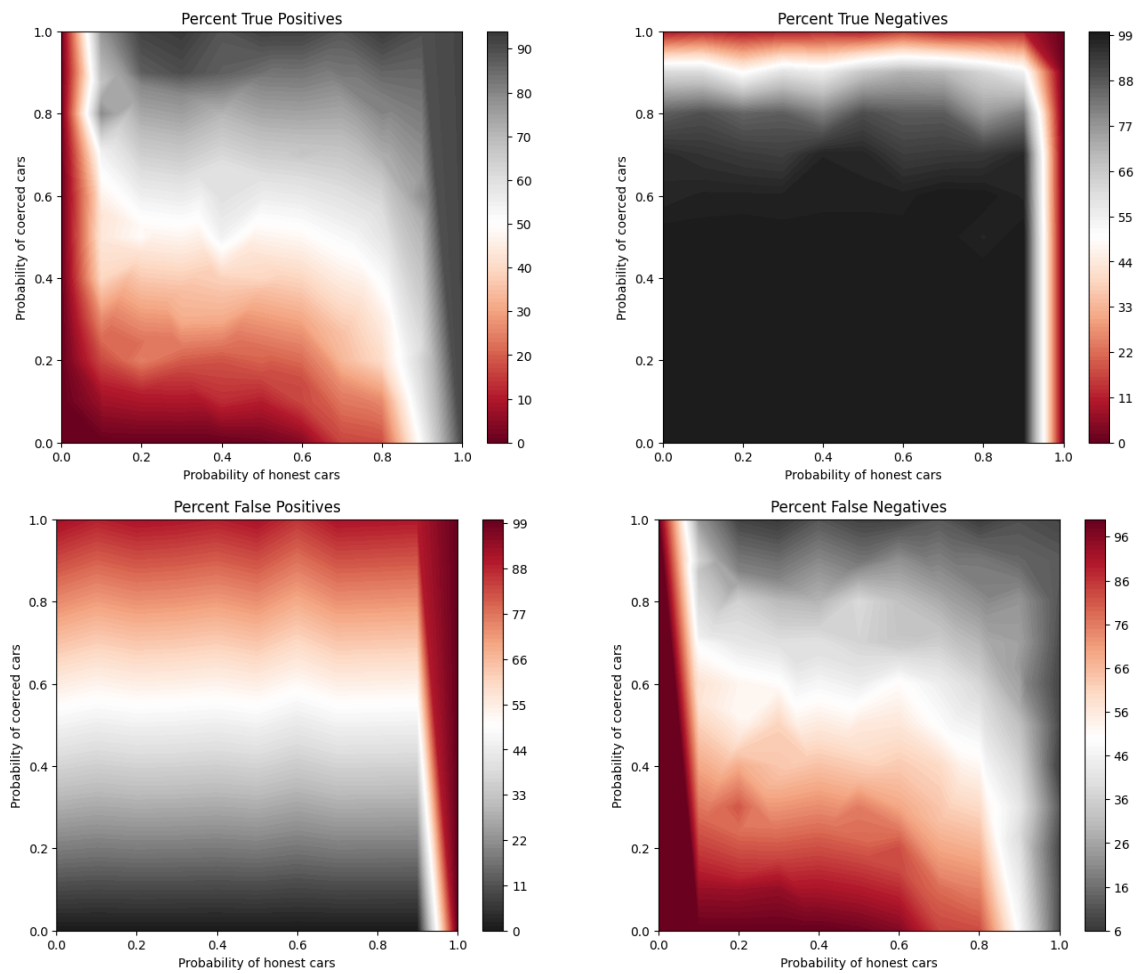


Figure 5.15: Low density performance of TPoP with a tree of height 1 and branching factor 6, and a threshold = 100%.

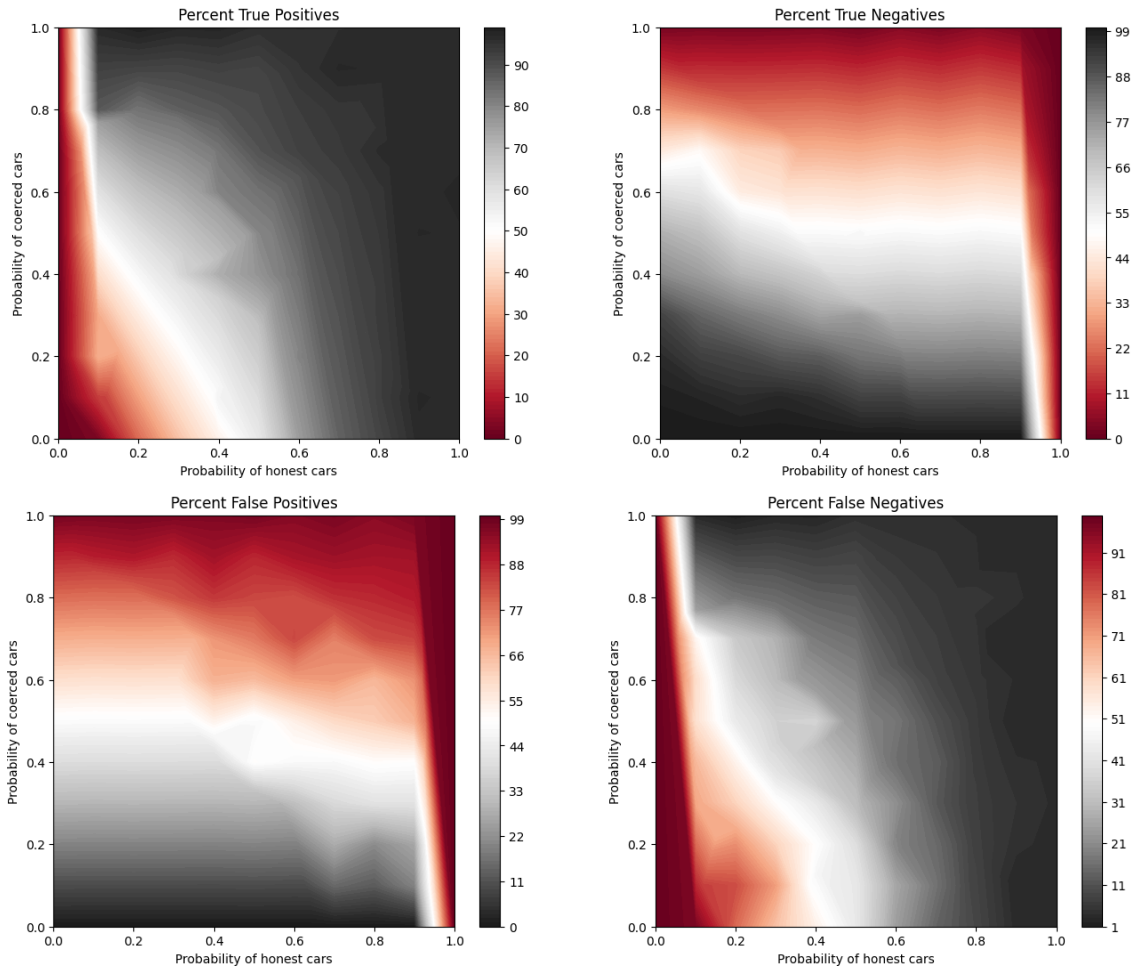


Figure 5.16: Low density performance of TPoP with a tree of height 2 and branching factor 2, and a threshold = 40%.

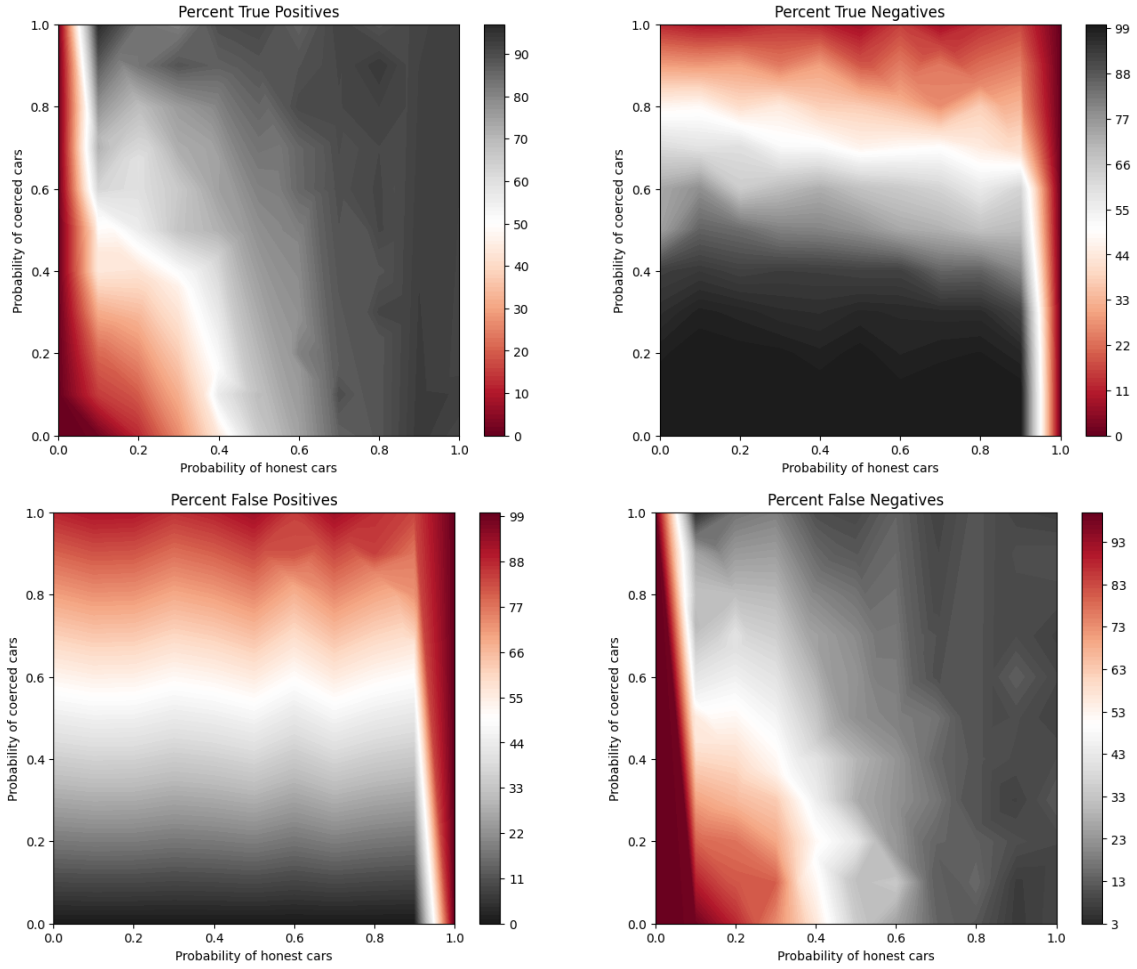


Figure 5.17: Low density performance of TPoP with a tree of height 1 and branching factor 6, and a threshold = 40%.

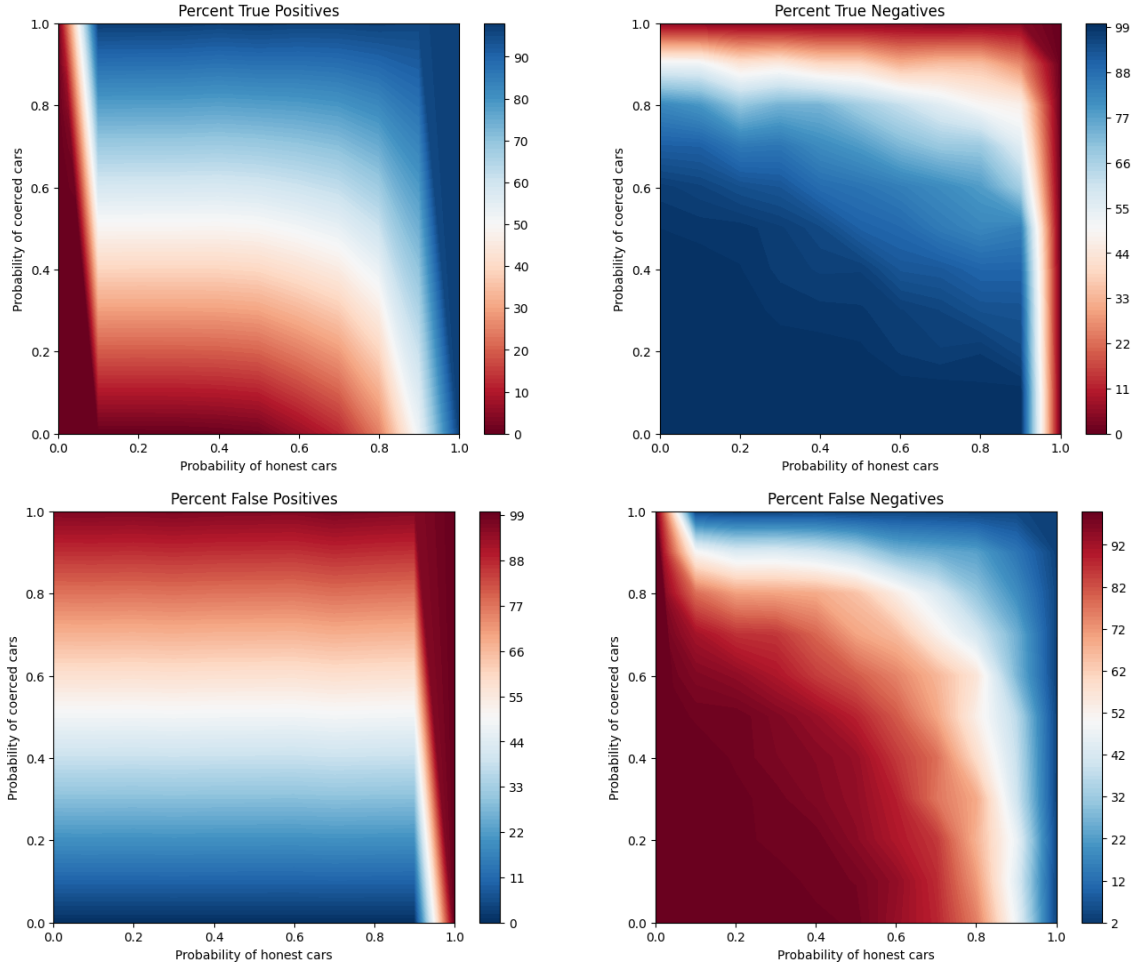


Figure 5.18: High density performance of TPoP with a tree of height 2 and branching factor 2, and a threshold = 100%.

5.7 Performance of TPoP

From the results obtained, it can be observed that the performance of TPoP does not vary greatly when using parameter $h = 1, w_1 = 6$ versus $h = 2, w_1 = 2, w_2 = 2$, especially when the density is high. The former slightly out-performs the latter in both thresholds tested. This is consistent with the results shown in Figure 5.8. The main difference in performance is observed when varying the threshold parameter, t , and when varying the density. Increasing density increases the number of True Positives, given that honest agents will not be classified as dishonest for not having sufficient nodes in their tree. Decreasing the threshold improves the rate of True Positives (reliability) of T-PoP in very low density scenarios, but this comes at the expense of security. A lower threshold makes it easier for dishonest agents to submit a valid tree. As such, users may tune the threshold parameters to obtain a desired level of security or reliability, depending on the expected density of their system.

We highlight that the algorithm does not rely on high agent density or a sufficient number of honest actors to operate. We propose certain operating conditions to illustrate the security and reliability guarantees the algorithm can offer in the context of dense, smart

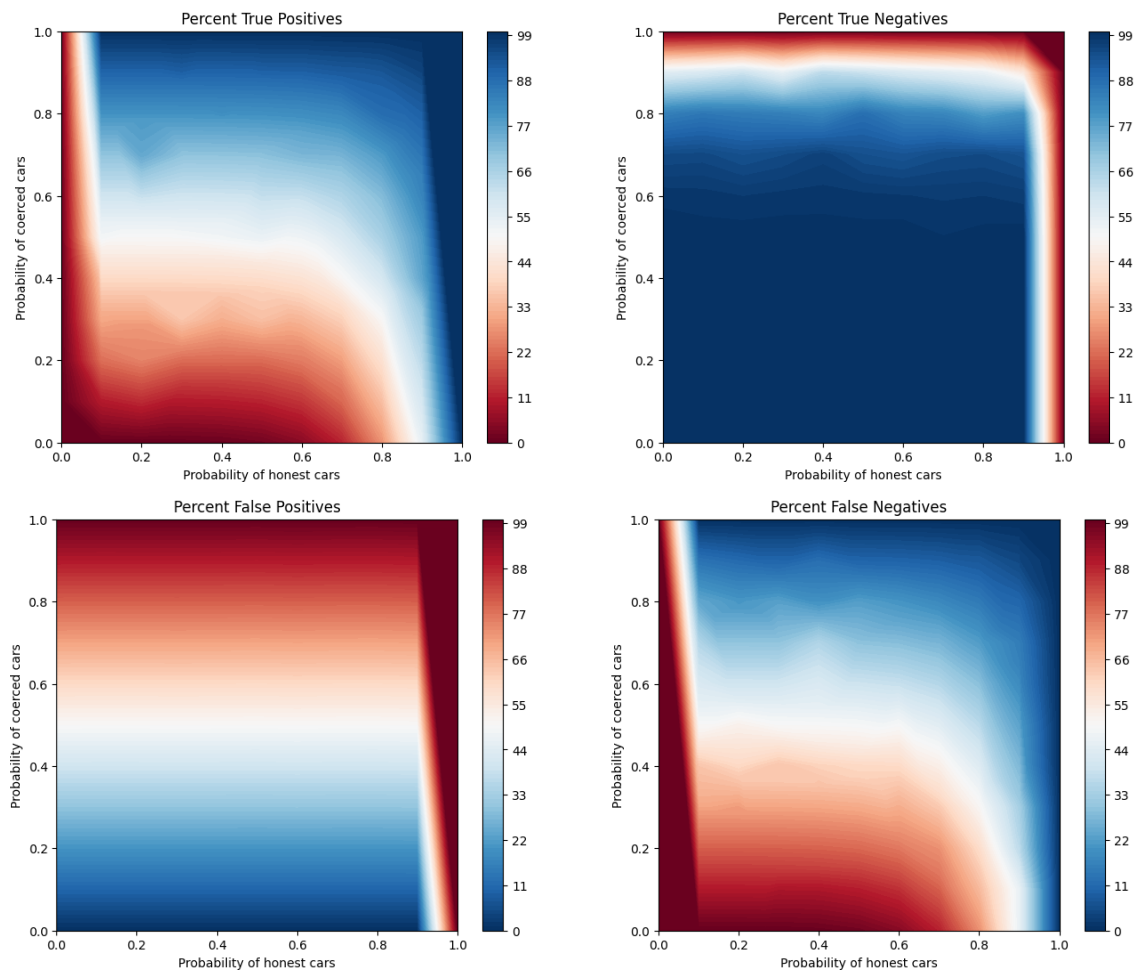


Figure 5.19: High density performance of TPOp with a tree of height 1 and branching factor 6, and a threshold = 100%.

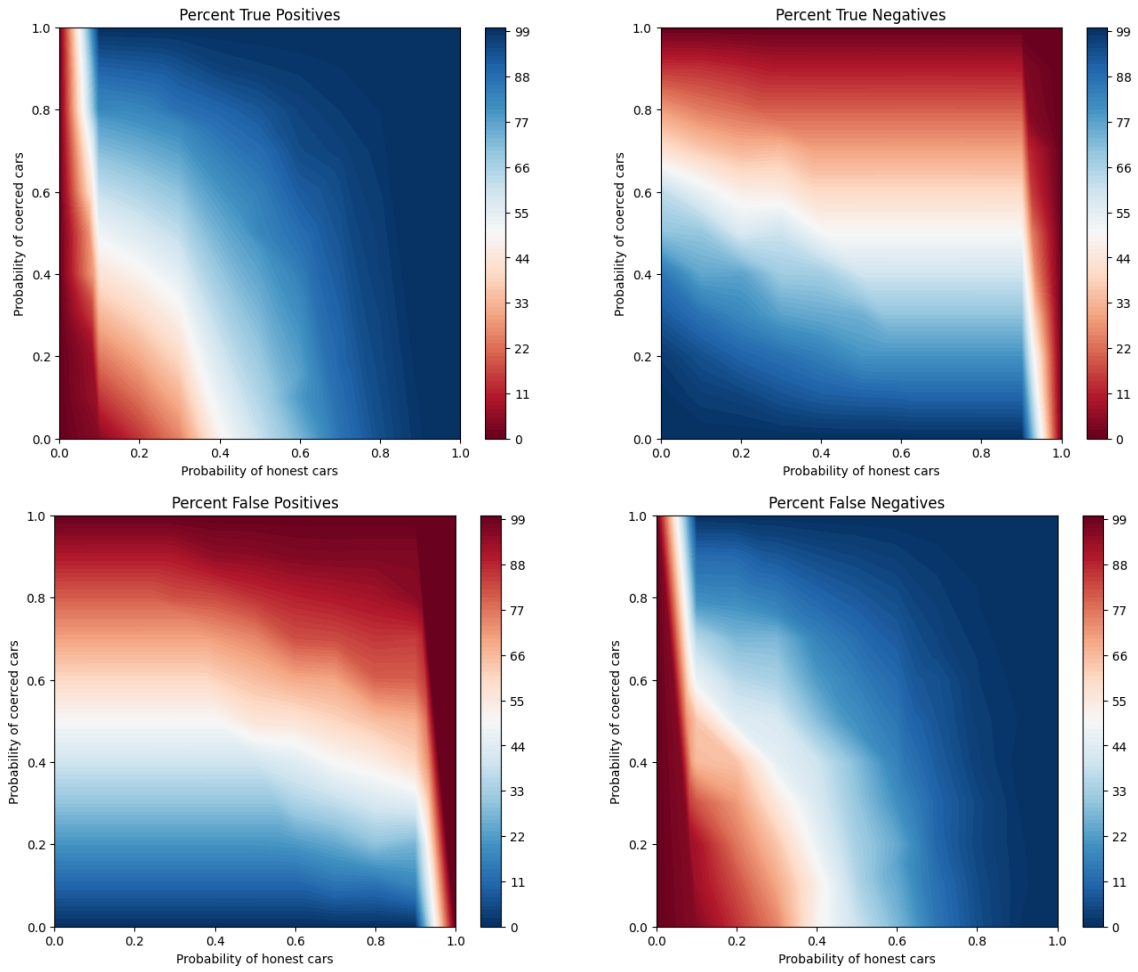


Figure 5.20: High density performance of TPoP with a tree of height 2 and branching factor 2, and a threshold = 40%.

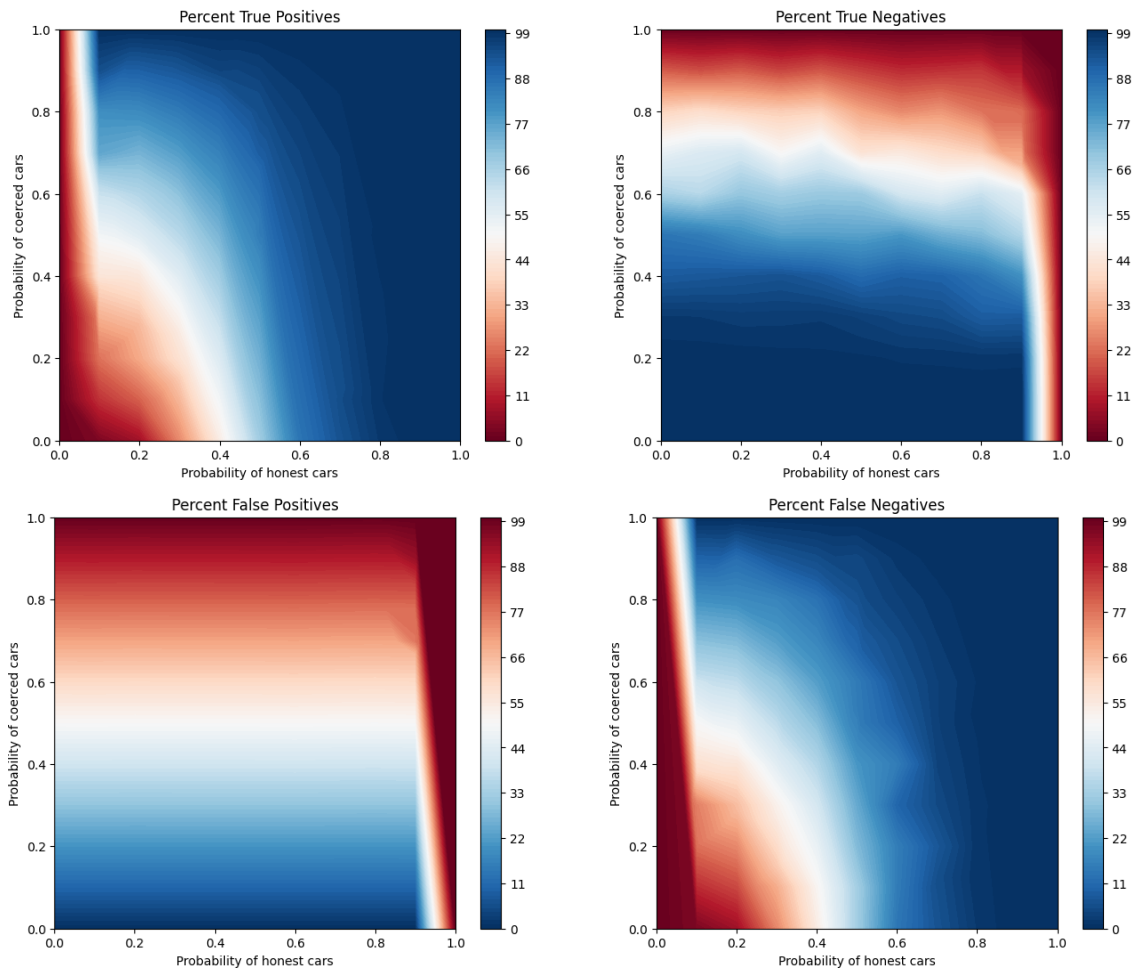


Figure 5.21: High density performance of TPOp with a tree of height 1 and branching factor 6, and a threshold = 40%.

cities. However, should the algorithm be used in a sparser, more adversarial scenario, these operational settings (the parameters of the tree size and threshold) can be tuned to achieved desired levels of performance. These parameters can be selected by using the mathematical model we propose, that has been empirically validated. We also test our algorithm on a low density scenario but retaining the same tree architecture used in the high density context, to show that even in worst case scenarios T-PoP still operates providing reasonable levels of performance.

5.8 Implementation Considerations

T-PoP’s practical implementations will vary depending on the application. It may be used in by vehicles to prove their position within a city, or it could be implemented in a smartphone, operated by individuals wishing to prove their location. It is also worth noting that T-PoP is not a highly resource intensive algorithm to execute (as shown in Section 5.4), thus, the hardware constraints are not a critical concern. Of significant importance are the identity management solution, the commitment scheme selected, the communication security protocol and the notarising of the position proofs. Subsequently, we proceed with providing some possible options, but we note that the final decision must be made considering the context specific needs: conflict-zones will have different security and privacy needs to smart cities.

Identity management is trivial using a centralised solution. T-PoP is a decentralised, peer to peer algorithm, so should a centralised trust assumption not be desirable, we suggest other alternatives: Proof of Humanity (PoH) is a decentralised project ‘*to provide a social identity verification system for humans on Ethereum*’ [400]. Users cannot submit duplicate identities, fake or deceased identities may be challenged and removed and identities are verified by other verified and registered users. The drawback of this mechanism is that the identity is (naturally) public. For reporting purposes this is not an issue: paired with a commitment scheme, the reporter’s position remains hidden, but the proof of humanity shows the reporter (prover) is a real and verified individual. PoH is built on the Ethereum blockchain, thus incurring transaction costs to the user wishing to obtain a proof of humanity, however, this is a one time cost, as users will not obtain more than one proof of humanity.

For privacy critical applications, we have shown that T-PoP may be computed under encryption, such that agents do not need to reveal their location publicly or even to each other. We showcase this with our proof-of-concept implementation using Zama’s fully homomorphic encryption library. This can be found in our [GitHub](#) repository. However, T-PoP does not rely on homomorphic encryption to operate, and including this implementation is simply to illustrate that anyone can compute T-PoP without actually revealing sensitive information, even to their witnesses or verifiers, other than the verification itself. We believe an interesting avenue of future research may be to bypass homomorphic encryption altogether and use T-PoP as a verification algorithm for non-interactive zero-knowledge proofs, or zk-SNARKs since these are strictly easier to verify than to prove.

These could be used for privacy critical situations instead of homomorphic encryption.

Regarding securing communication between agents in T-PoP, we advise using established protocols such as TLS 1.3. The protocol used should guarantee the following properties are met:

1. Confidentiality: Information is kept secret from all but authorized parties.
 2. Integrity: Messages have not been modified in transit.
 3. Message Authentication: The sender of the message is authentic.
 4. Non-repudiation: The sender of the message cannot deny the creation of the message.
- [391]

Finally, upon gaining a proof of position, this may be easily notarised in a DLT or the Inter-Planetary File System, along with the prover’s position commitment and prover’s tree, such that all other agents may verify that with the same inputs, they obtain the same T-PoP result. This must be accompanied with the prover and the witnesses’ unique signature to prevent forgery.

5.9 Limitations and Future Work

The computational complexity evaluation is currently theoretical and is an upper bound of the computational runtime expected. Future work should supplement this model with real-world data collection of the time complexity and communication overhead of TPoP on IoT devices. This is necessary to verify if TPoP’s real-world computational complexity matches our theoretical model. This real-world implementation is also necessary to verify the performance of TPoP under various operating conditions, and assess if our agent-based simulator can indeed emulate the behaviour of the algorithm. For future work, the aim is to implement TPoP on low-power IoT devices leveraging open-source protocols that enable mesh network designs. This is a requirement to allow for the construction of a tree of neighbours, since these have a mesh topology. Example protocols that satisfy these requirements are OpenThread [287] and Zigbee [180].

5.10 Conclusion

This chapter revisits and further develops the proposed T-PoP algorithm: an algorithm for agents to prove their position in a decentralised, collaborative and privacy-preserving manner. The analysis focuses on the robustness of the algorithm in an adversarial environment, showing that TPoP is robust to adversarial attacks. We provide a simple mathematical model to aid users in selecting the most suitable operating conditions of T-PoP, provided the security and reliability guarantees they wish to achieve, and the expected agent density in their system. The mathematical model is validated through the use of extensive Monte-Carlo agent-based simulations. The developed code-base is open-source, and the results along with the mathematical model and the agent-based simulator can be found

in the public [GitHub Repository](#). We also develop a framework to probabilistically detect platoons of malicious, collaborative agents, and we outline the algorithm’s computational complexity.

Chapter 6

Tree Consensus Algorithms

Abstract

This chapter generalises the Tree-Proof-of-Position algorithm, extending its application from proof of location statements to a broader consensus mechanism. It introduces a formal mathematical model of the algorithm, together with an agent-based simulator and a satisfiability solver implementation. The model is designed to guide users in selecting an appropriate tree architecture that achieves a desired balance between security and reliability.

6.1 Contributions Disclosure

I am solely responsible for all the work presented in this chapter.

6.2 Motivation

Consensus algorithms in decentralised protocols have been extensively studied, see for example [358], [115] and [308]. Most practical implementations adopt a consensus mechanism that allocates trustworthiness proportionally to wealth, or a proxy for wealth.¹ Whilst for many applications, this choice may be fit for purpose, we argue this is not the case for a data market implementation (Chapter 3). There is little indication to suggest that agents participating in a data market are more likely to be honest about their data source, veracity and quality if they are more wealthy. Furthermore, the design of our data market does not permit the purchase of shares of the market, or analogous stake, and therefore trust cannot meaningfully be allocated in proportion to financial investment.

In this work, the motivation is to explore alternative means to achieve consensus in a mutually distrusting environment, where information of other participants is limited, and

¹The wealth considered is either the one invested in the DLT application, or the one spent to acquire increased levels of computational power. The prior is akin to shareholder ownership models, where decision power or trustworthiness is granted to those who invest the most. Subsequent chapters explore the consequences of this choice to distribute power in the network.

reliance on centralised providers is bypassed. Faithful to the aim of enhancing fairness in decentralised applications, I advocate for trust being earned and not bought. After developing T-PoP to prove one’s position, it was of interest to extend the algorithm to any class of verifiable claims, and not just location ones only. In previous chapters, the user is able to obtain certain performance guarantees of the algorithm, given the algorithm’s operating conditions, the expected environment’s density and share of adversaries present.

Previously, it is noted that, although T-PoP is presented with the purpose of verifying location claims, the claims are not restricted to the physical nature, and can indeed be generalised. Hence, a natural extension arises to expand the use of T-PoP beyond location verification to generalised verifiable statements. This would enable using T-PoP as an algorithm to achieve consensus in a decentralised scenario. Thus, this chapter extends the logic and modelling framework of the proposed consensus algorithm beyond the scope considered previously. In the earlier formulation, agent behaviour was characterised by a set of relatively strict assumptions. Here, the behavioural model is broadened to account for more nuanced dynamics. For example, agents may act honestly under certain conditions but exhibit coerced behaviour under others. We no longer assume their behaviour is unchanging. Additionally, scenarios are considered in which an agent fails to respond altogether, neither endorsing nor rejecting a claim.

A further objective is to formalise the decision problem faced by prospective users of the algorithm: given a target level of security and reliability, which operating parameters should be selected? More specifically, this involves determining appropriate thresholds and tree dimensions. The model assumes that users possess an estimate of the expected agent density within the environment, as well as an upper bound on the proportion of adversarial agents. An associated question of interest is whether, under these constraints, a feasible configuration of tree parameters exists that can satisfy the desired security and reliability guarantees. This feasibility problem is naturally expressed as a satisfiability instance, which is addressed through the use of Satisfiability Modulo Theory (SMT) solvers, such as Z3.

6.3 Related Work

Consensus mechanisms are crucial in distributed ledgers to ensure agreement on the state of the ledger. For example, in the context of the branch of Computer Science known as *distributed systems*, consensus can be mapped to the fault-tolerant state-machine replication problem [416]. In such systems, the users in the network must come to an agreement as to what is the accepted state of the network. Furthermore, it is unknown which of these users are either faulty or malicious. This scenario is defined as a Byzantine environment, and the consensus mechanism used to address this issue must be Byzantine Fault Tolerant (BFT) [107].

In permissionless networks, probabilistic Byzantine consensus is achieved through the means of certain cryptographic primitives [527]. Commonly this is done by solving a

computationally expensive puzzle. This method, known as proof of work (PoW), although commonly used, suffers of a series of drawbacks, which have been described in [106], [293]. Aside from its poor scalability and mining races, this method incurs vast energy consumption costs [430]. In 2023, it was reported that Bitcoin consumed 121 tera-Watt hours that year², which is comparable to Norway’s annual consumption³ of energy that year.

In permissioned networks consensus is reached amongst a smaller subset of users in the network. This is done through BFT consensus mechanisms such as Practical BFT (PBFT) [108] and PAXOS [305]. Often the permissioned users are elected according to how much stake in the network they hold, following a proof of stake (PoS) method. This centralisation enables a higher throughput of transactions at the cost of higher messaging overhead, but ensures immediate consensus finality. They also require precise knowledge of the users’ membership. One such example is Ouroboros, a proof of stake protocol that divides time into epochs and slots, assigning slot leaders proportionally to stake for block production [285]. Its strengths lie in its formal security proofs, scalability, and energy efficiency, but it also faces challenges such as potential stake centralisation and concerns of fairness under dynamic network conditions. Meanwhile, in permissionless consensus protocols the guarantee of consensus is only probabilistic but does not require high node synchronicity or precise node memberships (ie: exact knowledge of which users are in the quorum), and are more robust [521] [527].

When considering consensus mechanisms for permissionless distributed ledgers, there exist a wide range of consensus mechanisms that are a hybrid combination of either PoS and PoW (eg: Snow White), or PoW-BFT (eg: PeerCensus) or PoS-BFT (eg: Tendermint). Each consensus mechanism places greater importance in achieving different properties. For example, Tendermint focuses on deterministic, secure consensus with accountability guarantees and fast throughput [100]. Snow White is a provably secure consensus mechanism that uses a reconfigurable PoS committee [74], and PeerCensus enables strong consistency in Bitcoin transactions, as opposed to eventual consistency [154].

There also exists a class of probabilistic consensus mechanisms, such as FPC [402], Optimal Algorithms for Byzantine Agreements [186], Randomised Byzantine Agreements [495] and Algorand [215]. The fact that they are probabilistic makes coercion of agents difficult for a malicious actor. To ensure malicious actors are selected by the consensus algorithm, they must know a priori the randomness used in the mechanism, or coerce a supra-majority of agents in the network. Furthermore, selecting users in a pseudo-random way can be argued to be a fairer method of selecting users, as opposed to choosing those with the greatest wealth or greatest computational power. Another consideration of fairness is made in [139], where the mechanism does not rely on a correct leader to terminate, therefore decentralising the mechanism.

In some examples described above, such as in [215], agents with greater wealth in the ledger are more likely to be selected by a Verifiable Random Function to form part of the

²Source: [Cambridge Centre for Alternative Finance](#).

³Source: [Ember Energy](#).

voting committee. This method retains voting power amongst the agents with the most stake, based on the assumption that the more stake an agent has in the system, the more incentive they have to behave honestly. The algorithm uses majority rule, and only has two possible outcomes: accept the transaction or not (timeout or temporary consensus) [215].⁴ Unlike [138], the work presented in this chapter allows participants to abstain, and can be extended to include as many preferences as desired, effectively providing a ranking system. This enables participants to add a score of trustworthiness to a prover's claim. It is also probabilistic, leveraging the advantages of this class of consensus protocols, and it is peer-to-peer, not relying on leadership elections or wealth to achieve finality.

6.4 Mathematical Model

We present a generalisable framework for how our algorithm, the Tree Consensus Algorithm (TCA), works. Previously, we restricted the state space of the possible combinations by enforcing certain logic on the behaviour of agents. In this work, we let go of these assumptions and relax the model in general. Instead of considering strict definitions of agent behaviour and conditionality rules, we simply consider that agents may choose to do one of the three: accept (a), reject (r) or ignore (i) another agent's claim. For ease of exposition we use the symbol R to denote the possible agent response, and C to denote the agent's possible claim. Therefore $R \in \{a, r, i\}$. We then consider that a claim may be true (t), false (f) or unverifiable (u). Thus $C \in \{t, f, u\}$. We assume child responses are i.i.d. across children and across parents. We enforce the following conditions:

$$\sum_{R \in \{a, r, i\}} \sum_{C \in \{t, f, u\}} p_{R,C} = 1 \quad (6.1)$$

$$\sum_{R \in \{a, r, i\}} p_R = 1 \quad (6.2)$$

$$\sum_{C \in \{t, f, u\}} p_C = 1 \quad (6.3)$$

These simply request that the agent performs one of the three actions, and that the claim has one of the three possible qualities. We assume that child responses are independent and identically distributed across siblings and across parents, and, for tractability, that responses are independent of the underlying claim. While in practice response behaviour may depend on claim veracity, the claim type is unobservable to us externally, and thus any conditional structure cannot be exploited by the algorithm. Independence therefore provides a principled baseline model: it captures the information actually available to the child verifying the parent, without presuming knowledge of the claim's truth.

⁴The disadvantages of majority rule systems such as First-Past-the-Post voting are known of and extensively documented [309], [84], [138]. A common critique of these voting systems is that they do not achieve proportional representation. Consequently, it could be argued that they are not an appropriate consensus mechanism for a context where the aim is to further decentralisation and fairness.

Response	Value	Probability
Accept	1	p_a
Reject	-1	p_r
Ignore	0	p_i

Table 6.1: Values assigned to each possible agent response.

Response \ Claim	True	False	Unverifiable
Accept	$p_a \cdot p_t$	$p_a \cdot p_f$	$p_a \cdot p_u$
Reject	$p_r \cdot p_t$	$p_r \cdot p_f$	$p_r \cdot p_u$
Ignore	$p_i \cdot p_t$	$p_i \cdot p_f$	$p_i \cdot p_u$

Table 6.2: Probability of each agent's possible response to varying claims.

For a prover to obtain a valid proof, the tree they construct must obtain a minimum number of approvals and have sufficient unique agents in the tree. In Chapters 4 and 5, the mathematical model measured the expected number of approvals, but did not consider rejections. To model this, we assign a weight to each possible response. These possible responses are summarised in Table 6.1.

The purpose of the mathematical model is to compute the probability that a certain tree will be considered truthful by the algorithm. Because TCA is a recursive, reverse breadth first search algorithm, we compute the probability of success recursively, starting from the lowest depth level. Let us begin with computing the probability that a single parent with a single child, at depth level d is deemed valid. A parent is deemed valid if it receives enough approvals, and we can compute the expected value for a single response from a child, X , as follows:

$$\mathbb{E}[X] = p_a(1) + p_r(-1) + p_i(0) = p_a - p_r \quad (6.4)$$

Where $X \in \{-1, 0, 1\}$. We then compute the expected value for the sum of the responses, S , that a parent in depth d , receives from its n_d many children.

$$S = \sum_{j=1}^{n_d} X_j \quad (6.5)$$

$$\mathbb{E}[S] = n_d \mathbb{E}[X] = n_d(p_a - p_r) \quad (6.6)$$

Recall that n_d is the number of children a parent in depth d has. The probability of a parent remaining in the tree, p_d is therefore:

$$p_d = P(S \geq n_d t) \quad (6.7)$$

To find p_d , we can compute this numerically for small values of n_d . For example, if $n_d = 2$, a parent can receive 3^2 different response combinations from its 2 children. Table 6.3 shows

Child 1 Response	Child 2 Response	Sum of responses	Probability of outcome
Accept	Accept	$1 + 1 = 2$	$p_a \cdot p_a$
Accept	Reject	$1 - 1 = 0$	$p_a \cdot p_r$
Accept	Ignore	$1 + 0 = 1$	$p_a \cdot p_i$
Reject	Reject	$-1 - 1 = -2$	$p_r \cdot p_r$
Reject	Accept	$-1 + 1 = 0$	$p_r \cdot p_a$
Reject	Ignore	$-1 + 0 = -1$	$p_r \cdot p_i$
Ignore	Accept	$0 + 1 = 1$	$p_i \cdot p_a$
Ignore	Reject	$0 - 1 = -1$	$p_i \cdot p_r$
Ignore	Ignore	$0 + 0 = 0$	$p_i \cdot p_i$

Table 6.3: Possible agent responses for a single parent with two children.

the possible permutations of responses available for a parent with two children. Then p_d can be evaluated following:

$$P(S \geq n_d t) = \sum_{k=\lceil n_d t \rceil}^{n_d} P(S = k) \quad (6.8)$$

In the example shown above, it can be computed numerically:

$$\Pr(S \geq n_d t) = \sum_{j \in \{-2, -1, 0, 1, 2\} : j \geq n_d t} \Pr(S = j),$$

where:

$$\begin{aligned} \Pr(S = 2) &= p_a^2, \\ \Pr(S = 1) &= 2p_a p_i, \\ \Pr(S = 0) &= 2p_a p_r + p_i^2, \\ \Pr(S = -1) &= 2p_r p_i, \\ \Pr(S = -2) &= p_r^2. \end{aligned}$$

Thus, explicitly,

$$\Pr(S \geq n_d t) = \begin{cases} 1, & n_d t \leq -2, \\ 1 - p_r^2, & -2 < n_d t \leq -1, \\ 1 - (p_r^2 + 2p_r p_i), & -1 < n_d t \leq 0, \\ p_a^2 + 2p_a p_i + p_i^2 + 2p_a p_r, & 0 < n_d t \leq 1, \\ p_a^2 + 2p_a p_i, & 1 < n_d t \leq 2, \\ p_a^2, & n_d t = 2. \end{cases}$$

However, generalising this behaviour is necessary, as the number of possible responses is 3^n , which rapidly grows as n increases. Therefore computing these values numerically is undesirable. Instead, we make use of a generating function to represent the probabilities of different response sums. These are often used to solve enumeration problems in combinatorics, and they work by encoding outcomes in the exponents and probabilities in the coefficients, turning a complex distribution into a tractable algebraic object. The generating function has a placeholder variable, x , where the exponent of x indicates the response value, and the coefficient of that term gives the probability that a parent receives that value. For a single child response to a parent, we define:

$$G(x) = p_a x^1 + p_r x^{-1} + p_i x^0 \quad (6.9)$$

Note each exponent is the value corresponding to the possible response value that a parent can receive: 1, -1, 0. Taking the derivative of $G(x)$ and substituting $x = 1$, we obtain the expected value of the response.

$$G'(x) = p_a - p_r x^{-2} \quad (6.10)$$

$$G'(1) = p_a - p_r(1)^{-2} = p_a - p_r = \mathbb{E}[X] \quad (6.11)$$

When there are n children, we can apply the same approach by simply taking the generating function and raising it to the power of n . In the expanded form of $G(x)^n$, each exponent of x corresponds to a possible total response sum of n children, and the coefficient of that term gives the probability of that outcome.

$$G(x)^n = (p_a x + p_r x^{-1} + p_i)^n \quad (6.12)$$

We choose the following notation to refer to the coefficient of the x^k term in the generating function $G(x)^n$:

$$[x^k]G(x)^n.$$

We can therefore define that:

$$[x^k]G(x)^n = P(S = k) \quad (6.13)$$

Meaning that, for n many children, the probability that a parent receives k approvals is the coefficient of the term x^k in $G(x)^n$. We denote the probability that a parent receives **enough** approvals as p_s , which we can compute by evaluating equation 6.14.

$$p_s = P(S \geq n_d t) = \sum_{[k=n_d t]}^{+n_d} [x^k]G(x)^{n_d} \quad (6.14)$$

We can therefore compute the expected value of total responses from n children, $\mathbb{E}[S]$, by taking the derivative of $G(x)^n$ and evaluating it at $x = 1$.

$$\left. \frac{d}{dx} G(x)^n \right|_{x=1} = \mathbb{E}[S]$$

Where

$$\frac{d}{dx} G(x)^n = n G(x)^{n-1} G'(x).$$

Evaluating at $x = 1$:

$$\mathbb{E}[S] = n G(1)^{n-1} G'(1).$$

Since $G(1) = p_a + p_r + p_i = 1$, this reduces to:

$$\mathbb{E}[S] = n(p_a - p_r).$$

Now, for a single depth level in the tree, we can compute the probability that enough **parents** will remain in the tree, $P(A_d)$. The number of parents in a depth level d is given by N_d .

$$N_d = \prod_{i=0}^{d-1} n_i, \quad N_0 = 1 \quad (6.15)$$

$$P(A_d) = \sum_{j=N_d t}^{N_d} \binom{N_d}{j} (p_s)^j (1 - p_s)^{N_d - j} \quad (6.16)$$

The final equation that models whether TCA will consider a prover's tree honest is

$$P(TCA = True) = \prod_{d=h-1}^0 P(A_d) \quad (6.17)$$

This is simply the product of enough parents being considered valid per depth level, across all depth levels. With this model, we can also compute the expected number of approvals in the entire tree, as shown in equation 6.18.

$$\mathbb{E}[A] = \sum_{d=0}^{h-1} N_d \cdot n_d \cdot (p_a - p_r), \quad N_0 = 1 \quad (6.18)$$

Equation 6.18 can also be restated as:

$$\mathbb{E}[A] = \sum_{d=0}^{h-1} N_d \cdot n_d \cdot \mathbb{E}[X] \quad (6.19)$$

recalling that $\mathbb{E}[X]$ is the expected value for a single agent's response.

6.5 Parameter Selection of TCA

Now we wish to provide users with a means to select a tree dimension to meet their reliability and security needs. We can frame this problem as an optimisation one, where the user provides the values of choice for certain constraints. Precisely, they are tasked with determining the maximum number of total agents allowed in the tree, M , and the maximum height, H . These two constraints are related to hardware capabilities. In Chapter 5 Section 5.4 it is shown that for tree architectures with the same number of nodes, the chosen height can affect the communication overhead between nodes. Finally, the user must provide a desired level of security and reliability. Previous definitions for these measures are given in equations 4.7.1 and 4.7.2 respectively. Another possible way to define them is

$$\text{Security} \equiv \Pr[p_a p_t \geq S]$$

$$\text{Reliability} \equiv \Pr[p_r p_f \geq R]$$

where S and R are user chosen values and $S, R \in [0, 1]$. Unfortunately, practically being able to measure p_t, p_f, p_i (the probability of a claim being true, false or unverifiable) is not realistic for a user. Indeed, if they could do this, there is no need for TCA. However, it should be possible to estimate a probability mass function for the possible outcomes in Table 6.2. This is akin to estimating the share of honest and of adversarial agents present in their application. With this estimate, it is sufficient to retrieve values for $p_a p_t$ and $p_r p_f$. It is therefore possible to reformulate this optimisation as a feasibility problem, where the objective is to determine whether there exists a solution that satisfies all user-specified constraints. If such a solution exists, an Satisfiability Modulo Theory (SMT) solver such as Z3 can be used to decide satisfiability and return a concrete assignment of variables [153]. We provide a preliminary implementation in our [GitHub repository](#), that also contains the

source code for all provided simulation results.

$$\max_{\theta} \quad \lambda \Pr(\text{TCA} = \text{False} \mid p_f = 1, \theta) + (1 - \lambda) \Pr(\text{TCA} = \text{True} \mid p_t = 1, \theta) \quad (6.20a)$$

$$\text{s.t.} \quad p_a p_t \geq R \quad (6.20b)$$

$$p_r p_f \geq S \quad (6.20c)$$

$$h \leq H \quad (6.20d)$$

$$\sum_{i=1}^j \prod_{j=0}^{h-1} n_j + 1 \leq M \quad (6.20e)$$

$$\theta = \{h, (n_1, \dots, n_{h-1}), t\} \quad (6.20f)$$

6.6 Evaluation

Figure 6.1 shows the probability of the Tree Consensus Algorithm accepting a given tree, for various tree dimensions. For simplicity, we consider only regular trees, where the branching factor is constant across height levels. The results are obtained from the mathematical model in equation 6.17. Similarly, Figure 6.2 shows the results of equation 6.18, the expected number of approvals in a tree, for varying tree architectures. For a given graph G , with E edges and V vertices, the total number of edges is $E + 1 = V$. The maximum value of $\mathbb{E}[A]$ is therefore E , and the minimum number is $-E$. This is consistent with the results shown in Figure 6.2

Figure 6.1 shows that the probability of an agent ignoring a claim, p_i , does not affect TCA's probability of success. This follows from the mathematical model constructed, as p_i has no influence TCA's classification from equation 6.17. It also shows that as the height h and branching factor n increase, the probability of a tree being successful decreases drastically. Although the expected total number of approvals in the tree $\mathbb{E}[A]$ grows exponentially with the tree height (for $n > 1$), the number of witnesses, W , also grows at the same exponential rate. Consequently, the relevant quantity is the normalized expectation

$$\frac{\mathbb{E}[A]}{W} = \mathbb{E}[X],$$

which is independent of h . By the law of large numbers, the observed approval to witnesses fraction A/W concentrates sharply around $\mathbb{E}[X]$ as both h and n increase. Thus, unless $\mathbb{E}[X]$ exceeds the acceptance threshold fraction, the probability of success converges rapidly to zero as the tree dimensions grow. Only when $\mathbb{E}[X]$ is close to 1 is the approval fraction sufficiently high to ensure success, and this situation arises precisely when p_a (the acceptance probability) is maximal. This explains why the feasible region of acceptance shrinks toward the apex of the ternary plots.

From Figure 6.1 what we can see is that increasing depth levels in the tree makes it increasingly difficult for a tree to be deemed honest by the algorithm. This can be observed by how the purple area increases in the ternary plots as the number of depth levels and

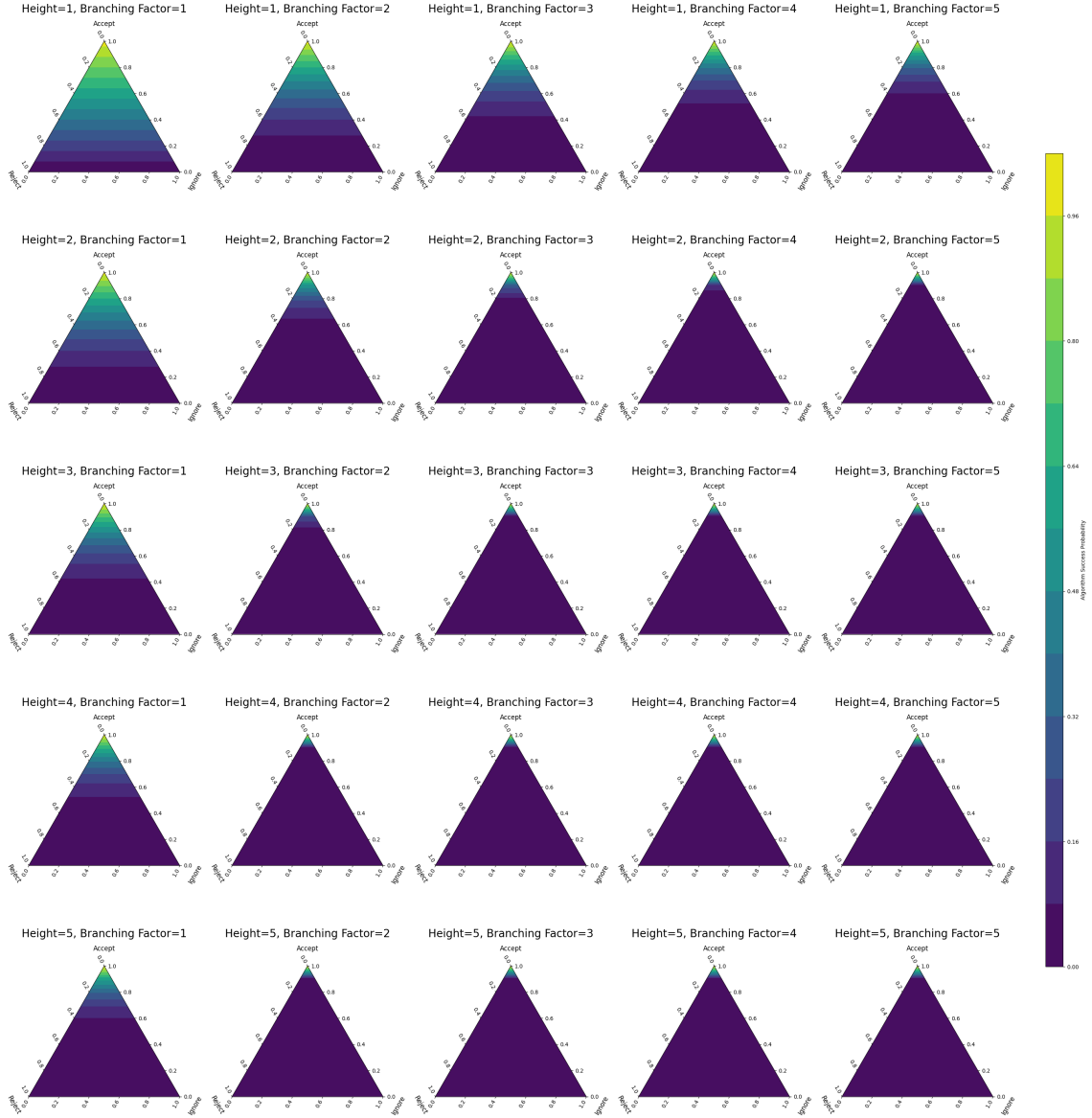


Figure 6.1: Probability of the Tree Consensus Algorithm accepting a tree, for various tree dimensions. The branching factor is kept constant across depth levels. Convention follows that apex means that the probability of accepting a claim, p_a , is 1, and the probability of rejecting, p_r , and ignoring the claim, p_i , is 0. The left corner vertex indicates $p_r = 1$, and right corner vertex means $p_i = 1$.

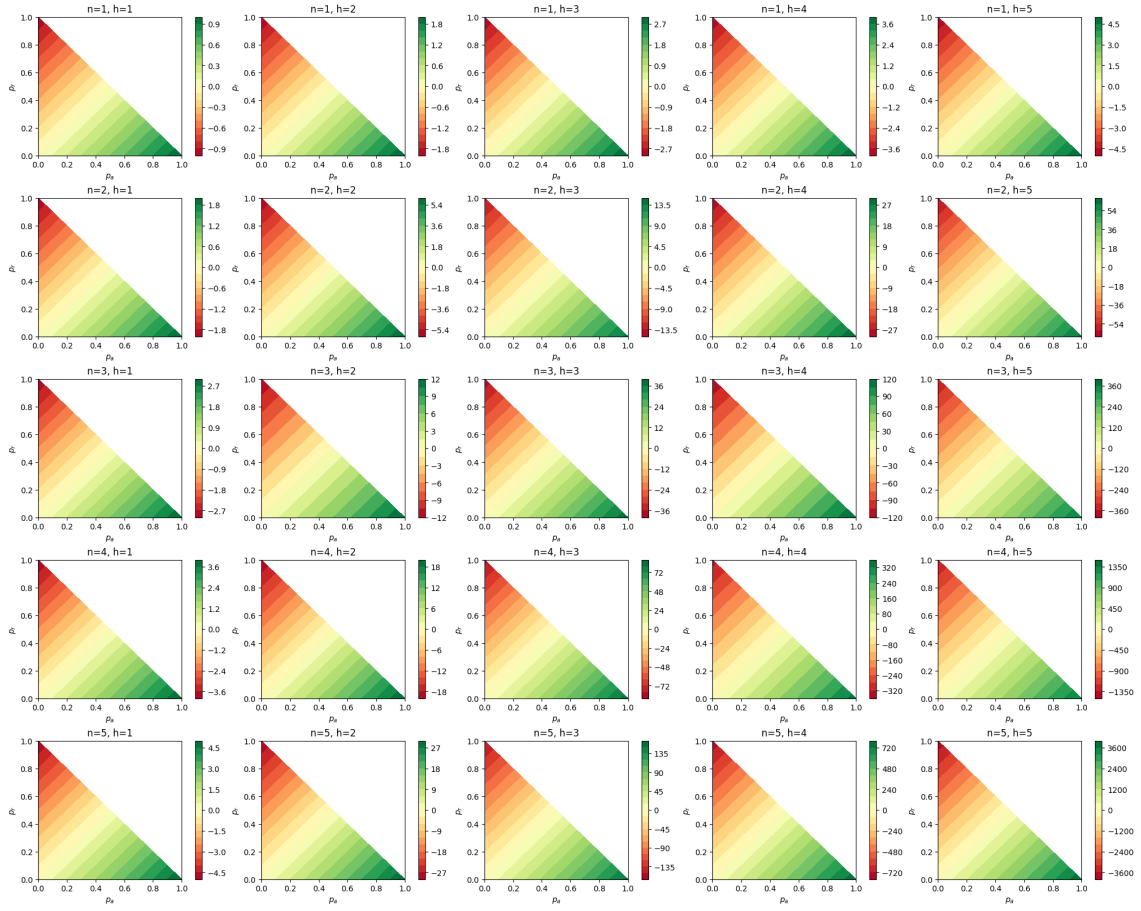


Figure 6.2: Expected number of approvals for a given tree, for various heights and branching factors. The branching factor is kept constant across depth levels.

branching factor increases. The probability of success is condensed in the top corner of the ternary plot, meaning trees are successful as the probability of agents being honest tends to 1. Both Figures 6.1 and 6.2 combined show that what is important for a tree to be successful in the algorithm, is not the absolute number of agents in the tree, whether this be by increasing the branching factor or depth levels. Rather, what determines the success of the tree is the proportion of honest agents to the number of agents in the tree. As this proportion tends to 1, so does the success probability of any given tree.

Chapter 7

Distributed Ledger Technologies: Challenges in Governance

Abstract

Distributed Ledger Technologies (DLTs) promise decentralisation, transparency, and security, yet the reality often falls short due to fundamental governance flaws. Poorly designed governance frameworks leave these systems vulnerable to coercion, vote-buying, centralisation of power, and malicious protocol exploits—threats that undermine the very principles of fairness and equity these technologies seek to uphold. This chapter surveys the state of DLT governance, identifies critical vulnerabilities, and highlights the absence of universally accepted best practices for good governance. By bridging insights from cryptography, social choice theory, and e-voting systems, we not only present a comprehensive taxonomy of governance properties essential for safeguarding DLTs but also point to technical solutions that can deliver these properties in practice. This work underscores the urgent need for robust, transparent, and enforceable governance mechanisms. Ensuring good governance is not merely a technical necessity but a societal imperative to protect the public interest, maintain trust, and realise the transformative potential of DLTs for social good.

7.1 Contributions Disclosure

I am responsible for the entirety of the work presented in this chapter, with guidance from William Sanders.

7.2 Motivation

Thus far, previous chapters have explored applications and implementations leveraging and built on Distributed Ledger Technologies. The focus has been to enable and further the goal of achieving decentralisation and delivering fair solutions, in contexts where these are yet in their infancy. However, any solution built on DLTs will only ever be as decentralised and fair as the DLT itself. The process of governance is what concerns decision-making

and distribution of power in DLTs, and thus is inextricably linked to delivering fairness, decentralisation and transparency. A nascent body of work has already been pointing at the concerns of DLTs governance practices, but we found no unified approach to categorising these, and little practical guidance to solve these issues present. It is these research gaps that fuel the motivation to explore and address governance related vulnerabilities of DLTs.

DLTs have enabled a paradigm shift in how users interact and transact online. They are comprised of users, that interact with the DLT network and receive some service from it, and service providers (miners and node operators) that maintain the DLT network by running its protocol and providing service to its users [424]. DLTs do not exist in a single server, but rather run on a web of connected nodes, operated by the service providers, which communicate following a protocol [484]. A protocol is simply a set of instructions that nodes execute to enable the functioning of the DLT. DLTs are run through code, and maintainable code requires updates [323]. They must have a mechanism through which protocols are updated, bugs are patched and the service is improved. This component is crucial to ensure DLTs adapt, develop and improve in a sustainable manner, as well as providing a much needed protection against threats [4]. Without this ability to mutate, vulnerabilities cannot be patched. Since all nodes must agree to use the same protocol, a DLT must rely on a consensus algorithm to decide which protocol updates will take place and when. Governance is an all-encompassing umbrella term that captures this ability: the decision-making process to change a DLT protocol [313].

Addressing governance pitfalls is critical. Cryptocurrencies have not only experienced unprecedented levels of growth¹, but are also facing institutional adoption². Although Satoshi Nakamoto conceived Bitcoin as a decentralised, peer-to-peer network, the reality is that it and many other cryptocurrencies are centralised [63] [426], manipulable [159] [52] and can be undermined due to their governance protocols [145], [267]. Governance is critical to delivering the promise of decentralisation, and without good governance, the vulnerabilities that may arise are dire. These issues must be addressed to preserve the original purpose of using cryptocurrencies for social good. Cryptocurrencies have become a matter of public interest: they are used by governments, their combined market worth is of \$3.71 trillion [191] and the entry barrier to invest in them is very low, meaning many with low technical and financial literacy can and do participate and invest in these technologies [54]. Whilst this is revolutionary, it is also highly dangerous: many victims have lost life's savings due to scams, hacks, heists and rug pulls with little to no compensation [474], with a number of such examples being described in Appendix B. Governance safeguards are necessary to ensure transparency, accountability and prevent monopolisation and theft of funds.

This work addresses the governance dimension of DLTs that is increasingly intersecting with

¹At time of writing (November 2024), Bitcoin price is up by 24,801.30% since it was created. Similarly, Ether increased by 1,985.73%, Cardano by 929.34% and Solana by 37,300% [191].

²Examples include countries regulating these assets such as the US [249], Japan [29] and the EU [551]), but also governments owning and investing in them as well [359] (El Salvador [204] and Brazil [417] actively investing, and other countries holding seized assets such as the US and UK [311]).

public policy, regulation, and societal well-being. Cryptocurrencies and distributed ledger technologies (DLTs) are no longer niche financial tools; they are evolving into systems that challenge traditional governance models and require new frameworks for oversight, trust-building, and public protection. By identifying vulnerabilities and proposing governance best practices, this work contributes to the interdisciplinary dialogue on how digital technologies transform governance structures, impact public institutions, and necessitate regulatory adaptations. Ultimately, the hope is for this research to inform policy-making, aid regulators in understanding systemic risks, and help ensure that these technological innovations serve, rather than exploit, the public interest.

7.3 Related Work

A comprehensive systematisation of knowledge on good governance properties can be found in Kiayias and Lazos [284] and Lie et al. [324]. The authors in [284] focus on governance properties, whereas [324] answer questions on what governance is, who is involved, when is it used and why. The work in [284] serves as a starting place for the work we present. Namely, we find their taxonomy of governance properties the most clear, tangible and concise thus far. However, they explicitly note that they do not provide specific techniques, algorithms or mechanisms to achieve good governance properties.

The authors in [275] consider blockchain governance in the context of IT governance, and study different decisional outcomes using the Nash equilibrium. Other research efforts such as [397], [58] and [152] aim to define governance and its components, although none delve deeply into tangible definitions of good governance properties.

In terms of vulnerabilities that may arise as a consequence of poor governance in DLTs, [437] claims that economic incentives alone do not suffice to enable good governance, and makes a case that well studied social choice and politics findings should be applied to governance in DLTs. The work in [72] outlines a number of challenges regarding DLT governance. Namely, it notes how poor governance models can cause oligarchies in DLTs, how incentives are crucial in ensuring accountability and notes how decision rights should be clearly defined in the governance protocol. The authors in [151] identify that governance in Bitcoin is not as decentralised and code-driven as expected, and uncovers that in fact the decision-making power is highly technocratic and determined by a small minority of code developers. None of the aforementioned address the formal security and privacy notions necessary in the governance protocol, or the vulnerabilities that arise in the absence of these. We find a substantial overlap in the study of governance vulnerabilities in Decentralised Autonomous Organisations (DAOs) and DLTs. The work in [36] and [146] outline how vote buying cartels may be automated as a consequence of poor governance models in DAOs. The authors in [184] also highlight a number of governance related vulnerabilities in DAOs. Specifically regarding purchaseable voting power, malicious proposals and flash loans exploits, which the work in [278] also discusses. A noteworthy body of research is carried out in [488], where the authors note the importance of classical security notions for e-voting should also hold in DAO governance mechanisms. We agree. However, they do not

provide guidance on how to achieve these primitives. We find it necessary to provide formal notions of good governance, because without these, the properties remain qualitative. It is challenging to discern whether good governance is in place, if no technical definition exists to benchmark against. In the absence of these, projects can claim to deliver desirable properties with no formal proof of such claims. The subsequent chapter explores the real world damages and negative impacts this can have on its participants.

In summary, some recent works have emerged outlining vulnerabilities that arise as a consequence of poor governance models in DAOs, but less so regarding DLT governance. Some cautionary warnings have been raised regarding DLT governance, but we find no work that explores the suite of possible attacks that may arise in DLTs in the absence of standard good governance properties.

7.4 Methodology

We employ a qualitative, analytical approach to identify vulnerabilities in Distributed Ledger Technologies arising from deficiencies in governance models.

Identification of Vulnerabilities

From work previously carried out regarding DAO vulnerabilities in [278], we identified their governance protocols enable a number of weaknesses as a consequence of flaws in their governance protocols. Other researchers corroborate these findings: [184], [36], [488]. We hypothesize that these vulnerabilities are also applicable to DLTs, since DAOs are implementations built on DLTs. We then proceeded to survey existing work following two criteria: governance related vulnerabilities, and governance properties of DLTs.

Identification of Governance Properties and Vulnerabilities

A comprehensive review of existing literature and frameworks on governance in DLTs was conducted to identify key governance vulnerabilities and properties. This was done following the PRISMA methodology [361]. We searched 4 databases using the keywords: *DLT*, *governance*, *vulnerabilities*, *attacks*, *properties*. The outcome of this search is summarised in Figure 7.1. The number of works identified was limited, we therefore widened our search to include DAO governance, of which only the work of [488], [184] and [146] were identified as relevant. Of the surveyed studies, only [284] focuses on governance *properties* of DLTs, and [218] on *vulnerabilities* of DLTs, although it does not explicitly discuss which of these vulnerabilities arise as a consequence of poor governance. The authors of [488] point to some formal notions of good governance in the context of DAOs, and implementations for governance that attempt to deliver these. These, in conjunction with [284] serve as the starting point for our work regarding our contributions on governance properties, and [146] and [218] for our contributions on governance related vulnerabilities.

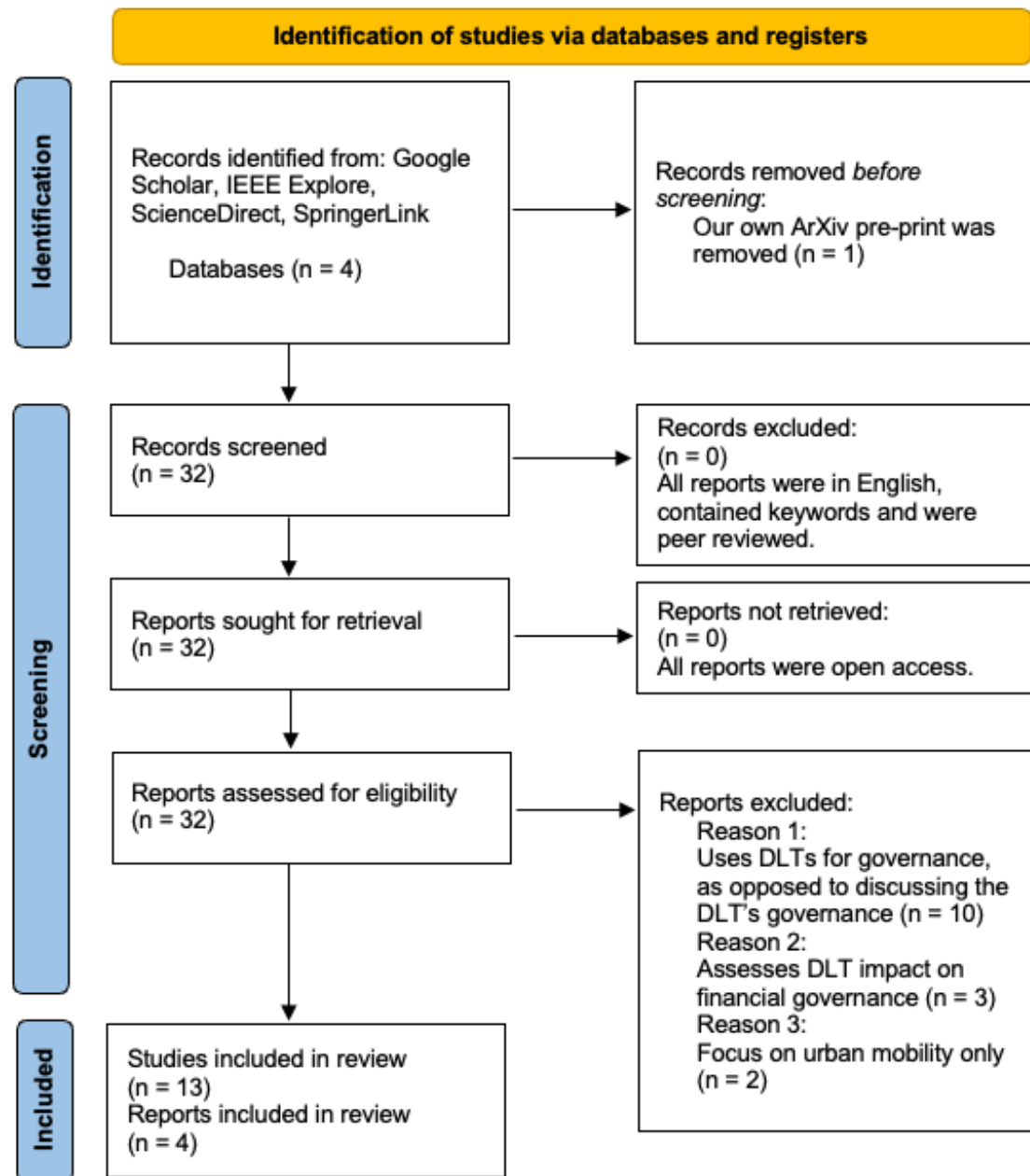


Figure 7.1: PRISMA flow diagram for systematic review of literature conducted.

Assessment of Governance Gaps

Using the governance properties as a benchmark, an analysis of governance models in various DLT systems was performed. This step involved examining case studies, technical documentation, and academic studies to identify instances where these governance properties were insufficiently implemented or entirely absent.

Linking Vulnerabilities to Absent Governance Properties

The identified governance failings were analysed to determine their impact on DLTs. Logical reasoning and case-based evidence were used to connect missing governance properties to specific vulnerabilities, such as vote buying, coercion, centralisation of power, incentives to heist the DLT and non-compliance with election outcomes.

Theoretical Justification

We then leveraged existing theories and frameworks in organisational governance, distributed systems, e-voting, cryptography and social choice theory to substantiate the connections between governance deficiencies and the emergence of vulnerabilities.

7.5 Governance Components

In this work, we refer to *governance* as the decision-making processes to enable the functioning and updating of a DLT protocol. The authors in [548] similarly state that the term governance refers to ‘the frameworks and processes that oversee decision-making, control, and the evolution of the system. This encompasses how rules are established, enforced, and modified within the decentralised network’. The work in [313] state that governance in DLTs is the ‘rule-making by the owners or participants of a system with the purpose of safeguarding their private interests’. Other approaches examine the intersection between blockchain governance and corporate governance [545] and borrow from the definitions of corporate governance [149]. For example, the work in [545] states: ‘Ultimately blockchains must rely on a governance process in which the users agree upon a set of requirements for the underlying software code to be changed, including provisions for dispute resolution, sanctions for violating the agreed upon rules, and procedures for enforcement of penalties’. Whilst there is no consensus on what is the definition of governance in the context of DLTs, most definitions overlap with the following critical functionalities: consensus, rules and decision-making.

Before proceeding with the outlined vulnerabilities, we establish definitions and nomenclature to aid with the explanation of governance related attacks. We present what we consider to be the minimum viable components to enable decision-making on-chain. These are borrowed from electronic voting literature, that also establish similar roles [8], [117], [127], [425].

- The voters.

- The votes.
- The vote tallying algorithm.
- The election officials: these are participants in charge of tallying the votes, maintaining the voting platform and publishing results.
- The bulletin board: the digital record of cast ballots.
- Stakeholders: they may or may not be allowed to vote, but they are in some way affected by the outcome of an election. This includes participants in the DLT that are in charge of implementing the election outcome.
- The platform to vote: the server, website, chain or other online location where votes are cast.

7.6 Vulnerabilities

In this section we outline the existing vulnerabilities that arise as a consequence of poor governance models. We highlight the intersection between DAO and DLT governance: both share similarities in their practical implementations. Indeed, this follows from the fact that many DAO projects are built on DLTs.

7.6.1 Exploits leveraging public votes

The right to a secret vote is recognized as a human right as per Article 21.3 of the Universal Declaration of Human Rights [506] in 1948. In the absence of it, voters may be coerced, bribed or sell their votes. Secret ballots are congruent to freedom, they ensure voters can express their will free of external influences and pressures. This is no different in DLT governance.

Voter Coercion

When ballots are public, everyone can know how a voter voted. This provides verifiability: everyone can check that the election outcome is correct and that only those allowed to vote voted. However, this forgoes freedom. Free will cannot be protected when a voter faces the reality of not abiding with their coercer's threats [316]. The coercer can easily check if their victim complied. Voters may chose to sell their votes even in the absence of coercion. In some cases, DLTs provide a unique set of conditions where the most beneficial strategy for some voters is to sell their vote. Other works have identified this issue in DAO governance [146], [36]. We highlight that this problem is prevalent in DLT governance too.

Currently, most DLTs have informal channels of discussion where participants debate on whether the proposal will be implemented, and from what time onwards it will be adopted if it has received enough support. These channels range from Slack, GitHub or Discord, to dedicated voting portals such as Snapshot and Tally, and often involve 'rough consensus'.

We stress, at current time of writing, none of these solutions offer secret ballots. No existing solution can be shown to satisfy formal notions of ballot secrecy, because votes are publicly posted on the ledger. Some have attempted to provide increased levels of privacy, but this privacy is short lived, and often compromised to achieve verifiability. An example is Snapshot's recently added feature of 'shielded voting', wherein votes remain private only until the end of the election. Then, 'when voting ends, all votes are revealed - as well as who voted for what' .³ This still enables voter coercion and vote selling.

Disincentive to vote against whales

Both users and miners vote with their feet on a DLT. When a proposal arises for a protocol update, both must agree for it to be implemented. Users may favour a certain proposal, but miners can refuse to service transactions following it. Similarly, if miners favour a proposal that users disagree with, the users may simply stop sending transactions on that chain, and miners will earn no fees. Let us consider a voter with much larger voting power than the rest a *whale*. An example could be a user with a large share of tokens or a collective with high mining power (a large mining pool) in the DLT. When a whale votes for a certain proposal, other agents are incentivised to follow suit. Otherwise a fork⁴ may arise, and eventually one of the two will outgrow the other and become the main chain. Miners processing transactions on the losing fork will not receive their reward. Consequently users' transactions will not get approved, and thus they too will eventually leave the fork. Because whales hold greater voting power, the fork they vote for will be the winning one with greater likelihood. Less influential voters will minimise their chance of losses by following the behaviour of whales, which is public, on-chain. This creates a herding effect, and a disincentive to vote true to one's beliefs, if these oppose those of a whale's.

For a voter in this situation, their only non-losing strategy is to abstain, but they are guaranteed a reward when they sell their vote. Given that votes are public, this sale can be verifiable: the vote buyer can check that they complied. Indeed, [36] and [146] show how vote buying can be automated through a DAO cartel. Their prototype for vote buying code is designed to buy DAO votes, although there is no impediment to using it to buy DLT votes.

Voter Information Asymmetry

When ballots are public, the first voter casting a vote has no information on what the election outcome will be, whereas the last voter to cast their vote knows the election outcome. Voters are unequal: the later you vote the more information you obtain regarding the election result. This allows more powerful voters to withhold their vote until they can guarantee that it is sufficient to sway the election result in their favour.

³Snapshot adds 'Shielded Voting'.

⁴A fork is a divergence in a DLT regarding the accepted version of transactions [526].

Another simpler exploit consists of an attacker purchasing enough voting power to approve a malicious proposal in their favour. Public voting facilitates this attack: the attacker knows exactly how much voting power they need to acquire to overturn an election in their favour. These attacks have been witnessed in DAOs a number of times ⁵, ⁶, ⁷, and are plausible in DLTs too. As shown in [525] and [210], miners' voting power (hash rate distribution) is highly centralised. In some cases, the voting power held by one agent can suffice to fork a network. In others, malicious users may leverage flash loans⁸ to rapidly obtain sufficient tokens, submit a proposal to their benefit, cash out and repay the loan, whilst still making a profit. Miners are incentivised to service this proposal: they stand to gain high transaction fees. Indeed, the behaviour of whales is known to affect the price of a token [341]: they have the power to send large scale transactions that may incentivize miners to act against the consensus protocol [319], [145].

7.6.2 Exploit leveraging non-binding elections

In a DLT context, we consider a binding election one in which the outcome of the election is enforced into the DLT. We note that, although informal voting channels may exist, these are in fact nothing other than a polling mechanism, because the only binding votes on a DLT protocol update are the transactions sent by users and approved by miners on-chain, following either one protocol or another. All other off-chain so-called voting platforms are operating under the assumption that miners and users will behave on-chain in the same way that they expressed they would off-chain. The New York Agreement is a clear example of how off-chain voting is not binding and how this assumption may not always hold [251].⁹ Anyone can vote, but if the agents responsible for implementing the election outcome disagree with the outcome, they have no obligation to implement it, even if the election was fair, free and correct. This is a vulnerability, as stakeholders may subvert an election outcome by refusing to comply. To prevent this, DLT governance protocols must be binding. To ensure this, elections must be carried out on-chain. On-chain voting brings a suite of challenges with it, which we outline in section 7.6.3.

⁵Build DAO's hostile governance takeover attack, Feb 2022.

⁶Beanstalk cryptocurrency project robbed after hacker votes to send themselves \$182 million.

⁷Sanctioned Tornado Cash DAO governance heisted by hacker.

⁸Loans written in smart contracts that enable participants to quickly borrow funds without the need for collateral [111].

⁹Bitcoin's New York Agreement was a planned upgrade of Bitcoin to increase the block size from 1MB to 2MB. The update garnered significant community support but large Bitcoin mining pools refused to implement it [236]. This update would have lowered miner's transaction fees by improving transaction efficiency. The result was that Bitcoin was forked: Bitcoin maintained the same block size and Bitcoin Cash, the fork, adopted the increased block size [530].

7.6.3 Exploits leveraging purchasable voting power

Plutocracy

1-dollar-1-vote mechanisms and any variant of these where voting power may be purchased will converge towards granting power to the most wealthy. Any governance mechanism that allows users to purchase voting power will inherently converge to plutocracy and centralise over time [419]. Whilst this issue is well known, DLTs continue to implement token based voting, for lack of a better alternative for a decentralised Sybil-protection¹⁰ system. DLT practitioners must compromise and choose which of the alternatives suits their application needs best: permissionless voting does not rely on a central trusted service to prevent Sybil attacks, and is often implemented by providing voting rights proportional to wealth or computational power. However, it is known to centralise voting power in the hands of the wealthy. Permissioned voting introduces reliance on a centralised trusted authority to issue voting rights, or requires a practical and functional decentralised identity management solution. This permissioned alternative is often implemented through reputation based voting schemes, soulbound tokens, NFTs, or a Know Your Customer solution, checking the validity of participants' identity.

On-Chain Voting

One of many concerns of on-chain voting is the transaction fees associated with a vote. No DLT currently offers zero transaction fees, thus, when miners or users wish to vote, they must pay to do so. This further contributes to exacerbate inequalities as only the wealthier can afford to do so. These fees often soar unpredictably [261], and may also discriminate between voters as an election is occurring. The authors in [395] have extensively documented many of the concerns of blockchain voting. Another concern pertaining to on-chain voting is the fact that censorship resistance cannot be assured [522]: miners may simply refuse to approve blocks (containing votes) of certain users. Conversely, malicious agents can spam the network with empty blocks, thus delaying the approval of the user's transactions. The feasibility of the latter highly depends on the access control mechanism of the DLT and the congestion control mechanism. Finally, current on-chain voting solutions do not offer ballot secrecy guarantees. The consequences arising from this are outlined in section 7.6.1.

It can therefore be observed that another compromise arises: binding elections are offered by on-chain voting, however, this means censorship resistance is not guaranteed, and participants must pay to vote.

7.6.4 Other vulnerabilities

There are other security concerns that arise when a combination of desirable properties are absent in the governance solution of the DLT.

¹⁰Sybil-protection mechanisms are those that prevent an attacker from creating multiple fake identities to gain unfair advantages in a network [360].

Bribe Automation

This exploit is a consequence of the following properties being absent: ballot secrecy, non-purchaseable voting power and fee-less voting. In DLTs where mining is competitive, if one mining pool gains a transaction, the other do not, so we can model the mining pools to be explicitly distinct from each other, and the smallest mining pool is of size 1 agent. Public votes make it evident to see how a mining pool voted by simply looking up the on-chain transactions. Less influential miners should either align their vote to those of the largest mining pools, or risk having their transactions end up in a fork and not receive the reward for the work they performed to include it. It is therefore easy for a briber to identify miners of less influence and bribe them to ‘vote’ for their desired protocol. The small sized mining pools will make a profit, and the briber can identify how many miners they must bribe to succeed in implementing their desired protocol. As the authors in [36] showed, paradoxically, the more decentralised the network, the easier it is to create a systematic bribing cartel. Small sized miners are cheaper to bribe than large mining pools. The authors even provide a protocol to automate this vote buying in DAOs. This can be re-purposed to buy a miner’s vote in a DLT. The bribe can be paid out as soon as the smart contract detects a transaction following the desired protocol of the briber.

Submitting Malicious Protocol Updates

Malicious protocol updates are not infrequent, and not only in DAOs or DLTs.¹¹ Some go undetected due to low participation rates in governance, in others, the protocol update is forcibly approved by a whale, and in other occasions, the malicious protocol is hidden or not evidently detectable by the community. The consequences of this can be dire: double spending attacks, delaying governance outcomes, creating new tokens or loss of funds are amongst the many attack vectors reported.

It is not feasible to characterise all the ways in which malicious protocol updates may be submitted. Governance is only one of the components necessary in the DLT to prevent these exploits, but a robust governance protocol does not guarantee protection against malicious protocol updates. Access control mechanisms, security and trust assumptions are other components that must be taken into consideration. We encourage DLT design practitioners to make choices in their governance protocol that will alleviate the risk of this exploit. Namely, removing incentives to submit malicious proposals, or to comply with bribes to implement them.

7.7 Governance Properties

7.7.1 Participation

First we begin by considering who may participate in the decision making process of DLTs. These actors can be classified into the following set of agents: users, service providers, proposers and decision-makers. We introduce some terminology for ease of exposition.

¹¹[Backdoor found in Library update, used by major Linux distributions.](#)

Definition 7.7.1 (User). *A user is a participant of the DLT network that interacts with the DLT to receive a service from it.*

Definition 7.7.2 (Service providers). *A service provider runs a set of agreed upon instructions (protocol) in their own machine that enables the functioning of the DLT.*

Definition 7.7.3 (Decision-makers). *A decision-maker is an agent that has the power to vote on a new proposal.*

Definition 7.7.4 (Proposer). *A proposer is an agent that can put forth a protocol update which will be subject to vote.*

Each of these sets of agents may overlap, service providers may also be proposers, users may also be decision-makers and so forth. The intersection of these sets is application specific to each DLT. There are two questions that must be addressed with regards to participation: suffrage and distribution of power. The former addresses who is allowed to vote, and the latter, how power is distributed amongst the voters. We advocate for equal suffrage. This property means everyone eligible to vote has the right to cast a vote, and that everyone's vote carries the same weight [451]. This property removes the possibility of weighted votes, thereby enforcing equal voting power distribution.

This property may be too strict in some contexts. Some DLTs may wish to grant voting rights proportional to reputation or trustworthiness. However, we strongly discourage practitioners from distributing voting power proportionally to wealth.

In summary, two important properties of governance participation are:

P1 Equal suffrage.

P2 Non-purchasable voting power distribution.

Note that **P2** is a weaker notion of **P1**. **P1** states each voter should receive one vote and all votes should be of equal weight. **P2** allows for votes of different weights, as long as the weight cannot be purchased.

Possible implementations: Property **P1** advocates for a one-person-one-vote mechanism to distribute voting power. This can be easily achieved in permissioned DLTs, since each participant's identity is known. In permissionless DLTs, this is more challenging to achieve if a decentralised structure is to be retained.¹²

This is because it is non-trivial to distribute unique identity credentials in a decentralised manner, and in a way in which a single user cannot easily obtain multiple identity credentials. Indeed, an ongoing research effort is dedicated to answering this question [170], [223]. The Proof of Humanity project aims to address this challenge. It is a peer-to-peer

¹²It can be argued that some permissionless DLTs are already not truly decentralised if the majority of voting power is held by a small number of entities and participation rights are assigned proportional to wealth. Stake distribution of Ethereum is thoroughly documented in [169], showing that centralised exchanges such as Coinbase and Binance hold approximately 25% of the total ETH staked.

protocol that issues unique identity credentials to its participants [400]. They each create an identity, and it is then approved by other validated participants in the protocol. It offers the benefit that it is a more decentralised solution (it distributes trust), but participants must wait for others to vouch for their profile, and it is built on Ethereum so it is only available on their ecosystem. Bridges are protocols that allow for the transfer of data or transactions across different blockchains, and may be leveraged in this case to enable Proof of Humanity on other compatible blockchains [312].

Semaphore is an Ethereum based zero-knowledge protocol that allows members to cast a vote without revealing their identity, whilst demonstrating they belong to a group of valid voters [443]. In this manner, voters can only participate if they prove in zero-knowledge that they own a valid, unique voter credential, and can vote without having to reveal it. It is possible to assign one person one vote in permissionless DLTs by combining a protocol such as Proof of Humanity with Semaphore. Proof of Humanity ensures only approved (real) individuals receive a unique identity credential. Then, voters can prove they own such a credential in zero-knowledge, meaning they do not need to publicly disclose their credential, through Semaphore. In [207], Andrea Gangemi and I propose combining these two protocols in the context of resolving disputes in Web3 platforms. We note both Proof of Humanity and Semaphore are only compatible with Ethereum. We encourage DLT engineers and researchers to prioritise delivering these functionalities across varying DLT architectures.

In summary, one-person-one-vote is a fair and democratic way of allocating voting rights but it will come at the expense of re-introducing some form of centralisation in permissionless DLTs to regulate who is considered a valid voter, and ensuring they only receive one unique identity credential.

7.7.2 Incentives

Many DLT protocols assume their governance system is secure due to the financial incentives offered to their participants. Unfortunately, in a number of cases these governance protocols are not clearly defined, documentation is sparse, outdated or spread across multiple sources [198], [258], [257]. In some cases, documentation that was used for this study was later removed [192], [362], making auditing challenging, especially if no archive of these sites is available. Some protocols are so complex it is non-trivial to outline a clear mathematical model of how all of the incentives interact and what the net best strategy for an agent would be [378]. As a result, undesirable scenarios may arise whereby agents are inadvertently incentivised to undermine the network [563]. There are two crucial incentives that must be clearly defined and identifiable in the governance protocol:

I1 Incentive to vote.

I2 Incentive to make useful and beneficial proposals.

Property **I1** encapsulates whether the voter has an incentive to participate honestly in the voting protocol, and **I2** ensures there is no incentive to make undermining proposals.

We then consider two more necessary properties:

- I3** Distributed liability: risk is appropriately distributed amongst the governance participants.
- I4** Binding outcome: the outcome of the election must be implemented, provided the outcome is valid.

I3 is necessary to encourage the implementation of valid election outcomes. If service providers have no liability for not complying, **I4** is hard to enforce. Service providers may simply choose to not implement the outcome, without incurring on any cost for doing so. DLT practitioners must select an appropriate function to distribute risk, which is context specific to their own application. We highlight, it does not suffice with distributing liability amongst agents: it must be clear and traceable how this liability is distributed as well. Election outcomes must also be binding, otherwise the governance mechanism is simply a polling or referendum mechanism.

Possible implementations: Some DLTs implement the property of distributed liability (**I3**) by introducing penalties for non-compliance (also known as slashing [241]), which also creates an incentive to participate honestly in the governance protocol. Implementing properties **I1** and **I2** is straightforward: direct rewards can be granted to participants upon voting or if the proposals they made receive sufficient support and are correctly implemented. Many cryptocurrencies already provide these rewards [193, 378, 198]. To implement **I4**, elections must be held on-chain, and those with the power to implement the election outcome must be able to participate in the election. The challenge lies in being able to characterise how all these incentives interact with each other, and whether other incentives in the cryptocurrency negatively interfere with them. For example, on-chain voting will introduce fees to cast a vote, and this may directly interfere with a financial incentive offered to voters for participating.

7.7.3 Voting Mechanism

Social choice theory has long studied the different properties that vote aggregation algorithms offer. Arrow famously demonstrated that no ordinal¹³ voting scheme can simultaneously satisfy three seemingly simple and desirable properties:

- V1** Independence of Irrelevant Alternatives: If A is preferred over B, then introducing an irrelevant option C should not affect the outcome.
- V2** Pareto efficient: if the majority of voters prefer A over B, the voting system should output A over B.
- V3** Non-dictatorial: there is no one voter that can single-handedly determine the election outcome.

¹³Ordinal voting schemes only capture preference, whereas cardinal schemes reflect how much more one candidate is preferred over another [486].

[31]. The Gibbard–Satterthwaite theorem further outlines that no ordinal voting scheme is simultaneously non-dictatorial and robust against tactical voting [433]. In other words, that it does not satisfy **V3** and **V4** where **V4** is defined as:

V4 Strategy-proofness: The best way for a voter to achieve their preferred outcome is to vote according to their preference.

This does not bode well for ordinal voting schemes, but none of these results apply to cardinal voting schemes. A caveat with cardinal voting schemes is that it is important to find a meaningful function to allocate value to preferences. One example is Quadratic Voting [304], whereby each extra vote for a candidate is quadratically more expensive for a voter. Another class of voting mechanisms are those that are probabilistic¹⁴. An example is the serial random dictator model or Maximum Entropy Voting [445], which returns the candidate with least ‘surprise’, or in other words, the least controversial choice according to the electorate’s preferences. Probabilistic voting mechanisms offer desirable properties such as **V1**, **V2** and **V4** (or slightly weaker notions of the latter two), and in some instances even **V3** [445], [42], [294], [41]. They also completely sidestep any of the theorems outlined by Arrow, Gibbard [214] and Gibbard–Satterthwaite.

Possible implementations: We strongly advocate for cardinal (provided a meaningful utility function to allocate preference intensity exists) or probabilistic schemes. These have been shown to satisfy the highest number of desirable properties by researchers in the field of social choice.

7.7.4 Security

In this section we present a non-exhaustive survey of formal notions of security for online voting schemes, and existing implementations that satisfy these. We highlight that there is no academic consensus as to what the formal, cryptographic definition of these notions should be. DLT practitioners should select the primitives that offer the security level necessary or desired for their application.

Let us proceed by outlining the core notions of secure voting schemes. They must ensure the following properties are satisfied [262], [379], [441]:

- S1** Ballot Secrecy: it should not be possible to determine how a voter voted from their ballot.
- S2** Verifiability: voters should be able to assert that the election outcome reflects the will of the voters.
- S3** Coercion resistance: a voter can not comply with the instructions of an adversary on how to vote, and this behaviour cannot be detected by the adversary.

Ballot Secrecy: The quest to formally define **S1** is non-trivial. A recent survey of formal

¹⁴A probabilistic voting scheme returns an outcome depending on a certain probability. The same election run multiple times may not always return the same winner [102].

definitions of Ballot Secrecy is available in [464]. In it, the author also presents a formal definition of ballot secrecy. It is the state-of-the-art and it accounts for an adversary that has the power to intercept ballots as they are cast. In the same work, Smyth also proves that a variant of Helios voting scheme, namely a variant of Helios 3.1.4 satisfies ballot secrecy. This variant of Helios is known as Helios'16 and can be found in [466].

Possible implementations: A number of voting scheme implementations exist that satisfy notions of ballot secrecy. Well studied proposals include Helios [8], JCJ [268], Civtas [127] and Athena [463]. All of these protocols offer encrypted ballots, but not all provably satisfy the same notions of ballot secrecy, some have weaker security (earlier versions of Helios), others stronger, (JCJ and Civtas) but pay a price for it with an increased computational complexity. We note that none of these schemes have been implemented on-chain. However, there are numerous voting smart contracts implemented, thus evidencing it is possible to tally votes on-chain. Whilst other on-chain voting schemes exist, most do not provide encrypted ballots. We find VoteCoin [438] and MACI [333] are the only on-chain voting schemes that provide encrypted ballots. Unfortunately, VoteCoin provides encrypted ballots only while the election is occurring. This forgoes secrecy once the election is over, enabling coercers to check if their victim complied. The same can be said about the work in [295]. In MACI, votes remain encrypted, but the coordinator tasked with tallying individually decrypts ballots. If these are intercepted or if the coordinator is not trustworthy, votes may be easily revealed. Note that other protocols exist that provide anonymous voting, whereby the vote is public but the identity of the voter is hidden. Examples include Snapshot [303] and Vocdoni [519]. This is **not** congruent to secret ballots and should not be considered an appropriate security measure especially in DLT implementations, where identities are at best pseudonymous [25].

In summary, DLT practitioners have a number of options to ensure their governance protocol satisfies notions of ballot secrecy. They may run off-the-shelf protocols such as Helios'16, Civtas, Athena or others previously mentioned. These can be run off-chain, and the results may be notarised on-chain leveraging zero-knowledge proofs. These can be used to demonstrate the election result was correctly computed. Alternatively, if on-chain voting is essential, practitioners may settle for options such as MACI or VoteCoin, which offer encrypted ballots, but these options are insecure and neither meet formal notions of ballot secrecy.

Verifiability: An informal notion of **S2** in electronic voting schemes is end-to-end verifiability. If this is satisfied, a voter should be able to verify their ballot was:

- cast as intended,
- recorded as cast,
- tallied as recorded.

[70]. Another informal notion of verifiability is universal and individual verifiability. The

authors in [297] state verifiability can be decomposed into:

- Individual verifiability: voters can check that their own ballots are recorded.¹⁵
- Universal verifiability: anyone can check that the tally of recorded ballots is computed properly.
- Eligibility verifiability: anyone can check that each tallied vote was cast by an authorized voter.

The work in [465] then provides new formal cryptographic definitions for the three aforementioned. Whilst the informal notion of verifiability is rather intuitive, formalising such notion has been an arduous endeavour. For a comprehensive review of the literature and a proposed formal notion of verifiability, the reader may refer to the work in [136].

Possible implementations: regarding available implementations, all e-voting schemes satisfy notions of verifiability (some weaker than others). It becomes a matter of deciding the level of security and the trust assumptions one can make in their system, to pick the right verifiability formalisation. Achieving verifiability alone is trivial, as can be seen from most blockchain voting implementations: simply publicly revealing votes yields verifiable voting schemes. The challenge lies in achieving both verifiability and secrecy, without sacrificing one in favour of the other. We stress: practitioners do not need to compromise, achieving both is indeed possible, and is in fact necessary. The possible implementations of ballot secrecy outlined earlier in 7.7.4 provide verifiability as well as ballot secrecy (granted that each adhere to different formal notions of both ballot secrecy and verifiability).

Coercion Resistance: The work in [268] introduced a paradigm shift in the concept of coercion-resistance. Prior to it, Benaloh and Tunistra introduce the notion of receipt-freeness as means to ensure voters cannot prove to a third party how they voted [69]. However, the authors in [268] consider a much more powerful adversary, one that may ‘demand of coerced voters that they vote in a particular manner, abstain from voting, or even disclose their secret keys’. They provide a formal notion of coercion resistance, and an election protocol that provably satisfies said notion. Note that this definition of coercion resistance partly captures censorship resistance¹⁶, as the adversary may demand of the voter to abstain. However, it does not consider an adversary that may suppress a ballot from the tallying phase. A recent survey of coercion resistant voting schemes can be found in [55] in section 6.5.1, and their respective time complexities is shown in Table 6.2. None of these schemes are implemented on-chain. The authors in [471] propose a coercion resistant voting protocol using blockchain to cast votes. They also propose a security notion of Vote Indistinguishability (VI), which is congruent to secret ballots, and show their voting protocol is VI secure.

¹⁵The property of individual verifiability is closely related to censorship resistance, but it is a weaker notion than the latter. Individual verifiability allows a voter to detect if their vote has been suppressed, whereas censorship resistance demands that votes cannot be suppressed in the first place.

¹⁶The property of censorship resistance means it should not be possible to suppress a valid voter’s ballot [523].

Possible implementations: for on-chain voting, DLT practitioners may implement [471]. Alternatively, one can select any of the aforementioned e-voting schemes that are coercion resistant (for example [127]), run them off-chain and notarise the result on-chain leveraging zero-knowledge proofs. Finally, a number of other coercion-resistant schemes have been proposed but not implemented. These may be implemented on-chain if the DLT of choice allows for smart contracts in layer 1, or similarly, implement them off-chain and notarise the result on-chain.

7.8 Evaluation

We proceed by selecting a diverse range of cryptocurrencies to evaluate varying types of governance models, DLT architectures and functionalities. Bitcoin and Ethereum serve as foundational examples of blockchain systems with informal or off-chain governance processes [133], [195]. Cardano is an interesting case-study, as its participants just recently approved a constitution to formalise its governance process.¹⁷ IOTA is an example of a Directed Acyclic Graph DLT, as opposed to a blockchain architecture [401]. Solana was selected due to its alternative consensus mechanism, Proof of History, as well as its high transaction per second rates and low transaction fees [543]. Avalanche is selected due to its consensus mechanism, the integration of subnetworks¹⁸, and the inclusion of multiple built-in blockchains [423]. Terra Luna was included as a case-study of a cryptocurrency that collapsed partly due to its governance structure, which failed to implement effective safeguards or timely interventions to address vulnerabilities in its algorithmic stable-coin model [321], [95].

¹⁷Cardano Constitutional Convention 2024.

¹⁸The subnetwork architecture is one in which participants validate different subsets of blockchains in the network [442].

Property	Bitcoin	Ethereum	IOTA	Polkadot	Cardano	Solana	Avalanche	Terra Luna
Equal suffrage	No	No	No	No	No	No	No	No
Non-purchaseable voting power	No	No	No	No	No	No	No	No
Incentive to vote	None	Social	Social	Monetary	Monetary	Monetary	Monetary	Monetary
Incentive to make beneficial proposals	None	Social	Social	Monetary	Monetary	Monetary	Monetary	Monetary
Distributed liability	Yes (Miners)	Yes (Miners, Validators)	Yes (Users)	Yes (Users, Validators)	Yes (Users, Validators)	Yes (Validators)	Yes (Validators)	Yes (Both, Organising Foundation)
Binding election outcome	No	No	Yes	Yes	Yes	Yes	Yes	Yes
Independence of irrelevant alternatives	No	No	No	No	No	No	No	No
Pareto efficient	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Non-dictatorial	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Strategy-proofness	No	No	No	No	No	No	No	No
Ballot secrecy	No	No	No	No	No	No	No	No
Verifiability	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Coercion resistance	No	No	No	No	No	No	No	No
Censorship resistance	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Partially

Table 7.1: Governance properties satisfied in varying cryptocurrencies.

For each of the selected cryptocurrencies, we survey their governance documentation and review the findings regarding each governance property below. The sources used for this section are the governance documentation pages of each of the cryptocurrencies: Bitcoin [133, 374], Ethereum [195, 194], IOTA [196, 258, 257], Polkadot [378], Cardano [193], Solana [198, 326], Avalanche [192, 423] and Terra Luna [362].

7.8.1 Participation

Regarding suffrage, whilst none of the selected cryptocurrencies satisfy equal suffrage, token-based systems still allow all token holders to participate in governance. In that sense, suffrage is not completely denied to any eligible participant. The inequality arises in *voting power*, not in the right to vote. All selected cryptocurrencies use some variant of token-based participation except Bitcoin, that grants participation rights proportionally to computational power. None deliver non-purchasable voting power: all selected examples use either Proof of Work, Proof of Stake or a variant of either to achieve consensus. In both of these systems, users may acquire greater power in the network by either purchasing more computational power or investing more wealth in the DLT network.

7.8.2 Incentives

Incentives to vote are delivered by Polkadot, Cardano, Solana, Avalanche and Terra Luna through direct financial compensation to the voters, whereas the incentive in IOTA is indirect. Stakeholders are motivated by the ecosystem’s success. Ethereum does not offer direct recompense to voters, but applications built on the Ethereum ecosystem often do, such as DAOs. Bitcoin offers no financial reward or alternative incentives for participants to vote in the decision-making processes.

Similarly, Bitcoin offers no rewards or incentives to make beneficial improvement proposals. Bitcoin lacks a formal governance process for proposals. Changes are made via Bitcoin Improvement Proposals (BIPs), which require volunteer effort without direct economic incentives for proposers [133]. No built-in mechanism rewards making beneficial proposals, so participants rely on altruism or indirect benefits, such as network improvement. Ethereum also lacks direct incentives to make proposals, which are submitted as Ethereum Improvement Proposals (EIPs) [195]. IOTA’s governance includes mechanisms for proposing changes, but there are no explicit financial incentives for proposers [196]. Participants may be motivated indirectly by the chance to improve the protocol or gain reputation within the ecosystem. The remaining cryptocurrencies all have mechanisms to fund winning proposals, with Terra Luna, Cardano and Polkadot providing said funding from their own treasury [362, 193, 378].

Bitcoin and Ethereum rely on voluntary adoption and off-chain agreement, making their governance outcomes non-binding. The rest of selected cryptocurrencies all implement on-chain voting, thereby making the outcome automatically binding.

Regarding liability, Bitcoin miners are responsible for consensus and validation. In Ethereum,

miners were responsible when Ethereum used Proof-of-Work for consensus, and validators¹⁹ are responsible since Ethereum shifted to Proof-of-Stake consensus. Participants indirectly manage the DLT in IOTA, since they can be both users and service providers at the same time. Validators and users share liability for governance and consensus in Polkadot, Cardano, Solana and Avalanche. However, Terra Luna operated differently prior to its collapse: both users and the Organising Foundation (Terraform Labs) assumed liability in the governance process.

7.8.3 Voting Mechanism

The table below summarises the voting aggregation algorithms used in each cryptocurrency. Most use a token-based variant of plurality voting (also known as first-past-the-post or majority voting). Majority voting is known to satisfy Pareto Efficiency and is non-dictatorial, but it is not independent of irrelevant alternatives and therefore is not strategy proof [346]. The exceptions are Bitcoin and Ethereum, neither of which use on-chain voting. Bitcoin nonetheless satisfies the same properties because it also utilises a majority voting scheme, although the 95% threshold is likely to reduce the influence of irrelevant alternatives in practise, compared to the 51% threshold. We are unable to characterise the properties that Ethereum’s voting scheme satisfies because there are no clear quorum or thresholds defined for adoption.

Cryptocurrency	Threshold for Proposal Approval
Bitcoin	95% of miner signalling.
Ethereum	‘Rough consensus’.
Solana	50%+1 of staked SOL (quorum rules may apply).
IOTA	50%+1 of Mana (majority-based).
Cardano	Planned: 50%+1 ADA + quorum (higher for major changes).
Polkadot	Conviction-weighted majority, often >66% for critical votes.
Avalanche	50%+1 AVAX (quorum-dependent).
Terra Luna	40% quorum; 50%+1 approval threshold.

Table 7.2: Voting thresholds for proposal approval across various cryptocurrencies.

7.8.4 Security

All selected cryptocurrencies reveal the votes publicly, therefore none satisfy notions of ballot secrecy. Conversely, verifiability is obviously satisfied, since all votes are public, everyone can check the election outcome is correct. The public nature of votes however, makes coercion resistance impossible. Because a voter can prove how they voted, they can be bribed or threatened to vote a certain way. On-chain voting means miners or validators are responsible for processing transactions containing votes. They have the power to exclude transactions from the DLT containing votes, and therefore formal notions of censorship resistance cannot be provably guaranteed. In practise, most miners or valida-

¹⁹Validators are the service providers in DLTs that use Proof-of-Stake for consensus.

tors do include valid transactions and follow the protocol, but censorship resistance is not guaranteed: researchers have found miners may still misbehave by reordering, inserting, or censoring transactions for their own gain [145].

7.8.5 Missing properties

From the evaluation of the selected cryptocurrencies, we find there are a number of currently disregarded good governance properties. These include:

- 1. Binding Elections:** some governance protocols carry out their decision-making process on off-chain platforms [195, 133], meaning the results are not binding. Similarly, some schemes allow participants to vote that are not responsible for implementing the results [133].
- 2. Censorship resistance:** it must not be possible to suppress voter's votes.
- 3. Incentive alignment:** the governance protocol must not create active incentives to attack it. In other words, if voters act following a strategy that maximises their benefit, this strategy should not be one that actively undermines fairness, security and decentralisation in the governance protocol.
- 4. Non-purchasable voting power:** voting rights should not be distributed proportionally to wealth.
- 5. Fee-less voting:** voters should not be required to pay to vote.
- 6. Ballot Secrecy:** votes must remain secret. Ballot secrecy should not come at the expense of verifiability.

In current DLTs, achieving all the above properties is not possible, since some are mutually exclusive. We encourage practitioners to consider the context specific security and performance needs of their DLT and select a compromise appropriately. We proceed by outlining which properties are mutually exclusive and why that is the case.

Binding elections and fee-less voting: to ensure that election outcomes must be implemented on the DLT, these must be held on-chain. However, on-chain voting sacrifices fee-less votes, because to cast a vote on-chain one must pay transaction fees. On-chain voting in DLTs also currently sacrifices ballot secrecy in favour of verifiability, although this need not be the case.

Binding elections and censorship resistance: on-chain voting cannot guarantee censorship resistance. However, this concern is application specific, as some DLTs may offer increased levels of censorship resistance due to their protocol design and network distribution. This is also dependant on how strong a notion of censorship resistance DLT practitioners wish to provide.

Non-purchasable voting power and permissionless access to the DLT: this is currently the most prevalent issue in DLT governance. DLTs were designed to avoid having to trust central authorities in order to deliver a service. In order to achieve this whilst being protected against resource attacks, they must still offer some protection against

Sybil attacks²⁰. In other words: because DLTs aim to be decentralised, they must avoid scenarios where any actor can access the network and attack it. Therefore, access to the DLT must be regulated. Currently, to avoid doing this by having to again rely on a centralised trust authority to grant access credentials, access is regulated through a proxy for wealth. The wealthier amass more voting power as a consequence. De-facto, this re-introduces centralisation in the DLT. We advocate for access control mechanisms that do not prioritise the wealthier, but are based on other criteria. An example is the Tree-Proof-of-Position mechanism presented in Chapter 4. This decentralised class of algorithms allows agents to prove they are in a given position in a collaborative manner. A scheme like this may be used to regulate access to DLTs, such that only agents with a valid proof of position may be granted participation rights.

Secret ballots and verifiability: current implementations of DLT governance also forgo ballot secrecy in favour of verifiability. Votes are cast publicly on-chain, or are publicly decrypted at the end of an election. A number of e-voting schemes have been developed to achieve both. Examples include: [8], [425], [127], and even proposals that address coercion resistance such as [268].

7.9 Conclusion

This chapter underscores the critical role of governance in ensuring the security, fairness, and sustainability of Distributed Ledger Technologies (DLTs). By identifying and categorising vulnerabilities that arise from poor governance, we have shown how these systems, despite their promise of decentralisation and transparency, remain susceptible to exploitation. The outlined threats—ranging from coercion and vote-buying to centralisation of power—highlight the urgent need for robust governance protocols that align with established principles of fairness, security, and accountability.

The key contributions include:

- A taxonomy of formalised governance properties that defines the minimum requirements for ensuring good governance in DLTs.
- A comprehensive analysis of vulnerabilities stemming from governance deficiencies and their implications for stakeholders.
- Guidance on technical solutions, bridging insights from cryptography, social choice theory, and e-voting systems to propose actionable pathways for practitioners to implement good governance properties.

This work is necessary because the stakes are high: as DLTs increasingly intersect with public policy, financial systems, and societal infrastructures, the absence of proper governance safeguards risks undermining trust, enabling scams, heists, hacks, centralisation of

²⁰A Sybil attack is one where an agent easily obtains multiple identities in the network to gain unfair advantages [360].

power and rug pulls. By addressing these vulnerabilities, we not only protect the integrity of DLTs but also uphold their potential to serve as tools for social good.

Future work in this area should concentrate on the development of scalable, technical solutions for delivering necessary governance properties within DLTs. Particular attention should be given to alleviating the tensions between properties that are mutually exclusive yet equally vital for good governance. A key point of improvement should focus on developing permissionless access to DLTs that does not enable purchasable voting power, and delivering fee-less participation. Equally important is the mathematical modelling of incentives in participation and proposal making.

Progress in governance design will also depend on interdisciplinary collaboration. Insights from regulatory, social, and technical perspectives must be integrated to produce governance mechanisms that are both theoretically sound and practically enforceable. Within this process, it is crucial to establish clear and tangible definitions of good governance in policy and regulation. Absent such definitions, projects cannot be held accountable, and systemic vulnerabilities will persist. Finally, there is a pressing need for standardisation efforts aimed at defining and promoting universally accepted frameworks for good governance in DLTs. Establishing such standards would provide common benchmarks against which systems can be evaluated and thus held accountable for.

We therefore call upon the DLT community—researchers, developers, users, and policy-makers—to prioritise governance as a fundamental component of system design. Only through collective effort and innovation can we ensure that DLTs fulfil their potential as equitable and secure platforms for a decentralised future.

Chapter 8

DAO Governance: Systemic Failures and a \$163 Million Heist

Abstract

DAO governance is currently broken. We survey the state of the art and find worrying conclusions. Vote buying, vote selling and coercion are easy. The wealthy rule, decentralisation is a myth. Hostile take-overs are incentivised. Ballot secrecy is non-existent or short lived, despite being a human right [506]. Verifiability is achieved at the expense of privacy. We present a case study of Aragon DAO's governance solution that is not fit for purpose. A year after this research was presented, Aragon DAO suffered an attack exploiting the shortcomings we analysed. \$163 million worth in tokens were extracted. We highlight necessary measures to prevent this from recurrence.

8.1 Contributions Disclosure

This work was developed by myself with the collaboration and supervision of Ben Smyth.

8.2 Motivation

Thus far, we have evaluated DLT's governance structures, under the assumption that without good governance, the architectures built on them are by extension vulnerable too. Does this assumption hold in practise? This work aims to uncover the answer. We find that when it comes to DAOs, unfortunately so. This chapter summarises the contributions developed in [277] and [278], which established the research starting point for Chapter 7. Here, we shift from DLTs in general to DAOs in particular, asking whether these organisations deliver on their promises of decentralisation and collective control.

DAOs must be understood within the wider Web3 movement from which they emerged. The Web3 revolution heralded an era of quick riches [369, 64]: participants sought to earn monetary gain, since they realised they are the product [403, 216, 283]. Advocates of the

movement wish to change the state of affairs: gone are the days where the users provide value and the services reap the reward [436, 75, 371, 535]. Users want a voice and a share of the returns [444]. The movement has spawned the creation of countless cryptocurrency projects, with the aim (allegedly) to deliver just that: decision-making power and financial reward to the participants, who are now also the service providers. DAOs have become a prominent choice of organisation, built on DLTs, to serve these purposes [415, 354]. Their structures are akin to co-operatives, where participants own a share of a resource, and vote on how the resource should be managed and on how the organisation should be run [457, 66].

Since their birth in 2016, the emergence of DAOs has only but increased. This increase does not show any signs of slowing down. According to the data provided by DeepDAO [157], where in 2018 there were 10 DAOs, by 2020 there were approximately 200 [494]. The influence and assets that DAOs hold has also increased. In 2021, the total Assets Under Management held by DAOs was \$520.7 million. Currently their total treasury has exploded to \$22.5 billion as of September 2025 [157]. Likewise, the number of DAO governance participants increased by 130 times from 13,000 to 11.8 million. Given this growth and financial prevalence, it is of utmost importance to study if the promises DAOs offer are actually delivered. Whilst cryptocurrencies and DLTs have been subject of extensive academic study, DAOs at time of conducting this research (June 2022), had not. Even in 2025, we found little on the crisis of DAO governance's state of affairs. In this work, we uncover the truth: wealthy minorities amass voting power, vote buying is legal, vote selling is incentivised, hostile takeovers frequent and coercion is easy.

8.3 DAO Governance: From Promise to Peril

It's 2016: DAOs are in their infancy, The DAO¹ has garnered attention having raised \$150 million of Ethereum tokens. Three months after launch, The DAO is hacked, a smart contract bug exploited, [160] \$50 million siphoned off their funds [160]. The aftermath raising questions over blockchain philosophy and the technology's future. Were funds obtained legally? 'Code is law' is regulation enforced by technology [239]. It underpins the functioning of DAOs and blockchain. If software is exploitable, then in theory, no law is broken. Victims lost their funds unfairly. Ethereum founder Vitalik Buterin proposed a soft-fork (a software update in the blockchain proposal) to right the 'wrong'. The solution was promptly abandoned; it too contained a bug, making it vulnerable to further attacks. The tokens amassed by the attacker gave them enough legislative power to enact decisions in The DAO. The alleged attacker responded by threatening to bribe miners to not comply with the soft-fork. They argued no smart contract rules were broken when obtaining the funds.

The DAO's value exceeded the cost of acquiring enough votes to take control, incentivising the heist. There is no need to break the laws established by the DAO to succeed.

¹Confusingly, *The DAO* is the name of a DAO.

Fast-forward to 2018: history repeats, another DAO is victim to poor governance. This time no bug was exploited, the attacker simply acquired enough tokens, bought the vote, approved their own proposal [274]. The coup drained nearly \$500,000 tokens from the Build Finance DAO.² The attacker covered their tracks using Tornado Cash, anonymising stolen funds. Token-based voting legalises coups—anyone can legitimately buy their way to power. Incentive makes takeovers inevitable if the cost is cheaper than the reward.

Democracies embrace one-person one vote. Acquiring multiple votes undermines fairness, equality. Token-based voting is incompatible with equality and fairness. Tokens are not a proxy for identity, their ownership is easily transferred. Wealth amasses tokens, buys legislative power, corrupts decision making [385]. A voting system that allows voters to buy more votes converges to plutocracy, the unwanted symptoms of centralisation, low representation of the electorate [48] and game theoretic incentives to attack the DAO [104].

Chapter 7, Section 7.6.1 introduces how these incentives negatively impact DLT governance, and the same issue arises in DAO governance. Wealthy agents can purchase voting power, and when it comes to voting, small to mid-sized token holder's votes are not as powerful. There is no incentive for small to mid-size token holders to vote against the wealthy agents, because to cast a vote on-chain, voters must also pay a transaction fee. Thus, the same incentive discussed earlier arises in DAOs: voters are incentivised to abstain [345], or worse, to sell their vote. The latter guarantees a winning strategy. In fact, the vote buying can be automated and verifiable [146]. Vote buyers can confirm their purchases, since votes are typically revealed during or after an election. Secondly, the ownership transfer of a vote is remarkably easy. The voting ability and power is linked strictly to tokens, not to an identity. Cryptocurrencies, by design, enable fast and simple transfer of said tokens. Vote-buying cartels can emerge: from simple smart contracts to pay out voters automatically upon proving compliance, to cartels buying trusted hardware executing vote buying code. Particularly, the latter is an attack vector that is essentially undetectable [36].

The insights gathered in [366] confirm the incentive to abstain, the dangers of public voting and the centralisation of power. DAO governance was studied with a focus on Dash DAO as a case study. Researchers accessed the voting history of Dash DAO's masternodes, given that these are public. Worryingly, IP identifiers, software version and wallet addresses were public too. Voting patterns of 4987 masternodes who participated in voting across 577 proposals were analysed. Researchers found that: 'some masternodes are not only abstaining from voting, but have disengaged from the voting process completely.' [366]. They also found a number of voters with almost identical IP Ports, strongly indicating that they are mounting Sybil attacks to gain voting power. Further to this, they analysed the voting patterns of the DAO participants. Results show that there are small-sized, dense clusters of masternodes with identical voting patterns. Although smaller in number compared to the rest of voters, if these minority clusters were to collude, 'they would have more voting power than the entire decentralised majority' [366]. The consequences of a pay-to-vote system are harsh: low turnout, centralisation, preclusion of free will, coups

²DAO Coup, Vice.

and coercion. A preliminary study found less than 1% of token holders control 90% of the vote [112].³ Are DAOs decentralised when controlled by a wealthy minority? Clearly not—the wealthy do not represent the masses.

8.4 Necessity of Secrecy

Historically, “*Americans [voted] with their voices – viva voce – or with their hands or with their feet. Yea or nay. Raise your hand. All in favor of Jones, stand on this side of the town common; if you support Smith, line up over there*” [315]. Everyone present could verify that only voters voted and that the count was correct. But free will must be ensured, as dictated by the United Nations [506], the Organisation for Security & Cooperation in Europe [387], and the Organization of American States [388]. Yet public votes forgo freedom, “The unfortunate voter is in the power of some opulent man; the opulent man informs him how he must vote. Conscience, virtue, moral obligation, religion, all cry to him, that he ought to consult his own judgement, and faithfully follow its dictates. The consequences of pleasing, or offending the opulent man, stare him in the face...the moral obligation is disregarded, a faithless, ..., pernicious vote is given” [492]. The need for voting privately became evident. In-person voting ensures this by providing identical ballots that are completed in a private booth, a concept first introduced successfully in Australian voting in 1856 [380].

Ballot Secrecy in e-voting

In e-voting, the concept of secret ballots emerged parallel to the development of such voting schemes, originating with David Chaum’s first proposal of an end-to-end verifiable voting scheme in 1981. In it, voter’s ballots were private, and all participants could check that the tallying operation was correctly performed [118].

Forgoing ballot secrecy is to regress centuries of progress, violate human rights and re-turning coercion and inequality as norms. With that in mind, we warn: DAOs are in dire straits...

8.5 DAO Voting: Survival of the Richest

DAO members vote remotely, online. One of the methods is on-chain voting, where the public nature of distributed ledgers is leveraged, using them as a shared and verifiable database. Proposals are encoded into smart contracts and submitted to the ledger as a transaction. A vote in favour or against new proposals is cast as a transaction on the ledger. Winning proposals are automatically executed. Votes, proposals and election outcomes are all publicly verifiable [370, 310]. On-chain voting makes elections outcomes binding, without relying on a trusted intermediary or a board to implement results. Guarantees of immutability are provided by the ledger: once the results are announced, these cannot be tampered with. Mounting an attack to re-write the blocks requires practically

³Chainalysis only studied ten DAOs, further study would establish general trends.

infeasible computational power. On-chain governance uses distributed ledgers as a public (or permissioned, depending on the protocol) bulletin board. Despite its desirable properties, it has been subject to criticism [298, 489]. Its detractors argue that blockchain voting not only fails to mitigate security risks present in e-voting, but also introduces additional risks [395]. We agree.

Worryingly, the vast majority of on-chain, smart contract votes do not satisfy ballot secrecy. At worst, votes are revealed as cast, and at best, these are publicly decrypted after the voting period ends. Values of a token can be artificially inflated or devalued, ‘pump and dumps’ become simple. Whales (entities or individuals with large amounts of tokens) can manipulate the value of a token with their behaviour. They can express intention with public votes, swaying token values to their favour. Just before the election closes, they change their intention, make a profit and cash out.

Information on how a wallet address voted, when, and how many tokens they staked to that vote is available for anyone in the ledger to see. Wallet addresses are pseudonymous, not anonymous [25], and it is possible to link wallet addresses to individuals from information such as their transaction history [82]. Cryptocurrency mixers such as Tornado Cash hide this, but has also been maliciously used to launder millions—the U.S. Department of the Treasury’s Office of Foreign Assets Control (OFAC) sanctioned the crypto-currency mixer⁴ and the developers were arrested.⁵ This does not bode the end of cryptocurrency mixers however, as alternative ones are still in use, although frequently subject to sanctions and criminal investigations [270, 481, 497]. On-chain transaction fees means voters pay to vote. Fees can vary throughout an election, meaning not all voters pay the same price to cast their vote. They can be victims of miners refusing to cast their votes, and only the wealthiest will survive the financial hurdles. Paying to vote or increasing the weight of their vote proportional to their wealth discriminates against those who cannot do so from the decision making process. What if a coup happens? Forking the chain brings little solace: election records can be reverted, actual events cannot, history cannot be changed; assets may have already been cashed out.

8.5.1 Off-Chain Voting and Hybrid Alternatives

Alternatives exist that don’t use blockchain to cast votes. The most popular example is Snapshot, which many DAOs use solely or in combination with on-chain voting to enable governance. Snapshot is decentralised, using IPFS as its main storage layer [259]. It offers the advantage of no fees to cast a vote whilst still being decentralised thanks to their storage system. The election outcome however, is not automatically binding. It has to be bought on-chain. Because of this, Snapshot is often used for polling. AragonDAO, Uniswap and MakerDAO are examples of DAOs using a hybrid governance solution [142, 7, 336].

⁴Tornado Cash Sanctioned, CNBC.

⁵Tornado Cash Developers Arrested, The Hacker News.

8.6 Current Solutions: Insecure and Insufficient

Despite the dire situation of DAO governance, we observe that a shy but steady shift is occurring in the space. A number of projects are emerging to address some of the aforementioned issues, although they are still in their infancy.

Snapshot is pairing with Orange Protocol to develop a reputation based voting mechanism [407]. Responding to inequality, communities such as Algorand [15] and Dream DAO [167] are transitioning towards a merit based voting system to actively encourage participation and development of the network, and distribute voting power amongst the developers, not the wealthy. Moving away from vote purchasing governance models is necessary to avoid plutocracies and centralisation and ‘legal’ fund siphoning.

To address ballot secrecy, VoteCoin presents an on-chain voting solution offering encrypted ballots during the election process [438]. Snapshot are also developing a similar feature, offering ‘shielded voting’ whereby votes are private only until the end of the election.⁶ Privacy in this case, is short lived. A number of issues remain: verifiability is achieved at the expense of privacy by naively decrypting votes publicly. An option exists to allow an auditor to decrypt votes, but this introduces a trust assumption of honesty of the auditor. VoteCoin also requires voters to pay to cast their ballot.

Another relevant case study is Aragon DAO’s new governance solution: Vocdoni. They provide an on-chain voting solution that uses two blockchains: the Ethereum blockchain for the election process creation or status update, and the Vochain blockchain (Vochain), where votes are cast [519]. Vochain uses the Proof of Authority Tendermint blockchain, so only trusted nodes can relay transactions. Due to the use of two blockchains, there is a need for an oracle to relay information from the Ethereum blockchain to Vochain, to signal new voting processes. At time of writing, the oracle nodes are run as trusted nodes, however, Vocdoni proposes a roadmap to substitute them with Zero-Knowledge Rollups⁷ to allegedly make them trustless. According to Vocdoni’s documentation: “One solution to this problem is to make use of Zero-Knowledge Rollups as a vote aggregation and mixing mechanism between voters and the blockchain. This would make it impossible for any third party to verify that a voter chose a specific option” [519]. This claim is incorrect. As shown in Figure 8.1, the node computing the Zero-Knowledge Rollup receives the vote unencrypted, so they must be a trusted node. If this is not the case, the node computing the Zero-Knowledge Rollup can very easily reveal how a user voted. While the voter ID remains private, the prover computing the Zero-Knowledge Rollup will still know how a voter voted, given that it is them who send the vote to the prover in the first place. Even if the identity that a voter provides is a wallet address, these are pseudonymous. Indeed, the only obfuscated information is the ID of the voter within the census. Instead, the voter sends a zero-knowledge proof⁸ of inclusion demonstrating that their ID belongs to the set

⁶Snapshot shielded voting.

⁷A Zero-Knowledge Rollup is a proof system used to compress a number of transactions into a batch, with cryptographic assurance that these are correct. A more detailed overview is presented in [35].

⁸A zero-knowledge proof is a way to prove that someone knows a piece of information without having

of accepted voters.

To understand the implications of this, we illustrate a parallel example for the reader: on the day of voting, anyone wishing to vote must cast their vote publicly, but what is hidden is their ID card, replaced instead by a proof that you hold a valid ID card and thus should be allowed to vote. Nonetheless, the voter must hand their public votes to the administrators, who can easily see how they voted, and could identify them, because indeed you were the one to hand them your vote. This implies that first, a great deal of trust must be placed on the administrators to not reveal your vote to malicious agents, and second, that no one else except the administrators will be able to observe your ballot as you cast it. Vocdoni addresses the second assumption by mentioning that a private transport channel would be used to send the votes to the prover. This assumption introduces a weaker notion of security, and the fact that the votes remain public in this channel means that this system cannot provide notions of ballot secrecy wherein the adversary is assumed to have the power of intercepting ballots during their collection. We would like to highlight that ballot secrecy does not equate to public votes with anonymous identities. Furthermore, the identities are not anonymous in Vocdoni, they are at best pseudonymous to the Zero-Knowledge Rollup prover, if the private transport channel is not compromised, and even making this assumption, voters would not be equal: later voters have more information with public votes. This is because Vocdoni does not support encrypted ballots with anonymous voting.

We outline another vulnerability related to the ‘self-sovereign’ identity management of Vocdoni. In their protocol every user creates their own key pair [519]. What is preventing users from selling their private key? In the anonymous voting, what is being hidden is the identity of the voter, and not their vote, so giving the voter the ability to generate their own identity would be parallel to allowing voters to create their own ID cards at an election. Instead of selling their vote, voters can sell their proof of census inclusion, that is directly generated from their identity. In fact, anyone can verify if this proof is invalid, so malicious agents attempting to coerce voters could easily check if they are being deceived. Similar to the Dark DAO vote buying cartels outlined by [36], identity buying cartels could emerge operating in the same manner. Indeed, black markets selling various types of identities already exist [510, 130]. The work in [37] challenges the assumption that private keys are held by a single individual, demonstrating how it is possible to loan out keys to multiple users in a way that remains undetected on chain. This possibility directly undermines the security of ‘self-sovereign identity’, precisely that a user will not sell or loan their own self-generated identity.

Vocdoni does provide the option of having encrypted votes, but the voter identity remains known. They do not currently support both anonymous and encrypted voting at the same time. Similar to VoteCoin and Snapshot’s proposals, verifiability is once again achieved at the cost of privacy by publicly decrypting the results. With Vocdoni’s anonymous voting, the ballots are public, as shown in Figure 8.2. We again reiterate that anonymous voting to reveal it [219].

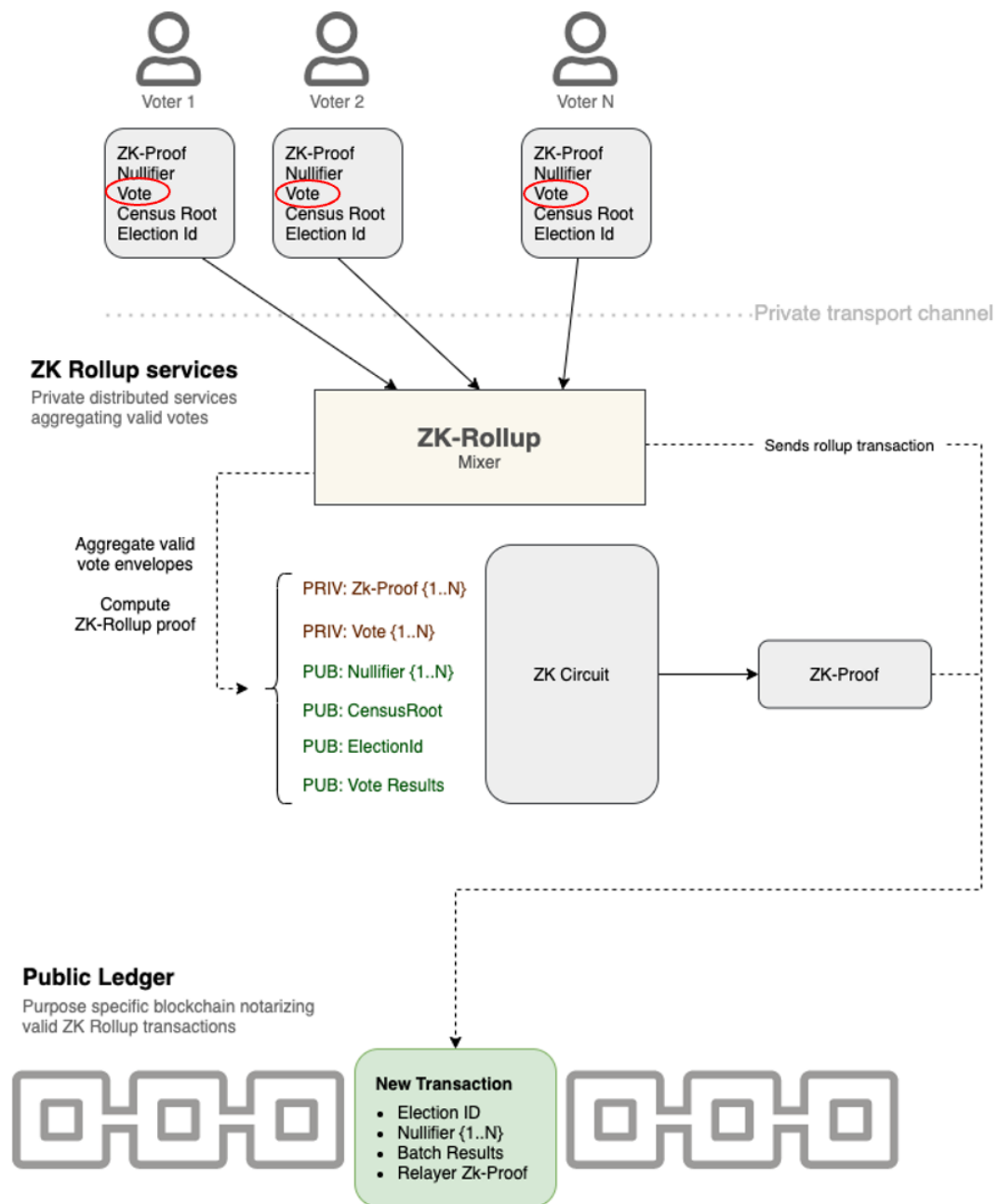


Figure 8.1: Vocdoni's Zero Knowledge Rollup Proposal. Source: [Vocdoni](#).

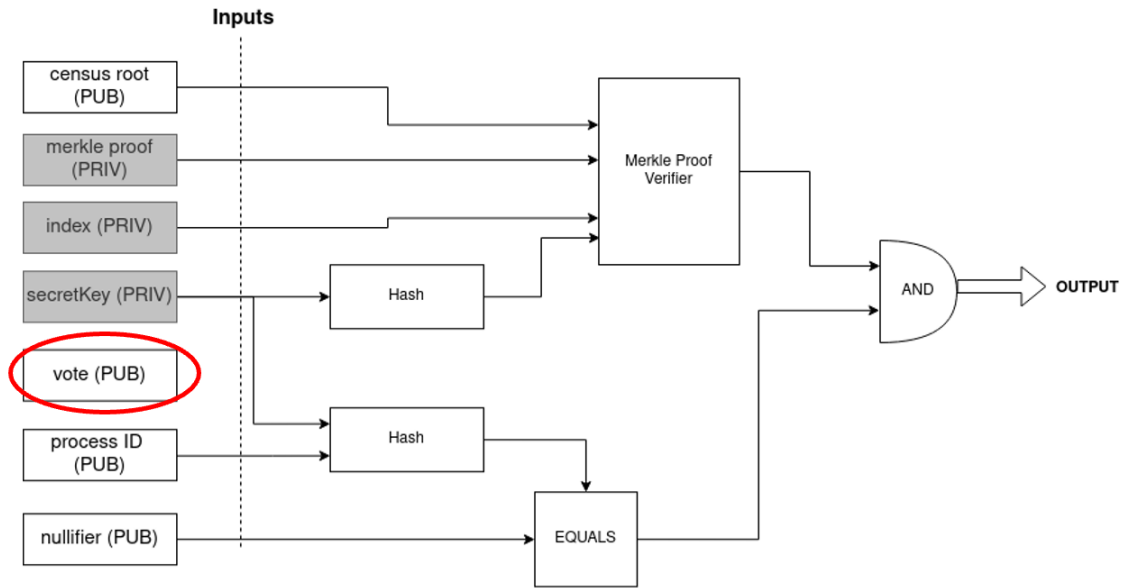


Figure 8.2: Vocdoni’s Anonymous Voting Schema. Source: [Vocdoni](#).

with public votes does not achieve ballot secrecy.

We summarise the state-of-the-art solutions in Table 8.1. The most used solution is on-chain smart contracts. It is a convenient option thanks to the existing integration platforms such as Tally and Boardroom providing a user-friendly platform to cast votes, tally them and summarise election outcomes. No options provide long term ballot secrecy. Voter’s identities are rarely kept private and verifiability comes at the expense of privacy in most.

Table 8.1: Current DAO governance solutions.

Voting Solution	Ballot Secrecy	Private Voter ID?	Public Vote Reveal	Fee to Cast Vote?
Snapshot	No	No	Yes	No
Vocdoni	No	Sometimes	Yes	Yes
VoteCoin	Temporarily	No	Yes	Yes
On-chain votes	No	No	Yes	Yes
MACI	No	Yes	No	Yes

8.7 Aftermath: the DAO Casualties

In the year after this research was completed (October 2022), a number of hostile takeovers and governance exploits of DAOs began. Aragon DAO, who’s voting platform Vocdoni is analysed above and shown to be insecure, suffered a governance exploit that cost the DAO \$163 million [394]. In October 2023, six wallet addresses purchased a majority of token supply and submitted a governance proposal to liquidate Aragon’s treasury. Despite objections from the DAO’s leadership, their token majority allowed them to pass the measure and withdraw more than 86,000 ETH—valued at roughly \$163 million [28]. This

group, later revealed to be called Risk Free Value (RFV) Raiders, is behind a number of DAO takeovers. They focus on those where game theory incentives exist to purchase sufficient voting power to control and thus sell the DAO treasury. The group sells the treasury and cashes out the profit. Other examples include Rook DAO, which resulted in a \$25 million governance takeover [420] and Mango Markets DAO, which suffered a \$110 million exploit. For the latter, the exploit was a ‘pump and dump’. The attackers were able to manipulate the market of the Mango governance token, by placing a large purchase of tokens, thus inflating the price. This made their futures positions on Mango appear extremely profitable. Mango let those unrealised gains count as collateral, so the attacker was able to borrow and withdraw over \$110 million from unrealised gains [469]. No malicious code or bug was exploited, the attacker used the protocol as designed, and leveraged game theory incentives to profit. Subsequently, the attacker used voting power purchased with own cashed out unrealised gains to submit and approve their own proposal in the Mango Markets DAO. The proposal encoded an offer to return some funds (\$67 million) to the now crashed Mango Market, minus a hefty bug bounty (\$47 million, the largest ever recorded), and the DAO agrees not to pursue criminal charges [377].

One of the participants, Avraham Eisenberg, was later arrested and charged with commodities fraud, commodities manipulation and wire fraud. Eisenberg has labelled himself as a ‘game theorist’ and argued that he was ‘involved in a team that operated a highly profitable trading strategy’ and that ‘all of his actions were legal open market actions, using the protocol as designed, even if the development team [of Mango] did not fully anticipate the consequences of setting the parameters the way they were’ [498]. Later, a federal ruling agreed, and overturned all of his convictions [480]. However, he still faces civil lawsuits from the Securities Exchange Commission (SEC) [505] and Commodity Futures Trading Commission (CFTC) [132].

Other examples of DAO governance heists include Build Finance DAO, the Beanstalk DAO and Tornado Cash DAO. The attacker of Beanstalk DAO followed the same rationale as the one from Build Finance DAO, but this time leveraging a flash loan to purchase sufficient voting rights. Beanstalk DAO lost \$181 million [243]. The governance takeover in Tornado Cash similarly leveraged purchasable voting power, as well as a malicious injection of code that went undetected [454]. The latter attacker included a function in a seemingly benign proposal that was approved, which allowed them to grant themselves voting tokens. With this minted voting power, the attacker then owned majority voting power in the DAO. In this case they simply sold the governance tokens, worth over \$4 million, causing the Tornado Cash token to plummet by 40% [339]. The attacker even used Tornado Cash Tumbler to obscure the transactions of their sold tokens [340]. Table 8.2 summarises the scale and timeline of these heists.

The scale of funds seized and liquidated is sobering. The ruling of Eisenberg’s case places responsibility on Mango Market’s architecture and guarantees. The court’s opinion emphasised Mango Markets’ absence of conventional safeguards: the platform had no terms of service, no explicit prohibition on market manipulation, and no requirement that loans

Table 8.2: DAO hacks and hostile takeovers exploiting governance models.

DAO	Amount Siphoned	Year
Beanstalk	\$181m	2022
Build Finance	\$470k	2022
The DAO	\$60m	2016
Tornado Cash	\$4m	2023
Aragon DAO	\$163m	2023
Mango Markets DAO	\$110m	2023

be repaid. On this basis, the court concluded that while Eisenberg exploited weaknesses in the system, he did not engage in a materially false statement or deception sufficient to constitute fraud [498]. This highlights that DAOs bear responsibility for providing sound governance and designing mechanisms that protect participants from exploitative but technically permissible strategies.

Where does this leave DAOs? Currently, they are a dangerous endeavour for participants. Governance documentation is frequently modified or even de-listed. Participants bear the responsibility of ensuring their DAOs have obligations to protect them. Current governance is not fit for purpose: it lends itself to coercion, plutocracy and vote buying. Incentives are poorly defined and it is naive to expect participants to not follow game theory incentives to maximise their own profit. Aside from the concern of governance, many DAOs lack formal security audits. Claims of privacy, trustless platforms, censorship and coercion resistance are loudly heralded, with little to no formal proof. The e-voting and social choice theory field has long worked on formal notions of these attributes, and DAO projects have yet to embrace them. The consequences of this are not anecdotal, they are persistent. Table 8.2 only shows DAO governance heists, but exchanges, cryptocurrencies and other projects in the space have been vulnerable to attack, as shown in Appendix B.

Chapter 9

Conclusion

The entirety of work presented in this doctoral study has the motivation to counter unfair and unethical practises in the advent of the Web3 era.¹

We now live in a world where data is a commodity: our digital footprint is continuously increasing, producing data is cheap and easy, infrastructure to store it is also increasing, and yet despite this, the lengths that some service providers go to to obtain more data is unprecedented. The nuance is important: not just more data is necessary, but more labelled, high quality and clean data. This resource has become a coveted good to feed and fuel the generative model boom, the attention economy [507] and the advertising industry [564]. These unethical case studies that motivated this work continued to arise, before, during and after completion of these research contributions, evidencing that they are not anecdotal and still pressing. I believe it is imperatively necessary to continue to audit nascent technologies, not just in a scientific vacuum, but in a wider societal context.

The Web3 decentralisation revolution wielded high hopes to combat financial centralisation of power. Protocols, architectures and communities emerged to organise and demonstrate that a democratised, fairer and more private alternative is possible. This space is flourishing with technical innovation, increasing at a perplexing rate. Cryptocurrencies boomed since 2017, exchanges emerged and grew to a multi-billion industry, DAOs hold billions in assets and welcome millions of decision-making participants. Cryptographic innovations such as zero-knowledge proofs are enabling scalability and privacy by design on DLTs, and since 2018 to date (2025) at least 16 new protocols have been developed and implemented on-chain. Each of these providing improvements on proof size, verification speed and trust assumptions needed.

What is the catch? DLTs have a dark side. Promises of decentralisation and security are made that do not hold in practise against scientific formalisations. Participants are vulnerable to scams, heists, pump and dumps and coercion. Since these technologies are

¹I do not wish for this work to be a philosophical debate on ethics or fairness, this statement is merely my personal opinion on current practises outlined in previous chapters. A reader that does not consider these scenarios unethical can chose evaluate the work presented solely against scientific evaluation criteria presented.

so new, legislation is yet to catch up. As a consequence, victims can go uncompensated for their losses. Perpetrators frequently go missing or are impossible to trace due to complex webs of transaction mixers and tumblers.

I embarked on this study because I too believe that centralised service providers should not reap the entirety of the reward from a resource its participants produce. The questionable, unethical practises these service providers engaged in due to lack of auditing deeply troubled me. The time context of this study coincided with the uptake of technologies such as DLTs, DAOs and other innovations in the field, and offered an opportunity to apply these technologies to further decentralisation and fairness in a privacy preserving manner.

The contributions presented in this work are in the hopes to address these issues from a technological and scientific standpoint. In some cases however, it became evident that technological development was not the barrier to solve the problem, whereas in others, research uncovered that it is not entirely possible to solve all issues without compromise.

It is my hope that this work serves to i) dispel unverified claims of system properties made without scientific validation; ii) reinforce the argument that DLTs may be used to further the goal of social good and not just yield high returns in wealth, and iii) contribute to and open a interdisciplinary dialogue across the fields of computer science, cryptography, and social choice to further this goal.

Bibliography

- [1] @BitGo. *BitGo's statement on the Bitfinex hack*. Tweet. Accessed: October 2, 2025. 2016. URL: <https://x.com/BitGo/status/760624908334346240>.
- [2] Irum Saeed Abbasi and Laila Al-Sharqi. "Media censorship: Freedom versus responsibility". In: *Journal of Law and Conflict Resolution* 7.4 (2015), pp. 21–24.
- [3] Abbas Acar et al. "A Survey on Homomorphic Encryption Schemes: Theory and Implementation". In: *ACM Comput. Surv.* 51.4 (July 2018). ISSN: 0360-0300. DOI: [10.1145/3214303](https://doi.org/10.1145/3214303). URL: <https://doi.org/10.1145/3214303>.
- [4] Accenture and the Depository Trust & Clearing Corporation (DTCC). *Governing DLT Networks: Distributed Ledger Technology Governance for Private Permissioned Networks*. White Paper. June 2019. URL: <https://www.accenture.com/content/dam/accenture/final/a-com-migration/pdf/pdf-2/Accenture-Governing-DLT-Networks.pdf>.
- [5] Samrat Acharya et al. "False Data Injection Attacks on Data Markets for Electric Vehicle Charging Stations". In: *Advances in Applied Energy* (2022), p. 100098. ISSN: 2666-7924. DOI: <https://doi.org/10.1016/j.adapen.2022.100098>. URL: <https://www.sciencedirect.com/science/article/pii/S2666792422000166>.
- [6] ACLED. *ACLED Codebook Documentation*. Tech. rep. Accessed: September 4, 2025. ACLED, 2025. URL: <https://web.archive.org/web/20250817052144/https://acleddata.com/methodology/acled-codebook>.
- [7] Hayden Adams et al. "Uniswap v3 core". In: *Tech. rep., Uniswap, Tech. Rep.* (2021).
- [8] Ben Adida. "Helios: Web-based Open-Audit Voting." In: *USENIX security symposium*. Vol. 17. 2008, pp. 335–348.
- [9] Anish Agarwal, Munther Dahleh, and Tuhin Sarkar. "A marketplace for data: An algorithmic solution". In: *Proceedings of the 2019 ACM Conference on Economics and Computation*. 2019, pp. 701–726.
- [10] Morteza Aghaee et al. "Interferometric single-shot parity measurement in InAs–Al hybrid devices". In: *Nature* 638.8051 (Feb. 2025), pp. 651–655. ISSN: 1476-4687. DOI: [10.1038/s41586-024-08445-2](https://doi.org/10.1038/s41586-024-08445-2). URL: <https://doi.org/10.1038/s41586-024-08445-2>.
- [11] Mamunur Rashid Akand et al. "Privacy-Preserving Proof-of-Location With Security Against Geo-Tampering". In: *IEEE Transactions on Dependable and Secure Computing* (2021).
- [12] Zuhaib Akhtar. "From Blockchain to Hashgraph: Distributed Ledger Technologies in the Wild". In: *2019 International Conference on Electrical, Electronics and Computer Engineering (UPCON)*. 2019, pp. 1–6. DOI: [10.1109/UPCON47278.2019.8980029](https://doi.org/10.1109/UPCON47278.2019.8980029).
- [13] Hosam Alamleh and Ali Abdullah S AlQahtani. "A cheat-proof system to validate gps location data". In: *2020 IEEE International Conference on Electro Information Technology (EIT)*. IEEE. 2020, pp. 190–193.
- [14] C. Alexander and M. Dakos. "A critical investigation of cryptocurrency data and analysis". In: *Quantitative Finance* 20.2 (2020), pp. 173–188. DOI: [10.1080/14697688.2019.1641347](https://doi.org/10.1080/14697688.2019.1641347).

- eprint: <https://doi.org/10.1080/14697688.2019.1641347>. URL: <https://doi.org/10.1080/14697688.2019.1641347>.
- [15] Algorand Foundation. *Update on next Governance voting session*. 2022. URL: <https://web.archive.org/web/20220429194150/https://algorand.foundation/news/governance-voting-update-g3>.
 - [16] Ammar Alkassar, Thomas Nicolay, and Markus Rohe. “Obtaining True-Random Binary Numbers from a Weak Radioactive Source”. In: *Computational Science and Its Applications – ICCSA 2005*. Ed. by Osvaldo Gervasi et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 634–646. ISBN: 978-3-540-32044-9.
 - [17] Ian Allison. “Divisions in Sam Bankman-Fried’s Crypto Empire Blur on His Trading Titan Alameda’s Balance Sheet”. In: *CoinDesk* (2022). Accessed: October 2, 2025. URL: <https://web.archive.org/web/20250424011235/https://www.coindesk.com/business/2022/11/02/divisions-in-sam-bankman-frieds-crypto-empire-blur-on-his-trading-titan-alamedas-balance-sheet>.
 - [18] Ala Altaweel, Hena Mukkath, and Ibrahim Kamel. “GPS Spoofing Attacks in FANETs: A Systematic Literature Review”. In: *IEEE Access* 11 (2023), pp. 55233–55280. DOI: [10.1109/ACCESS.2023.3281731](https://doi.org/10.1109/ACCESS.2023.3281731).
 - [19] Fernando Alvarez, David Argente, and Diana Van Patten. “Are cryptocurrencies currencies? Bitcoin as legal tender in El Salvador”. In: *Science* 382.6677 (2023), eadd2844. DOI: [10.1126/science.add2844](https://doi.org/10.1126/science.add2844). eprint: <https://www.science.org/doi/pdf/10.1126/science.add2844>. URL: <https://www.science.org/doi/abs/10.1126/science.add2844>.
 - [20] Saba Ameer et al. “Comparative analysis of machine learning techniques for predicting air quality in smart cities”. In: *IEEE Access* 7 (2019), pp. 128325–128338.
 - [21] Mostafa Amin-Naseri et al. “Evaluating the Reliability, Coverage, and Added Value of Crowdsourced Traffic Incident Reports from Waze”. In: *Transportation Research Record* 2672.43 (2018), pp. 34–43. DOI: [10.1177/0361198118790619](https://doi.org/10.1177/0361198118790619). eprint: <https://doi.org/10.1177/0361198118790619>. URL: <https://doi.org/10.1177/0361198118790619>.
 - [22] Michele Amoretti et al. “Blockchain-based proof of location”. In: *2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*. IEEE, 2018, pp. 146–153.
 - [23] Matthew Amy et al. “Estimating the Cost of Generic Quantum Pre-image Attacks on SHA-2 and SHA-3”. In: *Selected Areas in Cryptography – SAC 2016*. Ed. by Roberto Avanzi and Howard Heys. Cham: Springer International Publishing, 2017, pp. 317–337. ISBN: 978-3-319-69453-5.
 - [24] David A Anderson. “Freedom of the Press in Wartime”. In: *U. Colo. L. Rev.* 77 (2006), p. 49.
 - [25] Elli Androulaki et al. “Evaluating user privacy in bitcoin”. In: *Financial Cryptography and Data Security: 17th International Conference, FC 2013, Okinawa, Japan, April 1-5, 2013, Revised Selected Papers 17*. Springer, 2013, pp. 34–51.
 - [26] Marcin Andrychowicz et al. “On the malleability of bitcoin transactions”. In: *Financial Cryptography and Data Security: FC 2015 International Workshops, BITCOIN, WAHC, and Wearable, San Juan, Puerto Rico, January 30, 2015, Revised Selected Papers*. Springer, 2015, pp. 1–18.
 - [27] Apple. *User Privacy and Data Use – App Store Developer Documentation*. <https://developer.apple.com/app-store/user-privacy-and-data-use/>. Accessed: August 8, 2025. 2021.
 - [28] Aragon Association. *Aragon Repurposes DAO to Ensure Treasury Serves Its Mission*. <https://web.archive.org/web/20250804090439/https://blog.aragon.org/aragon->

- [repurposes-dao-to-ensure-treasury-serves-its-mission/](#). Accessed: September 24, 2025. May 2023.
- [29] Gaurav Arora. “Cryptoasset Regulatory Framework in Japan”. In: *Available at SSRN 3720230* (2020).
 - [30] Kenneth J Arrow. “A difficulty in the concept of social welfare”. ger. In: *The Journal of political economy* 58.4 (1950), pp. 328–346. ISSN: 0022-3808.
 - [31] Kenneth J Arrow. “A difficulty in the concept of social welfare”. In: *Journal of political economy* 58.4 (1950), pp. 328–346.
 - [32] Frank Arute et al. “Quantum supremacy using a programmable superconducting processor”. In: *Nature* 574.7779 (Oct. 2019), pp. 505–510. ISSN: 1476-4687. DOI: [10.1038/s41586-019-1666-5](#). URL: <https://doi.org/10.1038/s41586-019-1666-5>.
 - [33] Atlantic Council. *Central Bank Digital Currency Tracker*. Accessed: September 26, 2025. 2025. URL: <https://web.archive.org/web/20250203053930/https://www.atlanticcouncil.org/cbdctracker/>.
 - [34] Attorney General of British Columbia. *Overview Report - QuadrigaCX Final*. Accessed: October 2, 2025. 2020. URL: <https://web.archive.org/web/20211015235737/https://ag-pssg-sharedservices-ex.objectstore.gov.bc.ca/ag-pssg-cc-exh-prod-bkt-ex/246%5C%20-%5C%20vreview%5C%20Report%5C%20-%5C%20Quadriga%5C%20CX%5C%20Final.pdf>.
 - [35] Daniel Augot et al. *Zero-Knowledge: trust and privacy on an industrial scale*. 2022. URL: <https://inria.hal.science/hal-03512005/document>.
 - [36] James Austgen et al. *DAO Decentralization: Voting-Bloc Entropy, Bribery, and Dark DAOs*. 2023. arXiv: [2311.03530 \[cs.CR\]](#).
 - [37] James Austgen et al. *Liquefaction: Privately Liquefying Blockchain Assets*. 2024. arXiv: [2412.02634 \[cs.CR\]](#). URL: <https://arxiv.org/abs/2412.02634>.
 - [38] Zeta Avarikioti et al. *Bribe & Fork: Cheap Bribing Attacks via Forking Threat*. 2024. arXiv: [2402.01363 \[cs.CR\]](#). URL: <https://arxiv.org/abs/2402.01363>.
 - [39] Ferheen Ayaz et al. “A voting blockchain based message dissemination in vehicular ad-hoc networks (VANETs)”. In: *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE. 2020, pp. 1–6.
 - [40] Adam Back. *Hashcash-a denial of service counter-measure*. Accessed: June 6, 2025. 2002. URL: <http://www.hashcash.org/papers/hashcash.pdf>.
 - [41] Sophie Bade. “Random serial dictatorship: the one and only”. In: *Mathematics of Operations Research* 45.1 (2020), pp. 353–368.
 - [42] Sophie Bade. “Serial dictatorship: The unique optimal allocation rule when information is endogenous”. In: *Theoretical Economics* 10.2 (2015), pp. 385–410.
 - [43] Hajar Baghcheband, Carlos Soares, and Luis Paulo Reis. “Machine Learning Data Market Based on Multiagent Systems”. In: *IEEE Internet Computing* 28.4 (2024), pp. 7–13. DOI: [10.1109/MIC.2024.3399049](#).
 - [44] Hajar Baghcheband, Carlos Soares, and Luis Paulo Reis. “Shapley-Based Data Valuation Method for the Machine Learning Data Markets (MLDM)”. In: *Foundations of Intelligent Systems: 27th International Symposium, ISMIS 2024, Poitiers, France, June 17–19, 2024, Proceedings*. Poitiers, France: Springer-Verlag, 2024, pp. 170–177. ISBN: 978-3-031-62699-9. DOI: [10.1007/978-3-031-62700-2_16](#). URL: https://doi.org/10.1007/978-3-031-62700-2_16.
 - [45] Leemon Baird and Atul Luykx. “The Hashgraph Protocol: Efficient Asynchronous BFT for High-Throughput Distributed Ledgers”. In: *2020 International Conference on Omni-layer Intelligent Systems (COINS)*. 2020, pp. 1–7. DOI: [10.1109/COINS49042.2020.9191430](#).

- [46] Shaimaa Bajoudah, Changyu Dong, and Paolo Missier. “Toward a decentralized, trust-less marketplace for brokered IoT data trading using blockchain”. In: *2019 IEEE international conference on blockchain (Blockchain)*. IEEE. 2019, pp. 339–346.
- [47] Boaz Barak and Shai Halevi. “A model and architecture for pseudo-random generation with applications to /dev/random”. In: *Proceedings of the 12th ACM Conference on Computer and Communications Security*. CCS ’05. Alexandria, VA, USA: Association for Computing Machinery, 2005, pp. 203–212. ISBN: 1595932267. DOI: [10.1145/1102120.1102148](https://doi.org/10.1145/1102120.1102148). URL: <https://doi.org/10.1145/1102120.1102148>.
- [48] Tom Barbereau et al. “Decentralised Finance’s Unregulated Governance: Minority Rule in the Digital Wild West”. In: *Available at SSRN* (2022).
- [49] Elaine Barker and Quynh Dang. *NIST Special Publication 800-57 Part 3, Revision 1: Recommendation for Key Management — Part 3: Application-Specific Key Management Guidance*. NIST Special Publication 800-57 Pt.3, Rev. 1. National Institute of Standards and Technology, Jan. 2015. DOI: [10.6028/NIST.SP.800-57pt3r1](https://doi.org/10.6028/NIST.SP.800-57pt3r1). URL: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57Pt3r1.pdf>.
- [50] Elaine B. Barker and John M. Kelsey. *Recommendation for Random Number Generation Using Deterministic Random Bit Generators*. Tech. rep. NIST SP 800-90A Revision 1. National Institute of Standards and Technology, 2015, pp. 3–6. URL: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf>.
- [51] Julian E. Barnes, Eric Schmitt, and David D. Kirkpatrick. “‘Tell Your Boss’: Recording Is Seen to Link Saudi Crown Prince More Strongly to Khashoggi Killing”. In: *NY Times* (2018). Accessed: September 5, 2025. URL: <https://web.archive.org/web/20250617173530/https://www.nytimes.com/2018/11/12/world/middleeast/jamal-khashoggi-killing-saudi-arabia.html>.
- [52] Paul Barnes. “Crypto currency and its susceptibility to speculative bubbles, manipulation, scams and fraud”. In: *Journal of Advanced Studies in Finance (JASF)* 9.2 (18) (2018), pp. 60–77.
- [53] Brian Barrett. “An Artist Used 99 Phones to Fake a Google Maps Traffic Jam”. In: *Wired* (2020). Accessed: September 3, 2025. URL: <https://web.archive.org/web/20250318141332/https://www.wired.com/story/99-phones-fake-google-maps-traffic-jam/>.
- [54] Jamie Bartlett. *Cryptoqueen: How this woman scammed the world, then vanished*. Accessed: 2024-12-19. 2019. URL: <https://www.bbc.com/news/stories-50435014>.
- [55] Michele Battagliola et al. *Algebraic Construction for Multi-Party Protocols with Focus on Threshold Signatures*. 2024.
- [56] Ammar Battah et al. “Blockchain and NFTs for trusted ownership, trading, and access of AI models”. In: *IEEE Access* 10 (2022), pp. 112230–112249.
- [57] Ray Beaulieu et al. “The SIMON and SPECK lightweight block ciphers”. In: *Proceedings of the 52nd Annual Design Automation Conference*. DAC ’15. San Francisco, California: Association for Computing Machinery, 2015. ISBN: 9781450335201. DOI: [10.1145/2744769.2747946](https://doi.org/10.1145/2744769.2747946). URL: <https://doi.org/10.1145/2744769.2747946>.
- [58] Roman Beck, Christoph Müller-Bloch, and John Leslie King. “Governance in the blockchain economy: A framework and research agenda”. In: *Journal of the association for information systems* 19.10 (2018), p. 1.
- [59] BehindMLM. *Plus Token Wallet Review: Mobile crypto wallet Ponzi scheme*. Accessed: October 2, 2025. 2018. URL: <https://web.archive.org/web/20250620050056/https://behindmlm.com/mlm-reviews/plus-token-wallet-review-mobile-crypto-wallet-ponzi-scheme/>.

- [60] Rob Behnke. “Explained: The Africrypt “Hack””. In: *Halborn* (2021). Accessed: October 2, 2025. URL: <https://web.archive.org/web/20250519022524/https://www.halborn.com/blog/post/explained-the-africrypt-hack>.
- [61] Rob Behnke. “Explained: The Ronin Hack (March 2022)”. In: *Halborn* (2022). Accessed: October 2, 2025. URL: <https://web.archive.org/web/20250810043544/https://www.halborn.com/blog/post/explained-the-ronin-hack-march-2022>.
- [62] Rob Behnke. *Explained: The US Government - Bitfinex Hack (October 2024)*. Accessed: September 26, 2025. Nov. 2024. URL: <https://web.archive.org/web/20250427121712/https://www.halborn.com/blog/post/explained-the-us-government-bitfinex-hack-october-2024>.
- [63] Alireza Beikverdi and JooSeok Song. “Trend of centralization in Bitcoin’s distributed network”. In: *2015 IEEE/ACIS 16th international conference on software engineering, artificial intelligence, networking and parallel/distributed computing (SNPD)*. IEEE, 2015, pp. 1–6.
- [64] Russell Belk, Mariam Humayun, and Myriam Brouard. “Money, possessions, and ownership in the Metaverse: NFTs, cryptocurrencies, Web3 and Wild Markets”. In: *Journal of Business Research* 153 (2022), pp. 198–205. ISSN: 0148-2963. DOI: <https://doi.org/10.1016/j.jbusres.2022.08.031>. URL: <https://www.sciencedirect.com/science/article/pii/S0148296322007147>.
- [65] Frank Bell et al. “Data Sharing, Data Exchanges, and the Snowflake Data Marketplace”. In: *Snowflake Essentials*. Springer, 2022, pp. 299–328.
- [66] Cristiano Bellavitis, Christian Fisch, and Paul P Momtaz. “The rise of decentralized autonomous organizations (DAOs): a first empirical glimpse”. In: *Available at SSRN* (2022).
- [67] Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. *Interactive Oracle Proofs*. Cryptology ePrint Archive, Paper 2016/116. 2016. URL: <https://eprint.iacr.org/2016/116>.
- [68] Eli Ben-Sasson et al. *Scalable, transparent, and post-quantum secure computational integrity*. Cryptology ePrint Archive, Paper 2018/046. 2018. URL: <https://eprint.iacr.org/2018/046>.
- [69] Josh Benaloh and Dwight Tuinstra. “Receipt-free secret-ballot elections”. In: *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*. 1994, pp. 544–553.
- [70] Josh Benaloh et al. “End-to-end verifiability”. In: *arXiv preprint arXiv:1504.03778* (2015).
- [71] Josh Cohen Benaloh. “Secret Sharing Homomorphisms: Keeping Shares of a Secret Secret (Extended Abstract)”. In: *Advances in Cryptology — CRYPTO’ 86*. Ed. by Andrew M. Odlyzko. Berlin, Heidelberg: Springer Berlin Heidelberg, 1987, pp. 251–260. ISBN: 978-3-540-47721-1.
- [72] Gayan Benedict. “Challenges of DLT-enabled scalable governance and the role of standards”. In: *Journal of ICT Standardization* (2019), pp. 195–208.
- [73] Charles H. Bennett et al. “Strengths and Weaknesses of Quantum Computing”. In: *SIAM Journal on Computing* 26.5 (1997), pp. 1510–1523. DOI: [10.1137/S0097539796300933](https://doi.org/10.1137/S0097539796300933). eprint: <https://doi.org/10.1137/S0097539796300933>. URL: <https://doi.org/10.1137/S0097539796300933>.
- [74] Iddo Bentov, Rafael Pass, and Elaine Shi. “Snow White: Provably Secure Proofs of Stake”. In: *IACR Cryptol. ePrint Arch.* 2016.919 (2016).
- [75] Hal Berghel. “Malice domestic: The Cambridge Analytica dystopia”. In: *Computer* 51.05 (2018), pp. 84–89.
- [76] Daniel J Bernstein. “Cost analysis of hash collisions: Will quantum computers make SHARCS obsolete”. In: *SHARCS 9* (2009), p. 105.

- [77] Daniel J. Bernstein. “Post-quantum Cryptography”. In: *Encyclopedia of Cryptography, Security and Privacy*. Ed. by Sushil Jajodia, Pierangela Samarati, and Moti Yung. Cham: Springer Nature Switzerland, 2025, pp. 1846–1847. ISBN: 978-3-030-71522-9. DOI: [10.1007/978-3-030-71522-9_386](https://doi.org/10.1007/978-3-030-71522-9_386). URL: https://doi.org/10.1007/978-3-030-71522-9_386.
- [78] Daniel J. Bernstein. “The Salsa20 Family of Stream Ciphers”. In: *New Stream Cipher Designs: The eSTREAM Finalists*. Ed. by Matthew Robshaw and Olivier Billet. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 84–97. ISBN: 978-3-540-68351-3. DOI: [10.1007/978-3-540-68351-3_8](https://doi.org/10.1007/978-3-540-68351-3_8). URL: https://doi.org/10.1007/978-3-540-68351-3_8.
- [79] Ward Beullens. “Breaking rainbow takes a weekend on a laptop”. In: *Annual International Cryptology Conference*. Springer, 2022, pp. 464–479.
- [80] Aditya Bhaskara et al. “Data Exchange Markets via Utility Balancing”. In: *Proceedings of the ACM Web Conference 2024*. WWW ’24. Singapore, Singapore: Association for Computing Machinery, 2024, pp. 57–65. ISBN: 9798400701719. DOI: [10.1145/3589334.3645364](https://doi.org/10.1145/3589334.3645364). URL: <https://doi.org/10.1145/3589334.3645364>.
- [81] Dave Birnbaum. “Crypto Hot Wallets vs. Cold Wallets: The Key Differences”. In: *Forbes* (2024). Accessed: October 2, 2025. URL: <https://web.archive.org/web/20250426112603/https://www.forbes.com/sites/digital-assets/article/crypto-hot-wallets-vs-cold-wallets/>.
- [82] Alex Biryukov and Ivan Pustogarov. “Bitcoin over Tor isn’t a good idea”. In: *2015 IEEE Symposium on Security and Privacy*. IEEE, 2015, pp. 122–134.
- [83] Nir Bitansky et al. “From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again”. In: *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*. ITCS ’12. Cambridge, Massachusetts: Association for Computing Machinery, 2012, pp. 326–349. ISBN: 9781450311151. DOI: [10.1145/2090236.2090263](https://doi.org/10.1145/2090236.2090263). URL: <https://doi.org/10.1145/2090236.2090263>.
- [84] André Blais and Louis Massicotte. “Electoral systems”. In: *Comparing democracies 2* (1996), pp. 40–69.
- [85] Grant Blank and Bianca C. Reisdorf. “The Participatory Web”. In: *Information, Communication & Society* 15.4 (2012), pp. 537–554. DOI: [10.1080/1369118X.2012.665935](https://doi.org/10.1080/1369118X.2012.665935). eprint: <https://doi.org/10.1080/1369118X.2012.665935>. URL: <https://doi.org/10.1080/1369118X.2012.665935>.
- [86] Felipe Boeira, Mikael Asplund, and Marinho Barcellos. “Decentralized proof of location in vehicular ad hoc networks”. In: *Computer Communications* 147 (2019), pp. 98–110.
- [87] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. “Evaluating 2-DNF Formulas on Ciphertexts”. In: *Theory of Cryptography*. Ed. by Joe Kilian. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 325–341. ISBN: 978-3-540-30576-7.
- [88] Dan Boneh, Ben Lynn, and Hovav Shacham. “Short Signatures from the Weil Pairing”. In: *Journal of Cryptology* 17.4 (Sept. 2004), pp. 297–319. ISSN: 1432-1378. DOI: [10.1007/s00145-004-0314-9](https://doi.org/10.1007/s00145-004-0314-9). URL: <https://doi.org/10.1007/s00145-004-0314-9>.
- [89] Dan Boneh et al. *Zero Knowledge Proofs*. <https://rdi.berkeley.edu/zk-learning/>. Accessed: July 16, 2025. Berkley, 2023. URL: <https://rdi.berkeley.edu/zk-learning/>.
- [90] Michele Bonini et al. “Proof of Location through a Blockchain Agnostic Smart Contract Language”. In: *2023 IEEE 43rd International Conference on Distributed Computing Systems Workshops (ICDCSW)*. 2023, pp. 55–60. DOI: [10.1109/ICDCSW60045.2023.00016](https://doi.org/10.1109/ICDCSW60045.2023.00016).
- [91] Lorenz Bornholdt, Julian Reher, and Volker Skwarek. “Proof-of-Location: A method for securing sensor-data-communication in a Byzantine fault tolerant way”. In: *Mobile Communication - Technologies and Applications; 24. ITG-Symposium*. 2019, pp. 1–6.

- [92] Joppe Bos et al. “CRYSTALS-Kyber: a CCA-secure module-lattice-based KEM”. In: *2018 IEEE European symposium on security and privacy (EuroS&P)*. IEEE. 2018, pp. 353–367.
- [93] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. “(Leveled) Fully Homomorphic Encryption without Bootstrapping”. In: *ACM Trans. Comput. Theory* 6.3 (July 2014). ISSN: 1942-3454. DOI: [10.1145/2633600](https://doi.org/10.1145/2633600). URL: <https://doi.org/10.1145/2633600>.
- [94] Arnaud Braud et al. “The road to European digital sovereignty with Gaia-X and IDSA”. In: *IEEE network* 35.2 (2021), pp. 4–5.
- [95] Antonio Briola et al. “Anatomy of a Stablecoin’s failure: The Terra-Luna case”. In: *Finance Research Letters* 51 (2023), p. 103358.
- [96] Rachel Brodsky. “The Bizarre True Story Of ‘Bitcoin’s Bonnie & Clyde’ in Netflix’s Biggest Heist Ever”. In: *TIME* (2025). Accessed: October 1, 2025. URL: <https://web.archive.org/web/20250330081001/https://time.com/7200219/bitcoin-netflix-biggest-heist-ever/>.
- [97] Daniel R. L. Brown. “Generic Groups, Collision Resistance, and ECDSA”. In: *Designs, Codes and Cryptography* 35.1 (Apr. 2005), pp. 119–152. ISSN: 1573-7586. DOI: [10.1007/s10623-003-6154-z](https://doi.org/10.1007/s10623-003-6154-z). URL: <https://doi.org/10.1007/s10623-003-6154-z>.
- [98] Lars Brünjes and Murdoch J. Gabbay. “UTxO- vs Account-Based Smart Contract Blockchain Programming Paradigms”. In: *Leveraging Applications of Formal Methods, Verification and Validation: Applications*. Ed. by Tiziana Margaria and Bernhard Steffen. Cham: Springer International Publishing, 2020, pp. 73–88. ISBN: 978-3-030-61467-6.
- [99] William J Buchanan. *Security and So Many Things*. <https://asecuritysite.com/>. Accessed: July 16, 2025. Asecuritysite.com, 2025. URL: <https://asecuritysite.com/>.
- [100] Ethan Buchman. “Tendermint: Byzantine fault tolerance in the age of blockchains”. PhD thesis. University of Guelph, 2016.
- [101] Benedikt Bünz et al. “Bulletproofs: Short Proofs for Confidential Transactions and More”. In: *2018 IEEE Symposium on Security and Privacy (SP)*. May 2018, pp. 315–334. DOI: [10.1109/SP.2018.00020](https://doi.org/10.1109/SP.2018.00020).
- [102] Barry C Burden. “Deterministic and probabilistic voting models”. In: *American Journal of Political Science* (1997), pp. 1150–1169.
- [103] *Business Wire: News Release*. Accessed: October 2, 2025. Oct. 2018. URL: <https://web.archive.org/web/20181025054454/https://www.businesswire.com/news/home/20181024006083/ja/>.
- [104] Vitalik Buterin. *Notes on Blockchain Governance*. 2017. URL: <https://web.archive.org/web/20240413233016/https://vitalik.eth.limo/general/2017/12/17/voting.html>.
- [105] Ran Canetti, Oded Goldreich, and Shai Halevi. “The random oracle methodology, revisited”. In: *J. ACM* 51.4 (July 2004), pp. 557–594. ISSN: 0004-5411. DOI: [10.1145/1008731.1008734](https://doi.org/10.1145/1008731.1008734). URL: <https://doi.org/10.1145/1008731.1008734>.
- [106] Agostino Capponi, Sveinn Ólafsson, and Humoud Alsabah. “Proof-of-Work Cryptocurrencies: Does Mining Technology Undermine Decentralization?” In: *Manage. Sci.* 69.11 (Nov. 2023), pp. 6455–6481. ISSN: 0025-1909. DOI: [10.1287/mnsc.2023.4840](https://doi.org/10.1287/mnsc.2023.4840). URL: <https://doi.org/10.1287/mnsc.2023.4840>.
- [107] Miguel Castro, Barbara Liskov, et al. “Practical byzantine fault tolerance”. In: *OsDI*. Vol. 99. 1999. 1999, pp. 173–186.
- [108] Miguel Castro and Barbara Liskov. “Practical Byzantine fault tolerance and proactive recovery”. In: *ACM Transactions on Computer Systems (TOCS)* 20.4 (2002), pp. 398–461.
- [109] Miguel Castro and Barbara Liskov. “Practical byzantine fault tolerance and proactive recovery”. In: *ACM Transactions on Computer Systems (TOCS)* 20.4 (2002), pp. 398–461.

- [110] Wouter Castryck and Thomas Decru. “An Efficient Key Recovery Attack on SIDH”. In: *Advances in Cryptology – EUROCRYPT 2023*. Ed. by Carmit Hazay and Martijn Stam. Cham: Springer Nature Switzerland, 2023, pp. 423–447. ISBN: 978-3-031-30589-4.
- [111] Chainalysis. *\$197 Million Stolen: Euler Finance Flash Loan Attack Explained*. 2023. URL: <https://www.chainalysis.com/blog/euler-finance-flash-loan-attack/>.
- [112] Chainalysis. *The Chainalysis State of Web3 Report*. 2022. URL: <https://web.archive.org/web/20250404215010/https://go.chainalysis.com/rs/503-FAP-074/images/state-of-web3.pdf>.
- [113] Edwin Chan. “Microsoft helped NSA, FBI access user info: Guardian”. In: *Reuters* (July 2013). Accessed: August 8, 2025. URL: <https://www.reuters.com/article/us-usa-cybersecurity-microsoft-idUSBRE96A11R20130711/>.
- [114] Delaware Court of Chancery. *Memorandum Opinion Imposing Sanctions For Spoliation*. 2025. URL: <https://courts.delaware.gov/Opinions/Download.aspx?id=374350>.
- [115] Natalia Chaudhry and Muhammad Murtaza Yousaf. “Consensus Algorithms in Blockchain: Comparative Analysis, Challenges and Opportunities”. In: *2018 12th International Conference on Open Source Systems and Technologies (ICOSST)*. 2018, pp. 54–63. DOI: [10.1109/ICOSST.2018.8632190](https://doi.org/10.1109/ICOSST.2018.8632190).
- [116] David Chaum. “Blind Signatures for Untraceable Payments”. In: *Advances in Cryptology*. Ed. by David Chaum, Ronald L. Rivest, and Alan T. Sherman. Boston, MA: Springer US, 1983, pp. 199–203. ISBN: 978-1-4757-0602-4.
- [117] David Chaum et al. “Scantegrity: End-to-end voter-verifiable optical-scan voting”. In: *IEEE Security & Privacy* 6.3 (2008), pp. 40–46.
- [118] David L Chaum. “Untraceable electronic mail, return addresses, and digital pseudonyms”. In: *Communications of the ACM* 24.2 (1981), pp. 84–90.
- [119] Jung Hee Cheon et al. “Homomorphic Encryption for Arithmetic of Approximate Numbers”. In: *Advances in Cryptology – ASIACRYPT 2017*. Ed. by Tsuyoshi Takagi and Thomas Peyrin. Cham: Springer International Publishing, 2017, pp. 409–437. ISBN: 978-3-319-70694-8.
- [120] Alessandro Chiesa et al. “Marlin: Preprocessing zkSNARKs with Universal and Updatable SRS”. In: *Advances in Cryptology – EUROCRYPT 2020*. Ed. by Anne Canteaut and Yuval Ishai. Cham: Springer International Publishing, 2020, pp. 738–768. ISBN: 978-3-030-45721-1.
- [121] Ilaria Chillotti et al. “TFHE: Fast Fully Homomorphic Encryption Over the Torus”. In: *Journal of Cryptology* 33.1 (Jan. 2020), pp. 34–91. ISSN: 1432-1378. DOI: [10.1007/s00145-019-09319-x](https://doi.org/10.1007/s00145-019-09319-x). URL: <https://doi.org/10.1007/s00145-019-09319-x>.
- [122] Sedona Chinn, Daniel S Lane, and Philip S Hart. “In consensus we trust? Persuasive effects of scientific consensus communication”. In: *Public Understanding of Science* 27.7 (2018), pp. 807–823.
- [123] Konstantinos Christidis and Michael Devetsikiotis. “Blockchains and Smart Contracts for the Internet of Things”. In: *IEEE Access* 4 (2016), pp. 2292–2303. DOI: [10.1109/ACCESS.2016.2566339](https://doi.org/10.1109/ACCESS.2016.2566339).
- [124] Nicolas Christin. “Traveling the silk road: a measurement analysis of a large anonymous online marketplace”. In: *Proceedings of the 22nd International Conference on World Wide Web*. WWW ’13. Rio de Janeiro, Brazil: Association for Computing Machinery, 2013, pp. 213–224. ISBN: 9781450320351. DOI: [10.1145/2488388.2488408](https://doi.org/10.1145/2488388.2488408). URL: <https://doi.org/10.1145/2488388.2488408>.
- [125] Wolfie Christl, Katharina Kopp, and Patrick U Riechert. “How companies use personal data against people”. In: *Automated Disadvantage, Personalized Persuasion, and the Societal*

- Ramifications of the Commercial Use of Personal Information*. Wien: Cracked Labs 21 (2017). Accessed: August 6, 2025. URL: https://crackedlabs.org/dl/CrackedLabs_Christ1_DataAgainstPeople.pdf.
- [126] Shumo Chu and Sophia Wang. *The Curses of Blockchain Decentralization*. 2018. arXiv: 1810.02937 [cs.CR]. URL: <https://arxiv.org/abs/1810.02937>.
 - [127] Michael R Clarkson, Stephen Chong, and Andrew C Myers. “Civitas: Toward a secure voting system”. In: *2008 IEEE Symposium on Security and Privacy (sp 2008)*. IEEE. 2008, pp. 354–368.
 - [128] Coin Circle Observation. *Shilian Finance News: News Release*. Accessed: October 2, 2025. 2019. URL: <https://m.shilian.com/zixun/114617.html>.
 - [129] Alistair Coleman. “Ukraine conflict: Further false images shared online”. In: *British Broadcasting Corporation (BBC)* (2022). Accessed: September 4, 2025. URL: <https://www.bbc.com/news/60528276>.
 - [130] Keith Collins. “Here’s what your stolen identity goes for on the internet’s black market”. In: *Quartz* (2015). URL: <https://web.archive.org/web/20230127124419/https://qz.com/460482/heres-what-your-stolen-identity-goes-for-on-the-internets-black-market/>.
 - [131] European Commission. *SEDIMARK: SEcure Decentralised Intelligent Data MARketplace*. <https://cordis.europa.eu/project/id/101070074>. Grant Agreement No. 101070074. 2025.
 - [132] Commodity Futures Trading Commission. *CFTC Charges Avraham Eisenberg with Manipulative and Deceptive Scheme to Misappropriate Over \$110 million from Mango Markets*. <https://web.archive.org/web/20250422034408/https://www.cftc.gov/PressRoom/PressReleases/8647-23>. Accessed: September 24, 2025. Jan. 2023.
 - [133] Bitcoin Community. *Bitcoin Improvement Proposals*. <https://web.archive.org/web/20250128080254/https://github.com/bitcoin/bips>. Accessed: September 18, 2025. 2011.
 - [134] Thomas H Cormen et al. *Introduction to algorithms*. MIT press, 2022.
 - [135] Intel Corporation. *Intel Digital Random Number Generator (DRNG) Software Implementation Guide*. Technical Report 2.1. Accessed: September 27, 2025. Intel, 2018. URL: <https://web.archive.org/web/20250719025824/https://www.intel.com/content/dam/develop/external/us/en/documents/441-intel-r-drng-software-implementation-guide-final-aug7.pdf>.
 - [136] Véronique Cortier et al. “SoK: Verifiability Notions for E-Voting Protocols”. In: *2016 IEEE Symposium on Security and Privacy (SP)*. 2016, pp. 779–798. DOI: [10.1109/SP.2016.52](https://doi.org/10.1109/SP.2016.52).
 - [137] Sofia Cossar et al. *Proof of Humanity: Ethnographic Research of a “Democratic” DAO*. Tech. rep. Technical Report; 55 pages. European University Institute, Robert Schuman Centre, Apr. 2024. DOI: [10.2870/107946](https://doi.org/10.2870/107946). URL: <https://hdl.handle.net/1814/76793>.
 - [138] John C Courtney. “Plurality-Majority electoral systems: A review”. In: *Electoral Insight* 1.1 (1999), pp. 7–11.
 - [139] Tyler Crain et al. “DBFT: Efficient Leaderless Byzantine Consensus and its Application to Blockchains”. In: *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)*. 2018, pp. 1–8. DOI: [10.1109/NCA.2018.8548057](https://doi.org/10.1109/NCA.2018.8548057).
 - [140] Judd Cramer and Alan B. Krueger. “Disruptive Change in the Taxi Business: The Case of Uber”. In: *American Economic Review* 106.5 (May 2016), pp. 177–82. DOI: [10.1257/aer.p20161002](https://doi.org/10.1257/aer.p20161002). URL: <https://www.aeaweb.org/articles?id=10.1257/aer.p20161002>.
 - [141] Lane Crothers and Grace Burgener. “Insurrectionary Populism? Assessing the January 6 Attack on the U.S. Capitol”. In: *Populism* 4.2 (2021), pp. 129–145. DOI: [10.1163/](https://doi.org/10.1163/)

- 25888072-bja10025. URL: https://brill.com/view/journals/popu/4/2/article-p129_1.xml.
- [142] Luis Cuende and Jorge Izquierdo. *Aragon Network A Decentralized Infrastructure for Value Exchange*. 2017. URL: <https://web.archive.org/web/20210730083452/https://academy.bit2me.com/wp-content/uploads/2021/05/aragon-whitepaper.pdf>.
 - [143] Wei Dai. *B-money*. Accessed: June 6, 2025. 1998. URL: <http://www.weidai.com/bmoney.txt>.
 - [144] Phil Daian. “Analysis of the DAO Exploit”. In: *Hacking Distributed* (2016). Accessed: October 2, 2025. URL: <https://web.archive.org/web/20190214110957/https://hackingdistributed.com/2016/06/18/analysis-of-the-dao-exploit/>.
 - [145] Philip Daian et al. “Flash Boys 2.0: Frontrunning in Decentralized Exchanges, Miner Extractable Value, and Consensus Instability”. In: *2020 IEEE Symposium on Security and Privacy (SP)*. 2020, pp. 910–927. DOI: [10.1109/SP40000.2020.00040](https://doi.org/10.1109/SP40000.2020.00040).
 - [146] Philip Daian et al. “On-chain vote buying and the rise of Dark DAOs”. In: *Hacking, Distributed* (2018). URL: <https://web.archive.org/web/20180705080040/http://hackingdistributed.com/2018/07/02/on-chain-vote-buying/>.
 - [147] Ivan Damgård. “Commitment schemes and zero-knowledge protocols”. In: *School organized by the European Educational Forum*. Springer. 1998, pp. 63–86.
 - [148] Sheila Dang. “Snap shares plunge 25% as Apple privacy changes hit ads business”. In: *Reuters* (Oct. 2021). Accessed: August 8, 2025. URL: <https://www.reuters.com/technology/snap-revenue-falls-short-apple-privacy-changes-hurt-ads-business-2021-10-21/>.
 - [149] Sinclair Davidson. “Corporate governance in a crypto-world”. In: *Available at SSRN 4986409* (2024).
 - [150] Milind Dawande, Sameer Mehta, and Liying Mu. “Robin Hood to the Rescue: Sustainable Revenue-Allocation Schemes for Data Cooperatives”. In: *Production and operations management*. 32.8 (2023). ISSN: 1059-1478.
 - [151] Primavera De Filippi and Benjamin Loveluck. “The invisible politics of bitcoin: governance crisis of a decentralized infrastructure”. In: *Internet policy review* 5.4 (2016).
 - [152] Primavera De Filippi and Greg McMullen. “Governance of blockchain systems: Governance of and by Distributed Infrastructure”. PhD thesis. Blockchain Research Institute and COALA, 2018.
 - [153] Leonardo De Moura and Nikolaj Bjørner. “Z3: An efficient SMT solver”. In: *International conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer. 2008, pp. 337–340.
 - [154] Christian Decker, Jochen Seidel, and Roger Wattenhofer. “Bitcoin meets strong consistency”. In: *Proceedings of the 17th International Conference on Distributed Computing and Networking*. 2016, pp. 1–10.
 - [155] Christian Decker and Roger Wattenhofer. “Bitcoin Transaction Malleability and MtGox”. In: *Computer Security - ESORICS 2014*. Ed. by Mirosław Kutylowski and Jaideep Vaidya. Cham: Springer International Publishing, 2014, pp. 313–326. ISBN: 978-3-319-11212-1.
 - [156] M. DeCross et al. “Computational Power of Random Quantum Circuits in Arbitrary Geometries”. In: *Physical Review X* 15.2 (May 2025). ISSN: 2160-3308. DOI: [10.1103/PhysRevX.15.021052](https://doi.org/10.1103/PhysRevX.15.021052). URL: <http://dx.doi.org/10.1103/PhysRevX.15.021052>.
 - [157] DeepDAO. *DAO Organizations*. 2023. URL: <https://deepdao.io/organizations>.
 - [158] Sergi Delgado-Segura et al. “Analysis of the Bitcoin UTXO Set”. In: *Financial Cryptography and Data Security*. Ed. by Aviv Zohar et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2019, pp. 78–91. ISBN: 978-3-662-58820-8.

- [159] Anirudh Dhawan and Tālis J Putniņš. “A new wolf in town? Pump-and-dump manipulation in cryptocurrency markets”. In: *Review of Finance* 27.3 (2023), pp. 935–975.
- [160] Vikram Dhillon et al. “The DAO hacked”. In: *blockchain enabled applications: Understand the blockchain Ecosystem and How to Make it work for you* (2017), pp. 67–78.
- [161] Tim Dierks and Eric Rescorla. *The transport layer security (TLS) protocol version 1.2*. Tech. rep. IETF, 2008.
- [162] W. Diffie and M. Hellman. “New directions in cryptography”. In: *IEEE Transactions on Information Theory* 22.6 (1976), pp. 644–654. DOI: [10.1109/TIT.1976.1055638](https://doi.org/10.1109/TIT.1976.1055638).
- [163] José van Dijck, Thomas Poell, and Martijn de Waal. *The Platform Society*. Oxford University Press, Nov. 2018. ISBN: 9780190889760. DOI: [10.1093/oso/9780190889760.001.0001](https://doi.org/10.1093/oso/9780190889760.001.0001). URL: <https://doi.org/10.1093/oso/9780190889760.001.0001>.
- [164] Ariel Dobkin. “Information Fiduciaries in Practice: Data Privacy and User Expectations”. In: *Berkeley Technology Law Journal* 33.1 (2018). Accessed: August 6, 2025., pp. 1–52. ISSN: 10863818, 23804742. URL: <https://www.jstor.org/stable/26490153>.
- [165] David L Donoho and Peter J Huber. “The notion of breakdown point”. In: *A festschrift for Erich L. Lehmann* 157184 (1983), pp. 157–184.
- [166] John R Douceur. “The sybil attack”. In: *International workshop on peer-to-peer systems*. Springer. 2002, pp. 251–260.
- [167] DreamDAO. *The Dream DAO Merit System*. 2022. URL: <https://web.archive.org/web/20221223071951/https://dream-dao.gitbook.io/how-we-dao-dream-dao/merit-system>.
- [168] Amit Dua and Gaurav Kumar. *Mastering Zero-knowledge Proofs: Practical study of security, scalability, and privacy in blockchain and modern systems (English Edition)*. BPB Publications, 2024.
- [169] Dune. *Ethereum ETH Staking Distribution*. <https://dune.com/hildobby/eth2-staking>. Accessed: 2025-05-20.
- [170] Paul Dunphy, Luke Garratt, and Fabien Petitcolas. “Decentralizing digital identity: Open challenges for distributed ledgers”. In: *2018 IEEE European symposium on security and privacy workshops (EuroS&PW)*. IEEE. 2018, pp. 75–78.
- [171] Erika Duriakova et al. “A decentralised architecture for secure exchange of assets in data spaces: The case of SEDIMARK”. In: *Data in Brief* 61 (2025), p. 111757. ISSN: 2352-3409. DOI: <https://doi.org/10.1016/j.dib.2025.111757>. URL: <https://www.sciencedirect.com/science/article/pii/S2352340925004846>.
- [172] Cynthia Dwork and Aaron Roth. “The Algorithmic Foundations of Differential Privacy”. In: *Foundations and Trends® in Theoretical Computer Science* 9.3–4 (2014), pp. 211–407. ISSN: 1551-305X. DOI: [10.1561/04000000042](https://doi.org/10.1561/04000000042). URL: <http://dx.doi.org/10.1561/04000000042>.
- [173] Salam Al-E’mari and Yousef Sanjalawe. “A Review of Reentrancy Attack in Ethereum Smart Contracts”. In: *International Conference on Computing and Communication Networks*. Springer. 2023, pp. 53–70.
- [174] Kristine Eck. “In data we trust? A comparison of UCEDP GED and ACLED conflict events datasets”. In: *Cooperation and Conflict* 47.1 (2012), pp. 124–141. DOI: [10.1177/0010836711434463](https://doi.org/10.1177/0010836711434463). eprint: <https://doi.org/10.1177/0010836711434463>. URL: <https://doi.org/10.1177/0010836711434463>.
- [175] Malcolm Egan and Nir Oren. *Package Exchange Mechanisms with Quality Constraints for Data Markets*. Accessed: August 23, 2025. 2024. URL: <https://hal.science/hal-04536373/document>.

- [176] Parinya Ekparinya, Vincent Gramoli, and Guillaume Jourjon. “Impact of Man-In-The-Middle Attacks on Ethereum”. In: *2018 IEEE 37th Symposium on Reliable Distributed Systems (SRDS)*. 2018, pp. 11–20. DOI: [10.1109/SRDS.2018.00012](https://doi.org/10.1109/SRDS.2018.00012).
- [177] T. Elgamal. “A public key cryptosystem and a signature scheme based on discrete logarithms”. In: *IEEE Transactions on Information Theory* 31.4 (1985), pp. 469–472. DOI: [10.1109/TIT.1985.1057074](https://doi.org/10.1109/TIT.1985.1057074).
- [178] Elliptic Intel. “North Korea’s Lazarus Group identified as exploiters behind \$540 million Ronin bridge heist”. In: *Elliptic* (2022). Accessed: October 2, 2025. URL: <https://web.archive.org/web/20250402144634/https://www.elliptic.co/blog/540-million-stolen-from-the-ronin-defi-bridge>.
- [179] Equidem. *SCROLL. CLICK. SUFFER. The Hidden Human Cost of Content Moderation and Data Labelling*. Tech. rep. Accessed: August 8, 2025. Equidem.org, 2025. URL: <https://equidem.org/wp-content/uploads/2025/05/Equidem-Data-Workers-Report-May-29.pdf>.
- [180] Sinem Coleri Ergen. “ZigBee/IEEE 802.15. 4 Summary”. In: *UC Berkeley, September* 10.17 (2004), p. 11.
- [181] Ana Faguy. “Trump launches cryptocurrency with price rocketing”. In: *British Broadcasting Corporation (BBC)* (Jan. 2025). Accessed: September 26, 2025. URL: <https://www.bbc.com/news/articles/c9vmym2jvy9o>.
- [182] Benedikt Fauseweh. “Quantum many-body simulations on digital quantum computers: State-of-the-art and future challenges”. In: *Nature Communications* 15.1 (Apr. 2024), p. 2123. ISSN: 2041-1723. DOI: [10.1038/s41467-024-46402-9](https://doi.org/10.1038/s41467-024-46402-9). URL: <https://doi.org/10.1038/s41467-024-46402-9>.
- [183] Federal Trade Commission. *Complaint: In the Matter of Cambridge Analytica, LLC*. Accessed: August 6, 2025. Washington, D.C., July 24, 2019. URL: https://www.ftc.gov/system/files/documents/cases/182_3107_cambridge_analytica_administrative_complaint_7-24-19.pdf.
- [184] Rainer Feichtinger et al. “SoK: Attacks on DAOs”. In: *arXiv preprint arXiv:2406.15071* (2024).
- [185] Horst Feistel. “Cryptography and Computer Privacy”. In: *Scientific American* 228.5 (1973), pp. 15–23.
- [186] Paul Feldman and Silvio Micali. “Optimal algorithms for byzantine agreement”. In: *Proceedings of the twentieth annual ACM symposium on Theory of computing*. 1988, pp. 148–161.
- [187] Luca De Feo, David Jao, and Jérôme Plût. “Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies”. In: *Journal of Mathematical Cryptology* 8.3 (2014), pp. 209–247. DOI: [doi:10.1515/jmc-2012-0015](https://doi.org/10.1515/jmc-2012-0015). URL: <https://doi.org/10.1515/jmc-2012-0015>.
- [188] Niels Ferguson et al. “The skein hash function family”. In: *Submission to NIST (round 3)* 7.7.5 (2010), p. 3.
- [189] Amos Fiat and Adi Shamir. “How To Prove Yourself: Practical Solutions to Identification and Signature Problems”. In: *Advances in Cryptology — CRYPTO’ 86*. Ed. by Andrew M. Odlyzko. Berlin, Heidelberg: Springer Berlin Heidelberg, 1987, pp. 186–194. ISBN: 978-3-540-47721-1.
- [190] Financial Crisis Inquiry Commission. *The Financial Crisis Inquiry Report: Final Report of the National Commission on the Causes of the Financial and Economic Crisis in the United States*. Tech. rep. Accessed: June 2, 2025. Washington, D.C.: U.S. Government

- Printing Office, 2011. URL: https://fcic-static.law.stanford.edu/cdn_media/fcic-reports/fcic_final_report_full.pdf.
- [191] Forbes. *Crypto Prices - Forbes Digital Assets*. Accessed: 2024-12-20. 2024. URL: <https://web.archive.org/web/20250708012104/https://www.forbes.com/digital-assets/crypto-prices/>.
 - [192] Avalanche Foundation. *Avalanche Governance*. <https://docs.avax.network/learn/platform-overview/governance>. Accessed: December 19, 2024. 2019.
 - [193] Cardano Foundation. *Introduction to Cardano Protocol Governance*. <https://web.archive.org/web/20250715142133/https://developers.cardano.org/docs/governance/cardano-governance/governance-model/>. Accessed: September 18, 2025. 2024.
 - [194] Ethereum Foundation. *Ethereum Governance: The Formal Process*. <https://web.archive.org/web/20250313045656/https://ethereum.org/en/governance/#formal-process>. Accessed: December 19, 2024. 2017.
 - [195] Ethereum Foundation. *Ethereum Improvement Proposals*. <https://web.archive.org/web/20250328084838/https://eips.ethereum.org/>. Accessed: September 18, 2025. 2015.
 - [196] IOTA Foundation. *IOTA Rebased Governance Proposal*. <https://web.archive.org/web/20250127133036/https://govern.iota.org/t/iota-rebased-governance-proposal/1768>. Accessed: September 18, 2025. 2018.
 - [197] Ocean Protocol Foundation. *Ocean Protocol: Tools for the Web3 Data Economy*. 2020. URL: <https://oceanprotocol.com/tech-whitepaper.pdf>.
 - [198] Solana Foundation. *Solana Governance Overview*. <https://web.archive.org/web/20250514023311/https://docs.solana.com/cluster/governance>. Accessed: September 18, 2025. 2018.
 - [199] Pierre-Alain Fouque et al. “Falcon: Fast-Fourier lattice-based compact signatures over NTRU”. In: *Submission to the NIST’s post-quantum cryptography standardization process* 36.5 (2018), pp. 1–75.
 - [200] Tore Kasper Frederiksen et al. “On the complexity of additively homomorphic UC commitments”. In: *Theory of Cryptography Conference*. Springer. 2015, pp. 542–565.
 - [201] Yuki Furukawa. “Former Bitcoin Baron Mark Karpeles Gets Suspended Jail Term”. In: *Bloomberg* (2019). Accessed: October 2, 2025. URL: <https://web.archive.org/web/20200215153902/https://www.bloomberg.com/news/articles/2019-03-15/former-bitcoin-baron-mark-karpeles-gets-suspended-jail-term>.
 - [202] Ariel Gabizon, Zachary J. Williamson, and Oana Ciobotaru. *PLONK: Permutations over Lagrange-bases for Oecumenical Noninteractive arguments of Knowledge*. Cryptology ePrint Archive, Paper 2019/953. 2019. URL: <https://eprint.iacr.org/2019/953>.
 - [203] Tommaso Gagliardoni. *The Poly Network Hack Explained*. Accessed: October 2, 2025. 2021. URL: <https://web.archive.org/web/20230228203531/https://research.kudelskisecurity.com/2021/08/12/the-poly-network-hack-explained/>.
 - [204] Adityawardhan Gaikwad and Sushil Mavale. “The impact of cryptocurrency adoption as a legal tender in El salvador”. In: *International Journal of Engineering and Management Research* 11.6 (2021), pp. 112–115.
 - [205] Jay Gambetta. *The hardware and software for the era of quantum utility is here*. <https://web.archive.org/web/20250721223840/https://www.ibm.com/quantum/blog/quantum-roadmap-2033>. Accessed: August 1, 2025. 2023.
 - [206] Andrea Gangemi and Aida Manzano Kharman. “Private Dispute Resolution on Ethereum”. In: *2024 IEEE International Conference on Blockchain (Blockchain)*. 2024, pp. 183–192. DOI: [10.1109/Blockchain62396.2024.00032](https://doi.org/10.1109/Blockchain62396.2024.00032).

- [207] Andrea Gangemi and Aida Manzano Kharman. “Private Dispute Resolution on Ethereum”. In: *2024 IEEE International Conference on Blockchain (Blockchain)*. 2024, pp. 183–192. DOI: [10.1109/Blockchain62396.2024.00032](https://doi.org/10.1109/Blockchain62396.2024.00032).
- [208] Pedro Garcia Lopez, Alberto Montresor, and Anwitaman Datta. “Please, do not Decentralize the Internet with (Permissionless) Blockchains!” In: *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. 2019, pp. 1901–1911. DOI: [10.1109/ICDCS.2019.00188](https://doi.org/10.1109/ICDCS.2019.00188).
- [209] Fasica B. Gebrekidan. “Content moderation: The harrowing, traumatizing job that left many African data workers with mental health issues and drug dependency”. In: *The Data Workers’ Inquiry* (2024). Accessed: August 8, 2025. URL: <https://data-workers.org/wp-content/uploads/2024/09/Fasicas-Report-Accessible-1.pdf>.
- [210] Adem Efe Gencer et al. “Decentralization in bitcoin and ethereum networks”. In: *Financial Cryptography and Data Security: 22nd International Conference, FC 2018, Nieuwpoort, Curaçao, February 26–March 2, 2018, Revised Selected Papers 22*. Springer. 2018, pp. 439–457.
- [211] Craig Gentry. “Fully homomorphic encryption using ideal lattices”. In: *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*. STOC ’09. Bethesda, MD, USA: Association for Computing Machinery, 2009, pp. 169–178. ISBN: 9781605585062. DOI: [10.1145/1536414.1536440](https://doi.org/10.1145/1536414.1536440). URL: <https://doi.org/10.1145/1536414.1536440>.
- [212] Craig Gentry, Amit Sahai, and Brent Waters. “Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based”. In: *Advances in Cryptology – CRYPTO 2013*. Ed. by Ran Canetti and Juan A. Garay. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 75–92. ISBN: 978-3-642-40041-4.
- [213] Allan Gibbard. “Manipulation of schemes that mix voting with chance”. In: *Econometrica: Journal of the Econometric Society* (1977), pp. 665–681.
- [214] Allan Gibbard. “Manipulation of voting schemes: a general result”. In: *Econometrica: journal of the Econometric Society* (1973), pp. 587–601.
- [215] Yossi Gilad et al. “Algorand: Scaling byzantine agreements for cryptocurrencies”. In: *Proceedings of the 26th symposium on operating systems principles*. 2017, pp. 51–68.
- [216] George Gilder. *Life after Google: The fall of big data and the rise of the blockchain economy*. Simon and Schuster, 2018.
- [217] Nicolas Gisin et al. “Quantum cryptography”. In: *Rev. Mod. Phys.* 74 (1 Apr. 2002), pp. 145–195. DOI: [10.1103/RevModPhys.74.145](https://doi.org/10.1103/RevModPhys.74.145). URL: <https://link.aps.org/doi/10.1103/RevModPhys.74.145>.
- [218] Efthimios-Enias Gojka et al. “Security in distributed ledger technology: An analysis of vulnerabilities and attack vectors”. In: *Intelligent Computing: Proceedings of the 2021 Computing Conference, Volume 3*. Springer. 2021, pp. 722–742.
- [219] Oded Goldreich and Yair Oren. “Definitions and properties of zero-knowledge proof systems”. In: *Journal of Cryptology* 7.1 (1994), pp. 1–32.
- [220] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. “The Knowledge Complexity of Interactive Proof Systems”. In: *SIAM Journal on Computing* 18.1 (1989), pp. 186–208. DOI: [10.1137/0218012](https://doi.org/10.1137/0218012). eprint: <https://doi.org/10.1137/0218012>. URL: <https://doi.org/10.1137/0218012>.
- [221] Carla Goncalves, Pierre Pinson, and Ricardo J Bessa. “Towards data markets in renewable energy forecasting”. In: *IEEE Transactions on Sustainable Energy* 12.1 (2020), pp. 533–542.
- [222] Víctor González et al. “On the use of Blockchain to enable a highly scalable Internet of Things Data Marketplace”. In: *Internet of Things 22* (2023), p. 100722.

- [223] Geoff Goodell and Tomaso Aste. “A decentralized digital identity architecture”. In: *Frontiers in Blockchain* 2 (2019), p. 491305.
- [224] Mt. Gox. *Press Release: June 30, 2011*. Accessed: October 2, 2025. 2011. URL: https://web.archive.org/web/20110919162635/https://mtgox.com/press_release_20110630.html.
- [225] Robert M. Gray. “Entropy”. In: *Entropy and Information Theory*. Boston, MA: Springer US, 2011, pp. 61–95. ISBN: 978-1-4419-7970-4. DOI: [10.1007/978-1-4419-7970-4_3](https://doi.org/10.1007/978-1-4419-7970-4_3). URL: https://doi.org/10.1007/978-1-4419-7970-4_3.
- [226] Andy Greenberg. “A ‘Blockchain Bandit’ Is Guessing Private Keys and Scoring Millions”. In: *WIRED* (2019). URL: <https://web.archive.org/web/20250207005658/https://www.wired.com/story/blockchain-bandit-ethereum-weak-private-keys/>.
- [227] Andy Greenberg. “NSA’s Verizon Spying Order Specifically Targeted Americans, Not Foreigners”. In: *Forbes* (June 2013). Accessed: August 8, 2025. URL: <https://www.forbes.com/sites/andygreenberg/2013/06/05/nsas-verizon-spying-order-specifically-targeted-americans-not-foreigners/>.
- [228] David Gritten et al. “Iran: How we are uncovering the protests and crackdowns”. In: *British Broadcasting Corporation (BBC)* (2022). Accessed: September 4, 2025. URL: <https://web.archive.org/web/20250422110235/https://www.bbc.com/news/world-middle-east-63264159>.
- [229] Jens Groth. “On the Size of Pairing-Based Non-interactive Arguments”. In: *Advances in Cryptology – EUROCRYPT 2016*. Ed. by Marc Fischlin and Jean-Sébastien Coron. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 305–326. ISBN: 978-3-662-49896-5.
- [230] Lov K. Grover. “Quantum Mechanics Helps in Searching for a Needle in a Haystack”. In: *Phys. Rev. Lett.* 79 (2 July 1997), pp. 325–328. DOI: [10.1103/PhysRevLett.79.325](https://link.aps.org/doi/10.1103/PhysRevLett.79.325). URL: <https://link.aps.org/doi/10.1103/PhysRevLett.79.325>.
- [231] Hong Guo et al. “Truly random number generation based on measurement of phase noise of a laser”. In: *Phys. Rev. E* 81 (5 May 2010), p. 051137. DOI: [10.1103/PhysRevE.81.051137](https://link.aps.org/doi/10.1103/PhysRevE.81.051137). URL: <https://link.aps.org/doi/10.1103/PhysRevE.81.051137>.
- [232] Peter Gutmann and Stephan Neuhaus. *Replication of Quantum Factorisation Records with an 8-bit Home Computer, an Abacus, and a Dog*. Cryptology ePrint Archive, Paper 2025/1237. 2025. URL: <https://eprint.iacr.org/2025/1237>.
- [233] Robert Hackett and Declan Harty. “Apple’s ad changes wiped \$142 billion off Snap, Facebook, and other online ad giants”. In: *Fortune* (Oct. 2021). Accessed: August 8, 2025. URL: <https://fortune.com/2021/10/22/apple-snap-facebook-earnings-google-twitter-pinterest-ad-tracking/>.
- [234] Yassine Hamoudi, Qipeng Liu, and Makrand Sinha. “The NISQ Complexity of Collision Finding”. In: *Advances in Cryptology – EUROCRYPT 2024*. Ed. by Marc Joye and Gregor Leander. Cham: Springer Nature Switzerland, 2024, pp. 3–32. ISBN: 978-3-031-58737-5.
- [235] David J. Hand. “Aspects of Data Ethics in a Changing World: Where Are We Now?” In: *Big Data* 6.3 (2018). PMID: 30283727, pp. 176–190. DOI: [10.1089/big.2018.0083](https://doi.org/10.1089/big.2018.0083). eprint: <https://doi.org/10.1089/big.2018.0083>. URL: <https://doi.org/10.1089/big.2018.0083>.
- [236] Colin Harper. *Bitcoin Independence Day: How This Watershed Day Defines Community Consensus*. Accessed: 2024-12-19. 2019. URL: <https://bitcoinmagazine.com/culture/bitcoin-independence-day-how-this-watershed-day-defines-community-consensus>.
- [237] Huru Hasanova et al. “A survey on blockchain cybersecurity vulnerabilities and possible countermeasures”. In: *International Journal of Network Management* 29.2 (2019), e2060.

- [238] Aumyo Hassan and Sarah J Barber. “The effects of repetition frequency on the illusory truth effect”. en. In: *Cogn Res Princ Implic* 6.1 (May 2021), p. 38.
- [239] Samer Hassan and Primavera De Filippi. “The expansion of algorithmic governance: from code is law to law is code”. In: *Field Actions Science Reports. The journal of field actions* Special Issue 17 (2017), pp. 88–90.
- [240] Helmut Hasse. “Zur Theorie der abstrakten elliptischen Funktionenkörper III. Die Struktur des Meromorphismenrings. Die Riemannsche Vermutung.” In: *Journal für die reine und angewandte Mathematik* 1936.175 (1936), pp. 193–208. DOI: [doi:10.1515/crll.1936.175.193](https://doi.org/10.1515/crll.1936.175.193). URL: <https://doi.org/10.1515/crll.1936.175.193>.
- [241] Zhiguo He, Jiasun Li, and Zhengxun Wu. “Don’t trust, verify: The case of slashing from a popular ethereum explorer”. In: *Companion Proceedings of the ACM Web Conference 2023*. 2023, pp. 1078–1084.
- [242] Ethan Heilman et al. “Eclipse Attacks on Bitcoin’s Peer-to-Peer Network”. In: *24th USENIX Security Symposium (USENIX Security 15)*. Washington, D.C.: USENIX Association, Aug. 2015, pp. 129–144. ISBN: 978-1-939133-11-3. URL: <https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/heilman>.
- [243] Alex Hern. *Beanstalk cryptocurrency loses \$182m of reserves in flash ‘attack’*. 2022. URL: <https://web.archive.org/web/20250721235258/https://www.theguardian.com/technology/2022/apr/18/beanstalk-cryptocurrency-loses-182m-of-reserves-in-flash-attack>.
- [244] Jack D Hidary. *Quantum computing: an applied approach*. Vol. 1. Springer, 2021.
- [245] High Court of Justice, Business and Property Courts of England and Wales, Intellectual Property (ChD). *Getty Images (v) Stability AI Ltd, Neutral Citation: [2025] EWHC 38 (Ch)*. Accessed: August 8, 2025. Jan. 2025. URL: <https://www.judiciary.uk/wp-content/uploads/2025/01/Getty-Images-and-others-v-Stability-AI-14.01.25.pdf>.
- [246] House of Commons Standing Committee on Access to Information, Privacy and Ethics (ETHI). *Evidence of Standing Committee on Access to Information, Privacy and Ethics – Study of the Breach of Personal Information Involving Cambridge Analytica and Facebook*. Committee Evidence. Ottawa, Canada: House of Commons of Canada, 42nd Parliament, 1st Session, Number 101, Apr. 19, 2018. URL: <https://www.ourcommons.ca/Content/Committee/421/ETHI/Evidence/EV9771706/ETHIEV99-E.PDF>.
- [247] Yih-Chun Hu, A. Perrig, and D.B. Johnson. “Wormhole attacks in wireless networks”. In: *IEEE Journal on Selected Areas in Communications* 24.2 (2006), pp. 370–380. DOI: [10.1109/JSAC.2005.861394](https://doi.org/10.1109/JSAC.2005.861394).
- [248] Yih-Chun Hu, Adrian Perrig, and David B Johnson. “Wormhole attacks in wireless networks”. In: *IEEE journal on selected areas in communications* 24.2 (2006), pp. 370–380.
- [249] Scott D Hughes. “Cryptocurrency Regulations and Enforcement in the US”. In: *W. St. UL Rev.* 45 (2017), p. 1.
- [250] Thomas E Hull and Alan R Dobell. “Random number generators”. In: *SIAM review* 4.3 (1962), pp. 230–254.
- [251] Mark Hunter. *What Was the Bitcoin New York Agreement?* 2022. URL: <https://fullycrypto.com/what-was-the-bitcoin-new-york-agreement>.
- [252] Ziad Hussein, May A. Salama, and Sahar A. El-Rahman. “Evolution of blockchain consensus algorithms: a review on the latest milestones of blockchain consensus algorithms”. In: *Cybersecurity* 6.1 (Nov. 2023), p. 30. ISSN: 2523-3246. DOI: [10.1186/s42400-023-00163-y](https://doi.org/10.1186/s42400-023-00163-y). URL: <https://doi.org/10.1186/s42400-023-00163-y>.

- [253] Nick Hynes et al. “A demonstration of sterling: a privacy-preserving data marketplace”. In: *Proceedings of the VLDB Endowment* 11.12 (2018), pp. 2086–2089.
- [254] Independent Security Evaluators. *Ethercombing: Finding Secrets in Popular Places*. ISE. 2019. URL: <https://web.archive.org/web/20250305213148/https://www.ise.io/casestudies/ethercombing/>.
- [255] Information Commissioner’s Office. *Decision Notice: FS50808172 – Freedom of Information Act 2000*. Decision Notice. Wilmslow, UK, June 11, 2019. URL: <https://ico.org.uk/media2/migrated/decision-notices/2615143/fs50808172.pdf>.
- [256] IonQ, Inc. *IonQ Quantum Computers*. <https://web.archive.org/web/20250716043331/https://ionq.com/quantum-systems/compare>. Accessed: August 1, 2025. 2025.
- [257] <https://web.archive.org/web/20250207174759/https://govern.iota.org/c/governance-proposals/51>. Accessed: December 19, 2024.
- [258] IOTA Foundation. *Governance Voting Explained*. <https://wiki.iota.org/learn/governance/governance-tech/>. Accessed: December 19, 2024. 2023.
- [259] IPFS. *Snapshot IPFS Documentation*. 2022. URL: <https://web.archive.org/web/20230304193250/https://docs.ipfs.tech/case-studies/snapshot/>.
- [260] Ali Jafarnia-Jahromi et al. “GPS vulnerability to spoofing threats and a review of anti-spoofing techniques”. In: *International Journal of Navigation and Observation* 2012.1 (2012), p. 127072.
- [261] Archana Jain, Chinmay Jain, and Karolina Krystyniak. “Blockchain transaction fee and Ethereum Merge”. In: *Finance Research Letters* 58 (2023), p. 104507. ISSN: 1544-6123. DOI: <https://doi.org/10.1016/j.frl.2023.104507>. URL: <https://www.sciencedirect.com/science/article/pii/S1544612323008796>.
- [262] Roger Jardí-Cedó et al. “Study on poll-site voting and verification systems”. In: *Computers & Security* 31.8 (2012), pp. 989–1010.
- [263] Craig Jarvis. “Cypherpunk ideology: objectives, profiles, and influences (1992–1998)”. In: *Internet Histories* 6.3 (2022), pp. 315–342. DOI: [10.1080/24701475.2021.1935547](https://doi.org/10.1080/24701475.2021.1935547). eprint: <https://doi.org/10.1080/24701475.2021.1935547>. URL: <https://doi.org/10.1080/24701475.2021.1935547>.
- [264] Chitra Javali et al. “I am alice, i was in wonderland: secure location proof generation and verification protocol”. In: *2016 IEEE 41st conference on local computer networks (LCN)*. IEEE. 2016, pp. 477–485.
- [265] Staff. Al Jazeera. “What is Project Nimbus, and why are Google workers protesting Israel deal?” In: *Al Jazeera* (Apr. 2024). Accessed: August 8, 2025. URL: <https://www.aljazeera.com/news/2024/4/23/what-is-project-nimbus-and-why-are-google-workers-protesting-israel-deal>.
- [266] S. Josefsson and I. Liusvaara. *Edwards-Curve Digital Signature Algorithm (EdDSA)*. RFC 8032. Internet Engineering Task Force, Jan. 2017. URL: <https://www.rfc-editor.org/rfc/rfc8032>.
- [267] Aljosha Judmayer et al. “Pay-To-Win: Incentive Attacks on Proof-of-Work Cryptocurrencies.” In: *IACR Cryptol. ePrint Arch.* 2019 (2019), p. 775.
- [268] Ari Juels, Dario Catalano, and Markus Jakobsson. “Coercion-resistant electronic elections”. In: *Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society*. 2005, pp. 61–70.
- [269] United States Department of Justice. *U.S. v. Konstantin Ignatov Complaint*. Accessed: October 2, 2025. 2019. URL: <https://web.archive.org/web/20250203041737/https://www.justice.gov/usao-sdny/press-release/file/1141986/dl>.

- [270] United States Department of Justice Office of Public Affairs. *Individual Arrested and Charged with Operating Notorious Darknet Cryptocurrency “Mixer”*. <https://www.justice.gov/archives/opa/pr/individual-arrested-and-charged-operating-notorious-darknet-cryptocurrency-mixer>. Press Release No. 21-390. Apr. 2021.
- [271] Brittany Kaiser. *Written evidence of Brittany Kaiser to the Digital, Culture, Media and Sport Select Committee inquiry on Disinformation and “fake news”*. UK Parliament. Apr. 2018. URL: <https://www.parliament.uk/globalassets/documents/commons-committees/culture-media-and-sport/brittany-kaiser-parliamentary-testimony-final.pdf>.
- [272] Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg. “Constant-Size Commitments to Polynomials and Their Applications”. In: *Advances in Cryptology - ASIACRYPT 2010*. Ed. by Masayuki Abe. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 177–194. ISBN: 978-3-642-17373-8.
- [273] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography*. 2nd ed. Chapman & Hall/CRC, 2015. ISBN: 978-1-4665-7027-6. DOI: [10.5555/2700550](https://doi.org/10.5555/2700550). URL: https://eclass.uniwa.gr/modules/document/file.php/CSCYB105/Reading%20Material/%5BJonathan_Katz%2CYehuda_Lindell%5D_Introduction_to_Mo%282nd%29.pdf.
- [274] Liam Kelly. *Build Finance DAO Falls to Governance Takeover*. 2022. URL: <https://web.archive.org/web/20250329175614/https://decrypt.co/92970/build-finance-dao-falls-to-governance-takeover>.
- [275] Nida Khan et al. “Blockchain Governance: An Overview and Prediction of Optimal Strategies using Nash Equilibrium”. In: *arXiv preprint arXiv:2003.09241* (2020).
- [276] Aida Manzano Kharman and Ben Smyth. “DAO Governance Protocols and their Vulnerabilities”. In: *2024 6th Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*. 2024, pp. 1–2. DOI: [10.1109/BRAINS63024.2024.10732536](https://doi.org/10.1109/BRAINS63024.2024.10732536).
- [277] Aida Manzano Kharman and Ben Smyth. “DAO Governance Protocols and their Vulnerabilities”. In: *2024 6th Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*. IEEE. 2024, pp. 1–2.
- [278] Aida Manzano Kharman and Ben Smyth. *Perils of current DAO governance*. 2024. arXiv: [2406.08605](https://arxiv.org/abs/2406.08605) [cs.CY]. URL: <https://arxiv.org/abs/2406.08605>.
- [279] Aida Manzano Kharman et al. “An adversarially robust data-market for spatial, crowd-sourced data”. In: *arXiv preprint arXiv:2206.06299* (2022).
- [280] Aida Manzano Kharman et al. “An adversarially robust data-market for spatial, crowd-sourced data”. In: *Distrib. Ledger Technol.* (Nov. 2024). Just Accepted. DOI: [10.1145/3703464](https://doi.org/10.1145/3703464). URL: <https://doi.org/10.1145/3703464>.
- [281] Aida Manzano Kharman et al. “Robust Decentralised Proof-of-Position Algorithms for Smart City Applications”. In: *2023 62nd IEEE Conference on Decision and Control (CDC)*. 2023, pp. 112–119. DOI: [10.1109/CDC49753.2023.10384054](https://doi.org/10.1109/CDC49753.2023.10384054).
- [282] Aida Manzano Kharman et al. “Robust decentralised proof-of-position algorithms for smart city applications”. In: *2023 62nd IEEE Conference on Decision and Control (CDC)*. IEEE. 2023, pp. 112–119.
- [283] Ali M Al-Khouri et al. “Data ownership: who owns “my data””. In: *International Journal of Management & Information Technology* 2.1 (2012), pp. 1–8.
- [284] Aggelos Kiayias and Philip Lazos. “SoK: blockchain governance”. In: *Proceedings of the 4th ACM Conference on Advances in Financial Technologies*. 2022, pp. 61–73.
- [285] Aggelos Kiayias et al. “Ouroboros: A provably secure proof-of-stake blockchain protocol”. In: *Annual international cryptology conference*. Springer. 2017, pp. 357–388.

- [286] Lucianna Kiffer, Dave Levin, and Alan Mislove. “Stick a fork in it: Analyzing the ethereum network partition”. In: *Proceedings of the 16th ACM Workshop on Hot Topics in Networks*. 2017, pp. 94–100.
- [287] Hyung-Sin Kim, Sam Kumar, and David E Culler. “Thread/OpenThread: A compromise in low-power wireless multihop network architecture for the Internet of Things”. In: *IEEE Communications Magazine* 57.7 (2019), pp. 55–61.
- [288] Jason Kim and U.S. Department of Commerce, Bureau of Industry and Security. *Briefing on GPS/GNSS Export Controls*. Tech. rep. Presentation delivered to a working group of the National Space-Based Positioning, Navigation, and Timing (PNT) Advisory Board. U.S. Department of Commerce, Office of Space Commerce, 2022.
- [289] Kate Knibbs. “Meta Secretly Trained Its AI on a Notorious Piracy Database, Newly Unredacted Court Docs Reveal”. In: *Wired* (2025). URL: <https://www.wired.com/story/new-documents-unredacted-meta-copyright-ai-lawsuit/>.
- [290] Neal Koblitz. “Elliptic curve cryptosystems”. In: *Mathematics of computation* 48.177 (1987), pp. 203–209.
- [291] Paul C Kocher. “Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems”. In: *Annual international cryptology conference*. Springer. 1996, pp. 104–113.
- [292] Aleksandr Kogan. *Written evidence submitted by Aleksandr Kogan to the Digital Culture, Media and Sport Select Committee*. Evidence to the House of Commons *Digital, Culture, Media and Sport Select Committee*. Accessed: August 6, 2025. Apr. 2018. URL: <https://www.parliament.uk/documents/commons-committees/culture-media-and-sport/Written-evidence-Aleksandr-Kogan.pdf>.
- [293] Varun Kohli et al. “An analysis of energy consumption and carbon footprints of cryptocurrencies and possible solutions”. In: *Digital Communications and Networks* 9.1 (2023), pp. 79–89. ISSN: 2352-8648. DOI: <https://doi.org/10.1016/j.dcan.2022.06.017>. URL: <https://www.sciencedirect.com/science/article/pii/S2352864822001390>.
- [294] Olga Kosheleva, Vladik Kreinovich, and Guo Wei. “Ranking-based voting revisited: maximum entropy approach leads to Borda count (and its versions)”. In: *Behavioral Predictive Modeling in Economics* (2021), pp. 145–152.
- [295] Kristián Košťál et al. “Blockchain e-voting done right: Privacy and transparency with public blockchain”. In: *2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS)*. IEEE. 2019, pp. 592–595.
- [296] Kraken Security Labs. *Abusing Smart Contracts to Steal \$600 Million: How the Poly Network Hack Actually Happened*. Accessed: October 2, 2025. 2021. URL: <https://web.archive.org/web/20231004185549/https://blog.kraken.com/product/security/abusing-smart-contracts-to-steal-600-million-how-the-poly-network-hack-actually-happened>.
- [297] Steve Kremer, Mark Ryan, and Ben Smyth. “Election verifiability in electronic voting protocols”. In: *Computer Security—ESORICS 2010: 15th European Symposium on Research in Computer Security, Athens, Greece, September 20-22, 2010. Proceedings 15*. Springer. 2010, pp. 389–404.
- [298] Nir Kshetri and Jeffrey Voas. “Blockchain-Enabled E-Voting”. In: *IEEE Software* 35.4 (2018), pp. 95–99. DOI: [10.1109/MS.2018.2801546](https://doi.org/10.1109/MS.2018.2801546).
- [299] KuCoin. *KuCoin CEO Livestream Recap - Latest Updates About Security Incident*. Accessed: October 2, 2025. 2020. URL: <https://web.archive.org/web/20250611181105/https://www.kucoin.com/announcement/en-kucoin-ceo-livestream-recap-latest-updates-about-security-incident>.

- [300] Anikender Kumar and P Goyal. “Forecasting of daily air quality index in Delhi”. In: *Science of the Total Environment* 409.24 (2011), pp. 5517–5523.
- [301] Ralf Küsters, Tomasz Truderung, and Andreas Vogt. “Accountability: definition and relationship to verifiability”. In: *Proceedings of the 17th ACM Conference on Computer and Communications Security*. CCS ’10. Chicago, Illinois, USA: Association for Computing Machinery, 2010, pp. 526–535. ISBN: 9781450302456. DOI: [10.1145/1866307.1866366](https://doi.org/10.1145/1866307.1866366). URL: <https://doi.org/10.1145/1866307.1866366>.
- [302] Michael Kwet. “How US Big Tech supports Israel’s AI-powered genocide and apartheid”. In: *Al Jazeera* (May 2024). Accessed: August 8, 2025. URL: <https://www.aljazeera.com/opinions/2024/5/12/how-us-big-tech-supports-israels-ai-powered-genocide-and-apartheid>.
- [303] Snapshot Labs. *Snapshot Documentation*. 2023. URL: <https://web.archive.org/web/20230209223901/https://docs.snapshot.org/>.
- [304] Steven P. Lalley and E. Glen Weyl. “Quadratic Voting: How Mechanism Design Can Radicalize Democracy”. In: *AEA Papers and Proceedings* 108 (May 2018), pp. 33–37. DOI: [10.1257/pandp.20181002](https://doi.org/10.1257/pandp.20181002). URL: <https://www.aeaweb.org/articles?id=10.1257/pandp.20181002>.
- [305] Leslie Lamport. “Paxos Made Simple”. In: *ACM SIGACT News (Distributed Computing Column)* 32, 4 (Whole Number 121, December 2001) (Dec. 2001), pp. 51–58. URL: <https://www.microsoft.com/en-us/research/publication/paxos-made-simple/>.
- [306] Leslie Lamport, Robert Shostak, and Marshall Pease. “The Byzantine generals problem”. In: *Concurrency: the works of leslie lamport*. 2019, pp. 203–226.
- [307] Nikolaos Laoutaris. “Why Online Services Should Pay You for Your Data? The Arguments for a Human-Centric Data Economy”. In: *IEEE Internet Computing* 23.5 (2019), pp. 29–35. DOI: [10.1109/MIC.2019.2953764](https://doi.org/10.1109/MIC.2019.2953764).
- [308] Bahareh Lashkari and Petr Musilek. “A Comprehensive Review of Blockchain Consensus Mechanisms”. In: *IEEE Access* 9 (2021), pp. 43620–43652. DOI: [10.1109/ACCESS.2021.3065880](https://doi.org/10.1109/ACCESS.2021.3065880).
- [309] Jean-François Laslier. “And the loser is... plurality voting”. In: *Electoral systems*. Springer, 2012, pp. 327–351.
- [310] Kibin Lee et al. “Electronic voting service using block-chain”. In: *Journal of Digital Forensics, Security and Law* 11.2 (2016), p. 8.
- [311] Shaun Paul Lee. *Government Bitcoin Holdings*. Accessed: 2024-12-20. 2024. URL: <https://www.coingecko.com/research/publications/government-bitcoin-holdings>.
- [312] Sung-Shine Lee et al. “SoK: Not Quite Water Under the Bridge: Review of Cross-Chain Bridge Hacks”. In: *2023 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. 2023, pp. 1–14. DOI: [10.1109/ICBC56567.2023.10174993](https://doi.org/10.1109/ICBC56567.2023.10174993).
- [313] Vili Lehdonvirta and Robleh Ali. “Governance and regulation”. In: *Distributed Ledger Technology: beyond block chain: A report by the UK Government Chief Scientific Adviser*. Government Office for Science, 2016, pp. 40–45.
- [314] Vladimir Ilyich Lenin. “Adventurism”. In: *Collected Works*. Vol. 20. Translated quote: "Put no faith in words; subject everything to the closest scrutiny". Moscow: Progress Publishers, 1972, pp. 356–359. URL: <https://web.archive.org/web/20250312021602/https://www.marxists.org/archive/lenin/works/1914/jun/09.htm>.
- [315] Jill Lepore. “Rock, Paper, Scissors: How we used to vote”. In: *The New Yorker* (2008).
- [316] Annabelle Lever. “Mill and the secret ballot: Beyond coercion and corruption”. In: *Utilitas* 19.3 (2007), pp. 354–378.

- [317] Qinya Li et al. “Capitalize Your Data: Optimal Selling Mechanisms for IoT Data Exchange”. In: *IEEE Transactions on Mobile Computing* 22.4 (2023), pp. 1988–2000. DOI: [10.1109/TMC.2021.3113387](https://doi.org/10.1109/TMC.2021.3113387).
- [318] Wenting Li et al. “Securing Proof-of-Stake Blockchain Protocols”. In: *Data Privacy Management, Cryptocurrencies and Blockchain Technology*. Ed. by Joaquin Garcia-Alfaro et al. Cham: Springer International Publishing, 2017, pp. 297–315. ISBN: 978-3-319-67816-0.
- [319] Kevin Liao and Jonathan Katz. “Incentivizing blockchain forks via whale transactions”. In: *Financial Cryptography and Data Security: FC 2017 International Workshops, WAHC, BITCOIN, VOTING, WTSC, and TA, Sliema, Malta, April 7, 2017, Revised Selected Papers 21*. Springer. 2017, pp. 264–279.
- [320] Joshua Liebow-Feesser. *Randomness 101: LavaRand in Production*. *The Cloudflare Blog*. Nov. 2017. URL: <https://web.archive.org/web/20250613162419/https://blog.cloudflare.com/randomness-101-lavarand-in-production/>.
- [321] Jiageng Liu, Igor Makarov, and Antoinette Schoar. *Anatomy of a run: The terra luna crash*. Tech. rep. National Bureau of Economic Research, 2023.
- [322] Tianyi Liu et al. “Pianist: Scalable zkRollups via Fully Distributed Zero-Knowledge Proofs”. In: *2024 IEEE Symposium on Security and Privacy (SP)*. 2024, pp. 1777–1793. DOI: [10.1109/SP54263.2024.00035](https://doi.org/10.1109/SP54263.2024.00035).
- [323] Xing Liu, Bahar Farahani, and Farshad Firouzi. “Distributed ledger technology”. In: *Intelligent internet of things: From device to fog and cloud* (2020), pp. 393–431.
- [324] Yue Liu et al. “A systematic literature review on blockchain governance”. In: *Journal of Systems and Software* 197 (2023), p. 111576.
- [325] Ramon Lobato. *Netflix nations: The geography of digital distribution*. New York University Press, 2019.
- [326] Lostin. *Solana Governance: A Comprehensive Analysis*. <https://web.archive.org/web/20250326180623/https://www.helius.dev/blog/solana-governance--a-comprehensive-analysis>. Accessed: September 18, 2025. 2025.
- [327] Charles Lu et al. “Data Acquisition via Experimental Design for Data Markets”. In: *Advances in Neural Information Processing Systems*. Ed. by A. Globerson et al. Vol. 37. Curran Associates, Inc., 2024, pp. 118086–118118. URL: https://proceedings.neurips.cc/paper_files/paper/2024/file/d5e8326bbec25e1c608787d24488521b-Paper-Conference.pdf.
- [328] Xiaofeng Lu, Cheng Jiang, and Pan Wang. “A Survey on Consensus Algorithms of Blockchain Based on DAG”. In: *Proceedings of the 2024 6th Blockchain and Internet of Things Conference*. BIOTC ’24. Fukuoka, Japan: Association for Computing Machinery, 2024, pp. 50–58. ISBN: 9798400717000. DOI: [10.1145/3688225.3688232](https://doi.org/10.1145/3688225.3688232). URL: <https://doi.org/10.1145/3688225.3688232>.
- [329] Yingjin Lu, Xianglan Jiang, and Lu Jiao. “The Privacy Protection for Shapley Value”. In: *2011 International Conference on Computational and Information Sciences*. 2011, pp. 428–430. DOI: [10.1109/ICCIS.2011.296](https://doi.org/10.1109/ICCIS.2011.296).
- [330] Scott Lundberg. *SHAP library documentation*. Accessed: 2021. 2018. URL: <https://shap.readthedocs.io/en/latest/index.html>.
- [331] Scott M Lundberg and Su-In Lee. “A Unified Approach to Interpreting Model Predictions”. In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc., 2017, pp. 4765–4774. URL: <http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf>.
- [332] Wanying Luo and Urs Hengartner. “VeriPlace: A Privacy-Aware Location Proof Architecture”. In: *Proceedings of the 18th SIGSPATIAL International Conference on Advances in*

- Geographic Information Systems*. GIS '10. San Jose, California: Association for Computing Machinery, 2010, pp. 23–32. ISBN: 9781450304283. DOI: [10.1145/1869790.1869797](https://doi.org/10.1145/1869790.1869797). URL: <https://doi.org/10.1145/1869790.1869797>.
- [333] Vitalik Buterin et al. *MACI: Minimal Anti-Collusion Infrastructure*. 2023. URL: <https://web.archive.org/web/20240331183835/https://maci.pse.dev/docs/introduction/>.
 - [334] Roger Sewell; David MacKay; and Iain McLean. “Probabilistic electoral methods, representative probability, and maximum entropy”. In: *Voting matters* (26 2009), pp. 16–38.
 - [335] Haitham Mahmoud et al. “A Framework for Decentralized, Real-Time Reputation Aggregation in IoV”. In: *IEEE Internet of Things Magazine* 6.2 (2023), pp. 44–48.
 - [336] Maker Team. *The Maker Protocol: MakerDAO’s Multi-Collateral Dai (MCD) System*. 2017. URL: [https://web.archive.org/web/20220414140839/https://makerdao.com/whitepaper/White%20Paper%20-The%20Maker%20Protocol_%20MakerDAO%E2%80%99s%20Multi-Collateral%20Dai%20\(MCD\)%20System-FINAL-%20021720.pdf](https://web.archive.org/web/20220414140839/https://makerdao.com/whitepaper/White%20Paper%20-The%20Maker%20Protocol_%20MakerDAO%E2%80%99s%20Multi-Collateral%20Dai%20(MCD)%20System-FINAL-%20021720.pdf).
 - [337] Gianclaudio Malgieri and Bart Custers. “Pricing privacy – the right to know the value of your personal data”. In: *Computer Law & Security Review* 34.2 (2018), pp. 289–303. ISSN: 2212-473X. DOI: <https://doi.org/10.1016/j.clsr.2017.08.006>. URL: <https://www.sciencedirect.com/science/article/pii/S0267364917302819>.
 - [338] Mary Maller et al. “Sonic: Zero-Knowledge SNARKs from Linear-Size Universal and Updatable Structured Reference Strings”. In: *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. CCS ’19. London, United Kingdom: Association for Computing Machinery, 2019, pp. 2111–2128. ISBN: 9781450367479. DOI: [10.1145/3319535.3339817](https://doi.org/10.1145/3319535.3339817). URL: <https://doi.org/10.1145/3319535.3339817>.
 - [339] Shaurya Malwa. *Attacker Takes Over Tornado Cash DAO With Vote Fraud, Token Slumps 40%*. 2023. URL: <https://web.archive.org/web/20230526052359/https://www.coindesk.com/tech/2023/05/21/attacker-takes-over-tornado-cash-dao-with-vote-fraud-token-slumps-40/>.
 - [340] Shaurya Malwa. *Tornado Cash DAO Attacker Starts to Move Ether, TORN Tokens*. 2023. URL: <https://web.archive.org/web/20250216054016/https://www.coindesk.com/tech/2023/05/25/tornado-cash-dao-attacker-starts-to-move-ether-torn-tokens>.
 - [341] Viktor Manahov. “Cryptocurrency liquidity during extreme price movements: is there a problem with virtual money?” In: *Commodities*. Chapman and Hall/CRC, 2022, pp. 731–762.
 - [342] MK Manoylov. “A complete timeline of FTX: From Alameda’s spiraling debt to its dramatic implosion”. In: *The Block* (2022). Accessed: October 2, 2025. URL: <https://web.archive.org/web/20250121083844/https://www.theblock.co/post/256106/a-complete-timeline-of-ftx-from-alamedas-spiraling-debt-to-its-dramatic-implosion>.
 - [343] Aida Manzano Kharman and William Sanders. “Vulnerabilities that arise from poor governance in Distributed Ledger Technologies”. In: *Digit. Gov.: Res. Pract.* (May 2025). Just Accepted. DOI: [10.1145/3731566](https://doi.org/10.1145/3731566). URL: <https://doi.org/10.1145/3731566>.
 - [344] Aida Manzano Kharman et al. “Tree Proof-of-Position Algorithms”. In: *IEEE Internet of Things Journal* 12.11 (2025), pp. 16393–16409. DOI: [10.1109/JIOT.2025.3532161](https://doi.org/10.1109/JIOT.2025.3532161).
 - [345] Dino Mark, Vlad Zamfir, and Emin Gün Sirer. “A Call for a Temporary Moratorium on the DAO”. In: *Hacking, Distributed* (2016).
 - [346] E Maskin. *Arrow’s IIA Condition, May’s Axioms, and the Borda Count*. Tech. rep. Working paper, 2020.
 - [347] Hartwig Mayer. “ECDSA security in bitcoin and ethereum: a research survey”. In: *Coin-Faabrik*, June 28.126 (2016), p. 50.

- [348] Andrew McLennan. “Randomized preference aggregation: Additivity of power and strategy proofness”. In: *Journal of Economic Theory* 22.1 (1980), pp. 1–11.
- [349] Sameer Mehta et al. “How to sell a data set? Pricing policies for data monetization”. In: *Information Systems Research* 32.4 (2021), pp. 1281–1297.
- [350] Julia Meier and Benedikt Schuppli. “The DAO Hack and the Living Law of Blockchain”. In: *Digitalisierung–Gesellschaft–Recht: Analysen und Perspektiven von Assistierenden des Rechtswissenschaftlichen Instituts der Universität Zürich* (2019), pp. 27–43.
- [351] ML Menéndez et al. “The Jensen-Shannon divergence”. In: *Journal of the Franklin Institute* 334.2 (1997), pp. 307–318.
- [352] A. J. Menezes, Paul C. Van Oorschot, and Scott A. Vanstone. *Handbook of applied cryptography*. eng. Discrete mathematics and its applications. Boca Raton: CRC Press, 1997. ISBN: 9780429466335.
- [353] Alfred Menezes. “An introduction to pairing-based cryptography”. In: *Recent trends in cryptography* 477 (2009), pp. 47–65. URL: <https://www.math.uwaterloo.ca/~ajmeneze/publications/pairings.pdf>.
- [354] Ralph Merkle. “DAOs, democracy and governance”. In: *Cryonics Magazine* 37.4 (2016), pp. 28–40.
- [355] Milagros Miceli, Julian Posada, and Tianling Yang. “Studying Up Machine Learning Data: Why Talk About Bias When We Mean Power?” In: *Proc. ACM Hum.-Comput. Interact.* 6.GROUP (Jan. 2022). DOI: [10.1145/3492853](https://doi.org/10.1145/3492853). URL: <https://doi.org/10.1145/3492853>.
- [356] Victor S. Miller. “Use of Elliptic Curves in Cryptography”. In: *Advances in Cryptology — CRYPTO ’85 Proceedings*. Ed. by Hugh C. Williams. Berlin, Heidelberg: Springer Berlin Heidelberg, 1986, pp. 417–426. ISBN: 978-3-540-39799-1.
- [357] Yifei Min et al. “Learn to match with no regret: reinforcement learning in Markov matching markets”. In: *Proceedings of the 36th International Conference on Neural Information Processing Systems*. NIPS ’22. New Orleans, LA, USA: Curran Associates Inc., 2022. ISBN: 9781713871088.
- [358] Du Mingxiao et al. “A review on consensus algorithm of blockchain”. In: *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 2017, pp. 2567–2572. DOI: [10.1109/SMC.2017.8123011](https://doi.org/10.1109/SMC.2017.8123011).
- [359] Eddie Mitchell. *Top Government Bitcoin Holders: US, Bhutan, El Salvador, and More*. Accessed: 2024-12-19. 2024. URL: <https://www.ccn.com/news/crypto/bhutan-top-crypto-nations-full-list-of-countries/>.
- [360] Aziz Mohaisen and Joongheon Kim. “The sybil attacks and defenses: a survey”. In: *arXiv preprint arXiv:1312.6349* (2013).
- [361] David Moher et al. “Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement”. In: *International journal of surgery* 8.5 (2010), pp. 336–341.
- [362] Terra Money. *Terra Governance*. <https://docs.terra.money/Concepts/Governance.html>. Accessed: December 19, 2024. 2019.
- [363] Dustin Moody et al. “Report on pairing-based cryptography”. In: *Journal of research of the National Institute of Standards and Technology* 120 (2015), p. 11. URL: <https://nvlpubs.nist.gov/nistpubs/jres/120/jres.120.002.pdf>.
- [364] Dustin Moody et al. *Transition to post-quantum cryptography standards*. Tech. rep. National Institute of Standards and Technology, 2024. URL: <https://nvlpubs.nist.gov/nistpubs/ir/2024/NIST.IR.8547.ipd.pdf>.
- [365] Allison Morrow. “‘Complete failure.’ Filing reveals staggering mismanagement inside FTX”. In: *CNN Business* (2022). Accessed: October 2, 2025. URL: <https://web.archive.org/>

- web/20250730185352/https://edition.cnn.com/2022/11/17/business/ftx-ceo-complete-failure/index.html.
- [366] Lawrence Mosley et al. “Towards a systematic understanding of blockchain governance in proposal voting: A Dash case study”. In: *Blockchain: Research and Applications* (2022), p. 100085.
 - [367] Janine Mügge et al. “Empowering end-of-life vehicle decision making with cross-company data exchange and data sovereignty via Catena-X”. In: *Sustainability* 15.9 (2023), p. 7187.
 - [368] Sadiq Muhammed T and Saji K Mathew. “The disaster of misinformation: a review of research in social media”. In: *International journal of data science and analytics* 13.4 (2022), pp. 271–285.
 - [369] Jack Anthony Murray. “Sell your cards to who: Non-fungible tokens and digital trading card games”. In: *AoIR Selected Papers of Internet Research* (2021).
 - [370] Robert Muth and Florian Tschorsch. “Empirical analysis of on-chain voting with smart contracts”. In: *International Conference on Financial Cryptography and Data Security*. Springer. 2021, pp. 397–412.
 - [371] Kelsie Nabben. “Decentralised Autonomous Organisations (DAOs) as Data Trusts: A general-purpose data governance framework for decentralised data ownership, storage, and utilisation”. In: *Available at SSRN* (2021).
 - [372] Taishi Nakai et al. “The Blockchain Trilemma Described by a Formula”. In: *2023 IEEE International Conference on Blockchain (Blockchain)*. 2023, pp. 41–46. DOI: [10.1109/Blockchain60715.2023.00016](https://doi.org/10.1109/Blockchain60715.2023.00016).
 - [373] Satoshi Nakamoto. *Bitcoin open source implementation of P2P currency*. <https://satoshi.nakamotoinstitute.org/posts/p2pfoundation/1/>. Online forum post. Feb. 2009.
 - [374] Satoshi Nakamoto. *Bitcoin: A peer-to-peer electronic cash system*. Accessed: June 6, 2025. 2008. URL: <https://bitcoin.org/bitcoin.pdf>.
 - [375] Neha Narula, Willy Vasquez, and Madars Virza. “zkledger: Privacy-preserving auditing for distributed ledgers”. In: *15th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 18)*. 2018, pp. 65–80.
 - [376] National Institute of Standards and Technology. *Digital Signature Standard (DSS)*. Tech. rep. FIPS PUB 186-4. U.S. Department of Commerce, July 2013. URL: <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf>.
 - [377] Danny Nelson and Nikhilesh De. *Mango Markets Exploiter Eisenberg Arrested in Puerto Rico*. 2023. URL: <https://web.archive.org/web/20250207153143/https://www.coindesk.com/policy/2022/12/27/mango-markets-exploiter-eisenberg-arrested-in-puerto-rico>.
 - [378] Polkadot Network. *Polkadot Governance*. <https://web.archive.org/web/20220126214930/https://wiki.polkadot.network/docs/learn-governance>. Accessed: September 18, 2025. 2016.
 - [379] Peter G Neumann. “Security criteria for electronic voting”. In: *NCSC 93* (1993), pp. 478–482.
 - [380] Terry Newman. “Tasmania and the secret ballot”. In: *Australian Journal of Politics & History* 49.1 (2003), pp. 93–101.
 - [381] Kervins Nicolas et al. “Blockchain System Defensive Overview for Double-Spend and Selfish Mining Attacks: A Systematic Approach”. In: *IEEE Access* 9 (2021), pp. 3838–3857. DOI: [10.1109/ACCESS.2020.3047365](https://doi.org/10.1109/ACCESS.2020.3047365).
 - [382] Michael A. Nielsen and Isaac L. Chuang. “Quantum computers: physical realization”. In: *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010, pp. 277–352.

- [383] Mohammad Reza Nosouhi et al. “Blockchain for secure location verification”. In: *Journal of Parallel and Distributed Computing* 136 (2020), pp. 40–51.
- [384] Mohammad Reza Nosouhi et al. “SPARSE: privacy-aware and collusion resistant location proof generation and verification”. In: *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE. 2018, pp. 1–6.
- [385] Edward Jr Ongweso. “Democratic DAO Suffers Coup, New Leader Steals Everything”. In: *Vice* (2022). URL: <https://web.archive.org/web/20250404074831/https://www.vice.com/en/article/democratic-dao-suffers-coup-new-leader-steals-everything/>.
- [386] Ontario Securities Commission. *QuadrigaCX: The Ontario Securities Commission’s Investigation into QuadrigaCX*. Accessed: October 2, 2025. 2020. URL: <https://web.archive.org/web/20220216020217/https://www.osc.ca/quadrigacxreport/>.
- [387] Organization for Security and Cooperation in Europe. *Document of the Copenhagen Meeting of the Conference on the Human Dimension of the CSCE*. 1990. URL: <https://www.osce.org/files/f/documents/9/c/14304.pdf>.
- [388] Organization of American States. *American Convention on Human Rights, "Pact of San Jose", Costa Rica*. 1969. URL: <https://www.refworld.org/docid/3ae6b36510.html>.
- [389] Bjorn Oude Roelink, Mohammed El-Hajj, and Dipti Sarmah. “Systematic review: Comparing zk-SNARK, zk-STARK, and bulletproof protocols for privacy-preserving authentication”. In: *Security and Privacy* 7.5 (2024), e401.
- [390] Kazim Rifat Özyilmaz, Mehmet Doğan, and Arda Yurdakul. “IDMoB: IoT data marketplace on blockchain”. In: *2018 crypto valley conference on blockchain technology (CVCBT)*. IEEE. 2018, pp. 11–19.
- [391] Christof Paar, Jan Pelzl, and Tim Güneysu. *Understanding Cryptography: From Established Symmetric and Asymmetric Ciphers to Post-Quantum Algorithms*. 2nd ed. Springer, 2024. ISBN: 978-3-662-69006-2. DOI: [10.1007/978-3-662-69007-9](https://doi.org/10.1007/978-3-662-69007-9). URL: <https://link.springer.com/content/pdf/10.1007/978-3-662-69007-9.pdf>.
- [392] Pascal Paillier. “Public-Key Cryptosystems Based on Composite Degree Residuosity Classes”. In: *Advances in Cryptology — EUROCRYPT ’99*. Ed. by Jacques Stern. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 223–238. ISBN: 978-3-540-48910-8.
- [393] Chris Panella. “Russian forces have hit on a cheap way to foil US precision weapons in Ukraine”. In: *Business Insider* (2024). Accessed: September 3, 2025. URL: <https://web.archive.org/web/20250728173848/https://www.businessinsider.com/russian-cheap-electronic-warfare-keeps-beating-us-precision-weapons-2024-4>.
- [394] Max Parasol. *Raider investors are looting DAOs — Nouns and Aragon share lessons learned*. <https://web.archive.org/web/20250818172613/https://cointelegraph.com/magazine/raider-investors-daos-nouns-aragon/>. Accessed: September 24, 2025. July 2024.
- [395] Sunoo Park et al. “Going from bad to worse: from internet voting to blockchain voting”. In: *Journal of Cybersecurity* 7.1 (2021), tyaa025.
- [396] Geraint Parry. “Trust, distrust and consensus”. In: *British journal of political science* 6.2 (1976), pp. 129–142.
- [397] Rowan van Pelt et al. “Defining blockchain governance: A framework for analysis and comparison”. In: *Information Systems Management* 38.1 (2021), pp. 21–41.
- [398] Mathew Penrose. *Random geometric graphs*. Vol. 5. OUP Oxford, 2003.
- [399] Billy Perrigo. “Exclusive: OpenAI Used Kenyan Workers on Less Than \$2 Per Hour to Make ChatGPT Less Toxic”. In: *Time* (2023). Accessed: August 8, 2025. URL: <https://time.com/6247678/openai-chatgpt-kenya-workers/>.

- [400] Stuart James. *Proof of Humanity - An Explainer*. Accessed: September 8, 2025. 2021. URL: <https://web.archive.org/web/20250213200723/https://blog.kleros.io/proof-of-humanity-an-explainer/>.
- [401] Serguei Popov. *The Tangle: White Paper*. Accessed: September 18, 2025. 2017. URL: <https://cryptoverze.s3.us-east-2.amazonaws.com/wp-content/uploads/2018/11/10012054/IOTA-MIOTA-Whitepaper.pdf>.
- [402] Serguei Popov and William J Buchanan. “Fpc-bi: Fast probabilistic consensus within byzantine infrastructures”. In: *Journal of Parallel and Distributed Computing* 147 (2021), pp. 77–86.
- [403] Jason Potts and Ellie Rennie. “Web3 and the creative industries: How blockchains are reshaping business models”. In: *A Research Agenda for Creative Industries*. Edward Elgar Publishing, 2019, pp. 93–111.
- [404] Bart Preneel, Vincent Rijmen, and Antoon Bosselaers. “Recent Developments in the Design of Conventional Cryptographic Algorithms”. In: *State of the Art in Applied Cryptography: Course on Computer Security and Industrial Cryptography Leuven, Belgium, June 3–6, 1997 Revised Lectures*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 105–130. ISBN: 978-3-540-49248-1. DOI: [10.1007/3-540-49248-8_4](https://doi.org/10.1007/3-540-49248-8_4). URL: https://doi.org/10.1007/3-540-49248-8_4.
- [405] R. Pries et al. “A New Replay Attack Against Anonymous Communication Networks”. In: *2008 IEEE International Conference on Communications*. 2008, pp. 1578–1582. DOI: [10.1109/ICC.2008.305](https://doi.org/10.1109/ICC.2008.305).
- [406] Committee to Protect Journalists. *Israel kills Al Jazeera journalists in targeted Gaza City airstrike*. Accessed: September 4, 2025. 2025. URL: <https://cpj.org/2025/08/israel-kills-al-jazeera-journalists-in-targeted-gaza-city-airstrike/>.
- [407] Orange Protocol. *Orange Protocol Documentation*. Tech. rep. Orange, 2022. URL: <https://web.archive.org/web/20230128060903/https://docs.orangeprotocol.io/overview>.
- [408] Yiannis Psaras and David Dias. “The InterPlanetary File System and the Filecoin Network”. In: *2020 50th Annual IEEE-IFIP International Conference on Dependable Systems and Networks-Supplemental Volume (DSN-S)*. 2020, pp. 80–80. DOI: [10.1109/DSN-S50200.2020.00043](https://doi.org/10.1109/DSN-S50200.2020.00043).
- [409] Sarah Rainsford. “Romanian court annuls result of presidential election first round”. In: *British Broadcasting Corporation (BBC)* (Dec. 2024). Accessed: August 6, 2025. URL: <https://www.bbc.co.uk/news/articles/cn4x2epppego>.
- [410] Aitazaz Ali Raja et al. “A market for trading forecasts: A wagering mechanism”. In: *International Journal of Forecasting* 40.1 (2024), pp. 142–159. ISSN: 0169-2070. DOI: <https://doi.org/10.1016/j.ijforecast.2023.01.007>. URL: <https://www.sciencedirect.com/science/article/pii/S0169207023000079>.
- [411] Clionadh Raleigh et al. “Introducing ACLED: An armed conflict location and event dataset”. In: *Journal of peace research* 47.5 (2010), pp. 651–660.
- [412] Gowri Sankar Ramachandran, Rahul Radhakrishnan, and Bhaskar Krishnamachari. “Towards a decentralized data marketplace for smart cities”. In: *2018 IEEE International Smart Cities Conference (ISC2)*. IEEE, 2018, pp. 1–8.
- [413] Botlhokwa N. D. Ranta. “The Unknown Women of Content Moderation”. In: *The Data Workers’ Inquiry* (2024). Accessed: August 8, 2025. URL: <https://data-workers.org/ranta/>.
- [414] Mohammad Rasouli and Michael I Jordan. “Data Sharing Markets”. In: *arXiv preprint arXiv:2107.08630* (2021).

- [415] Michel Rauchs et al. “Distributed ledger technology systems: A conceptual framework”. In: *Available at SSRN 3230013* (2018).
- [416] Michel Raynal. “Communication and agreement abstractions for fault-tolerant asynchronous distributed systems”. In: *Synthesis Lectures on Distributed Computing Theory* 1.1 (2010), pp. 1–273.
- [417] Jamie Redman. *Brazil Introduces Bill to Establish Strategic Bitcoin Reserve*. Accessed: 2024-12-20. 2024. URL: <https://news.bitcoin.com/brazil-introduces-bill-to-establish-strategic-bitcoin-reserve/>.
- [418] Fergal Reid and Martin Harrigan. “An Analysis of Anonymity in the Bitcoin System”. In: *Security and Privacy in Social Networks*. Ed. by Yaniv Altshuler et al. New York, NY: Springer New York, 2013, pp. 197–223. ISBN: 978-1-4614-4139-7. DOI: [10.1007/978-1-4614-4139-7_10](https://doi.org/10.1007/978-1-4614-4139-7_10). URL: https://doi.org/10.1007/978-1-4614-4139-7_10.
- [419] Wessel Reijers et al. “Now the code runs itself: On-chain and off-chain governance of blockchain technologies”. In: *Topoi* 40 (2021), pp. 821–831.
- [420] Rhizoo. *The Rook DAO Story: Postmortem of the \$25M governance takeover*. 2023. URL: <https://web.archive.org/web/20250218181844/https://mirror.xyz/rhizoo.eth/6ggZQ2g50pUVaAXEwYBldN2gT1CZxfr2D2sL9cn0eJc>.
- [421] Rigetti & Co, LLC. *Introducing the Novera QPU*. <https://web.archive.org/web/20250626220331/https://www.rigetti.com/novera>. Accessed: August 1, 2025. 2023.
- [422] Rigetti & Co, LLC. *Rigetti Computing Launches 84-Qubit Ankaa-3 System; Achieves 99.5% Median Two-Qubit Gate Fidelity Milestone*. <https://investors.rigetti.com/node/9886/pdf>. Accessed: August 1, 2025. 2024.
- [423] Team Rocket et al. *Scalable and Probabilistic Leaderless BFT Consensus through Metastability*. 2020. arXiv: [1906.08936 \[cs.DC\]](https://arxiv.org/abs/1906.08936). URL: <https://arxiv.org/abs/1906.08936>.
- [424] José Luis Romero Ugarte. “Distributed ledger technology (DLT): introduction”. In: *Banco de Espana Article* 19 (2018), p. 18.
- [425] Peter YA Ryan et al. “Prêt à voter: a voter-verifiable voting system”. In: *IEEE transactions on information forensics and security* 4.4 (2009), pp. 662–673.
- [426] Ashish Rajendra Sai et al. “Taxonomy of centralization in public blockchain systems: A systematic literature review”. In: *Information Processing & Management* 58.4 (2021), p. 102584.
- [427] Fahad Saleh. “Blockchain without Waste: Proof-of-Stake”. In: *The Review of Financial Studies* 34.3 (July 2020), pp. 1156–1190. ISSN: 0893-9454. DOI: [10.1093/rfs/hhaa075](https://doi.org/10.1093/rfs/hhaa075). eprint: <https://academic.oup.com/rfs/article-pdf/34/3/1156/36264598/hhaa075.pdf>. URL: <https://doi.org/10.1093/rfs/hhaa075>.
- [428] Idean Salehyan. “Best practices in the collection of conflict data”. In: *Journal of Peace Research* 52.1 (2015), pp. 105–109. DOI: [10.1177/0022343314551563](https://doi.org/10.1177/0022343314551563). eprint: <https://doi.org/10.1177/0022343314551563>. URL: <https://doi.org/10.1177/0022343314551563>.
- [429] Nichola Saminather. *Canadian Cryptocurrency Firm Collapsed Due to Ponzi Scheme by Late Founder: Regulator*. Accessed: October 2, 2025. 2020. URL: <https://www.reuters.com/article/technology/canadian-cryptocurrency-firm-collapsed-due-to-ponzi-scheme-by-late-founder-regu-idUSKBN23I3AF>.
- [430] Nishant Sapra et al. “Uncovering Bitcoin’s electricity consumption relationships with volatility and price: Environmental Repercussions”. In: *Journal of Environmental Management* 356 (2024), p. 120528. ISSN: 0301-4797. DOI: <https://doi.org/10.1016/j.jenvman.2024.120528>. URL: <https://www.sciencedirect.com/science/article/pii/S0301479724005140>.

- [431] Shayan Sardarizadeh and Alistair Coleman. “Ukraine conflict: Many misleading images have been shared online”. In: *British Broadcasting Corporation (BBC)* (2022). Accessed: September 4, 2025. URL: <https://www.bbc.com/news/60513452>.
- [432] Or Sattath. “On the insecurity of quantum Bitcoin mining”. In: *International Journal of Information Security* 19.3 (June 2020), pp. 291–302. ISSN: 1615-5270. DOI: [10.1007/s10207-020-00493-9](https://doi.org/10.1007/s10207-020-00493-9). URL: <https://doi.org/10.1007/s10207-020-00493-9>.
- [433] Mark Allen Satterthwaite. “Strategy-proofness and Arrow’s conditions: Existence and correspondence theorems for voting procedures and social welfare functions”. In: *Journal of economic theory* 10.2 (1975), pp. 187–217.
- [434] Jason Scharfman and Jason Scharfman. “Cryptocurrency Compliance and Operations Case Studies”. In: *Cryptocurrency Compliance and Operations: Digital Assets, Blockchain and DeFi* (2022), pp. 155–170.
- [435] Christophe Olivier Schneble, Bernice Simone Elger, and David Martin Shaw. “Google’s Project Nightingale highlights the necessity of data science ethics review”. In: *EMBO Molecular Medicine* 12.3 (2020), e12053. DOI: <https://doi.org/10.15252/emmm.202012053>. eprint: <https://www.embopress.org/doi/pdf/10.15252/emmm.202012053>. URL: <https://www.embopress.org/doi/abs/10.15252/emmm.202012053>.
- [436] Christophe Olivier Schneble, Bernice Simone Elger, and David Martin Shaw. “Google’s Project Nightingale highlights the necessity of data science ethics review”. In: *EMBO molecular medicine* 12.3 (2020), e12053.
- [437] Nathan Schneider. “Cryptoeconomics as a limitation on governance”. In: *Regulation and Governance* (2022).
- [438] Ludovit Scholtz. *VoteCoin*. 2022. URL: <https://web.archive.org/web/20220131162622/https://www.vote-coin.com/>.
- [439] Patrick Schueffel. “Alternative distributed ledger technologies Blockchain vs. Tangle vs. Hashgraph-A high-level overview and comparison”. In: *Tangle vs. Hashgraph-A High-Level Overview and Comparison (December 15, 2017)* (2017).
- [440] Securities and Exchange Commission. *SEC v. BitConnect, et al.* Accessed: October 2, 2025. 2021. URL: <https://web.archive.org/web/20250316092731/https://www.sec.gov/files/litigation/complaints/2021/comp-pr2021-172.pdf>.
- [441] Sabina Petride. 2002. URL: <https://www.cs.cornell.edu/courses/cs513/2002SP/proj.00.StuSolns/sp2580.htm>.
- [442] Kevin Sekniqi et al. *Avalanche Platform Whitepaper*. https://cdn.prod.website-files.com/5d80307810123f5ffbb34d6e/6008d7bbf8b10d1eb01e7e16_Avalanche%20Platform%20Whitepaper.pdf. Accessed: September 18, 2025. 2020.
- [443] *Semaphore projects*. <https://semaphore.pse.dev/>, Last accessed on 2024-04-10. 2022.
- [444] Alesha Serada, Jori Grym, and Tanja Sihvonen. “The Economy of Attention on Blockchain in the Brave Browser”. In: *Futures of Journalism*. Springer, 2022, pp. 49–62.
- [445] Roger Sewell, David MacKay, and Iain McLean. “Probabilistic electoral methods, representative probability, and maximum entropy”. In: *Voting matters* 26 (2009), pp. 16–38.
- [446] Cosimo Sguanci, Roberto Spatafora, and Andrea Mario Vergani. *Layer 2 Blockchain Scaling: a Survey*. 2021. arXiv: [2107.10881](https://arxiv.org/abs/2107.10881) [cs.DC]. URL: <https://arxiv.org/abs/2107.10881>.
- [447] Yahya Shahsavari, Kaiwen Zhang, and Chamseddine Talhi. “A Theoretical Model for Fork Analysis in the Bitcoin Network”. In: *2019 IEEE International Conference on Blockchain (Blockchain)*. 2019, pp. 237–244. DOI: [10.1109/Blockchain.2019.00038](https://doi.org/10.1109/Blockchain.2019.00038).
- [448] Claude E. Shannon. *A Mathematical Theory of Cryptography*. Memorandum MM 45-110-02. A Classified Memorandum for Bell Telephone Labs. Re-typed by Whitfield Diffie. Now

- declassified, hosted by the IACR Cryptology Museum. Accessed: July 14, 2025. Murray Hill, NJ: Bell Laboratories, Sept. 1945, pp. 92–94. URL: <https://www.iacr.org/museum/shannon45.pdf>.
- [449] Claude. Shannon. “Communication theory of secrecy systems”. In: *The Bell System Technical Journal* 28.4 (1949), pp. 656–715. DOI: [10.1002/j.1538-7305.1949.tb00928.x](https://doi.org/10.1002/j.1538-7305.1949.tb00928.x).
 - [450] LS Shapley. *Contributions to the Theory of Games (AM-28), Volume II*. Ed. by Harold William Kuhn and Albert William Tucker. Princeton University Press, 1953. ISBN: 9781400881970. DOI: [doi:10.1515/9781400881970](https://doi.org/10.1515/9781400881970). URL: <https://doi.org/10.1515/9781400881970>.
 - [451] Anna Howard Shaw. “Equal Suffrage—a Problem of Political Justice”. In: *The Annals of the American Academy of Political and Social Science* 56.1 (1914), pp. 93–98.
 - [452] Peter W. Shor. “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer”. In: *SIAM Review* 41.2 (1999), pp. 303–332. DOI: [10.1137/S0036144598347011](https://doi.org/10.1137/S0036144598347011). eprint: <https://doi.org/10.1137/S0036144598347011>. URL: <https://doi.org/10.1137/S0036144598347011>.
 - [453] Thomas Shrimpton and R. Seth Terashima. “A Provable-Security Analysis of Intel’s Secure Key RNG”. In: *Advances in Cryptology – EUROCRYPT 2015*. Ed. by Elisabeth Oswald and Marc Fischlin. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 77–100. ISBN: 978-3-662-46800-5.
 - [454] Sidhartha Shukla. *Sanctioned Crypto Mixer Tor*. 2023. URL: <https://www.bloomberg.com/news/articles/2023-05-21/sanctioned-crypto-mixer-tornado-cash-hijacked-by-hackers>.
 - [455] Thomas Silkjaer. *Is This \$3 Billion Crypto Ponzi Still Alive?* Accessed: September 26, 2025. 2020. URL: <https://web.archive.org/web/20201127092533/https://www.forbes.com/sites/thomassilkjaer/2020/07/06/is-this-3-billion-crypto-ponzi-still-alive/>.
 - [456] Phil Simon. *Too big to ignore : the business case for big data*. eng. 1st ed. Wiley & SAS business series. Hoboken, N.J: John Wiley & Sons, 2013. ISBN: 9781119204039.
 - [457] Alexandra Sims. *Blockchain and decentralised autonomous organisations (DAOs): the evolution of companies?* 2019.
 - [458] Meital Ben Sinai et al. “Exploiting social navigation”. In: *arXiv preprint arXiv:1410.0151* (2014).
 - [459] Nigel Smart. *Bootstrapping for Dummies*. Accessed: October 1, 2025. 2022. URL: <https://web.archive.org/web/20250221090855/https://www.zama.ai/post/what-is-bootstrapping-homomorphic-encryption>.
 - [460] Nigel Smart. *Cryptography: An Introduction*. 3rd ed. McGraw-Hill College, 2004. ISBN: 978-0077099879. URL: <https://www.cs.umd.edu/~waa/414-F11/IntroToCrypto.pdf>.
 - [461] Nigel. Smart. *Cryptography Made Simple*. eng. 1st ed. 2016. Information Security and Cryptography. Cham: Springer International Publishing, 2016. ISBN: 3-319-21936-7.
 - [462] Chris Smith. *The Biggest Heist Ever*. Documentary. 2024.
 - [463] Ben Smyth. “Athena: A verifiable, coercion-resistant voting system with linear complexity”. In: *Cryptology ePrint Archive* (2019).
 - [464] Ben Smyth. “Ballot secrecy: Security definition, sufficient conditions, and analysis of Helios”. In: *Journal of Computer Security Preprint* (2021), pp. 1–61.
 - [465] Ben Smyth, Steven Frink, and Michael R Clarkson. “Computational election verifiability: Definitions and an analysis of helios and JCJ”. In: *Cryptology ePrint Archive* (2015).
 - [466] Ben Smyth, Steven Frink, and Michael R Clarkson. *Election verifiability: cryptographic definitions and an analysis of Helios and JCJ*. Tech. rep. Huawei Technologies, IBM Watson, Cornell University, 2017.

- [467] Michael Sober et al. “A blockchain-based IoT data marketplace”. In: *Cluster computing* 26.6 (2023), pp. 3523–3545.
- [468] Anda I. Solea. “Why Romania’s election was annulled – and what happens next”. In: *The Conversation* (Dec. 2024). Accessed: August 6, 2025. URL: <https://theconversation.com/why-romanias-election-was-annulled-and-what-happens-next-245779>.
- [469] Solidus Labs. *The Mango Markets Exploit: An Order Book Analysis*. URL: <https://web.archive.org/web/20250210123109/https://www.soliduslabs.com/post/mango-hack>.
- [470] S B Sonu and A. Suyampulingam. “Linear Regression Based Air Quality Data Analysis and Prediction using Python”. In: *2021 IEEE Madras Section Conference (MASCONE)*. 2021, pp. 1–7. DOI: [10.1109/MASCONE51689.2021.9563432](https://doi.org/10.1109/MASCONE51689.2021.9563432).
- [471] Chiara Spadafora, Riccardo Longo, and Massimiliano Sala. “Coercion-resistant blockchain-based e-voting protocol”. In: *Cryptology ePrint Archive* (2020).
- [472] BBC Staff. “Google should not be in business of war, say employees”. In: *British Broadcasting Corporation (BBC)* (Apr. 2018). Accessed: August 8, 2025. URL: <https://www.bbc.co.uk/news/business-43656378>.
- [473] Florian Stahl, Fabian Schomm, and Gottfried Vossen. “Data Marketplaces: An Emerging Species.” In: *DB&IS*. 2014, pp. 145–158.
- [474] Erica Stanford. *Crypto Wars: Faked Deaths, Missing Billions and Industry Disruption*. Kogan Page, Limited, 2023. ISBN: 9781398610514. URL: <https://books.google.es/books?id=N7pIzwEACAAJ>.
- [475] Felix Sterk et al. “Understanding Car Data Monetization: A Taxonomy of Data-Driven Business Models in the Connected Car Domain”. In: *Wirtschaftsinformatik 2022 Proceedings*. Proceedings of the 17th International Conference on Wirtschaftsinformatik (WI 2022). Nürnberg (online): Association for Information Systems (AIS), Feb. 2022, p. 7. URL: https://aisel.aisnet.org/wi2022/digital_business_models/digital_business_models/7.
- [476] Robert Stevens. “KuCoin Has Found The Hackers Who Stole \$281 Million in Crypto”. In: *Decrypt* (2020). Accessed: October 2, 2025. URL: <https://web.archive.org/web/20201119123507/https://decrypt.co/43806/kucoin-has-found-the-hackers-who-stole-281-million>.
- [477] E. M. Stoudenmire and Xavier Waintal. “ch”. In: *Phys. Rev. X* 14 (4 Nov. 2024), p. 041029. DOI: [10.1103/PhysRevX.14.041029](https://doi.org/10.1103/PhysRevX.14.041029). URL: <https://link.aps.org/doi/10.1103/PhysRevX.14.041029>.
- [478] Maurice E Stucke. “Should we be concerned about data-opolies?” In: *Geo. L. Tech. Rev.* 2 (2017), p. 275. URL: <https://georgetownlawtechreview.org/should-we-be-concerned-about-data-opolies/GLTR-07-2018/>.
- [479] Eduardo Suárez. “BBC expert on debunking Israel-Hamas war visuals: “The volume of misinformation on Twitter was beyond anything I’ve ever seen””. In: *Reuters Institute* (2023). Accessed: September 4, 2025. URL: <https://reutersinstitute.politics.ox.ac.uk/news/bbc-expert-debunking-israel-hamas-war-visuals-volume-misinformation-twitter-was-beyond>.
- [480] Arun Subramanian. *United States of America, Plaintiff, -against- Avraham Eisenberg, Defendant: Opinion and Order, Case No. 23-cr-10 (AS)*. <https://web.archive.org/web/20250602094330/https://www.nysd.uscourts.gov/sites/default/files/2025-05/23cr10%20Opinion%20and%20Order.pdf>. Accessed: September 24, 2025. May 2025.
- [481] Mengqi Sun. “Crypto Mixer Samourai Wallet’s Co-founders Arrested for Money Laundering”. In: *The Wall Street Journal* (2024). Accessed September 23, 2025.

- [482] Xiaoqiang Sun et al. “A Survey on Zero-Knowledge Proof in Blockchain”. In: *IEEE Network* 35.4 (2021), pp. 198–205. DOI: [10.1109/MNET.011.2000473](https://doi.org/10.1109/MNET.011.2000473).
- [483] Ralph Sundberg and Erik Melander. “Introducing the UCDP georeferenced event dataset”. In: *Journal of peace research* 50.4 (2013), pp. 523–532.
- [484] Ali Sunyaev and Ali Sunyaev. “Distributed ledger technology”. In: *Internet computing: Principles of distributed systems and emerging internet-based technologies* (2020), pp. 265–299.
- [485] Saurabh Suratkar, Mahesh Shirole, and Sunil Bhirud. “Cryptocurrency wallet: A review”. In: *2020 4th international conference on computer, communication and signal processing (ICCCSP)*. IEEE. 2020, pp. 1–7.
- [486] Harrie de Swart et al. “Categoric and Ordinal Voting: An Overview”. In: *Theory and Applications of Relational Structures as Knowledge Instruments: COST Action 274, TARSKI. Revised Papers*. Ed. by Harrie de Swart et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 147–195. ISBN: 978-3-540-24615-2. DOI: [10.1007/978-3-540-24615-2_8](https://doi.org/10.1007/978-3-540-24615-2_8). URL: https://doi.org/10.1007/978-3-540-24615-2_8.
- [487] Nick Szabo. *The Idea of Smart Contracts*. 1997. URL: <https://nakamotoinstitute.org/library/the-idea-of-smart-contracts/>.
- [488] Joshua Z Tan et al. “Open problems in daos”. In: *arXiv preprint arXiv:2310.19201* (2023).
- [489] Ruhi Taş and Ömer Özgür Tanrıöver. “A systematic review of challenges and opportunities of blockchain for E-voting”. In: *Symmetry* 12.8 (2020), p. 1328.
- [490] Fitz Tepper. “Bitconnect, which has been accused of running a Ponzi scheme, shuts down”. In: *TechCrunch* (2018). Accessed: October 2, 2025. URL: <https://web.archive.org/web/20250207212450/https://techcrunch.com/2018/01/16/bitconnect-which-has-been-accused-of-running-a-ponzi-scheme-shuts-down/>.
- [491] Louis Tremblay Thibault, Tom Sarry, and Abdelhakim Senhaji Hafid. “Blockchain Scaling Using Rollups: A Comprehensive Survey”. In: *IEEE Access* 10 (2022), pp. 93039–93054. DOI: [10.1109/ACCESS.2022.3200051](https://doi.org/10.1109/ACCESS.2022.3200051).
- [492] Thomas Perronet Thompson. *An abridgement of the article on the ballot which appeared in the Westminster Review, No. XXV, for July, 1830, earnestly recommended to the serious consideration of every elector in the United Kingdom*. At the office of the National Political Union, 1832. URL: <https://jstor.org/stable/60209037> (visited on 08/30/2022).
- [493] Joe Tidy. “Ronin Network: What a \$600m hack says about the state of crypto”. In: *British Broadcasting Corporation (BBC)* (2022). Accessed: October 2, 2025. URL: <https://web.archive.org/web/20250821021414/https://www.bbc.com/news/technology-60933174>.
- [494] Jamel Toppin. *DAOs Aren’t A Fad — They’re A Platform*. 2022. URL: <https://web.archive.org/web/20220211074026/https://www.forbes.com/sites/jeffkauflin/2022/02/03/daos-arent-a-fad-theyre-a-platform/?sh=42ea7dc719d0>.
- [495] Sam Toueg. “Randomized byzantine agreements”. In: *Proceedings of the third annual ACM symposium on Principles of distributed computing*. 1984, pp. 163–178.
- [496] Matias Travizano et al. “Wibson: A Decentralized Data Marketplace”. In: *Proceedings of the SIGBPS Workshop on Blockchain and Smart Contract*. Also available as arXiv preprint arXiv:1812.09966. San Francisco, CA, USA: ACM Special Interest Group on Business Processes and Services (SIGBPS), Dec. 2018. URL: <https://arxiv.org/abs/1812.09966>.
- [497] U.S. Department of the Treasury. *U.S. Treasury Issues First-Ever Sanctions on a Virtual Currency Mixer, Targets DPRK Cyber Threats*. <https://web.archive.org/web/20250904231737/https://home.treasury.gov/news/press-releases/jy0768>. Press release. Accessed September 23, 2025. June 2022.

- [498] TRM Labs. *BREAKING: Federal Judge Overturns All Criminal Convictions in Mango Markets Case Against Avraham Eisenberg*. <https://web.archive.org/web/20250529144146/https://www.trmlabs.com/resources/blog/breaking-federal-judge-overturns-all-criminal-convictions-in-mango-markets-case-against-avraham-eisenberg>. Accessed: September 24, 2025. May 2025.
- [499] TRM Labs. *Ilya Lichtenstein and Heather 'Razzelekh' Morgan Sentenced In The Bitfinex Case As Government Recovers \$10 Billion in Stolen Funds*. Accessed: September 26, 2025. Nov. 2024. URL: <https://web.archive.org/web/20250225084242/https://www.trmlabs.com/post/ilya-lichtenstein-sentenced-for-role-in-bitfinex-hack-in-razzelekh-case-as-government-recovers-about-10-billion-in-stolen-funds>.
- [500] Yoichi Tsuchiya and Naoki Hiramoto. "How cryptocurrency is laundered: Case study of Coincheck hacking incident". In: *Forensic Science International: Reports* 4 (2021), p. 100241.
- [501] Meltem Sönmez Turan et al. *Recommendation for the Entropy Sources Used for Random Bit Generation*. Tech. rep. NIST SP 800-90B. National Institute of Standards and Technology (NIST), 2018. URL: <https://doi.org/10.6028/NIST.SP.800-90B>.
- [502] U.S. Homeland Security Investigations. *Bitfinex Hacker Sentenced in Money Laundering Conspiracy Involving Billions in Stolen Cryptocurrency*. Accessed: October 2, 2025. 2024. URL: <https://web.archive.org/web/20250126233355/https://www.dhs.gov/hsi/news/2024/11/20/bitfinex-hacker-sentenced-money-laundering-conspiracy-involving-billions-stolen>.
- [503] U.S. Department of Justice. *BitConnect Founder Indicted in Global \$2.4 Billion Cryptocurrency Scheme*. Accessed: October 2, 2025. 2022. URL: <https://web.archive.org/web/20220228000646/https://www.justice.gov/opa/pr/bitconnect-founder-indicted-global-24-billion-cryptocurrency-scheme>.
- [504] U.S. Deputy Secretary of Defense. *Establishment of an Algorithmic Warfare Cross-Functional Team (Project Maven)*. Accessed: August 8, 2025. 1010 Defense Pentagon, Washington, DC 20301-1010, Apr. 26, 2017. URL: https://www.ftc.gov/system/files/documents/cases/182_3107_cambridge_analytica_administrative_complaint_7-24-19.pdf.
- [505] U.S. Securities and Exchange Commission. *Complaint — U.S. Securities and Exchange Commission v. Avraham Eisenberg*. <https://web.archive.org/web/20250710024107/https://www.sec.gov/files/litigation/complaints/2023/comp-pr2023-13.pdf>. Accessed: September 24, 2025. 2023.
- [506] UN General Assembly. "Universal declaration of human rights". In: *UN General Assembly* 302.2 (1948), pp. 14–25.
- [507] United Nations Economist Network. *New Economics For Sustainable Development: Attention Economy*. Tech. rep. Accessed: August 8, 2025. United Nations, 2023. URL: https://www.un.org/sites/un2.un.org/files/attention_economy_feb.pdf.
- [508] United States Department of Justice. *Statement of Facts: Two Arrested for Alleged Conspiracy to Launder \$4.5 Billion in Stolen Cryptocurrency*. <https://web.archive.org/web/20250909011355/https://www.justice.gov/archives/opa/press-release/file/1470211/dl?inline==>. Accessed: September 26, 2025. 2022.
- [509] Northern District of California United States District Court. *Order Re Meta's Sealing Requests (Case No. 23-cv-03417-VC)*. <https://storage.courtlistener.com/recap/gov.uscourts.cand.415175/gov.uscourts.cand.415175.373.0.pdf>. Accessed: August 8, 2025. Jan. 2025.
- [510] US Department of Justice. *SSNOB Marketplace, A Series Of Websites That Listed More Than 20 Million Social Security Numbers For Sale, Seized And Dismantled In International*

- Operation. 2022. URL: <https://www.justice.gov/usao-mdfl/pr/ssndob-marketplace-series-websites-listed-more-20-million-social-security-numbers-sale>.
- [511] Chris Vallance. “Africrypt brothers deny involvement in Bitcoin ‘heist’”. In: *British Broadcasting Corporation (BBC)* (2021). Accessed: October 2, 2025. URL: <https://web.archive.org/web/20250604074353/https://www.bbc.com/news/technology-57582805>.
 - [512] Matthew Van Niekerk and Roderik van der Veer. *Global market for local data*. 2018. URL: <https://www.allcryptowhitepapers.com/wp-content/uploads/2018/11/Databroker-DA0.pdf>.
 - [513] Lieven M. K. Vandersypen et al. “Experimental realization of Shor’s quantum factoring algorithm using nuclear magnetic resonance”. In: *Nature* 414.6866 (Dec. 2001), pp. 883–887. ISSN: 1476-4687. DOI: [10.1038/414883a](https://doi.org/10.1038/414883a). URL: <https://doi.org/10.1038/414883a>.
 - [514] Marie Vasek, Micah Thornton, and Tyler Moore. “Empirical Analysis of Denial-of-Service Attacks in the Bitcoin Ecosystem”. In: *Financial Cryptography and Data Security*. Ed. by Rainer Böhme et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 57–71. ISBN: 978-3-662-44774-1.
 - [515] Marie Vasek et al. “The Bitcoin brain drain: a short paper on the use and abuse of bitcoin brain wallets”. In: *Financial Cryptography and Data Security, Lecture Notes in Computer Science*. Springer (2016).
 - [516] Jan Vermeulen. “Surprising new information about South Africa’s Bitcoin Brothers”. In: *Mybroadband* (2025). Accessed: October 2, 2025. URL: <https://web.archive.org/web/20250618205607/https://mybroadband.co.za/news/cryptocurrency/594856-surprising-new-information-about-south-africas-bitcoin-brothers.html>.
 - [517] Friedhelm Victor and Sebastian Zickau. “Geofences on the blockchain: Enabling decentralized location-based services”. In: *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE. 2018, pp. 97–104.
 - [518] Vivek Vishnumurthy, Sangeeth Chandrakumar, and Emin Gun Sirer. “Karma: A secure economic framework for peer-to-peer resource sharing”. In: *Workshop on Economics of Peer-to-peer Systems*. Vol. 35. 6. 2003.
 - [519] Vocdoni. *Vocdoni Documentation*. 2021. URL: <https://docs.vocdoni.io/architecture/general.html>.
 - [520] Paul Voigt and Axel Von dem Bussche. “The eu general data protection regulation (gdpr)”. In: *A practical guide, 1st ed., Cham: Springer International Publishing* 10.3152676 (2017), pp. 10–5555.
 - [521] Marko Vukolić. “The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication”. In: *International workshop on open problems in network security*. Springer. 2015, pp. 112–125.
 - [522] Anton Wahrstätter et al. “Blockchain Censorship”. In: *arXiv preprint arXiv:2305.18545* (2023).
 - [523] Anton Wahrstätter et al. “Blockchain censorship”. In: *Proceedings of the ACM Web Conference 2024*. 2024, pp. 1632–1643.
 - [524] John Walker. *HotBits: Genuine Random Numbers Generated by Radioactive Decay*. <https://web.archive.org/web/20250405011259/https://www.fourmilab.ch/hotbits/>. Accessed: September 27, 2025. 1996.
 - [525] Canhui Wang, Xiaowen Chu, and Yang Qin. “Measurement and Analysis of the Bitcoin Networks: A View from Mining Pools”. In: *2020 6th International Conference on Big Data Computing and Communications (BIGCOM)*. 2020, pp. 180–188. DOI: [10.1109/BigCom51056.2020.00032](https://doi.org/10.1109/BigCom51056.2020.00032).

- [526] Shengling Wang, Chenyu Wang, and Qin Hu. “Corking by forking: Vulnerability analysis of blockchain”. In: *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE. 2019, pp. 829–837.
- [527] Wenbo Wang et al. “A Survey on Consensus Mechanisms and Mining Strategy Management in Blockchain Networks”. In: *IEEE Access* 7 (2019), pp. 22328–22370. DOI: [10.1109/ACCESS.2019.2896108](https://doi.org/10.1109/ACCESS.2019.2896108).
- [528] Wenbo Wang et al. “A Survey on Consensus Mechanisms and Mining Strategy Management in Blockchain Networks”. In: *IEEE Access* 7 (2019), pp. 22328–22370. DOI: [10.1109/ACCESS.2019.2896108](https://doi.org/10.1109/ACCESS.2019.2896108).
- [529] Euan Ward and Kate Lyons. “Who’s buying Vanuatu’s passports? Crypto moguls, disgraced business people and wanted men”. In: *The Guardian* (2021). Accessed: October 2, 2025. URL: <https://web.archive.org/web/20250605061148/https://www.theguardian.com/world/2021/jul/15/whos-buying-vanuatus-passports-crypto-moguls-wanted-men>.
- [530] Nick Webb. “A fork in the blockchain: income tax and the bitcoin/bitcoin cash hard fork”. In: *North Carolina Journal of Law & Technology* 19.4 (2018), p. 283.
- [531] Georgia Wells, Jeff Horwitz, and Deepa Seetharaman. “Facebook Knows Instagram Is Toxic for Teen Girls, Company Documents Show”. In: *Wall Street Journal* (Sept. 2021). Accessed: August 6, 2025. Available at: <https://www.congress.gov/117/meeting/house/114054/documents/HHRG-117-IF02-20210922-SD003.pdf>. URL: <https://www.wsj.com/tech/personal-tech/facebook-knows-instagram-is-toxic-for-teen-girls-company-documents-show-11631620739>.
- [532] Andreas G. Wertgen and Tobias Richter. “Source credibility and plausibility are considered in the validation of textual information: evidence from a social media context”. In: *Journal of Cognitive Psychology* 35.2 (2023), pp. 183–204. DOI: [10.1080/20445911.2022.2149757](https://doi.org/10.1080/20445911.2022.2149757). eprint: <https://doi.org/10.1080/20445911.2022.2149757>. URL: <https://doi.org/10.1080/20445911.2022.2149757>.
- [533] Ron White and Timothy Edward Downs. *How computers work*. eng. 9th ed. Place of publication not identified: Que Pub, 2007. ISBN: 9780768676709.
- [534] Christopher Whyte. “Deepfake news: AI-enabled disinformation as a multi-level public policy challenge”. In: *Journal of Cyber Policy* 5.2 (2020), pp. 199–217. DOI: [10.1080/23738871.2020.1797135](https://doi.org/10.1080/23738871.2020.1797135). eprint: <https://doi.org/10.1080/23738871.2020.1797135>. URL: <https://doi.org/10.1080/23738871.2020.1797135>.
- [535] Alex Wilhelm. “Datacy raises \$2.4M to help consumers monetize their own damn data”. In: *Tech Crunch* (2021). URL: <https://web.archive.org/web/20220702013551/https://techcrunch.com/2021/05/26/datacy-raises-2-4m-to-help-consumers-monetize-their-own-damn-data/>.
- [536] Wired Ideas. “The Web3 Decentralization Debate Is Focused on the Wrong Question”. In: *Wired* (2022). Accessed: September 4, 2025. URL: <https://web.archive.org/web/20250315173917/https://www.wired.com/story/web3-blockchain-decentralization-governance/>.
- [537] Gavin Wood et al. “Ethereum: A secure decentralised generalised transaction ledger”. In: *Ethereum project yellow paper* 151.2014 (2014), pp. 1–32.
- [538] Wei Wu et al. “Blockchain Based Zero-Knowledge Proof of Location in IoT”. In: *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*. 2020, pp. 1–7. DOI: [10.1109/ICC40277.2020.9149366](https://doi.org/10.1109/ICC40277.2020.9149366).
- [539] Wei Wu et al. “Blockchain based zero-knowledge proof of location in iot”. In: *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE. 2020, pp. 1–7.

- [540] Christopher Wylie. *Cambridge Analytica and the Future of Data Privacy: Written Testimony for the Senate Judiciary Committee*. Statement for the Record. United States Senate Committee on the Judiciary. May 16, 2018. URL: <https://www.judiciary.senate.gov/imo/media/doc/05-16-18%20Wylie%20Testimony.pdf>.
- [541] Pengcheng Xia et al. “Characterizing cryptocurrency exchange scams”. In: *Computers & Security* 98 (2020), p. 101993. ISSN: 0167-4048. DOI: <https://doi.org/10.1016/j.cose.2020.101993>. URL: <https://www.sciencedirect.com/science/article/pii/S0167404820302662>.
- [542] Tiancheng Xie et al. “zkBridge: Trustless Cross-chain Bridges Made Practical”. In: *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. CCS ’22. Los Angeles, CA, USA: Association for Computing Machinery, 2022, pp. 3003–3017. ISBN: 9781450394505. DOI: [10.1145/3548606.3560652](https://doi.org/10.1145/3548606.3560652). URL: <https://doi.org/10.1145/3548606.3560652>.
- [543] Anatoly Yakovenko. “Solana: A new architecture for a high performance blockchain v0.8.13”. In: *Whitepaper* (2018).
- [544] Andrew C. Yao. “Theory and application of trapdoor functions”. In: *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*. 1982, pp. 80–91. DOI: [10.1109/SFCS.1982.45](https://doi.org/10.1109/SFCS.1982.45).
- [545] David Yermack. “Corporate Governance and Blockchains”. In: *Review of Finance* 21.1 (2017), pp. 7–31.
- [546] Joseph Young. *QuadrigaCX Wallets Found Empty, Mystery Deepens*. Accessed: October 2, 2025. 2019. URL: <https://web.archive.org/web/20190305165314/https://finance.yahoo.com/news/quadrigacx-wallets-found-empty-mystery-102140761.html>.
- [547] Abraham Yuval. “‘Lavender’: The AI machine directing Israel’s bombing spree in Gaza”. In: *+972 Magazine* (Apr. 2024). Accessed: August 8, 2025. URL: <https://www.972mag.com/lavender-ai-israeli-army-gaza/>.
- [548] Markos Zachariadis, Garrick Hileman, and Susan V Scott. “Governance and control in distributed ledgers: Understanding the challenges facing blockchain technology in financial services”. In: *Information and organization* 29.2 (2019), pp. 105–117.
- [549] Thomas Zeitzoff. “How Social Media Is Changing Conflict”. In: *Journal of Conflict Resolution* 61.9 (2017), pp. 1970–1991. DOI: [10.1177/0022002717721392](https://doi.org/10.1177/0022002717721392). eprint: <https://doi.org/10.1177/0022002717721392>. URL: <https://doi.org/10.1177/0022002717721392>.
- [550] Dirk Zetzsche and Areti Nikolakopoulou. “Crypto Custody: An Empirical Assessment”. In: *Journal of Financial Regulation* 11.1 (Apr. 2025), pp. 73–97. ISSN: 2053-4841. DOI: [10.1093/jfr/fjaf004](https://doi.org/10.1093/jfr/fjaf004). eprint: <https://academic.oup.com/jfr/article-pdf/11/1/73/62834127/fjaf004.pdf>. URL: <https://doi.org/10.1093/jfr/fjaf004>.
- [551] Dirk A Zetzsche et al. “The Markets in Crypto-Assets regulation (MiCA) and the EU digital finance strategy”. In: *Capital Markets Law Journal* 16.2 (2021), pp. 203–225.
- [552] Rong Zhang and Wai Kin (Victor) Chan. “Evaluation of Energy Consumption in Block-Chains with Proof of Work and Proof of Stake”. In: *Journal of Physics: Conference Series* 1584.1 (July 2020), p. 012023. DOI: [10.1088/1742-6596/1584/1/012023](https://doi.org/10.1088/1742-6596/1584/1/012023). URL: <https://dx.doi.org/10.1088/1742-6596/1584/1/012023>.
- [553] Lianna Zhao et al. “Secure Access Control for DAG-Based Distributed Ledgers”. In: *IEEE Internet of Things Journal* 9.13 (2022), pp. 10792–10806. DOI: [10.1109/JIOT.2021.3128025](https://doi.org/10.1109/JIOT.2021.3128025).
- [554] Yuxi Zhao et al. “Data poisoning attacks and defenses in dynamic crowdsourcing with online data quality learning”. In: *IEEE Transactions on Mobile Computing* 22.5 (2021), pp. 2569–2581.

- [555] Shuyuan Zheng, Yang Cao, and Masatoshi Yoshikawa. “Secure Shapley Value for Cross-Silo Federated Learning”. In: *Proc. VLDB Endow.* 16.7 (Mar. 2023), pp. 1657–1670. ISSN: 2150-8097. DOI: [10.14778/3587136.3587141](https://doi.org/10.14778/3587136.3587141). URL: <https://doi.org/10.14778/3587136.3587141>.
- [556] Yao Zheng et al. “SHARP: Private Proximity Test and Secure Handshake with Cheat-Proof Location Tags.” In: *ESORICS*. Springer. 2012, pp. 361–378.
- [557] Jiawei Zhou et al. “Synthetic Lies: Understanding AI-Generated Misinformation and Evaluating Algorithmic and Human Solutions”. In: *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. CHI ’23. Hamburg, Germany: Association for Computing Machinery, 2023. ISBN: 9781450394215. DOI: [10.1145/3544548.3581318](https://doi.org/10.1145/3544548.3581318). URL: <https://doi.org/10.1145/3544548.3581318>.
- [558] Liyi Zhou et al. “SoK: Decentralized Finance (DeFi) Attacks”. In: *2023 IEEE Symposium on Security and Privacy (SP)*. 2023, pp. 2444–2461. DOI: [10.1109/SP46215.2023.10179435](https://doi.org/10.1109/SP46215.2023.10179435).
- [559] Zhichao Zhu and Guohong Cao. “Applaus: A privacy-preserving location proof updating system for location-based services”. In: *2011 Proceedings IEEE INFOCOM*. IEEE. 2011, pp. 1889–1897.
- [560] Zhichao Zhu and Guohong Cao. “Toward Privacy Preserving and Collusion Resistance in a Location Proof Updating System”. In: *IEEE Transactions on Mobile Computing* 12.1 (2013), pp. 51–64. DOI: [10.1109/TMC.2011.237](https://doi.org/10.1109/TMC.2011.237).
- [561] Zhichao Zhu and Guohong Cao. “Toward privacy preserving and collusion resistance in a location proof updating system”. In: *IEEE Transactions on Mobile Computing* 12.1 (2011), pp. 51–64.
- [562] Andreas Zolnowski, Towe Christiansen, and Jan Gudat. “Business Model Transformation Patterns of Data-Driven Innovations.” In: *European Conference on Information Systems*. Vol. 2016. Accessed: August 6, 2025. 2016, p. 146. URL: http://aisel.aisnet.org/ecis2016_rp/146.
- [563] Matthew A Zook and Joe Blankenship. “New spaces of disruption? The failures of Bitcoin and the rhetorical power of algorithmic governance”. In: *Geoforum* 96 (2018), pp. 248–255.
- [564] Shoshana Zuboff. *The age of surveillance capitalism : the fight for a human future at the new frontier of power*. eng. London: Profile Books, 2019. ISBN: 9781781256855.

Appendix A

An example of an MEV election

We illustrate a sample election of our proposed modification of MEV. Let us consider an election with three candidates: $A = \{a_i, a_j, a_k\}$ and only one winner is needed ($K = 1$). The possible outcomes are shown in Table A.1.

$\mathcal{O}$	o_1	o_2	o_3
Preferred $\mathcal{P}$	(a_i)	(a_j)	(a_k)
Non-Preferred $\mathcal{NP}$	(a_j, a_k)	(a_i, a_k)	(a_i, a_j)

Table A.1: The lower-cardinality ordering set when $A = \{a_i, a_j, a_k\}$ and $K = 1$. Agents in the same brackets are given the same rank in an ordering.

Each agent constructs a vote for their preferred outcome. The possible outcomes in this case can be o_1, o_2 or o_3 . A example of a vote for outcome o_1 is the pairwise preference matrix $S(o_1)$, displayed in (A.1).

$$S(o_1) = \begin{array}{c|ccc} & a_i & a_j & a_k \\ \hline a_i & 0 & 1 & 1 \\ a_j & 0 & 0 & 1/2 \\ a_k & 0 & 1/2 & 0 \end{array} \quad (\text{A.1})$$

Each agent constructs their vote. The election administrator (the agent tasked with tallying votes) receives all the votes and computes the average vote by aggregating all the votes (following A.2).

$$S(A) := \frac{1}{N} \sum_{a_i \in A} S(a_i). \quad (\text{A.2})$$

The administrator then solves the optimisation problem:

$$\begin{aligned}
\pi^* = \max_{\pi} & - \sum_{o \in \mathcal{O}} \pi(o) \log \pi(o) \\
\text{s.t.} & \sum_{o \in \mathcal{O}} \pi(o) \cdot S(o) = S(A) \\
& \sum_{o \in \mathcal{O}} \pi(o) = 1 \\
& \pi(o) \geq 0 \quad \forall o \in \mathcal{O}
\end{aligned} \tag{A.3}$$

This will return the probability of selecting each of the three possible outcomes:

$$\pi^* = \pi^*(o_1), \pi^*(o_2), \pi^*(o_3)$$

Suppose the administrator computes π^* from the optimisation problem in equations A.3. Assuming $\pi^*(o_1) = 0.3$, $\pi^*(o_2) = 0.4$ and $\pi^*(o_3) = 0.3$, to find the winning candidate ordering o^* , the administrator simply samples o^* from π^* . Consider a prize wheel as shown in Figure A.1. The wheel includes $|\mathcal{O}|$ wedges where each wedge represents one ordering o and takes the share of $\pi^*(o)$. To obtain the winning outcome ordering, simply spin the wheel and o^* is the wedge where the red arrow stops, i.e., o_1 in Figure A.1.

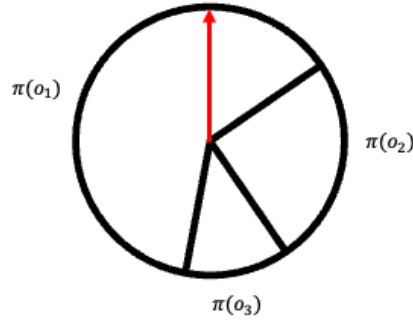


Figure A.1: A prize wheel for sampling an outcome ordering $o \in \mathcal{O}$ from a probability measure π .

To summarise:

1. Firstly, each agent a_i constructs their vote, the pairwise preference matrix $S(i)$, from the data point x and reputation $r_{i \rightarrow j}$.
2. Then, the election administrator (the agent running the election) calculates an average of all agents' pairwise preference matrix, $S(A)$.
3. Then, the administrator constructs a low-cardinality ordering set of agents $\mathcal{O}$ from K , the number of necessary winners needed.
4. For every possible ordering of candidates o , the administrator constructs a theoretical pairwise preference matrix, $S(o)$.

5. Then, the administrator solves the optimisation problem to maximise entropy as defined in equation A.3. The outcome is a probability measure π^* , of a given ordering. This probability measure also adheres to the *Representative Probability* property.
6. Finally, the administrator samples an outcome ordering o^* from π^* , using a "prize-wheel" sampling, as shown in Figure A.1. This ordering is the final election outcome.

Appendix B

Instances of Fraud in Web3 Applications

Mt. Gox

DLT application type: cryptocurrency exchange.

Years: 2011 to 2014.

Estimated amount lost: \$460 million.

Nature of event: attackers were able to siphon funds from the Mt. Gox exchange over the years leveraging a number of exploits. They gained unauthorized access to the private keys of Mt. Gox's wallets. This breach allowed them to transfer Bitcoins from the exchange's wallets to their own addresses.

Causes: The breach of Mt. Gox's accounts was facilitated by inadequate security measures, such as storing private keys without proper encryption [224]. Management was negligent, the CEO Mark Karpeles was found guilty of manipulating account data to inflate the value of Mt. Gox by \$ 33.5 million [201]. Mt. Gox also cites the issue of transaction malleability¹ as one of the culprits of their losses, but the authors of [155] disprove this claim.

OneCoin

DLT application type: cryptocurrency.

Years: 2014 to 2017.

Estimated amount lost: \$4 billion.

Nature of event: the OneCoin fraud was advertised as a revolutionary cryptocurrency, but in fact all transactions and tokens were handled by a centralised company server, there was no blockchain technology involved. The scam was both a Ponzi and a pyramid scheme. It relied on new investors to pay for the earlier ones, and it offered no real product [474]. Withdrawal of funds was guarded with high entry barriers and limitations: the exchange

¹Transaction malleability is a vulnerability in a blockchain network that allows an attacker to change a transaction's unique identifier before it's confirmed [26].

was handled privately by OneCoin through xcoinx.com centralised exchange. This site was often listed as ‘under maintenance’, and only allowed users to cash out up to 1.5% of the user’s total OneCoins [269, p. 12].

Causes: centralised and poorly audited power. The scheme was a fraud and relied on false advertising.

The DAO

DLT application type: DAO.

Year: 2016.

Estimated amount lost: \$60 million.

Nature of event: an attacker exploited the smart contract code in The DAO to set up an iterative loop of withdrawing funds leveraging a re-entrancy attack². This allowed the attacker to siphon funds from The DAO treasury [160].

Causes: a smart contract vulnerability.

Bitfinex

DLT application type: cryptocurrency exchange.

Year: 2016.

Estimated amount stolen: \$72 million.

Nature of event: attackers later identified by the US government were able to send transactions to their own address from other users’ wallets [502]. Bitfinex had a multi-signature wallet address service provided by BitGo, but this service itself was not hacked [499], according to BitGo [1]. According to Brett Johnson, former cybercriminal turned US Secret Services informant, the hackers had access to Bitfinex’s servers for months and modified the back-end code so it would send the attackers the users’ passwords as they (the users) logged on to the exchange [462]³. Chris Janczewski, lead investigator of the IRS on this case, states they then executed a script to transfer over the funds from users’ compromised wallets and deleted their access logs [462]⁴.

Causes: the exact technical vulnerability that enabled this hack is unclear [62]. The lead perpetrator is now under witness protection [96].

Bitconnect

DLT application type: cryptocurrency.

Years: 2016 to 2018.

Estimated amount lost: \$2.4 billion (for US investors alone) [503].

Nature of event: Bitconnect was another fraud that lured investors with unrealistic promises of investment returns. The premise of Bitconnect was encouraging users to lend their purchased Bitconnect tokens back to the platform, which would then be used to

²For an explanation of how a re-entrancy attack works, the reader may refer to the following sources: [173, p. 56], [144].

³Statement made at minute 6:35.

⁴Statement made at minute 6:45.

trade Bitconnect tokens against the price volatility of Bitcoin with an alleged trading bot. The trading bot was never publicly disclosed or audited, and it is unlikely it ever existed [440]. Bitconnect offered unrealistic investment returns: depending on how much the user invested, this could be from 47.5% a month to 570% a year, with a minimum guaranteed of 90% return a year [474], [434]. This interest could then also be compounded. Investors had to lock up their tokens for 120 to 299 days, and the majority of these were traded on Bitconnect’s own exchange, because Bitconnect imposed high transaction fees to anyone wishing to cash out their tokens. The scam operated as both a Ponzi and a pyramid scheme: it offered high referral bonuses to recruit new investors and invested large sums in extravagant marketing and advertising tactics [490] [440].

Causes: centralised and poorly audited power. This scheme was another fraud that also relied on false advertising.

Coincheck

DLT application type: cryptocurrency exchange.

Year: 2018

Estimated amount lost: \$530 million.

Nature of event: hackers infected a company terminal with malware to steal the private keys of the hot wallet⁵ of the exchange containing the funds [500].

Causes: inadequate security measures and storing large quantities of tokens in a hot wallet, known to be less secure than cold wallets [485].

QuadrigaCX

DLT application type: cryptocurrency exchange.

Year: 2019.

Estimated amount lost: \$190 million.

Nature of event: the founder of the exchange, Gerald Cotten, operated it as a Ponzi scheme, using new deposits to cover old withdrawals [429]. He continuously misappropriated client funds for his own personal use and transferred funds to accounts in other exchanges [386]. The exchange held insufficient funds to cover what was owed to clients. He was the sole individual with access to the private keys of the exchange’s holdings [34]. Cotten created fake accounts under aliases, credited with fictitious balances, to trade against real customers [386]. These fictitious trades artificially increased the platform’s trading volume, misleading investors about its liquidity and profitability. Audits revealed that QuadrigaCX’s cold wallets were empty, contradicting claims that funds were securely stored [546].

Causes: centralisation of power, lack of oversight and transparency and fraud. The court-appointed monitor, Ernst & Young, found no meaningful records of accounting or governance, highlighting the extent of mismanagement.

⁵A hot wallet is one that is connected to the internet, as opposed to a cold wallet, that is offline [81].

PlusToken

DLT application type: cryptocurrency and exchange.

Years: 2018 to 2019.

Estimated amount lost: \$4 billion.

Nature of event: very similar to the nature of OneCoin and Bitconnect. Scammers began cultivating member loyalty by offering advice on cryptocurrency trading in small group chats. They continued by hosting large events [103], and eventually launched their PlusToken platform: this included a native PlusToken, an exchange and a wallet service [474]. Similar to Bitconnect, PlusToken promised unrealistic amounts of profit if users deposited their PlusTokens in the PlusToken platform, and a novel trading bot would trade their tokens against other cryptocurrencies and make a profit based on arbitrage [128]. Users were charged higher fees if they wanted to cash out their tokens from the platform, and were offered substantial rewards for bringing in new referrals. The scheme was a pyramid scheme, with rewards for referrals going up to 10 levels of depth [59]. The founders stole funds from the exchange and left a message saying ‘sorry we have run’ [455].

Causes: I have been unable to find a source outlining the exact technical method that the scammers used to drain the funds from the PlusToken exchange. It can be speculated that they were able to steal funds because they likely held centralised access to the main exchange accounts, but to the best of my knowledge, there is no official PlusToken documentation, or government investigation sources that outline how funds were siphoned.

KuCoin

DLT application type: cryptocurrency exchange.

Year: 2020.

Estimated amount lost: \$280 million [476].

Nature of event: the hackers obtained private keys to KuCoin’s hot wallets, enabling them to transfer funds to their own addresses [299].

Causes: lack of security audits to identify vulnerabilities in private key management, storage and access.

Poly Network

DLT application type: cross-chain network.

Year: 2021.

Estimated amount lost: \$600 million.

Nature of event: an attacker exploited the mismanagement of access rights in two important smart contracts of the Poly Network: one that had the right to execute cross-chain transactions (‘EthCrossChainManager’, or M in short), and one that kept a record of the public keys of the ‘keepers’, that are nodes that manage wallets and authenticate transactions coming from other blockchains (‘EthCrossChainData’, or D in short). Contract M actually grants edit rights into contract D, or in other words, contract D can be accessed through contract M. This is the first vulnerability. The second is that through contract

M, the public keys listed in contract D could be overwritten, the attacker could pass a maliciously crafted input to M, which then called D's function to overwrite the keeper keys [203]. At a high level, the attacker sends a transaction request that M verifies against the keeper keys defined in D, but the attacker is able to construct their own key pair that is written into D, and M therefore accepts because a) M checks a variable that takes a user-defined input, so the attacker can find an input that will get accepted; b) the attacker can access contract D through M and c) finding a valid user input is greatly facilitated due to the low security level of the contract check [296].

Causes: the mismanagement of access rights within two crucial smart contracts of the Poly Network, a flaw in the logic of the contract that greatly reduced the difficulty of a brute force attack⁶ and allowing an important input variable of the smart contract to be user-defined. These oversights can be summarised with a lack of security audits of smart contracts.

Africrypt

DLT application type: investment firm.

Year: 2021.

Estimated amount lost: \$3.6 billion at peak valuation.

Nature of event: the founders claimed the firm's wallets had been hacked and drained. The siphoned funds were obfuscated through cryptocurrency mixers. Reports alleged that some of the wallets into which the stolen funds were transferred into were owned by the founders, although this has not been officially confirmed [60]. The founders subsequently went missing, and one was later arrested in Switzerland, 'trying to access safety deposit boxes they rented' [516]. The founders denied involvement, although employees of the firm reported losing access to the exchange's back-end platforms a week before the alleged hack [60]. The founders have also discouraged victims from pursuing legal action [511], and after disappearing purchased citizenship from Vanuatu and created a shell company in UK named Clandestine Limited [529].

Causes: unknown. There has been no formal evidence that the founders siphoned the accounts, but also no proof that the accounts were hacked.

Ronin Network

DLT application type: cross-chain bridge.

Year: 2022.

Estimated amount lost: \$620 million.

Nature of event: attackers targetted the private keys of five out of the nine necessary accounts with the power to validate transactions [493]. With this they were able to approve transactions moving funds out of the network [178]. Although Ronin Network alleged that it was likely a social engineering attack, it was later reported that they had centralised the ability to approve transactions in order to service the increase in demand of the network

⁶This is because the attacker only needed to find an input that produced a collision for first 4 bytes of the output [296].

[493, 61]. It was reported that this choice lead them to ‘loosen security procedures’ which were not later tightened.

Causes: increased centralisation and lack of security audits.

FTX

DLT application type: cryptocurrency exchange.

Years: 2022

Estimated amount lost: \$8 billion.

Nature of event: the exchange was operating fraudulent practises on many levels. Alameda Research was a trading firm funded by the FTX founder, Sam Bankman-Fried, led by himself and his close colleagues. Alameda received preferential treatment in the execution of trades on the FTX platform (allowing Alameda trades to front-run those of the customers on FTX) [342]. It also contained a bug for 6 months that meant that when a customer withdrew funds on FTX, Alameda’s debt did not change [342]. Alameda’s debt to creditors was repaid with customer funds. The founder specifically requested from developers to include a function that granted permission to pay for various FTX expenses from Alameda’s accounts. Later it was reported that Alameda’s funds were held in FTT, a native token of the FTX exchange, as opposed to in stable-coins or fiat currency [17]. This triggered the panic in customers to withdraw their funds, only to find that the exchange was not solvent to pay back customers, as they had a negative balance of \$8 billion.

Causes: fraudulent practises, lack of corporate governance, audits and misallocation of customer funds [365].