Li, Ning; Yan, Zheng; Wang, Mingjun; Yang, Laurence T.

Securing Communication Data in Pervasive Social Networking Based on Trust with KP-ABE

Published in:
ACM Transactions on Cyber-Physical Systems

DOI:
10.1145/3145624

Published: 01/09/2018

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Li, N., Yan, Z., Wang, M., & Yang, L. T. (2018). Securing Communication Data in Pervasive Social Networking Based on Trust with KP-ABE. *ACM Transactions on Cyber-Physical Systems*, *3*(1), Article 9. https://doi.org/10.1145/3145624

Securing Communication Data in Pervasive Social Networking based on Trust with KP-ABE

NING LI, Xidian University
ZHENG YAN, Xidian University and Aalto University
MINGJUN WANG, Xidian University
LAURENCE T. YANG, St Francis Xavier University

Pervasive Social Networking (PSN) intends to support instant social activities in a pervasive way at anytime and anywhere. In order to protect crucial social activities, ensure communication dependability and enhance user privacy, securing pervasive social communications becomes especially important. However, neither centralized nor distributed solutions can protect PSN communications as expected. How to automatically control data access in a trustworthy and efficient way is an important security issue. In this paper, we propose a scheme to guarantee communication data security in PSN based on two dimensions of trust in a flexible manner on the basis of Key-Policy Attribute-based Encryption (KP-ABE). Its advantages and performance are justified and evaluated through extensive analysis on security, computation complexity, communication cost, scalability and flexibility, as well as scheme implementation. In addition, we develop a demo system based on Android mobile devices to test our scheme in practice. The results demonstrate its efficiency and effectiveness. Comparison with our previous work based on CP-ABE [Yan and Wang 2014] further shows its feasibility to be applied into PSN.

Categories and Subject Descriptors: **C.2.0 [General]** Security and Protection; **C.2.4 [Computer-Communication Networks]**: Distributed Systems.

General Terms: Design, Security, Algorithms, Performance

Additional Key Words and Phrases: access control, attribute-based encryption, reputation, social networking, trust

ACM Reference Format:

Ning Li, Zheng Yan, Mingjun Wang, and Laurence T. Yang. 2016. Secure Communication Data in Pervasive Social Networking based on Trust with KP-ABE. *ACM Trans. on Internet Technology*.

1. INTRODUCTION

Pervasive Social Networking (PSN) intends to support instant social activities in a pervasive way at anytime and anywhere. It makes use of heterogeneous networks self-organized by Mobile Ad Hoc Networks (MANETs), mobile cellular networks and the Internet as a supporting platform to offer social networking services. PSN is a typical cyber-physical social system. It has such specific characteristics as network carrier adaptability, social interaction ubiquity, and social service intelligence [Feng et al. 2017]. PSN is an essential complement to traditional online social networking. Through PSN, people can seek services from not only online services and remote friends, but also nearby people. Familiar strangers who do not know with each other but regularly meet in a same public place can chat, play, recommend and assist with each other by relying on PSN. PSN is very valuable for mobile users since it can

This work is sponsored by the National Key Research and Development Program of China (grant 2016YFB0800700), 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 B16037 and B08038), and Academy of Finland (grant 308087).

Author's addresses: N. Li, the State Key Laboratory on Integrated Services Networks, Xidian University, China, email: ningtaiye@yeah.net; Z. Yan (Corresponding Author), the State Key Laboratory on Integrated Services Networks, School of Cyber Engineering, Xidian University, China and the Department of Communications and Networking, Aalto University, Finland, email: zyan@xidian.edu.cn; M. Wang, the State Key Laboratory on Integrated Services Networks, Xidian University, China, email: wangmingjun1987@hotmail.com. L. T. Yang, St Francis Xavier University, Canada, email: ltyang@gmail.com.

extremely extend their social experiences. For example, PSN can be applied to provide instant recommendations, assistance, and urgent rescues among strangers in vicinity. It can also support instant resource sharing. This kind of services brings people from online communications to physical interaction.

In spite of the significant benefits, PSN faces unique security threats compared with traditional online social networking. First, users are mostly strangers, not only acquaintances in PSN. Therefore, in reciprocal activities, the users need to make sure that the communication partners are trustworthy. Trust plays a crucial role in PSN for reciprocal activities. It concerns the level of belief and dependence put into an entity based on direct or indirect knowledge and experiences on the entity. Trust can help people overcome uncertainty, avoid risk and encourage "trusted social behaviors". Thus, figuring out user trust becomes important to make a decision for secure and dependable social networking. Second, due to the heterogeneity of PSN, communication data in PSN are facing higher risk in terms of malicious eavesdropping and accessing by untrustworthy nodes compared with online social networking. It is crucial to secure communication data in PSN to carry out data sensitive communications. However, a centralized solution based on a trusted server may not be applicable in many scenarios (e.g., rescues, disasters, and military activities) although the server has sufficient processing capability. The drawback of this design is the server could become the object of DoS/DDoS attacks. The dependability and reliability of the whole system could be terribly impacted if the server is crashed. Obviously, setting up a backup server cannot solve the problem of connection availability and dependability. On the other hand, applying a pure distributed solution to secure PSN also suffers from some disadvantages. Due to the dynamic topology of PSN and the changes of user trust, the keys used for securing PSN could be frequently updated and distributed to eligible users. This causes heavy communication costs and extra processing loads, thus seriously influences PSN performance. How to efficiently and automatically secure communication data with security and dependability (i.e., trustworthiness) in PSN is a significant research issue.

Our previous work used Ciphertext-Policy Attributed-based Encryption (CP-ABE) to secure communication data in PSN based on trust attribute [Yan and Wang 2014]. Due to the complexity of message encryption in CP-ABE, it is not so efficient for PSN data transmission and protection, especially for mobile PSN nodes.

In this paper, we consider the scenes above and propose using two dimensions of trust: Global Trust (GT) and Local Trust (LT) to guarantee communication data security in PSN in a flexible manner on the basis of Key-Policy Attribute-based Encryption (KP-ABE). GT is evaluated by a Trusted Server (TS) based on historically accumulated social networking information; LT evaluated by PSN nodes based on locally collected information in social interactions. GT is more accurate than LT since it is evaluated based on globally collected information linked to a unique node identifier. It is used for PSN data access control when the TS is available. The TS generates public key *PK_GT* for encrypting messages and private key *SK_GT* for each node to decrypt ciphertext. If the connectivity of TS is not available, the LT will be applied to secure communication data in PSN. Each node generates local public key *PK_LT* and issues private key *SK_LT* to other nodes. In addition, we can use both GT and LT to secure communication data in PSN in a more secure way. In particular, the computation complexity of PSN message encryption is not changed with the maximum levels of GT and LT, which makes the scheme efficient and practical for communication data protection in PSN.

Different from previous work, our scheme adopts multi-dimensional trust levels to control communication data access in PSN in order to flexibly adapt to PSN topology changes and ensure PSN dependability. It is also very simple by comparing it with other ABE based access control solutions since it only applies one type of attribute: trust, to secure PSN communication data for achieving high efficiency and advanced dependability. Furthermore, applying KP-ABE is more suitable than CP-ABE in PSN in terms of the communication cost of encrypted data packages. Specifically, the contributions of this paper can be summarized as below:

- We motivate securing communication data in PSN based on general trust and/or local trust in a flexible manner on the basis of KP-ABE. PSN communication data access can be controlled based on trust in either a centralized or distributed way or both.
- We analyze the security of the proposed scheme, justify its performance through extensive analysis on its computation complexity, communication cost, scalability and flexibility. We further compare its performance with our previous work that used CP-ABE [Yan and Wang 2014] to show its advantages.
- We also develop a demo system based on Android mobile phones to show the practicability and real performance of the scheme.

We organize the rest of the paper as below. Section II briefly overviews related work. Section III introduces the system and threat models and our design goals, followed by the detailed description of our scheme in Section IV. Section V analyzes and evaluates the performance of the scheme. Finally, a conclusion and future work are presented in the last section.

2. RELATED WORK

2.1 Pervasive Social Networking

There are many research activities related to social networking and computing. A number of recent studies focus on social communications in mobile domains.

In academia, Social networking based on MANETs was investigated. Stanford MobiSocial Group developed Junction, which is a mobile ad hoc and multiparty platform for MANET applications [Junction]. A Micro-blog [Micro-blog] service helped users to post micro-blogs tagged by locations. ETHz Systems Group developed a pervasive social communication platform named AdSocial [AdSocial]. Ott et al. analyzed a floating content concept based on a theoretical framework to study the fundamental quantities of an ephemeral content sharing service in opportunistic networks [Ott et al. 2011; Hyytiä et al. 2011].

In industry, many companies conducted researches in the area of PSN. For example, Intel Berkeley Lab conducted a project named Familiar Stranger based on mobile devices. It could extend our feelings and relationships with strangers that we regularly observe but do not interact in public places [Familiar Stranger]. EZSetup system developed by Microsoft Research Asia made a mobile user find services provided by his/her neighbors [EZSetup]. Nokia Instant Community (NIC) provided an instant social networking platform to allow people in vicinity to communicate, get to know, and share information with each other [Nokia Instant Community]. But the above work in both academia and industry did not take social communication data security into considerations.

2.2 PSN Data Access Control

Data privacy protection and access control have been widely studied in the literature [Beach et al. 2009]. Common solution is data encryption [Chen and Faruq 2008]. The privacy issue of social networking was addressed in a number of applications through privacy setting configuration and execution of privacy policies defined by users. PeopleFinder allowed users to share their locations with others and refine privacy policies over time [Miluzzo et al. 2008]. In CenceMe [Miluzzo et al. 2008], a mobile social networking application using sensors, users can manually disable certain sensors in their mobile phones in order to preserve privacy.

However, existing work did not seriously consider trust, security and privacy in PSN. Traditional centralized social networking systems (e.g., Facebook) cannot satisfy instant social networking demands, especially when users do not have the Internet connection, but with location proximity. Most existing systems have not taken user privacy and data security into considerations. They lack management on trust for the purpose of security assurance and privacy enhancement. Thus, PSN is hard to be accepted by mobile users to perform official and crucial social communications that demand high security. In order to provide trustworthy pervasive social networking, we need to solve a number of issues with regard to trust, security and privacy [Yan 2013]. Most of previous work did not apply trust to control access to communication data in PSN although trust is the basis for establishing a secure social networking environment [Yan and Wang 2014].

Data access control with encryption only permits eligible users to decrypt data. The best way is to encrypt data once, distribute decryption keys once, and control each user to only decrypt its own authorized data. Considering the scenarios of PSN, the changes of network topology and user trust make key management complicated in order to achieve the expected level of security. Pure symmetric key encryption is not suitable because it is hard to support group social networking and manage keys in a distributed way. Data access control designed based on the symmetric key encryption to support various policies is neither flexible nor convenient. Public key encryption is not suitable for PSN either, especially in community based instant social activities. The reason is the public key encryption is not efficient for multicasting or broadcasting data to a group of users since data has to be encrypted by a data sender for each target receiver.

2.3 ABE based Data Protection in Mobile Networking

Attribute-based Encryption (ABE) [Goyal et al. 2006; Bethencourt et al. 2007; Müller et al. 2008; Sahai et al. 2005] is a new cryptographic technique for data protection. In ABE, data is encrypted based on an attribute-based access structure, such that only the users whose attributes satisfy the access structure can decrypt the data. Access control identifies users based on a set of attributes rather than an exact identity. There are two branches in ABE [Goyal et al. 2006]: KP-ABE [Goyal et al. 2006] and CP-ABE [Goyal et al. 2006; Bethencourt et al. 2007]. CP-ABE encrypts data according to an access control policy, formulated as a Boolean formula over the attributes. In KP-ABE, a secret key is generated by associating it with a set of attributes, thus it can decrypt the ciphertext if the attributes of the key owner satisfy a pre-applied access structure.

Recently, ABE has been applied into many areas, e.g., cloud data protection [Yu et al. 2010; Wang et al. 2011], secure vehicular ad hoc networks [Huang et al. 2009; Chen et al. 2010; Liu et al. 2016,], wireless sensor networks and IoT [Chatterjee et al. 2015; Wang et al. 2014].

However, ABE was seldom applied for securing PSN except our previous work that uses CP-ABE [Yan and Wang 2014]. The main reason is the complexity of an ABE based scheme, which makes its adoption by mobile devices challenging. The work presented in this paper focuses on securing communication data in PSN based on trust by applying KP-ABE. To date, our scheme is one of the scanty holistic access control schemes to secure communication data based on trust in PSN.

3. PROBLEM STATEMENT

3.1 System and Security Models

Fig. 1 illustrates the system model of PSN. The system contains two kinds of entities: a Trusted Server (TS) that is trusted to provide identity management, key management and trust management. TS can evaluate the GT of a node accurately by collecting sufficient information; the PSN nodes that interact with each other for pervasive social networking. Each node can collect local information in order to evaluate trust in other nodes, i.e., LT. It has functions and capability to manage LT, identities and keys of other nodes. It is important to secure communication data in PSN because integrity and confidentiality of some social communications should be stringently ensured.

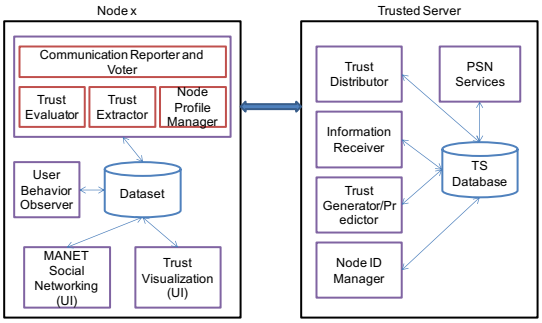


Fig. 1. System model of PSN

The nodes in PSN may not behave honestly, thus cannot be fully trusted with each other. Some malicious nodes may eavesdrop communication data to pursue personal benefits. Secure communications among trustworthy nodes in PSN are expected. Each node registers itself into TS, and then TS distributes a public key (*PK_GT*) to all nodes and issues a secret key based on the GT of each node. In addition, each node also has a public/secret key pair based on LT. The public key (*PK_LT*) of a node based on its current identifier is shared if needed for the purpose of authentication and secure communications. Since PSN nodes are mostly strangers, delegation is not allowed among them.

Additional assumptions based on our previous work about trust evaluation in PSN [Yan et al. 2013] include: TS knows each node's unique identifier and its pseudonyms. Each node reports its social networking records to TS according to pseudonyms. Thus, it is possible for TS to iteratively evaluate each node's general trust based on all collected information under a unique identifier. Social networking among nodes is performed based on pseudonyms in order to ensure node privacy. Each node can anonymously authenticate GT values of other nodes [Feng et al. 2017; Yan et al. 2016; Yan et al. 2013; Yan et al. 2014] and locally evaluate other node trust based on social networking experiences in an iterative way according to pseudonyms and GT. We further assume that each node has a unique identifier

registered at TS. The TS issues a number of certified pseudonyms to the node for it to selectively use during PSN communications. The pseudonyms issued to different nodes are different. Thus, the TS can match the node's pseudonyms to its unique identifier and make one node's pseudonyms different from those of another node.

3.2 Design Goals

To secure communication data in PSN, our design should achieve the following goals regarding security and performance: 1) Security: communication data in PSN can be only accessed by eligible nodes that are trustworthy enough; the data access is controlled according to the GT evaluated by TS or the LT evaluated locally by a node or both GT and LT; 2) Personalization: the scheme allows each node to define its own policies to control its data access, e.g., setting up trust level threshold or specifying a trust level for guiding personal data access control; 3) Lightweight: the scheme should be lightweight with regard to computation and communication overhead.

4. SCHEME

We introduce a scheme to secure communication data based on two dimensions of trust in PSN. First, we describe the notations, preliminaries and definitions related to the scheme. Then, we introduce the keys used in the scheme. Finally, we present the detailed scheme to control communication data access by applying KP-ABE in different scenarios.

4.1 Notations, Preliminaries and Definitions

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

- **Non-degenerate**: $e(g, g) \neq 1$ for generator g.
- **Bilinear**: for all $r, s \in G$, and $a, b \in Z_p$, $e(r^a, s^b) = e(r, s)^{ab}$.
- **Computable**: there is an efficient algorithm to compute $e(r, s)$ for any $r, s \in G$.

Definition 1: General Trust level (GT) is the trust level that is evaluated by the TS according to all accumulated social networking information based on a unique node identifier.

Definition 2: Local Trust level (LT) is the trust level that is evaluated by the PSN individual node according to the social networking information locally collected based on node pseudonyms.

The values of trust can be divided into discrete levels. We use gt_i represent the i-th level of GT, $i \in (0, I_{gt}]$, where I_{gt} is the maximum level of GT. Similarly, lt_i denotes the i-th level of LT, $i \in (0, I_{lt}]$, where I_{lt} is the maximum level of LT.

4.2 Keys and System Setup

In system setup, TS generates public key PK_GT that is shared within all nodes and used by the nodes to encrypt and decrypt symmetric key DEK_{gt}; a master key MK that is only known by TS. For each node u, TS generates its secret key $SK\_(gt\_u, u), i \in (0, I_{gt}]$ based on gt_u, where gt_u represents u's general trust level. Symmetric keys DEK_{gt} and DEK_{lt} are randomly selected by a node to encrypt communication data (i.e., PSN messages) controlled by GT and LT, respectively. When TS is unavailable and nodes want to build up a secure session in a distributed manner, each node u generates public key $PK\_(LT, u)$ and personalized secret

attribute key $SK\_(lt\_(u,u'),u,u'), lt\_(u,u') \in (0, I_{lt}]$, where $lt\_(u,u')$ represents the local trust level of u' evaluated by u, and $SK\_(lt\_(u,u'),u,u')$ is a secret attribute key of attribute LT issued to eligible node u' by node u. Then, u issues $SK\_(lt\_(u,u'),u,u')$ to node u'. The secret key $SK\_(lt\_(u,u'),u,u')$ is used in the decryption operation related to $PK\_(LT,u)$.

When TS is available, either GT or LT or both could be used for PSN data access control. Each node owns a pair of keys for controlling data access based on GT: PK_GT for encryption and decryption operations and $SK\_(gt\_u,u)$ used to decrypt DEK_{gt}. Corresponding to GT, DEK_{gt} is used in symmetric decryption to get plain communication data. For attribute LT related to node u, public key $PK\_(LT,u)$ is generated by node u and used to encrypt DEK_{lt}, aiming to control data access based on the local trust level evaluated by u. The corresponding secret attribute keys for decrypting the cipher-key encrypted by $PK\_(LT,u)$ are personalized for eligible nodes and issued by node u. To prevent collusion, every node u' gets different secret attribute key $SK\_(lt\_(u,u'),u,u')$ that only it can use. In case that both GT and LT are applied to control data access in PSN, the above keys are used in an integrated way, refer to Section 4.3 for details. The keys $SK\_(gt\_u,u)$ and $PK\_(LT,u)$ of node u are bound to its unique identifier, which can be a pseudonym. This binding is crucial for the verification of node trust level. Similarly, $SK\_(lt\_(u,u'),u,u')$ is bound to the unique identifiers of node u and u'. Table I summarizes the keys used in the proposed scheme.

Table I. System Keys

Key	Description	Usage
PK_GT	The public key of attribute GT generated by TS;	Encryption and decryption of DEK_{gt} and personalized secret attribute key generation for node u; shared within all nodes;
MK	The master key of TS about GT, which is only known by TS;	For key generation;
DEK_{gt}	The randomly selected symmetric key, whose access is controlled by GT;	Encryption of PSN communication data;
$SK\_(gt\_u, u)$	The secret key of node u regarding GT;	Decryption to get DEK_{gt};
$PK\_(LT,u)$	The public key of LT generated by u;	Encryption of symmetric key DEK_{lt};
$MK(u)$	The master key of node u about LT, which is only known by node u;	Key generation at node u;
$SK\_(lt\_(u,u'),u,u')$	The secret key of LT for node u' issued by u;	Decryption of DEK_{lt} of node u;
DEK_{lt}	The randomly selected symmetric key, whose access is controlled by LT.	Encryption of PSN communication data.

4.3 Scheme Algorithms

The proposed scheme contains a number of fundamental algorithms: *IssueGeneralTrustPK*, *IssueGeneralTrustSK*, *CreateLocalTrustPK*, *IssueLocalTrustSK*, *Encrypt* and *Decrypt*.

IssueGeneralTrustPK(). This algorithm chooses a bilinear multiplicative group G_1 of prime order p, selects g as the generator of G_1, and generates a bilinear map $e: G_1 \times G_1 \rightarrow G_2$. A security parameter k determines the size of the groups. It defines universe of attributes $U = (gt\_1, gt\_2, \ldots, gt\_i, \ldots, gt\_I_{gt})$, where gt_i represents the i-th level of GT, $i \in (0, I_{gt}]$ and I_{gt} is the maximum level of GT. For each gt_i, the

algorithm uniformly randomly selects a number denoted t_{gt_i} from Z_p and uniformly chooses y at random in Z_p. Then, PK_GT and MK are generated as below:

$PK\_GT = \left\{G_1, G_2, e, g, Y = e(g,g)^y, T_1 = g^{t_{gt\_1}}, \dots, T_{I_{gt}} = g^{t_{gt\_I_{gt}}}\right\}$,

$MK = \{y, t_{gt\_1}, \dots, t_{gt\_I_{gt}}\}$.

IssueGeneralTrustSK(PK_GT, u). This algorithm is automatically executed by TS whenever it is time to issue a secret GT key. Based on node u's GT, for instance, gt_u, it sets the access policy of node u: u can decrypt the ciphertexts encrypted by gt_i where $gt\_i \le gt\_u$. It issues an access control tree T with root r. For r, it further sets a threshold value to be 1, and chooses polynomial q_r with degree $d_r = 0$, sets $q_r(0) = y$. Each leaf in tree T contains attribute t_{gt_i}, where $gt\_i \le gt\_u$. The algorithm generates $D_i = g^{\frac{q_r(0)}{t_{gt\_i}}}$ for leaf i with attribute t_{gt_i}. The secret key is:

$SK\_(gt\_u, u) = \left\{D_1 = g^{\frac{q_r(0)}{t_{gt\_1}}}, \dots, D_i = g^{\frac{q_r(0)}{t_{gt\_i}}}, \dots, D_{N_{gt}} = g^{\frac{q_r(0)}{t_{gt\_u}}}\right\}$.

CreateLocalTrustPK(LT, u). This algorithm is executed by node u to control the access of its data (e.g., PSN chatting messages) in a distributed manner. The algorithm checks the LT related policies. If this is the case, the algorithm outputs a public attribute key for the LT of user u, denoted $PK\_(LT,u)$, otherwise NULL. Similar to *IssueGeneralTrustPK()*, the algorithm chooses a bilinear multiplicative group G_1' of prime order p', and selects g_1 as a generator of G_1'. In addition, it selects a bilinear map $e_1 : G_1' \times G_1' \to G_2'$. A security parameter k' determines the size of the groups. We define the universe of LT attributes $U' = (lt\_1, lt\_2, \dots, lt\_i, \dots, lt\_I_{lt})$, lt_i represents the i-th level of LT, $i \in (0, I_{lt}]$, where I_{lt} is the maximum level of LT. For each lt_i, the algorithm uniformly randomly selects number t_{lt_i} from $Z_{p'}$ and uniformly chooses $y' \in Z_{p'}$ at random. Then, we have:

$PK\_(LT,u) = \begin{Bmatrix} G_1', G_2', e_1, g_1, Y' = e_1(g_1, g_1)^{y'}, \\ T_1' = g_1^{t_{lt\_1}}, \dots, T_{I_{gt}} = g_1^{t_{lt\_I_{lt}}} \end{Bmatrix}$, $MK(u) = \{y', t_{lt\_1}, \dots, t_{lt\_I_{lt}}\}$.

IssueLocalTrustSK(PK_(LT,u),u'). This algorithm is executed by node u. It outputs the secret key of node u', $SK\_(lt\_(u,u'), u, u')$. This process is executed by node u based on trust evaluation on u'. Similarly, an access control tree T' is issued with root r'. For r', the algorithm sets a threshold value to be 1, chooses polynomial $q_{r'}$ with degree $d_{r'} = 0$, and sets $q_{r'}(0) = y'$. Each leaf in tree T' has attribute t_{lt_i}. $D_i' = g_1^{t_{lt\_i}}$ is generated for leaf i with attribute t_{lt_i}. Then, $SK\_(lt\_(u,u'), u, u')$ can be generated as:

$$SK\_(lt\_(u,u'), u, u') = \left\{D_1' = g_1^{\frac{q_{r'}(0)}{t_{lt\_1}}}, \dots, D_i' = g_1^{\frac{q_{r'}(0)}{t_{lt\_i}}}, \dots, D_{N_{lt}}' = g_1^{\frac{q_{r'}(0)}{t_{lt\_(u,u')}}}\right\}.$$

The generated key is sent to u' by u through a secure channel, e.g., using RSA.

For the data access controlled by GT, the Encrypt and Decrypt algorithms are described as below.

Encrypt1(PK_GT, gt_i, DEK_{gt}, M). The algorithm takes as input PK_GT, gt_i, DEK_{gt} and message M. It encrypts DEK_{gt} with attribute gt_i and encrypts M with symmetric key DEK_{gt}. It selects a random value $s \in Z_p$ and generates ciphertext CT_{gt} as:

$CT_{gt} = \{gt\_i, C_{gt}' = DEK_{gt} \times Y^s, C_{gt\_i} = g^{s \times t_{gt\_i}}, C_{gt}'' = (M)_{DEK_{gt}}\}$.

Decrypt1(PK_GT, $SK\_(gt\_u, u)$, CT_{gt}). The algorithm takes as input PK_GT, $SK\_(gt\_u, u)$, and encrypted message CT_{gt}. It decrypts CT_{gt} with $SK\_(gt\_u, u)$ and outputs plaintext M. This process is conducted at node u as follows.

$Decrypt1(CT_{gt}, PK\_GT, SK\_(gt\_u, u), u) = e(D_i, C_{gt\_i}) = e\left(g^{\frac{q_r(0)}{t_{gt\_i}}}, g^{s \times t_{gt\_i}}\right) =$

$e(g,g)^{s \times q_r(0)} = e(g,g)^{y \times s} = Y^s$,

$DEK_{gt} = \frac{C'_{gt}}{Y^s}$.

Then, the algorithm decrypts C''_{gt} with symmetric key DEK_{gt} and gets plaintext M.

For PSN data access controlled by LT, the Encrypt and Decrypt algorithms are described as below.

Encrypt2($PK\_(LT,u)$, lt_i, DEK_{lt}, M*).* This algorithm takes as input $PK\_(LT,u)$, lt_i, DEK_{lt} and message M. The algorithm encrypts DEK_{lt} with $PK\_(LT,u)$ and M with DEK_{lt}. It selects random value $s' \in Z_p$, and generates ciphertext CT_{lt} as:

$CT_{lt} = \{lt\_i, C'_{lt} = DEK_{lt} \times Y'^{s'}, C_{lt\_i} = g_1^{s' \times t_{lt\_i}}, C''_{lt} = (M)_{DEK_{lt}}\}$.

Decrypt2($PK\_(LT,u)$, $SK\_(lt\_(u,u'),u,u')$, CT_{lt}*).* This algorithm takes as input $PK\_(LT,u)$, $SK\_(lt\_(u,u'),u,u')$, and encrypted message CT_{lt}. It decrypts CT_{lt} with $SK\_(lt\_(u,u'),u,u')$ and outputs plaintext M at node u as below.

$Decrypt2(PK\_(LT,u), SK\_(lt\_(u,u'),u,u'), CT_{lt}) = e(D'_i, C_{lt\_i}) =$

$e\left(g_1^{\frac{q_{r'}(0)}{t_{lt\_i}}}, g_1^{s' \times t_{lt\_i}}\right) = e(g_1,g_1)^{s' \times q_{r'}(0)} = e(g_1,g_1)^{s' \times y'} = Y'^{s'}$,

$DEK_{lt} = \frac{C'_{lt}}{Y'^{s'}}$.

Then, we decrypt C''_{lt} with symmetric key DEK_{lt} and get plaintext M.

For PSN communication data access controlled by both GT and LT, the Encrypt and Decrypt algorithms are performed as follows.

Encrypt3(PK_GT, $PK\_(LT,u)$, gt_i, DEK_{gt}, lt_i, DEK_{lt}, M*).* This algorithm takes as input PK_GT and $PK\_(LT,u)$, gt_i, lt_i, DEK_{gt}, DEK_{lt}, and M. M is encrypted by DEK_{gt} and then, DEK_{gt} is encrypted with gt_i using PK_GT to output ciphertext CT_1. Then CT_1 is encrypted with DEK_{lt} that is encrypted with lt_i using $PK\_(LT,u)$ to output CT_2, as shown below.

$CT_1 = \{gt\_i, C' = DEK_{gt} \times Y^s, C_{gt\_i} = g^{s \times t_{gt\_i}}\}$,

$CT_2 = \{lt\_i, C'' = DEK_{lt} \times (Y')^{s'}, C_{lt\_i} = g_1^{s' \times t_{lt\_i}}, (CT_1)_{DEK_{lt}}, (M)_{DEK_{gt}}\}$.

Decrypt3(PK_GT, $PK\_(LT,u)$, $SK\_(gt\_u,u)$, $SK\_(lt\_(u,u'),u,u')$, CT_2*).* This algorithm is executed by the node to decrypt the PSN communication data controlled by both GT and LT. It firstly calls *Decrypt2* to decrypt DEK_{lt}, then uses DEK_{lt} to get CT_1. Next, it calls *Decrypt1(PK_GT,* $SK\_(gt\_u,u)$, CT_1*)* to decrypt DEK_{gt}, and finally uses DEK_{gt} to get plaintext M.

4.4 Procedures

We illustrate the procedure of securing communication data in PSN based on trust in three cases: 1) controlling communication data access with GT when TS is available; 2) controlling communication data access with LT in a distributed way when TS is not available; 3) controlling communication data access with both GT and LT with the support of TS.

Case1: Secure PSN Communications with GT

In this case, node u wants to secure its communication data with other nodes based on GT. It generates DEK_{gt} and uses PK_GT and $SK\_(gt\_u,u)$ to encrypt and decrypt DEK_{gt}, respectively. PK_GT and $SK\_(gt\_u,u)$ are generated by TS based on the general trust level of node u. Only the node with $gt\_u \geq gt\_i$ can decrypt DEK_{gt} to

get the plain PSN data. Herein, gt_i is a threshold defined in access policy about GT for access control. Fig. 2 shows the detailed procedure.

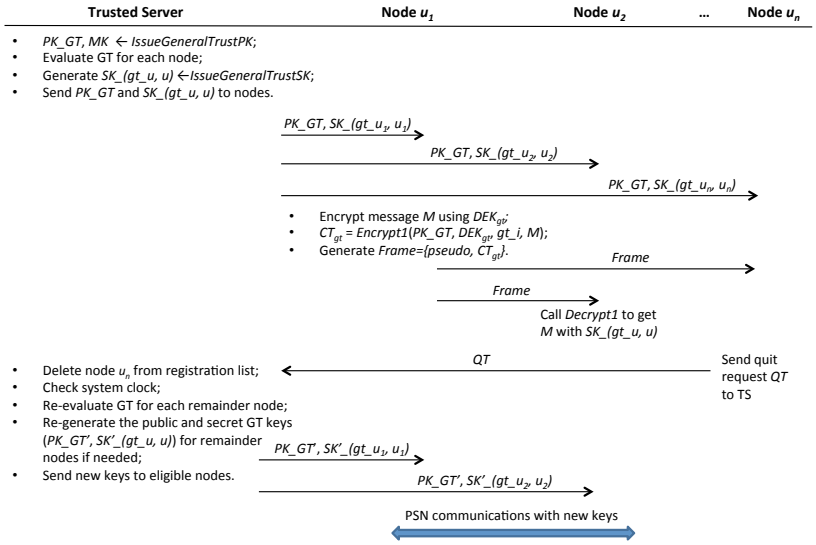


Fig. 2. Control PSN data access with GT

GT Related Key Generation:

- TS calls *IssueGeneralTrustPK* to generate PK_GT and MK. PK_GT is shared within all nodes, MK is kept by TS only. When a new node joins the system. TS assigns it a unique identifier u, sends PK_GT to it and initiates its GT value.
- TS evaluates the GT of a node using the method proposed in [Yan et al. 2013; Yan et al. 2014].
- TS calls *IssueGeneralTrustSK* to generate $SK\_(gt\_u, u)$ for eligible node u based on its GT and TS sends $SK\_(gt\_u, u)$ to node u.
- TS stores the node unique identifier u, the current pseudonym applied by u, gt_u and $SK\_(gt\_u, u)$ in its database by attaching a valid period to them.

Secure PSN Communications: node u uses DEK_{gt} to encrypt communication data M. It calls *Encrypt1* to encrypt DEK_{gt} with PK_GT and generates a message frame with a node pseudonym, $Frame = \{pseudo, CT\}$. It then sends the frame to other nodes in PSN. Once received the frame from a source node, a receiver node calls *Decrypt1* to decrypt CT. Only those nodes with $gt\_u \geq gt\_i$ can decrypt DEK_{gt} and get M.

Quit from PSN: A node may quit from the PSN at any time by informing TS. The TS accepts this request and informs other nodes (e.g., by not issuing its GT token for node authentication [Yan et al. 2016]). Later on, TS will not send any new keys to it when its old keys are expired.

Re-issue GT Keys: Due to node revocation and trust changes, related keys should be updated in order to ensure security. In this case, TS periodically updates PK_GT and $SK\_(gt\_u, u)$ as below:

- TS checks the validity period of PK_GT;
- If it runs out, TS re-evaluates the GT levels of remainder nodes;
- TS re-generates PK_GT and MK, and calls *IssueGeneralTrustSK* to issue a new secret key for each eligible node. Then it resets the validity period of new keys;
- TS sends the new keys and valid pseudonyms to the eligible nodes. The new keys are applied once their validity period starts.

Case 2: Secure PSN communications with LT

In this case, a node wants to secure its communication data based on LT. It generates DEK_{lt} and uses corresponding public key $PK\_(LT, u)$ to encrypt DEK_{lt}. For eligible node u', u generates $SK\_(lt\_(u,u'),u,u')$ based on the trust level of u' evaluated by u. $SK\_(lt\_(u,u'),u,u')$ is used to decrypt DEK_{lt}. Only the node with $lt\_(u,u') \geq lt\_i$, (lt_i is the threshold defined by access policy regarding LT) can decrypt DEK_{lt} and then get the plain PSN data, as shown in Fig. 3. The detailed procedure is similar to Case 1 except that the node itself handles $PK\_(LT,u)$ and $SK\_(lt\_(u,u'),u,u')$ generation, issuing and revocation based on LT by calling *CreateLocalTrustPK* and *IssueLocalTrustSK*. *Encrypt2* and *Decrypt2* are applied in this case.

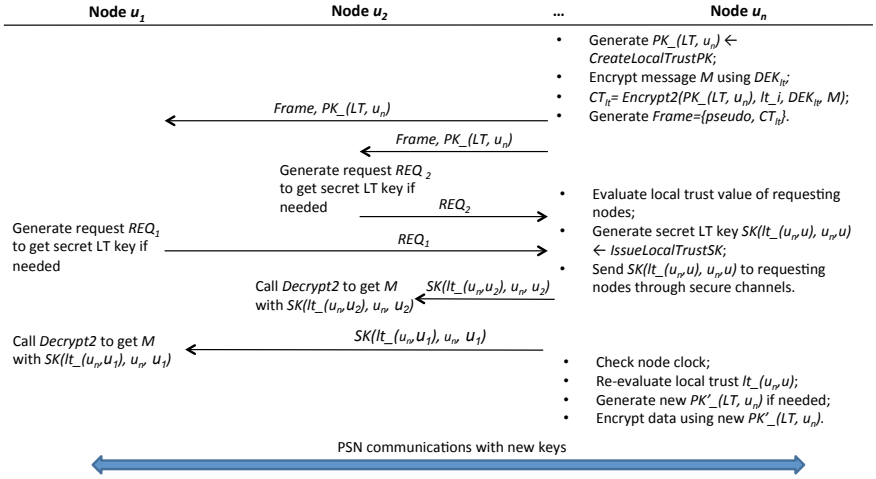


Fig. 3. Control PSN data access with LT

Case 3: Secure PSN Communications with both GT and LT:

In this case, a node wants to secure its communication data based on both GT and LT. Node u generates DEK_{gt} and uses PK_GT and $SK\_(gt\_u',u')$ to encrypt and decrypt DEK_{gt}, respectively. Meanwhile, u also generates DEK_{lt} and uses $PK\_(LT,u)$ to encrypt DEK_{lt}. Eligible node u' gets $SK\_(lt\_(u,u'),u,u')$ generated by u based on the LT of u' evaluated by node u. The PSN data is encrypted by both GT and LT. Only the node with $gt\_u' \geq gt\_i$ and $lt\_(u,u') \geq lt\_i$ can decrypt DEK_{gt} and DEK_{lt} and then get the plain PSN data, as shown in Fig. 4 for details.

The first step is the same as GT Related Key Generation as described in Case 1.

LT Related Key Generation: The operation is executed at the node that wants to control PSN data access based on both GT and LT as below.

- Evaluate the LT of other nodes using the method proposed in [Yan et al. 2013; Yan et al. 2014].
- Call *CreateLocalTrustPK* to create $PK\_(LT,u)$ and *IssueLocalTrustSK* to generate $SK\_(lt\_(u,u'),u,u')$ for eligible node u' based on its LT. Then, node u sends $SK\_(lt\_(u,u'),u,u')$ to node u'.
- Store the information about u', e.g., current pseudonym, $lt\_(u,u')$, $SK\_(lt\_(u,u'),u,u')$ in a light database by linking to a validity time period.

Secure PSN Communications: node u generates DEK_{gt} and DEK_{lt}. It encrypts communication data M by calling *Encrypt3*. Once receiving a data frame from a source node, a receiver node calls *Decrypt3* to get plain communication data M.

Quit from PSN: A node may quit from the PSN at any time by informing TS. The TS accepts its request and informs other nodes. Later on, TS will not send any new keys to it when its old keys are expired.

Re-issue GT Keys: this step is performed in the same way as described in Case 1.

Re-issue LT keys: node u periodically checks the clock, updates LT through local trust re-evaluation, and re-generates $PK\_(LT,u)$ and $SK\_(lt\_(u,u'),u,u')$.

The new keys are applied once their validity period starts.

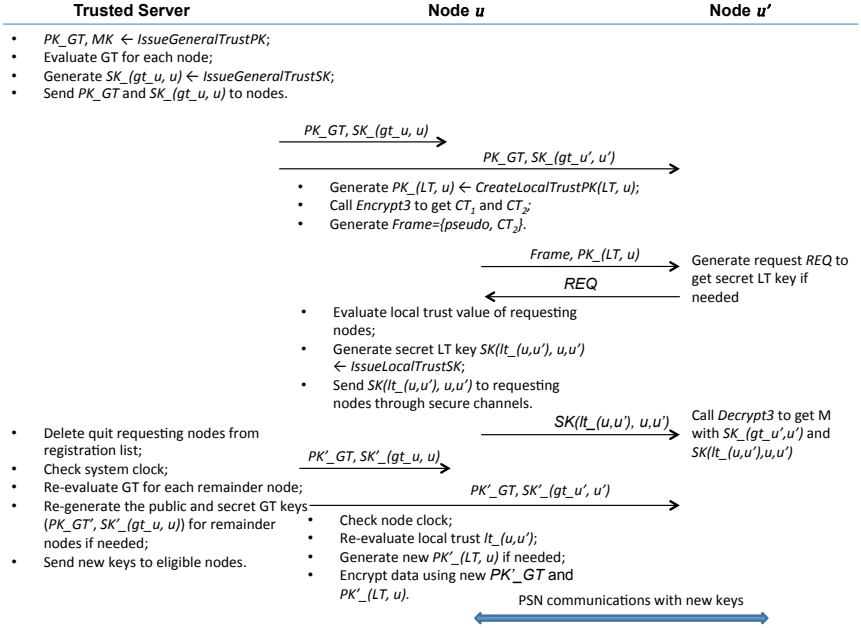


Fig. 4. Control PSN data access with both GT and LT

5. PERFORMANCE EVALUATION & SYSTEM IMPLEMENTATION

5.1 Security Analysis

Fine-Grainedness of Access Control

The scheme can achieve fine-grained access control. TS and nodes are able to define access policies and generate corresponding secret keys to control access to the encrypted data. Various factors that impact trust can be considered in the process of trust evaluation. The access policies can be flexibly defined based on GT and/or LT by TS and PSN nodes. With this way, the scheme not only guarantees the fine-grained access control, but also simplifies the attribute structure of the secret key by only considering the trust levels. Meanwhile, a data sender can set the data access privilege based on GT or LT or both according to the security demand of the data. Specifically, trust can be configured into different levels in different social networking contexts in order to realize context-aware data access control in PSN. For example, in the case that a person would like to send a message to his family that is not readable by college friends and vice versa a message to college friends which is not readable by family, our scheme can support this kind of practical demand by setting different community groups with different trust levels in different PSN communication situations.

Data Confidentiality

In the proposed scheme, data confidentiality is ensured by three data encryption algorithms, i.e., *Encrypt1*, *Encrypt2* and *Encrypt3*. In all these algorithms, the PSN

data firstly are encrypted using a symmetric key DEK_s, and then DEK_s is encrypted using our variant of KP-ABE algorithm. Assuming that the symmetric key algorithm is secure, e.g., using a standard algorithm such as Advanced Encryption Standard (AES), the data confidentiality of our proposed scheme merely relies on the security of our variant of KP-ABE algorithm. In *Encrypt1* and *Encrypt2*, two KP-ABE algorithms constructed on independent group structures are used respectively, so the data Confidentiality of *Encrypt1* and *Encrypt2* relies on their relevant KP-ABE algorithms. The KP-ABE algorithms used are the same as the algorihtm in [Goyal et al. 2006], so the security of our scheme only based on *Encrypt1* and *Encrypt2* is provably secure under the attribute-based Selective-Set model given the Decisional Bilinear Diffie-Hellman (DBDH) problem is hard.

In Encrypt3, we combine two KP-ABE algorithms together in order to achieve data access control based both GT and LT. In what follows, we prove that Encrypt3 is as secure as *Encrypt1* and *Encrypt2*, i.e., KP-ABE algorithm in [Goyal et al. 2006]. We sketch the security proof of our scheme based on *Encrypt3* by definin three games as follow:

Game 1: This is the security game of our scheme based on *Encrypt1*.

Game 2: This is the security game of our scheme based on *Encrypt2*.

Game 3: This game is our scheme based on *Encrypt3*.

Proof: If there is a polynomial time adversary $\mathcal{A}$ that wins the semantic security game of Game 3 under the attribute-based Selective-Set model with non-negligible advantage ϵ, we can use it to build a polynomial time simulator $\mathcal{B}$ to win the semantic security game of Game 1 and to build a polynomial time simulator $\mathcal{C}$ to win Game 2. In the semantic security game, the adversary submits two equal length challenge message M_0 and M_1. The challnger flips a random coin $b \in \{0,1\}$ and encrypts M_b. The challenge cihpertext is then given to the adversary. The adversary is asked to output his guess $b' \in \{0,1\}$ of the random coin b. If $b^{'} = b$ the adversary wins. During the game, the adversary is given public parameters and allowed to issue queries for private keys for many access policies except the declared access policy the adversary wishes to be challenged upon. Assuming the adversary $\mathcal{A}$ is able to win the semantic security Game 3 with non-negligible advantage, i.e., $Adv_{\mathcal{A}} = |Pr\left[b^{'} \leftarrow \mathcal{A}(CT_1, CT_2, PK\_GT, PK\_(LT,u), SK_j) = b\right]$ -0.5|, where $\mathrm{SK_j}$ denotes the set of all the secret key queries. We build a polynomial time simulator $\mathcal{B}(CT_{gt}, PK\_GT, SK_j^{'})$ to challenge Game 1 and a polynomial time simulator $\mathcal{C}(CT_{lt}, PK\_(LT,u), SK_j^{''})$ to challenge Game 2. In Game 1, simulator $\mathcal{B}$ gets ciphertext CT_{gt}, which contains more ciphertext contents than CT_1 in Game 3, so simulator $\mathcal{B}$'s advantage is higher than ϵ. In Game 2, simulator $\mathcal{C}$ gets CT_{lt}, which is the same as CT_2 except content $(M)_{DEK_{gt}}$. Since simulator $\mathcal{C}$ has no knowledge of DEK_{gt}, its advantage in Game 2 will not be affected by $(M)_{DEK_{gt}}$. Therefore, the advantage of simulator $\mathcal{C}$ in Game 2 is the same as adversary $\mathcal{A}$. Game 1 and Game 2 both base on KP-ABE and are provably secure under the attribute-based Selective-Set model, thus Game 3 is secure under the same model.

5.2 Performance Analysis and Discussions

Computation Complexity

We analyze the computation complexity of the following fundamental algorithms: *IssueGeneralTrustPK*, *IssueGeneralTrustSK*, *CreateLocalTrustPK*, *IssueLocalTrustSK*, *Encrypt* and *Decrypt*.

For the algorithms *IssueGeneralTrustPK* and *CreateLocalTrustPK*, their computation complexity is related to the input maximum level of GT or LT. These two algorithms execute one random generation operation (Rand) and one exponentiation operation (Exp) on group G_1 or G_1' for each GT or LT level (the level is linear from 1 to the maximum level). Moreover, extra one random generation operation and one exponentiation are carried out to generate Y and Y'. So, the computation complexity of the two algorithms is $\mathcal{O}(I_{gt})$ and $\mathcal{O}(I_{lt})$, respectively.

For the algorithms *IssueGeneralTrustSK* and *IssueLocalTrustSK*, their computation complexity is related to the GT level or the LT level of the node. Supposed that the GT level of a node is N_{gt}, $N_{gt} \in (0, I_{gt}]$, N_{gt} exponentiation operations are executed at the node. The computation complexity of the algorithm *IssueGeneralTrustSK* is $\mathcal{O}(N_{gt})$. This is because the access tree contains a root and a number of N_{gt} leaves. The algorithm generates $D_i = g^{\frac{q_r(0)}{t_{gt\_i}}}$ for each leaf i with attribute t_{gt_i}. The computation cost for each leaf is constant. Similarly, the computation complexity of the algorithm *IssueLocalTrustSK* is $\mathcal{O}(N_{lt})$, where N_{lt} represents the LT level of a node.

In Case 1, the main computation overhead of encryption operation is the encryption of the message using DEK_{gt} as well as the encryption of DEK_{gt} using the *Encrypt1* algorithm. The complexity of the former depends on the size of the underlying message and is inevitable for any cryptographic methods. The algorithm *Encrypt1* requires one random generation operation and two exponentiation operations on group G_1, since we only use one attribute gt_i to encrypt DEK_{gt}. So, the computation complexity of the encryption operation is $\mathcal{O}(1)$.

For decryption operation, the main computation is the decryption of DEK_{gt} using the *Decrypt1* algorithm and the decryption of the message using DEK_{gt}. The complexity of the later depends on the size of the underlying message and is inevitable for any cryptographic methods. For the former, the computation contains two parts: traversing the access tree and one pairing operation (Pair). Thus, its computation complexity is $\mathcal{O}(1)$.

In Case 2, the encryption and decryption algorithms (*Encrypt2* and *Decrypt2*) are similar to the relative algorithms in the case based on GT. The computation complexity of the encryption operation and decryption operation is both $\mathcal{O}(1)$.

In Case 3, the main computation overhead of *Encrypt3* consists of four parts: twice symmetric encryption with DEK_{gt} and DEK_{lt}, twice attribute encryption with gt_i and lt_i. Since the size of CT_1 is very small, about 270 Bytes, the computation complexity of generating CT_2 mainly depends on the size of the PSN data. For each attribute based encryption operation, it requires one random generation operation and two exponentiation operations on group. Thus, the computation complexity is $\mathcal{O}(1)$.

In the *Decrypt3* algorithm, firstly DEK_{lt} is derypted with $SK\_(lt\_(u, u_1)$, then CT_1 is gained with DEK_{lt}. Further, DEK_{gt} is decrypted with $SK\_(gt\_u, u)$, and then the PSN communication data M is achieved with DEK_{gt}. Since the size of CT_1 is small, about 270 Bytes, the complexity of the symmetric decryption mainly depends on the size of PSN data, which is inevitable. For each attribute based decryption operation, the computation contains two parts: traversing the access tree and one pairing operation on group. Therefore, its computation complexity is $\mathcal{O}(1)$.

Table II summarizes the computation complexity of each system operation in the proposed scheme and compares with our previous work [Yan and Wang 2014] based on CP-ABE.

Table II. Comparison of Computation Complexity

Algorithms	Operations	Our scheme	Scheme with CP-ABE [Yan and Wang 2014]
IssueGeneralTrustPK	$(I_{gt}+1)*Rand+(I_{gt}+1)*Exp$	$\mathcal{O}(I_{gt})$	$\mathcal{O}(I_{gt})$
IssueGeneralTrustSK	$N_{gt}*Exp$	$\mathcal{O}(N_{gt})$	$\mathcal{O}(1)$
IssueLocalTrustPK	$(I_{lt}+1)*Rand+(I_{lt}+1)*Exp$	$\mathcal{O}(I_{lt})$	$\mathcal{O}(I_{lt})$
IssueLocalTrustSK	$N_{lt}*Exp$	$\mathcal{O}(N_{lt})$	$\mathcal{O}(1)$
Encryption	$1*Rand+2*Exp$	$\mathcal{O}(1)$	$\mathcal{O}(L)$
Decryption	$1*Pair$	$\mathcal{O}(1)$	$\mathcal{O}(1)$

Note: I_{gt}: the maximum level of GT; N_{gt}: the GT level of a node; I_{lt}: the maximum level of LT; N_{lt}: the LT level of a node; L: the number of conjunctions related to access policy, $L \le I_{gt} \times I_{lt}$.

Obviously, our scheme based on KP-ABE mainly costs in secret attribute key generation, which is related to the levels of node's GT and LT. The scheme based on CP-ABE mainly costs in encryption operation. We note that the secret key generation is mainly executed by a server with a strong processing capacity or by mobile PSN node devices, which does not performed frequently, while the PSN data encryption needs to be performed very frequently. Thus, the scheme proposed in this paper can save a lot of computation costs compared with the work in [Yan and Wang 2014] although the generation of secret attribute key takes longer time. Thus, the proposed scheme is more suitable to be applied in many practical applications of PSN, e.g., social chatting and service inquiry. In these cases, the generation of secret attribute key is not frequent, especially when a session of social networking has been built up.

Communication Cost

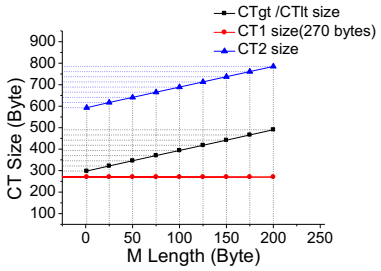


Fig. 5. CT size

The size of ciphertext is an essential aspect with regard to communication cost. In the proposed scheme, the structure of the ciphertext is simple and the size is also reasonable, as shown in Fig. 5. CT can be either CT_{gt}, CT_{lt}, or $CT_1 \parallel CT_2$. If the size of symmetric keys is 128 bits and the length of PSN data is one byte, the size of ciphertext CT_{gt} is about 298 bytes, as shown in Fig. 8. CT_{gt} is lineally increased with the PSN data size. For the data access control based on LT, the size of ciphertext CT_{lt} is the same as CT_{gt}. While for the data access control based on both GT and LT, the ciphertext CT_1 keeps 270 bytes and CT_2 consists of three parts: the result of ABE, the symmetric encryption result of CT_1, the symmetric encryption result of PSN communication data M. The length of CT_2 is about 593 bytes if PSN data size is one byte. And CT_2 is also lineally increased with the PSN data size.

The communication cost of our scheme is small and acceptable. Additionally, we apply the access policy and trust evaluation to assist the decisions on the need of communications, key generation and exchanges in order to minimize communication costs in various situations.

Scalability

In the proposed scheme, we concern multiple trust levels in the data access control. The result is the computation complexity of encryption and decryption can be greatly reduced. The number of nodes in PSN does not impact communication data encryption and decryption, thus the scheme supports scalability from the view of the size of PSN.

Our scheme is scalable for securing communication data to satisfy various access control demands in PSN with trust management support. We integrate trust evaluation into fine-grained access control in order to reduce the complexity of cryptographic computation. In the process of trust evaluation, we can take into account many complicated attributes that should be considered in the access control policy. Thus, our scheme can achieve sound operation performance and therefore suitable to be applied into PSN.

The goal of scalability can be further supported because the complexity of each operation in the scheme does not depend on the number of nodes in the system, shown in Table II. The computation complexity of symmetric key encryption and decryption is $O(1)$. Therefore, our scheme is ideal for securing communication data in PSN.

Flexibility

The proposed scheme can flexibly control communication data access in PSN in a centralized or distributed manner by adapting to the topology of PSN. The scheme supports data access control based on centralized trust evaluation or distributed trust evaluation or both in PSN. If TS is available, we could use GT or both GT and LT to control data access. If not, LT can be applied for securing PSN.

Personalized Access Control

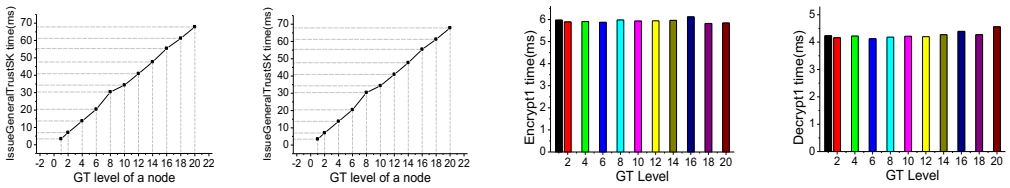
The scheme provides data access control in PSN with trust management support. It supports personal access control policies defined by each individual node. Each node can set its own data access policy (use different trust levels to control PSN data in different social networking contexts). TS and the node itself manage the corresponding keys. The data access policy plays an important role in the design of general trust evaluation and local trust evaluation.

5.3 Demo Implementation and Performance Evaluation

We implemented the proposed scheme with C language based on Pairing Based Cryptography (PBC) Library [Pairing Based Cryptography] on a workstation with Intel Pentium CPU G630 and 2-GB RAM, running Ubuntu 14.04. Furthermore, we transplanted the core code of the scheme into Android mobile phones through cross-compiler. Furthermore, we developed a demo system to demonstrate the applicability of the scheme. We adopted Android mobile phones (a ZTE v5 mobile phone and a Samsung GT-I9128 mobile phone are used in our demo) as PSN nodes and used a desktop workstation to serve as the TS. We also applied a second desktop workstation to play as a service provider to provide mobile services, e.g., hotel information provision. At system setup, TS was initiated by generating a number of system credentials.

Operation Performance

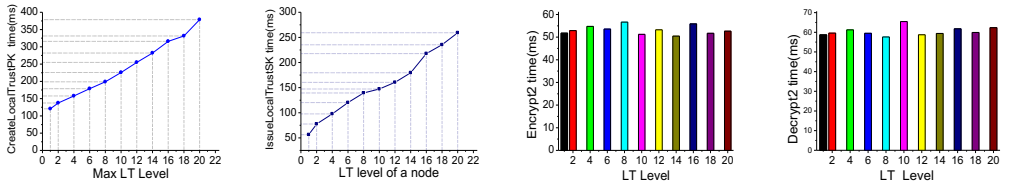
We tested the operation time of the algorithms in our demo system. The implementation used a 160-bit elliptic curve group based on the super singular curve $y^2 = x^3 + x$ over a base 512-bit finite field. As shown in Fig. 6, the GT public key generation time is lineally increased with the maximum level of GT. The GT secret key generation time is also lineally increased with the GT level of a node. The encryption time and decryption time based on GT is almost constant, which is not changed with the GT level of a node. Their operation time is about 6 milliseconds (ms) and 4.5ms respectively at the workstation with Intel Pentium CPU G630 and 2-GB RAM, running Ubuntu 12.04.



(a) (b) (c) (d)

Fig. 6. (a) Operation time of *IssueGeneralTrustPK*; (b) Operation time of *IssueGeneralTrustSK*; (c) Operation time of *Encrypt1*; (d) Operation time of *Decrypt1*

We also tested the operation time of the algorithms related to LT in the mobile phone ZTE v5, as shown in Fig. 7. We observed that the LT public key generation time is lineally increased with the maximum level of LT. The LT secret key generation time is also lineally increased with the LT level of a node. We can see that the encryption time and decryption time based on LT is almost constant, which is not changed with the level of LT. The execution time is about 55ms and 60ms respectively in the tested Android phone. Due to limited computation capacity at the Android phone, the operation time of data encryption and decryption at the Android phone is about 10 times of that in the workstation. But this fact did not impact user experiences very much. In the user test of our demo system, participants cannot feel any delay of encryption and decryption during chatting and service inquiry.

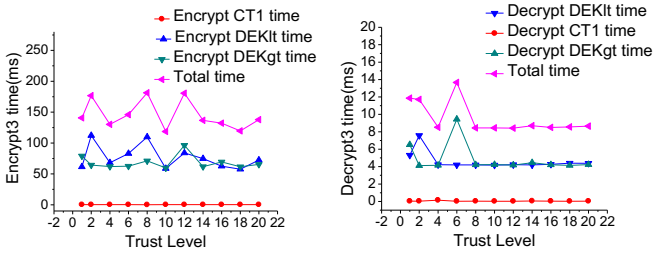


(a) (b) (c) (d)

Fig. 7. (a) Operation time of *CreateLocalTrustPK*; (b) Operation time of *IssueLocalTrustSK*; (c) Operation time of *Encrypt2*; (d) Operation time of *Decrypt2*

We tested the operation time of encryption and decryption based on both GT and LT in a service query initiated by the ZTE mobile phone and responded by the service provider server run in the workstation, as shown in Fig. 8. The PSN node user can control that only the services with sufficient levels of both GT and LT can access the service, thus reply a service query. Fig. 8.a shows the operation time of encryption in the Android phone based on both GT and LT. Because of its limited computing capacity, the total operation time is jittered, but keeps about 160ms as a whole. Fig. 8.b shows the operation time of decryption at the service provider server. We can see that the decryption time is almost stable, about 9ms in total.

The operation time we tested is consistent with our analysis on computation complexity. Due to limited computation capacity at the Android phone, it takes more time to run the algorithms in the Android phone than in the workstation.



(a) (b)

Fig.8: (a) Operation time of encryption in an Android phone based on both GT and LT; (b) Operation time of decryption at a service provider server based on both GT and LT

Demo and Test

We developed an Android application to implement the functions of a PSN node that contains the following five modules, as shown in Fig. 9.

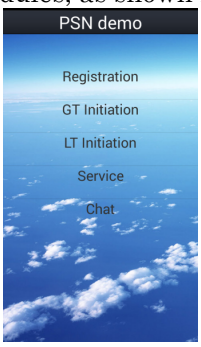
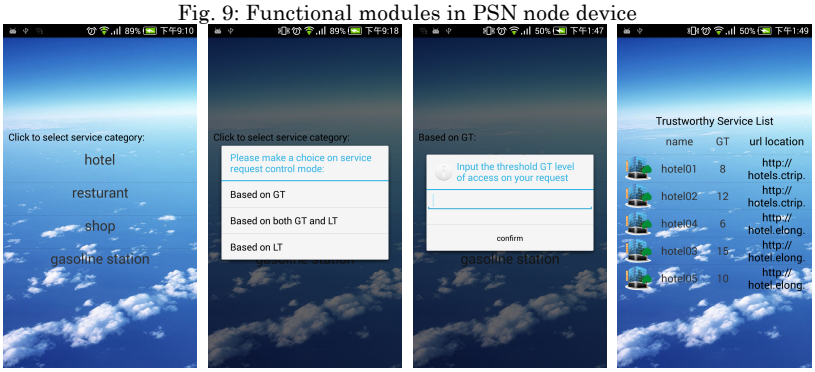


Fig. 9: Functional modules in PSN node device



(a) (b) (c) (d)

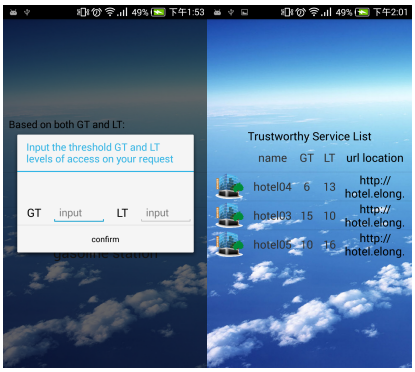
Fig. 10. (a) Main user interface of service module; (b) Three access control manners; (c) Access policy setting with GT; (d) A result list of services satisfying access policy GT ≥ 5

1) Registration: This module helps a PSN node device register into TS. During registration, the TS initiates the GT of the device and records its information into TS database. In our implementation, we set the maximum level of GT as 20.

2) GT Initiation: This module sends a request from a device to TS in order to allow the TS to generate credentials for the device. After getting such a request, the TS checks the information of the PSN device and generates a secret key based on its GT. And then, TS sends the GT public key and the GT secret key to the device via a secure channel.

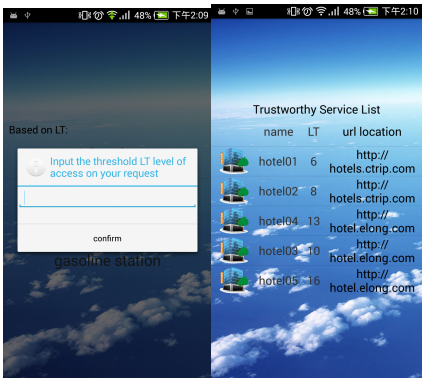
3) LT Initiation: This module initiates the LT keys for PSN node devices. It generates a public key of LT and issues corresponding secret keys to other nodes based on the

LT information recorded in the local node device. We also set the maximum level of LT as 20 in the implementation.



(a) (b)

Fig. 11. (a) Service access policy based on GT and LT; (b) The result list of services with policy $GT \geq 5$ *and* $LT \geq 10$



(a) (b)

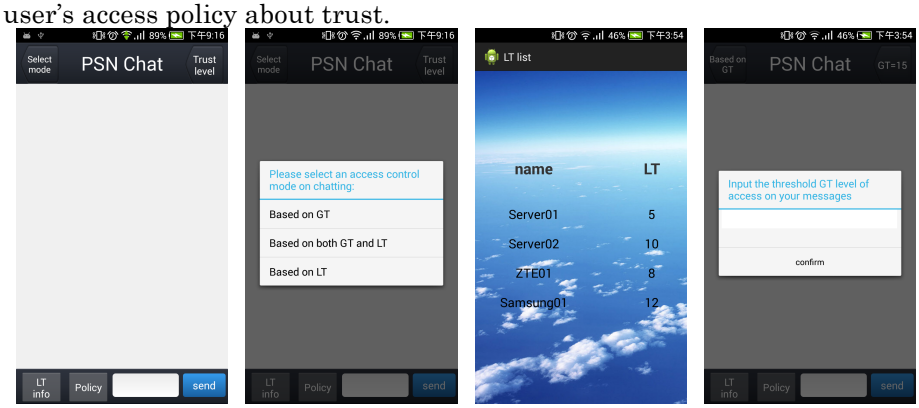
Fig. 12. (a) Service access policy based on LT; (b) The result list of services with policy $LT \geq 5$

4) Service: We implemented a service module to allow the PSN device to query and access various web services based on trust, e.g., hotel services, restaurant services, and mobile shops. A service query is encrypted with a policy related to trust levels (GT, LT or both) as decided by the device user. After inputting the threshold access trust level (e.g., $GT \geq 5$), the query will be encrypted. Then the device broadcasts the query to service providers. The services whose trust levels satisfy the access policy can decrypt the query and get the plain user request. Thus, it is possible for them to provide corresponding services. Only trustworthy services can be displayed in the screen of the PSN device. A user can click their URLs to access them. Fig. 10.a shows the user interface of the service module. A user can select the access control mode of a service query through a button on the top-left. Then the user sets the access policy after choosing the type of services, as shown in Fig. 10.b and Fig. 10.c. Fig. 10.d shows the service query result with the policy $GT \geq 5$. Fig. 11 shows the service query result based on both GT and LT with the policy $GT \geq 5$ *and* $LT \geq 10$. Fig. 12 shows the query result based on LT with the policy $LT \geq 5$.

5) Chat: We also implemented a chatting room to demonstrate the scheme as well. In the chatting room, a device user can choose attribute GT or LT or both to control the access of chatting messages by clicking a button "Select mode" on the top-left of the UI, shown in Fig. 13.a and Fig. 13.b. The button "LT info" allows the user to see LT information of other nodes as shown in Fig. 13.c. And the button "Policy" on the right of "LT info" makes the user able to set the access policy of chatting messages, shown in Fig. 13.d. The node device's GT level is displayed on the top-right during chatting. Only those nodes whose trust satisfied with the access policy can decrypt the ciphertext of chatting messages.

Fig. 14 illustrates an access controlled chatting room set between two Android phones, a Samsung phone A and a ZTE phone B. Fig. 14.a and 14.b are the screenshots of phone A. We use two figures to show A's messages because of the problem of displaying all related messages. Fig. 14.c is the screenshot of phone B. In the showed example: A's GT is 10; and the message sent by A is controlled by GT with a threshold $GT \geq 10$ at the beginning; B's GT is 15 and the message sent by B is controlled by GT with $GT \geq 1$. At beginning, messages sent by A can be displayed by B successfully because B's GT is 15, which is higher than 10. But when A set its access policy as $GT \geq 16$, higher than B's $GT = 15$, the warning message "*** Your

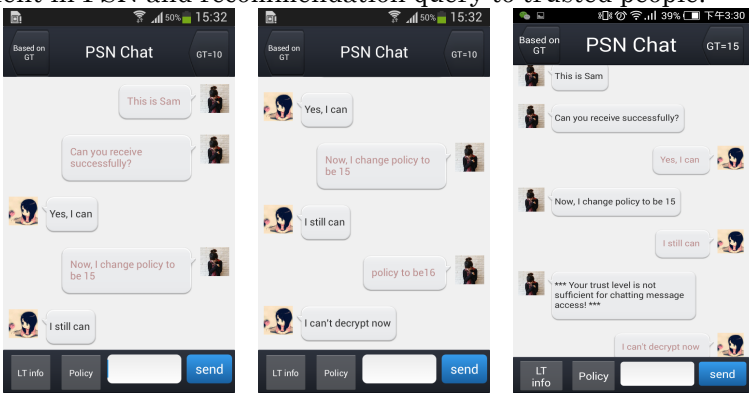
trust level is not sufficient for chatting message access! ***" is displayed by B. This test proves that the implemented scheme can control chatting messages based on the PSN user's access policy about trust.



(a) (b) (c) (d)

Fig. 13: (a) Main user interface of chatting room; (b) Three access control modes on chatting; (c) LT information of other nodes; (d) UI of access control policy setting

We can see that the above demo illustrates the usage and applicability of the scheme. More application use cases can be supported by the scheme. Some examples are controlling the access of group chatting messages by trustworthy group members; location sharing with trusted friends; media sharing with a group of people in PSN; secure payment in PSN and recommendation query to trusted people.



(a) (b) (c)

Fig. 14: (a) Phone A's screenshot; (b) Phone A's screenshot; (c) Phone B's screenshot

User Study

We further studied the user acceptance of our scheme through a small-scale user study. There were 10 participants (40% female) between 22-25 years old with different technical backgrounds participating in our experiment. We installed the demo system in the Android phones of all participants. After usage, each participant was asked to fill in a questionnaire designed based on Technology Acceptance Model (TAM) to feedback perceived ease of use, perceived usefulness, interface, playfulness and acceptance attitude of the demo system. A 5-point Likert scale was applied. The interview result showed that our demo system achieved satisfactory evaluation scores with regard to perceived ease of use ($AVE = 4.2, SD = 0.31$), perceived usefulness ($AVE = 4.4, SD = 0.38$), user interface ($AVE = 3.5, SD = 0.35$), playfulness ($AVE = 4.0, SD = 0.40$), and attitude ($AVE = 3.9, SD = 0.89$). Based on the TAM, we can see that our scheme was welcome by the participants due to good user experiences.

6. DISCUSSIONS

We further discuss generality and limitation of the proposed scheme in this section.

Generality: The proposed scheme takes the advantage of GT and/or LT to secure communication data and flexibly control data access in PSN in either a centralized or a distributed manner. It can protect PSN communication data in a pervasive way based on trust evaluation and management. It can easily secure group social networking by performing group based social trust evaluation. It can support specific communication requirements by embedding such requirements into trust evaluation and management. In addition, data confidentiality can be also considerred in the justification of trust levels for the purpose of controlling data access based on the level of data confidentiality. Moreover, context information, e.g., location and time, can be taken into account in order to control data access with specific context constraits. For example, trust evaluation can be performed based on context requirements. Thus, it is possible for our scheme to support such cases where the relationships are ad hoc and one user would like to send a message readable by people around or by past contacts but by anyone with high enough global and/or local trust level.

Limitation: This work is based on our previous results about trust evaluation in PSN [Yan 2013, Yan et al. 2013, Yan et al. 2014]. At present, we perform trust evaluation according to the behaviors of PSN users and use trust as an attribute to secure and control communication data access. In PSN, context is linked to different social activities. Thus, a more sophisticated policy should be constructed based on context in PSN to achieve a more flexible and fine-grained data protection and access control. In future work, we will improve our scheme by taking context into consideration in both trust evaluation and data access control. Another line of research is to investigate a distributed trust evaluation scheme without any dependency on a centralized server.

7. CONCLUSIONS

In this paper, we presented a scheme to secure communication data in PSN based on the general trust and/or the local trust in a flexible manner by applying KP-ABE. The scheme achieves controlling PSN data access based on trust either in a centralized or distributed manner. We analyzed the security of the proposed scheme under the existing security model and evaluated its effectiveness through extensive analysis on computation complexity, communication cost, scalability and flexibility. We also compared its performance with our previous work using CP-ABE [Yan and Wang 2014] in order to show its suitability for PSN data protection due to its efficiency on communication data encryption. We further developed a demo system to evaluate its operation performance and illustrate the applicability and use cases of the scheme. The results show the correctness of theoretic analysis and the efficiency of the scheme for securing communication data in PSN in practice. For future work, we are going to apply the proposed scheme into more application scenarios in order to motivate its social acceptance and deployment and investigate a distributed trust evaluation scheme with context-awareness.

REFERENCES

AdSocial. ETHz Systems Group, http://www.iks.inf.ethz.ch/publications/files/mobicom08_demo.pdf.

Ari Ahtiainen, Kari Kalliojarvi, Mika Kasslin, Kari Leppanen, Andreas Richter, Paivi Ruuska, and Carl Wijting. 2009. Awareness networking in wireless environments. IEEE Vehicular Technology Magazine, 4, 3 (September 2009), 48-54. DOI: 10.1109/MVT.2009.933475.

Giuseppe Ateniese, Kevin Fu, Matthew Green, and Susan Hohenberger. 2006. Improved proxy re-

encryption schemes with applications to secure distributed storage. ACM Trans. Inf. Syst. Secur. 9, 1 (February 2006), 1-30. DOI: 10.1145/1127345.1127346

Aaron Beach, Mike Gartrell, and Richard Han. 2009. Solutions to Security and Privacy Issues in Mobile Social Networking. In Proceedings of the International Conference on Computational Science and Engineering (CSE'09). Vancouver, BC, Canada, 1036-1042. DOI: 10.1109/CSE.2009.243.

John Bethencourt, Amit Sahai, and Brent Waters. 2007. Ciphertext-Policy Attribute-Based Encryption. In Proceedings of the IEEE Symposium on Security and Privacy. IEEE, Berkeley, CA, 321-334. DOI: 10.1109/SP.2007.11

Guanling Chen and Rahman Faruq. 2008. Analyzing Privacy Designs of Mobile Social Networking Applications. In Proceedings of the 2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC'08). IEEE/IFIP, Shanghai, China, 83-88. DOI: 10.1109/EUC.2008.156.

Nanxi Chen, Mario Gerla, Dijiang Huang and Xiaoyan Hong. 2010. Secure, selective group broadcast in vehicular networks using dynamic attribute based encryption. In Proceedings of the 9th IFIP Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net' 2010). IEEE. Juan Les Pins, France, 1-8. DOI: 10.1109/MEDHOCNET.2010.5546877.

Richard Chow, Philippe Golle, Markus Jakobsson, Elaine Shi, Jessica Staddon, Ryusuke Masuoka, and Jesus Molina. 2009. Controlling data in the cloud: outsourcing computation without outsourcing control. In Proceedings of the 2009 ACM workshop on Cloud computing security (CCSW'09). ACM, New York, NY, USA, 85-90. DOI: 10.1145/1655008.1655020.

EZSetup. http://research.microsoft.com/en-us/groups/wn/mssn.aspx.

Familiar Stranger. Intel Berkeley Lab http://www.paulos.net/research/intel/familiarstranger/index.html.

Wei Feng, Zheng Yan, Haomeng Xie. 2017. Anonymous authentication on trust in pervasive social networking based on group signature. IEEE Access, 2017. Doi: 10.1109/ACCESS.2017.2679980

Eu-Jin Goh, Hovav Shacham, Nagendra Modadugu, and Dan Boneh. 2003. SiRiUS: Securing Remote Untrusted Storage. In Proceedings of Network and Distributed Systems Security Symposium (NDSS'03). San Diego, California, USA, 131-145.

Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. 2006. Attribute-based encryption for fine-grained access control of encrypted data. In Proceedings of the 13th ACM conference on Computer and communications security (CCS'06). ACM, New York, NY, USA, 89-98. DOI: 10.1145/1180405.1180418

Esa Hyytiä, Jorma Virtamo, Pasi Lassila, Jussi Kangasharju, and Jörg Ott. 2011. When does content float? Characterizing availability of anchored information in opportunistic content sharing. In Proceedings of IEEE 2011 INFOCOM. IEEE, Shanghai, China, 3137-3145. DOI: 10.1109/INFCOM.2011.5935160.

Dijiang Huang and Mayank Verma. 2009. ASPE: attribute-based secure policy enforcement in vehicular ad hoc networks. Ad Hoc Networks. 7, 8 (November 2009), 1526-1535. DOI: 10.1016/j.adhoc.2009.04.011.

Junction. Stanford MobiSocial Group, http://openjunction.org/.

Mahesh Kallahalla, Erik Riedel, Ram Swaminathan, Qian Wang, and Kevin Fu. 2003. Plutus: Scalable Secure File Sharing on Untrusted Storage. In Proceedings of the 2nd USENIX Conference on File and Storage Technologies (FAST'03). USENIX Association, Berkeley, CA, USA, 29-42.

Qin Liu, Chiu C. Tan, Jie Wu, and Guojun Wang. 2012. Efficient information retrieval for ranked queries in cost-effective cloud environments. In Proceedings of IEEE 2012 INFOCOM. IEEE, Orlando, FL, USA, 2581-2585. DOI: 10.1109/INFCOM.2012.6195657.

Xuejiao Liu, Yingjie Xia, Wenzhi Chen, Yang Xiang, Mohammad Mehedi Hassan and Abdulhameed Alelaiwi. 2016. SEMD: Secure and efficient message dissemination with policy enforcement in VANET. Journal of Computer and System Sciences. 82, 8 (December 2016), 1316-1328. DOI: 10.1016/j.jcss.2016.05.006.

Blaze Matt, Gerrit Bleumer, and Martin Strauss. 1998. Divertible protocols and atomic proxy cryptography. In Proceedings of International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT'98). Espoo, Finland, 127-144. DOI: 10.1007/BFb0054122.

Green Matthew and Giuseppe Ateniese. 2007. Identity-based proxy re-encryption. Applied Cryptography and Network Security, 288-306. DOI: 10.1007/978-3-540-72738-5_19.

Micro-blog. SyNRG in Duke University, http://synrg.ee.duke.edu/microblog.html.

Emiliano Miluzzo, Nicholas D. Lane, Kristóf Fodor, Ronald Peterson, Hong Lu, Mirco Musolesi, Shane B. Eisenman, Xiao Zheng, and Andrew T. Campbell. 2008. Sensing meets mobile social networks: the design, implementation and evaluation of the CenceMe application. In Proceedings of the 6th ACM conference on Embedded network sensor systems (SenSys'08). ACM, New York, NY, USA, 337-350. DOI: 10.1145/1460412.1460445

Sascha Müller, Stefan Katzenbeisser, and Claudia Eckert. 2009. Distributed attribute-based encryption. In Proceedings of the 11th Annual International Conference on Information Security and Cryptology (ICISC'08). Springer, Seoul, Korea, 20-36. DOI: 10.1007/978-3-642-00730-9.

Nokia Instant Community (NIC), https://lausanne.nokiaresearch.com/nic/

Jörg Ott, Esa Hyytiä, Pasi Lassila, Jussi Kangasharju, and Sougata Santra. 2011. Floating content for probabilistic information sharing. Pervasive and Mobile Computing, 7, 6 (December 2011), 671-689.

DOI: 10.1016/j.pmcj.2011.09.001.

Pairing Based Cryptography. https://Stanford.edu/pbc/.

Matthew Pirretti, Patrick Traynor, Patrick McDaniel, and Brent Waters. 2006. Secure attribute-based systems. In Proceedings of the 13th ACM conference on Computer and communications security (CCS'06). ACM, New York, NY, USA, 99-112. DOI: 10.1145/1180405.1180419

Norman Sadeh, Jason Hong, Lorrie Cranor, Ian Fette, Patrick Kelley, Madhu Prabaker, and Jinghai Rao. 2009. Understanding and capturing people's privacy policies in a mobile social networking application. Personal Ubiquitous Comput. 13, 6 (August 2009), 401-412. DOI: 10.1007/s00779-008-0214-3.

Amit Sahai and Brent Waters. 2005. Fuzzy identity-based encryption. In Proceedings of the 24th International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT'05). Aarhus, Denmark, 457-473. DOI: 10.1007/11426639_27.

Chatterjee Santanu and Ashok Kumar Das. 2015. An effective ECC - based user access control scheme with attribute - based encryption for wireless sensor networks. Security and Communication Networks. 8.9 (June 2015), 1752-1771. DOI: doi: 10.1002/sec.1140.

Kamara Seny and Kristin Lauter. 2010. Cryptographic cloud storage. In Proceedings of the International Conference on Financial Cryptograpy and Data Security (FC'10). Tenerife, Canary Islands, Spain, 136-149. DOI: 10.1007/978-3-642-14992-4_13.

Zhiguo Wan, Jun'e Liu, Robert H. Deng. 2012. HASBE: a hierarchical attribute-based solution for flexible and scalable access control in cloud computing. IEEE Transactions on Information Forensics and Security. 7, 2 (April 2012), 743-754. DOI: 10.1109/TIFS.2011.2172209.

Guojun Wang, Qin Liu, and Jie Wu. 2010. Hierarchical attribute-based encryption for fine-grained access control in cloud storage services. In Proceedings of the 17th ACM conference on Computer and communications security (CCS'10). ACM, New York, NY, USA, 735-737. DOI: 10.1145/1866307.1866414

Guojun Wang, Qin Liu, Jie Wu, and Minyi Guo. 2011. Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers. Computers & Security. 30, 5 (July 2011), 320-331. DOI: 10.1016/j.cose.2011.05.006.

Xinlei Wang, Jianqing Zhang, Eve M. Schooler and Mihaela Ion. 2014. Performance evaluation of attribute-based encryption: Toward data privacy in the IoT. In Proceedings of the IEEE International Conference on Communications (ICC' 2014). IEEE. NSW, Sydney, Australia, 725-730. DOI: 10.1109/ICC.2014.6883405.

Zheng Yan. 2013. Trust Management in Mobile Environments – Usable and Autonomic Models", IGI Global, Hershey, Pennsylvania, USA. DOI: 10.4018/978-1-4666-4765-7.

Zheng Yan, Yu Chen, and Yue Shen. 2013. A practical reputation system for pervasive social chatting. Journal of Computer and System Sciences, 79, 5 (August 2013), 556-572. DOI: 10.1016/j.jcss.2012.11.003.

Zheng Yan, Chen Yu, and Yue Shen. 2014. PerContRep: a practical reputation system for pervasive content services. The Journal of Supercomputing. 70, 3 (December 2014), 1051-1074. DOI: 10.1007/s11227-014-1116-y.

Zheng Yan, Wei Feng, Pu Wang. 2016. Anonymous authentication for trustworthy pervasive social networking. IEEE Trans. on Computational Social Systems, 2, 3, (February 2016), 88-98. DOI: 10.1109/TCSS.2016.2519463

Zheng Yan and Mingjun Wang. 2014. Protect pervasive social networking based on two-dimensional trust levels. IEEE Systems Journal, PP, 9, 1-12. DOI: 10.1109/JSYST.2014.2347259.

Zheng Yan, Mingjun Wang, Valtteri Niemi, and Raimo Kantola. 2013. Secure pervasive social networking based on multi-dimensional trust levels. In Proceedings of the IEEE Conference on Communications and Network Security (CNS'13). IEEE, Washington D.C., USA, 100-108. DOI: 10.1109/CNS.2013.6682697.

Shucheng Yu, Cong Wang, Kui Ren, and Wenjing Lou. 2010. Achieving secure, scalable, and fine-grained data access control in cloud computing. In Proceedings of the IEEE 2010 INFOCOM. IEEE, San Diego, CA, US, 1-9. DOI: 10.1109/INFCOM.2010.5462174

Shucheng Yu, Cong Wang, Kui Ren, and Wenjing Lou. 2010. Attribute based data sharing with attribute revocation. In Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security (ASIACCS'10). ACM, New York, NY, USA, 261-270. DOI: 10.1145/1755688.1755720

Miao Zhou, Yi Mu, W. Susilo, Man Ho Au, and Jun Yan. 2011. Privacy-Preserved Access Control for Cloud Computing. In Proceedings of the IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom'11). IEEE, Changsha, China, 83-90. DOI: 10.1109/TrustCom.2011.14.