Rapid, User-Transparent, and Trustworthy Device Pairing for D2D-Enabled Mobile Crowdsourcing

Cong Zhao, Shusen Yang, Xinyu Yang, and Julie McCann

Abstract—Mobile Crowdsourcing is a promising service paradigm utilizing ubiquitous mobile devices to facilitate large-scale crowdsourcing tasks (*e.g.* urban sensing and collaborative computing). Many applications in this domain require Device-to-Device (D2D) communications between participating devices for interactive operations such as task collaborations and file transmissions. Considering the private participating devices and their opportunistic encountering behaviors, it is highly desired to establish secure and trustworthy D2D connections in a fast and autonomous way, which is vital for implementing practical Mobile Crowdsourcing Systems (MCSs). In this paper, we develop an efficient scheme, Trustworthy Device Pairing (TDP), which achieves user-transparent secure D2D connections and reliable peer device selections for trustworthy D2D communications. Through rigorous analysis, we demonstrate the effectiveness and security intensity of TDP in theory. The performance of TDP is evaluated based on both real-world prototype experiments and extensive trace-driven simulations. Evaluation results verify our theoretical analysis and show that TDP significantly outperforms existing approaches in terms of pairing speed, stability, and security.

Index Terms—Mobile Crowdsourcing, D2D Communications, User-Transparent Pairing, Trustworthiness.

1 Introduction

SMART mobile devices have promoted the proliferation of Mobile Crowdsourcing Systems (MCSs), which facilitates large-scale crowdsouring tasks by exploiting the capacities of ubiquitous mobile users in a collaborative way [1]. Typical MCS applications include mobile crowdsensing [2], [3], [4], [5], information searching [6], crowdsourced content caching and sharing [4], [5], and collaborative mobile could computing [7], [8], *etc.* Many of these applications require exploiting the collaborative interactions among individual participants in geographical proximity. For instance, in collaborative searching applications (*e.g.* finding a missing child [6]), local searching results are required to be shared among surrounding participators.

Such geographically proximal interactions can be achieved by using Device-to-Device (D2D) communication (*e.g.* WiFi direct and Bluetooth), which is much more agile and cost-effective compared with using traditional cellular networks, due to its shorter transmission delay, lower power consumption, and ignorable financial cost [4], [9]. For instance, in D2D-enabled mobile crowdsensing systems [4], [10], [11], [12], a participating mobile user can report his or her sensory data to the server via other participating phones through opportunistic D2D transmissions rather than through cellular communications directly, which significantly improves the network throughput and reduces the system financial cost. It has also been shown that D2D communication is promising in cellular traffic offloading in crowdsourced video streaming [1], [2] and content sharing in mobile social networking [5]. In addition, D2D communications are considered as the fundamental communication infrastructure in more and more emerging mobile could computing architectures [6], [7].

There exist an increasing number of efforts from both academia and industry to develop D2D related communication techniques, including efficient neighbor discovery [13], secure device pairing [14], [15], [16], D2D enabled cellular networks [17], incentivisation for D2D communications [4], [9], [18], and D2D standards such as WiFi direct, Bluetooth Smart, and LTE direct. However, these approaches cannot be directly applied to achieve D2D-enabled MCSs, because they cannot simultaneously satisfy the following three requirements in practice:

1. **Secure D2D Connection.** Since participating devices are privately held and temporarily recruited, there is no prior trust relation between them. Therefore, establishing a shared secret key for the firstly encountered devices (*i.e.* device pairing) is vital for secure collaborative interactions.

2. **Rapid and User-Transparent Connection.** Due to the opportunistic human contact behaviors, it is highly desired to achieve rapid, autonomous, and user-transparent device pairing processes. However, existing device pairing approaches are based on either physical interactions (*e.g.* button clicks) between firstly encountered devices or common contextual information gathered gradually [15], [16], and are therefore unsuitable for practical MCSs.

3. **Addressing Peer Diversity.** In a MCS, a participating device could have multiple potential D2D peer devices and need to choose the optimal one (or a set of optimal ones). This optimization decision should depend on the trustworthiness and other metrics (*e.g.* transmission data rate) of each candidate. As a result, a mechanism estimating the trustworthiness of a certain device as a D2D peer is

- *C. Zhao, S. Yang (Corresponding Author), and X. Yang are with Xi'an Jiaotong University, P.R.China. E-mail: zhaocong@stu.xjtu.edu.cn, cuiangmu@gmail.com, yxyphd@mail.xjtu.edu.cn*
- *J. McCann is with the Department of Computing, Imperial College London, UK. E-mail: j.mccann@imperial.ac.uk*

necessary to guarantee the quality of D2D communications.

1.1 Contribution

In this paper, we present both theoretical and practical studies to simultaneously achieve above three requirements. Our contributions are summarized as follows:

- We develop Trustworthy Device Pairing (TDP), the first device pairing scheme for trustworthy opportunistic D2D communications in MCSs. TDP achieves rapid and user-transparent device paring, which is not supported by the off-the-shelf protocols or state-of-art user-transparent approaches, without introducing extra pairing labors. In addition, TDP introduces a new trust management method for reliable peer device selections based on online trustworthiness estimation of D2D peers.
- Through rigorous security analysis, we demonstrate that TDP is immune to six potential security threats, including *Passive Eavesdropping attacks*, *Impersonating attacks*, *Man-in-the-Middle attacks*, *Trust Forging attacks*, *Independent Negative Attacks*, and *Collusive attacks*. Particularly, besides guaranteeing a comparable security intensity to the off-the-shelf protocols, TDP can also resist the *Traffic-Oriented attacks* that are detrimental to D2D communications in MCSs.
- We implement TDP in the Android operating system and the OMNeT++ simulator, and conduct real-world experiments and extensive trace-driven simulations to evaluate the performance of TDP. Results demonstrate that TDP significantly outperforms existing approaches in terms of pairing speed and stability, and it can effectively eliminate the impact of *Traffic-Oriented attacks* in D2D-enabled MCSs.

1.2 Paper Organization

The next section presents related work. Section 3 presents the system model. The device trust estimation method in D2D-enabled MCSs is presented in Section 4. Section 5 presents the Trustworthy Device Pairing scheme. Real-world experiments and extensive simulations are discussed in Section 6. And we conclude this paper in Section 7. Discussions on parameter determinations and security proofs are placed in Appendices A and B respectively, which can be found in the supplemental material.

2 Related Work

D2D Communications: There exist a large body of D2D communication based schemes, including efficient neighbor discovery [13], D2D-enabled cellular networks [17], incentivisation for D2D [4], [9], [18], network protocols [19], [20], etc. Among those, the work [4], [9], [21] specifically consider MCSs with D2D communications. However, none of these work considers security issues.

Device Pairing: Device pairing focuses on establishing shared keys between freshly encountered devices. Some approaches [22], [23], [24], [25] rely on user-interactions to distribute initial trust credentials. However, in D2D-enabled MCSs, mobile users are normally strangers who are unlikely to conduct physical interactions. Since user-transparency is more preferable for spontaneous pairing, the proximity-based approach [14], [15], [16] attracts significant attentions, which authenticates adjacent devices leveraging locally sensed contextual information. However, this approach commonly assumes that adversaries have no sustained access to legal contextual information, which is not acceptable for highly open MCSs. One of the most relevant work is the non-interactive device pairing approach in [26], which, however, requires asynchronous broadcasting of public credentials that may lead to unpredictable pairing time fluctuations. In summary, none of No existing device pairing approach can achieve both secure and user-transparent D2D connections, as our TDP.

Trust Management: For participatory distributed systems, trust management is an effective approach to enhance system performance by determining the potential gain of self-interested entities according to their trustworthiness [27], [28]. In traditional P2P systems, the trust framework is usually applied for either enabling access control [29] or encouraging benign resource sharing [30]. SepRep [31] establishes a general Quality-of-Service (QoS) based trust framework for heterogeneous P2P networks. For mobile P2P networks, [32] proposes M-trust for accurate, robust and light-weight trust rating aggregation. In proliferating MCS applications, existing trust frameworks [33], [34] mainly focus on evaluating the participator contribution from the data quality perspective. In [35], a reputation-based trust framework is used to estimate the trustworthiness of a mobile device in location authentications, which is collaboratively conducted by proximal devices using Bluetooth communications. Overall, existing approaches maintain application-specific trustworthiness for self-interested entities, which is usually estimated using upper-level metrics without considering link-layer constrains. They cannot be directly used to evaluate the trustworthiness of D2D communications in MCSs.

The workshop abstract [36] presents the very initial idea of our TDP, while this paper establishes a much more complete and systematic trust management scheme with comprehensive theoretical analysis. Furthermore, real-world experiments and extensive trace-driven simulations are also conducted to study the practical performance of TDP in realistic D2D-enabled MCSs.

3 System Model

In this section, we present the model of general D2D-enabled MCSs and potential attacks to opportunistic D2D communications.

3.1 D2D-enabled MCS Architecture

The typical architecture of a D2D-enabled MCS is shown in Fig.1: multiple participating personal mobile devices communicate with the Backend Server (BS) through the Internet via WiFi or cellular access points, and they communicate with devices opportunistically encountered through outband D2D radios such as Bluetooth and WiFi direct.

Generally, the BS is responsible for the maintenance of the entire MCS by establishing system regulations, publishing crowdsourcing applications, publishing tasks and

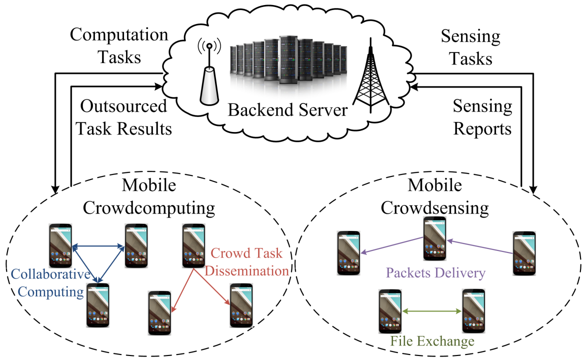


Fig. 1: Architecture of D2D-Enabled MCSs.

recording device profiles. Let $\mathcal{N}$ be the set of all regulated MCS task types (*e.g.* mobile social networking, crowdsensing, collaborative computing).

To participate in crowdsourcing tasks, personal mobile devices need to install the crowdsourcing application and register to the BS. Let $\mathcal{D}$ be the set of all registered devices. Driven by a specific crowdsourcing task, any pair of registered devices can establish a secure D2D connection for further communications, which is denoted as a D2D transaction for the rest of the paper. A device $a \in \mathcal{D}$ can have a sequence of D2D transactions $i = \{1, 2, ...\}$ with any other device in $\mathcal{D}$.

For the establishment of secure D2D connections, each device $a \in \mathcal{D}$ possesses a unique device pairing credential, which is verifiable to both the BS and any other registered device encountered. Meanwhile, a 'trustvalue' is maintained for each device according to its behavior in previous D2D transactions, which is treats as its trustworthiness to fulfill future transaction requests. Here, the 'trustvalue' is the predominant concern of the peer selection when there are multiple candidates.

3.2 Security Threats and Attack Models

Considering the sensitivity of personal mobile devices, *neither devices opportunistically encountered nor the BS should be fully trusted in MCSs*.

Intuitively, potential attacks launched by malicious devices within the D2D-enabled MCS can be divided into two categories: Connection-Oriented (CO) attacks and Traffic-Oriented (TO) attacks. On one hand, CO attackers intend to compromise the pairing process between encountered devices for illegal communication contents or MCS services. In fact, the prevention of CO attacks is the predominant concentration of security mechanism of off-the-shelf D2D communication protocols (*e.g* Bluetooth [25] and WiFi direct [24]). On the other hand, the TO attackers are more interested in manipulating D2D traffics in MCSs for simply unfair rewards (*e.g.* popularity, credits) or future advanced attacks (*e.g.* false data injection attacks). Unlike CO attacks, as far as we know, there is still no systematic discussion on the impact and prevention of TO attacks in MCSs from the perspective of opportunistic D2D communications.

We formally model these potential attacks as follows (let there be devices $a, b, m \in \mathcal{D}$):

1. Passvie-Eavesdropping Attack (CO_1): any third party adversary m, who is not the intended end a or b of a D2D transaction, tries to decrypt eavesdropped data;

2. Impersonating Attack (CO_2): any adversary m, except the device impersonated a, tries to use a's credential to establish D2D connections with others;

3. Man-in-the-Middle Attack (CO_3): any third party adversary m, except both ends of a D2D transaction a and b, tries to intercept and replace a (b)'s credential to establish D2D connections with b (a) in middle without being noticed;

4. Trust Forging Attack (TO_1): any adversary m forges its trustvalue to attract unfair D2D transaction requests;

5. Independent Negative Attack (TO_2): any adversary m intentionally replies negative ratings (continuously or intermittently) to downgrade others' trustvalue and consequently attract unfair D2D transaction requests;

6. Collusive Attack (TO_3): any set of adversaries $\mathcal{M}$ intentionally reply positive ratings to members within $\mathcal{M}$ and negative ratings to others (continuously or intermittently) to attract unfair D2D transaction requests.

Moreover, we should treat the BS as 'benign but curious': generally, it will jeopardize neither the device pairing nor trust management process for public credibility; meanwhile, it is curious about the content of D2D transactions among devices. In this case, the BS may try to launch CO_1 attacks as a third party adversary.

4 Device Trust Estimation in D2D-Enabled MCSs

In D2D-enabled MCSs, the reliability of a device to fulfill future D2D requests can be reflected by its behavior in historical transactions. In this paper, we denote this reliability as the trustworthiness of a device, which is treated as the basic metric for the peer selection in device pairing process (will be discussed latter).

In this section, we develop an online method to: (1) accurately estimate the trustworthiness of participating devices according to their historical behaviors; (2) selectively profile the device behavior during D2D transactions; (3) adjust the device trustworthiness using behavior profiles at real-time.

4.1 Device Trustworthiness

We use the term of 'trustvalue' to quantitatively represent the trustworthiness of each device, and, for the authenticity concern, it can only be adjusted by the BS after each transaction according to the device behavior.

Mathematically, let the trustvalue of each device $a \in \mathcal{D}$ after its i^{th} D2D transaction be a $|\mathcal{N}|$-dimensional vector $\boldsymbol{t}_a(i)$. Here, the value of the n^{th} $(1 \leqslant n \leqslant |\mathcal{N}|)$ entry of $\boldsymbol{t}_a(i)$ is a value within the range of $[0,\ 1]$, which represents the trustworthiness of a fulfilling the n^{th} type of D2D request (for the corresponding type of MCS task).

Since the trustworthiness of a device should be gradually built over a period of benign behaviors, we use the widely-adopted Gompertz function [37] in reputation mapping to compute the device trustvalue:

$$\boldsymbol{t}_a(i) = G(G^{-1}(\boldsymbol{t}_a(i-1)) + \Delta\boldsymbol{t}_a(i)), \tag{1}$$

where, for a given vector $\boldsymbol{x}$, $G(\boldsymbol{x})$ denotes calculating the value of the Gompertz function:

$$g(x) = e^{-e^{-c_g x}} \tag{2}$$

for each entry of vector $\boldsymbol{x}$, and the parameter c_g determines the sensitivity of trustvalue variation along time. $\Delta \boldsymbol{t}_a(i)$ is the estimation of a's behavior at its i^{th} transaction, which is computed by the BS according to behavior profiles uploaded by participating devices (will be explained later).

4.2 Device Behavior Profiling during D2D Transactions

For each D2D transaction, both ends of the transaction profile necessary information that can objectively reflect each other's behavior, which is then recorded in a D2D receipt respectively for future trustvalue adjustment at the BS. Intuitively, for a reasonable profiling of the i^{th} D2D transaction of a device $a \in \mathcal{D}$ (with its peer device b), the following four factors should be recorded in a's receipt (omitting index i for concision): the Quality-of-Service (QoS) provided by the D2D link q_{ab}, the credibility of b from a's perspective c_{ab}, the rating from b on a's behavior r_{ba}, and the type of the current transaction $\boldsymbol{\lambda}_a$. Following is their specific definitions.

QoS: We use the link-layer QoS value q_{ab} to denote the quality of the D2D link constructed by a given pair of encountered devices a and b. Different QoS metrics are used according to different D2D transactions, such as transmission delay, data rate, or packet loss rate. For example, image-based applications would require high data rate, while event-monitoring tasks need small delay. In our system, q_{ab} is a normalized value that can be obtained by both a and b during their transactions.

We use link-layer QoS because it can precisely reflect the application-layer QoS (due to the one-hop nature of D2D transactions), and it is much easier to be estimated in practice. For instance, if the minimal possible delay for transmitting a given-size packet over a D2D link is 10 ms (computed using datasheet), and the actual per-packet delay is 40 ms (measured online), then the QoS value can be normalized as $q_{ab} = 10/40 = 0.25$.

Credibility: The credibility estimation between encountered devices is important for the evaluation of current behavior of each other, considering their historical knowledge. In practice, it is reasonable for a person to consider a stranger credible if their judges to any common third party are similar. Inspired by this, we estimate c_{ab} based on the comparability of contact histories of both device a and device b. Without loss of generality, let device a's contact history $\mathcal{E}_a$ be a set of devices that a has paired with during a trust management cycle regulated by the BS. For each device $x \in \mathcal{E}_a$, a value e_{ax} that denotes the percentage of positive ratings from a to x is also maintained by a, which is recorded in a vector $\boldsymbol{\theta}_a$ according to the encountering order of device x. e_{ax} will be refreshed according to the rating from device a to device x whenever they finish a D2D transaction. Specifically, c_{ab} is defined as:

$$c_{ab} = (\alpha s_{ab}^2 + (1-\alpha) d_{ab}^2) w_{ab}, \tag{3}$$

where s_{ab} is the similarity between percentages of positive ratings separately replied by a and b to other devices, d_{ab} is the relative diversity of percentages of positive ratings separately replied by a and b to other devices, α within the range of $[0,\ 1]$ is used to adjust the focus of the credibility estimation, and w_{ab} is the damping factor that is inversely proportion of the intimacy between a and b.

In Eq.(3), s_{ab} and d_{ab} are defined as:

$$s_{ab} = 1 - \sqrt{\frac{\sum_{x \in \mathcal{E}_{ab}} (e_{ax} - e_{bx})^2}{|\mathcal{E}_{ab}|}}, \tag{4}$$

$$d_{ab} = 1 - |H(\boldsymbol{\theta}_a) - H(\boldsymbol{\theta}_b)|, \tag{5}$$

where $\mathcal{E}_{ab} = \mathcal{E}_a \cap \mathcal{E}_b$, and $H(\boldsymbol{\theta})$ is calculated as:

$$H(\boldsymbol{\theta}) = \sqrt{\frac{\sum_{1 \leqslant y \leqslant dim(\boldsymbol{\theta})} (\theta_y - \theta_{(y+1) \bmod dim(\boldsymbol{\theta})})^2}{dim(\boldsymbol{\theta})}}, \tag{6}$$

where $dim(\boldsymbol{\theta})$ is $\boldsymbol{\theta}$'s dimensions, and θ_y is $\boldsymbol{\theta}$'s y^{th} entry. w_{ab} is used to restrict the credibility estimation between devices that contact more frequently than normal, which is defined as:

$$w_{ab} = \begin{cases} 1 & \text{if } \sigma_{ab} = 0 \\ -c_w(\sigma_{ab}/\bar{\sigma})^2 + 1 & \text{if } 0 < \sigma_{ab} \leqslant \lfloor \sqrt{1/c_w}\bar{\sigma} \rfloor \\ 0 & \text{if } \sigma_{ab} > \lfloor \sqrt{1/c_w}\bar{\sigma} \rfloor \end{cases}, \tag{7}$$

where parameter c_w determines the sensitivity of credibility damping, σ_{ab} is the transaction count between a and b, and $\bar{\sigma}$ is the average transaction count of all devices within a trust management cycle regulated by the BS.

Rating: The rating indicates the satisfactory of a device on the behavior of its peer device in the current D2D transaction, which is determined by the comparison of the QoS and credibility of the current transaction with that of historical transactions with any device. For the transaction between devices a and b:

$$r_{ab} = \begin{cases} 1 & \text{if } \beta q_{ab}^2 + (1-\beta) c_{ab}^2 \geqslant \beta \bar{q}_a^2 + (1-\beta) \bar{c}_a^2 \\ -1 & \text{if } \beta q_{ab}^2 + (1-\beta) c_{ab}^2 < \beta \bar{q}_a^2 + (1-\beta) \bar{c}_a^2 \end{cases}, \tag{8}$$

where $\bar{q}_a$ and $\bar{c}_a$ are the EWMAs of historical QoS and credibility that are locally updated by device a after each transaction, and β within the range of $[0,\ 1]$ is used to adjust the focus of the rating decision.

Transaction Type: The type of the current transaction is profiled to estimate the trustworthiness of a device fulfilling D2D requests from different type of MCS tasks respectively.

For a normalized and comparable profiling, we define $\boldsymbol{\lambda}_a$ as a $|\mathcal{N}|$-dimensional vector to represent a D2D transaction that is initiated from device a. If the current transaction is a type-n transaction, the n^{th} entry of $\boldsymbol{\lambda}_a$ is a positive value within the range of $(0,\ 1]$, while all other entries are 0. The value of the n^{th} entry indicates resources (computing/communicating) that are contributed by such a transaction, *e.g.* bandwidth, which is determined according to the criteria regulated by the BS.

4.3 Device Trustworthiness Adjustment at the BS

According to the uploaded receipts, the BS can estimate the behavior of a device in a D2D transaction. For instance, for the i^{th} transaction of a device $a \in \mathcal{D}$ (with its peer device b), the BS can obtain q_{ab}, c_{ab}, r_{ba} and $\boldsymbol{\lambda}_a$ from a's receipt.

From the BS's perspective, one more thing should be considered is that D2D transactions required by different crowdsourcing tasks lead to different device trustvalue adjustment patterns (*i.e.* which end of the transaction should have a trustvalue adjustment). For instance, for an outsourced collaborative computing task, both ends of the D2D transaction should have trustvalue adjustments because of their equal contributions to task offloading.

Without loss of generality, for $\boldsymbol{\lambda}_a$, the trustvalue adjustment pattern $\boldsymbol{p}_a$ is defined as an element of set $\mathcal{P} = \{(0,0),(0,1),(1,0),(1,1)\}$ that indicates whether the initiator (indicated by the 1^{st} entry, role $\boldsymbol{r}_a = (1,0)$) or the peer (indicated by the 2^{nd} entry, role $\boldsymbol{r}_a = (0,1)$) should have a trustvalue adjustment. Let Λ be the whole set of the transaction types ($\boldsymbol{\lambda}$). It is viable for the BS to establish a mapping:

$$f : \Lambda \rightarrow \mathcal{P} \tag{9}$$

that indicates which end of the transaction should have a trustvalue adjustment according to the transaction type.

In this case, for the BS, the behavior estimation of device a in its i^{th} transaction $\Delta \boldsymbol{t}_a(i)$ in Eq.(1) is computed as (omitting index i for concision):

$$\Delta \boldsymbol{t}_a = q_{ab} c_{ab} r_{ba} f(\boldsymbol{\lambda}_a) \boldsymbol{r}_a \boldsymbol{\lambda}_a. \tag{10}$$

4.4 Trustvalue Bootstrapping

For the bootstrapping of the device trustvalue, an initial value of e^{-1} ($g(0)$) is issued when the device joins the MCS. Besides, for two devices freshly encountered (*e.g.* a and b, where $|\mathcal{E}_a|$, $|\mathcal{E}_b|$ or $|\mathcal{E}_{ab}|$ is 0), the weight factor of rating β is set as 1 to neglect the credibility factor. Therefore, any device that freshly joins the MCS will not be starved and is guaranteed a fair probability to be involved as long as behaving well in D2D transactions.

4.5 Discussion on the Trust Estimation Method

In this subsection, we firstly provide justifications for the trustvalue definition and device behavior profiling, then we conduct comparisons between our method and the Josang model [38] in the D2D-enabled MCS scenario.

4.5.1 Device Trustvalue Accumulation

Intuitively, we estimate the trustworthiness of a device performing D2D transactions according to its historical behaviors. The trustvalue of a device is determined by the accumulation of its quantified behavior profiles (Eq.(1)).

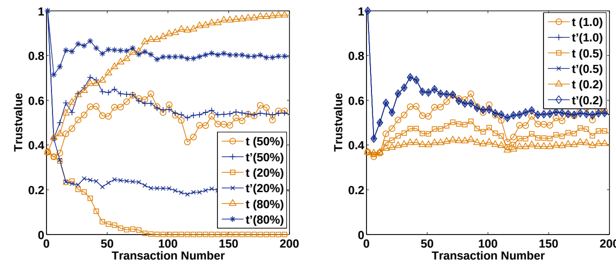
Considering that any freshly joint device with continuously positive (negative) behaviors deserves a rapid trustvalue enhancement (decrement), and that the trustvalue itself should be normalized for effective comparisons, we use a type of sigmoid function, *i.e.* the Gompertz Function [37] $g(x) = e^{-e^{-c_g x}}$ (see Fig.A1 (a) in Appendix A.1), to map the behavior accumulation to the device trustvalue. Any other function with similar properties, *e.g.* the Logistic Function [39] $l(x) = 1/(1+e^{-c_l x})$ (see Fig.A1 (a) in Appendix A.1), is also applicable in trustworthiness mapping.

One thing should be noted is that the value of c_g determines the sensitivity of trustvalue variations. We provide an instance for c_g determination in Appendix A.1.

4.5.2 Device Credibility Estimation

Intuitively, to stimulate more D2D transactions, devices with benign and stable behaviors in extensive active scopes should be guaranteed higher credibilities.

The first metric of credibility estimation is the comparability of the behavior profiles separatively maintained by both ends of the transaction. We assume that devices strictly following MCS regulations receive relatively similar ratings from different devices, then the comparability is quantified by the similarity (Eq.(4)) and the relative diversity (Eq.(5)) of their contact histories. Specifically, the similarity estimation encourages devices to objectively reflect other's behavior through justifiable ratings, while that the diversity estimation restricts devices from downgrading other's trustvalue with extreme or discriminative ratings.



(a) Impact of Positive Ratings (b) Impact of D2D Profiles

Fig. 2: Comparisons between TDP and the Josang Model.

Besides, to prevent the potential trustvalue boost caused by massive meaningless D2D transactions between collusive devices, the intimacy between both ends of a transaction is treated as a damping factor of their credibility (Eq.(7)), *i.e.* suspiciously frequent contacts lead to rapid credibility decreases. We use a type of monotonic descending function, *i.e.* a quadratic function $w(\sigma) = -c_w(\sigma/\bar{\sigma})^2 + 1$, to represent the relation between the device contact number σ and the damping factor w (see Fig.A2 (a) in Appendix A.2). Any other function with similar properties, *e.g.* a cubic function $w(\sigma) = -c_w(\sigma/\bar{\sigma})^3 + 1$ (see Fig.A2 (a) in Appendix A.2), is also applicable.

One thing should be noted is that the value of c_w determines the sensitivity of credibility damping on the device intimacy. We provide an instance for c_w determination in Appendix A.2.

4.5.3 Comparison with the Josang Trust Model

In this part, we compare the effectiveness of our trust estimation method and that of the Josang trust model [38], one of the most representative trust estimation methods, in D2D-enabled MCS scenarios.

Specifically, the Josang model [38] leverages the beta probability density function to map the reputation of self-interested entities, whose expectation is treated as entity trustworthiness on fulfilling future transactions. According to [38], we calculate the trustvalue based on the Josang model of each device a after its i^{th} transaction as:

$$t'_a(i) = \frac{Count_p(i) + 1}{Count_p(i) + Count_n(i) + 2}, \tag{11}$$

where $Count_p(i)$ and $Count_n(i)$ denote the number of positive and negative ratings that a receives after its i^{th} transaction, respectively.

For comparison, we conducted two sets of numerical simulations to study the impact of the positive rating percentage and the D2D profile on the accuracy and sensitivity of trustworthiness mapping of our method and the Josang model. For each round of simulation, we recorded the trustvalue variation (t for our method, and t' for the Josang model) of a single device a in 200 transactions. For our method, we set $t_0 = e^{-1}$, and t was updated after each transaction based on the rating and D2D profiles (*i.e.* QoS and credibility); for the Josang model, we set $t'_0 = 1$, and t' was updated after each transaction based on the rating only.

In the first set of simulations, we set that device a got a positive rating at different probabilities (*i.e.* 50%, 20% and 80%) and a random D2D profile within the range of $[0, 1]$ in a transaction. As shown in Fig.2 (a), both models can maintain a distinguishable trustvalue with a similar adjusting trend. However, their sensitivities are obviously different: for more extreme cases (*i.e.* 20% and 80%), our method guarantees a gradual accumulation of the device trustworthiness, whereas the Josang model just presents a rapid trust convergence; for the neutral case (*i.e.* 50%), our device trustworthiness is adjusted more sensitively because of more objective device behavior estimations.

In the second set of simulations, we set that device a got a positive rating with a 50% probability and a random D2D profile within different ranges (*i.e.* $[0, 1]$, $[0, 0.5]$ and $[0, 0.2]$) in a transaction. As shown in Fig.2 (b), TDP manages to maintain a more distinguishable and sensitive trustvalue compared to the Josang model. In D2D-enabled MCSs, it is not objective to estimate the trustworthiness of a device only based on ratings from other peers. Counting more detailed information into trustworthiness estimation not only guarantees a better objectivity but also provides devices a fair chance to attract more transactions with better behaviors.

5 The Trustworthy Device Pairing (TDP) Scheme

With the accurate trustworthiness estimated by the method in Section 4, in this section, we develop TDP to realize rapid, user-transparent, and trustworthy device pairing for D2D-enabled MCSs.

5.1 Overview

Considering the security threats in Subsection 3.2, TDP establishes a Certificate-Less Public Key Cryptography (CL-PKC) [40] framework to issue a unique private-public key pair to each registered device as the D2D credential. Since the credential is collaboratively generated by the BS and the registering device, it is verifiable to any other registered device as well as the BS.

Based on the credential, any pair of registered devices encountered can negotiate a shared key spontaneously without contacting the BS, which introduces no extra labor compared with existing device pairing methods. Besides, because a part of the credential is privately held by the device only, the 'benign but curious' BS cannot restore such shared keys to decrypt the transaction contents. Moreover, according to the D2D receipts authenticated by the credential, participating devices of D2D transactions can have justifiable trustvalue adjustments from the BS based on their behaviors, which will guarantee a well-behaving device a fair advantage in peer selections in future D2D transactions.

Specifically, TDP consists of three components: *device registration*, *device pairing*, and *device trustvalue management*, whose basic procedure is illustrated in Fig.3. Specific operations are presented as follows.

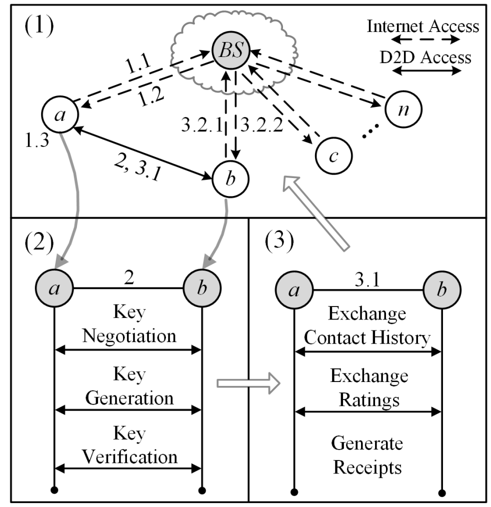


Fig. 3: Architecture of TDP.

TABLE 1: System Parameters of TDP.

Parameter	Meaning	Note
E	an elliptic curve on a finite filed F_q	treated as a cyclic addition group G_q; q: a large prime as the group order; P: the generator of G_q
x	a random positive integer $< q$ ($x \in_R Z_q^*$)	the master private key, only possessed by the BS
P_{pub}	a point on E ($P_{pub} = xP$)	the master public key; xP: the point multiplication on E
$h_{0,1,2}$	one-way hash functions	$h_0 : \{0,1\}^* \to Z^*$; $h_1 : \{0,1\}^* \to \{0,1\}^*$; $h_2 : \{0,1\}^* \to Z^*$

5.2 Device Registration

In this process, the BS generates and issues a D2D credential to each registering device.

For the system initialization, the BS determines all system parameters in Table 1. Then, the parameter set $\Omega = \{F_q, G_q, P, P_{pub}, h_0, h_1, h_2\}$ and the bootstrapping trustvalue $\boldsymbol{t}_0$ are deployed in the crowdsourcing application.

To join the MCS, personal mobile devices need to install the crowdsourcing application and register to the BS. Without loss of generality, let device a register to the BS. Specific operations (Step 1.1-1.3 in Fig.3) are demonstrated in Fig.4, where the operator'$\|$' denotes the concatenation operation.

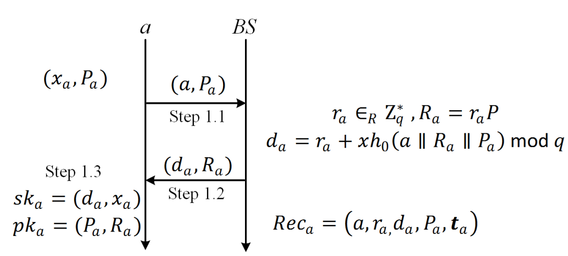


Fig. 4: Device Registration (Step 1 in Fig.3).

Here, device a determines its local partial key pair (x_a, P_a), where $P_a = x_a P$. Then the BS extracts an authenticated partial key pair (d_a, R_a) according to P_a. Finally, (sk_a, pk_a), where $sk_a = (d_a, x_a)$ and $pk_a = (P_a, R_a)$,

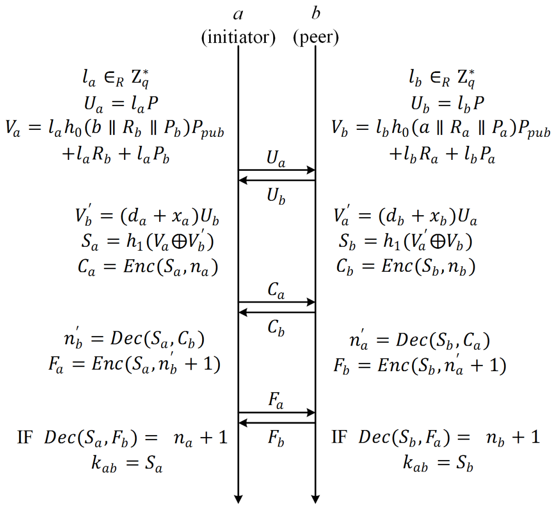


Fig. 5: Device Pairing (Step 2 in Fig.3).

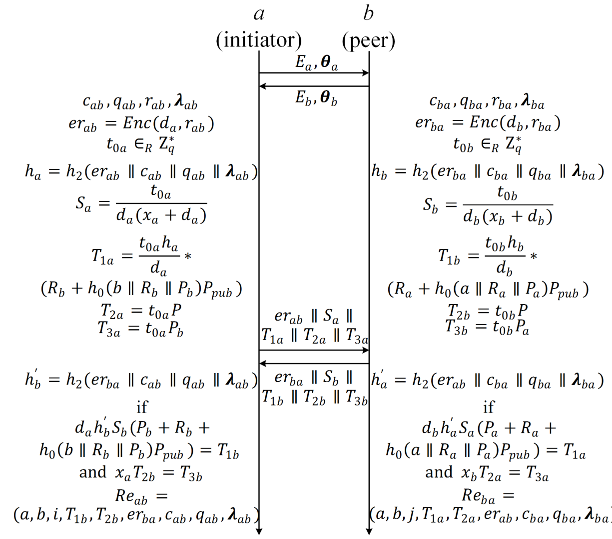


Fig. 6: D2D Receipt Generation (Step 3 in Fig.3).

is treated as a's credential. Meanwhile, the BS records $(a, r_a, d_a, P_a, \boldsymbol{t}_a)$.

5.3 Device Pairing

In this process, any pair of registered devices encountered use their credentials to negotiate a shared key for a secure D2D connection.

To accomplish a specific kind of D2D transaction, the initiator a selects a peer b among multiple available candidates considering their device trustvalues and other states (*e.g.* D2D channel signal strength). Then a and b start the device pairing. Specific operations (Step 2 in Fig.3) are demonstrated in Fig.5, where '$\oplus$' denotes the xor operation, and $Enc(x, y)$ and $Dec(x, y)$ denote the symmetric encryption and decryption with x as the key and y as the plaintext. The shared key k_{ab} is generated according to a and b's credentials. $l_{a(b)}$ and $n_{a(b)}$ are random numbers guaranteeing the freshness of k_{ab} and the Challenge-Response messages C and F, respectively.

5.4 Device Trustvalue Management

In this process, each end of a finished D2D transaction generates a verifiable receipt to profile the current transaction, which is uploaded to the BS for trustvalue adjustment.

D2D Transaction Receipt Generation: After the i^{th} D2D transaction of device a (the j^{th} of device b), a and b generate receipts that record both the transaction information and the profiles of each other's behavior.

Specific operations (Step 3.1 in Fig.3) are demonstrated in Fig.6. Note that all interactions between a and b are encrypted by k_{ab} (the symmetric cryptography operation (Enc, Dec) and the transaction indexes (i, j) are omitted for conciseness). Here, a and b exchange the contact histories $\mathcal{E}_a$ and $\mathcal{E}_b$ (together with $\boldsymbol{\theta}_a$ and $\boldsymbol{\theta}_b$) to calculate each other's credibility $c_{ab} = c_{ba}$. Then, considering the QoS of the D2D link $q_{ab} = q_{ba}$ and the knowledge of historical transactions $\bar{q}_{a(b)}$ and $\bar{c}_{a(b)}$, device $a(b)$ generates a rating $r_{ab(ba)}$ to evaluate $b(a)$'s behavior. For the integrity and accountability concerns, the rating is encrypted before exchanged that only the generator and the BS can decrypt. $T_{1,2,3}$ are signatures that are generated using the credentials of a, b, and the BS combined, while t_0 is used to guarantee the signature freshness. $Re_{ab(ba)}$ is the receipt of device $a(b)$.

Trustvalue Adjustment: Since D2D receipts will expire after a certain period of time, participating devices will upload locally verified receipts to the BS (Step 3.2.1 in Fig.3) when there is a cost-effective Internet access (*e.g.* free WiFi).

When device a uploads Re_{ab}, the BS can get the transaction information and behavior profiles including $a, b, c_{ab}, q_{ab}, \boldsymbol{\lambda}_{ab}$ and $r_{ba} = Dec(d_b, er_{ba})$. To verify the validity of Re_{ab}, the BS computes

$$h_b' = h_2(er_{ba} \| c_{ab} \| q_{ab} \| \boldsymbol{\lambda}_{ab}), \tag{12}$$

then, according to the partial key pairs of a and b, the BS verifies whether

$$\frac{d_a h_b'}{d_b} T_{2b} = T_{1b} \tag{13}$$

holds. If Eq.(13) holds, the receipt is valid.

With a verified receipt, according to Eq.(1) and (10), the trustvalue of device a can be adjusted.

For the integrity and authenticity concerns, the BS generates a signature for each adjusted device trustvalue. For device a discussed above, the BS generates

$$T_{4bs} = \frac{l_{bs} \boldsymbol{t}_a \boldsymbol{u}}{x + r_a}, \quad T_{5bs} = l_{bs} P, \tag{14}$$

where $l_{bs} \in_R Z_q^*$ is used to guarantee the signature freshness, $\boldsymbol{t}_a$ is the adjusted trustvalue of device a, and $\boldsymbol{u}$ is a $|\mathcal{N}|$-dimensional unit vector.

Then $\boldsymbol{t}_a \| T_{4bs} \| T_{5bs}$ is replied to device a (Step 3.2.2 in Fig.3). Any device that encounters a can verify whether

$$T_{4bs}(P_{pub} + R_a) = \boldsymbol{t}_a \boldsymbol{u} T_{5bs} \tag{15}$$

holds. If Eq.(15) holds, the trustvalue is valid.

Until now, TDP realizes the secure device pairing and the online trust management in D2D-enabled MCSs.

5.5 Security Analysis

In this subsection, we conduct theoretical analysis to demonstrate the security intensity of TDP confronting potential threats in D2D-enabled MCSs (Subsection 3.2). Note that we consider the Computational Diffie-Hellman Problem on Elliptic Curves (ECDHP) [41] as intractable for any adversary with finite resources, which is clarified as follows:

Let E be an elliptic curve on a finite field F_q, whose generator is P: given P, aP and bP, where $a, b \in Z_q^*$, the computation of abP is intractable.

5.5.1 Confronting CO Attacks

Adversaries launch CO attacks to compromise the device pairing process between registered devices. Without loss of generality, let an adversary $m \in \mathcal{M}$ ($\mathcal{M} \subset \mathcal{D}$) be a registered device, who has a verifiable credential (sk_m, pk_m), that tries to compromise the device pairing process between benign devices $a, b \in \mathcal{D}$. In Appendixes B.1 to B.3, we demonstrate TDP's immunity to all CO attacks in Subsection 3.2, which indicates its comparable security intensity to the off-the-shelf protocols [24], [25].

5.5.2 Confronting TO Attacks

Adversaries launch TO attacks to manipulate D2D traffics. Without loss of generality, let a set of adversaries $\mathcal{M} \subset \mathcal{D}$ be multiple registered devices with verifiable credentials. They try to, either independently or collusively, attract unfair D2D transaction requests than benign devices in $\mathcal{D}$ do. In Appendixes B.4 to B.6, we demonstrate TDP's resistance to all TO attacks in Subsection 3.2, which can effectively prevent traffic manipulations in D2D-enabled MCSs.

5.6 Case Studies

In this part, we conduct two case studies to demonstrate how to apply TDP to mobile collaborative computing tasks [7], [8] and mobile crowdsensing tasks [4], [9], [21].

Collaborative Computing Tasks: In a mobile crowd-computing task, the BS outsources computing requests to registered devices through the crowdsourcing application. Let device a accept the task and recruit peer devices for collaborative computing. By investigating the status of nearby devices, including the trustvalue in terms of collaborative computing, a determines its peer devices and then conducts device pairing with each one of them.

During the pairing process between a and one of the recruited devices b, both of them use registered credentials (sk, pk) to negotiate, generate and verify a shared key k_{ab}, which is used to secure D2D communications during the task. After the computing task, recruited devices reply ratings to demonstrate their satisfactory. To generate the receipt, a and b exchange $\mathcal{E}_a$ and $\mathcal{E}_b$ using the established secure connection. Then receipts from a and b, *i.e.* Re_{ab} and Re_{ba}, are generated using the credentials of a, b and the BS.

When there is a cost-effective access to the Internet (*e.g.* free WiFi at work, home or a cafe), a and b separately upload the receipts to the BS. After authenticating the receipts, the BS will adjust their trustvalues according to Eq.(1) and (10). For the collaborative computing, since both ends of the transaction equally contribute, both ends will have a trustvalue adjustment (*i.e.* $f(\boldsymbol{\lambda}_{ab}) = (1, 1)$). Then the BS generates signatures for their adjusted trustvalues. Finally, the adjusted trustvalues and signatures are replied to a and b for future D2D transactions.

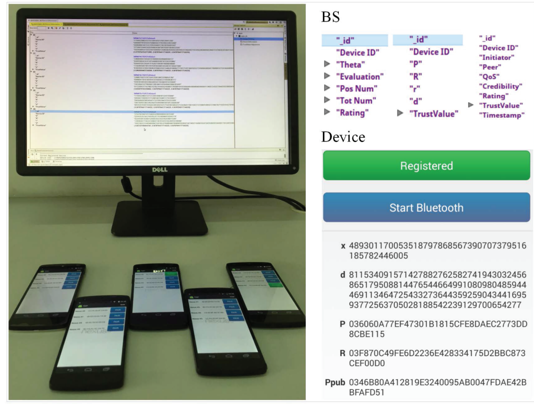


Fig. 7: D2D-Enabled MCS Prototype Illustration.

TABLE 2: Implementation Details of TDP.

Schemes	BTDP	WTDP
D2D Channel	Bluetooth v3.0 [25]	WiFi direct v1.5 [24]
ROM	4.20 MB	7.47 MB
RAM	19.29 MB	23.54 MB
Cryptography Primitives	ECC-secp160r1, AES-128 [42], [43]	ECC-secp160r1, AES-128 [42], [43]

Mobile Crowdsensing Tasks: In a mobile crowdsensing task, the BS outsources a sensing request to a participating device a to collect and report sensing data. Perhaps there are Internet-access or cost issues and a requires a peer device to deliver its data packets in a D2D manner. By investigating the status of nearby devices, including the trustvalue in terms of packet delivery, a selects a peer device b and conducts device pairing with it.

The device pairing and D2D receipt generation in such a mobile crowdsensing task are similar to that in the last case, therefore we make no further discussion for conciseness.

For the packet delivery, only the device that initiates the data transmission, which is device a in this case, will have an 'actual' trustvalue adjustment (*i.e.* $f(\boldsymbol{\lambda}_{ab}) = (1, 0)$). Therefore the peer device that receives packets will try further deliveries to get a trustvalue adjustment. Then the BS generates a signature for a's adjusted trustvalue. Finally, the adjusted trustvalue and signatures are replied to a and b (no change) for future D2D transactions.

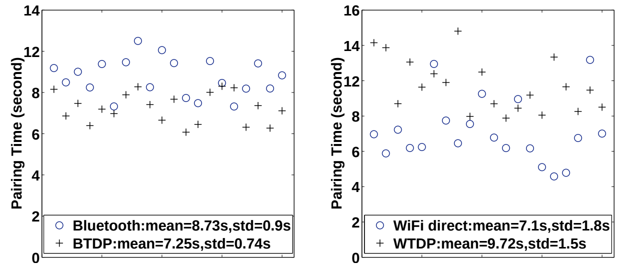
6 Evaluation

In this section, we evaluate the performance of TDP by both prototype experiments and trace-driven simulations.

6.1 Real-world Experiments

As shown in Fig.7, we constructed a MSC prototype consisting of:

- **A PC based BS**: A Dell OPTIPLEX 9010 with Linux Mint 17.3-64bit, Intel Core i5-3470 3.2GHz CPU, and 8GB RAM.
- **Five Android devices**: Google Nexus5 with Android 4.4.4, Qualcomm Snapdragon 800 2.3GHz CPU, 2GB RAM, and 16GB ROM.



(a) Bluetooth Pairing (b) WiFi direct Pairing

Fig. 8: Off-the-Shelf Pairing Time Comparison.

Devices can communicate with the BS through WiFi and communicate with each other through D2D channels.

As shown in Table 2, we implemented two versions of TDP as independent security services upon *user-transparent but insecure* D2D connections on Android mobile devices: BTDP is based on a Bluetooth connection in the `JustWork` mode [25], and WTDP is based on a WiFi direct connection in the `ServiceDiscovery` mode [24] whose mandatory manual authentication of WiFi Protected Setup (WPS) is blocked. Our implementation adopted the cryptography primitives in [42], [43].

In prototype experiments, we evaluated the performance of device pairing and trust management processes.

6.1.1 Device Pairing Study

In this set of experiments, we evaluated the performance of TDP device pairing process. Specifically, we considered *pairing time*, *i.e.* the time consumed from the device discovery to the establishment of a shared key, as the performance metric to measure the pairing speed. We compared BTDP and WTDP with original Bluetooth and WiFi direct with default settings [1]. As shown in Fig.8 (a) and (b), 20 pairing time samples were collected for BTDP (and original Bluetooth) and WTDP (and original WiFi direct) respectively.

Fig.8 (a) shows that both the mean and standard deviation (std) of the pairing time of BTDP are smaller than that of original Bluetooth pairing. This demonstrates that the pairing process of BTDP is not only faster, but also more stable, in comparison with Bluetooth. The autonomous key negotiation of BTDP avoids undetermined delays caused by manual confirmations in original Bluetooth pairing. It shortens the pairing time and limits any potential fluctuations other than that caused by Bluetooth device discovery itself.

From Fig.8 (b), we can see that WTDP outperforms WiFi Direct in terms of pairing stability (by avoiding potential manual operation delays), but its average pairing speed is slower. Different from the BTDP implemented on the *insecure but autonomous* Bluetooth connections (*i.e.* the `JustWork` mode of Bluetooth [25]), there exists no such WiFi direct connecting mode available for WTDP [2]. Therefore, we had to implement WTDP in the `ServiceDiscovery` mode [24] of WiFi direct for autonomous device discovery, and block the mandatory manual confirmation in the Android Wifip2p service. This inefficient implementation somehow emulated *insecure but autonomous* WiFi direct connections, but leading to significant performance degradation of our approach, *i.e.* each recorded pairing time of WTDP consists of both an entire round of original WiFi direct pairing and our upper-layer security service.

TABLE 3: User-Transparent Pairing Time.

Related Schemes	Miettinen *et al.* [15]	Truong *et al.* [16]	Lentz *et al.* [26]	TDP (B/W)
Secret Extraction/s	120	10	No need	No need
Key Negotiation/s	Not considered	Not considered	Not specified	7.25/9.72
Fluctuation/%	Not specified	Not specified	0-100	10.2/15.4

Table 3 compares the pairing time of TDP and state-of-the-art user-transparent device pairing approaches that are promising in implementing D2D-enabled MCSs. The proximity-based approaches in [15] and [16] collect contextual data for 120s and 10s respectively to generate key materials, and the time for key negotiation and authentication is not considered. The non-interactive approach in [26] pairs up two devices based on a series of asynchronous broadcasts of device public key materials with a 60s or 120s interval. it is possible that a pairing time fluctuation up to 100% will be introduced. Compared with them, TDP has no requirement on contextual information collection or device proximity, and its pairing fluctuation is well restricted without the synchronization concern.

6.1.2 Trust Management Study

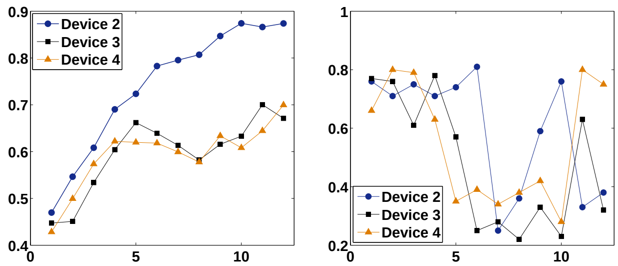
This set of experiments for trust management evaluation were based on BTDP pairing. Specifically, we adopted the average RSSI of the Bluetooth channel measured by both ends at the start of device pairing process as the metric for QoS estimation. The RSSI values were normalized, *i.e.* proportionally mapped from the range of [-100 dBm, 0dBm] to [0, 1]. According to Eq.(8), the rating that reflects the device behavior during a D2D transaction can be either QoS-driven or Credibility-Driven by adjusting the value of β. Therefore, we conducted two sets of experiments that separately focused on QoS ($\beta = 0.8$) and Credibility ($\beta = 0.2$), meanwhile we monitored the variation of device trustvalue and the direction of D2D traffic within the system.

Each set of experiments had 40 D2D transactions. At an experiment round, registered devices (Devices 1-5) took turns to post a transaction request and autonomously paired with the most trustworthy device that was visible. One of the devices (in turn) was excluded from candidates for a single round. During the experiments, only the trustvalue of transactions that lead to trust variations at the peer side was considered. We set the average transaction count $\bar{\sigma} = 30$. As a result, the trust accumulation sensitivity $c_g = 0.5$ and the damping sensitivity $c_w = 0.5$ according to Eq.(A1) and (A2) respectively. The credibility sensitivity $\alpha = 0.5$.

Fig.9 shows the results of the QoS-driven experiments. Here, each device was carried by a student roaming around the laboratory to introduce QoS variations. The majority of D2D requests were attracted by Devices 2-4. For Devices 3 and 4, a higher QoS boosts their trustvalue and vice

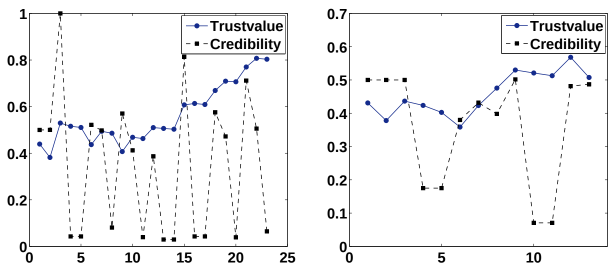
1. Note that we did our best to shorten the reactive time whenever there was a physical interaction (*i.e.* manual confirmations for pairing requests: PIN comparison, pairing button click).

2. The default WiFi direct pairing is in the `ActiveScan` mode [24], which requires both users to scan simultaneously for device discovery.



(a) Trustvalue (b) QoS

Fig. 9: QoS-Driven Trust Management.



(a) Device 2 Profiles (b) Device 3 Profiles

Fig. 10: Credibility-Driven Trust Management.

versa. Moreover, since Device 2 has a trustvalue boost at the beginning due to our device encountering setting, it stays in a dominant position all the time compared with others. Even it provides lower QoS for a while, its trustvalue has no severely but only a slightly drop since almost all of contact histories of others are with Device 2. To avoid such kind of monopoly, β should be well adjusted to offer devices with inherently lower QoS a chance for a trustvalue boost with higher credibilities.

Fig.10 shows the results of Credibility-driven experiments. Here, Devices 4 and 5 were set to reply negative ratings in all D2D transactions to introduce credibility differences. The majority of D2D requests were attracted by Devices 2 and 3. Devices 4 and 5 only attracted two requests because of low credibilities. For Devices 2 and 3, transactions with a credibility around 0.1 were conducted with Device 4 or 5. The influence of ratings with abnormal credibilities on the trustvalue of Devices 2 and 3 is nearly negligible. When they conduct transactions with other devices, a higher credibility boosts their trustvalue and vice versa. Actually, to encourage users refining their QoS, β should be well adjusted to offer devices with lower credibilities a chance for a trustvalue boost with higher QoS.

6.2 Trace-Driven Simulations

To study the practical performance of TDP in larger-scale MCSs, we constructed extensive trace-driven simulations based on OMNeT++ 4.6 [44], by using real human mobility data collected from the IEEE Infocom'05 conference [45] and the ACM Sigcomm'09 conference [46]. The settings of these two sets of simulations are as follows:

- **The Infocom'05 Trace** was used for constructing a D2D-enabled MCS with 41 registered devices. Here, the connecting time of D2D channels between encountered devices was used to estimate the D2D QoS. According to the simulation result with no attacker, we set the average transaction count $\bar{\sigma} = 110$. Therefore sensitivities $c_g = 0.14$ and $c_w = 0.5$ according to Eq.(A1) and (A2) respectively. For Eq.(3), we set $\alpha = 0.5$ to equally treat the similarity and diversity, and $\beta = 0.1$ for Eq.(8) to compensate the inherent unfairness on the device QoS. According to duration of the Infocom'05 trace, each round of simulations lasted for 30000 simulation seconds, which was treated as a single trustvalue management cycle.
- **The Sigcomm'09 Trace** has nearly twice node records than the Infocom'05 trace. By using this trace, a D2D-enabled MCS with 76 devices was constructed to evaluate the scalability of TDP. Here, the volume of data transmitted through D2D channels was used to estimate the D2D QoS. In this set of simulations, we set $\bar{\sigma} = 31$, $c_g = 0.5$, $c_w = 0.5$, $\alpha = 0.5$ and $\beta = 0.1$. Each round of simulations lasted for 15000 simulation seconds.

In all simulations, devices can spontaneously register to the BS and publish or accept D2D requests. The trustvalue was treated as the unique metric for peer selections. Meanwhile, since the type of D2D transactions has no impact on the performance of TDP, we only considered the trustvalue of transactions that lead to trust variations at the peer side.

6.2.1 *Performance Metrics*

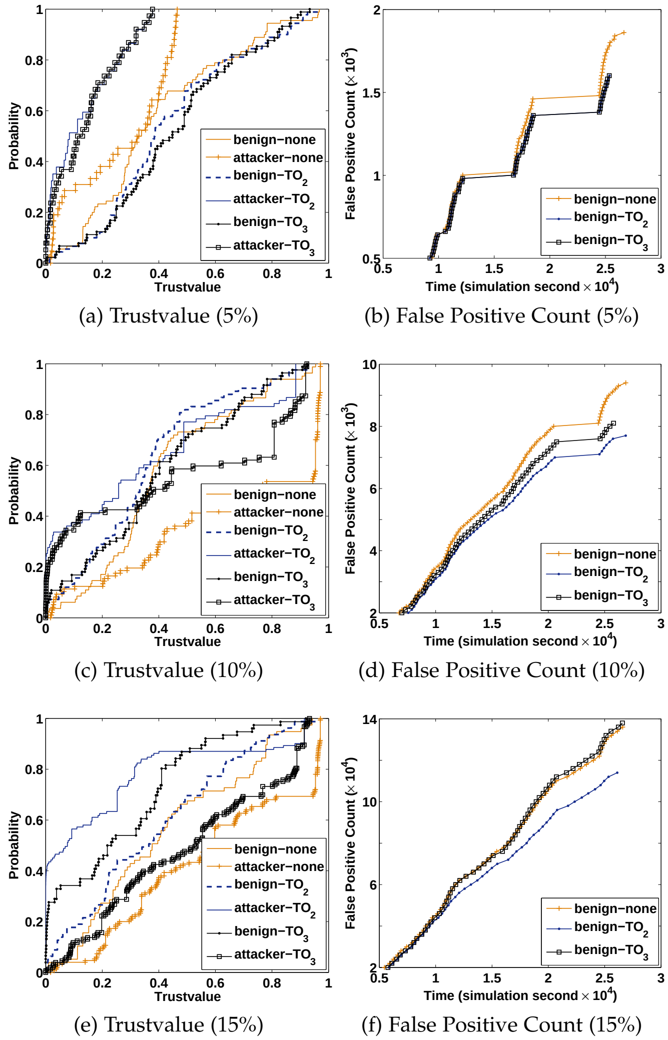
Since the goal of TO attackers in D2D-enabled MCSs is to get unfair advantages in peer selections over benign devices, we collected the following two metrics to evaluate the effectiveness of TDP against potential traffic manipulations:

- **Device Trustvalue**, the time-varying device trustvalue, whose cumulative distribution indicates whether malicious behaviors of TO attackers downgrade the trustworthiness of benign devices. Ideally, TDP should be able to automatically downgrade the trustvalue of TO attackers without disturbing (downgrading) the trustvalue of benign devices;
- **False Positive Transactions Numbers**, the number of transactions attracted by TO attackers from benign devices, which quantifies the impact of traffic manipulations of TO attackers. Ideally, TDP should be able to automatically limit the number of transactions attracted by TO attackers.

In our simulations, participating devices in the MCS were divided into two categories: attackers and benign devices, whose trustvalue and false positive transactions were recorded separately for analysis. For comparison, we ran two simulations with no attacker for both the Infocom and Sigcomm traces (*i.e.* the original scenario). For the rest of the paper, we use the Cumulative Distribution Figure (CDF) to present the trustvalue of both benign devices and TO attackers, and we use the Time Variance Figure (TVF) to present the number of false positive transactions attracted by TO attackers.

6.2.2 *Impact of Attacker Percentage*

In this set of simulations, different percentages (5%, 10%, and 15%) of registered devices were randomly selected as

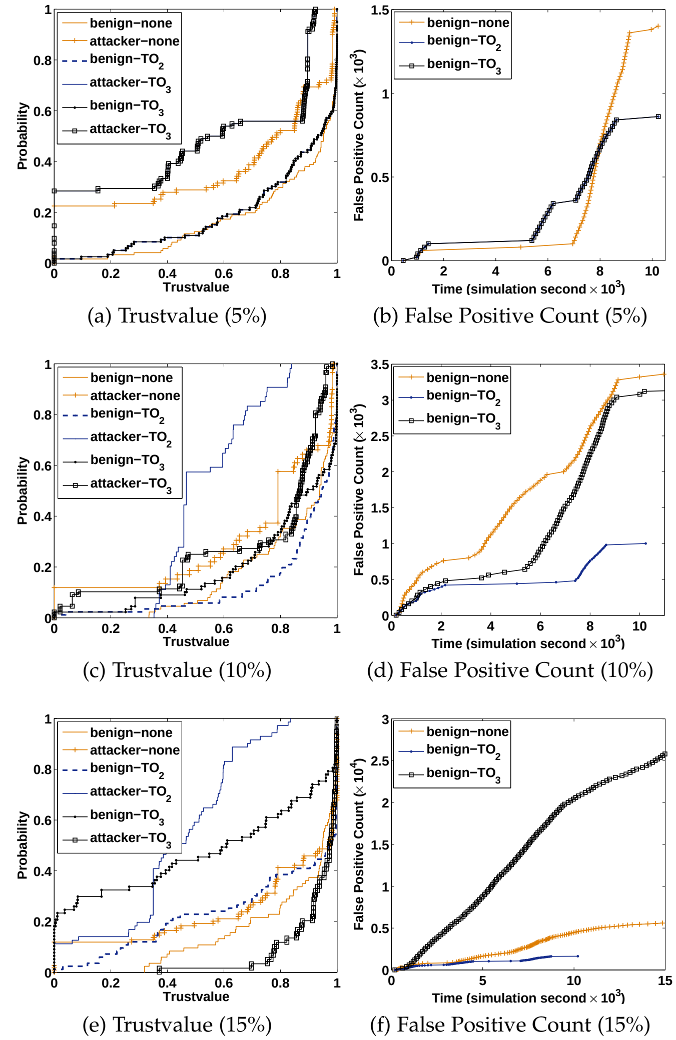


(a) Trustvalue (5%) (b) False Positive Count (5%)

(c) Trustvalue (10%) (d) False Positive Count (10%)

(e) Trustvalue (15%) (f) False Positive Count (15%)

Fig. 11: Impact of Different Attacker Percentages (Infocom).



(a) Trustvalue (5%) (b) False Positive Count (5%)

(c) Trustvalue (10%) (d) False Positive Count (10%)

(e) Trustvalue (15%) (f) False Positive Count (15%)

Fig. 12: Impact of Different Attacker Percentages (Sigcomm).

attackers that were continuously launching TO attacks. The results of the Infocom and Sigcomm simulations are shown in Fig.11 and Fig.12 respectively.

For both sets of simulations, the impact of TO_2 attacks is well eliminated in all three scenarios: the average trustvalue of benign devices is not obviously affected compared with the original scenario (*e.g.* benign-TO_2 ($0.46/9.5\%^{\uparrow}$) vs benign-none (0.42) in Fig.11 (a)). However, the average trustvalue of attackers is severely downgraded because of their TO_2 attacks (*e.g.* attacker-TO_2 ($0.13/50\%^{\downarrow}$) vs attacker-none (0.26) in Fig.11 (a)). Meanwhile, the false positive transaction number also decreases when there are TO_2 attacks (*e.g.* benign-TO_2 ($1612/13.5\%^{\downarrow}$) vs benign-none (1864) in Fig.11 (b)).

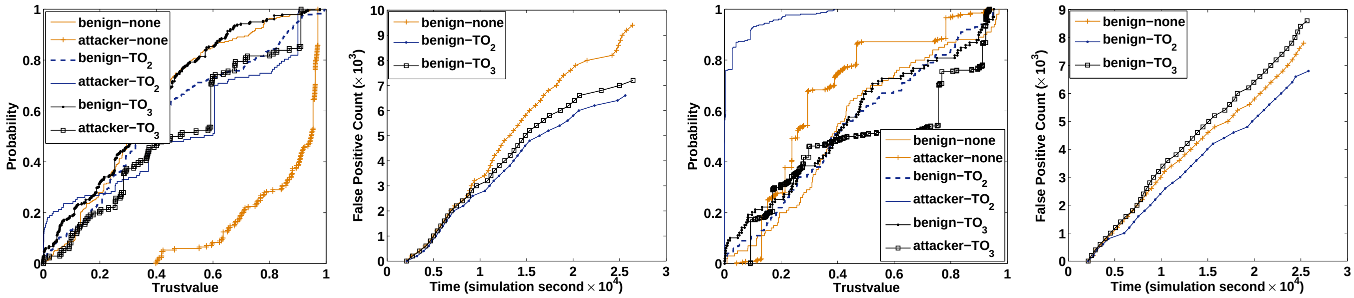
For the Infocom simulations, the impact of TO_3 attacks is well eliminated in all three scenarios. For the Sigcomm simulations, when the attacker percentage is 15%, the impact of TO_3 attacks cannot be neglected. However, in general, it is well accepted that devices in the MCS should form a relatively good community with small percentage of malicious devices (*e.g.* 4% in [47] and 10% in [48]). For the scenario with the attacker percentage up to 10%, TDP manages to effectively eliminate the impact of TO_3 attacks.

6.2.3 Impact of Attackers with Different Properties

In this set of simulations, considering that TO attacks from more trustworthy or popular devices may cause deeper impacts, we specifically selected a fixed percentage of devices with different properties as continuous TO attackers, *i.e.* top 7% devices in average trustvalue (TopTV) or top 7% devices in transaction count (TopTC) in the original scenario. The results of the Infocom and Sigcomm simulations are shown in Fig.13 and Fig.14 respectively.

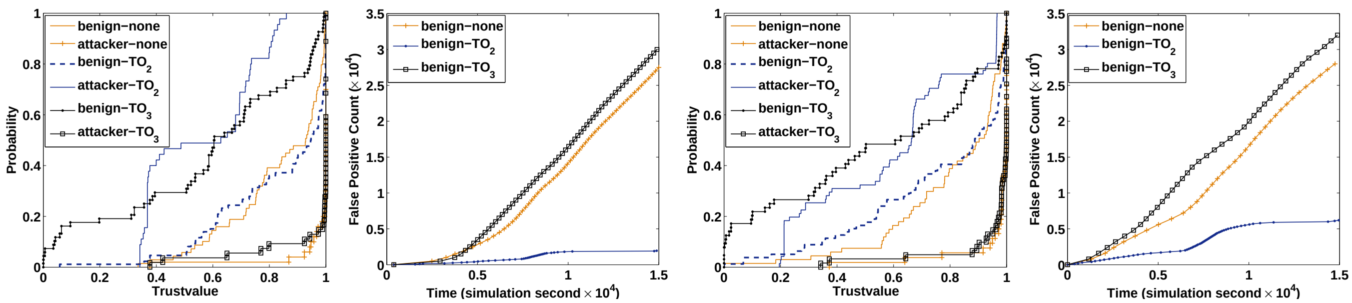
For both sets of simulations, the impact of TO_2 attacks is well eliminated in all two scenarios: there is no obvious impact on the average trustvalue of benign devices (*e.g.* benign-TO_2 ($0.41/9.8\%^{\uparrow}$) vs benign-none (0.37) in Fig.13 (a)), while the average trustvalue of attackers is severely downgraded (*e.g.* attacker-TO_2 ($0.44/48.2\%^{\downarrow}$) vs attacker-none (0.85) in Fig.13 (a)), and the number of false positive transactions also decreases correspondingly (*e.g.* benign-TO_2 ($6737/29.7\%^{\downarrow}$) vs benign-none (9579) in Fig.13 (b)).

For the Infocom simulations, the impact of TO_3 attacks from TopTV devices is well eliminated according to Fig.13 (a) and (b). In the TopTC scenario, attackers have a boost in terms of the average trustvalue (*i.e.* attacker-TO_3 ($0.53/65.6\%^{\uparrow}$) vs attacker-none (0.32) in Fig.13 (c))



(a) Trustvalue (TopTV) (b) False Positive Count (TopTV) (c) Trustvalue (TopTC) (d) False Positive Count (TopTC)

Fig. 13: Impact of Attackers with Different Properties (Infocom).



(a) Trustvalue (TopTV) (b) False Positive Count (TopTV) (c) Trustvalue (TopTC) (d) False Positive Count (TopTC)

Fig. 14: Impact of Attackers with Different Properties (Sigcomm).

and the number of attracted transactions (*i.e.* attacker-TO_3 ($8758/9.9\%^{\uparrow}$) vs attacker-none (7969) in Fig.13 (d)). However, the impact on the average trustvalue of benign devices is negligible (*i.e.* benign-TO_3 ($0.40/7.0\%^{\downarrow}$) vs benign-none (0.43) in Fig.13 (c)). For the Sigcomm simulations with TO_3 attacks, the average trustvalue of attackers remains similar, and the number of attracted transactions has a slightly boost in all two scenarios. In fact, the impact of TO_3 attacks on benign devices is difficult to be restricted only when the collusive attackers possess predominant popularities (*i.e.* $\frac{6\%}{1.17\%} = 512.8\%$ of the average popularity of benign devices for the Infocom TopTC scenario, $\frac{4.42\%}{1.05\%} = 421\%$ and $\frac{5.66\%}{1.02\%} = 555\%$ for the Sigcomm TopTV and TopTC scenarios respectively). However, considering it is irrational for the most popular devices to collude in reality because of the low cost-efficiency, TDP manages to effectively eliminate the impact of TO_3 attacks from devices with rationally advanced trustworthiness or popularity.

6.2.4 *Impact of Attacking Intensity*

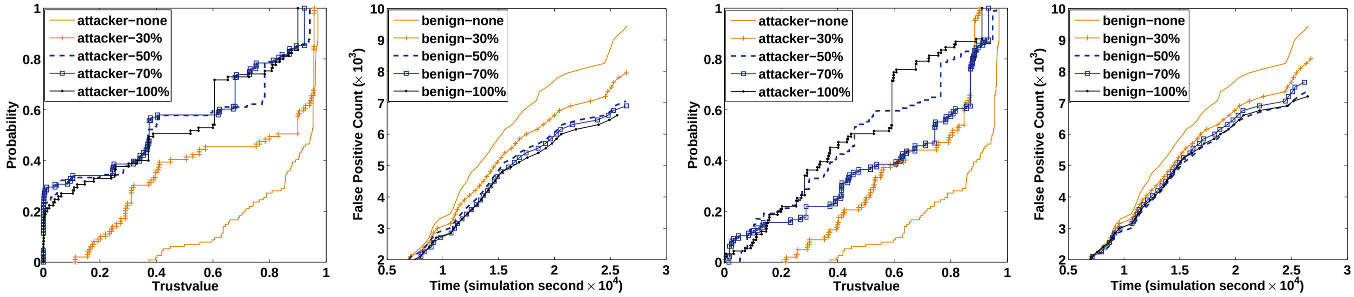
In this set of simulations, to study the impact of TO attacks with different intensities, we randomly selected a fixed percentage (7%) of devices as attackers that launch TO attacks in each D2D transaction with different probabilities (*i.e.* 0, 30%, 50%, 70%, and 100%). The results of the Infocom and Sigcomm simulations are shown in Fig.15 and Fig.16 respectively.

For the Infocom simulations, the impacts of both TO_2 and TO_3 attacks are well eliminated: compared with the original scenario, the average trustvalue of attackers is in inverse proportion to the attack intensity (*e.g.* attacker-30% ($0.66/22.4\%^{\downarrow}$), attacker-50% ($0.49/42.4\%^{\downarrow}$) vs attacker-none (0.85) in Fig.15 (a)). The number of false positive transactions also decreases in direct proportion to the attack intensity (*e.g.* benign-30% ($8047/16.0\%^{\downarrow}$), benign-50% ($7156/25.3\%^{\downarrow}$) vs attacker-none (9579) in Fig.15 (b)).

For the Sigcomm simulations, the results are basically the same except for the 30% scenarios. For both 30% TO_2 and TO_3 attacks, attackers have a slightly boost in terms of the average trustvalue and the number of attracted transactions. In fact, since the device contact in the Sigcomm trace is relatively sparse compared to the Infocom trace (*i.e.* 31 vs 110 in terms of the average transaction count of participating devices), system's reaction to TO attacks is correspondingly slower. However, the impacts of the 30% TO attacks on the trustvalue of benign devices are negligible (*e.g.* benign-30% ($0.85/1.2\%^{\uparrow}$) vs benign-none (0.86) in Fig.16 (a)). Therefore, TDP manages to effectively eliminate the impact of TO attacks with different intensities.

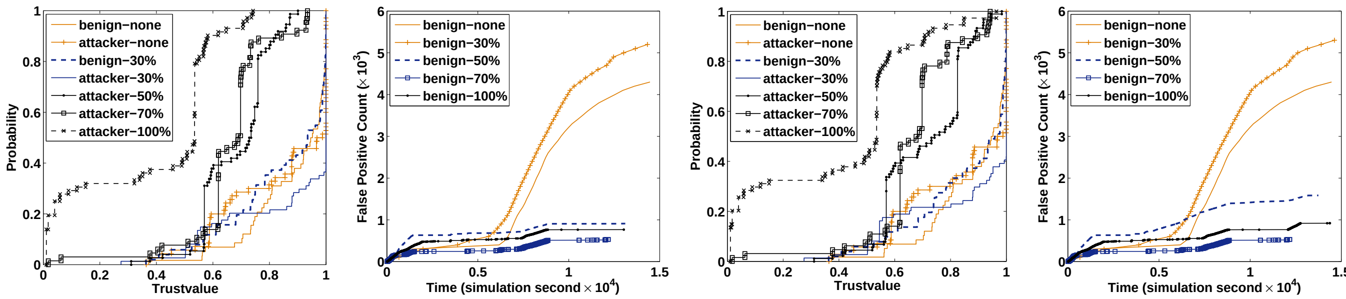
7 Conclusion

In this paper, we develop TDP, a rapid, user-transparent, and trustworthy device pairing scheme for Mobile Crowdsouring Systems (MCS) with opportunistic Device-to-Device (D2D) communications. We first develop a formal method to estimate the trustworthiness of participating devices at realtime. Then, we propose a CL-PKC based credential framework that allows each encountered device to spontaneously establish a secure D2D connection with the most trustworthy peer nearby. We theoretically prove that TDP manages to achieve a comparable security intensity with the off-the-shelf protocols and the immunity to the TO attacks that are not considered by existing approaches. Through real-world experiments using Android devices, we show that TDP outperforms existing approaches in terms of pairing speed



(a) Trustvalue (TO_2) (b) False Positive Count (TO_2) (c) Trustvalue (TO_3) (d) False Positive Count (TO_3)

Fig. 15: Impact of Attacks with Different Intensities (Infocom).



(a) Trustvalue (TO_2) (b) False Positive Count (TO_2) (c) Trustvalue (TO_3) (d) False Positive Count (TO_3)

Fig. 16: Impact of Attacks with Different Intensities (Sigcomm).

and stability. In addition, extensive trace-driven simulation results verify that TDP manages to effectively prevent the traffic manipulation of TO attackers in large-scale MCSs. Interesting future work is to apply TDP to specific MCS applications such as D2D-enabled mobile social networking and mobile cloud computing.

Acknowledgments

This work is sponsored by Intel Corporation, China 1000 Young Talents Program, Young Talent Support Plan of Xi'an Jiaotong University, and NSF China (NSFC) under Grant 61572398. We thank the anonymous reviewers for their helpful and insightful comments and suggestions, which significantly contribute to improving the quality of the paper.

References

[1] J. Ren, Y. Zhang, K. Zhang, and X. Shen, "Exploiting mobile crowdsourcing for pervasive cloud services: challenges and solutions," *IEEE Commun. Mag.*, vol. 53, no. 3, pp. 98–105, 2015.

[2] D. Zhang, H. Xiong, L. Wang, and G. Chen, "Crowdrecruiter: Selecting participants for piggyback crowdsensing under probabilistic coverage constraint," in *Proc. ACM Ubicomp*, 2014, pp. 703–714.

[3] R. Gao, M. Zhao, T. Ye, F. Ye, Y. Wang, K. Bian, T. Wang, and X. Li, "Jigsaw: Indoor floor plan reconstruction via mobile crowdsensing," in *Proc. ACM Mobicom*, 2014, pp. 249–260.

[4] S. Yang, U. Adeel, and J. McCann, "Backpressure meets taxes: Faithful data collection in stochastic mobile phone sensing systems," in *Proc. IEEE Infocom*, 2015, pp. 1490–1498.

[5] S. Nawaz, C. Efstratiou, and C. Mascolo, "Parksense: A smartphone based sensing system for on-street parking," in *Proc. ACM Mobicom*, 2013, pp. 75–86.

[6] H. Shin, T. Park, S. Kang, B. Lee, J. Song, Y. Chon, and H. Cha, "Cosmic: designing a mobile crowd-sourced collaborative application to find a missing child in situ," in *Proc. ACM Mobiheld*, 2014, pp. 389–398.

[7] Y. Li, M. Qian, D. Jin, P. Hui, Z. Wang, and S. Chen, "Multiple mobile data offloading through disruption tolerant networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 7, pp. 1579–1596, 2014.

[8] D. G. Murray, E. Yoneki, J. Crowcroft, and S. Hand, "The case for crowd computing," in *Proc. ACM Mobiheld*, 2010, pp. 39–44.

[9] M. Karaliopoulos, O. Telelis, and I. Koutsopoulos, "User recruitment for mobile crowdsensing over opportunistic networks," in *Proc. IEEE Infocom*, 2015, pp. 2254–2262.

[10] S. Yang, U. Adeel, and J. A. Mccann, "Selfish mules: Social profit maximization in sparse sensornets using rationally-selfish human relays," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 6, pp. 1124–1134, 2013.

[11] Y. Han, T. Luo, D. Li, and H. Wu, "Competition-based participant recruitment for delay-sensitive crowdsourcing applications in d2d networks," *IEEE Trans. Mobile Comput.*, pp. 1–1, 2016.

[12] U. Adeel, S. Yang, and J. A. McCann, "Self-optimizing citizen-centric mobile urban sensing systems," in *Proc. ICAC*, 2014, pp. 161–167.

[13] M. Bakht, M. Trower, and R. H. Kravets, "Searchlight: won't you be my neighbor?" in *Proc. ACM Mobicom*, 2012, pp. 185–196.

[14] A. Varshavsky, A. Scannell, A. LaMarca, and E. De Lara, "Amigo: Proximity-based authentication of mobile devices," in *Proc. Springer-Verlag Ubicomp*, 2007, pp. 253–270.

[15] M. Miettinen, N. Asokan, T. D. Nguyen, A.-R. Sadeghi, and M. Sobhani, "Context-based zero-interaction pairing and key evolution for advanced personal devices," in *Proc. ACM CCS*, 2014, pp. 880–891.

[16] H. T. T. Truong, X. Gao, B. Shrestha, N. Saxena, N. Asokan, and P. Nurmi, "Comparing and fusing different sensor modalities for relay attack resistance in zero-interaction authentication," in *Proc. IEEE Percom*, 2014, pp. 163–171.

[17] J. Liu, S. Zhang, H. Nishiyama, N. Kato, and J. Guo, "A stochastic geometry analysis of d2d overlaying multi-channel downlink cellular networks," in *Proc. IEEE Infocom*, 2015, pp. 46–54.

[18] J. Li, R. Bhattacharyya, S. Paul, S. Shakkottai, and V. Subramanian, "Incentivizing sharing in realtime d2d streaming networks: A mean field game perspective," in *Proc. IEEE Infocom*, 2015, pp. 2119–2127.

[19] S. Yang, X. Yang, and H. Yang, "A cross-layer framework for position-based routing and medium access control in heterogeneous mobile ad hoc networks," *Telecommunication Systems*, vol. 42, no. 1-2, pp. 29–46, 2009.

[20] X. Yang, X. Fan, W. Yu, X. Fu, and S. Yang, "Hlls: a history information based light location service for manets," *Computer Networks*, vol. 56, no. 2, pp. 731–744, 2012.

[21] M. Xiao, J. Wu, L. Huang, Y. Wang, and C. Liu, "Multi-task assignment for crowdsensing in mobile social networks," in *Proc. IEEE Infocom*, 2015, pp. 2227–2235.

[22] M. Li, S. Yu, J. D. Guttman, W. Lou, and K. Ren, "Secure ad hoc trust initialization and key management in wireless body area networks," *ACM Trans. Sensor Networks*, vol. 9, no. 2, pp. 53–55, 2013.

[23] P. Aditya, V. Erdélyi, M. Lentz, E. Shi, B. Bhattacharjee, and P. Druschel, "Encore: Private, context-based communication for mobile social apps," in *Proc. ACM Mobisys*, 2014, pp. 135–148.

[24] "Wi-fi peer-to-peer (p2p) technical specification v1.5," 2014.

[25] https://www.bluetooth.com/specifications.

[26] M. Lentz, V. Erdélyi, P. Aditya, E. Shi, P. Druschel, and B. Bhattacharjee, "Sddr: Light-weight, secure mobile encounters," in *Proc. USENIX Security*, 2014, pp. 925–940.

[27] H. Mousa, S. B. Mokhtar, O. Hasan, O. Younes, M. Hadhoud, and L. Brunie, "Trust management and reputation systems in mobile participatory sensing applications: A survey," *Computer Networks*, 2015.

[28] C. Selvaraj and S. Anand, "A survey on security issues of reputation management systems for peer-to-peer networks," *Computer Science Review*, vol. 6, no. 4, pp. 145–160, 2012.

[29] M. Blaze, J. Feigenbaum, and J. Lacy, "Decentralized trust management system," *Procs.of IEEE Conf.security & Privacy*, vol. 30, no. 1, pp. 164–173, 1996.

[30] E. Damiani, "A reputation-based approach for choosing reliable resources in peer-to-peer networks," in *Proc. ACM CCS*, 2002, pp. 207–216.

[31] X. Chu, X. Chen, K. Zhao, and J. Liu, "Reputation and trust management in heterogeneous peer-to-peer networks," *Springer Telecommun Syst*, vol. 44, no. 3-4, pp. 191–203, 2010.

[32] B. Qureshi, G. Min, and D. Kouvatsos, "A distributed reputation and trust management scheme for mobile peer-to-peer networks," *Elsevier Computer Communications*, vol. 35, no. 5, pp. 608–618, 2012.

[33] K. L. Huang, S. S. Kanhere, and W. Hu, "Are you contributing trustworthy data?: the case for a reputation system in participatory sensing," in *Proc. ACM MSWiM*, 2010, pp. 14–22.

[34] X. Wang, W. Cheng, P. Mohapatra, and T. Abdelzaher, "Enabling reputation and trust in privacy-preserving mobile sensing," *IEEE Trans. Mobile Comput.*, vol. 13, no. 12, p. 1, 2014.

[35] M. Talasila, R. Curtmola, and C. Borcea, "Collaborative bluetooth-based location authentication on smart phones," *Elsevier Pervasive & Mobile Computing*, vol. 17, no. PA, pp. 43–62, 2015.

[36] C. Zhao, F. Shi, R. Huang, X. Yang, and J. McCann, "Trustworthy device pairing for opportunistic device-to-device communications in mobile crowdsourcing systems," in *Proc. ACM S3*, 2015, pp. 4–6.

[37] J. F. Kenney and E. S. Keeping, "Mathematics of statistics-part one," 1954.

[38] A. Josang and R. Ismail, "The beta reputation system," in *Proc. Bled Conference on Electronic Commerce*, 2002, pp. 1–14.

[39] R. H. Jones, "Probability estimation using a multinomial logistic function," *Journal of Statistical Computation & Simulation*, vol. 3, no. 4, pp. 315–329, 1975.

[40] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," *Lecture Notes in Computer Science*, vol. 2894, no. 2, pp. 452–473, 2003.

[41] A. Joux and K. Nguyen, "Separating decision diffie–hellman from computational diffie–hellman in cryptographic groups," *Springer J CRYPTOL*, vol. 16, no. 4, pp. 239–247, 2003.

[42] http://www.bouncycastle.org/java.html.

[43] *SEC 2: Recommended Elliptic Curve Domain Parameters*, 2000.

[44] https://omnetpp.org.

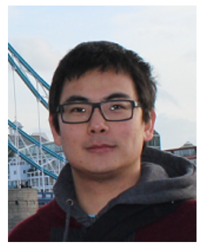
[45] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "CRAWDAD dataset cambridge/haggle (v. 2009-05-29)," Downloaded from http://crawdad.org/cambridge/haggle/20090529, May 2009.

[46] A.-K. Pietilainen and C. Diot, "CRAWDAD dataset thlab/ sigcomm2009 (v. 2012-07-15)," Downloaded from http://crawdad.org/ thlab/sigcomm2009/ 20120715, Jul. 2012.

[47] H. Shen, Y. Lin, K. Sapra, and Z. Li, "Enhancing collusion resilience in reputation systems," *IEEE Trans. Parallel Distrib. Syst.*, pp. 1–1, 2015.

[48] X. Li, F. Zhou, and X. Yang, "Scalable feedback aggregating (sfa) overlay for large-scale p2p trust management," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 10, pp. 1944–1957, 2012.

Cong Zhao received his BS degree from Xi'an Jiaotong University (XJTU) in 2012. He is currently a PhD candidate at the Department of Computer Science and Technology of XJTU. From 2014-2015, Cong has been an international exchange PhD student in the Department of Computing at Imperial College London. His research interests include network security, computing economics, and people-centric sensing.



Shusen Yang received his PhD in Computing from Imperial College London in 2014. He is currently a professor in the Institute of Information and System Science at Xi'an Jiaotong University (XJTU). He also holds an academic association as an honorary research fellow at Imperial College London. Before joining XJTU, Shusen worked as a Lecturer (Assistant Professor) at University of Liverpool from 2015 to 2016, and a Research Associate at Intel Collaborative Research Institute (ICRI) for sustainable connected cities from 2013 to 2014. His research interests include mobile and wireless networks, networks with human in the loop, and data-driven networked systems. Shusen is a recipient of China "1000 Young Talents Program". He is a member of IEEE and ACM.

Xinyu Yang received the B.S., M.S., and Ph.D. degrees from Xi'an Jiaotong University in 1995, 1997, and 2001, respectively. Currently, he is a professor in the Department of Computer Science and Technology, Xi'an Jiaotong University. His main research interests lie in the areas of wireless networks and network security. He was awarded the title of "New Century Excellent Talent" by the Chinese Ministry of Education in 2009. He is a member of ACM and IEEE.

Julie A. McCann is a Professor in Computer Systems at Imperial College. Her research centers on highly decentralized and self-organizing scalable algorithms for spatial computing systems e.g. wireless sensing networks. She leads both the Adaptive Embedded Systems Engineering Research Group and the Intel Collaborative Research Institute for Sustainable Cities, and is currently working with NEC and others on substantive smart city projects. She has received significant funding though bodies such as the UK's EPSRC, TSB and NERC as well as various international funds, and is an elected peer for the EPSRC. She has actively served on, and chaired, many conference committees and is currently Associative Editor for the ACM Transactions on Autonomous and Adaptive Systems. She is a Fellow of the BCS.