Flexible data access control in D2D communications

Published in:
Future Generation Computer Systems

DOI:
10.1016/j.future.2017.08.052

Published: 01/01/2018

Please cite the original version:
Flexible data access control in D2D communications

Zheng Yan \(^{a,b,*}\), Haomeng Xie \(^a\), Peng Zhang \(^a\), Brij B. Gupta \(^c\)

\(^{a}\) State Key Laboratory on Integrated Services Networks, School of Cyber Engineering, Xidian University, China
\(^{b}\) Department of Communications and Networking, Aalto University, Finland
\(^{c}\) National Institute of Technology Kurukshetra, India

**Highlights**

- A scheme to secure D2D communications on the basis of multiple trust levels.
- Flexible data access control for all D2D communication scenarios.
- A robust trust evaluation method for D2D communications.
- Performance evaluation through security proof, analysis, and simulations.

**Abstract**

Device-to-Device (D2D) communications have been regarded as an advanced technology for the next generation mobile communication networks and wireless systems (5G). It is essential to secure D2D communication data for resisting malicious attacks. However, secure D2D communications among mobile devices have not been well solved. By paying attention to the important role of trust in securing D2D communications, in this paper, we propose a scheme using either a General Trust (GT) level issued by a core network or a Local Trust (LT) level evaluated by a device or both to control D2D communication data by applying Attribute-Based Encryption (ABE). This scheme realizes secure data communications among mobile devices under the legacy system model of Long-Term Evolution (LTE). Performance analysis and evaluation demonstrate that the proposed scheme is effective with regard to security, computation complexity, communication cost, flexibility and scalability.

© 2017 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

**1. Introduction**

Nowadays, the development of wireless and mobile communications has promoted the emergence of new technologies. They aim to significantly improve network performance in such aspects as overall throughput, spectrum utilization and energy consumption. At the same time, business services such as content-based networking have also made great progress, which encourages us to invent new technologies to meet the demands of users. Device-to-Device (D2D) communications have been considered as one of the most important technologies for short-range communications and will play a key role in the next generation mobile communication networks and wireless systems (5G).

Device-to-device (D2D) communications refer to a communication technology that can make direct communications among devices without direct involvement of fixed network infrastructures, like Access Points (APs), Base Stations (BSs), WiFi-Direct [1] and Bluetooth, and may operate in a purely autonomous way. D2D communications can be implemented as a constrained or controlled underlaying network of LTE-Advanced networks in the same cellular spectrum [2]. At the same time, in order to meet the demands of market such as context awareness and proximity services, mobile operators have developed new business models and new application scenarios based on D2D communications, such as pervasive social networking and emergency services.

As a promising technology, D2D communications have attracted a lot of attentions in academia, industry and recent standardization organizations. In academia, D2D communications are considered as an underlay of LTE-Advanced networks [2] from the beginning. Many researchers focus on studying D2D application scenarios, resource allocation [3], communication mode selection [4,5], interference control and power control [6]. In industry, the development of D2D communications is active. For example, a D2D communication subsystem called FlashLinQ [7,8] has been developed to make short-range communications possible. It is expected to not only complement traditional services of...
cellular-based networks, but also provide a new scalable platform of new applications such as advertising, secure mobile payment and content sharing. Meanwhile, the standardization of the third Generation Partnership Program (3GPP) on D2D communications is underway. The problems related to D2D communications will be further normalized.

Despite the significant benefits of D2D communications, this new networking scenario poses unique security threats to D2D services. Comparing with the conventional communications between devices and BSs, direct communications among proximate devices are more susceptible due to the following reasons: (1) limited computing capacity for security-related protection in mobile devices, (2) semi-autonomous or fully-autonomous security management, and (3) relay transmission architecture. These factors may seriously hinder the rapid deployment of D2D communications in practice. However, both academia and industrial standardization organizations have not carefully researched the security of D2D communications. Secure data communications among a number of mobile devices is one of the most important security issues in D2D communications. But based on our survey [9], this issue has not yet been well solved in a flexible way to support all D2D communication scenarios no matter a core network is available or unavailable. A new security scheme is expected in practice.

In this paper, we utilize two-dimensional trust levels: a General Trust (GT) level evaluated by the core network or a Local Trust (LT) level assessed by a device itself or both to control D2D communication data access based on Attribute-Based Encryption (ABE) [10–12]. Each user device would select at least one kind trust level of GT or LT to secure communications. In case that the core network is available and a user device would like to only utilize GT to control its data access, the data is encrypted and decrypted by GT-related keys. In case that the core network is not available, each user device generates and issues LT-related keys to eligible devices. When the core network is available and a user device would like to utilize both GT and LT to control its data access, the attribute keys are generated under the control of both GT and LT.

Compared with existing key agreement schemes [13–16] in D2D communications, our proposed scheme has no need to negotiate session keys among pieces of User Equipment (UE), which greatly decreases corresponding communication overhead. Furthermore, it can realize secure communications no matter the core network is available or not. In particular, fine-grained access control is supported in this scheme. One UE can only communicate with the pieces of UE that satisfy with its access policy. Last but not the least, the proposed scheme protects identity privacy. Neither other pieces of UE nor ProSe App Server (PAS) that evaluates GT know the real ID of the UE. But in other key agreement schemes, the real identity of UE is often required for authentication in D2D communications.

Specifically, the contributions of the paper are summarized as below:

- We propose a new scheme to secure D2D communications on the basis of two-dimensional trust levels;
- Our scheme can control data access in D2D communications no matter whether the core network functionalities are available or not;
- We elaborate a trust evaluation method and demonstrate its practicability through performance analysis and simulations.
- We demonstrate that our scheme is secure and effective in terms of computation efficiency, communication cost, flexibility and scalability through security proof, performance analysis, and simulations.

The rest of the paper is organized as below. We review related work in Section 2. In Section 3, we introduce system structure, application scenarios and design goals. Section 4 provides detailed description of our scheme. We evaluate the performance of our scheme and discuss its advantages seriously through security proof, performance analysis, and simulations in Section 5. Finally, a conclusion is presented in the last section.

2. Related work

In this section, we review security schemes of D2D communications proposed in recent years.

2.1. Authentication and key management

Zhang et al. [13] proposed a secure data sharing protocol for D2D communications in LTE-Advanced (LTE-A), which realizes confidentiality, integrity, and non-repudiation. However, for data sharing, a content providing server must be installed in the cellular network and all mobile devices need to register into it. Although the content providing server is completely honest, it is exposed to attacks. Sheng et al. [14] proposed a scheme of shared secret key for D2D communications between two devices on the basis of Diffie–Hellman Key Exchange (DHKE). However, its authentication process is finally completed through visual or lingual comparison, which makes it inconvenient in practice. Moreover, the scheme cannot deal with more than two-device communications. Goratti et al. [15] presented a secure communication protocol that sets up direct links among D2D devices, which addresses compatibility with LTE specifications. In [17], Wang et al. proposed a universal authentication and key agreement protocol for securing D2D communications. This protocol supports user roaming and inter-operator operations. By adopting Diffie–Hellman Key Exchange (DHKE) algorithm and employing message authentication code, the protocol achieves privacy-preserving session key generation and mutual authentication between D2D users. But this protocol can only work in the context that two mobile devices are in the coverage of mobile core network entities (e.g., BSs). It cannot support group D2D communications.

Hsu et al. [16] introduced an authenticated key exchange protocol for D2D communications in both network-absent and network-covered scenarios. It achieves end-to-end security and accountable group anonymity to network operators and guarantees revocability and traceability. This group-anonymous D2D communication scheme adopted Diffie–Hellman key exchange, identity-based encryption, hash functions and symmetric encryption. In [18], Kwon et al. proposed a novel D2D authentication protocol based on Ciphertext-Policy Attribute-Based Encryption (CP-ABE). By utilizing CP-ABE, this scheme establishes a secure initial key and allows communicating partners to mutually authenticate and derive a link key in a secure manner in a multi-hop network environment. However, it does not consider trust evaluation and uses trust levels as specific attributes to control communication data access. This protocol cannot work when a group manager is not available.

2.2. Access control

Huang et al. [19] studied the access control issue of D2D communications in cellular networks. In order to reduce the interference caused by D2D communications to cellular communications, this scheme assigns different priority levels to the access requests of cellular communications and D2D communications. Yue et al. [20] innovatively proposed a secrecy capacity scheme for cellular communication access control. The above two schemes only considered cellular communication access control. In both studies, Fine-grained Access Control was supported in the security
public key cryptography is inefficient for relaying data to a symmetric key should be generated for any two mobile devices. Not available, symmetric keys are difficult to be managed because the core network is unable for D2D communications. Specifically, if the core network is not capable of processing the security and network domains as shown in Fig. 1 requests further investigation. However, conventional symmetric encryption schemes and public key cryptography based schemes are not suitable for D2D communications. Specifically, if the core network is not available, symmetric keys are difficult to manage because a symmetric key should also be generated for any two mobile devices. And public key cryptography is inefficient for relaying data to a group of devices because data should be respectively encrypted for each device. ABE is a new encryption scheme [10–12,25]. It is divided into two branches, Key-Policy ABE (KP-ABE) [11] and Ciphertext-Policy ABE (CP-ABE) [10,12]. In KP-ABE, a secret key is associated with an access policy based on a set of attributes. While in CP-ABE, ciphertexts are encrypted according to an access control policy, formulated as a Boolean formula over the attributes. The policy construction assures that only an entity whose attributes satisfy the access control policy can decrypt the ciphertext with its secret attribute key. ABE has been applied to secure data access in cloud and social networking [26], but it is seldom applied in D2D communications, especially on the basis of trust levels that play an important role in secure communications.

3. Problem statement

3.1. System structure

The system structure of the proposed scheme is compatible with the current standardization, as shown in Fig. 2. It involves two different entities: D2D UE and core network functionalities that consist of Evolved Packet Core (EPC), ProSe Function Server (PFS) and ProSe App Server (PAS). UE communicates with each other with the support of the core network functionalities (e.g., EPC) for device authentication and key management. We apply PFS to reserve a device’s unique real ID and its corresponding pseudonyms and to issue the device a trust token that contains the UE pseudonym, GT level and corresponding keys. UE can directly connect to PAS to access various D2D related services and report D2D communication records and local trust levels to it. One function that can be realized by the PAS is GT evaluation on UE in order to figure out its trustworthiness. We assign this task to the PAS because the trust evaluation is performed based on D2D communication behaviors regarding various D2D services.

We assume that the trust and security of Domain A and Domain B can be ensured based on existing security technologies, e.g., as specified in 3GPP. UE can authenticate with each other based on the trust token issued by PFS, which contains a device pseudonym and its GT level. Our scheme focuses on establishing a confidential channel for D2D communications in Domain C in order to flexibly support various D2D communication scenarios. This issue has not been well solved at present as discussed in Section 2 and our survey [9].

3.2. Application scenarios and use cases

We classify the application scenarios and use cases of D2D communications into three representative types, as illustrated in Fig. 3.

In-Coverage: the user equipment UE1 and UE2 are located in the coverage of network entities, which control the communications between two pieces of UE. A mobile operator controls over user authentication of D2D communications, resource allocation, connection establishment and security management. This D2D link shares a cellular licensed spectrum with a normal cellular connection (Device to BS) in coordination with the core network. Typical use cases of this scenario are local traffic offloading from local data services controlled by the core network, such as local content sharing and Machine to Machine (M2M) communications.

Relay-Coverage: when a user device (UE4 or UE5) is at the edge of BS coverage, it can relay its information through other covered devices (UE3) to communicate with the BS. Relay-Coverage can extend the coverage of a core network and further improve the Quality of Services at a cellular edge. In this case, the core network fully manages D2D communications as same as the “In-Coverage” scenario. The band used in the D2D link in this scenario is also the
cellular licensed spectrum shared with conventional communications.

**Out-of-Coverage:** when the core network is absent, D2D devices (UE6, UE7 and UE8) can connect and communicate with each other autonomously. This kind of scenario called “Out-of-Coverage”. A typical use case is Emergence Communication Network when an emergent situation occurs. For example, network entities have been damaged by natural disaster (e.g., flood or earthquake). This D2D communication scenario can serve as a technical component for providing such services as public protection, disaster relief, national security and public safety. This scenario looks similar to Mobile Ad-hoc Networks (MANET). However, their key difference lies in its D2D link works on a reserved cellular licensed spectrum for an LTE-based public safety network, while MANET works on an unlicensed Industrial, Scientific and Medical (ISM) spectrum, which makes it under more severe interference comparing with D2D communications.

3.3. Design goals and assumptions

Secure communication data among mobile devices is an important security issue in D2D communications. This issue has not been well solved in a flexible way to support all above D2D communication scenarios. For securing D2D communications, our design should achieve the following performance and security goals that are summarized based on our survey [9]:

- **Flexibility:** the proposed scheme can control communication data access no matter if the core network functionalities are available or not. It should flexibly support D2D data access in either a purely distributed way or a purely centralized way or both.
- **Privacy:** the proposed scheme should support identity privacy in D2D communications. Herein, we assume that devices use the pseudonyms issued by the PFS to communicate with each other and evaluate LT levels. A number of pseudonyms are generated for one UE and different pieces of UE are assigned different pseudonyms by the PFS. A unique anonymous ID is applied to take the place of the real ID to make it unknown to the PAS. Thus, UE cannot know the real IDs of other pieces of UE during D2D communications. When GT levels are evaluated at PAS, PAS sends the pseudonym of an evaluated UE to the PFS in order to get its unique anonymous ID. Thus, we protect identity privacy of UE from PAS.

- **Security:** only eligible nodes with sufficient trust levels can communicate with each other.
- **Lightweight:** the proposed scheme should be lightweight with regard to computation complexity and communication cost.

3.4. Preliminary and Notations

Bilinear pairing: Let $G$ and $G_T$ be two cyclic multiplicative groups with the same prime order $p$, that is, $|G| = |G_T| = p$. Let $g$ be a generator of $G$. Let us have a bilinear map $e : G \times G \rightarrow G_T$, with the following properties:

1. Bilinear: for all $u, v \in G$ and $a, b \in \mathbb{Z}_*$, $e(u^a, v^b) = e(u, v)^{ab}$.
2. Non-degenerate: $e(g, g) \neq 1$ for the generator $g$.
3. Computable: there is an efficient algorithm to compute $e(u, v)$ for any $u, v \in G$.

The notations used in the proposed scheme are described in Table 1.

4. Proposed Scheme

4.1. Trust Evaluation

We apply a hybrid trust evaluation method for realizing data access control based on trust levels issued by the PAS or devices. As shown in Fig. 4, at each UE, the Communication Behavior Observer records UE communication behaviors. The UE communicates with the PFS through the D2D Networking UI. The Communication Reporter and Voter transmit the communication behaviors and local trust levels to the PAS. And a device can also vote for other devices through it. The Trust Evaluator evaluates the local trust level of UE based the GT levels and local communication information. The Trust Extractor receives the trust tokens issued by the PFS. The Dataset stores the data of other functional blocks in a secure manner. And the UE utilizes the UE Profile Manager to preserve UE identity and other personal information.

At the PAS, the Information Receiver collects LT levels and communication records reported by UE. The PAS contacts the PFS to map the UE’s pseudonyms with its uniquely registered ID in order to perform GT level evaluation based on LT levels and the feedback from UE. For protecting the privacy of UE from the PAS, the PFS can generate a unique anonymous ID for each UE and inform this
ID to the PAS, where the Trust Generator calculates the UE’s GT levels accordingly. The Trust Distributor transmits the GT levels with their unique anonymous IDs to the PFS. Thus, the PFS can issue a number of new UE pseudonyms (by request or periodically) and trust tokens to UE and inform the new pseudonyms to all legitimate pieces of UE.

Unlinkability: In our scheme, when transmitting a message, some parameters such as pseudonym and trust value may be utilized by attackers to judge whether two messages are from a same node. However, in a session with the same node, the pseudonym and trust level are changed. The messages from the same node could apply different pseudonyms (up to the sender’s willingness). Only the PFS knows the unique real ID of UE. And there is no other information that the attacker can take advantage of. Thus, we claim that our scheme can provide unlinkability to some extent. Note that a number of pseudonyms are generated for one UE and different pieces of UE apply different pseudonyms.

Trust Evaluation at UE
In this part, we propose the algorithms for trust evaluation at UE and PAS. At UE, the local trust evaluation is based on the communication time ct, times cn, and user votes cv during communications with UE pseudonyms.

To generate the local trust T (i → j) made by UEi to UEj, we consider to aggregate three trust impact factors together. The first is the sum of previous local trust value T′ (i → j) and the general trust value T (j), i.e., T (i → j) + T (j), which serves as the initial trust value for current D2D communications. The second is the trust value generated on the basis of current D2D communication experiences. Since there could be multiple votes during the communications, we integrate them by adding the product of the votes from UEi to UEj, the communication time ct (i, j), and times cn (i, j); at the lth vote. ct (i, j) and cn (i, j) at the lth vote relates to the preciseness of the opinion (cv (i, j)) of UE. The third factor is modeled based on the votes on UEi provided by other pieces of UE than UEi, which is certified by the opinion deviation factor od (i ↔ k, j) that indicates the difference of opinions between UEj and UEi with regard to UEj. The opinion deviation factor indicates the opinion deviation of two pieces of UE on a target UE. Applying this factor makes it easy to figure out the UE that hold different opinions from the trust evaluating UE. Thus, applying it in trust evaluation can avoid the negative influence of bad mouthing attack. We apply Formula (1) to perform local trust evaluation by considering the above factors that are available at each UE. Formula (2) is used for calculating od (i ↔ k, j).

\[
T (i ↔ j) = \frac{1}{2}(T (i → j) + T (j)) + \beta f \left( \frac{1}{k} \sum_{l=1}^{K} \left[ cv (i → j, l) * ct (i, j, l) * cn (i, j, l) \right] \right),
\]

\[
y f \left( \frac{1}{k} \sum_{l=1}^{K} \left( \frac{1}{l} \sum_{i=1}^{l} cv (k → j, l) * ct (k, j, l) * cn (k, j, l) \right) \right) + od (i ↔ k, j) \]

\[
o d (i ↔ k, j) = 1 - f \left( \frac{1}{k} \sum_{l=1}^{K} \frac{1}{l} \sum_{i=1}^{l} cv (k → j, l) * ct (k, j, l) * cn (k, j, l) \right) \}
\]

where \( f(x) = 2/(1 + e^{-x}) - 1 \) is the Sigmoid function that is used to normalize an arbitrary value into \((0, 1)\). \( l \) denotes the total number of votes made by UEj on UEi; \( l ' \) denotes the total number of votes by UEj on UEi. \( \alpha, \beta, \gamma \) are parameters to indicate the weights of different contributions. Note that \( \alpha + \beta + \gamma = 1 \).

Trust Evaluation at ProSe App Server (PAS)
Formula (3) is applied to generate the personalized trust of UEj \( T (i → j) \) evaluated by UEj by considering the voting-afterwards and time decaying. The voting-afterwards is the vote on UE sent to the PAS after D2D communications based on interaction experiences.

\[
T (i → j) = \frac{1}{O} \sum_{m} T (i → j)_{LM} + V (j)_{l} \cdot e^{-\frac{tm}{m}},
\]

where \( O = \sum_{m} T (i → j)_{LM} \cdot e^{-\frac{tm}{m}} \cdot V (j)_{l} \) is the voting-afterwards of UEj on UEi at time \( t_{m} \); \( t \) is the calculation time of UE trust; parameter \( \tau \) is used to control time decaying. \( T (i → j)_{LM} \) is the local trust of UEj reported by UEj at time \( t_{m} \), with the voting-afterwards \( V (j)_{l} \)' attached. If \( V (j)_{l} \) is not provided by the UE, we set \( V (j)_{l} = 0.5 \). Note that \( V (j)_{l} \in [0, 1] \) and \( T (i → j)_{LM} \in [0, 1] \).

To get the general trust of UEj, denoted as \( T (j) \), we aggregate \( T (i → j) \) of all UE based on (4). The general reputation \( T (j) \) plays as the credibility of \( T (i → j) \) in the aggregation. Meanwhile, we also consider the influence of the number of trust evaluation contributors on the general trust evaluation, since the more contributors, the more convinced the aggregation result is.

\[
T (j) = \frac{g (K)}{K} \sum_{i=1}^{K} T (i) * T (i → j), (i \neq j).
\]

Table 1

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK, MR</td>
<td>The master key pairs of PFS;</td>
</tr>
<tr>
<td>ID</td>
<td>The unique identifier of UE;</td>
</tr>
<tr>
<td>Pseu</td>
<td>The pseudonym of UE;</td>
</tr>
<tr>
<td>SKsys</td>
<td>The secret key of the PFS;</td>
</tr>
<tr>
<td>H(idm)</td>
<td>The hash function like SHA_1 with regard to SKsys;</td>
</tr>
<tr>
<td>SK(i, u)u</td>
<td>The key pairs of UE u;</td>
</tr>
<tr>
<td>GTi</td>
<td>The general trust level i;</td>
</tr>
<tr>
<td>LTi</td>
<td>The local trust level i;</td>
</tr>
<tr>
<td>It</td>
<td>The maximum level of GT;</td>
</tr>
<tr>
<td>LT</td>
<td>The maximum level of LT;</td>
</tr>
<tr>
<td>PK(GT)</td>
<td>The public attribute key based on GT;</td>
</tr>
<tr>
<td>SK(GT, u)u</td>
<td>The secret attribute key based on GT, that PFS generates for UE u;</td>
</tr>
<tr>
<td>PK(LT, u)u</td>
<td>The public attribute key based on LT, generated by UE u;</td>
</tr>
<tr>
<td>SK(LT, u, u′)u</td>
<td>The secret attribute key based on LT, that u generate for u’;</td>
</tr>
<tr>
<td>M</td>
<td>The message that UE broadcasts;</td>
</tr>
<tr>
<td>AA</td>
<td>The access control policy.</td>
</tr>
</tbody>
</table>
trust values at UE and the PAS, we can set the levels of LT and GT for each UE. We can then configure the total number of trust levels.

4.2. Scheme of access control

We secure D2D communications with the support of either distributed access control or centralized access control or both. Herein, the GT level is evaluated by PAS according to the LT levels and communication feedback based on the unique UE anonymous ID. LT is evaluated by UE itself according to D2D communication experiences and behaviors based on a UE pseudonym.

To realize the proposed scheme, we apply CP-ABE to encrypt and decrypt data encryption keys based on trust evaluation. Optionally, we can also use KP-ABE to implement our scheme. But applying CP-ABE saves computation cost of secret attribute key generation.

We first describe the basic algorithms used in our scheme: Setup, UERegistration, IssueGeneralTrustPK, IssueGeneralTrustSK, IssueLocalTrustPK, IssueLocalTrustSK, Encrypt, and Decrypt. The notations used in the scheme are described in Table 1.

1. Setup:
This algorithm is conducted in the PFS and generates system parameters like PK and MK, as well as a secret key for the PFS. The details are described in Algorithm 1.

Algorithm 1. Setup
1. Select a symmetric elliptic curve \( e: G \times G \rightarrow G_T \);
2. Randomly select an element \( P \in G \);
3. Randomly select an element \( y \in Z_r \);
4. Generate \( PK = (G, G_T, e, g, P, e(g, g)^y) \);
5. Randomly select an element \( SK_{PFS} \in Z_r \), and a hash function \( H_{PFS} : \{0, 1\}^* \rightarrow Z_r \);
6. Generate \( MK = (g^y, SK_{PFS}) \).

Herein, MK and \( H_{PFS} \) are kept secret by the PFS, while PK is public for each UE.

2. UERegistration(PK, MK, ID)
This algorithm is conducted in the PFS. It assigns a unique anonymous ID and reserves a number of pseudonyms for UE. PFS also generates a public key and a secret key of UE (\( PK_u \) and \( SK_u \)), and issues them to the eligible UE subsequently. The details are described in Algorithm 2. Note that a number of pseudonyms are generated for one UE and different pieces of UE are assigned different pseudonyms by the PFS.

Algorithm 2. NodeRegistration
1. Generate pseudonyms for UE: \( Pseu_u \);
2. Select a random secret for UE: \( s_u \in Z_r \);
3. Generate a public key of the UE: \( PK_u = g^{s_u} \);
4. Generate a secret key of the UE: \( SK_u = MK \cdot P^{s_u} = g^y \cdot P^{s_u} \).

3. IssueGeneralTrustPK(PK, MK, SK_{PFS})
This algorithm is conducted in the PFS. It generates \( PK \) whenever \( PK \) reaches the period of its validity. It checks the access policy of PFS first. And if the result is positive, it returns \( PK \) that consists two parts for each eligible general trust level \( i \in [0, I_g] \), where \( I_g \) is the max level of GT, else it returns NULL. The details are depicted in Algorithm 3.

Algorithm 3. IssueGeneralTrustPK
1. \( PK(GT_i) = PK(GT_i) \cdot g^{H_{SK_{PFS}}(GT_i)} \); \( PK(GT_i) \cdot g^{H_{SK_{PFS}}(GT_i)} \); \( PK(GT_i) = PK(GT_i) \).
2. \( PK(GT_i) = PK(GT_i) \).

4. IssueGeneralTrustSK(PK, GT, SK_{PFS}, PK_u)
This algorithm is executed in the PFS whenever UE registers into the PFS or SK \( (GT, u) \) becomes invalid. It generates \( SK \) for the UE according to its attribute \( GT_u \) as described in Algorithm 4.

Algorithm 4. IssueGeneralTrustSK
\[
SK(GT_u, u) = PK_u H_{SK_{PFS}}(GT_u) = g^{u H_{SK_{PFS}}(GT_u)}.
\]

Significantly, UE can verify the validity of this algorithm through the equation as below:
\[
e(SK_u, PK(GT_u)) = PK(GT_u) \cdot e(P, SK(GT_u, u)).
\]

5. IssueLocalTrustPK(PK, LT, SK)
This algorithm is conducted in UE whenever LT controls the data access policy. It randomly chooses a hash function \( H_{SK_{u}} : \{0, 1\}^* \rightarrow Z_r \), which is different from \( H_{SK_{PFS}} \). Then it generates \( PK \) for each LT level, \( i \in [0, I_l] \), where \( I_l \) is the max level of LT. The details are depicted in Algorithm 5.

For simplification, we set \( I_g = I_l = I \) in our scheme.

Algorithm 5. IssueLocalTrustPK
1. \( PK(LT_i, u') = PK(LT_i, u') \cdot g^{H_{SK_{u}}(LT_i)} \);
2. \( PK(LT_i, u') \); \( PK(LT, u) = \{PK(LT, u)\} \).

6. IssueLocalTrustSK(PK, LT, SK)
This algorithm is conducted in UE whenever LT controls its data access policy. It checks LT of \( u' \) evaluated by UE. If the LT of \( u' \) satisfies UE's requirements, then UE generates and issues local trust secret key \( SK(LT_i, u, u') \) as described in Algorithm 6.

Algorithm 6. IssueLocalTrustSK
1. \( SK(LT_i, u, u') = PK_{u'} H_{SK_{u}}(LT_i) = g^{u'} H_{SK_{u}}(LT_i) \).

7. Encrypt(PK, M, AA, PK(GT), PK(LT, u))
This algorithm is conducted in UE whenever it would like to send messages. It encrypts a message by applying policy AA to generate the corresponding ciphertext CT. Significantly, it can utilize either GT or LT or both to create the access policy. When the UE only utilizes GT to control its data access, we can simplify this algorithm as Encryption(PK, M, AA, PK(GT)) and AA is defined as \( AA = \bigvee_{j=g^j GT} g^{j I_G} \), where \( I_G \) represents the minimum GT level in which the UE is specified for data access control. When the UE only utilizes LT to control its data access, we can simplify this algorithm as Encrypt(PK, M, AA, PK(LT, u)) and AA is described as \( AA = \bigvee_{j=I_l} g^{j I_L} \), where \( I_l \) is the threshold that the UE sets for data access control. When a user device utilizes both GT and LT to control its data access, this algorithm keeps its original form and we describe AA as \( AA = \bigvee_{j=I_l} g^{j I_L} \). For example, in the case that both GT and LT are applied to control data access and \( I = 4 \), \( I_G = I_l = 3 \), AA can be indicated as \( AA = (GT = 3 \land LT = 3) \lor (GT = 3 \land LT = 4) \lor (GT = 4 \land LT = 3) \lor (GT = 4 \land LT = 4) \). The UE with \( GT \geq 3 \) and \( LT \geq 3 \) can decrypt the message. This algorithm returns CT, that consists three parts for each attribute involved in AA. The details are depicted in Algorithm 7.
When we only use GT to control the data access policy, we set
\( PK(LT_k, u) = PK(LT_k, u') = 1 \) and we set \( PK(GT_j) = PK(GT_j) = 1 \) if data access is only controlled by LT.

8. **Decrypt** (PK, CT, AA, SKPFS, SKGT, u', SKLT, u')

This algorithm is conducted in UE whenever it reserves a message from UE. The algorithm first checks that AA belongs to which kind of access control policy, and then detects whether the given attributes of UE meet the demands of AA. If the result is eligible, it returns the original message to UE, otherwise, it returns NULL.

The details are depicted in Algorithm 8.

**Algorithm 8. Decrypt**

1. If AA is satisfied, \( M = E_t = e \left( E'_t, SKPFS, SKLTk, u' \right) \cdot e \left( E''_t, SKLTk, u' \right) \).

2. Else, return NULL.

When we only use GT to control the data access policy, we set \( PK(LT_k, u) = PK(LT_k, u') = 1 \) and we set \( PK(GT_j) = PK(GT_j) = 1 \) if data access is only controlled by LT.

8. **Decrypt** (PK, CT, AA, SKPFS, SKGT, u', SKLT, u')

This algorithm is conducted in UE whenever it reserves a message from UE. The algorithm first checks that AA belongs to which kind of access control policy, and then detects whether the given attributes of UE meet the demands of AA. If the result is eligible, it returns the original message to UE, otherwise, it returns NULL.

The details are depicted in Algorithm 8.

**Algorithm 8. Decrypt**

1. If AA is satisfied, \( M = E_t = e \left( E'_t, SKPFS, SKLTk, u' \right) \cdot e \left( E''_t, SKLTk, u' \right) \).

2. Else, return NULL.

When we only use GT to control the data access policy, we set \( PK(LT_k, u) = PK(LT_k, u') = 1 \) and we set \( PK(GT_j) = PK(GT_j) = 1 \) if data access is only controlled by LT.

8. **Decrypt** (PK, CT, AA, SKPFS, SKGT, u', SKLT, u')

This algorithm is conducted in UE whenever it reserves a message from UE. The algorithm first checks that AA belongs to which kind of access control policy, and then detects whether the given attributes of UE meet the demands of AA. If the result is eligible, it returns the original message to UE, otherwise, it returns NULL.

The details are depicted in Algorithm 8.

**Algorithm 8. Decrypt**

1. If AA is satisfied, \( M = E_t = e \left( E'_t, SKPFS, SKLTk, u' \right) \cdot e \left( E''_t, SKLTk, u' \right) \).

2. Else, return NULL.

When we only use GT to control the data access policy, we set \( PK(LT_k, u) = PK(LT_k, u') = 1 \) and we set \( PK(GT_j) = PK(GT_j) = 1 \) if data access is only controlled by LT.

8. **Decrypt** (PK, CT, AA, SKPFS, SKGT, u', SKLT, u')

This algorithm is conducted in UE whenever it reserves a message from UE. The algorithm first checks that AA belongs to which kind of access control policy, and then detects whether the given attributes of UE meet the demands of AA. If the result is eligible, it returns the original message to UE, otherwise, it returns NULL.

The details are depicted in Algorithm 8.

**Algorithm 8. Decrypt**

1. If AA is satisfied, \( M = E_t = e \left( E'_t, SKPFS, SKLTk, u' \right) \cdot e \left( E''_t, SKLTk, u' \right) \).

2. Else, return NULL.

Significantly, it is easy to verify the validity of this algorithm as below [12]: Let \( \alpha_t = H_{SKPFS} \left( GT_j \right) + H_{SKLTk} \left( LT_k \right) \). And then, \( E_t = M \cdot e(g, g)^{\alpha_t R_{K_i}} \).

**Algorithm 7. Encrypt**

1. \( CT_i = < E_t = M \cdot (PK(GT_j)^{\alpha_t} \cdot PK(LT_k, u')^{R_{K_i}})^{R_{i}}, \quad E_t' = P^{R_{i}}, \quad E_t'' = (PK(GT_j) \cdot PK(LT_k, u')^{R_{i}})^{R_{i}} > \). 

2. \( CT = \{ CT_i \} \) (herein, \( i \) refers to the \( i \)th V conjunction that AA consists of).

When we only use GT to control the data access policy, we set \( PK(LT_k, u) = PK(LT_k, u') = 1 \) and we set \( PK(GT_j) = PK(GT_j) = 1 \) if data access is only controlled by LT.

8. **Decrypt** (PK, CT, AA, SKPFS, SKGT, u', SKLT, u')

This algorithm is conducted in UE whenever it reserves a message from UE. The algorithm first checks that AA belongs to which kind of access control policy, and then detects whether the given attributes of UE meet the demands of AA. If the result is eligible, it returns the original message to UE, otherwise, it returns NULL.

The details are depicted in Algorithm 8.

**Algorithm 8. Decrypt**

1. If AA is satisfied, \( M = E_t = e \left( E'_t, SKPFS, SKLTk, u' \right) \cdot e \left( E''_t, SKLTk, u' \right) \).

2. Else, return NULL.

Significantly, it is easy to verify the validity of this algorithm as below [12]: Let \( \alpha_t = H_{SKPFS} \left( GT_j \right) + H_{SKLTk} \left( LT_k \right) \). And then, \( E_t = M \cdot e(g, g)^{\alpha_t R_{K_i}} \).

**Algorithm 7. Encrypt**

1. \( CT_i = < E_t = M \cdot (PK(GT_j)^{\alpha_t} \cdot PK(LT_k, u')^{R_{K_i}})^{R_{i}}, \quad E_t' = P^{R_{i}}, \quad E_t'' = (PK(GT_j) \cdot PK(LT_k, u')^{R_{i}})^{R_{i}} > \). 

2. \( CT = \{ CT_i \} \) (herein, \( i \) refers to the \( i \)th V conjunction that AA consists of).

When we only use GT to control the data access policy, we set \( PK(LT_k, u) = PK(LT_k, u') = 1 \) and we set \( PK(GT_j) = PK(GT_j) = 1 \) if data access is only controlled by LT.

8. **Decrypt** (PK, CT, AA, SKPFS, SKGT, u', SKLT, u')

This algorithm is conducted in UE whenever it reserves a message from UE. The algorithm first checks that AA belongs to which kind of access control policy, and then detects whether the given attributes of UE meet the demands of AA. If the result is eligible, it returns the original message to UE, otherwise, it returns NULL.

The details are depicted in Algorithm 8.

**Algorithm 8. Decrypt**

1. If AA is satisfied, \( M = E_t = e \left( E'_t, SKPFS, SKLTk, u' \right) \cdot e \left( E''_t, SKLTk, u' \right) \).

2. Else, return NULL.

When we only use GT to control the data access policy, we set \( PK(LT_k, u) = PK(LT_k, u') = 1 \) and we set \( PK(GT_j) = PK(GT_j) = 1 \) if data access is only controlled by LT.

8. **Decrypt** (PK, CT, AA, SKPFS, SKGT, u', SKLT, u')

This algorithm is conducted in UE whenever it reserves a message from UE. The algorithm first checks that AA belongs to which kind of access control policy, and then detects whether the given attributes of UE meet the demands of AA. If the result is eligible, it returns the original message to UE, otherwise, it returns NULL.

The details are depicted in Algorithm 8.

**Algorithm 8. Decrypt**

1. If AA is satisfied, \( M = E_t = e \left( E'_t, SKPFS, SKLTk, u' \right) \cdot e \left( E''_t, SKLTk, u' \right) \).

2. Else, return NULL.
Step 2. UE Registration: UE registers into the core network system with its unique identifier ID. PFS assigns a number of pseudonyms and a unique anonymous ID to it. PFS further generates PK_u and SK_u, and issues them to UE subsequently.

Step 3. Trust Evaluation: PAS collects D2D communication records (e.g., local trust values and communication feedback). It requests unique anonymous IDs from the PFS, which maps UE pseudonyms to their unique anonymous IDs and provides the IDs to the PAS. The PAS then evaluates the GT levels based on the UE unique anonymous IDs.

Step 4. Issue Attribute Keys: The PAS sends the GT evaluation results to the PFS, which runs the IssueLocalTrustPK algorithm to create PK_lu and SK_lu based on the PAS policy, and then issue them to the eligible UE when previous keys have reached the end of validity period.

Step 5. Secure D2D Communications (through encryption): At UE, it encrypts D2D communication data using a symmetric key (DEK) [11] first and then encrypts DEK based on personal policy on LT by calling the Encrypt algorithm and then generates a data frame including the encrypted data and a UE pseudonym (Pseu), described as Frame = \{Pseu, AA, CT, (Data)|DEK\}.

Step 6. Secure D2D Communications (through decryption): If UE is eligible to access the encrypted data (e.g., its GT level exceeds the access threshold set by the sender), it can decrypt the data with its personal secret key about LT issued by the PFS by calling the Decrypt algorithm first and then decrypt the encrypted data using DEK.

Step 7. Trust Re-evaluation: Later on, UE can send LT levels and feedback votes to the PAS. These records are linked to UE pseudonyms. At the PAS, these records are processed and aggregated for evaluating the GT of the UE based on its unique anonymous ID (after getting them from the PFS). Then the process goes to Step 2.

Case 2. Data Access only Controlled by LT

In the case that we only utilize LT to control data access when the core networks are not available, the attribute keys based on LT are generated and issued by UE. The UE issues secret attribute keys to other pieces of UE that meet the demands of its access policy. This secure channel establishment can be applied into the scenarios of Out-of-Coverage or Relay-Coverage with distrusted relaying UE. The procedure is shown in Fig. 6 and described as below. The procedures of System Setup and UE Registration are the same as Case 1.

Step 1. Communication Initiation: UE sets its access control policy AA with LT and a threshold \(LT_{th}\), which decides whether to communicate with other pieces of UE. Meanwhile, it runs the IssueLocalTrustPK algorithm to generate PK_lu and SK_lu.

Step 2. Secure D2D Communications (through encryption): Similar to Case 1, the UE encrypts data using a symmetric key (DEK) first and then encrypts DEK based on personal policy on LT by calling the Encrypt algorithm and then generates a data frame, described as Frame = \{Pseu, AA, CT, (Data)|DEK\}.

Step 3. Encryption Key Request: If UE receives the Frame from UE, the UE checks whether \(LT_{th}\) satisfies with AA, it issues IssueLocalTrustSK and issues SK_lu to the UE.

Step 4. Issue Secret Attribute Key: After the UE receives the request from the UE, it checks whether \(LT_{th}\) exceeds the demand of AA. If the \(LT_{th}\) satisfies with AA, it issues IssueLocalTrustSK and issues SK_lu to the UE.

Step 5. Secure D2D Communications (through decryption): Any eligible UE can decrypt the data with its personal secret key about LT issued by the UE by calling the Decrypt algorithm first and then decrypting the encrypted data using DEK.

Step 6. Trust Evaluation: During D2D communications, UE collects D2D communication records and evaluate LT based on UE pseudonyms.

Case 3. Data Access Controlled by both GT and LT
Fig. 7. D2D communications in a secure channel controlled by both GT and LT.

**We utilize both GT and LT to control data access when the core network is available in this case. This secure channel establishment can be applied to the scenarios of In-Coverage or Relay-Coverage with trusted relaying UE. The procedure is illustrated in Fig. 7 and described below.**

The first four steps are System Setup, UE Registration, Trust Evaluation, Issue Attribute Keys. They are the same as those in Case 1.

**Step 5. Secure D2D Communications (through encryption):** UE creates its access control policy AA on the basis of GT and LT. It generates PK (LT, u) by calling IssueLocalTrustPK and encrypts data with DEK. It further encrypts DEK using two attribute public keys by calling Encrypt(PK(M, AA, PK(GT), PK(LT, u))). Then it broadcasts its data frame to other pieces of UE.

If UE wants to communicate with the UE, it should send a request to the UE. Step 6 and Step 7 are Decryption Key Request and Issue Secret Attribute Key. They are the same as those in Case 2.

**Step 8. Secure D2D Communications (through decryption):** Any eligible UE can decrypt the data with both SK(GT, u) and SK(LT, u, u') by calling the Decrypt algorithm first to get DEK and then decrypt the encrypted data using DEK.

**Step 9. Trust Evaluation:** During D2D communications, UE collects D2D communication records and evaluate LT based on UE pseudonyms. Later on, UE can send D2D LT levels and feedback votes to the PAS. These records are linked to UE pseudonyms. At the PAS, these records are processed and aggregated for evaluating the GT of the UE based on its unique anonymous ID (after getting them from the PFS). Then the process goes to Step 2.

**4.4. Further discussion**

D2D communications over roaming and inter-operators are very common in the real world [17]. In the cases of roaming and inter-operators, the home network should first authenticate its subscribed devices and then securely negotiate a common set of system parameters with regard to PK, MK and shared SK_{PFS}, then
we can apply the designed algorithms to perform secure D2D communications based on the proposed scheme. In another line of our study [17], we proposed a Universal Authentication and Key Agreement protocol for D2D communications (UAKE-D2D) to achieve secure D2D communications among two devices with support on user roaming and inter-operator operation. The scheme proposed in this paper can support trustworthy group communications.

5. Security analysis and performance evaluation

5.1. Security proof

The security goal of our scheme is to realize effective data access control that only the UE with eligible attributes can communicate with other pieces of UE.

Fine-grained Access Control: We can apply either GT or LT or both to control the access policy in our scheme, and realize fine-grained access control in the security domain C. We use dimensions of trust levels instead of hierarchical attribute structure to release the complexity of ABE. We utilize a number of trust impact factors to evaluate trust levels, and thus simplify the access policy by making it only based on trust levels. Our scheme greatly reduces computation complexity since the attributes that are adopted in encryption and decryption are simple. It can be flexibly applied into all three scenarios of D2D communications.

Data Confidentiality: The propose scheme first encrypts D2D communication data using DEK and then encrypts DEK with the Encrypt algorithm. In the case that the symmetric key encryption (e.g., AES) is secure, the confidentiality of our scheme only depends on the Encrypt algorithm based on ABE.

We prove the security of our scheme by setting up a game of two parties. One is an adversary A who wants to get advantages in the game, and the other is challenger C who wants to control data access policy as either the core network or UE in D2D communications.

Setup: The challenger C calls the Setup algorithm to generate system parameters and issues PK to the adversary A.

Phase 1: A sends a request to C for getting a number of UE keys. C calls the UERegistration algorithm to generate and issue PK for each piece of UE. And each UE can ask C for attribute keys according to either GT or LT or both by calling corresponding algorithms.

Challenge: A sends an access policy and two messages (M1 and M1) to C. None of UE registered in Phase 1 satisfies the access policy. Otherwise, C aborts. Then C throws a coin b to set b as 0 or 1 and encrypts the message MB by calling the Encrypt algorithm. Finally, it submits the ciphertext to A.

Phase 2: Like Phase 1, A can request for an arbitrary number of SK (GT, u) and/or SK (LT, u, v) for each UE. The only limited factor is the fact that the trust levels associated with the secret attribute keys of each UE cannot satisfy the access policy. Otherwise, C aborts.

Guess: A guesses b as b.

We describe the advantage of A using a probabilistic equation like Advantage = Pr [b = b] − 0.5, where 0.5 is probability of correct random guess. If the advantage is negligible in polynomial time, the proposed scheme is secure.

We define a modified game [26] in order to further prove the security of the proposed scheme. First, we assume that adversary A1 in the original game has nonnegligible advantages in polynomial time. With the help of A1, we can construct another adversary A2 in the modified game, which has nonnegligible advantages in polynomial time. Then, if we prove such A2 is not available, there exists no polynomial-time adversary with nonnegligible advantages in the original game.

5.2. Performance analysis

In this section, we analyze the performance of both trust evaluation method and access control scheme in terms of computation complexity, communication cost, flexibility and scalability.

Table 2 Computation complexity.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Computation complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setup</td>
<td>O(1)</td>
</tr>
<tr>
<td>UERegister</td>
<td>O(1)</td>
</tr>
<tr>
<td>IssueGeneralTrustPK</td>
<td>O(1)</td>
</tr>
<tr>
<td>IssueGeneralTrustSK</td>
<td>O(1)</td>
</tr>
<tr>
<td>IssueLocalTrustPK</td>
<td>O(1)</td>
</tr>
<tr>
<td>IssueLocalTrustSK</td>
<td>O(1)</td>
</tr>
<tr>
<td>Encrypt</td>
<td>O(N)</td>
</tr>
<tr>
<td>Decrypt</td>
<td>O(1)</td>
</tr>
</tbody>
</table>

I denotes the maximum trust level; N denotes the number of V conjunctions that AA consists of.

The way to create A2 from A1 was provided in [26]. We can deduce that A2 has a nonnegligible advantage 0.5e from the fact that A1 has a nonnegligible advantage ϵ. Furthermore, in [12], it has been proved that such A2 does not exist. So, we can conclude that all of polynomial-time adversaries A in the original game have negligible advantages in our security model. Thus, our scheme achieves security.

The propose scheme first encrypts D2D communication data using DEK and then encrypts DEK with the Encrypt algorithm. The complexity of Encrypt is up to the size of data and cannot be clear away in any cryptographic algorithm. The Encrypt algorithm contains an exponentiation operation of G and G, which have negligible advantages in our security model.

The IssueGeneralTrustSK and IssueLocalTrustSK algorithms are both contain one exponentiation operation of G and a same operation of G for each attribute. So, the computation complexity of these two algorithms is O(1), where I is the maximum trust level.

In our scheme, a data sender encrypts D2D communication data using DEK first and then encrypts DEK by calling the Encrypt algorithm. The complexity of DEK is up to the size of data and cannot be clear away in any cryptographic algorithm. The Encrypt algorithm contains an exponentiation operation of GT and two same operations of G for each conjunction in AA. Thus, its computation complexity is O(N), where N indicates the number of V conjunction involved in AA.

For each communication message, the Decrypt algorithm contains only two time-consuming bilinear pairing operations. There is no impact with the access policy. So the computation complexity of Decrypt is O(1).

We summarize the entire computation complexity in Table 2. Without considering the impact of the number of UE pieces and messages, the computation complexity of our scheme is mainly up to the maximum number of trust levels and the access policy.

Communication Cost: In the trust evaluation, the communication cost derives from the communication time ct, times cn, user votes cv and LT levels. Every factor is limited and regarded as an integer value or a floating value that generally occupies 32 bits in a 32-bit system.
The main communication cost in the access control scheme is ciphertext size. We encapsulate CT into the Frame = \{Pseud, AA, CT, [Data]DEK\}. Herein, CT = \{CT_i\}, where CT is consist of three elements. Each \(\lor\) conjunction involved in AA will generate a CT, and AA contains N \(\lor\) conjunctions. Considering the maximum trust level is not very big in practice, the size of the frame is reasonable. Furthermore, we can use trust evaluation and access policy to decide whether UE should generate and issue secret attribute keys about LT to other pieces of UE. It further decreases the communication cost.

**Scalability:** Using ABE, the data can be accessed by a number of UE pieces that satisfies the access policy. This scheme supports data access by many pieces of UE at the same time. It supports unicast, multicast and broadcast to any number of communication devices. Community based group communications can be fully supported in a secure way by applying the proposed scheme, which has not been well solved in the literature [8]. Furthermore, reduced complexity benefits scheme scalability.

**Flexibility:** The proposed scheme can control data access in D2D communications no matter if the core network functionalities are available or not. It supports purely distributed access control, purely centralized access control or hybrid access control by establishing a secure communication channel controlled by either the general trust or the local trust or both for supporting all three D2D communication scenarios: In-Coverage, Relay-Coverage and Out-of-Coverage. Meanwhile, the scheme supports one-to-one, one-to-many, and many-to-one D2D communications, as well as group communications in different D2D scenarios. For example, in Case 3 that data access controlled by both GT and LT, UE_A, UE_B and UE_C are all located in the same local network and have access policy \(AA = (GT = 3 \land LT = 3) \lor (GT = 3 \land LT = 4) \lor (GT = 4 \land LT = 3) \lor (GT = 4 \land LT = 4)\). The trust levels of these pieces of UE all satisfy AA. Then, they can communicate with each other as a group. Similar types of D2D communications could also happen in Case 1 and Case 2.

5.3. Performance evaluation

**Performance Test of Trust Evaluation**

First, we measure the impact of each trust influencing factor on LT value \(T(i \leftrightarrow j)\) and GT value \(T(j)\), respectively. In order to evaluate the impact of every factor on \(T(i \leftrightarrow j)\), we set \(\alpha = 0.4, \beta = 0.4\) and \(\gamma = 0.2\) in our performance test. This setting is appropriate because previous local trust value \(T(i \leftrightarrow j)\) and the general trust value \(T(j)\) are important references on current trust evaluation. Current D2D communication experiences are decisive factors. Thus, we gave them high weights. We set \(\gamma = 0.2\) for reducing the negative influence of bad mouthing attacks.

We first evaluated \(T(i \leftrightarrow j)\) with different previous local trust values \(T'(i \rightarrow j)\) and the general trust values \(T(j)\). We set the second factor that refers to current D2D communication experiences \(f(\sum_{i=1}^{L} \{ct(i,j) \land ct(i,j) \land cn(i,j)\})\) and the third factor that refers to votes from other pieces of UE \(f(\sum_{k=1}^{K} \{ct(k,j) \land ct(k,j) \land cn(k,j)\}) = od(i \leftrightarrow k, j))\) as the maximum value 1 after normalization to simplify the simulation process. It shows that \(T(i \leftrightarrow j)\) changes linearly with \(T'(i \rightarrow j)\) or \(T(j)\), which conforms to our design in Fig. 8.

Then we set \(T'(i \rightarrow j), T(j)\) and the third factor as the maximum value 1. The influence of current D2D communication experiences on \(T(i \leftrightarrow j)\) is shown in Fig. 9(a). The impact of votes from other pieces of UE on \(T(i \leftrightarrow j)\) is shown in Fig. 9(b), in which the LT value is close to 1 unlimitedly with the growth of the number of positive votes.

Bad mouthing attack denotes that a dishonest UE deliberately cracks up a bad UE or defame a good one. Although this proposed scheme cannot detect bad mouthing attack, it can resist it to some extent. We assume that dishonest votes occupy 0%, 10%, 20%, 30% and 40% of all votes from other pieces of UE to UE. We tested two cases: (a) honest votes are \(1/L \sum_{i=1}^{L} ct(k \rightarrow j) \land ct(k \rightarrow j) \land cn(k, j)\) that maps to 0.9 after normalization \((f(0.2) = 0.9)\), dishonest votes are \(1/L \sum_{i=1}^{L} ct(k \rightarrow j) \land ct(k \rightarrow j) \land cn(k, j)\) that maps to 0.1 after normalization \((f(0.2) = 0.1)\), (b) honest votes are 0.1 and dishonest votes are 0.9 after normalization.

**Table 3**

<table>
<thead>
<tr>
<th>Attacker percentage</th>
<th>0%</th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) Honest votes are 0.9 and dishonest notes are 0.1 after normalization</td>
<td>LT Value</td>
<td>0.9</td>
<td>0.894</td>
<td>0.885</td>
<td>0.875</td>
</tr>
<tr>
<td>Relative error</td>
<td>0</td>
<td>0.7%</td>
<td>1.6%</td>
<td>2.8%</td>
<td>4%</td>
</tr>
<tr>
<td>(b) Honest votes are 0.1 and dishonest notes are 0.9 after normalization</td>
<td>LT Value</td>
<td>0.1</td>
<td>0.1015</td>
<td>0.103</td>
<td>0.1045</td>
</tr>
<tr>
<td>Relative error</td>
<td>0</td>
<td>1.5%</td>
<td>3%</td>
<td>4.5%</td>
<td>6%</td>
</tr>
</tbody>
</table>

Table 3 shows that the performance of the LT evaluation under the bad mouthing attack in the above two cases. For example, in Case 1, when the dishonest votes occupy 10% of all votes, \(od(i \leftrightarrow k, j)_{dishonest} = 1 - f(0.2) = 0.1213\) and \(od(i \leftrightarrow k, j)_{honest} = 0.2\). Then \(T(i \leftrightarrow j) = 1/2*0.4/0.9 + 0.4/0.9 + 0.2*0.9 = 0.894\). We observe that the dishonest votes have little influence on the accuracy of trust evaluation. If there are 40% dishonest votes, the relative error of vote value is \((2.94 - 0.2)/2.94 * 100\% = 37.3\%\) in Case (a), and \((0.2 + 0.6 + 2.94 + 0.4 - 0.2)/0.2 * 100\% = 548\%\) in Case (b). While the evaluation result based on the proposed evaluation...
method is about 0.86 in Case (a) with the relative error 4%; and is about 0.106 in Case (b) with the relative error 6%. This implies that the proposed LT evaluation method can effectively resist the bad mouthing attack.

We further evaluated the impact of $g(K)$, $T(i)$, and $\bar{T}(t \rightarrow j)$, on GT level. We assume the total number of registered pieces of UE in the system is 1000. We assigned maximum values to $T(i)$ and $\bar{T}(t \rightarrow j)$ and evaluated the influence of $g(K)$ on the GT level. Fig. 10 shows that the reliability of GT evaluation grows as the number of trust evaluation contributors increases. And it can reach about 0.106 in Case (b) with the relative error 6%. This implies that our method is about 0.86 in Case (a) with the relative error 4%; and is about 0.106 in Case (b) with the relative error 6%. This implies that the proposed LT evaluation method can effectively resist the bad mouthing attack.

We assume that the number of trust evaluation contributors is above 3 and $g(K)$ is close to 1. Fig. 11 shows the impact of $T(i)$ and $\bar{T}(t \rightarrow j)$ on GT evaluation. GT value changes linearly with $T(i)$ or $\bar{T}(t \rightarrow j)$, which is accord with our design.

Performance Test of Access Control

Now we implemented our access control scheme in C language using a Pairing-Based Cryptography (PBC) library (http://crypto.stanford.edu/pbc/) in a laptop running 64-bit Ubuntu Linux 16.04 with 2.5 GHz Inter Core i5 Quad-CPU and 8.0G RAM. We implement the scheme on the elliptic curve $y^2 = x^3 + x$ that has a high running speed on a 512-b finite field. In this circumstance, the pairing operation consumed 3.6 ms (ms) and the exponentiation operation costed 0.6 ms in group $G_T$ and 3.5 ms in group $G$.

First, we explored that the consuming time of generating public attribute keys with different maximum trust levels. It shows that the time changed linearly with the maximum trust levels in Fig. 12. It follows theoretic analysis.

Table 4 shows the time of generating secret attribute keys with different trust levels when the maximum trust level of GT or LT is 20. We discovered that trust levels have no impact on the secret attribute keys generation time. As there exists an exponentiation operation of $G$ in each of the two algorithms of generating secret keys, the consuming time of these two algorithms is coincident as well. By comparing the secret key generation time with the session key generation time (10 ms when the size of security parameter is 1024 bits) in [17], we can see that our scheme is more efficient than [17] regarding secure channel establishment. Because few schemes support D2D group communications, it is hard to perform other performance comparison.

The operation time of Encrypt is mainly up to the maximum trust level and the access policy. We evaluated this algorithm in two cases: (1) different maximum numbers of trust levels under a constant access policy; (2) different threshold levels of trust levels and the access policy. We evaluated this algorithm in [17]. We discover that trust levels have no impact on the secret attribute keys generation time. As there exists an exponentiation operation of $G$ in each of the two algorithms of generating secret keys, the consuming time of these two algorithms is coincident as well. By comparing the secret key generation time with the session key generation time (10 ms when the size of security parameter is 1024 bits) in [17], we can see that our scheme is more efficient than [17] regarding secure channel establishment. Because few schemes support D2D group communications, it is hard to perform other performance comparison.

The operation time of Encrypt is mainly up to the maximum trust level and the access policy. We evaluated this algorithm in two cases: (1) different maximum numbers of trust levels under a constant access policy; (2) different threshold levels of trust levels and the access policy. We evaluated this algorithm in two cases: (1) different maximum numbers of trust levels under a constant access policy; (2) different threshold levels of trust levels and the access policy. We evaluated this algorithm in two cases: (1) different maximum numbers of trust levels under a constant access policy; (2) different threshold levels of trust levels and the access policy.
6. Conclusions

In this paper, we presented a scheme to secure D2D communications based on two-dimensional trust levels by applying ABE. It can realize secure data communications among a number of mobile devices. Our scheme uses either a GT level issued by the core networks or a LT level evaluated by a device itself or both to control D2D communication data access. Only the devices holding the eligible trust level of GT and/or LT can access the data. Moreover, UE evaluates local trust levels with pseudonyms in order to enhance communication privacy. We proved our scheme as secure. Performance analysis showed that it can achieve high efficiency. The evaluation based on simulations further demonstrated that the theoretic analysis is correct and our scheme can achieve high efficiency, especially for data decryption.

Acknowledgments

This work is sponsored by the National Key Research and Development Program of China (grant 2016YFB0800704), the NSFC (grants 61672410 and U1536202), the Project Supported by Natural Science Basic Research Plan in Shaanxi Province of China (Program No. 2016ZDJC-06), the 111 project (grants B08038 and B16037), and Academy of Finland (grant 308087).

References
