POSTER: Euclidean Distance Based Encryption: How to Embed Fuzziness in Biometric Based Encryption

Fuchun Guo
University of Wollongong
NSW, Australia
fuchun@uow.edu.au

Willy Susilo
University of Wollongong
NSW, Australia
wsusilo@uow.edu.au

Yi Mu
University of Wollongong
NSW, Australia
ymu@uow.edu.au

ABSTRACT

We introduce a new encryption notion called *Euclidean Distance based Encryption* (EDE). In this notion, a ciphertext encrypted with a vector and a threshold value can be decrypted with a private key of another vector, if and only if the Euclidean distance between these two vectors is less than or equal to the threshold value. Euclidean distance is the underlying technique in the pattern recognition and image processing community for image recognition. The primary application of this encryption notion is to enable an identity-based encryption that incorporates biometric identifiers, such as fingerprint, face, hand geometry, vein and iris. In that application, usually the input biometric will not be exactly the same during the enrollment and encryption phases. In this poster, we propose this new encryption notion and study its construction. We show how to generically and efficiently construct an EDE from an inner-product encryption (IPE) with reasonable size of private keys and ciphertexts. We also propose a new IPE scheme that is equipped with a specific characteristic to build EDE, namely the need for short private key. Our IPE scheme achieves the shortest private key compared to existing IPE schemes in the literature, where our private key is composed of two group elements only.

Categories and Subject Descriptors

E.3 [**Data Encryption**]: Public Key Cryptosystems

General Terms

Security, Algorithms

Keywords

Identity-based Encryption, Biometrics, Euclidean Distance

1. INTRODUCTION

Identity-based encryption (IBE) system is a public key encryption in which any identity information of a user can be set as the user's public key. In the literature, the most commonly used identity information is an *arbitrary string*, such as name, email address or social security number. The arbitrary string can be seen as a *text-based identity* that comprises a combination of alphabet and numbers in various orders.

Text-based approach is quite commonly used in the real world applications to represent an identity. Nevertheless, there are some limitations with a text-based approach, and it is merely impractical in some situations. Biometric traits such as fingerprint, face, iris and vein can be also used to represent the identities of users due to their unique features. In contrast to traditional text-based identities, people do not need to remember their biometrics information. With the advance of technology, biometrics readers have been rapidly developed and deployed.

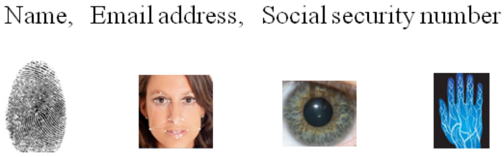

Figure 1: Text-based identities vs biometric identities.

It is widely accepted that biometric identities offer many interesting features, and therefore in this work we are motivated to incorporate the biometric identities as identities in an identity-based encryption notion. Therefore, our main question is: *Can we use biometrics in an identity-based encryption?* More precisely, first a user acquires a private key of his/her biometric, which is gathered during the key generation phase (which is part of the enrollment phase). Subsequently, the user's biometric will be captured again during the encryption phase to produce the ciphertext. We aim to enable the decryption of the ciphertext, if and only if the biometric trait gathered during the key generation phase and the biometric trait used during the encryption phase belong to the same user, i.e., the two biometric traits are "close" to each other.

To tackle the fuzziness of the biometric measurements, it is widely accepted to utilize an image processing technique (or pattern recognition). The idea of "distance" is fundamental to image processing and pattern recognition. Euclidean distance is a very useful distance measurement that has been adopted for a variety of image recognition (see [1]). Given two biometrics represented with two images, the image recognition algorithm firstly processes images into vectors. Then, it calculates the Euclidean distance of these two vectors, and compares the distance with a threshold value. If the distance is less than or equal to the threshold value, the algorithm outputs the "*Match*" result to indicate that the two biometrics are considered to be from the same user. Otherwise, it returns "*Mismatch*" to claim the two biometrics are from different users.

In cryptography research community, the use of biometric in identity-based encryption system (IBE) was first mentioned by Sahai and Waters [2]. They formalized the notion called Fuzzy IBE, which allows error-tolerance property of a private key of a biometric to decrypt a ciphertext encrypted with a slightly different biometric. This work eventually evolved to the development of attribute-based encryption, which has been used to develop many interesting applications. We note that it is unfortunate that the notion of fuzzy IBE only considers a simple set overlap distance in judging the similarity of two biometrics, which are treated as sets of descriptive attributes. We note that this test (recognition) approach is *not* the widely accepted technique to measure the difference of biometrics in the pattern recognition and image processing community. Hence, although fuzzy IBE has been claimed to incorporate biometric in IBE system, unfortunately this is not the approach that is used in practice.

1.1 Our Contributions

We introduce *Euclidean Distance based Encryption* (EDE). In this encryption notion, a private key of vector $\vec{y}$ can decrypt a ciphertext encrypted with another vector $\vec{x}$ and a threshold value t, if and only if the Euclidean distance between $\vec{x}$ and $\vec{y}$ is less than or equal to t. The primary motivation of this work is to bride the gap between biometric based encryption and pattern recognition.

In this poster, we propose this encryption notion and study its construction. The adopted Euclidean distance measurement is called weighted squared Euclidean distance, which is a generalization of (squared) Euclidean distance. We show how to generically and efficiently construct an EDE from an inner-product encryption (IPE) with reasonable size of private keys and ciphertexts. Given any integer k_0 defined by the system generator, each EDE key has $k_0 + 1$ numbers of IPE keys and each EDE ciphertext has $\lceil \frac{t}{k_0} \rceil$ numbers of IPE ciphertexts. A proper k_0 can be selected to balance the size between private keys and ciphertexts. We also propose a new IPE scheme equipped with a specific characteristic to build EDE, namely the need for a short private key. The private key of our IPE scheme comprises of two group elements only, compared to the best efficient IPE scheme in the literature with nine group elements [3]. The EDE instantiation from our IPE will therefore save more than 75% secure memory for private key storage. We prove the security of our IPE scheme with payload security in the selective security model under the Decision Bilinear Diffie-Hellman assumption.

2. DEFINITION OF EDE

Let $\vec{x} = (x_1, x_2, \cdots, x_n)$ and $\vec{y} = (y_1, y_2, \cdots, y_n)$ be two n-length vector, where all values are from the real number space $\mathbb{R}$. Let $f_i \in \mathbb{R}$ be the normalization factor for the ith value. The Euclidean distance $d_E(\vec{x}, \vec{y})$ and weighted squared Euclidean distance $d_W(\vec{x}, \vec{y})$ are defined as follows.

$$d_E(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2},\ d_W(\vec{x}, \vec{y}) = \sum_{i=1}^{n} f_i(x_i - y_i)^2.$$

In this work, we adopt the *weighted squared Euclidean distance* in the EDE construction, which is a generalization of (weighted squared) Euclidean distance. Throughout the rest of this poster, we use $d(\vec{x}, \vec{y})$ to denote the weighted squared Euclidean distance between $\vec{x}$ and $\vec{y}$.

To be able to embed all values in vectors to the exponents, we need to adjust all values to integers. For example, suppose $x_i, y_i \in [0, 1)$ and each is a decimal number represented with three digits. We can multiply each value with integer 1,000 to change all decimals into integers. In this poster, we directly assume $\mathbb{S}, \mathbb{F} \subseteq \mathbb{Z}_p$ for some prime number p of a group order.

The EDE comprises of the following four algorithms.

Setup. The setup algorithm takes as input the security parameter λ and distance parameters $(n, f_1, f_2, \cdots, f_n)$. It returns a master public/secret key pair (mpk, msk).

KeyGen. The key generation algorithm takes as input msk and an n-length vector $\vec{y}$. It returns a private key $sk_{\vec{y}}$ for $\vec{y}$.

Encryption. The encryption algorithm takes as input mpk, an n-length vector $\vec{x}$, a threshold value t and a message M. It outputs a ciphertext $CT = \mathsf{Enc}[\vec{x}, t, M]$.

Decryption. The decryption algorithm takes as input a ciphertext CT for $(\vec{x}, t)$, the master public key mpk and the private key $sk_{\vec{y}}$ of $\vec{y}$. It attempts to decrypt the ciphertext and outputs the message if $d(\vec{x}, \vec{y}) \leq t$. Otherwise, it simply returns the symbol $\perp$.

Correctness Consider all $(mpk, msk, n, f_1, f_2, \cdots, f_n, \vec{x}, t)$ and $(\vec{y}, sk_{\vec{y}})$. Suppose $CT = \mathsf{Enc}[\vec{x}, t, M]$. If $d(\vec{x}, \vec{y}) \leq t$, we have the decryption on CT using $sk_{\vec{y}}$ will output the message M.

Security Without a valid private key $sk_{\vec{y}}$ for any vector $\vec{y}$ satisfying $d(\vec{x}, \vec{y}) \leq t$, it requires that an adversary cannot distinguish the message in $CT = \mathsf{Enc}[\vec{x}, t, M]$.

The security model (semantic security) for EDE is similar to the fuzzy IBE. Let $\vec{x}^*$ be the challenge vector and t^* be the challenge threshold value. An adversary can only query private keys for $\vec{y}$ satisfying $d(\vec{x}^*, \vec{y}) > t^*$. Otherwise, the adversary can trivially win the game by decrypting the challenge ciphertext by itself.

3. GENERIC CONSTRUCTION OF EDE

In this section, we first revisit the definition of inner-product encryption (IPE). Then, we show how to generically and efficiently construct EDE from an IPE with reasonable size of private keys and ciphertexts.

3.1 Inner-Product Encryption

An inner-product encryption comprises of the following four algorithms $\mathsf{IPE.Setup}$, $\mathsf{IPE.Key}$, $\mathsf{IPE.Enc}$, $\mathsf{IPE.Dec}$.

- The $\mathsf{IPE.Setup}[\lambda, n]$ (setup) algorithm takes as input the security parameter λ and an integer n. It outputs a master public/secret key pair ($\mathsf{IPE.mpk}$, $\mathsf{IPE.msk}$).

- The $\mathsf{IPE.Key}[\vec{z}, \mathsf{IPE.msk}]$ (key generation) algorithm takes as input an n-length vector $\vec{z} = (z_1, z_2\ , \cdots, z_n) \in \Sigma^n$ and ($\mathsf{IPE.mpk}$, $\mathsf{IPE.msk}$). It outputs a private key $\mathsf{IPE.sk}_{\vec{z}}$ of $\vec{z}$. Here, Σ is the space of all vectors.

- The $\mathsf{IPE.Enc}[\vec{w}, M]$ (encryption) algorithm takes as input an n-length vector $\vec{w} = (w_1, w_2, \cdots, w_n) \in \Sigma^n$, a message M and $\mathsf{IPE.mpk}$. It outputs a ciphertext denoted by $\mathsf{IPE.CT}$.

- The $\mathsf{IPE.Dec}[\mathsf{IPE.CT}, \mathsf{IPE.sk}_{\vec{z}}]$ (decryption) algorithm takes as input $\mathsf{IPE.CT}$ encrypted with $\vec{w}$ and the private key $\mathsf{IPE.sk}_{\vec{z}}$ of $\vec{z}$. It outputs the message iff the inner product, denoted by $\langle \vec{w}, \vec{z} \rangle$, is equal to zero. More precisely, successful decryption requires $\langle \vec{w}, \vec{z} \rangle = w_1z_1 + w_2z_2 + \cdots + w_nz_n = 0$[1].

3.2 Our Generic Construction

Katz, Sahai and Waters [4] proposed a generic vector transformation to support polynomial equations, which is also suitable for

[1] More precisely, the inner product considers $\langle \vec{w}, \vec{z} \rangle = 0 \bmod p$, where p is the order of group.

our EDE construction. However, the generic transformation is not very efficient due to the long vector transformation. Here, we give specific vector transformations based on the formula of Euclidean distance. We transform n-length vectors $\vec{x}, \vec{y}$ into the $n+2$-length vectors $\vec{w}, \vec{z}$, respectively, under the integer l defined as follows.

$$\vec{w}_l = (\ x_1\ ,\ x_2\ ,\ \cdots\ ,\ x_n\ ,\ -l+\sum_{i=1}^{n} f_i x_i^2\ ,\ \ 1\ \ )$$

$$\vec{z}_l = (-2f_1y_1, -2f_2y_2, \cdots, -2f_ny_n,\ 1\ ,\ l+\sum_{i=1}^{n} f_i y_i^2).$$

It is not hard to verify the transformation satisfies the relationship between inner product and distance $\langle \vec{w}_{l_1}, \vec{z}_{l_2} \rangle = d(\vec{x}, \vec{y}) + l_2 - l_1$.

With the above vector transformation, the generic construction of EDE from an IPE is defined as follows.

Setup. Taking as input the security parameter λ and distance parameters $(n, f_1, f_2, \cdots, f_n)$, the setup algorithm runs the algorithm $\mathsf{IPE.Setup}[\lambda, n+2]$ to generate $\mathsf{IPE.mpk}$ and $\mathsf{IPE.msk}$. Next, it chooses an integer k_0. The mpk and msk are defined as

$$mpk = (\mathsf{IPE.mpk}, n, f_1, f_2, \cdots, f_n, k_0),\ \ msk = \mathsf{IPE.msk}.$$

KeyGen. The key generation algorithm takes as input an n-length vector $\vec{y} = (y_1, y_2, \cdots, y_n)$ and the master secret key msk. It runs the algorithm $\mathsf{IPE.Key}[\vec{z}_l, \mathsf{IPE.msk}] : l = 0, 1, 2, \cdots, k_0$. The private key $sk_{\vec{y}}$ of $\vec{y}$ is $sk_{\vec{y}} = \Big\{\mathsf{IPE.sk}_{\vec{z}_l} : l = 0, 1, 2, \cdots, k_0\Big\}$.

Encryption. The encryption algorithm takes as input the master public key mpk, an n-length vector $\vec{x}$, a threshold value t and a message M. Let $l_0 = \lceil \frac{t}{k_0} \rceil$ (the minimum integer not less than $\frac{t}{k_0}$). It runs the algorithm $\mathsf{IPE.CT}_l = \mathsf{IPE.Enc}[\vec{w}_l, M]$ for $l = k_0, 2k_0, \cdots, (l_0-1)k_0, t$. The ciphertext encrypted with $(\vec{x}, t)$ is

$$CT = \Big\{\mathsf{IPE.CT}_l : l = k_0, 2k_0, \cdots, (l_0-1)k_0, t\Big\}.$$

Decryption. The decryption takes as input a ciphertext CT for $(\vec{x}, t)$, the master public key mpk and the private key $sk_{\vec{y}}$ of $\vec{y}$. If $d(\vec{x}, \vec{y}) = j > t$, it simply returns the symbol $\perp$. Otherwise, suppose $(i-1)\cdot k_0 \le j < i \cdot k_0$ holds for some $i \in \{1, 2, \cdots, l_0\}$. The decryption algorithm decrypts the message M by running the algorithm $\mathsf{IPE.Dec}[\mathsf{IPE.CT}_{i\cdot k_0}, \mathsf{IPE.sk}_{\vec{z}_{i\cdot k_0 - j}}]$ for $i \le l_0 - 1$ or running the algorithm $\mathsf{IPE.Dec}[\mathsf{IPE.CT}_t, \mathsf{IPE.sk}_{\vec{z}_{t-j}}]$ for $i = l_0$.

In our generic construction, each EDE key is composed of k_0+1 numbers of IPE keys and each EDE ciphertext comprises of $\lceil \frac{t}{k_0} \rceil$ numbers of IPE ciphertexts. We can choose a proper k_0 to balance the private key size and ciphertext size.

Correctness. The decryption is correct when the distance satisfies $d(\vec{x}, \vec{y}) = j \le t$. If $(i-1)k_0 \le j < ik_0$ for some $i \le l_0 - 1$, we have $\langle \vec{w}_{i\cdot k_0}, \vec{z}_{i\cdot k_0 - j} \rangle = 0$ and $0 < ik_0 - j \le k_0$. Otherwise we have $\langle \vec{w}_t, \vec{z}_{t-j} \rangle = 0$ and $0 = t - t \le t - j \le t - (l_0 - 1)\cdot k_0 \le k_0$. Therefore, we have $\mathsf{IPE.sk}_{\vec{z}_{i\cdot k_0 - j}} \in sk_{\vec{y}}$ or $\mathsf{IPE.sk}_{\vec{z}_{t-j}} \in sk_{\vec{y}}$. The ciphertext hence can be decrypted with $sk_{\vec{y}}$. It is not hard to verify that the ciphertext cannot be decrypted when $d(\vec{x}, \vec{y}) > t$ and the IPE scheme is secure, because for all $l_1 \le t$ and $l_2 \ge 0$

$$\langle \vec{w}_{l_1}, \vec{z}_{l_2} \rangle = d(\vec{x}, \vec{y}) + l_2 - l_1 > 0 \text{ and then } \langle \vec{w}_{l_1}, \vec{z}_{l_2} \rangle \neq 0.$$

4. NEW IPE WITH SHORT PRIVATE KEYS

In this section, we propose a new IPE with the shortest private key compared to existing IPE schemes in the literature. It can be adopted to reduce to the private key size of EDE scheme.

$\mathsf{IPE.Setup}$: The setup algorithm takes as input a security parameter λ and an integer n to denote the length of vector. It first chooses a pairing group $\mathbb{PG} = (\mathbb{G}, \mathbb{G}_T, g, p, e)$. The algorithm then chooses random α_i, β from $\mathbb{Z}_p$ for all $i = 1, 2, \cdots, n$. Finally, for all $i = 1, 2, \cdots, n$, it computes group elements $g_i = g^{\alpha_i}$ and $u = e(g, g)^{\beta}$. The $\mathsf{IPE.mpk}$ and $\mathsf{IPE.msk}$ are defined as:

$$\mathsf{IPE.mpk} = (\mathbb{PG},\ g_i,\ u),\ \ \mathsf{IPE.msk} = (\alpha_1, \alpha_2, \cdots, \alpha_n, \beta).$$

$\mathsf{IPE.Key}$: The key generation algorithm takes as input an n-length vector $\vec{z} = (z_1, z_2, \cdots, z_n) \in \mathbb{Z}_p^n$ and the master public/secret key pair $(\mathsf{IPE.mpk}, \mathsf{IPE.msk})$. The algorithm randomly chooses $t \in \mathbb{Z}_p$ and computes the private key $\mathsf{IPE.sk}_{\vec{z}}$ as follows.

$$\mathsf{IPE.sk}_{\vec{z}} = \Big(g^{\beta + t\sum_{i=1}^{n} \alpha_i z_i},\ \ g^t\Big) \in \mathbb{G} \times \mathbb{G}.$$

$\mathsf{IPE.Enc}$: The encryption algorithm takes as input an n-length vector $\vec{w} = (w_1, w_2, \cdots, w_n) \in \mathbb{Z}_p^n$, a message $M \in \mathbb{G}_T$ and the master public key $\mathsf{IPE.mpk}$. It chooses random r, s from $\mathbb{Z}_p$ and creates the ciphertext as follows.

$$\mathsf{IPE.CT} = \Big(u^r \cdot M,\ \ g^r,\ \ g_1^r g^{sw_1},\ g_2^r g^{sw_2},\ \ \cdots,\ g_n^r g^{sw_n}\Big).$$

$\mathsf{IPE.Dec}$: Suppose that $\mathsf{IPE.CT} = (C_m, C_0, C_1, \cdots, C_n)$ is a ciphertext encrypted with $\vec{w}$ and we have a private key $\mathsf{IPE.sk}_{\vec{z}}$ for $\vec{z}$ satisfying $\langle \vec{w}, \vec{z} \rangle = 0$. The decryption algorithm decrypts the message by $M = C_m \cdot e(g^{\beta + t\sum_{i=1}^{n} \alpha_i z_i}, C_0)^{-1} e(g^t,\ \prod_{i=1}^{n} C_i^{z_i})$.

The proposed IPE is provably secure in the selective security model with payload property under the DBDH assumption (given g, g^a, g^b, g^c, Z, to decide whether Z is equal to $e(g,g)^{abc}$ or not). The sketch of proof is described as follows. Let $\vec{w}^* = (w_1^*, \cdots, w_n^*)$ be the challenge vector. Randomly choose $\eta, \eta_i \in \mathbb{Z}_p$ and set $\alpha_i = -\eta w_i^* b + \eta_i$, $\beta = bc$. Subsequently, we can simulate the $\mathsf{IPE.mpk}$. For a private key query on $\vec{z}$ satisfying $\langle \vec{w}^*, \vec{z} \rangle \neq 0$, set t as $t = \frac{1}{\langle \vec{w}^*, \vec{z} \rangle \eta} c + t'$ where t' is a random number. Hence, we can simulate the private key of $\vec{z}$. In the challenge ciphertext, we set $r = a$ and $s = \eta \cdot ab$, where u^r contains $e(g,g)^{abc}$ and we embed Z into it. If $Z = e(g,g)^{abc}$, we have simulated a valid challenge ciphertext. Otherwise, it is a one-time encryption such that we can use the guess of the adversary to solve the hard problem.

5. CONCLUSION

We introduced the notion of Euclidean distance based encryption (EDE). The primary motivation of proposing this encryption notion is to bridge the gap between biometric based encryption and pattern recognition. We proposed a generic construction of EDE from IPE with reasonable size of private keys and ciphertexts. We also proposed a new IPE with the shortest private key composed of two group elements only, for the need of short EDE keys. We note that our new IPE is of an independent interest.

6. REFERENCES

[1] A. K. Jain and D. Maltoni, *Handbook of Fingerprint Recognition*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2003.

[2] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *EUROCRYPT 2005*, ser. LNCS, vol. 3494. Springer, 2005, pp. 457–473.

[3] T. Okamoto and K. Takashima, "Achieving short ciphertexts or short secret-keys for adaptively secure general inner-product encryption," in *CANS 2011*, ser. LNCS, Springer, 2011, pp. 138–159.

[4] J. Katz, A. Sahai, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," in *EUROCRYPT 2008*, ser. LNCS, vol. 4965. Springer, 2008, pp. 146–162.