Hybrid Encryption Scheme Using Terminal Fingerprint and Its Application to Attribute-Based Encryption Without Key Misuse

Chunlu Chen, Hiroaki Anada, Junpei Kawamoto, Kouichi Sakurai

Hybrid Encryption Scheme using Terminal Fingerprint and its Application to Attribute-based Encryption without Key Misuse

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

Abstract: Internet services make sharing digital contents faster and easier but raise an issue of illegal copying and distribution of those digital contents at the same time. A lot of public key encryption schemes solve this issue. However, the secret key is not completely protected i.e. these kinds of encryption methods do not prevent illegal copying and distribution of secret keys. In this paper, we propose a hybrid encryption scheme that employ terminal fingerprints. This scheme is a template to avoid such misuse of secret keys, and can be applied to, for example, attribute-based encryption schemes. There terminal fingerprint information is used to create a second encryption key and secret key. Since the terminal fingerprint is assumed to be unchangeable and unknowable, we ensure that our secret keys are valid in the terminal where such secret keys were created.

Keyword: Key misuse, Terminal fingerprint, Re-encryption

1. Introduction

In the last few years, the amount of data stored in the cloud server has been increasing day by day with the rapid development 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 the security of the data stored in the cloud became a critical task. Typically, data stored in the cloud must be encrypted in order to achieve this goal of ensuring 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, public key encryption can enhance security, the complexity of key management is a big issue in this kind of cryptography.

Although public key cryptography can help us to protect the massage, it also can be used to make illegal acts such as transferring and copying secret key unauthorized. Furthermore, it is possible to copy secret key from other users illegally. It is difficult to identify the source of leaking or the responsible entity if the secret key leaks.

Various methods are proposed to utilize unique information for secret key generation to prevent this behavior, bur the leakage of secret key is still a weak point of encryption waiting to be solved in the near future. 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" [1]. PUF system receives a random code, and generates a unique random code as a response. Due to differences in the manufacturing process, the produced chip cannot be imitated and copied.

Kumar et al. [2] 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 massage through the uniqueness of the secret key from copying and other 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 [3]. Furthermore, biometric can be used as a unique, unalterable secret key but the safety is still taken seriously.

Jain, Anil et al. [4] analyzed and evaluated these biometric authentication systems. Moreover, biometric authentication is also used in various fields, for example, Uludag et al. [5] 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.

Terminal Fingerprint

In general, the type of font, screen resolution and network environment are different for each browser terminal that is used to receive fingerprint information. This information can be used as the feature points to identify the terminal. The various sets of features possessed by the browser in this way is referred as browser fingerprint [6, 7, 8]. In this paper the terminal fingerprint is assumed to be unchangeable and unextractable.

Terminal fingerprint has been applied in a variety of locations. For example, terminal fingerprint is used to track the behavior of users on the web by collecting the trend of web sites that the user has accessed. As a result, it is possible to provide advertisements tailored to the interest and favorite of the user. It has also been made

applicable to the risk-based authentication. The authentication terminal fingerprints are taken at the time of login of the user, and save. The terminal fingerprints are compared with those of the previous log. If it is significant difference, it will be determined that there is a high possibility of access from another terminal, which causes a higher strength authentication.

The hardware based authentication and Biometric based authentication methods mentioned above ensure the uniqueness of the key. These still cannot guarantee the security of keys. The update of hardware based authentication requires the replacement of the hardware itself, which will increases system cost. Biometric based authentication is impossible to alter but it is possible to be copied.

In order to meet the point, this paper utilizes the terminal fingerprint information because every terminal fingerprint information 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 information of the user can be utilized as a secret key, and it is never revealed outside even once. Unless the owner leaks information, otherwise the security of the key is guaranteed. Safety of the secret key is increased in this way.

For this purpose, we propose a hybrid encryption scheme that consists of a common-key encryption scheme and two public key encryption schemes. The hash value of a terminal fingerprint will be used as a secret key in the second public key scheme. In this paper, we employ Waters' CP-ABE [9] as the first encryption scheme, but any public key encryption scheme could be used as the first. Our scheme does not only utilize terminal fingerprint for generating unique secret key, but also updates itself according to user settings with relatively low cost to keep the freshness of the terminal fingerprint.

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

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: $G_1 \times G_1 \rightarrow G_2$. The bilinear map e has the following properties:

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

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) [10].

Definition 1 (Access Structure) *Let* $P = \{P_1, P_2, \ldots, P_n\}$ *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, monotone assess structures) is a collection (respectively, monotone collection)* Γ *of nonempty subsets of* P, *i.e.,* $\Gamma \subset 2^P \setminus \{\emptyset\}$. *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) [10]) *A secret-sharing scheme* Π *over a set of parties* $\mathcal{P}$ *is called linear (over* Z_p*) if*

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

2. There exists a matrix M with ℓ *rows and n columns called the share-generating matrix for* Π. *For all i = 1, ..., ℓ, 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* $v = (s, r_2, \ldots, r_n)$, *where* $s \in Z_p$ *is the secret to be shared, and* $r_2, \cdots, r_n \in Z_p$ *are randomly chosen, then Mv is the vector of* ℓ *share of the secret s according to* Π. *The share* $(Mv)_i$ *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 $I \subset \{1,2,\ldots,\ell\}$ 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 $\sum_{i \in I} \omega_i \lambda_i = s$.

2.3 CP-ABE

There are a lot of studies on enhance the security of system. Cheung and Newport [11] proposed CP-ABE scheme based on DBDH problem using the CHK techniques [12], 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. Naruse et al. [13] 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. 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 included in 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, but the public key distribution center will increase the workload. Hinek et al. [16] 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$(\boldsymbol{\lambda}, \mathbf{U}) \rightarrow (\mathbf{PK}, \mathbf{MK})$: 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$(\mathbf{PK}, \mathbf{M}, \mathbf{W}) \rightarrow \mathbf{CT}$: The Encrypt algorithm takes the public parameter PK, a message M, and an access structure was input. It output a cipher text CT.

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

Decrypt$(\mathbf{CT}, \mathbf{SK}) \rightarrow \mathbf{M}$: 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. Our System Model

In this section, we propose a hybrid encryption scheme. Then, we propose an attribute-based encryption scheme without key misuse. Finally, we provide a concrete realization of our 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 information 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.

3.1 Our Hybrid Encryption Scheme

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 =

$(\mathrm{CKE}, \mathrm{PKE1}, \mathrm{PKE2}, H)$. 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. And 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.

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

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

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

3.2 Our Concrete Construction of ABE without Key Misuse

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* information. The advantages of this scheme, confirmation of the terminal fingerprint information 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 3.1.

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

DO.Setup$(\boldsymbol{v}, \boldsymbol{w}) \rightarrow \mathbf{FK}$ **:** 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

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

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

Auth.Setup $(\boldsymbol{\lambda}) \rightarrow \mathbf{PK}, \mathbf{MK}$ **:** 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: $\mathrm{G}_1 \times \mathrm{G}_1 \rightarrow \mathrm{G}_2$. It then chooses two random exponents $a, b \in Z_p$ and hash function $\mathrm{H}: \{0,1\}^* \rightarrow \mathrm{G}$ as

input. The Common key is published as

$$\text{PK} = g, g^b, e(g,g)^a$$

The system master secret key is published as

$$\text{MK} = g^a$$

Auth.Ext$(\mathbf{MK}, \boldsymbol{S}) \rightarrow \mathbf{SK}$**:** 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

$$SK = (g^{a+bt}, g^t, (K_X)_{X \in S}), \qquad \forall_{X \in S}\ K_X = H(X)^t$$

U.Setup$(\mathbf{SK}, \boldsymbol{f}) \rightarrow \mathbf{F}, \mathbf{D}$**:** 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. $DE \equiv 1 \bmod (p-1)(q-1)$. The user's terminal-fingerprint public key is $F = (N, E)$. The user keeps *D* as the user's terminal-fingerprint secret key.

Auth.Enc$(\mathbf{PK}, \mathbf{FK}, \boldsymbol{W}) \rightarrow \mathbf{FT}$ **:** The Auth.Enc algorithm takes the public parameter PK, common key FK, and an access structure (W, ρ) over the all of attributes to encrypts a message *M*. The function ρ associates row of *W* to attributes. Where *W* is an $\ell \times n$ matrix. First the algorithm generates a vector $\varkappa = (s, y_2, \cdots, y_n) \in Z_p^n$ and $r_1, r_2, \cdots, r_\ell \in Z_p$ 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 $\lambda_i = \varkappa \cdot W_i$ from 1 to ℓ.

It output a ciphertext FT as

$$\text{FT} = (FKe(g,g)^{as}, g^s, \widehat{Cs}),$$

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

Auth.ReEnc$(\mathbf{FT}, \mathbf{F}) \rightarrow \mathbf{FT}'$ **:** The Auth.ReEnc algorithm takes the cipher text FT and user's terminal-fingerprint public key F as input.

The re-cipher text is published as

$$\text{FT}' = (\text{FT})^{\text{E}} \bmod N,$$

$$\text{Where } (\text{FT})^{\text{E}} = (FKe(g,g)^{as\text{E}}, g^{s\text{E}}, (\widehat{Cs})^{\text{E}}).$$

U.Dec$(\mathbf{FT}', \boldsymbol{D}) \rightarrow \mathbf{FT}$ **:** The U.Dec algorithm takes as input the cipher text FT′ and *D*. The decryption algorithm first computes.

The decryption algorithm computes

$$(\text{FT}')^D = (\text{FT}^{\text{E}})^D = \text{FT} \bmod N.$$

U.ReDec$(\mathbf{FT}, \mathbf{SK}) \rightarrow \mathbf{FK}$ **:** The U.ReDec algorithm takes the cipher text FT and

secret key SK as input. The secret key for an attribute set S, and the cipher text FT for access structure (W, ρ). Suppose that *S* satisfies the access structure and define *I* as $\{i = \rho(i) \in s\}, I \in \{1,2,\ldots, \ell\}$ for $\prod$, 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

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

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

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*.

4. Discussion

This paper 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 attacks. If the encrypted data is published, our scheme also resists attacks from colluding users. If the attacker did not know the terminal fingerprint information of the legitimate user, they wouldn't be able to get the secret key.

In this study, we proposed a cryptosystem to improve security. In the proposed method, the data server only sends re-cipher text and private information to the user, while the data server has to send both cipher text and secret key. In addition, the user creates secret key and re-encrypt key using the private information. Henceforth, user keeps the secret key and sends the re-encrypt key to the data server, and then the data server use the re-encrypt key to re-encrypt cipher text. Finally, data server sends back the re-cipher text to the user.

The proposed cryptosystem utilizes the terminal fingerprint information of the user. The terminal fingerprint is assumed to be unchangeable and unknowable. Also, only the key generation, encryption and decryption programs running on the trusted terminal can get the value of the fingerprints. The proposed scheme is built on the above-mentioned conditions. Here, the terminal fingerprint information is different for each user. It can be used as a user ID, and you can guarantee the anonymity of the user's own information. Misuse of the terminal fingerprint, such as transfer of the secret key, is incorrect behavior and meaningless,. Since the secret key that has legitimate user includes their terminal fingerprint information, the terminal fingerprint information is different in the other terminal, and the secret key is revoked. Safety of the secret key is increased in this way.

We proposed a hybrid encryption scheme in which a public key encryption scheme can be utilized. It is also easy to add, update and delete user's information. Then, we do not need a credible third party to guarantee the security of encryption and authenticate a user. In this scheme, the secret key is generated and stored by the user, protecting the secret key against communication channel attack.

Our scheme requires that each user provide their own encryption terminal information to key management center. If there is large number of simultaneous user application, the workload of management center can be quite heavy. So in the future we should consider decreasing the computational complexity of re-encrypted.

Finally, the system ensures that the key cannot be copied, forwarded and etc. if there the safety of the security key is provided.

5. Conclusion and Future Work

In this study, we combine user terminal fingerprint data with a public key and secret key pair. Furthermore, we proposed a cryptographic scheme to update the secret key during decryption phase using terminal fingerprint information. As a result, the secret key is protected by ensuring that it does not operate except in the generated terminal key pair, even if an attacker eavesdrops the user secret key.

The encryption and decryption time can be optimized by proposing suitable algorithm as the future work. Furthermore, the security issue of our proposed method is that if the user connects to the Internet, the terminal fingerprint can be eavesdropped by an attacker. Hence, the proper solution should be proposed to mitigate this issue.

Acknowledgements

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.
(Authors: Chunlu Chen, Hiroaki Anada, Junpei Kawamoto, Kouichi Sakurai)

References

[1] Suh, G. Edward, and Srinivas Devadas. "Physical unclonable functions for device authentication and secret key generation." *Proceedings of the 44th annual Design Automation Conference*. ACM, 2007.

[2] Kumar, Sandeep S., et al. "The butterfly PUF protecting IP on every FPGA." *Hardware-Oriented Security and Trust, 2008. HOST 2008. IEEE International Workshop on*. IEEE, 2008.

[3] Jain, Anil, Lin Hong, and Sharath Pankanti. "Biometric

identification." *Communications of the ACM* 43.2 (2000): 90-98.

[4] Jain, Anil K., Karthik Nandakumar, and Abhishek Nagar. "Biometric template security." *EURASIP Journal on Advances in Signal Processing* 2008 (2008): 113.

[5] Uludag, Umut, et al. "Biometric cryptosystems: issues and challenges." *Proceedings of the IEEE* 92.6 (2004): 948-960.

[6] W. C. Nick Doty. (24 Feb 2015). *Fingerprinting Guidance for Web Specification Authors (Unofficial Draft)*. Available: http://w3c.github.io/fingerprinting-guidance/

[7] Aggarwal, Gaurav, et al. "An Analysis of Private Browsing Modes in Modern Browsers." *USENIX Security Symposium*. 2010.

[8] Eckersley, Peter. "How unique is your web browser?." *Privacy Enhancing Technologies*. Springer Berlin Heidelberg, 2010.

[9] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Security and Privacy, 2007. SP'07. IEEE Symposium on*, 2007, pp. 321-334.

[10] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Public Key Cryptography–PKC 2011*, ed: Springer, 2011, pp. 53-70.

[11] L. Cheung and C. Newport, "Provably secure ciphertext policy ABE," in *Proceedings of the 14th ACM conference on Computer and communications security*, 2007, pp. 456-465.

[12] R. Canetti, S. Halevi, and J. Katz, "Chosen-ciphertext security from identity-based encryption," in *Advances in Cryptology-Eurocrypt 2004*, 2004, pp. 207-222.

[13] T. Naruse, M. Mohri, and Y. Shiraishi, "Attribute Revocable Attribute-Based Encryption with Forward Secrecy," presented at the Proc. of the 2014 information processing society of Japan, Japan, 2014.

[14] J. Li, K. Ren, and K. Kim, "A2BE: Accountable Attribute-Based Encryption for Abuse Free Access Control," *IACR Cryptology ePrint Archive,* vol. 2009, p. 118, 2009.

[15] J. Li, K. Ren, B. Zhu, and Z. Wan, "Privacy-aware attribute-based encryption with user accountability," in *Information Security*, ed: Springer, 2009, pp. 347-362.

[16] Hinek, M. Jason, et al. "Attribute-Based Encryption with Key Cloning Protection." *IACR Cryptology ePrint Archive* 2008 (2008): 478.