2005

Secure key extraction in computer networks

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

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

Xinyi Huang
Nanjing Normal University, xh068@uow.edu.au

Secure key extraction in computer networks

Abstract

Security of computer networks normally relies on a trusted authority who is responsible for setting up the system and distributing cryptographic keys. If the trusted authority is compromised due to an attack, then the security of the entire system will be compromised. In this paper, we will look into this issue in terms of identity-based (or ID-based) cryptography. Since the introduction of identity-based (or ID-based) cryptography in 1984 by Shamir, IDbased cryptography has attracted many research due to its simplicity. However, ID-based cryptography suffers from several drawbacks, namely the requirement of having a secure channel during the key extraction and a complete trust to be placed on a trusted authority (or a so-called Private Key Generator PKG). In this paper, we overcome these problems by proposing a new key extraction algorithm that does not have these two limitations. We are only concentrating on the key extraction problem and hence, our schemes are applicable in any other ID-based scheme that has similar structure, such as Boneh-Franklin ID-based encryption scheme, etc.

Disciplines

Physical Sciences and Mathematics

Publication Details

Susilo, W., Mu, Y. & Huang, X. (2005). Secure key extraction in computer networks. In T. A. Wysocki (Eds.), International Symposium on Digital Signal Processing and Communication Systems & Workshop on the Internet, Telecommunications and Signal Processing (pp. 96-102). Noosa: DSP for Communication Systems.

Secure Key Extraction in Computer Networks

Willy Susilo[1], and Yi Mu[1], and Xinyi Huang[2]
[1]School of Information Technology and Computer Science
University of Wollongong, Wollongong, NSW 2522, Australia
Email: {wsusilo,ymu}@uow.edu.au

[2]College of Mathematics and Computer Science
Nanjing Normal University, Nanjing, P.R. China
Email: xinyinjnu@126.com

Abstract

Security of computer networks normally relies on a trusted authority who is responsible for setting up the system and distributing cryptographic keys. If the trusted authority is compromised due to an attack, then the security of the entire system will be compromised. In this paper, we will look into this issue in terms of identity-based (or ID-based) cryptography. Since the introduction of identity-based (or ID-based) cryptography in 1984 by Shamir, ID-based cryptography has attracted many research due to its simplicity. However, ID-based cryptography suffers from several drawbacks, namely the requirement of having a secure channel during the key extraction and a complete trust to be placed on a trusted authority (or a so-called Private Key Generator PKG). In this paper, we overcome these problems by proposing a new key extraction algorithm that does not have these two limitations. We are only concentrating on the key extraction problem and hence, our schemes are applicable in any other ID-based scheme that has similar structure, such as Boneh-Franklin ID-based encryption scheme, etc.
Keywords: ID-based cryptography, bilinear pairing, key extraction.

1 Introduction

In computer networks, the communications among nodes should be protected against various attacks. To ensure a sound protection, various cryptographic methods can be deployed and some secure cryptographic-key distribution mechanisms should be applied. However, such methods and mechanisms usually rely on a trusted network authority who may be a certificate authority responsible for setting up the system and/or distributing cryptographic keys. The obvious concern is that the entire system could be compromised if the trusted authority is compromised due to an attack from adversaries.

In a traditional certificate based public key cryptography, a key generation procedure invariantly contains a function F, where the public key $\mathcal{PK}$ is defined in terms of the secret key $\mathcal{SK}$ as $\mathcal{PK} = F(\mathcal{SK})$, where F is an efficient and one-way function that maps from the private key space to the public key space. Due to the one-wayness of the function F, the public key $\mathcal{PK}$ always contains a part that *looks* random. In practice, verification of a user's public key is certified with a certificate issued by a certification authority (CA). Any participant who would like to use a public key must first verify the correctness of the corresponding certificate to ensure the validity of the public key. Nevertheless, this issue has lead into a different problem called trust relationship, when many CAs are involved. Public key infrastructure (PKI) is an important infrastructure that is used to manage the trust relationship between entities in a hierarchical manner. As a consequence, certificate-based public key cryptosystems require a large amount of storage and computing time to store and verify certificates.

In [13], Shamir suggested that the public keys are chosen from the users' identities, such as e-mail address, IP address, etc. and hence, it is named an identity-based public-key cryptography (or ID-based cryptography, for short). In an ID-based cryptography, the private key is computed by a *key extraction* algorithm, which is defined as $\mathcal{SK} = F(\mathsf{master\text{-}key}, \mathsf{ID})$. Note that the ID is the public key $\mathcal{PK}$ in a traditional public key cryptography. The so-called master-key is the long term secret key that is owned by the Private Key Generator (PKG). Note that since ID is used to replace the usual public key, then the authenticity of the public key is no longer a problem. However, there is a new protocol introduced in this system, called the *key extraction* algorithm, which is a *service* offered by a trusted PKG to system wide users. This service is essentially an authentication service: the resulting private key from this algorithm provides the key owner with a credential for his/her ID-based

public key to be recognized and used by other users in the system. Essentially, before the secret key for an identity is released, the PKG must conduct a thorough check of the identity information of the user. This check may include some physical means of identification, which is similar to the identification check before a CA issues a public key certificate to a user in a traditional public key cryptography setting.

In the same paper [13], Shamir provided a concrete ID-based signature scheme, but he questioned the existence of an ID-based encryption (IBE) scheme. It was not known whether IBE scheme exists, until shown recently by Boneh and Franklin that an efficient IBE can be constructed from bilinear pairings on "weak" elliptic curves [2]. Interestingly, the key extraction algorithm in [2] is known later on as a short signature scheme based on bilinear pairing [3].

Problems with ID-based Cryptography

Despite of all the nice features that ID-based cryptography can offer, ID-based cryptography suffers from several drawbacks.

Firstly, an inherent problem in ID-based cryptography is the *key escrow* problem. Since all user's private keys are generated by the PKG, they must place an absolute trust to the PKG. The trust must be absolute, complete and unconditional. Essentially, this means that the PKG can read all the private communications or forge all of their signatures. Hence, as noted by Shamir in his seminal paper in [13], ID-based cryptography is suitable in a closed environment, for example in an organization environment in which the employer has the complete ownership of the information communicated to and from the employees, then the employer can play the role of PKG.

We note that this is not a new issue. This problem has been addressed in several work in the literature by employing multiple authority approach [2, 5] or by using some user-chosen secret information [1, 14, 7, 10]. It was noted in [2] that if the master-key is distributed to multiple PKGs and a private key is computed in a threshold manner, then the key escrow problem of a single PKG can be prevented. However, in practice, having many PKGs who will generate a secret key to a user is quite a burden. A different approach by generating a new private key by adding multiple private keys was proposed in [5]. Nevertheless, in this scheme, PKGs have no countermeasure against user's illegal usage. A certificate-based encryption (and later, signature scheme) was proposed by Gentry in [7]. In this scheme, the user's secret key is computed by adding some user-chosen secret information, but in fact, it becomes a certificate-based scheme losing the advantage of ID-based cryptography. Similar, but different, approach was taken by Al-Riyami and Paterson in [1] by proposing a certificateless cryptography. In certificateless cryptography, the need of certificate has been completely removed. After the key extraction algorithm is invoked (by an interactive protocol between the PKG and the user), the user needs to add his user-chosen secret information and later on, publish his public key. Again, in this scheme, the advantage of ID-based cryptography has been removed since the public key is required to be published by the user. In a more recent work by Lee et. al. [11], the secret key is constructed by a PKG together with several key privacy authorities (KPAs). In practice, this approach is not very practical.

The second problem in ID-based cryptography is still related to the key extraction protocol, namely the requirement to have a *secure channel* between the PKG and the user during the protocol. We note that the resulting key extracted from this protocol is the secret key that is only known by the user. If this key is somehow leaked, then anyone can either sign on behalf of the user or read the message intended to the user. Therefore, the existence of a secure channel between the PKG and the user is essential. In practice, a secure channel needs to be established by using a public key infrastructure, and hence, it returns to the problem of having a traditional cryptosystem (by requiring each party to provide certificate, etc.). This problem has recently been addressed Sui *et. al.* [6], and their proposed solution is by adding randomness to the identity that is sent by the user during the key extraction algorithm. However, we note that this approach is not practical and somehow it violates the idea of the key extraction algorithm itself. As noted earlier, in the key extraction algorithm, the PKG needs to identify the correctness of the ID, and in the approach that was taken in [6], the PKG cannot verify the authenticity of the ID since the information that is sent by the user is randomized.

Our Contribution

In this paper, we revisit the notion of key extraction algorithm and solve the two existing problems in ID-based cryptography mentioned in the previous section. In particular, our key extraction algorithm will not require the user to place his/her trust to a single PKG and more importantly, our key extraction algorithm does not require the existence of a secure channel. Moreover, our key extraction algorithm will work with any previously known schemes, such as Boneh-Franklin's IBE scheme [2]. To the best of our knowledge, this is the first scheme that solves the two problems in ID-based cryptography at the same time.

Organization of The Paper

In the next section, we provide some required back-

ground concepts and some related works in this area. Then, we describe our key extraction algorithm in stages. In section 3, we describe our first key extraction protocol. In this protocol, we have successfully removed the need of a secure channel. However, the PKG's key escrow problem still exists. In section 4, we improve our first scheme by removing escrowing problem. The only assumption that we make is the PKGs will not collude against the user. Section 5 concludes the paper.

2 Preliminaries

In this section, we briefly review the basic concepts on bilinear pairing together with some complexity assumptions, while introducing the notations used throughout this paper.

2.1 Basic Concepts on Bilinear Pairings

Let $\mathbb{G}_1, \mathbb{G}_2$ be cyclic additive groups generated by P_1, P_2, respectively, whose order are a prime q. Let $\mathbb{G}_M$ be a cyclic multiplicative group with the same order q. We assume there is an isomorphism $\psi : \mathbb{G}_2 \to \mathbb{G}_1$ such that $\psi(P_2) = P_1$. Let $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_M$ be a bilinear mapping with the following properties:

1. *Bilinearity*: $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ for all $P \in \mathbb{G}_1, Q \in \mathbb{G}_2, a, b, \in \mathbb{Z}_q$.
2. *Non-degeneracy*: There exists $P \in \mathbb{G}_1, Q \in \mathbb{G}_2$ such that $\hat{e}(P, Q) \neq 1$.
3. *Computability*: There exists an efficient algorithm to compute $\hat{e}(P, Q)$ for all $P \in \mathbb{G}_1, Q \in \mathbb{G}_2$.

For simplicity, hereafter, we set $\mathbb{G}_1 = \mathbb{G}_2$ and $P_1 = P_2$. We note that our scheme can be easily modified for a general case, when $\mathbb{G}_1 \neq \mathbb{G}_2$.

A Bilinear pairing instance generator is defined as a probabilistic polynomial time algorithm $\mathcal{IG}$ that takes as input a security parameter ℓ and returns a uniformly random tuple $param = (p, \mathbb{G}_1, \mathbb{G}_M, \hat{e}, P)$ of bilinear parameters, including a prime number p of size ℓ, a cyclic additive group $\mathbb{G}_1$ of order q, a multiplicative group $\mathbb{G}_M$ of order q, a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_M$ and a generator P of $\mathbb{G}_1$. For a group $\mathbb{G}$ of prime order, we denote the set $\mathbb{G}^* = \mathbb{G} \setminus \{\mathcal{O}\}$ where $\mathcal{O}$ is the identity element of the group.

2.2 Complexity Assumptions

Definition 1 Bilinear Diffie-Hellman (BDH) Problem.
Given a randomly chosen $P \in \mathbb{G}_1$, as well as aP, bP and cP (for unknown randomly chosen $a, b, c \in \mathbb{Z}_q$), compute $\hat{e}(P, P)^{abc}$.

Definition 2 Decisional Diffie-Hellman (DDH) Problem.
Given a randomly chosen $P \in \mathbb{G}_1$, as well as aP, bP, cP, for some $a, b, c \in \mathbb{Z}_q^*$, decide whether $c \stackrel{?}{=} ab$ holds with equality.

It is well-known that DDH problem in $\mathbb{G}_1$ is easy, by performing MOV reduction, that states the discrete logarithm problem (DLP) in $\mathbb{G}_1$ is no harder than the DLP in $\mathbb{G}_2$.

Definition 3 Decisional Bilinear Diffie-Hellman (DBDH) Problem.
Given a randomly chosen $P \in \mathbb{G}_1$, as well as aP, bP, cP and r, for some $a, b, c, \in \mathbb{Z}_q^*$ and $r \in \mathbb{G}_M$, decide whether $r \stackrel{?}{=} \hat{e}(P, P)^{abc}$ holds with equality.

Definition 4 Decisional Hash Bilinear Diffie-Hellman (DHBDH) Problem.
Given a randomly chosen $P \in \mathbb{G}_1$, as well as aP, bP, cP and r, for some $a, b, c, r \in \mathbb{Z}_q^*$, decide whether $r \stackrel{?}{=} h(\hat{e}(P, P)^{abc})$, where $h : \mathbb{G}_M \to \mathbb{Z}_q^*$, holds with equality.

Definition 5 Decisional Hash Bilinear Diffie-Hellman Assumption.
If $\mathcal{IG}$ is a DHBDH parameter generator, the advantage $\mathtt{Adv}_{\mathcal{IG}}(\mathcal{A})$ that an algorithm $\mathcal{A}$ has in solving the DHBDH problem is defined to be the probability that the algorithm $\mathcal{A}$ outputs "yes" when $r \stackrel{?}{=} h(\hat{e}(P, P)^{abc})$ holds on inputs $\mathbb{G}_1, \mathbb{G}_M, \hat{e}, P, aP, bP, cP$, where $(\mathbb{G}_1, \mathbb{G}_M, \hat{e})$ is the output of $\mathcal{IG}$ for sufficiently large security parameter ℓ, P is a random generator of $\mathbb{G}_1$ and a, b, c are random elements of $\mathbb{Z}_q$ and $h : \mathbb{G}_M \to \mathbb{Z}_q^*$. The DHBDH assumption is that $\mathtt{Adv}_{\mathcal{IG}}^{DHBDH}(\mathcal{A})$ is negligible for all efficient algorithms $\mathcal{A}$.

2.3 Boneh-Franklin's ID-based Encryption Scheme

Using the bilinear pairing, an ID-based encryption (IBE) scheme can be designed. For completeness, we review the construction of an IBE scheme due to Boneh-Franklin [2] as follows.
In general, there are four algorithms in ID-based cryptosystem as follows.

- `Setup`. A deterministic algorithm that is run by a trusted authority to generate global system parameters and *master key*.
- `Extract`. A deterministic algorithm that is run by a trusted authority on inputting the

master key together with an arbitrary bit string $\mathsf{ID} \in \{0,1\}^*$, to generate the user's private key $\mathcal{S}_{\mathsf{ID}}$. That is, $\mathcal{S}_{\mathsf{ID}} \leftarrow \mathsf{Extract}(\mathsf{ID})$.

- `Encrypt`. A probabilistic algorithm that encrypts a message under the public identity ID. That is, $C \leftarrow \mathsf{IDEncrypt}(m, \mathsf{ID})$.

- `Decrypt`. A deterministic algorithm that receives a ciphertext and a private key $\mathcal{S}_{\mathsf{ID}}$, to generate the corresponding plaintext. That is,

$$m \leftarrow \mathsf{IDDecrypt}(C, \mathcal{S}_{\mathsf{ID}}).$$

The ID-based cryptosystem proposed by Boneh and Franklin is as follows.

- `Setup`. PKG generates two groups $(\mathbb{G}_1, +)$ and $(\mathbb{G}_M, \cdot)$ of prime order q and a mapping pair $e : (\mathbb{G}_1, +)^2 \rightarrow (\mathbb{G}_M, \cdot)$. He also selects an arbitrary generator $P \in \mathbb{G}_1$. Then, he picks $s \in \mathbb{Z}_q$ and set $P_{pub} = sP$, where s denotes the master key. Finally, two cryptographically strong hash functions are selected: $F : \{0,1\}^* \rightarrow \mathbb{G}_1$, $H : \mathbb{G}_M \rightarrow \{0,1\}^n$, where n denote the size of the plaintext message space. The system parameters and their descriptions are $(\mathbb{G}_1, \mathbb{G}_M, \hat{e}, q, P, P_{pub}, F, H)$.

- `Extract`. After performing physical identification of Bob and making sure the uniqueness of ID, PKG generates Bob's secret key as follows. PKG computes $\mathsf{Q} = F(\mathsf{ID})$ and $\mathcal{S}_{\mathsf{ID}} = s\mathsf{Q}$. $\mathcal{S}_{\mathsf{ID}}$ is Bob's secret key.

- `Encrypt`. To send an encrypted message to Bob, Alice first obtains the system parameter and Bob's identity to compute $\mathsf{Q} = F(\mathsf{ID})$. Then, to encrypt a message $m \in \{0,1\}^n$, Alice picks $r \in Z_q$ and computes $g_{\mathsf{ID}} = \hat{e}(\mathsf{Q}, rP_{pub})$ and $C = (rP, m \oplus H(g_{\mathsf{ID}}))$. The ciphertext is $C = (rP, m \oplus H(g_{\mathsf{ID}}))$.

- `Decrypt`. Let $C = (U, V)$ be a ciphertext received by Bob. To decrypt C using his private key $\mathcal{S}_{\mathsf{ID}}$, Bob computes $V \oplus H(\hat{e}(\mathcal{S}_{\mathsf{ID}}, U))$.

The security of this scheme relies on BDH assumption in the random oracle model [2].

In this paper, we are concentrating on the key extraction algorithm `Extract`. We aim to demonstrate a key extraction algorithm that does not suffer from the problems mentioned above. To the best of our knowledge, our scheme is the first scheme that offers a key extraction algorithm without having the two problems mentioned earlier.

3 The Basic Scheme

In this section, we present our basic key extraction algorithm that preserves the advantages of ID-based cryptography. Intuitively, our scheme works as follows. There are two PKGs in the system. It is assumed that the PKGs will not collude against the user. Each user needs to visit both PKGs to obtain his/her secret key (via the key extraction algorithm). The secret key is delivered by the PKG to the user via a public channel (and hence, no secure channel required). We employ the technique from Joux's tripartite Diffle-Hellman key agreement [8, 9] and a blind signature scheme [4, 12] to construct our proposed scheme. The aim of this basic scheme is to eliminate the need of any secure channel.

3.1 Model

As mentioned in [2], to avoid a complete trust to a single PKG, multiple PKGs can be employed. In our model, we strictly use two PKGs, namely PKG_0 and PKG_1. We assume that both PKGs do not collude against the user.

In the key extraction algorithm, each user visits each PKG to obtain his/her partial secret keys. The secret keys are delivered via a public channel (and hence, no secret channel is required). Only the user, who can retrieve the secret key, can use the secret key in any ID-based cryptosystem.

Remarks: Though our scheme does not need secret channel, in practice we may still want to employ such a channel for simplicity.

3.2 The Scheme

The system parameter for the scheme is identical to Boneh-Franklin's scheme, namely $(\mathbb{G}_1, \mathbb{G}_2, \hat{e}, q, P, P_{pub}, F, H)$, which is publicly available. Each PKG selects a secret key $s_i \in \mathbb{Z}_q^*$ and computes

$$P_{pub_i} = s_i P$$

for $i = 0, 1$. Hence, the tuple $(P, P_{pub_0}, P_{pub_1})$ is part of the system parameter to replace (P, P_{pub}) in a single PKG setting. Define an additional hash function $h : \mathbb{G}_2 \rightarrow \mathbb{Z}_q^*$. Hence, the complete system parameter is $(\mathbb{G}_1, \mathbb{G}_2, \hat{e}, q, P, P_{pub_0}, P_{pub_1}, F, H, h)$.

Key Extraction Algorithm

There are three steps in this algorithm.

Step 1.

Let ID denote a user's uniquely identifiable identity. The user performs the following.

1. Select a random $r \in \mathbb{Z}_q^*$.
2. Compute $R = rP$.
3. Send ID and R to the PKG_i.

Step 2.
Upon each user's visit, the PKG_i, $i = 0, 1$, will perform the following.

1. Compute $\mathsf{Q} = F(\mathsf{ID})$, where $\mathsf{Q} \in \mathbb{G}_1$, and Q is the user's unique ID-based public key.
2. Compute the user's secret key as
$$\mathcal{S}_{\mathsf{ID}}^i = h(\hat{e}(P_{pub_{i\oplus 1}}, R)^{s_i}) \cdot s_i\mathsf{Q}$$
3. Send $\mathcal{S}_{\mathsf{ID}}^i$ to the user via a public channel (note that this value can be just broadcasted).

Step 3.
After visiting both PKGs, the user obtains two secret keys namely $\mathcal{S}_{\mathsf{ID}}^0$ and $\mathcal{S}_{\mathsf{ID}}^1$, where
$$\mathcal{S}_{\mathsf{ID}}^0 = h(\hat{e}(P_{pub_1}, R)^{s_0}) \cdot s_0\mathsf{Q}$$
and
$$\mathcal{S}_{\mathsf{ID}}^1 = h(\hat{e}(P_{pub_0}, R)^{s_1}) \cdot s_1\mathsf{Q}$$
The authenticity of the partial secret key $\mathcal{S}_{\mathsf{ID}}^i$ can be verified by testing whether
$$\hat{e}(\mathcal{S}_{\mathsf{ID}}^i, P) \stackrel{?}{=} \hat{e}(\mathsf{Q}, P_{pub_i})^{h(\hat{e}(P_{pub_0}, P_{pub_1})^r)}$$
holds with equality. If both partial signatures are authenticated, then the user will compute
$$\mathcal{S}_{\mathsf{ID}} = \frac{\mathcal{S}_{\mathsf{ID}}^0 + \mathcal{S}_{\mathsf{ID}}^1}{h(\hat{e}(P_{pub_0}, P_{pub_1})^r)}$$
to obtain his secret key. The user's secret key is of the form
$$\mathcal{S}_{\mathsf{ID}} = s_0\mathsf{Q} + s_1\mathsf{Q}$$
Correctness.
The correctness of the secret key extraction is justified as follows.

$$\begin{aligned}
\mathcal{S}_{\mathsf{ID}} &= \frac{\mathcal{S}_{\mathsf{ID}}^0 + \mathcal{S}_{\mathsf{ID}}^1}{h(\hat{e}(P_{pub_0}, P_{pub_1})^r)} \\
&= \frac{h(\hat{e}(P,P)^{s_0 s_1 r}) \cdot s_0\mathsf{Q} + h(\hat{e}(P,P)^{s_0 s_1 r}) s_1\mathsf{Q}}{h(\hat{e}(P,P)^{s_0 s_1 r})} \\
&= \frac{h(\hat{e}(P,P)^{s_0 s_1 r})(s_0\mathsf{Q} + s_1\mathsf{Q})}{h(\hat{e}(P,P)^{s_0 s_1 r})} \\
&= s_0\mathsf{Q} + s_1\mathsf{Q} \\
&= (s_0 + s_1)\mathsf{Q}
\end{aligned}$$

∎

Remarks:

- We note that the secret key obtained from our extraction algorithm can be used in any other ID-based system that has similar structure, such as Boneh-Franklin's IBE system, with respect to the public key P_{pub_0} and P_{pub_1}.
- To illustrate the encryption algorithm in Boneh-Franklin's IBE, the encryption and decryption algorithms are defined as follows.
  - `Encrypt`. To send an encrypted message to Bob, Alice first obtains the system parameter and Alice's identity to compute $\mathsf{Q} = F(\mathsf{ID})$. Then, to encrypt a message $m \in \{0,1\}^n$, Bob picks $s \in Z_q$ and computes $g_{\mathsf{ID}} = \hat{e}(\mathsf{Q}, sP_{pub_0} + sP_{pub_1})$ and $C = (sP, m \oplus H(g_{\mathsf{ID}}))$. The ciphertext is $C = (sP, m \oplus H(g_{\mathsf{ID}}))$.
  - `Decrypt`. Let $C = (U, V)$ be a ciphertext received by Bob. To decrypt C using his private key $\mathcal{S}_{\mathsf{ID}}$, Bob computes $V \oplus H(\hat{e}(\mathcal{S}_{\mathsf{ID}}, U))$.

The correctness of the decryption algorithm is obvious and hence, it is omitted.

3.3 Security Analysis

Theorem 1 *Our proposed key extraction algorithm does not require the existence of any secure channel.*

Proof. The partial secret key provided by each PKG_i is in the form of
$$\mathcal{S}_{\mathsf{ID}}^i = h(\hat{e}(P_{pub_{i\oplus 1}}, R)^{s_i}) \cdot s_i\mathsf{Q}$$
The only participant who can derive the partial secret key $s_i\mathsf{Q}$ from $\mathcal{S}_{\mathsf{ID}}^i$ is only the user who knows r, where $R = rP$. This is due to Joux's tripartite Diffie-Hellman key agreement [8]. Hence, no secure channel is required. ∎

Lemma 1 *In our first scheme, each PKG can perform key escrow to the user's secret key.*

Proof. The partial secret key is derived from a tripartite key agreement among PKG_0, PKG_1 and the user. Hence, when PKG_0 sends the user's partial secret key $\mathcal{S}_{\mathsf{ID}}^0$, PKG_1 can always obtain this partial secret key as well. We will show how to eliminate this problem in our second scheme. ∎

Theorem 2 *Our key extraction algorithm is unforgeable if DHBDH problem is hard.*

Proof. To show the unforgability of our scheme, let us assume there is an attacker $\mathcal{A}$ who obtains the partial key extraction from the PKG and successfully retrieves the partial secret key without knowing either the PKGs' secret keys nor the user's secret key. We will show how to construct an algorithm $\mathcal{B}$ that will invoke the attacker $\mathcal{A}$ in the simulation. The purpose of $\mathcal{B}$ is to solve an instance of DHBDH problem.

For clarity of the presentation, let us recall the ability of the attacker $\mathcal{A}$ together with the goal of

the algorithm $\mathcal{B}$. Having received a partial secret key $\mathcal{S}_{\mathsf{ID}}^i$, where

$$\mathcal{S}_{\mathsf{ID}}^i = h(\hat{e}(P_{pub_{i\oplus 1}}, R)^{s_i}) \cdot s_i\mathsf{Q}$$

$\mathcal{A}$ can derive $s_i\mathsf{Q}$ without knowing any of the secret key involved. The purpose of the algorithm $\mathcal{B}$ is to decide whether $r \stackrel{?}{=} h(\hat{e}(P,P)^{abc})$ holds for the tuple (aP, bP, cP, r), where $a, b, c, r \in \mathbb{Z}_q^*$. The simulation is as follows.

Preparation.

$\mathcal{B}$ prepares the public keys of the PKGs as $P_{pub_0} = aP$ and $P_{pub_1} = bP$. Then, the public parameter $(\mathbb{G}_1, \mathbb{G}_2, \hat{e}, q, P, P_{pub_0}, P_{pub_1}, F, H)$ is provided to $\mathcal{A}$.

Key Extraction.

To simulate the key extraction, $\mathcal{B}$ computes

$$\mathcal{S} = r \cdot R$$

where $R \in \mathbb{G}_1^*$ is chosen randomly. Then, $\mathcal{B}$ sends $\mathcal{S}$ together with cP to $\mathcal{A}$. Receiving $\mathcal{S}, cP$ from $\mathcal{B}$, $\mathcal{A}$ can output δ, where

$$\delta = \frac{\mathcal{S}}{h(\hat{e}(P,P)^{abc})}.$$

Then, $\mathcal{B}$ will perform the following.

- If $\delta \stackrel{?}{=} R$ holds, then output "yes", to indicate that $r \stackrel{?}{=} h(\hat{e}(P,P)^{abc})$ holds with equality.
- Otherwise, output "no".

The success probability of $\mathcal{B}$ is the same as $\mathcal{A}$. Hence, we obtain the contradiction and complete the proof. ∎

4 The Second Scheme: Removing the Key Escrow

In this section, we extend our basic scheme to remove the ability of each PKG to derive the user's partial secret key during the partial key extraction, and hence, the key escrowing problem can be eliminated. The scheme is very similar to our first scheme mentioned in the previous section, with several modifications as follows.

Key Extraction Algorithm

There are three steps in this algorithm.

Step 1. The same as our first scheme.

Step 2.

Upon each user's visit, the PKG_i, $i = 0, 1$, will perform the following.

1. Compute $\mathsf{Q} = F(\mathsf{ID})$, where $\mathsf{Q} \in \mathbb{G}_1$, and Q is the user's unique ID-based public key.
2. Select a random point $T_i \in \mathbb{G}_1$.
3. Compute the user's secret key as
$$\mathcal{S}_{\mathsf{ID}}^i = h(\hat{e}(T_i, R)^{s_i}) \cdot s_i\mathsf{Q}$$
4. Send $\mathcal{S}_{\mathsf{ID}}^i$ and T_i to the user via a public channel (these values can be just broadcasted).

Step 3.

After visiting both PKGs, the user obtains two secret keys namely $\mathcal{S}_{\mathsf{ID}}^0$ and $\mathcal{S}_{\mathsf{ID}}^1$, where

$$\mathcal{S}_{\mathsf{ID}}^0 = h(\hat{e}(T_0, R)^{s_0}) \cdot s_0\mathsf{Q} \text{ and}$$

$$\mathcal{S}_{\mathsf{ID}}^1 = h(\hat{e}(T_1, R)^{s_1}) \cdot s_1\mathsf{Q}$$

together with T_0 and T_1. The authenticity of the partial secret key $\mathcal{S}_{\mathsf{ID}}^i$ can be verified by testing whether

$$\hat{e}(\mathcal{S}_{\mathsf{ID}}^i, P) \stackrel{?}{=} \hat{e}(\mathsf{Q}, P_{pub_i})^{h(\hat{e}(T_i, P_{pub_i})^r)}$$

holds with equality. If both partial signatures are valid, then the user will compute

$$\mathcal{S}_{\mathsf{ID}} = \frac{\mathcal{S}_{\mathsf{ID}}^0}{h(\hat{e}(T_0, P_{pub_0})^r)} + \frac{\mathcal{S}_{\mathsf{ID}}^1}{h(\hat{e}(T_1, P_{pub_1})^r)}$$

to obtain his complete secret key. The user's secret key is of the form

$$\mathcal{S}_{\mathsf{ID}} = s_0\mathsf{Q} + s_1\mathsf{Q}$$

4.1 Security Analysis

Theorem 3 *Our second scheme does not permit each PKG to reveal the partial secret key sent by the other PKG.*

Proof. Unlike our first scheme, our second scheme removes the ability of the other PKG to read the partial secret key sent over the public channel. This is due to the use of two randomly chosen points by each PKG. ∎

Theorem 4 *Our second scheme is unforgeable iff DBDH problem is hard.*

Proof. The proof is very similar to the proof of Theorem 2 and therefore it is omitted. ∎

Remarks:

- In our second scheme, each PKG cannot retrieve the user's partial secret key (and hence, the complete secret key). However, it was noted in [13] that key escrow is sometimes needed, for instance under a court order. In our scheme, we assume that the PKGs will not collude against the user. Nevertheless, under a court order, both PKGs can be called upon and hence, they can cooperatively decrypt a ciphertext to recover the required plaintext that is encrypted using the private key (in the IBE scheme).

4.2 Extension to Multiple PKGs

It is easy to see that our second scheme can be easily extended to support more than two PKGs. The idea is illustrated as follows. During the key extraction algorithm, each PKG_i will compute the user's secret key as

$$\mathcal{S}_{\mathsf{ID}}^i = h(\hat{e}(T_i, R)^{s_i}) \cdot s_i \mathsf{Q}$$

for a random $T_i \in \mathbb{G}_1$, and finally send $\mathcal{S}_{\mathsf{ID}}^i$ and T_i to the user via a public channel. After collecting the partial secret keys from n PKGs, the user can retrieve his/her public key by computing

$$\mathcal{S}_{\mathsf{ID}} = \sum_{i=1}^{n} \frac{\mathcal{S}_{\mathsf{ID}}^i}{h(\hat{e}(T_i, P_{pub_i})^r)}.$$

The computed secret key can be verified as follows

$$\hat{e}(\mathcal{S}_{\mathsf{ID}}, P) \stackrel{?}{=} \prod_{i=1}^{n} \hat{e}(\mathsf{Q}, P_{pub_i})$$

where P_{pub_i} denotes the PKG_i's public key.

5 Conclusion

In an ID-based cryptography, key extraction algorithm assumes that the PKG is trusted and can always perform key-escrow. Additionally, a *secure channel* between the PKG and the user is required during the key extraction algorithm. In this paper, we firstly proposed a key extraction algorithm that overcomes these two limitations. We provide our security proof under a standard assumption.

References

[1] S. S. Al-Riyami and K. G. Paterson. Certificateless Public Key Cryptography. *Advances in Cryptography - Asiacrypt 2003, Lecture Notes in Computer Science 2894*, pages 452 – 473, Springer-Verlag, Berlin, 2003.

[2] D. Boneh and M. Franklin. Identity-based Encryption from the Weil Pairing. *Lecture Notes in Computer Science 2139*, pages 213 – 229, Springer-Verlag, Berlin, 2001.

[3] D. Boneh, B. Lynn, and H. Shacham. Short Signatures from the Weil Pairing. *Advanced in Cryptology - Asiacrypt 2001, Lecture Notes in Computer Science 2248*, pages 514 – 532, Springer-Verlag, Berlin, 2001.

[4] D. Chaum. Blind Signatures for Untraceable Payments. *Advances in Cryptology - Crypto '82*, pages 199 – 203, Springer-Verlag, Berlin, 1983.

[5] L. Chen, K. Harrison, N. P. Smart, and D. Soldera. Applications of Multiple Trust Authorities in Pairing Based Cryptosystems. *InfraSec 2002, Lecture Notes in Computer Science 2437*, pages 260 – 275, Springer-Verlag, Berlin, 2002.

[6] A. fen Sui, S. S. Chow, L. C. Hui, S. Yiu, K. Chow, W. Tsang, C. Chong, K. Pun, and H. Chan. Secure and Anonymous Identity-Based Key Issuing without Secure Channel. *Cryptology ePrint Archive, Report 2004/322*, 2004.

[7] C. Gentry. Certificate-Based Encryption and the Certificate Revocation Problem. *Advances in Cryptology - Eurocrypt 2003, Lecture Notes in Computer Science 2656*, pages 272 – 293, Springer-Verlag, Berlin, 2003.

[8] A. Joux. A One Round Protocol for Tripartite Diffle-Hellman. *Proceedings of ANTS IV, Lecture Notes in Computer Science 1838*, pages 385 – 394, Springer-Verlag, Berlin, 2000.

[9] A. Joux. A One Round Protocol for Tripartite Diffle-Hellman. *Journal of Cryptology, Volume 17 Number 4*, pages 263 – 276, Springer-Verlag, Berlin, 2004.

[10] B. G. Kang, J. H. Park, and S. G. Hahn. A Certificate-Based Signature Scheme. *CT-RSA 2004 2964*, pages 99 – 111, Springer-Verlag, Berlin, 2004.

[11] B. Lee, C. Boyd, E. Dawson, K. Kim, J. Yang, and S. Yoo. Secure Key Issuing in ID-based Cryptography. *Australasian Information Security Workshop (AISW 2004)*, pages 60 – 74, 2004.

[12] D. Pointcheval and J. Stern. Security Arguments for Digital Signatures and Blind Signatures. *Journal of Cryptology*, 13(3): 361 – 396, Springer-Verlag, Berlin, 2000.

[13] A. Shamir. Identity-based Cryptosystems and Signature Schemes. *Advances in Cryptology - Crypto '84, Lecture Notes in Computer Science 196*, pages 47 – 53, Springer-Verlag, Berlin, 1985.

[14] D. H. Yum and P. J. Lee. Generic Construction of Certificateless Signature. *Information Security and Privacy, ACISP 2004, Lecture Notes in Computer Science 3108*, pages 200 – 211, Springer-Verlag, Berlin, 2004.