A Hybrid Encryption Scheme with Key-cloning Protection: User / Terminal Double Authentication via Attributes and Fingerprints

Chunlu Chen[1,2*], Hiroaki Anada[2], Junpei Kawamoto[1,2], and Kouichi Sakurai[1,2]
[1]Kyushu University, Fukuoka, Japan
{chunlu.chen, kawamoto, sakurai}@inf.kyushu-u.ac.jp
[2] Institute of Systems, Information Technologies and Nanotechnologies, Fukuoka, Japan
anada@isit.or.jp

Abstract

Internet service has enabled digital contents to be shared faster and easier, but on the other side it raised an issue of illegal copy of the digital contents. Public key encryption schemes solve this issue partially. However, there is still a weak point that the secret key is not completely protected; that is, public key encryption schemes suffer from illegal copy of secret keys (the key-cloning problem). In this paper, first, we discuss the usability of terminal fingerprints for key-cloning protection. Next, we propose a hybrid encryption scheme using terminal fingerprints to protect the secret keys from the key-cloning. Based on an assumption that the terminal fingerprint is unchangeable and unextractable even by the user of the terminal, our hybrid encryption scheme can be effectively used as a method of the key-cloning protection. Then, we instantiate our hybrid encryption scheme as a combination of the attribute-based encryption scheme and the RSA encryption scheme; the attribute-based encryption scheme functions as a mechanism for authentication of user's attributes, and the RSA encryption scheme functions as a mechanism for authentication of a terminal device. Terminal fingerprint that is a feature of the device is used to generate a secret key of the RSA encryption.

Keywords: Key misuse, Terminal fingerprint, Re-encryption

1 Introduction

As the rapid development of information construction and with the application of information network technology growing popularity, information security has become important issues affecting the security and the efficiency of the network. On one hand, network information technology makes the world wide range of information exchanges convenient increasingly and fast. At the same time, it brings more opportunities on business and scientific research. However, once the important information, (such as national secrets, commercial secrets or personal privacy) is intercepted or tampered with, the state, enterprises or individuals will suffer huge losses. In addition, illegal invasion and illegal profits for the purpose of information crime are increasing. It also brings challenges to the safe operation and further development of the network. Such as information leakage, information theft, data tampering, data addition, computer virus etc. Usually, for the computer crime, it is difficult to leave evidence of the crime. This was also greatly stimulated the occurrence of High-tech computer crime cases. With the rapid increase of computer crime, the computer system especially the network system, is facing great threat and becoming one of the serious social problems.

In recent years, the storing data in the cloud serves has been incseasing more and more with the rapid growth of the Internet in order to reduce the cost of using local storage and data sharing. However, information disclosure and trust issues arise in third party management cloud servers. Therefore, improving

Journal of Internet Services and Information Security (JISIS), volume: 6, number: 2 (May 2016), pp. 23-36
*Corresponding author: Kyushu University, Fukuoka, Japan, Tel: +81-92-802-3639

the security of stored data is becoming a critical task. Typically, data stored in the cloud must be encrypted to ensure the data security. Public-key cryptography is one of the methods to encrypt data, which uses a pair of keys, a secret key (SK) and a public key (PK). Although, it can guarantee the security and can help us to protect the message, the complexity of key management is a big issue. And it can also be used to make illegal acts, such as transferring and copying of unauthorized secret key. Secret Key can be copied from other users illegally but it is difficult to identify the responsible entity and the source if the secret key leaks. Although there are various methods to utilize for preventing the secret key generation, the leakage of secret key is still a weak point.

1.1 Related Work

At this stage, authentication for the protection of user security can generally be divided into the following ways. The first way is to verify physical layer aspects, which is to determine whether the user is a legitimate user through the hardware of user's terminal. For example chip verification, or USB key verification etc. A typical example has been studied as PUF. In this approach, the terminal hardware utilization ensures the uniqueness of the secret key. And by doing so, if prevents secret key from copy and lost. But it has some problems. The first problem is that the update will increase the cost of overall system and the loss of terminal means the secret key will be permanently lost as well. The second way is by a variety of application-layer encryption like Biometric authentication or PKI/CA applications. In recent years, many researchers have been studying on biometric authentication. It can reveal the secrets of the human body and at it can also let us use it in our daily lives as well. This technology greatly improves the facilitation of human's social life, but we need to pay attention to how to protect "confidentia" information in biometric authentication system. And, PKI/CA technology has long been proposed and utilized as an authentication method and it is well-known and has been widely utilized in Digital rights management (DRM). But the decryption requires sessions to obtain authorization, and the CA must be a trusted third party. The following discusses some of the existing technologies in detail. Three related technologies are introduced as follows.

Hardware Certification

Physical Unclonable Function (PUF) achieved by a physical device using differential extraction of the chip manufacturing process inevitably leads to generate an infinite number, unique and unpredictable "secret key"[19]. These secret keys are randomly generated. PUF system uses the password / response mechanism to authentication. After PUF system receives a random 64-bit code, it will generate a unique random 64 (or longer) code as a response. Due to the differences in chip manufacturing process, even the chips manufactures cannot replicate the same set of passwords on another chip. Therefore, PUF technology makes chip to have the function of anti-imitation.

Since none of the two PUF circuits' delay characteristic are identical, the transmission speed of the signal is not same. Thus two signals pass the PUF at different time too. At the end of the PUF, an arbiter is placed to determine which signal arrives first and determine the output "1" or "0". The same input for two PUF may produce different output. At the same time, since the input signal determines the signal transmission path in the PUF, different inputs will produce different outputs.

Kumar et al.[13] designed a system, where PUF output defines and gives a certain input, while other PUFs produce different outputs. According to the uniqueness of this chip output, it can be widely utilized in smart cards, bank cards and so on. In this way, we can protect message through the uniqueness of the secret key from copying and other doing illegal activities.

Biometric authentication

Biometric technology consists of using computers and optics, acoustics, biosensors and other high-tech tools to retrieve the body's natural physiological characteristics (such as a fingerprint, finger vein, face, iris, etc.) and behavioral characteristics (e.g. handwriting, voice, gait, etc.) to identify personal identity. Biometric technology is not easy to forget, good security performance, and not copy or stolen "portable" and can be used anywhere[10]. Furthermore, biometric can be used as a unique, unalterable secret key but the safety is still taken seriously.

Jain, Anil et al.[11] analyzed and evaluated these biometric authentication systems. Moreover, biometric authentication is also used in various fields, for example, Uludag et al.[20] proposed the biometric authentication, which can be used to construct a digital rights management system.

In fact, in the present life, the biometric authentication has been very widely utilized, such as bank card fingerprint authentication, and face authentication in customs. Although biometrics brought us convenience, biometrics privacy protection has become an important research challenge.

PKI/CA

Digital certificate, as the core of PKI / CA technology, can encrypt, decrypt, digital sign and signature verify online transmitted information. Therefore, it ensures that the information cannot be accessed except for the sending and receiving parties and cannot be tampered during transmission. The recipient can confirm the identity of the sender via digital certificate and the senders cannot deny their information.

PKI is also used for various occasions in our lives. We can use PKI when we want to use "identity data," "complete transactions," "transaction undeniable" or "confidential". The PKI application goes beyond these. Those popular service, such as online banking and online shopping, also rely on the PKI mechanism and legal status given by digital signature, in order to ensure the existence and identity of transaction parties to the transaction record to confirm. The interests of business and consumers are to be protected, as well as using the internet to achieve the security purpose of transmitting information. PKI applications will come into contact with daily life and also include a network filing, automated highway toll smart cards, e-mail encryption signature, internet shopping and building access control systems.

1.2 Challenging Issue

There are a lot of works detailing the knowledge in this area [1, 12, 17], but in those works a trusted third party (mostly by national regulators or major well-known companies) are formed and released. However, in the small network groups, the trusted third party is difficult to achieve. So, there is still a need for ensuring security of systems without the use of third-party supervision. Here we note that the hardware-based authentication and Biometric authentication mentioned above ensure the uniqueness of the key. But these still cannot guarantee the safety (and hence, security) of keys. The update of hardware-based authentication requires the replacement of the hardware itself, which increases the cost. Biometric authentication is impossible to alter but it is possible to be copied.

1.3 Our Contribution

Our first contribution is to focus on terminal fingerprints and use it to realize key-cloning protection. Here we will use the terminology a terminal fingerprint as a set of feature points of a terminal with unchangeability and unextractability. We also require that a terminal fingerprint should possess universal composability with other fingerprints. For example, the screen resolution, network environment, memory, CPU are different to each terminal. These information can be used as the feature points to identify the terminal, and hence they can be considered to be a terminal fingerprint of the terminal. In contrast, the various sets of features possessed by browser is called browser fingerprints [7, 3, 8]. We can say that the difference between our terminal fingerprints and browser fingerprints lies in the discrepancy that

browser fingerprints (such as type of fonts, instaled plug-ins and etc.) are changeable and difficult to control.

Our second contribution is to propose a hybrid encryption scheme that consists of two public key encryption schemes, PKE1 and PKE2. As an instance, PKE1 can be ciphertext-policy attribute-based encryption (CP-ABE) scheme, which is for decryption by user with authenticated attributes. On the other hand, PKE2 can be the RSA scheme, which is for decryption by authenticat terminal. The virtue to utilize the terminal fingerprint is that every terminal fingerprint is different for an attacker. Even if attacker launches a collusion attack, it still cannot be decoded. Hence, in the proposed scheme, the terminal fingerprint of the user can be utilized as a secret key, and it never leak outside even once, hence the security of the secret key is guaranteed. Thus, safety of the secret key is increased in this way. Therefore, our hybrid encryption scheme guarantees the secret key to be unclonable, and cannot be used by illegal user.

In this paper, the above contributions are discussed and stated in detail compared with Chen et al.[5].

1.4 Comparison with Existing Work

The terminal fingerprint we selected is different for each terminals. Saito et al.[18] proposed Current Status and Issues of Web Browser Fingerprinting. In this paper the case of mobile compared with the PC, there are some fingerprints that cannot be taken (for example flash). Its lower unique for each feature point. To this weakness, the combining of 21 feature points are obtained by the mobile fingerprinted data, and it can improve the unique rate up to 96.5%. This can ensure that the terminal fingerprint we used is unique and cannot be copied. We can choose terminal information as a secret key which we want. For example, we can choose not only browser information as secret key for onetime session also hardware information as secret key for multiple sessions. This way can be more convenient to use, according to the environment, situation and user's needs to create the secret key.

And for this study, we propose a hybrid encryption scheme that needs to re-encrypt ciphertext. And the process of encryption uses terminal fingerprint information. Our scheme can support any public key encryption scheme. However, in our scheme we will use Waters' CP-ABE scheme[2] first. This scheme is not only using the terminal fingerprint information to generate unique secret key, but also updates itself according to user settings with relatively low cost to keep the freshness of the terminal fingerprint information.

1.5 Organization of This Paper

The rest of this paper is structured as follows. Section 2 introduces background information, formal definitions. Section 3 introduces CP-ABE system model and our system model. Section 4 describes our encryption scheme. Section 5 discusses the security and advantage of the proposed scheme. Finally, conclusion and future work in section 6.

2 Preliminaries

In this section, we give background information on bilinear maps and our cryptographic assumption.

2.1 Bilinear Maps

We present a few facts related to groups with efficiently computable bilinear maps. Let G_1 and G_2 be two multiplicative cyclic groups of prime order p. Let g be a generator of G_1 and e be a bilinear map $e : G_1 \times G_1 \to G_2$. The bilinear map e has the following properties:

1. Bilinearity: for all u, $v \in G_1$ and, $a, b \in Z_p$, we have $e\left(U^a, V^b\right) = e(U,V)^{ab}$,

2. Non-degeneracy: $e(g,g) \neq 1$.

2.2 Access Structure and Linear Secret Sharing Scheme

We will review here the definition of access structure and Linear Secret Sharing Schemes (LSSS)[21].

Definition 1 (Access Structure) *Let $P = (P_1, P_2, ..., P_n)$ be a set of attributes. A collection $\Gamma \subset 2^P$ is said to be monotone if Γ is closed under superset, i.e. if $\forall B, C$ if $B \in \Gamma$ and $B \subset C$, then $C \in \Gamma$. An access structure $(respectively, monotoneassessstructures)$ is a collection $(respectively, monotonecollection)$ Γ of nonempty subsets of P, i.e., $\Gamma \subset 2^P \backslash \{\phi\}$. The members of Γ are called authorized sets, and the sets not in Γ are called unauthorized sets.*

Definition 2 (Linear Secret Sharing Schemes (LSSS)[21]) *A secret-sharing scheme Π over a set of parties P is called linear (over Z_p) if*

1. The shares for each party form a vector over Z_p,

2. There exists a matri $\times M$ with ℓ rows and n columns called the share-generating matrix for Π. For all $i = 1, ..., \ell$, the i-th row of M, we let the function ρ defined the party labeling row i as $\rho(i)$. When we consider the column vector $\gamma = \{s, r_2, ..., r_n\}$, where $s \in Z_p$ is the secret to be shared, and $r_2, ..., r_n \in Z_p$ are randomly chosen, then M_γ is the vector of ℓ share of the secret s according to Π. The share $\left(M_\gamma\right)_i$ belongs to party $\rho(i)$.

Here the Π is a Linear Secret Sharing Schemes(LSSS) composed of Γ. Let s be any attribute set of authenticated user, and define $I \subset \{1, 2, ..., \ell\}$ as $\{i; \rho(i) \in S\}$. For Π , there exist a structure $\{\omega_i \in Z_p\}$ that if $\{\lambda_i\}$ are valid shares of any secret s, than $\sum_{i \in I} \omega_i \lambda_i = s$.

3 System Model

In this section, we will review the the properties of CP-ABE scheme, Naruse's proposed re-encryption scheme and propose our hybrid encryption scheme.

3.1 CP-ABE

There are a lot of studies on enhance the security of system. Cheung and Newport[6] proposed CP-ABE scheme based on DBDH problem using the CHK techniques[4], which satisfies IND-CPA secure and pioneers the achievement of IND-CCA secure. In this method, a user's secret key is generated by calculating user attributes and system attributes. Li et al.[14] proposed an encryption system using trusted third party, who issues authentication information embed user key to achieve better safety in decryption phase than CP-ABE. However, it is difficult to implement due to the complexity of the computational process required from the third party. Finally, Li et al.[15] proposed encryption scheme crowded includes the ID of the user attribute, decrypts it when ID authentication is also carried out at the same time. Although this scheme can improve the safety, the public key distribution center will increase the workload. Hinek et al.[9] proposed a tk-ABE(token-based attribute-based encryption) scheme that includes a token server to issue a token for a user to decrypt the cipher text, thus making the key cloning meaningless.

Our proposal scheme aims to increase the safety of the secret key without third party. When the cipher text corresponds to an access structure and secret key corresponds to a set of attributes. Only if the attributes in the set of attributes is able to fulfill the access structure.

An (Ciphertext-policy) Attribute Based Encryption scheme consists of four fundamental algorithms: Setup, Encrypt, KeyGen, and Decrypt.

Setup (λ, U) $\rightarrow$ (PK, MK) : The Setup algorithm takes security parameter λ and an attribute universe U as input. It outputs the public parameter PK and the system master secret key MK.

Encrypt (PK, M, W) $\rightarrow$ CT : The Encrypt algorithm takes the public parameter PK, a message M, and an access structure was input. It output a cipher text CT.

KeyGen (MK, S) $\rightarrow$ SK : The KeyGen algorithm takes the master secret key MK and a set S of attributes as input. It output a secret key SK.

Decrypt (CT, SK) $\rightarrow$ M : The Decrypt algorithm takes as input the cipher text CT and the secret key SK. If the set S of attributes satisfies the access structure W then the system will output the message M.

3.2 CP-ABE Scheme with Re-Encryption

Naruse et al.[16] proposed a new CP-ABE mechanism with re-encryption. Their method is based on the CP-ABE scheme to make the cipher text and has re-encryption phase to protect the message. Although this scheme is used to achieve an updated and relocated mechanism, but it can increase the calculation of system because the updated or withdrawal is needed to calculate the re-ciphertext and secret key.

We will review here the Naruse's scheme which consists of five fundamental algorithms: Setup, Ext, Enc, ReEnc and Dec.

Setup (λ) $\rightarrow$ (PK, MK, RK) : The Setup algorithm takes security parameter λ as input. It outputs the public parameter PK, the system master secret key MK and the re-encrypt key RK.

Ext (MK, S) $\rightarrow$ SK : The Ext algorithm takes the master secret key MK and user's attributes set S as input. It output a secret key SK.

Enc (PK, M, W) $\rightarrow$ CT' : The Enc algorithm takes the public parameter PK, a message M, and an access structure was input. It output a cipher text CT'.

ReEnc (RK, CT', S) $\rightarrow$ CT : The ReEnc algorithm takes the re-encrypt key RK, cipher text CT' and user's attributes set S as input. It output a cipher text CT.

Dec (CT, SK) $\rightarrow$ M : The Dec algorithm takes as input the cipher text CT and the secret key SK. If the set S of attributes satisfies the access structure W then the system will output the message M.

3.3 Our Hybrid Encryption Scheme

In this section, we propose an attribute-based encryption scheme without key misuse.
Our system consists of three parts:

– User needs to provide their attributes information and legitimate manner to use the content. They also need to manage the terminal fingerprint that their own;

– Data server needs to manage the attribute information, a common key and public parameter PK and issue the secret key that contains the attribute information of the user;

– Document sender needs to issue the common key and encrypt the contents.

We propose a hybrid encryption scheme HybENC that uses terminal fingerprint. HybENC consists of a common-key encryption scheme, CKE, two public key encryption schemes, PKE1 and PKE2, and a hash function, H : HybENC = (CKE, PKE1, PKE2, H). Informally, CKE is used for fast encryption and decryption of data of large size such as pictures and movies. PKE1 is used to encrypt the common key of CKE. Later, PKE1 will be replaced with an attribute-based encryption. Finally, PKE2 is used to re-encrypt the common key of CKE; fingerprint is used here as the secret key of PKE2 through a hash function.

Formally, our HybENC is described as follows.

HybENC.Key $(\lambda) \rightarrow$ **FK, (PK1, SK1), (PK2, SK2)**: The HybENC.Key algorithm takes a security parameter λ as input. It calculates keys as follows;
CKE.Key (λ) $\rightarrow$ FK, PKE1.Key (λ) $\rightarrow$ (PK1, SK1), H_λ (fingerprint) $\rightarrow$ SK2, PKE2.Key (SK2) $\rightarrow$ PK2. Then it outputs keys; FK, (PK1, SK1), (PK2, SK2).

HybENC.Enc (FK, PK1, PK2, m) $\rightarrow$ **CT, CT2**: The HybENC.Enc algorithm takes keys FK, PK1, PK2 and a plaintext m as input. It calculates cipher texts as follows;
CKE.Enc (FK, m) $\rightarrow$ CT,PKE1.Enc (PK1, m1 := FK) $\rightarrow$ CT1, PKE2.Enc (PK2, m2 := CT1) $\rightarrow$ CT2. Then it outputs cipher texts; CT, CT2.

HybENC.Dec (FK, SK1, SK2, CT, CT2) $\rightarrow$ **m**: The HybENC.Dec algorithm takes keys FK, SK1, SK2 and cipher texts CT, CT1, CT2 as input. It executes decryption as follows;
PKE2.Dec (SK2, CT2) $\rightarrow$ m2 = CT1, PKE1.Dec (SK1, CT1) $\rightarrow$ m1 = FK, CKE.Dec (FK, CT) $\rightarrow$ m. Then it outputs the decryption result m.

4 Our Construction

We apply the above template of our hybrid encryption scheme to a scheme in the attribute-based setting. Plaintext is encrypted by using the attribute information and terminal fingerprint. The advantages of this scheme, confirmation of the terminal fingerprint is difficult to use except by authorized users.

We now give our construction by employing Water's CP-ABE as PKE1 in our hybrid encryption in Section 3.3.

In our construction the set of users is $U = \{1, 2, ..., n\}$ and the attribute universe is $A = \{1, 2, ..., \ell\}$. A random exponent for encryption is denoted as $s \in Z_p$. Note that secret keys below are randomized to avoid collusion attacks.

C.Setup $(v, w) \rightarrow$ **FK :** The DO.Setup algorithm will choose a prime order p with generator q in the system. Next it will choose two random exponents $v, w \in Z_p$ as input. The common key is published by the Diffie-Hellman key exchange

$$FK = (q^v)^w \ mod \ p = (q^w)^v \ mod \ p$$

C.Enc (FK, m) $\rightarrow$ **CT :** The common-key encryption, C.Enc algorithm takes FK and a plaintext m as input. It outputs a ciphertext CT.

Auth.Setup $(\lambda) \rightarrow$ **PK1, MK :** The Auth.Setup algorithm will choose a bilinear group G_1 of prime order p with generator g, and e be a bilinear map, $e : G_1 \times G_1 \rightarrow G_2$. It then chooses two random exponents $a, b \in Z_p$ and hash function $H : \{0,1\}^* \rightarrow G$ as input. The Common key is published as

$$PK1 = g, g^b, e(g,g)^a$$

The system master secret key is published as

$$MK = g^a$$

Auth.Ext (MK, S) → SK1 : The Auth.Ext algorithm takes the master secret key MK and a set of attributes S as input. And algorithm chooses a random $t \in Z_p$ for each user. It creates the secret key as

$$SK1 = \left(g^{a+bt}, g^t, (K_X)_{X\in S}\right), \quad \forall_{X\in S}\, K_X = H(X)^*$$

U.Setup (f) → PK2, SK2 : The U.Setup algorithm takes user's fingerprint information f. Then it calculates the hash value $H(f) = D$(in this paper we use the RSA encryption for our re-encryption). It chooses two primes p,q. Make N = pq. Next it computes E s.t. $DE \equiv 1\ mod\ (p-1)(q-1)$. The user's terminal-fingerprint $F = (N,E)$ as public key PK2. The user keeps D as the user's terminal-fingerprint secret key SK2.

Auth.Enc (PK1, FK, W) → CT1 : The Auth.Enc algorithm takes the public parameter PK1, common key FK, and an LSSS access structure (W,ρ) over the all of attributes to encrypts the common key FK. The function ρ associates row of W to attributes.

Where W is an $\ell \times n$ matrix. First the algorithm generates a vector $\gamma = (s, y_2, ..., y_n) \in Z_p^n$ and $r_1, r_2, ..., r_\ell \in Z_p$ randomly. The vector is made for sharing the encryption exponent s. then W_i is the vector corresponding to the i-th row of W, calculates $\lambda_i = \gamma W_i$ from 1 to ℓ.
It output a ciphertext FT as

$$CT1 = (FKe(g,g)^{as},\ g^s,\ \widehat{Cs}),$$
$$\widehat{Cs} = (g^{b\lambda_1}H(X_{\rho_1})^{r_1}, g^{r_1}), (g^{b\lambda_2}H(X_{\rho_2})^{r_2}, g^{r_2}), ...(g^{b\lambda_\ell}H(X_{\rho_\ell})^{r_\ell}, g^{r_\ell}).$$

Auth.ReEnc (CT1, PK2) → CT2 : The Auth.ReEnc algorithm takes the cipher text CT1 and user's terminal-fingerprint public key PK2 as input.
The re-cipher text is published as

$$CT2 = (CT1)^E\ mod\ N$$
$$Where\ (CT1)^E = (FKe(g,g)^{asE}, g^{sE}, (\widehat{Cs})^E)$$

U.Dec (CT2, SK2) → CT1 : The U.Dec algorithm takes as input the cipher text CT2 and SK2. The decryption algorithm first computes.
The decryption algorithm computes

$$(CT2)^D = \left(CT1^E\right)^D = CT1\ mod\ N$$

U.ReDec (CT1, SK1) → FK : The U.ReDec algorithm takes the cipher text CT1 and secret key SK1 as input. The secret key for an attribute set S, and the cipher text FT for LSSS access structure (W,ρ). Suppose that S satisfies the access structure and define $I \subset \{1,2,...,\ell\}$ as $\{i; \rho(i) \in S\}$. For Π , there exist a structure $\{\omega_i \in Z_p\}$ that if $\{\lambda_i\}$ are valid shares of any secret s, than $\sum_{i\in I}\omega_i\lambda_i = s$. The U.ReDec algorithm will output the common key FK.
The re-decryption algorithm computes

$$\frac{e(g^s, g^{a+bt})}{\prod_{i\in I}(e(g^{b\lambda_i}H(X_{\rho_i})^{r_i}, g^t)e(H(X_{\rho_i})^t, g^{r_i}))} = \frac{e(g,g)^{as}e(g,g)^{bts}}{\prod_{i\in I}e(g,g)^{bt\omega_i\lambda_i}} = e(g,g)^{as}$$

$$\frac{FKe(g,g)^{as}}{e(g,g)^{as}} = FK$$

C.Dec (FK, CT) → m : The C.Dec algorithm takes the common key FK and the cipher text CT as input. It output the message m.

5 Analysis

In this paper, the confidentiality of the shared data has been encrypted and protected. It is almost impossible for the secret key to leak. Because the ABE scheme for the chosen plaintext attack is safe, our scheme based on the ABE scheme is safe too. Even if the encrypted data is published, the scheme can withstand the attacks from colluding users. The attacker can not get the secret key because they cannot get the legitimate user's terminal fingerprint information.

5.1 Security Analysis

This study shows that confidentiality of the shared data that has been encrypted can be protected and it is difficult to reveal the secret keys in the proposed scheme. The proposed scheme is secure against chosen-plaintext attacks because the underlying ABE scheme is secure against chosen-plaintext attack. So in this section, we will discuss the security model based on CP-ABE. If the encrypted data is published, our scheme also resists attacks from colluding users. If the attacker did not know the terminal fingerprint of the legitimate user, they wouldn't be able to get the secret key.

We analyze the security of the RSA scheme. The system chooses two primes p,q randomly, and make $N = pq$ and $\varphi(N) = \varphi(p)\varphi(q) = (p–1)(q–1)$. Choose integer E such that $1 < E < \varphi(N)$ and $\gcd(e,\varphi(N)) = 1$. Next it computes D s.t.$DE \equiv 1 \bmod (p-1)(q-1)$. D is the modular multiplicative inverse of $E(\bmod (p-1)(q-1))$. To obtain D, the easiest way is to decompose the N into p and q, but for a long time no one has found a polynomial time algorithm to decompose a large integer factor. At the same time, no one has been able to prove that the algorithm does not exist. Until now no one has been able to prove that factor decomposition of N is the only way to derive C from M. Thus, today it is generally believed that as long as the N is large enough, then the scheme is safe.

Now we give the security proof of our porposed scheme.

Theorem 1: *If DBDH problem is Intractable, then our scheme is safe in the following attack model.*

Proof: Provided a polynomial of the adversary A exists, In the AB-sSet model, it can take advantage of ε to break our scheme. Construct a simulator B. It takes advantage of $\varepsilon' = (1/2)\varepsilon$ to complete the attack game of DBDH. The simulation procedure is as follows:

The simulator is carried out according to the following: the challenger set group G_1, G_2 and a highly efficient bilinear map e, g be a generator of G_1. The Challenger throws a coin privately. The value is assigned to u. If $u = 0$, challenger makes $(g, A = g^a, B = g^b, C = g^c, Z = e(g,g)^{abc})$, otherwise $(g, A = g^a, B = g^b, C = g^c, Z = e(g,g)^z)$, and a,b,c,z are random.

Init: Simulator B running adversary A. The adversary A choose he wants to challenge the set of attributes s, random number E and informs the Simulator B.

Setup: Simulator B specifies common parameters $g_1 = A, g_2 = B$. Then choose random n-th-order polynomial $f(x)$. And compute the n-th-order polynomial $u(x) = -x^n$, and $x \in s$. For other $x, u(x) \neq -x^n$. Such structure ensures the $u(x) = -x^n$ for arbitrary $x \in s$.

Next, the simulator set $t_i = g_2^{u(i)}\, g^{f(i)}$, $i \in \{1, ..., n+1\}$. Because the $f(x)$ is an n-th-order polynomial, so all of the t_i are random and independent. And $T_i = g_2^{i^{n+u(i)}} g^{f(i)}$.

Phase1: Perform the following operation.

Enc: This phase adversary A can adaptively select some attribute set s to carry out the secret key generation query, it also can adaptively select some ciphertext CT' to query the decryption. First, the attribute

set s of adversary query must satisfy the $T'(s) = 0$, then the simulator will return the secret key. The specific operation of the simulator is as follows:

First, for every node in access tree choose a d_x-th-order polynomial q_x. And set the following equation $y = q_r(0) = a$. Then, let $i = att(x)$, the private key corresponding to each leaf node is calculated using the following operation:

–When $i \in s$, random number $r_x \in Z_p$,

$$D_x = g_2^{q_x(0)} T(i)^{r_x}; R_x = g^{r_x}$$

–When $i \notin s$,$i^n + u(i) \neq 0$, let $g_3 = g^{q_x 0}$,

$$D_x = g_3^{\frac{-f(i)}{i^{n+u(i)}}} (g_2^{i^{n+u(i)}} g^{f(i)}); R_x = g_3^{\frac{-1}{i^{n+u(i)}}} g^{r_x{}'}$$

Let $r_x = r_x{}' - \frac{q_x 0}{i^{n+u(i)}}$, then through the computation

$$D_x = g_2^{q_x 0} (g_2^{i^{n+u(i)}} g^{f(i)})^{r_x{}' - \frac{q_x 0}{i^{n+u(i)}}} = g_2^{q_x 0} T(i)^{r_x}$$

$$R_x = g_3^{\frac{-1}{i^{n+u(i)}}} g^{r_x{}'} = g^{r_x{}' - \frac{q_x 0}{i^{n+u(i)}}} = g^{r_x}$$

Therefore, the simulator can construct the secret key according to the access tree, and the secret key is same with the secret key created by the original scheme, then adversary A can not be distinguished.

Re-Enc: The adversary A is to query the ciphertext CT^* to simulator B. Simulator B utilizes the random number E to encrypt the ciphertext CT^* and sends ciphertext CT^* to adversary A.

Challenge: The adversary A submits two messages m_0, m_1 of same length. Then the simulator random selectes one of them $m_v (v \in \{0,1\})$, and use's attribute set s and random number E to encrypt. Then output Re-Ciphertext as

$$CT = (CT')^E; CT' = (s, CT^* = m_v \cdot Z, CT^{**} = g^c, \{E_i = C^{f(i)}\}_{i \in s})$$

–If $u = 0$, than $Z = e(g,g)^{abc}$, than output legitimate ciphertext as

$$CT = (CT')^E; CT' = (s, CT^* = m_v \cdot e(g,g)^{abc}, CT^{**} = g^c, \{E_i = T(i)^c\}_{i \in s})$$

–If $u = 1$, than $Z = e(g,g)^z, CT^* = m_v \cdot e(g,g)^z$,

Since Z is random, in the adversary A's view CT^* is a random value that does not contain any useful information.

Phase2: Repeated phase1 of the operation.

Guess: Adversary A output guesses $v' \in \{0,1\}$. Then the simulator output guess u' based on adversary A's guess. If $v' = v$, then simulator output $u' = 0$, the simulator determines that the BDH- tuple is challenged. Otherwise, the simulator output $u' = 1$, it is challenged by random 4 tuples.

–When $\mu = 1$, the correct probability by simulator guess is $Pr[u' = u | u = 1] = \frac{1}{2}$

–When $\mu = 0$, the correct probability by simulator guess is $Pr[u' = u | u = 0] = \frac{1}{2} + \varepsilon$ The probability of the simulator to solve the DBDH problem is

$$\frac{1}{2} Pr[u' = u | u = 1] + \frac{1}{2} Pr[u' = u | u = 0] - \frac{1}{2} = \frac{1}{2} \varepsilon$$

Table 1: Notations

Notations	Definition	Notations	Definition
Ce	pairing operation,e:$G_1 \times G_1 \to G_2$	C_c	Common-key encryption / decryption
C_R	RSA algorithms	G_1	Group or group operation(exponentiation,multiplication) ($i = 1,2$), g is a random generator of G_1,
S	The minimum satisfying set of interior nodes an access structure (include the root)	A_c	The set of attributes involved in a policy of a ciphertext
A_u	The set of attributes of a user	L_*	Bit-length of element in *
$h(d)$	The hash value of user's terminal fingerprint d	R	The number of update attributes (secret key and ciphertext)

Table 2: Comparision of CP-ABE scheme, Naruse's scheme and Our scheme

	CP-ABE scheme[2]	Naruse's scheme[16]	Our scheme
Public-key size	$3L_{G1} + L_{G2}$	$(3\|A_c\|+1)L_{G1} + L_{G2}$	$3L_{G1} + L_{G2}$
Commom-key size	-	-	L_c
Secret-key size	$(2\|A_u\|+1)L_{G1}$	$(2\|A_c\|+1)L_{G1}$	$(2\|A_u\|+1)L_{G1} + L_{h(d)}$
Ciphertext size	$(2\|A_c\|+1)L_{G1} + L_{G2}$	$(\|A_c\|+1)L_{G1} + L_{G2}$	$(2\|A_c\|+1)L_{G1} + L_{G2}$
Encryption computational time	$(2\|A_c\|+1)G_1 + 2G_2$	$(\|A_c\|+2)G_2$	$(2\|A_c\|+1)G_1 + 2G_2$
Re-Encryption computational time	-	$\|R\|G_2$	$\|h(d)\|G_2$
Decryption computational time	$2\|A_u\|C_e + (2\|S\|+2)G_2$	$(A_c+1)C_e + (A_c+1)G_2$	$C_c + 2\|A_u\|C_e + (2\|S\| + 2)G_2 + \|h(d)\|C_R$
Re-Decryption computational time	-	-	$C_c + \|h(d)\|C_R$

5.2 Efficiency

We analyze the efficiency of the scheme from following aspects: the size of the secret key, the size of the private key and the time required for the encryption and decryption. The ciphertext size and the key size are easy to calculate. They are linearly related to the number of leaf nodes and the number of user attributes.For each leaf node of the encrypted access tree, encryption algorithm perform two exponentiations. Key generation algorithm needs to perform the multiplication times is, two times the number of user attributes. In decryption operation, for each leaf node of access tree is required two times to bilinear map operations, and for each node of the access tree would need a multiplication.

In this section, we give table1 and table2 to compare the properties of CP-ABE scheme, Naruse's proposed re-encryption scheme and our scheme. Our scheme is different from another scheme. Firstly, our scheme uses the common key encryption scheme which allows calculate on of the amount of the ciphertext unrelated to encryption and decryption. That is whether you want to encrypt the contents that show; how long you have, and it does not spend too much time to encrypt and decrypt. Secondly, in our proposal data center to addition or revoke users only need to re-calculation of the encrypted text but do not need to update system's secret key. Finally, although our proposal system calculated amount of CP-ABE scheme has increased, we can still guarantee the security of secret key in the system

5.3 Discussion

In this study, a cryptosystem is proposed to improve the security and in that method, the user can receive only re-ciphertext and private information, while the server is sending both ciphertext and secret key. After receiving, the user will create both personal secret key and re-encrypted key and will keep the personal secret key. After that, the re-encrypted key will be sent back to the server and the server will use that key to re-encrypt ciphertext. Finally, the server will send that re-ciphertext back to the user.

The cryptosystem utilizes the terminal fingerprint, which is assumed to be unchangeable and unextractable. But also the key generation, encryption and decryption can get the value of thet fingerprints. However, it is different for each user but if can be used as a user ID and can give the guarantee for the user's information. Since the secret key has legitimate user includes their terminal fingerprint, the key cannot be used by other users.

In our hybrid encryption scheme, the public key encryption scheme can be utilized and easy to update and delete user's information. Therefore, a credible third party is not needed to guarantee the security and authentication of a user. In this scheme, the secret key will be generated and protected from against the communication channel attack.

It is requires to the own encryption terminal information to key management center by each user. The workload can be quite heavy if it has a large number of user application.

6 Conclusion and Future Work

In conclusion, the secret key does not operate except in the generated terminal fingerprint key pair and it can be protacted even if an attacker eavesdrops the user secret key.

As a future work, the encryption and decryption time should be optimized by using proper algorithms and the computational complexity of re-encrypted key should be decseased. Furthermore, the proper solution shoule be proposed wehn the user connects the internet, the terminal fingerprint can be eavesdropped by an attacker. Hence, the proper solution should be proposed to mitigate this issue. Hence, if the secred key is ensure, the key cannot be copied and forwarded and system will be ensure.

Acknowledgments

The second author is partially supported by Grants-in-Aid for Scientific Research of Japan Society for the Promotion of Science; Research Project Number:15K00029.
The fourth author is partially supported by Grants-in-Aid for Scientific Research of Japan Society for the Promotion of Science; Research Project Number: 15H02711.

References

[1] C. Adams and S. Lloyd. *Understanding PKI: concepts, standards, and deployment considerations*. Addison-Wesley Professional, 2003.

[2] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In *Proc. of the 2007 IEEE Symposium on Security and Privacy (SP'07), Berkeley, CA, USA*, pages 321–334. IEEE, May 2007.

[3] G. A. E. Bursztein, C. Jackson, and D. Boneh. An analysis of private browsing modes in modern browsers. In *Proc. of the 19th USENIX Security Symposium (Security'10), Washington, DC, USA*, August 2010.

[4] R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In *Proc. of the 2004 International Conference on the Theory and Applications of Cryptographic Techniques (Eurocrypt'04), Interlaken, Switzerland*, volume 3027 of *Lecture Notes in Computer Science*, pages 207–222. Springer Berlin Heidelberg, May 2004.

[5] C. Chen, H. Anada, J. Kawamoto, and K. Sakurai. Hybrid encryption scheme using terminal fingerprint and its application to attribute-based encryption without key misuse. In *Proc. of the 3rd IFIP TC 5/8 International Conference, ICT-EurAsia 2015, and the 9th IFIP WG 8.9 Working Conference, CONFENIS 2015, Held as Part of WCC 2015, Daejeon, Korea*, volume 9357 of *Lecture Notes in Computer Science*, pages 255–264. Springer International Publishing, October 2015.

[6] L. Cheung and C. Newport. Provably secure ciphertext policy abe. In *Proc. of the 14th ACM conference on Computer and Communications Security (CCS'07), Alexandria, VA, USA*, pages 456–465. ACM, October 2007.

[7] N. Doty. Fingerprinting guidance for web specification authors. W3C, Unofficial Draft 13, November 2015. [Online; Accessed on May 3, 2016] `https://w3c.github.io/fingerprinting-guidance/`.

[8] P. Eckersley. How unique is your web browser? In *Proc. of the 10th International Symposium on Privacy Enhancing Technologies (PETS'10), Berlin, Germany*, volume 6205 of *Lecture Notes in Computer Science*, pages 1–18. Springer Berlin Heidelberg, July 2010.

[9] M. J. Hinek, S. Jiang, R. Safavi-Naini, and S. F. Shahandashti. Attribute-based encryption with key cloning protection. Cryptology ePrint Archive, Report 2008/478, 2008. [Online; Accessed on May 3, 2016] `https://eprint.iacr.org/2008/478`.

[10] A. Jain, L. Hong, and S. Pankanti. Biometric identification. *Communications of the ACM*, 43(2):90–98, 2000.

[11] A. K. Jain, K. Nandakumar, and A. Nagar. Biometric template security. *EURASIP Journal on Advances in Signal Processing*, 2008:579416:1–579416:17, January 2008.

[12] A. Kahate. *Cryptography and network security*. Tata McGraw-Hill Education, 2013.

[13] S. S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen, and P. Tuyls. The butterfly puf protecting ip on every fpga. In *Proc. of the 2008 IEEE International Workshop on Hardware-Oriented Security and Trust (HOST'08), Anaheim, CA, USA*, pages 67–70. IEEE, June 2008.

[14] J. Li, K. Ren, and K. Kim. A2be: Accountable attribute-based encryption for abuse free access control. Cryptology ePrint Archive, Report 2009/118, 2009. [Online; Accessed on May 3, 2016] `http://eprint.iacr.org/2009/118`.

[15] J. Li, K. Ren, B. Zhu, and Z. Wan. Privacy-aware attribute-based encryption with user accountability. In *Proc. of the 12th International Conference on Information Security (ISC'09), Pisa, Italy*, volume 5735 of *Lecture Notes in Computer Science*, pages 347–362. Springer Berlin Heidelberg, September 2009.

[16] T. NARUSE, M. MOHRI, and Y. SHIRAISHI. Attribute revocable attribute-based encryption with forward secrecy. *Journal of Information Processing*, 55(10):2256–2264, 2014.

[17] F. Piper. *Cryptography*. Wiley Online Library, 2002.

[18] T. Saito, K. Takasu, T. Yamada, N. Takei, T. Ishikawa, R. Hosoi, K. Yasuda, and K. Takahashi. Current status and issues of web browser fingerprinting. In *Proc. of the 2015 Computer Security Symposium, Nagasaki, Japan*, pages 663–670. Information Processing Society of Japan, October 2015.

[19] G. E. Suh and S. Devadas. Physical unclonable functions for device authentication and secret key generation. In *Proc. of the 44th Annual Design Automation Conference (DAC'07), San Diego, CA, USA*, pages 9–14. ACM, June 2007.

[20] U. Uludag, S. Pankanti, S. Prabhakar, and A. K. Jain. Biometric cryptosystems: issues and challenges. *Proceedings of the IEEE*, 92(6):948–960, 2004.

[21] B. Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In *Proc. of the 14th International Conference on Practice and Theory in Public Key Cryptography (PKC'11), Taormina, Italy*, volume 6571 of *Lecture Notes in Computer Science*, pages 53–70. Springer Berlin Heidelberg, March 2011.

Author Biography

Chunlu Chen received the B.E. from Inner Mongolia University of science and Technology, China. And master of Informatics in 2014 from Kyushu University, Japan. She is interested in information security.

Hiroaki Anada received his B.E. and M.E. from Waseda University, and Ph.D. in Informatics from Institute of Information Security (IISEC), Japan. He is with Institute of Systems, Information Technologies and Nanotechnologies (ISIT),Japan and a visitor researcher at IISEC. He is interested in interactive proofs.

Junpei Kawamoto received his B.Eng. in 2007, Master of Informatics in 2008, and Ph.D. in Informatics in 2012 all from Kyoto University, Japan. He was also a research fellow of the Japan Society for the Promotion of Science between 2009-2011, an expert researcher of Japan's National Institute of Information and Communications Technology in 2012, then a Postdoctoral Fellow at the University of Tsukuba between 2012-2013. He is currently an Assistant Professor at Kyushu University in Fukuoka, Japan and concurrently working with the Institute of Systems & Information Technologies and Nanotechnologies, Japan. He is interested in database security and privacy preserving data mining.

Kouichi Sakurai received the B.S. degree in mathematics from the Faculty of Science, Kyushu University in 1986. He received the M.S. degree in applied science in 1988, and the Doctorate in engineering in 1993 from the Faculty of Engineering, Kyushu University. He was engaged in research and development on cryptography and information security at the Computer and Information Systems Laboratory at Mitsubishi Electric Corporation from 1988 to 1994. From 1994, he worked for the Dept. of Computer Science of Kyushu University in the capacity of associate professor, and became a full professor there in 2002. He is concurrently working also with the Institute of Systems & Information Technologies and Nanotechnologies, as the chief of Information Security laboratory, for promoting research co-operations among the industry, university and government under the theme "Enhancing IT-security in social systems". He received Minister of Economy, Trade and Industry Award in 2012.