This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Yan, Zheng; Shi, Wangyang

CloudFile

Published in:
Journal of Network and Computer Applications

DOI:
[10.1016/j.jnca.2016.11.016](https://doi.org/10.1016/j.jnca.2016.11.016)

Published: 15/05/2017

Document Version
Peer reviewed version

Published under the following license:
CC BY-NC-ND

Please cite the original version:
Yan, Z., & Shi, W. (2017). CloudFile: A Cloud Data Access Control System based on Mobile Social Trust. *Journal of Network and Computer Applications*, 86, 46-58. <https://doi.org/10.1016/j.jnca.2016.11.016>

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

CloudFile: A Cloud Data Access Control System based on Mobile Social Trust

Zheng Yan^{1,2}, Wangyang Shi¹

- ^{1.} The State Key Laboratory on Integrated Services Networks, School of Cyber Engineering, Xidian University, Xi'an 710071, China
- ^{2.} The Department of Communications and Networking, Aalto University, Espoo 02150, Finland

Corresponding author: Zheng Yan, Email: zyan@xidian.edu.cn

Abstract— Personal data management at cloud with mobile devices confronts a number of practical challenges such as security, efficiency and usability. Many existing solutions suffer from high computation and communication costs, thus impractical to be deployed in mobile domains. Notably, high popularity of mobile social networking provides a good platform to assess social trust in a pervasive way, which can be applied to automatically manage personal data access at the cloud with sound performance and effectiveness. This paper presents the design and implementation of a personal cloud data access control system named CloudFile based on trust evaluation in mobile social networking. The CloudFile uses a Key Policy-Attribute Based Encryption (KP-ABE) scheme to control personal data access based on social trust in order to guarantee the safety of mobile cloud data storage. System performance tests show that the CloudFile is flexible for controlling data access, secure for personal data storage, and efficient to be deployed by mobile devices. Particularly, we investigate user acceptance on the CloudFile system. Based on user feedback, we conclude that the CloudFile is accepted with regard to perceived ease of use, practicality, user interface, playfulness and intuitive attitude.

Index Terms— Cloud Storage, Data Access Control, Attribute-Based Encryption, Trust Management, Mobile Social Networking

1 INTRODUCTION

In recent years, cloud computing has caught a wide attention in both academia and industry. As a new service model, the cloud computing provides a platform to offer various services such as data storage, software development, computations, and so on based on user demands. One of the most popular cloud services is data storage. Nowadays, more and more mobile users need to store their personal data at the cloud due to limited storage spaces in their personal mobile devices.

However, the risk of personal data leakage at the cloud scares users. Personal privacy intrusion and information disclosure could bring a huge loss to users. This retards user adoption on cloud data storage services. Cloud data security becomes more and more severe. If the cloud server is not secure and data stored in a cloud server is plain text [1], after hackers break through the barrier system of Cloud Service Providers (CSPs), stealing user data will be very simple. So it has been suggested that the sensitive data stored at the cloud should be encrypted [1]. Even if the cloud server has been compromised, the data stored there are still safe.

Researchers have proposed a variety of solutions for protecting access to personal data in cloud. However, schemes based on Access Control List (ACL) suffer from the drawback that the computation complexity grows line-

arly with the number of data-groups [2] or the number of users in the ACL [3]. Traditional symmetric encryption or public key encryption cannot achieve fine-grained data access control. In order to overcome the above problems, schemes based on Attribute-Based Encryption (ABE) [5, 6] were proposed to provide flexible and generic cloud data access control [13-15, 18, 19, 22, 24]. However, the computation cost of these schemes is generally high due to the complexity of attribute structure. Thus, few of the existing solutions have been practically deployed, especially in mobile domains. Existing studies lack investigation on a lightweight and efficient solution to achieve fine-grained mobile cloud data access control with sound usability and practicality. Current research is still at the stage of academic study. More efforts are needed to explore applicable schemes with regard to mobile cloud data access control that can be implemented with mobile devices. Notably, little work was proposed to control the access of data stored at the cloud based on the trust assessed in mobile social networking [13-15, 17-19], while trust plays an important role to help people avoid risk and uncertainty.

Mobile social networking statistics reflect the level of trust among users. With the rapid growth of mobile communications and networking and the wide usage of mobile devices, people nowadays perform various social activities

and networking using their mobile devices in a pervasive way, e.g., calling through mobile cellular networks, chatting via the mobile Internet, sending short messages using SMS, and conducting various instant social activities via mobile social networking applications (e.g., WeChat and WhatsApp). Trust relationships in different contexts can be assessed based on mobile social networking activities, feedback, behaviors and experiences. For example, people rarely communicate with strangers, but contact familiar people such as friends and family members frequently. Their feedback on social activities reflects their opinions on interacted counterparties. Most people would like to share their personal importance, happiness, sadness and opinions with those closest to them due to trust. For example, when facing a serious issue, a mobile user normally would like to get advice from his family and good friends through phone calls. For solving some trouble, a user firstly contacts the person he trusts most, no matter where he is. Mobile social networking helps people overcome time and region and allow them to communicate efficiently and conveniently. We would like to argue that the social trust that is reflected from mobile social networking could be utilized to control private data access.

Using social trust to control access of the personal data stored at the cloud has a number of advantages [23]. First, trust relationships can be automatically evaluated by mobile devices based on mobile social networking statistics. This provides great usability for cloud data access control. Second, using trust to control data access is simple and lightweight because it replaces the complexity of hierarchical attribute-based fine-grained access control [22] with trust based one. We simplify the attribute structure by only considering one attribute: trust. This design not only reduces the complexity of access policy description, but also keeps its expressivity. Trust can be evaluated according to many factors. But the computation of trust evaluation is much less complicated than embedding the trust-factor related attributes into the attribute structure of an ABE scheme. Third, a scheme to control personal data access in cloud based on social trust can take the advantages of both social networking and cloud computing. Concretely, we analyze social networking data in order to figure out user trust relationships, with which we further control data access at the cloud in an automatic and autonomic way.

This paper presents the design and implementation of CloudFile: a personal cloud data access control system based on trust evaluation in mobile social networking. The CloudFile uses a Key Policy-Attribute Based Encryption (KP-ABE) scheme to control personal data access based on social trust evaluation in order to guarantee the safety of mobile cloud data storage. Concretely, the trust level with regard to a concrete context (e.g., health treatment) can be assessed based on social networking activities, feedback, behaviors and experiences. According to the trust level and its linked context, the data owner encrypts his/her personal data by setting a trust threshold and specifying access context for the encrypted data. The data owner issues decryption keys to eligible users. Thus, personal data access (with a condition such as the trust level being higher than the pre-defined threshold in a specific context) can

be fully controlled by the data owner. Due to the dynamic changes of trust levels, the data owner informs CSP the blacklist of the users who are not eligible to access the data according to concurrent trust evaluation results. The users who do not satisfy the access policies cannot access the data any more through the control of CSP.

The CloudFile is original and differs substantially from existing work. It applies context-aware social trust as a specific attribute to realize fine-grained and efficient access control on cloud data. It can be implemented and practically utilized by mobile users. Specifically, the main contributions of this paper can be summarized as follows:

- 1) We motivate controlling encrypted cloud data access based on the trust evaluated through mobile social networking. The CloudFile is one of the first systems to control cloud data access via mobile devices based on KP-ABE by using social trust levels as control attributes in a context-aware manner;

- 2) We analyze the performance of the CloudFile with regard to security, computation complexity and communication cost. The result shows its effectiveness;

- 3) We implement the CloudFile using Android phones and prove its efficiency and user acceptance through system performance tests and a user study.

The rest of the paper is organized as follows. Section II gives a brief overview of related work. Section III introduces the system and our design goals. Then we provide the detailed design of the CloudFile in Section IV. Section V analyzes the security, effectiveness and efficiency of the CloudFile, followed by the system performance test results. Finally, a conclusion is presented in the last section.

2 RELATED WORK

2.1 Access Control on Cloud Data

The issue of access control on cloud data was introduced due to data encryption for security protection [1-3]. Different cryptographic mechanisms are applied for this purpose. Using an ACL, a data owner classifies data into a data-group and then encrypts each data-group with a symmetric key that is shared with the users in the ACL [4]. The main problem of this method is the number of keys grows linearly with the number of data-groups. Key management becomes complicated due to trust relationship change between users and the data owner. Because this causes the symmetric key revoked, thus impacts other users in the same ACLs. Combining traditional symmetric key and public key cryptographic systems, the symmetric key used for data encryption can be issued using the public keys of users in an ACL [5]. Therefore, only the users in the ACL can recover the data using their private keys. But the cost of symmetric key encryption grows linearly with the number of users in the ACL. This approach cannot efficiently support frequent changes of trust relationships, either.

Role Based Access Control (RBAC) provides flexibility on access control management corresponding to an organization's policy and structure. Zhou et al. proposed a Role-Based Encryption (RBE) [28] scheme that enforces RBAC policies on encrypted data stored in cloud with an efficient

user revocation mechanism. Only the users with appropriate roles specified by a RBAC policy can decrypt the data. A dynamic role based access control framework was proposed by integrating trusted computing with RBAC in cloud computing [29]. Most of RBAC mechanisms for cloud computing cannot satisfy various data access demands based on trust and fine-grained access control inside a role [30, 31].

Attribute based access control is an important technical division for securing cloud data [6-9, 12]. In an ABE system, data is encrypted with an attribute-based access structure, such that only the users whose attributes satisfy the access structure can decrypt the data. ABE is classified into two branches, Key-Policy ABE (KP-ABE) [7] and Ciphertext-Policy ABE (CP-ABE) [6, 8] depending on how attributes and policies are associated with ciphertexts and decryption keys.

Secure cloud data storage schemes based on ABE can achieve flexibility, scalability and fine-grained access control [13-15, 18, 19, 22, 24]. An example is a Hierarchical Attribute-Set-Based Encryption (HASBE) for data access control in cloud computing by extending Ciphertext-Policy Attribute-Set-Based Encryption (ASBE) with a hierarchical structure of users [22]. Yu et al. proposed an access control scheme based on KP-ABE, Proxy Re-Encryption (PRE) and lazy re-encryption for efficient user revocation [13]. This scheme locates most of computational overhead to cloud servers and provides user access privilege confidentiality and user secret key accountability. Fine-grained access control is supported gracefully by applying KP-ABE. Hierarchical ABE (HABE) proposed by Wang, Liu and Wu achieves fine-grained access control on cloud data by combining Hierarchical Identity-Based Encryption (HIBE) and CP-ABE [14, 18]. A secure overlay cloud storage system named FADE was proposed to provide fine-grained access control and assure deletion for outsourced data on cloud [25].

However, existing schemes based on ABE normally used a complicated attribute structure to achieve fine-grained access control. The computation cost of key management and data protection is heavy. This makes the existing schemes not suitable to be applied in practice, especially in mobile domains. Notably, trust plays a decisive role in data sharing. Using social trust to control cloud data access is a useful supplement of prior arts. Our previous work explored using trust and reputation to control cloud data access in a flexible way based on CP-ABE [11, 26]. But we have not yet investigated its real performance in implementation and the practicability of system deployment [11, 26, 27].

2.2 User Revocation

User revocation is not a trivial task. The key problem is that the revoked users still retain the keys issued earlier, and thus can still decrypt ciphertexts. The work in [10] proposed to require the data owner to periodically re-encrypt the data, and re-distribute new keys to authorized users. This approach is obviously inefficient due to the heavy processing overhead introduced to a data owner.

A better solution is to let the data owner delegate a

third party to execute some computational intensive tasks, e.g., re-encryption, while leaking the least information. For example, the work in [13] combined KP-ABE and PRE to delegate most of the computation tasks in revocation to CSP. The work in [14] combined PRE and a CP-ABE system to achieve a scalable revocation mechanism in cloud computing. Attribute revocation was supported in [15]. It requires that once a user is revoked from a system, the data owner should send PRE keys to the CSP, with which the CSP can be delegated to execute re-encryption. The main problem of this approach is that the data owner should be online in order to send the PRE keys to the CSP in a timely fashion. Yang et al. constructed a new multi-authority CP-ABE scheme with efficient decryption and proposed an efficient attribute revocation method for multi-authority cloud storage [17].

In our proposed scheme, revocation is implemented through the cooperation between the data owner and the CSP. The data owner performs key re-generation if necessary.

2.3 Trust Based Access Control

There are many reputation management systems in the literature [16]. Lin et al. proposed a mutual trust based access control (MTBAC) model [20] by taking both user behavior trust and cloud service credibility into consideration. However, most existing work did not consider how to control personal data access based on trust over the cloud [21], especially by collaborating with mobile social networking to automatically gain trust evidence for trust evaluation [21, 24], and further applying the trust to automatically control cloud data access.

3 THE PROPOSED SYSTEM

3.1 CloudFile Overview

The CloudFile simply consists of two types of entities: user mobile devices and CSP for data storage, as shown in Figure 1. Users use the mobile devices to interact with the CSP for personal file upload, download and management. Mobile devices perform a variety of social communications and networking activities.

Fig. 1. A system model

The CSP stores encrypted data of users and allows data access according to the control strategy of a data owner (i.e., a system user). It performs by following system design and protocols. But it is curious on the contents of user

data and tries to gain profits from these data. Internal data leakage could happen at the CSP. Meanwhile, the CSP could be the target of any network intrusions and attacks. Thus, the CSP cannot be fully trusted.

3.2 Notations/Preliminaries and Definitions

TABLE 1: NOTATIONS

Symbols	Description
$N_{c(i,j)}$	The number of calls made by u_i to u_j ;
$N_{m(i,j)}$	The number of messages sent by u_i to u_j ;
$N_{l(i,j)}$	The number of interactions initiated by u_i to u_j ;
$TL(u_i, u_j)$	The trust value of u_j assessed by u_i ;
w_1	The weight parameter to show the importance of voice call, $w_1 + w_2 + w_3 = 1$;
w_2	The weight parameter to show the importance of messaging;
w_3	The weight parameter to show the importance of instant interaction;
$pl(i, j)$	The priority level of u_j in u_i 's social networks, e.g., a family member or a close friend may have the highest priority level in various contexts. The value of $pl(i, j)$ is decided by the social importance and the role of u_j for u_i ;
$pu(i, j)$	The punishment factor of u_j in u_i 's view, e.g., due to bad social interaction/communication experiences;
$\theta(N)$	The Rayleigh cumulative distribution function $\theta(N) = 1 - e^{-N^2/2\sigma^2}$ to model the impact of an integer number N , where $\sigma > 0$, is a parameter that inversely controls how fast number N impacts the increase of $\theta(N)$;
$f(x)$	The Sigmoid function $f(x) = 1/(1 + e^{-\alpha x})$, it is used to normalize a value x into $(0, 1)$. Parameter α is applied to control the normalization;
I_{tl}	The maximum level of trust level;
PK_TL	The public key of trust level;
MK	The master key;
SK_TL_u, u	The secret key of user u ;
DEK	The symmetric key for encrypting a data file;
M	The plaintext of a data file;
CT	The ciphertext of a data file;
$R(t_e)$	The reputation value of CSP at t_e ;
$V(k, t)$	The vote of user k on CSP at time t ;
$C(k, t)$	The credibility of user k w.r.t. feedback provision at time t .

Table 1 summarizes the notations used in this paper.

Bilinear pairing: Let G_1 and G_2 be two cyclic multiplicative groups with the same prime order p , i.e., $|G_1| = |G_2| = p$. Let g be a generator of G_1 . A bilinear map $e: G_1 \times G_1 \rightarrow G_2$ has the following properties:

Bilinear: for all $u, v \in G_1$ and $a, b \in \mathbb{Z}_p$, $e(u^a, v^b) = e(u, v)^{ab}$.

Non-degenerate: $e(g, g) \neq 1$ for the generator g .

Computable: there is an efficient algorithm to compute $e(u, v)$ for any $u, v \in G_1$.

Definition: Trust Level (TL) is the trust evaluated by a user based on mobile social networking activities, feedback, behaviors and experiences. The value of trust can be divided into a number of discrete levels, e.g., i represents the i -th level of TL, $i \in (0, I_{tl})$, where I_{tl} is the maximum level of TL.

To effectively secure private data, we resort to controlling the data access based on the trust levels assessed in mobile social networking by applying the KP-ABE technique [7], particularly distributed ABE [8]. We select KP-ABE not CP-ABE to demonstrate the CloudFile system

because of the following reasons. First, the data encryption based on KP-ABE is not changed with the size of encrypted data, or the policy of data access control. Thus, the communication cost can be well controlled since this greatly impacts the performance of file uploading and downloading with mobile devices. Second, although applying KP-ABE spends more time on key generation than CP-ABE, this does not impact much on the user cost regarding cloud storage service usage and networking usage since it is impacted by the size of stored or uploaded data. The users could benefit more from a KP-ABE scheme than a CP-ABE scheme.

3.3 Trust Evaluation based on Mobile Social Networking

Trust can be evaluated based on the clue showed in mobile social networking. We classify the types of social networking and communications as: a) mobile voice calls; b) voice/video calls via mobile Internet (e.g., VoIP); c) short messages; d) instant social messages; e) pervasive interactions based on local connectivity. The number of voice calls (called and received), the number of interactions and the number of messages (sent and received) reflect the social closeness between two persons and their personal trust with each other.

We propose a trust evaluation function (1) to calculate the trust value $TL(u_i, u_j)$ of two users u_i and u_j based on mobile social networking. The notations used for trust evaluation are depicted in Table 1. The numbers of different types of mobile social communications ($N_{c(i,j)} + N_{c(j,i)}$, $N_{m(i,j)} + N_{m(j,i)}$, $N_{l(i,j)} + N_{l(j,i)}$) are considered in the trust evaluation of user u_i on u_j by applying the Rayleigh cumulative distribution function $\theta(N)$. $\theta(N)$ is used to model the impact of an integer number N on the trust level $TL(u_i, u_j)$ of u_j assessed by u_i . Its parameter σ ($\sigma > 0$) is applied to inversely control how fast N influences $TL(u_i, u_j)$. Meanwhile, the importance weight (w_i) of each type of social networking is used to further tailor the impact of the Rayleigh cumulative distribution function on the evaluation. Furthermore, the priority level of u_j in u_i 's social networks ($pl(i, j)$) that indicates the degree of social closeness is applied to weight the above social networking statistics. In particular, we apply a punishment factor ($pu(i, j)$) of u_j in the view of u_i to reduce $TL(u_i, u_j)$ due to bad social interaction experiences of u_i with u_j . Finally, the Sigmoid function is used to normalize the trust value that is calculated based on its old value and newly collected social networking statistics as described above into $(0, 1)$. Thus, $TL(u_i, u_j)$ is evaluated as below by aggregating all above:

$$TL(u_i, u_j) = f\{TL(u_i, u_j) + pl(i, j) \times \{w_1 \cdot \theta(N_{c(i,j)} + N_{c(j,i)}) + w_2 \cdot \theta(N_{m(i,j)} + N_{m(j,i)}) + w_3 \cdot \theta(N_{l(i,j)} + N_{l(j,i)})\} - pu(i, j)\} \quad (1)$$

Particularly, the above trust level can be optionally linked to a concrete context based on semantic analysis on messaging texts, voice recognition or context detection. That is $N_{c(i,j)}$, $N_{m(i,j)}$ and $N_{l(i,j)}$ are recorded based on different context categories, thus different $TL(u_i, u_j)$ is generated regarding different contexts. Based on the trust evaluation in different contexts, the data owner can control data

access automatically.

3.4 Secure Data Access based on KP-ABE

Let G_1 be a bilinear group with order p and generator g of G_1 . A bilinear map $e: G_1 \times G_1 \rightarrow G_2$ is generated with K as a security parameter that determines the size of the groups. The algorithms used for secure data access based on KP-ABE are described in details as below.

GenerateTrustPK(). This algorithm defines an attribute collection $U = (1, 2, \dots, i, \dots, I_{tl})$, where i represents the i -th level of trust, $i \in (0, I_{tl}]$ and I_{tl} is the maximum level of TL. For each trust level i , the algorithm uniformly selects a number denoted t_i from Z_p randomly and chooses y uniformly at random in Z_p . Then, public key of trust level PK_{TL} and master key MK are generated at each data owner as below:

$$PK_{TL} = \{G_1, G_2, e, g, Y = e(g, g)^y, T_1 = g^{t_1}, \dots, T_{I_{tl}} = g^{t_{I_{tl}}}\}, \quad (2)$$

$$MK = \{y, t_{T_1}, \dots, t_{I_{tl}}\}. \quad (3)$$

IssueTrustSK(PK_{TL}, TL_u). This algorithm is executed by the device of a data owner to issue a secret TL key. Based on user u 's trust level, for instance, TL_u , it sets the access policy of u : u can only decrypt the ciphertexts encrypted by TL i where $i \leq TL_u$. The algorithm 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_i , where $i \leq TL_u$. The algorithm generates

$D_i = g^{\frac{q_r(0)}{t_i}}$ for leaf i with attribute t_i . The secret key is:

$$SK_{(TL_u, u)} = \left\{ D_1 = g^{\frac{q_r(0)}{t_1}}, \dots, D_i = g^{\frac{q_r(0)}{t_i}}, \dots, D_{N_{gt}} = g^{\frac{q_r(0)}{t_{N_{gt}}}} \right\}. \quad (4)$$

Encrypt(PK_{TL}, TL_i, DEK, M). The algorithm takes as input public key PK_{TL} , trust level TL_i , symmetric key DEK and message M . It encrypts DEK with trust level i and encrypts M with symmetric key DEK . It selects a random value $s \in Z_p$ and generates ciphertext CT as:

$$CT = \{i, C'_{tl} = DEK \times Y^s, C_i = g^{s \times t_i}, C''_{tl} = (M)_{DEK}\}. \quad (5)$$

Decrypt(PK_{TL}, SK_(TL_u, u), CT). The algorithm takes as input PK_{TL} , $SK_{(TL_u, u)}$, and encrypted message CT . It decrypts CT with $SK_{(TL_u, u)}$ and outputs plaintext M . This process is conducted at user u as follows.

$$Decrypt(CT, PK_{TL}, SK_{(TL_u, u)}) = e(D_i, C_i) =$$

$$e\left(g^{\frac{q_r(0)}{t_i}}, g^{s \times t_i}\right) = e(g, g)^{s \times q_r(0)} = e(g, g)^{y \times s} = Y^s, \quad (6)$$

$$DEK = \frac{C'_{tl}}{Y^s}. \quad (7)$$

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

3.5 Additional Access Control by CSP and CSP Reputation Generation

Due to the dynamic change of the trust level, earlier eligible users could become distrusted. In this case, the data owner will not allow them to access his/her data any more although the secret attribute keys have been issued and their valid access period has not expired. For controlling

data access by distrusted users, the data owner announces a blacklist of the users who are not eligible to access the data in the valid access period. Thus, those users whose current trust levels are below a threshold (i.e., do not satisfy the access control conditions) or blacklisted cannot access the data any more through the control of the CSP. The CSP can also block data access after the access time period expires and remind the data owner to upload newly encrypted data.

One approach to ensuring the CSP to perform the above additional access control by following the data owner's instruction is to apply a reputation mechanism. Each CSP's reputation is evaluated according to user feedback and published in order to encourage and ensure good behaviors of CSP. A disreputable CSP will easily lose its customers and profits. Many existing reputation mechanisms can be applied to evaluate a CSP's reputation based on user feedback [23]. Suppose user k 's voting on the CSP at time t is $V(k, t)$, and the user's own credibility of providing feedback at time t is $C(k, t)$, we can calculate the reputation value $R(t_e)$ of the CSP at time t_e by weighting $V(k, t)$ with $C(k, t)$ in order to overcome some potential attacks, such as bad mouthing attacks. $C(k, t)$ can be generated based on the performance of user k 's vote by considering all the feedback provided by all users who experienced the CSP's data storage service. Meanwhile, we also take into account the influence of the time of voting on the reputation calculation since the latest votes impact the reputation more than old ones. Logically, we should pay more attention to the current votes. Herein, we use $e^{-\frac{|t_e - t|}{\tau}}$ to model vote decaying over time in reputation calculation. Parameter τ is used to control such decaying. Obviously, the total number of votes (K) is also an impact factor on reputation. The bigger the number, the more convinced the reputation is generated. Again, we apply the Rayleigh cumulative distribution function $\theta(K)$ as defined in Table 1 to model the impact of K on $R(t_e)$. The final value of $R(t_e)$ is calculated by summing $V(k, t) \cdot C(k, t) \cdot e^{-\frac{|t_e - t|}{\tau}}$ and getting its average by dividing the sum with $\sum_{k=1}^K C(k, t) e^{-\frac{|t_e - t|}{\tau}}$ (i.e., the sum of the tail of $V(k, t)$), and further tailoring the above result with $\theta(K)$. Thus, we have the following formula to calculate $R(t_e)$.

$$R(t_e) = \frac{\theta(K)}{\rho} \sum_{k=1}^K V(k, t) \cdot C(k, t) \cdot e^{-\frac{|t_e - t|}{\tau}}, \quad (8)$$

where $\rho = \sum_{k=1}^K C(k, t) e^{-\frac{|t_e - t|}{\tau}}$. The reputation generation could be offered by a third party, e.g., another cloud computing service or an authorized trusted party. Its computation cost has no influence on any user devices.

3.6 Procedure

Figure 2 illustrates the procedure of data access control based on the trust evaluated in mobile social networking. We suppose that user $u1$ saved its sensitive personal data at the CSP data center, while user $u2$ would like to access it with the authorization of $u1$.

Step 1: $u1$ conducts trust evaluation based on mobile social networking activities, feedback, behaviors and experiences. This trust evaluation can be periodically executed or triggered by bad experiences.

Step 2: Based on the trust evaluation results, $u1$ sets its

sensitive data access policy (such as a trust level threshold). It encrypts DEK with trust level i and encrypts M with symmetric key DEK by calling $Encrypt(PK_{TL}, TL_i, DEK, M)$.

Step 3: u_2 would like to access u_1 's data. It requests the CSP that checks the validity of its ID in order to decide whether it should forward its request to u_1 if it is not in the blacklist of u_1 . u_2 can also request u_1 directly about data access.

Step 4: u_1 checks the eligibility of u_2 in order to generate personalized secret keys for u_2 by calling $IssueTrustSK(PK_{TL}, TL_u)$ to access the data stored at the CSP. By considering the trust level of u_2 together with context information, context-aware access control based on TL can also be supported in the CloudFile.

Step 5: u_2 accesses the encrypted u_1 data with the issued secret key. But before initiating the access, the CSP firstly checks the access policy defined by u_1 and the blacklist of u_1 to find out if u_2 is inside it. If not, the CSP allows the access by providing corresponding encrypted data CT . Then u_2 can decrypt the encrypted data by calling $Decrypt(PK_{TL}, SK_{(TL_u, u)}, CT)$.

Step 6: u_1 re-evaluates trust based on latest mobile social networking evidence. If there are users who have been issued the secret keys but are not eligible any more, u_1 will put them into its data access blacklist and inform the CSP.

Step 7: u_1 feedbacks the performance of the CSP by voting and commenting it to a third trusted party for CSP reputation generation.

Fig. 2. A procedure of data access control based on the trust evaluated in mobile social networking

4. ANALYSIS & PERFORMANCE EVALUATION

The security goal of our scheme is to guarantee that only the users whose trust satisfies the access control policies of the data owner in the context of data usage can access the data stored at the cloud.

4.1 Security Proofs

Fine-Grained Access Control

Our scheme can achieve fine-grained access control. Various access policies regarding TL can be defined and

enforced. We remove the complexity of hierarchical attribute-based fine-grained access control [22] by replacing it with the policy based on trust level. Fine-grained control is achieved by evaluating the trust levels according to many factors based on mobile social networking; thus simplifying the attribute structure of ABE. This design not only reduces the complexity of the access policy description, but also keeps its expressivity. The computation complexity can also be greatly reduced. Trust level calculation does not contain the exponentiation operation in the Sigmoid function $f(x)$ since we only need to match the trust value into a concrete level (refer to [23] for a concrete implementation). The computation of trust level is much less complicated and faster than embedding the trust-factor related attributes into the attribute structure of the KP-ABE scheme. The proposed scheme for data access control can be flexibly applied into many scenarios by cooperating with a trust management framework. Context-awareness can be easily supported with context-aware trust evaluation.

Data confidentiality

In the CloudFile, the user data are encrypted with a symmetric key DEK , which is encrypted using the KP-ABE algorithm. Assuming that the symmetric key algorithm is secure, e.g., using a standard algorithm such as AES, the data confidentiality of the CloudFile relies on the security of the KP-ABE algorithm. Regarding the security proof of KP-ABE, refer to [7]. KP-ABE was proved secure under an attribute-based selective-set model by reducing to the hardness of the Decisional Bilinear Diffie-Hellman (BDH) assumption. The proof shows that no polynomial-time adversary can get a non-negligible advantage to decrypt the ciphertext of KP-ABE.

4.2 Performance Analysis

This section analyzes the performance of the proposed scheme in terms of computation complexity and communication cost.

Computation Complexity

We analyze the computation complexity of the proposed scheme in terms of the following algorithms: GenerateTrustPK, IssueTrustSK, Encrypt and Decrypt.

For the algorithms Encrypt, and Decrypt, each contains a constant number of exponentiation operations on group G_1 regarding DEK encryption and decryption. So the computation complexity of them is $\mathcal{O}(1)$. Computation complexity of encryption and decryption of data M depends on the size of M , which is inevitable in each scheme.

The algorithms GenerateTrustPK and IssueTrustSK, require one exponentiation operation on group G_1 for each conjunction in the access policy. So the computation complexity of these algorithms is $\mathcal{O}(W)$, where W denotes the number of conjunctions, $W \leq I_{ti} + 1$.

Table 2 summarizes the computation complexity of each system operation in the proposed scheme and compares it with two other schemes based on KP-ABE [13] and CP-ABE [22].

TABLE 2: COMPARISON OF COMPUTATION COMPLEXITY

Operation	Our	HASBE	Yu's scheme
-----------	-----	-------	-------------

	scheme	[22]	[13]
Setup	—	$\mathcal{O}(1)$	$\mathcal{O}(Y)$
User Initiation (Public Key Generation)	$\mathcal{O}(I_{tl})$	$\mathcal{O}(2M + s)$	$\mathcal{O}(Y)$
Key Encryption	$\mathcal{O}(1)$	$\mathcal{O}(2Y + X)$	$\mathcal{O}(S)$
Key Decryption	$\mathcal{O}(1)$	$\mathcal{O}(2Y + 2X)$	$\mathcal{O}(\max(Y , N))$
Secret Key Generation	$\mathcal{O}(W)$	N.A.	$\mathcal{O}(Y)$
Revocation	$\mathcal{O}(W)$	$\mathcal{O}(1)$	$\mathcal{O}(Y)$

Notes: W : the number of conjunction in a specified access policy, $W \leq I_{tl}$, I_{tl} is the maximum numbers of trust levels; Y : the number of leaf nodes in an access policy tree; S : the attribute set; M : the number of attributes in S ; X : translating nodes of access policy tree; N : the number of group multiplication operations.

The comparison shows that our scheme is very simple and efficient since the computation complexity is $\mathcal{O}(1)$ for encrypting and decrypting DEK . We note that the value of W is normally reasonable since I_{tl} is generally not big, e.g. below 10 in many applications. The execution time with only one attribute specified policies performs much more efficiently than other schemes with more than one attribute.

Communication Cost

An essential part of the communication cost relates to the size of ciphertext. The ciphertext contains the encrypted data that is inevitable in a cryptography method, and the encrypted DEK . The size of the latter part of the ciphertext keeps constant, which is about 270 bytes if the size of DEK is 128 bits. Additionally, we apply the access policy and trust evaluation to assist decisions on key generation and exchange in order to minimize communication cost in various situations. From the communication cost point of view, applying the scheme based on KP-ABE is more economic and practical than a scheme based on CP-ABE. This is because the cipherkey size based on CP-ABE is larger than that based on KP-ABE. The cipherkey size is linearly increased with the threshold of trust level if applying CP-ABE, while it is constant if KP-ABE is applied.

5 SYSTEM IMPLEMENTATION

We implemented the CloudFile system and demonstrated its main functions. The KP-ABE algorithms were implemented with C language based on Pairing Based Cryptography (PBC) Library [32] on a workstation. Furthermore, we transplanted the core code of the algorithms into Android mobile phones through cross-compiler. We adopted Android mobile phones (HUAWEI Honor 3C", Android 4.3, 1G RAM, Snapdragon MSM8612, Quad-Core, clock speed 1.2GHz) in our prototype system as mobile user client devices. The CSP data storage server was simulated by a workstation, Compaq Presario CQ40, Window7 X86, Pentium(R) Dual-Core CPU T4300, clock speed 2.1GH, 2GB RAM. At system setup, every mobile device was initiated by generating a number of its own credentials. The mobile phones connect the CSP server through WiFi.

5.1 Main System Functions

Fig. 3. File upload: (a) file upload UI; (b) upload processing progress; (c) upload success UI with recorded file encryption time and uploading time

Fig. 4. UI for setting trust threshold and blacklist/whitelist for file access

Fig. 5. Read and delete file

From a user client point of view, the prototype system contains system registration and login function, user file management function, and trust evaluation function, as well as big file process function. The file management function is responsible for uploading a user file to the

cloud server after encrypting it (refer to Figure 3) by setting up access control policies (e.g., an eligible trust threshold, an access blacklist and an access whitelist), as shown in Figure 4. A user personal file stored at the server can be downloaded, opened and deleted from the server by his/her mobile device, as shown in Figure 5. The encrypted personal file saved at the cloud can be accessed and opened by its owner or eligible users through decryption. The downloading time and decryption time are prompted, as shown in Figure 6.

Fig. 6. Access and download a file from cloud server

5.3 Trust Evaluation

Fig. 7. (a) Personal contact list with social trust values; (b) Details of mobile social networking statistics of a contact with its cloud data access link

The mobile client of the CloudFile can automatically evaluate trust of personal contacts based on mobile social networking activities based on SMS communications, mobile calls, social chat, etc. The social trust value of each personal contact is visualized and dynamically updated in a contact list, as shown in Figure 7a. The details social networking statistics of a personal contact are also provided with a link to access the files owned by the contact and stored at the cloud (see Figure 7b). We developed a chatting room (shown in Figure 8a) inside the CloudFile to collect social chatting data (see Figure 8b) because it is hard to extract such data from an existing third party de-

veloped social networking application (e.g., WeChat).

Fig. 8. (a) Chatting UI; (b) Chatting records

Fig. 9. (a) Files of a contact; (b) File access request

Fig. 10. (a) File access key generation and issuing; (b) File access rejection

The CloudFile offers a mobile user to access the files of his contacts that are stored at the cloud (refer to Figure 9a). In this case, the file owner will get an access request (refer to Figure 9b). Based on the owner's access control policy, it issues a secret key to the file requester (Figure 10a) or rejects the request (Figure 10b). The file access with the issued secret key from its owner is shown in Figure 11, which includes the requester gets the secret key from the data owner for encrypted file downloading and decryption.

Fig. 11. Access another user's file from cloud: (a) Get issued secret key; (b) Download file; (c) Download completion with download time and decryption time

5.4 Big File Process

Fig. 12. File splitting process

Mobile applications about network disk access generally restrict the maximum size of user files due to limited phone memory and network bandwidth. For handling big files in practice, the CloudFile offers a function to split and stitch big files to support big file encryption, uploading, downloading and decryption. A big file can be encrypted separately after split it into several small pieces. When a user accesses a big file, he downloads small file pieces, decrypting them and then stitching them to obtain the original one.

By splitting a big file into a plurality of small files, uploading speed is relatively faster than uploading the big file directly. The CloudFile supports big file storage at the cloud using the above method in order to avoid system crash caused by memory leakage, thus improve user experiences. Figure 12 shows the process of file splitting and Figure 13 shows the process of file stitching.

Fig. 13. File stitching process

6 SYSTEM PERFORMANCE TESTS

6.1 Operation Performance

We tested the operation time of the algorithms in our prototype 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. The system performance test includes the following several experiments about the operation time of user setup, KP-ABE key generation, file encryption, file upload, file download, file decryption. The process of file storage at the cloud includes plain file encryption and encrypted file upload. The process of file access from the cloud includes encrypted file download and file decryption. File encryption/decryption time contains file reading time, KP-ABE key encryption/decryption time, and AES encryption/decryption time and file writing time. We tested the operation time with regard to different file sizes: 10MB, 50MB, 100MB, 150MB, 200MB, 250MB, and 300MB (MB: MegaByte). Table 3 lists execution time of important processes in the Cloud-File system.

TABLE 3: EXECUTION TIME OF IMPORTANT PROCESSES

Table 3.1: File uploading and downloading time

File size (MegaByte)	Encryption time (MilliSecond)	Upload time (MilliSecond)	Download time (MilliSecond)	Decryption time (MilliSecond)
10	1513	2647	7491	1142
50	5637	14443	32666	6128
100	14773	31458	73884	10938
150	29559	44595	116677	23969
200	39044	58681	137673	40738
250	46415	68489	147882	52277
300	53588	90448	175354	64002

Table 3.2: File encryption time

File size (MegaByte)	File read time (MilliSecond)	Kpabe encryption time (MilliSecond)	AES encryption time (MilliSecond)	Write file time (MilliSecond)
10	532	47	515	419
50	723	46	2553	2315
100	5366	53	5001	4353
150	8252	47	7661	13599
200	10819	50	9936	18239

250	13223	47	12788	20357
300	15925	51	14167	23445

Table 3.3: File decryption time

File size (MegaByte)	File read time (MilliSecond)	Kpabe decryption time (MilliSecond)	AES decryption time (MilliSecond)	Write file time (MilliSecond)
10	140	21	547	434
50	605	15	2696	2812
100	1268	21	5299	4350
150	4288	15	8033	11633
200	6389	15	11760	22574
250	7699	19	14543	30016
300	9187	19	17137	37659

KP-ABE Public Key Generation (GenerateTrustPK)

User setup generates a masker key and a public KP-ABE key by calling the algorithm GenerateTrustPK, which executes only once if no key revocation happens. Its execution time is lineally increased with the maximum level of trust (I_t), as shown in Figure 14.

Fig. 14. Operation time of KP-ABE public key generation in setup

KP-ABE Secret Key Generation (IssueTrustSK)

Fig. 15. Operation time of KP-ABE secret key generation

For granting a user to access a file stored at the cloud, the file owner issues the KP-ABE secret key to the eligible user who satisfies the access control policy regarding the trust level. The operation time of KP-ABE secret key generation (the algorithm IssueTrustSK) increases lineally with the trust level of the eligible user, as shown in Figure 15.

KP-ABE Encryption and Decryption

KP-ABE encryption and decryption time is not changed with the size of encrypted content (e.g., *DEK*). Refer to Figure 16, the encryption time is about 50ms and decryption time is around 20ms in the prototype system.

Fig. 16. KP-ABE encryption and decryption time

File Encryption

We adopted AES for encrypting and decrypting files in our system (*DEK* key size is 128 bits). The file encryption time with AES is proportional to the size of the file. For a file with 200MB, the AES encryption time is roughly 10 seconds. The entire file encryption process consists of four parts (as shown in Figure 17): AES key encryption with KP-ABE; plaintext file reading (read a file into memory); file encryption with the AES key; output ciphertext of the file and the cipherkey.

Fig. 17. File encryption time

As can be seen from Figure 17, except the encryption time of AES key with KP-ABE, the time of other parts of the entire file encryption process depends on the size of the file. The total time of encryption for a file with 200MB size takes about 38 seconds, in which about 10 seconds are spend to read the file, 10 seconds for AES encryption, and 18 seconds for outputting the ciphertext file. Compared to the above time, the *DEK* encryption time with KP-ABE is only about 50 milliseconds, which does not change with the size of the file and can be almost ignored. Based on this real system test, we can see that our designed scheme based on KP-ABE is very suitable for securing cloud data storage for big data since the size of file has no any impact on the KP-ABE encryption.

File Decryption

The file decryption time with AES also depends on the size of the file. It is proportional to the size of the file. For a file with 200MB, the AES decryption time is roughly 12 seconds. The entire file decryption process consists of four parts (as shown in Figure 18): AES key decryption with KP-ABE; encrypted file reading; file decryption with the AES key; output plaintext.

As can be seen from Figure 18, except the decryption time of AES key with KP-ABE, the time of other parts of the entire file decryption process depends on the size of the file. The total time of decryption for a file with 200MB size takes about 40 seconds, in which about 6.7 seconds are spent to read the file, 11 seconds for AES decryption, and 22 seconds for outputting the plaintext file. Compared to the above time, *DEK* decryption time is very trivial, only about 21 milliseconds, which does not change with the size of the file and can be almost ignored. Again, we can see that our designed scheme based on KP-ABE is very suitable for big cloud data storage and access since the size of file has no any impact on the KP-ABE decryption.

Fig. 18. File decryption time

File Upload and Download

We tested the time of file uploading and downloading based on the prototype system. The results are shown in Figure 19 and Figure 20, respectively. Both uploading and downloading time increases with the size of the file. It takes several minutes if the file is big, e.g., 200MB.

Fig. 19. File uploading time

Fig. 20. File downloading time

The above test results were influenced by the processing capability of a mobile phone and the transmission speed WiFi network.

We observed that AES encryption and decryption time is linearly changed with file size. The time spent for file read, write, upload and download has some fluctuations, which is not strictly linearly changed with the file size. This phenomenon is caused by some external causal factors, such as unexpected network condition variation or other operations executed at the mobile phone during network data transmission. All of above caused a relatively large deviation of testing results.

Based on the above system performance tests, we can see that the CloudFile system is suitable for cloud data storage handled by mobile phones. It also has potential to handle big file storage at the cloud. Note that mobile users often use their mobile phones for mobile social networking and wireless communications. The files saved at the mobile phones are normally not big. For example, a photo taken by a mobile phone has only several megabytes and a video is mostly less than 100 megabytes. Based on our tests, the encryption and decryption of these kinds of multimedia files are very efficient (in second level). The time spent in file uploading and downloading is also reasonable. Thus, the CloudFile can be deployed and used in practice.

6.2 User Acceptance

We further applied Technology Acceptance Model (TAM) to study user acceptance of the CloudFile system. TAM indicates that usefulness, ease of use and playfulness lead to user acceptance [33]. This theory and its extension [34] also show that good interface leads to good perceived usefulness and ease of use; playfulness causes better acceptance (attitude). We designed a questionnaire based on TAM (shown in Table 4) and conducted an interview with it after 20 participants installed the CloudFile at their personal mobile phones and used it for a couple of weeks. We aimed to evaluate the perceived ease of use, perceived usefulness, interface, playfulness and attitude in terms of the CloudFile in order to study its social acceptance. The participants were asked to express their agreement on the questionnaire items described in Table 4 by applying a 5-point Likert scale: very disagree (1), disagree (2), neutral (3), agree (4) and very agree (5).

TABLE 4: USER ACCEPTANCE QUESTIONNAIRE

Purpose	Items
Perceived ease of use (PEU)	Q1: It is easy for me to use CloudFile.
	Q2: CloudFile doesn't influence my usage of other mobile applications.
	Q3: It is very easy to manage my data through CloudFile.
Perceived usefulness (PU)	Q4: CloudFile can store my files as I expect.
	Q5: CloudFile can control other people to access my personal files based on my settings.
	Q6: CloudFile is a useful and valuable application.
Interface (I)	Q7: CloudFile's UI is simple and easy to understand.
	Q8: CloudFile user interface design is good.
	Q9: CloudFile provides a very good user interface for personal file management.
Playfulness (P)	Q10: CloudFile is an interesting application.
	Q11: CloudFile is an exciting application for file management at cloud via my mobile phone.
	Q12: PSNController provides an interesting way to manage my personal data.
Attitude	Q13: I would like to use CloudFile.

(A)	Q14: CloudFile is a cool app.
	Q15: CloudFile offers me a secure and convenient way to store and control my data at cloud.

The 20 participants of the user experiment have different backgrounds. We collected participants' information at the beginning of the experiment regarding their age, gender, occupation, technical background, time of phone usage, mobile phone storage experiences and cloud data storage experiences, shown in Table 5. Among them, ten are female and ten are male, aging between 20 and 30 years old. Each participant owned an Android phone. 17 of them experienced lacking memory storage in their mobile phones. They said they had to delete some files (e.g., photos, videos, and chatting log files) due to memory limitation. Eight participants used cloud storage (e.g., a network disk) in the past. All participants preferred unlimited cloud storage spaces, senseless upload and download time and secure personal data storage at cloud.

TABLE 5: PERSONAL INFORMATION OF PARTICIPANTS

ID	Age	Gender	Occupation	Technical Background	Usage Time of Mobile Phone per Day (hours)	Have you experienced a situation with insufficient phone storage ?	Have you ever used cloud storage via a mobile phone?
1	19	F	Student	Liberal art	5	Y	N
2	21	F	Student	Mathematics	4	Y	Y
3	20	M	Student	Science	4	Y	Y
4	22	M	Student	Engineering	3	Y	N
5	24	F	Student	Law	3	Y	N
6	23	F	Student	Law	3	N	N
7	25	N	Student	Science	3.5	Y	Y
8	24	M	Programmer	Science	2	Y	N
9	27	M	Salesman	Engineering	2.5	Y	N
10	27	M	Programmer	Mathematics	2	N	N
11	28	M	Doctor	Medicine	1.5	Y	N
12	24	F	Administrator	Literature	3.5	Y	N
13	23	F	Student	Mathematics	3	Y	Y
14	28	M	Officer	Liberal art	4.5	N	N
15	27	F	Lawyer	law	2	Y	N
16	30	M	Designer	Liberal art	3.5	Y	Y
17	32	F	Designer	Liberal art	3	Y	Y
18	29	F	Manager	Engineering	3.5	Y	Y
19	26	M	Artist	Art	3	Y	N
20	24	F	Student	Art	4	Y	Y

The interview result is shown in Figure 21 with Standard Deviation (SD), we can see that the CloudFile had satisfactory evaluation scores with regard to perceived ease of use (Q1 to Q3; AVE=4.3, SD=0.40), perceived usefulness (Q4 to Q6; AVE=4.0, SD=0.44), and attitude (Q13 to Q15; AVE=3.7, SD=0.46). User interface (Q7 to Q9; AVE=3.5, SD=0.41) and playfulness (Q10 to Q12; AVE=3.0, SD=0.40) requested further improvement. In particular, perceived ease of use, perceived usefulness and attitude got AVE scores over 3.7. This implied that the CloudFile is a very useful application and it is convenient to use. The participants liked using it with positive attitudes. We also found that the standard deviation is small on each testing item, which implies opinion consistence of the participants. Based on the TAM, we can conclude that the CloudFile prototype system was well accepted by the participants in terms of usefulness and ease of use. But we also notified that the CloudFile's current user interface should be further improved in order to make it more attractive. How to

improve its joyfulness is another interesting future work.

Fig. 21. TAM test result

7 CONCLUSION

In this paper, we proposed the CloudFile: a personal cloud data access control system based on trust evaluation in mobile social networking by applying KP-ABE in order to guarantee the safety of mobile cloud data storage. The CloudFile is one of the first to control cloud data access via mobile devices by using social trust levels as control attributes. We analyzed the performance of the CloudFile with regard to security, computation complexity and communication cost. The prototype implementation and system performance tests conformed theoretical analysis and showed its practical efficiency for securing cloud data with mobile devices. Another advantage of the CloudFile is it can support big file upload to and download from the cloud. User acceptance study further showed its advantages in terms of usefulness and ease of use.

With regard to future work, we are going to further improve and extend the current implementation to make the CloudFile support cloud data access control based on the reputation evaluated by a third trusted party authorized by the data owner when he/she is offline. We also plan to integrate the CloudFile into a commercial social networking application in order to make use of mobile social networks to secure cloud storage services.

ACKNOWLEDGMENT

This work is sponsored by the National Key Foundational Research and Development on Network and Space Security, China (grant 2016YFB0800704), the NSFC (grants 61672410 and U1536202), the PhD grant of the Ministry of Education, China (grant JY0300130104), 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 the Aalto University. The corresponding author is Zheng Yan.

REFERENCES

- [1] R. Chow, P. Golle, M. Jakobsson, E. Shi, J. Staddon, R. Masuoka, J. Molina, "Controlling data in the cloud: outsourcing computation without outsourcing control", *Proc. of the ACM Workshop on Cloud Computing Security*, 2009, pp. 85–90.
- [2] S. Kamara, K. Lauter, "Cryptographic cloud storage", *Proc. of Financial Cryptography and Data Security (FC)*, 2010, pp. 136–149.
- [3] Q. Liu, C. Tan, J. Wu, G. Wang, "Efficient information retrieval for ranked queries in cost-effective cloud environments", *Proc. of INFOCOM*, 2012, pp. 2581–2585.
- [4] M. Kallahalla, et al., "Plutus: Scalable secure file sharing on untrusted storage", *Proc. of the USENIX Conference on File and Storage Technologies (FAST)*, 2003, pp. 29–42.
- [5] E. Goh, H. Shacham, N. Modadugu, D. Boneh, "Sirius: Securing remote untrusted storage", *Proc. of NDSS*, 2003, pp. 131–145.
- [6] J. Bethencourt, A. Sahai, B. Waters, "Ciphertext-policy attribute based encryption", *Proc. of IEEE S&P*, 2007, pp. 321–334.
- [7] V. Goyal, O. Pandey, A. Sahai, B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data", *Proc. of the 13th ACM CCS*, 2006, pp. 89–98.
- [8] S. Muller, S. Katzenbeisser, C. Eckert, "Distributed attribute-based encryption", *Proc. of the 11th Annual Int. Conf. on Information Security and Cryptology*, 2008, pp. 20–36.
- [9] A. Sahai, B. Waters, "Fuzzy identity-based encryption", *Proc. of 24th International Conference on the Theory and Application of Cryptographic Techniques*, 2005, pp. 457–473.
- [10] M. Pirretti, P. Traynor, P. McDaniel, B. Waters, "Secure attribute based systems", *Journal of Computer Security*, vol. 18, no. 5, pp. 799–837, 2010.
- [11] Z. Yan, X. Li, M. Wang, A.V. Vasilakos, "Flexible Data Access Control based on Trust and Reputation in Cloud Computing", *IEEE Transactions on Cloud Computing*, 2015. Doi: 10.1109/TCC.2015.2469662.
- [12] M. Ambrosin, M. Conti, T. Dargahi, "On the feasibility of attribute-based encryption on smartphone devices", *Proc. of IoT-Sys'15*, 2015, pp. 49–54.
- [13] S. Yu, C. Wang, K. Ren, W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing", *Proc. of the IEEE INFOCOM*, 2010, pp. 534–542.
- [14] G. Wang, Q. Liu, J. Wu, M. Guo, "Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers", *Computers & Security*, vol. 30, no. 5, pp. 320–331, 2011.
- [15] S. Yu, C. Wang, K. Ren, W. Lou, "Attribute based data sharing with attribute revocation", *Proc. of the ACM ASIACCS*, 2010, pp. 261–270.
- [16] Z. Yan (ed.), *Trust Modeling and Management in Digital Environments: from Social Concept to System Development*, IGI Global, Hershey, Pennsylvania, 2010.
- [17] K. Yang, X. Jia, K. Ren, B. Zhang, R. Xie, "DAC-MACS: Effective Data Access Control for Multiauthority Cloud Storage Systems", *IEEE Trans. on Information Forensics and Security*, vol. 8, no. 11, pp. 1790–1801, 2013.
- [18] G. Wang, Q. Liu, J. Wu, "Hierarchical attribute-based encryption for fine-grained access control in cloud storage services", *Proc. of the 17th ACM CCS*, 2010, pp. 735–737.
- [19] M. Zhou, Y. Mu, W. Susilo, J. Yan, "Piracy-preserved access control for cloud computing", *Proc. of IEEE TrustCom11*, 2011, pp. 83–90.
- [20] G. Lin, D. Wang, Y. Bie, M. Lei, "MTBAC: a mutual trust based access control model in cloud computing", *China Communications*, vol. 11, no. 4, pp. 154 – 162, 2014.
- [21] Z. Yan, M. Wang, V. Niemi, R. Kantola, "Secure pervasive social networking based on multi-dimensional trust levels", *IEEE CNS 2013*, 2013, pp.100–108.
- [22] Z. Wan, J. Liu, R.H. Deng, "HASBE: a hierarchical attribute-based solution for flexible and scalable access control in cloud computing", *IEEE Trans. on Info. Forensics & Security*, vol. 7, no. 2, pp. 743–754, 2012.
- [23] Z. Yan, Y. Chen, Y. Shen, "A practical reputation system for pervasive social chatting", *Computer and System Sciences*, vol. 79, no. 5, pp. 556–572, 2013.
- [24] Z. Yan, *Trust Management in Mobile Environments – Usable and Autonomic Models*, IGI Global, Hershey, Pennsylvania, 2013.
- [25] Y. Tang, P.P.C. Lee, J.C.S. Lui, R. Perlman, "Secure overlay cloud storage with access control and assured deletion," *IEEE Trans. on Dependable and Secure Computing*, vol. 9, no. 6, pp. 903–916, 2012.
- [26] Z. Yan, X. Li, R. Kantola, "Personal data access based on trust assessment in mobile social networking", in *Proc. of IEEE TrustCom2014*, 2014, pp. 989 – 994.
- [27] Z. Yan, X. Li, R. Kantola, "Controlling cloud data access based on reputation", *Mobile Networks and Applications*, vol. 20, No. 6, pp. 828–839, December 2015.
- [28] L. Zhou, V. Varadharajan, M. Hitchens, "Achieving secure role-based access control on encrypted data in cloud storage", *IEEE Trans. on Information Forensics and Security*, vol. 8, no. 12, pp. 1947–1960, 2013.
- [29] W. Wang, J. Han, M. Song, X. Wang, "The design of a trust and role based access control model in cloud computing", in *Proc. of 6th International Conference on Pervasive Computing and Applications*, 2011, pp. 300–334.
- [30] T. Zhu, W. Liu, J. Song, "An efficient role based access control system for cloud computing", *Proc. of IEEE CIT2011*, 2011, pp. 97–102.
- [31] S. Yang, P. Lai, J. Lin, "Design role-based multi-tenancy access control scheme for cloud services", *Proc. of International Symposium on Biometrics and Security Technologies*, 2013, pp. 273–279.
- [32] Pairing-Based Cryptography (PBC) library. Available: <http://crypto.stanford.edu/pbc/>

- [33] F. D. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology", *MIS Quarterly*, 13(3), 1989, pp. 319-340.
- [34] V. Venkatesh, H. Bala, "Technology acceptance model 3 and a research agenda on interventions", *Decision Sciences*, 39(2), 2008, pp. 273-315.

Zheng Yan received the BEng degree in electrical engineering and the MEng degree in computer science and engineering from the Xi'an Jiaotong University, Xi'an, China in 1994 and 1997, respectively, the second MEng degree in information security from the National University of Singapore, Singapore in 2000, and the licentiate of science and the doctor of science in technology in electrical engineering from Helsinki University of Technology, Helsinki, Finland in 2005 and 2007. She is currently a professor at the Xidian University, Xi'an, China and a visiting professor at the Aalto University, Espoo, Finland. She authored more than 140 publications and solely authored two books. She is the inventor and co-inventor of 48 patents and patent applications. Her research interests are in trust, security and privacy, social networking, cloud computing, networking systems, and data mining. Prof. Yan serves as an editor/guest editor, an organization and program committee member for numerous reputable journals, conferences and workshops. She is a senior member of the IEEE.

Wangyang Shi received his B.Eng. degree in telecommunication engineering from Xi'an University of Posts and Telecommunications, Xi'an, China in 2012. He got his M.Eng degree in information security from the School of Telecommunications Engineering, Xidian University in 2016. His research interests are in cloud security, social networking and gaming.