# $(n,t)$ Threshold Key Generation in Identity-based Cryptosystems*

Fu-Kuo Tseng
Department of Computer Science
National Chiao Tung University
fktseng@cs.nctu.edu.tw

Rong-Jaye Chen
Department of Computer Science
National Chiao Tung University
rjchen@cs.nctu.edu.tw

***Abstract*—In recent years, considerable concern has arisen over the security of the master key in ID-based cryptosystems. The Boneh-Franklin scheme proposed a distributed PKG arrangement. The scheme distributes partial master keys among different PKGs using techniques of threshold cryptography. In this paper, we will discuss t-out-of-n key generation and related issues in Boneh-Franklin ID-based cryptosystem.**

***Index Terms*—ID-based cryptosystems, secret splitting, secret sharing, threshold cryptography**

## I. Introduction

The past decade has witnessed growing interest in the pairing-based cryptosystems [8]. One of the important findings is the Weil pairing [8] which can be used to construct an ID-based (public-key) cryptosystem. In this system, the public key of a user is some unique identifiers of the user, such as the e-mail address or the telephone number. However, the corresponding private key is generated by a trusted third party called Private Key Generator (PKG) using the system secret called the master key. Therefore, the PKG must be highly trusted since it can generate the private key of any system user. In some circumstances, we cannot tolerate the full possession of the master key by a single PKG. Therefore, the secret sharing scheme and many of its variants have been proposed to eliminate this weakness. The primary goal of this paper is to examine individual secret sharing schemes. It is hoped that this study could lead to a better understanding of both identity-based cryptosystems and secret sharing schemes. The remainder of the paper is organized as follows: Section Ⅱ reviews bilinear pairings and ID-based cryptosystems. Section Ⅲ introduces the basic secret sharing schemes and how they are used in Boneh-Franklin scheme. In Section Ⅳ, we discuss related issues when deploying secret sharing scheme in Boneh-Franklin scheme. Finally, Section Ⅴ gives our conclusions and presents future works.

* This paper was supported in part by National Science Council of Taiwan under Contract No. NSC98-2221-E-009-079-MY3 and by Industrial Technology Research Institute of Taiwan under Contract No. T2-98020-1.

## II. Preliminaries

In this section, we briefly describe the basic definition and properties of (admissible) bilinear pairings [8] and ID-based cryptosystems [1].

### *A.* (Admissible) Bilinear Pairing

Let $G_1$ be a cyclic additive group generated by $P$, whose order is a prime $q$, and $G_2$ be a cyclic multiplicative group of the same order $q$. Let $a$, $b$ be the elements in $Z_q^*$. A bilinear pairings is a map $e: G_1 \times G_1 \to G_2$ with the following properties:

- Bilinear: $\forall P,Q \in G_1, e(aP,bQ) = e(P,Q)^{ab}$.
- Non-degenerate: $\exists P,Q$ such that $e(P,Q) \neq 1$
- Computable:
  $\forall P,Q \in G_1$, $e(P,Q)$ is efficiently-computable ■

The security of the scheme using bilinear pairings relies on the hardness of the computational problem called Bilinear Diffie-Hellman problem (BDH). No algorithm is now available to be able to efficiently solve BDH. We assume BDH is computationally intractable.

❖ **Bilinear Diffie-Hellman (BDH) Problem:**
Given groups $G_1$ of prime order $q$, a bilinear map $e$ and one generator $P$ of $G_1$, $a, b, c \in Z_q^*$, and given $aP$, $bP$, $cP \in G_1$, compute $e(P,P)^{abc}$ is hard ■

*B.* ID-based cryptosystems

The concept of identity-based cryptosystems is first proposed by Shamir [7]. The scheme utilizes user's identity (ID) as public key rather than meaningless string in the digital certificate. Therefore, the public key can be inferred by the user's identifier and no public key certificate is needed. One user authenticates itself to Private Key Generator (PKG) by showing physical identification tokens and obtains the corresponding private key. The sender encrypts messages using the recipient's ID as public key. The recipient decrypts messages using its private key corresponding to its ID. An identity-based encryption scheme consists of four randomized algorithms: Setup, Extract, Encrypt and Decrypt as shown below and Figure 1.

**Setup:**
The PKG selects system parameters and one system master key.

**Extract:**
The system user Bob authenticates himself to PKG and obtains the private key corresponding to his identity.

**Encrypt:**
The sender Alice produces ciphertext C by encrypting plaintext *M* using Bob's identity.

**Decrypt:**
The receiver Bob decrypts ciphertext C using his private key and obtains plaintext *M*.

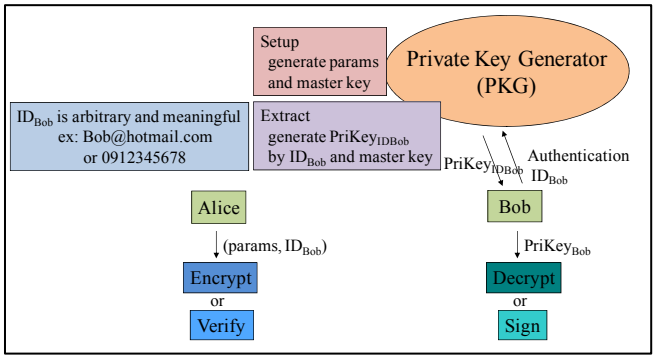


Figure 1 Operations defined in Identity-based cryptosystems

At first, Shamir [6] constructed an identity-based signature scheme (IBS) using RSA functions; however, he was unable to construct an identity-based encryption (IBE) scheme. The first efficient and secure ID-based encryption scheme was proposed by Boneh and Franklin [1] in 2003. (Cock [2] constructs another IBE scheme the same year using integer factorization problem. However, the scheme is inefficient since it encrypts messages in a bit-by-bit fashion yielding a very long ciphertext. Thus, in this paper, we focus only on pairing-based identity-based cryptosystems that is widely used and discussed in the research field.) The main idea of Boneh-Franklin scheme is that the sender and the receiver can retrieve the same session key by using a bilinear map of information available and use this shared session key to protect the messages. The four randomized algorithms are described as below and depicted in Figure 2.

**Setup:** (performed by PKG)
The PKG selects system master key $s_0$ and system parameters $\langle G_1, G_2, H_1, H_2, P, Q_0, e \rangle$

(1) PKG generates cyclic groups $G_1, G_2$ of order $q$, bilinear map $e: G_1 \times G_1 \to G_2$ and generator $P$ in $G_1$

(2) PKG picks master key. $s_0 \in F_q^*$
Computer system public key $Q_0 = s_0 P$

(3) PKG picks cryptographic hash functions
$H_1: \{0,1\}^* \to G_1$, $H_2: G_2 \to \{0,1\}^*$

**Extract:** (performed by PKG)

(1) Bob authenticates himself to PKG using his identity $ID_{Bob}$ and some physical identity credentials like the identification card or driver's license.

(2) Once authenticated by PKG, Bob is given his private key $S_{Bob} = s_o P_{Bob}$ corresponding to his public key $P_{Bob} = H_1(ID_{Bob})$.

**Encrypt:** (performed by encrypter)
Given a plaintext $M$, the recipient's identity $ID_{Bob}$

(1) Alice selects a random number $r \in F_q^*$

(2) Alice computes Bob's public key as
$P_{Bob} = H_1(ID_{Bob})$

(3) Alice produces ciphertext by computing
$C = \langle rP, M \oplus H_2(e(P_{Bob}, Q_0)^r) \rangle$

**Decrypt:** (performed by decrypter)
Given a ciphertext $C' = \langle U, V \rangle$
Bob compute $M' = V \oplus H_2(e(S_{Bob}, U))$

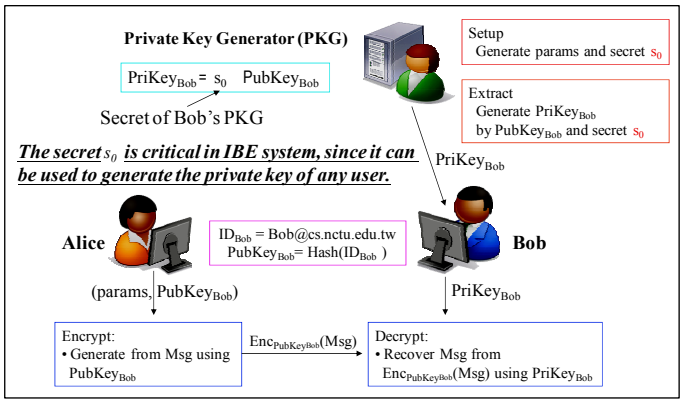


Figure 2 Boneh-Franklin IBE scheme

## III. t-out-of-n Master-key Generation

The security concern of ID-based cryptosystems is the inherent weakness called *key escrow*. It originates from the need of PKG to generate the private key of the system users Therefore, PKG knows the master key of the system; hence it can generate the private key of any system user and perform unauthorized operations such as decryption or signing. In this section, several secret sharing schemes for (master) key generation are introduced and how these schemes integrate with Boneh -Franklin ID-based cryptosystem is also presented.

The first attempt to mend key-escrow weakness is by using *secret splitting* [11]. In this scheme, the secret information is divided into multiple shares which are given to each individual. All of the individuals with a share have to agree on merging all the shares and retrieve the secret. In this case, the number of shares needed to recover the secret equals to the number of total shares, say $n$, which yields an $n$-out-of-$n$ key generation scheme. This scheme is analogous to splitting treasure map into shares and distributing to all the explorers. Only if all the shares join together can they find the position of the treasure. It is noted that the scheme need a trusted third party who perform the map-splitting process. They can designate this job to their chief of the village. An example is shown in Figure 3.

$$\begin{array}{r} Secret: 010010110100 \\ xor)\ Share_1: 101001000110 \\ \hline Share_2: 111011110010 \end{array}$$

Figure 3 Secret splitting between 2 shareholders

If there are two users, say Alice and Bob, who want to share a secret. They first find a trusted third party (TTP). If the secret is represented as binary number of length $m$, TTP first generates a share of length $m$ randomly and performs exclusive-or operation (*XOR*) with the secret to gain another share. TTP then gives one of the shares to Alice and the other to Bob. Alice and Bob can collaborate later to reconstruct the secret by *XOR*ing their shares. None of them knows the whole secret without the help of the other share holder.

More generally, if more shares are needed, say $n$, TTP needs to generates $n-1$ shares of length $m$ randomly and *XOR*ing all of them together with the secret to gain the $nth$ share. TTP then distributes $n$ share to each of the $n$ participants. An example is depicted in Figure 4.

$$\begin{array}{r} Secret: 010010110100 \\ Share_1: 101001000110 \\ Share_2: 011101011011 \\ \vdots \\ xor)Share_{n-1}: 101010011110 \\ \hline Share_n: 111011110010 \end{array}$$

Figure 4 Secret splitting among $n$ shareholders

Sometimes, it might be impractical to gather all the participants to recover the secret. More importantly, if any one of the shares is missing, the secret will no longer be reconstructed. Therefore, the second technique called the *threshold scheme* [10] is used. In this scheme, the secret information is also broken up into multiple shares which are given to two or more individuals. However, the scheme provides a way to reconstruct the secret without the appearance of all the shares. More specifically, if any subsets of shares with size equal to or larger than $t$ are sufficient to reconstruct the secret in the scheme, we call it a $t$-out-of-$n$ threshold scheme.

One version of $t$-out-of-$n$ threshold scheme is proposed by Shamir [6]. It makes use of the idea that two points are sufficient to define a line, three points are sufficient to decide a parabola, and so forth. That is, we need t points to uniquely decide a polynomial of degree $t-1$. An example is depicted in Figure 5.

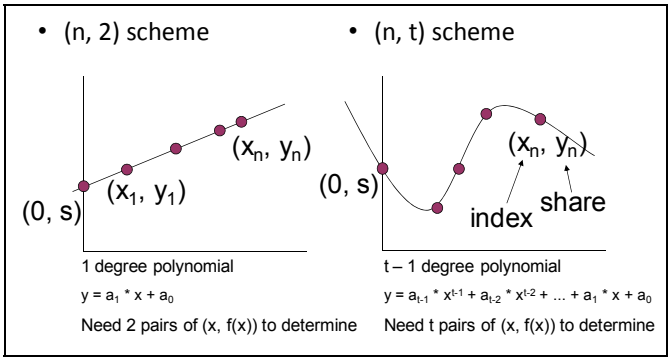


Figure 5 Concept of Shamir secret sharing

If we want to construct a Shamir $(n,t)$ threshold scheme, there are two phases in the scheme.

**Setup:** (Perform by TTP)

(1) TTP chooses a prime of order $q$ in $G_1$.

(2) TTP defines the coefficient $a_0 = s = f(0)$

(3) TTP chooses $a_1 \cdots a_{t-1}$ at random from $Z_q^*$

(4) For all the shareholder $i$,

TTP computes $f(i) = \sum_{j=0}^{t-1} a_j x^j (mod\ q)$

(5) TTP gives $f(i)$ to shareholder $i$ for $i \in \{1,\cdots,n\}$

**Reconstruct:** (Performed by TTP or each of the share-holder)

(1) TTP collects all the shares of the shareholders or each shareholder gives its share to the others.

(2) TTP (or each shareholder) reconstructs secret by computing $s = \sum_{j=1}^{t} (\prod_{i \neq j} \frac{0-i}{j-i} f(i))(mod\ q)$ ■

An example of threshold (*6, 3*) scheme is shown in Figure 6. The upper part describes the *Setup* phase, and the lower two blocks shaded with light yellow color show the *Reconstruction* phase.

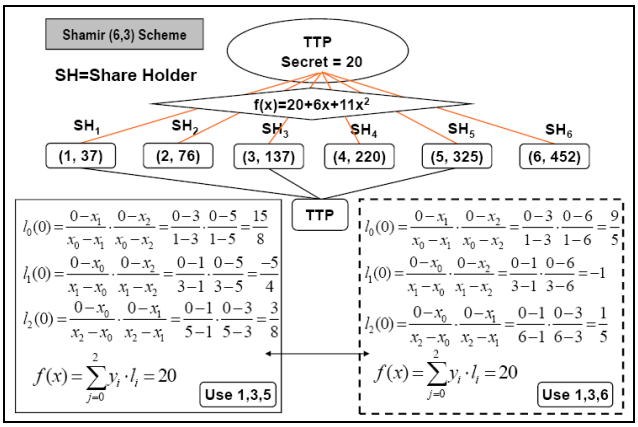


Figure 6 Example of Shamir (6, 3) threshold scheme

### ❖ Modified Boneh-Franklin Scheme

There is one main PKG who acts as TTP and another $n$ PKG $i$, $i = 1,\cdots,n$ who act as shareholder.

**Modified Setup:** (performed by PKG)

The PKG (act as TTP) selects master key $s_0$ and system parameters $\langle G_1, G_2, H_1, H_2, P, Q_0, e \rangle$

(1) PKG generates cyclic groups $G_1, G_2$ of order $q$, bilinear map $e: G_1 \times G_1 \rightarrow G_2$ and generator $P$ in $G_1$

(2) PKG picks cryptographic hash functions $H_1: \{0,1\}^* \rightarrow G_1$, $H_2: G_2 \rightarrow \{0,1\}^*$

(3) PKG chooses prime of order $q$ in $G_1$.

(4) PKG picks master key $s_0 \in F_q^*$ and defines coefficient $a_0 = s_0 = f(0)$

(5) PKG chooses $a_1 \cdots a_{t-1}$ at random from $Z_q^*$

(6) For all the PKG $i$,

PKG computes $f(i) = \sum_{j=0}^{t-1} a_j x^j (mod\ q)$

(7) PKG gives $f(i)$ to PKG $i$ for $i \in \{1,\cdots,n\}$

(8) PKG $i$ publishes its public key $Q_0 = s_0^i P$, where $s_0^i$ is the share of PKG $i$

**Modified Extract:** (performed by PKG $i$)

(1) Bob authenticates himself to PKG $i$ using identity $ID_{Bob}$ and some physical credentials.

(2) Once passed, PKG $i$ issues Bob's private key share $s_0^i P_{Bob}$ corresponding to his public key $P_{Bob} = H_1(ID_{Bob})$

(3) Bob reconstructs its private key by computing

$$S_{Bob} = s_0 P_{Bob} = \sum_{j=1}^{t} (\prod_{i \neq j} \frac{0-i}{j-i} s_0^i P_{Bob})(mod\ q)$$ ■

Once users acquire their own private key, the cryptographic operations are the same as original scheme such as decryption and signing.

## IV. Discussion

In the secret splitting scheme, the security relies heavily on randomness of the shares. In computer, the built-in RND function does not produce true randomness [5]. There are some suggested physical methods which are nevertheless time-consuming,

such as coin flipping, dice throwing, and so on. For computational method, pseudo-random number generators can be used. They create a long sequence of random numbers with satisfying properties; however, the sequence will be repeated eventually. Besides, the safety of each share is also important. At least one share should be kept in secret for each shareholder since less than $n$ shares yield no information about the secret.

In secret sharing [10], the number of threshold should be decided carefully. If the number is set too high, it would be a great burden for the system users, and limits the scale of the system. However, if the threshold number is set too low**,** the capability to resist malicious PKGs would be weakened. Besides, the validity of the share has to be guaranteed. That is, the PKG may give you false shares during *Setup* phase. Some share holders may cheat the others by giving fake shares during *Reconstruct* phase. In Boneh-Franklin paper [1], they mention that these actions can be detected by using the fact that Decision Diffie-Hellman (DDH) is easy in $G_1$. During *Setup* phase, each of the $n$ PKG publishes its $s_i P$ as witness of its share of the master key. When one user requests his private key, he can verify that the response from the $ith$ PKG is valid by testing the following equation:

$$e(s_0^i P_{ID}, P) \stackrel{?}{=} e(P_{ID}, s_0^i P)$$

Thus, if the equation fails to hold for specific $i$, the misbehaving PKG $i$ are caught**;** otherwise, the PKG is considered honest. We call this kind of secret sharing *verifiable* since the share can be further verified after distributed.

In the original threshold scheme, the share will be valid forever; however, some PKGs might be compromised leaving the share revealed. It should also be noted that an ex-employee knows the secret but he is not privileged to possess it. We would like the share to remain valid for only a period of time and can be updated periodically. This is sometimes called a *proactive* property.

There are also adversaries that hinder the others from recovering the secret. This kind of adversary may give false information**,** and even deliberately stop functioning during the transaction. We would like to finish the setup and reconstruction process in the appearance of this kind of malicious adversaries and remove invalid contribution from the result. This is usually achieved by defining computation and communication protocols. If one cannot follow the protocols, it will be ruled out of the group of the shareholder.

What is more, some people may argue that it is the PKG that picks the master key; therefore, it can still generate any private key of the system user. This problem can be eliminated through the following two approaches. One approach is to destroy the master key immediately after the main PKG distributes all the shares. This can be achieved by adding this action into the setup function. The other approach makes use of one protocol called DKG (Distributed Key Generation), $n$ PKG can collaborate to decide the master key and each of them have one share of this secret and none of them know the whole secret. This practice can get rid of one main PKG (act as a TTP) with the cost of a little more operations and message exchange when running the DKG protocol during *Setup* phase. These two approaches can be used to remove the property called *key-escrow* inherited from the ID-based cryptosystem [9].

Sometimes we may need hierarchical management infrastructure. The idea is from the nature of human society, and many global systems have adopted this management mechanism like DNS servers or Certificate Authorities (CA) in certificate-based cryptosystems. Gentry and Silverberg provide the first efficient and secure Hierarchical IBE (HIBE) [3]. If HIBE is needed, what we should do is to integrate secret sharing scheme within HIBE functionality.

## V. Conclusion

In this paper, we have presented the pairing-based ID-based cryptosystems together with threshold cryptography. We also present how to integrate and deploy the threshold schemes within Boneh-Franklin's scheme. There are still many issues that need considering: secret share update, se-

cret share verification, and robustness of the system when malicious TTP or shareholder exists**,** to name but a few. We also consider that the hierarchical IBE may be used when deploying IBE in a large network. There are still many circumstances to figure out and different requirements to meet. To sum up, our research will keep on dedicating to constructing more secure, efficient and escrow-free ID-based cryptosystems in the real world.

## REFERENCE


[1] Boneh D., Franklin M., "Identity Based Encryption from Weil Pairing," SIAM Journal of Computing, 32, 586-615, 2003.

[2] Cocks C., "An Identity Based Encryption Scheme Based on Quadratic Residues," Cryptography and Coding, LNCS 2260, pp. 360-363, 2001.

[3] Gentry C., Silverberg A., "Hierarchical ID-Based Cryptography," ASIACRYPT 2002, Springer-Verlag, LNCS #2501 (2002).

[4] Joux A., "The Weil and Tate Pairings as Building Blocks for Public Key Cryptosystems," ANTS2002, LNCS 2369, pp. 20-32, 2002.

[5] Menezes A., Oorschot P., and Vanstone S., "Handbook of Applied Cryptography," CRC Press, 1996, Chapter 5 Pseudorandom Bits and Sequences

[6] Shamir A., "How to share a secret," Communications of the ACM, 22:612-613, 1979.

[7] Shamir A., "Identity-Based Cryptosystems and Signature Schemes," Proceedings of Crypto '84, pp. 47-53, 1984.

[8] The Pairing-Based Crypto Lounge http://www.larc.usp.br/~pbarreto/pblounge.html

[9] The Risk of Key Recovery, Key escrow, and Trusted Third Party Encryption http://users.telenet.be/d.rijmenants/en/secretsplitting.htm

[10] The Secret Sharing http://en.wikipedia.org/wiki/Secret_sharing

[11] The Secret Splitting http://users.telenet.be/d.rijmenants/en/secretsplitting.htm