Efficient Multi-receiver Identity-Based Encryption and Its Application to Broadcast Encryption

Joonsang Baek, Reihaneh Safavi-Naini, and Willy Susilo

Centre for Information Security Research,
School of Information Technology and Computer Science,
University of Wollongong,
Wollongong NSW 2522, Australia
{baek,rei,wsusilo}@uow.edu.au

Abstract. In this paper, we construct an efficient "multi-receiver identity-based encryption scheme". Our scheme only needs one (or none if precomputed and provided as a public parameter) pairing computation to encrypt a single message for n receivers, in contrast to the simple construction that re-encrypts a message n times using Boneh and Franklin's identity-based encryption scheme, considered previously in the literature. We extend our scheme to give adaptive chosen ciphertext security. We support both schemes with security proofs under precisely defined formal security model. Finally, we discuss how our scheme can lead to a highly efficient public key broadcast encryption scheme based on the "subset-cover" framework.

Keywords: Multi-Receiver Identity-Based Encryption, Formal Security Analysis, Public Key Broadcast Encryption

1 Introduction

Motivation. Assume that there are n receivers, numbered $1, \ldots, n$, and that each of them keeps a private and public key pair denoted by (sk_i, pk_i). A sender then encrypts a message M_i directed to receiver i using pk_i for $i = 1, \ldots, n$ and sends $(C_1, \ldots, C_n)$ as a ciphertext. Upon receiving the ciphertext, receiver i extracts C_i and decrypts it using its private key sk_i. This setting of public key encryption is generally referred to as "multi-receiver (recipient) public key encryption" in the literature [2, 3, 16].

Now consider a situation where "Identity-Based Encryption (IBE)" [7, 10] is incorporated to the above setting. In this setting, the public key pk_i is replaced by receiver i's identifier information (identity) $\mathtt{ID}_i$, which will be used as encryption key. Receiver i has a private key associated with $\mathtt{ID}_i$, obtained from the trusted Private Key Generator (PKG), so that it can correctly decrypt C_i. This setting, which we call "multi-receiver identity-based encryption", is a main theme of this paper.

As one can easily see, any multi-receiver public key encryption scheme can be transformed into a natural *broadcast encryption* scheme: Receivers are given

S. Vaudenay (Ed.): PKC 2005, LNCS 3386, pp. 380–397, 2005.

private/public key pairs which may be generated by the sender. A *single* message M is then encrypted by running the multi-receiver encryption algorithm with all messages M_i for $i = 1, \ldots, n$ set to M to produce a ciphertext which is sent to all receivers.

In the non-identity-based setting, the above broadcast encryption has received a great attention from the research community, while relatively little research has been done on the identity-based setting. One may, however, argue that the natural construction of a broadcast encryption scheme derived from the multi-receiver public key encryption scheme can trivially be transformed into the one in the identity-based setting. That is, a single message M is encrypted n times using $\mathtt{ID}_i$ for $i = 1, \ldots, n$ and the resulting ciphertext $(C_1, \ldots, C_n)$ is sent to the receivers. However, what one should not overlook here is that when the most widely used IBE scheme proposed by Boneh and Franklin [7] is employed to realize such scheme, we need at least n bilinear pairing computations, which is very expensive. (In fact, this was suggested in [18] and [11]).

Our Contributions. Following the above discussion, a natural question one can ask is how to design a multi-receiver identity-based encryption scheme that broadcasts a message with a high-level of computational efficiency while retaining security. In this paper, we answer this question affirmatively, providing an *efficient* multi-receiver IBE scheme that only requires "one" (or "none" if precomputed) pairing computation to encrypt a single message for multiple receivers. We provide formal security notions for multi-receiver IBE schemes based on the "selective identity attack" model in which an attacker outputs ahead of time the identities of multiple receivers that it wishes to challenge [8, 4]. We then prove that our schemes are secure against chosen plaintext attack (CPA) and adaptive ciphertext attack ("CCA2 [5]") in the random oracle model [6] assuming the standard assumptions related to the Bilinear Diffie-Hellman problems [7] are computationally hard. Finally, we show how our schemes lead to very efficient public key broadcast encryption schemes based on the "subset-cover" framework [18]. As an independent interest, we discuss in Section 5 how the selective identity attack model plays an important role in obtaining an efficient reduction in the security analysis of our efficient multi-receiver IBE schemes.

Related Work. The concept of multi-receiver public key encryption was independently formalized by Bellare, Boldyreva, and Micali [2], and Baudron, Pointcheval, and Stern [1]. Their main result is that the security of public key encryption in the single-receiver setting implies the security in the multi-receiver setting. Hence, for example, one can construct a semantically secure multi-receiver public key encryption scheme by simply encrypting a message under n different public keys of a semantically secure single-receiver public key encryption scheme. Later, Kurosawa proposed a technique called "randomness re-use" to improve the computational efficiency and bandwidths of an ElGamal [13] version of multi-receiver public key encryption scheme. Kurosawa's work was refined in [3] in a sense that a general test to determine whether a given public key encryption scheme permits the randomness re-use to build up an efficient multi-receiver encryption scheme.

To our knowledge, identity-based encryption in the multi-receiver setting has not been much treated in the literature. Chen, Harrison, Soldera, and Smart [9], and Smart [21] considered the problem of "conjunction" and "disjunction" of private keys associated with multiple identities in Boneh and Franklin's IBE scheme. In terms of conjunction, a user who has all the private keys associated with the identities that were used to encrypt a message can decrypt the ciphertext. Regarding disjunction, a user who possesses one of the private keys associated with identities that were used to create the ciphertext can decrypt. The authors of [9] and [21] showed how Boneh and Franklin's IBE scheme can be modified to solve the conjunction and disjunction problems efficiently. Especially, Smart presented a scheme that realizes the general logic formula called "conjunctive-disjunctive normal form (CDNF)" and showed how it can be used in access control to broadcast encrypted data. However, one criticism about the work of [9] and [21] is that their schemes are not supported by appropriate formal security model and proofs. Although our motivation is somewhat similar to those of [9] and [21] in terms of realizing "disjunction" in identity-based encryption, our constructions are different from theirs and importantly, we provide formal model and security proofs for our schemes.

Our work is also related to broadcast encryption [14] based on the "subset-cover" framework proposed by Naor, Naor, and Lotspiech [18]. In Section 6, we discuss it in detail.

2 Definitions for Multi-receiver Identity-Based Encryption

Model. We present a generic model for multi-receiver IBE schemes. Note that in the multi-receiver IBE setting, either a single message or multiple messages can be encrypted. However, *throughout the rest of the paper including the following definition, we assume that a single message is broadcast to the multiple receivers*, which leads to interesting schemes and applications.

Definition 1 (Multi-receiver IBE). A generic multi-receiver IBE scheme for broadcasting a single message, denoted by Π, consists of the following algorithms.

- PKG's key generation algorithm KeyGen: The PKG runs this algorithm to generate a PKG's master key and a common parameter, denoted by mk_{PKG} and cp_{PKG} respectively. Note that cp_{PKG} is given to all interested parties while mk_{PKG} is kept secret.
- PKG's private key extraction algorithm $\mathsf{Extract}$: Providing an identity $\mathtt{ID}$ received from a user and its master key mk_{PKG} as input, the PKG runs this algorithm to generate a private key associated with $\mathtt{ID}$, denoted by $sk_{\mathtt{ID}}$. We write $S_{\mathtt{ID}} = \mathsf{Extract}(mk_{\mathrm{PKG}}, \mathtt{ID})$.
- Encryption algorithm $\mathsf{Encrypt}$: Providing multiple identities $(\mathtt{ID}_1, \ldots, \mathtt{ID}_n)$ of the receivers, the PKG's common parameter cp_{PKG}, and a plaintext message M as input, the sender runs this algorithm to generates a ciphertext C which is an encryption of M under $(\mathtt{ID}_1, \ldots, \mathtt{ID}_n)$. We write $C = \mathsf{Encrypt}(cp_{\mathrm{PKG}}, (\mathtt{ID}_1, \ldots, \mathtt{ID}_n), M)$.

- Encryption algorithm $\mathsf{Encrypt}$: Providing multiple identities $(\mathtt{ID}_1, \ldots, \mathtt{ID}_n)$ of the receivers, the PKG's common parameter cp_{PKG}, and a plaintext message M as input, the sender runs this algorithm to generates a ciphertext C which is an encryption of M under $(\mathtt{ID}_1, \ldots, \mathtt{ID}_n)$. We write $C = \mathsf{Encrypt}(cp_{\mathrm{PKG}}, (\mathtt{ID}_1, \ldots, \mathtt{ID}_n), M)$.
- Decryption Algorithm $\mathsf{Decrypt}$: Providing its private key $sk_{\mathtt{ID}_i}$, the PKG's common parameter cp_{PKG}, and a ciphertext C as input, the receiver numbered i runs this algorithm to get a decryption D, which is either a certain plaintext message or a "*Reject*" message. We write $D = \mathsf{Decrypt}(cp_{\mathrm{PKG}}, sk_{\mathtt{ID}_i}, C)$

Security Notions. We present security notions for multi-receiver IBE schemes. In these notions, we consider the "selective identity attack" [8] in which an attacker commits ahead of time the identity that it intends to attack, which is slightly weaker than the model proposed in [7], where the attacker adaptively chooses the identity that will be challenged on rather than outputting it at the beginning.

We assume that this type of attacker outputs ahead of time a number of identities (of the receivers) that it wishes to attack, which we call a "*selective multi-ID attack* ". We then define "indistinguishability of encryptions under selective multi-ID, chosen plaintext attack", which we refer to as "IND-sMID-CPA" as follows.

Definition 2 (IND-sMID-CPA). Let A denote an attacker. Let Π be a generic multi-receiver IBE scheme. Consider the following game in which A interacts with the "Challenger":

Phase 1: A outputs target multiple identities, denoted by $(\mathtt{ID}_1^*, \ldots, \mathtt{ID}_n^*)$.
Phase 2: The Challenger runs the PKG's key generation algorithm $\mathsf{KeyGen}_{\mathrm{PKG}}(k)$ to generate a master key and a common parameter $(mk_{\mathrm{PKG}}, cp_{\mathrm{PKG}})$. The Challenger gives cp_{PKG} to A while keeps mk_{PKG} secret from A.
Phase 3: A issues a number of private key extraction queries, each of which is denoted by $\mathtt{ID}$. Upon receiving $\mathtt{ID}$, the Challenger runs the private key extraction algorithm to get $S_{\mathtt{ID}} = \mathsf{Extract}(mk_{\mathrm{PKG}}, \mathtt{ID})$. A restriction here is that $\mathtt{ID} \neq \mathtt{ID}_i^*$ for $i = 1, \ldots, n$.
Phase 4: A outputs a target plaintext pair (M_0, M_1). Upon receiving (M_0, M_1), the Challenger picks $\beta \in \{0, 1\}$ at random and creates a target ciphertext $C^* = \mathsf{Encrypt}(cp_{\mathrm{PKG}}, (\mathtt{ID}_1^*, \ldots, \mathtt{ID}_n^*), M_\beta)$. The Challenger returns C^* to A.
Phase 5: A issues a number of private key extraction queries as in Phase 3.
Phase 6: A outputs its guess $\beta' \in \{0, 1\}$.

We define A's guessing advantage $\mathbf{Adv}_{\Pi}^{\mathrm{IND-sMID-CPA}}(A) = |\Pr[\beta' = \beta] - \frac{1}{2}|$. A breaks IND-sMID-CPA of Π with (t, q_{ex}, ϵ) if and only if the guessing advantage of A that makes q_{ex} private key extraction queries is greater than ϵ within running time t. The scheme Π is said to be (t, q_{ex}, ϵ)-IND-sMID-CPA secure if there is no attacker A that breaks IND-sMID-CPA of Π with (t, q_{ex}, ϵ).

We now define "indistinguishability of encryptions under selective multi-ID, *adaptive* chosen ciphertext attack", which we refer to as "IND-sMID-CCA".

Definition 3 (IND-sMID-CCA). Let A denote an attacker. Let Π be a generic multi-receiver IBE scheme. Phases 1, 2, 4, and 6 of the attack game for IND-sMID-CCA are identical to those of IND-sMID-CPA. We only describe Phase 3 and 5 in the following:

Phase 3: A issues private key extraction queries as in Phase 3 of IND-sMID-CPA. Additionally, it issues decryption queries for target identities, each of which is denoted by $(C, \mathtt{ID}_i^*)$ for some $i \in [1, n]$. Upon receiving this, the Challenger generates a private key associated with $\mathtt{ID}_i^*$, which is denoted by $sk_{\mathtt{ID}_i^*}$, and returns $D = \mathsf{Decrypt}(cp_{\mathrm{PKG}}, S_{\mathtt{ID}_i}, C)$ to A.
Phase 5: As in Phase 3, A issues a number of private key extraction and decryption queries for target identities. However, this time, A is not allowed to issue a target ciphertext C^* as a decryption query.

We define A's guessing advantage $\mathbf{Adv}_{\Pi}^{\mathrm{IND-sMID-CCA}}(A) = |\Pr[\beta' = \beta] - \frac{1}{2}|$. A breaks IND-sMID-CCA of Π with $(t, q_{ex}, q_d, \epsilon)$ if and only if the guessing advantage of A that makes q_{ex} private key extraction queries and q_d decryption queries is greater than ϵ within running time t. The scheme Π is said to be $(t, q_{ex}, q_d, \epsilon)$-IND-sMID-CCA secure if there is no attacker A that breaks IND-sMID-CCA of Π with $(t, q_{ex}, q_d, \epsilon)$.

3 Bilinear Pairing and Related Computational Problems

As preliminaries, we review the bilinear pairing and related computational problems on which our efficient multi-receiver IBE schemes are based.

Definition 4 (Bilinear Pairing). An admissible bilinear pairing [7], which we denote by "$\hat{e}$", is defined over two groups of the same prime-order q denoted by $\mathcal{G}$ and $\mathcal{F}$ in which the Computational Diffie-Hellman problem is intractable. We will use an additive notation to describe the operation in $\mathcal{G}$ while we will use a multiplicative notation for the operation in $\mathcal{F}$. In practice, the group $\mathcal{G}$ is implemented using a group of points on certain elliptic curves, each of which has a small MOV exponent [17], and the group $\mathcal{F}$ will be implemented using a subgroup of the multiplicative group of a finite field. The admissible bilinear map has the following properties. 1) Bilinear: $\hat{e}(aP_1, bP_2) = \hat{e}(P_1, P_2)^{ab}$, where $P_1, P_2 \in \mathcal{G}$ and $a, b \in \mathbb{Z}_q^*$; 2) Non-degenerate: $\hat{e}$ does not send all pairs of points in $\mathcal{G} \times \mathcal{G}$ to the identity in $\mathcal{F}$. (Hence, if P is a generator of $\mathcal{G}$ then $\hat{e}(P, P)$ is a generator of $\mathcal{F}$).; 3)Computable: For all $P_1, P_2 \in \mathcal{G}$, the map $\hat{e}(P_1, P_2)$ is efficiently computable.

We now review the "Bilinear Decision Diffie-Hellman (BDDH)" problem, which is a "decisional" version of the Bilinear Diffie-Hellman problem on which Boneh and Franklin's IBE scheme [7] is based.

Definition 5 (BDDH). Let $\mathcal{G}$ and $\mathcal{F}$ be two groups of the same prime order q. Let P be a generator of $\mathcal{G}$. Suppose that there exists a bilinear map $\hat{e} : \mathcal{G} \times \mathcal{G} \rightarrow \mathcal{F}$. Let A be an attacker. A tries to solve the following problem: *Given* (P, aP, bP, cP, κ) *for uniformly chosen* $a, b, c \in \mathbb{Z}_q^*$ *and* $\kappa \in \mathcal{F}$*, decide whether* $\kappa = \hat{e}(P, P)^{abc}$.

We define A's guessing advantage $\mathbf{Adv}_{\mathcal{G}}^{\mathrm{BDDH}}(A)$ by

$$\Pr[A(P, aP, bP, cP, \hat{e}(P, P)^{abc}) = 1] - \Pr[A(P, aP, bP, cP, \gamma) = 1],$$

where $\gamma \in \mathcal{F}$ is chosen uniformly at random. A solves the BDDH problem with (t, ϵ) if and only if the guessing advantage of A is greater than ϵ within running time t. The BDDH problem is said to be (t, ϵ)-intractable if if there is no attacker A that solves the BDDH problem with (t, ϵ).

It is widely believed that the BDDH problem is computationally hard [4, 8]. Hence, we can define a Gap-Bilinear Diffie-Hellman (Gap-BDH) problem which belongs to the new class of computational problems, called "Gap Problems" proposed by Okamoto and Pointcheval [19]. Informally, the intractability of the Gap-BDH means that it is hard to compute a Bilinear Diffie-Hellman key $\hat{e}(P,P)^{abc}$ of (P, aP, bP, cP) although one has access to a BDDH oracle that, given a tuple (P, aP, bP, cP, κ), decides whether $\kappa = \hat{e}(P,P)^{abc}$. A formal definition follows.

Definition 6 (Gap-BDH). Let $\mathcal{G}$ and $\mathcal{F}$ be two groups of order the same prime order q. Let P be a generator of $\mathcal{G}$. Suppose that there exists a bilinear map $\hat{e} : \mathcal{G} \times \mathcal{G} \to \mathcal{F}$. Let A be an attacker. A tries to solve the following problem: *Given (P, aP, bP, cP), compute a Bilinear Diffie-Hellman key $\hat{e}(P,P)^{abc}$ with the help of the Bilinear Decisional Diffie-Hellman (BDDH) oracle, which, given (P, aP, bP, cP, κ), outputs 1 if $\kappa = \hat{e}(P,P)^{abc}$ and 0 otherwise.*

We define A's advantage $\mathbf{Adv}_{\mathcal{G}}^{\mathrm{Gap-BDH}}(A) = \Pr[A(P,aP,bP,cP) = \hat{e}(P,P)^{abc}]$. A solves the Gap-BDH problem with (t, q_o, ϵ) if and only if the guessing advantage of A that makes q_o BDDH-oracle queries is greater than ϵ within running time t. The Gap-BDH problem is said to be (t, q_o, ϵ)-intractable if if there is no attacker A that solves the Gap-BDH problem with (t, q_o, ϵ).

4 Proposed Schemes

CPA Secure Scheme. We present our efficient multi-receiver IBE scheme based on the bilinear pairing. Our scheme is motivated by the binary-tree scheme of Canetti, Halevi, and Katz [8], which bears some similarities to Gentry and Silverberg's [15], and Boneh and Boyen's [4] hierarchical IBE schemes. However, the purpose and structure of our scheme are different from those of all the previous ones.

- $\mathsf{KeyGen}_{\mathrm{PKG}}$: Choose two groups $\mathcal{G} = \langle P \rangle$ and $\mathcal{F}$ of the same prime order q. Construct a bilinear pairing $\hat{e} : \mathcal{G} \times \mathcal{G} \to \mathcal{F}$. Choose $Q \in \mathcal{G}^*$ uniformly at random. Choose $s \in \mathbb{Z}_q^*$ uniformly at random and compute $T = sP$. Also, select a hash function $H_1 : \{0,1\}^* \to \mathcal{G}^*$. Return $cp_{\mathrm{PKG}} = (q, \mathcal{G}, \mathcal{F}, \hat{e}, P, Q, T, H_1)$ and $mk_{\mathrm{PKG}} = (q, \mathcal{G}, \mathcal{F}, \hat{e}, P, s)$ as a PKG's common parameter and a master key respectively.
- $\mathsf{Extract}(mk_{\mathrm{PKG}}, \mathtt{ID})$: Compute $S_{\mathtt{ID}} = sH_1(\mathtt{ID})$. Return $S_{\mathtt{ID}}$ as a private key associated with identity $\mathtt{ID}$.
- $\mathsf{Encrypt}(cp_{\mathrm{PKG}}, (\mathtt{ID}_1, \ldots, \mathtt{ID}_n), M)$: Choose $r \in \mathbb{Z}_q^*$ uniformly at random and compute $C = (U, V_1, \ldots, V_n, W, \mathcal{L})$ such that

$$(U, V_1, \ldots, V_n, W, \mathcal{L}) = (rP, rH_1(\mathtt{ID}_1) + rQ \ldots, rH_1(\mathtt{ID}_n) + rQ, \hat{e}(Q,T)^r M, \mathcal{L}),$$

where $\mathcal{L}$ is a label that contains information about how "V_i" is associated with each receiver. Return C as a ciphertext. (Notice that $\hat{e}(Q,T)$ can be precomuted and provided as a PKG's common parameter. In this case, there is no need for the sender to perform a pairing computation).

- $\mathsf{Decrypt}(cp_{\mathrm{PKG}}, S_{\mathtt{ID}_i}, C)$ for some $i \in [1,n]$: Parse C as $(U, V_1, \ldots, V_n, W, \mathcal{L})$. Using $\mathcal{L}$, find appropriate V_i. Then, compute

$$M = \frac{\hat{e}(U, S_{\mathtt{ID}_i})}{\hat{e}(T, V_i)} W$$

and return M as a plaintext.

It is easy to see that the above decryption algorithm is consistent. Indeed, if C is a valid ciphertext,

$$\begin{aligned}\frac{\hat{e}(U, S_{\mathtt{ID}_i})}{\hat{e}(T, V_i)} W &= \frac{\hat{e}(rP, sH_1(\mathtt{ID}))}{\hat{e}(sP, rH_1(\mathtt{ID}_i) + rQ)} W = \frac{\hat{e}(rP, sH_1(\mathtt{ID}))}{\hat{e}(rP, sH_1(\mathtt{ID}_i) + sQ)} W \\ &= \frac{\hat{e}(rP, sH_1(\mathtt{ID}))}{\hat{e}(rP, sH_1(\mathtt{ID}_i))\hat{e}(rP, sQ)} \hat{e}(Q,T)^r M = M.\end{aligned}$$

Security Analysis. We now prove that the hardness of the BDDH problem (Definition 5) is sufficient for the above scheme to be IND-sMID-CPA secure in the random oracle model [6].

Theorem 1. *The above scheme is $(t, q_{H_1}, q_{ex}, \epsilon)$-IND-sMID-CPA secure in the random oracle model assuming that the BDDH problem is (t', ϵ')-intractable, where $t' > t + q_{H_1} O(\tau_1)$. ($\tau_1$ denotes the computing time for an exponentiation in $\mathcal{G}$).*

Proof. Assume that an attacker A breaks IND-sMID-CPA of the above scheme with probability greater than ϵ within time t making q_{ex} private key extraction queries. We show that using A, one can construct an attacker B for solving the BDDH problem (Definition 5).

Suppose that B is given (q, $\mathcal{G}$, $\mathcal{F}$, P, aP, bP, cP, κ) as an instance of the BDDH problem. By ϵ' and t', we denote B's winning probability and running time respectively. B can simulate the Challenger's execution of each phase of IND-sMID-CPA game for A as follows.

[Simulation of Phase 1] Suppose that A outputs target multiple identities ($\mathtt{ID}_1^*$, $\ldots$, $\mathtt{ID}_n^*$).

[Simulation of Phase 2] B sets $Q = bP$ and $T = cP$, and gives A (q, $\mathcal{G}$, $\mathcal{F}$, $\hat{e}$, P, T, Q, H_1) as the PKG's common parameter, where H_1 is a random oracle controlled by B as follows.

Upon receiving a random oracle query $\mathtt{ID}_j$ to H_1:

- If there exists $(\mathtt{ID}_j, l_j, L_j)$ in $\mathsf{H_1List}$, return L_j. Otherwise, do the following:
  - If $\mathtt{ID}_j = \mathtt{ID}_i^*$ for some $i \in [1,n]$, compute $L_j = l_j P - Q$.
  - Else choose $l_j \in \mathbb{Z}_q^*$ uniformly at random and compute $L_j = l_j P$.
  - Put $(\mathtt{ID}_j, l_j, L_j)$ in $\mathsf{H_1List}$ and return L_j as answer.

[Simulation of Phase 3] B answers A's private key extraction queries as follows.

Upon receiving a private key extraction query $\mathtt{ID}_j$ (Note that by the assumption of the IND-sMID-CPA game, $\mathtt{ID}_j \neq \mathtt{ID}_i^*$ for $i = 1, \ldots, n$).:

- If $(\mathtt{ID}_j, l_j, L_j)$ exists in $\mathsf{H_1List}$, compute $S_{\mathtt{ID}_j} = l_j T$. Otherwise do the following:
  - Choose $l_j \in \mathbb{Z}_q^*$ uniformly at random and compute $S_{\mathtt{ID}_j} = l_j T$.
  - Put $(\mathtt{ID}_j, l_j, L_j)$ in $\mathsf{H_1List}$ and return $S_{\mathtt{ID}_j}$ as answer. (Note that $S_{\mathtt{ID}_j} = l_j T = l_j cP = c l_j P = c H_1(\mathtt{ID}_j)$ for all $j \neq i$).

[Simulation of Phase 4] B creates a target ciphertext C^* as follows.

Upon receiving (M_0, M_1):

- Choose $\beta \in \{0, 1\}$ at random.
- Search $\mathsf{H_1List}$ to get l_i that corresponds to $\mathtt{ID}_i^*$ for $i = 1, \ldots, n$.
- Compute $l_i aP$ for $i = 1, \ldots, n$ and κM_β.
- Return $C^* = (aP, l_1 aP, \ldots, l_n aP, \kappa M_\beta)$ as a target ciphertext. Note here that.

[Simulation of Phase 5] B answers A's random oracle/private key extraction queries as in Phase 3.

[Simulation of Phase 6] A outputs its guess β'. If $\beta' = \beta$, B outputs 1. Otherwise, it outputs 0.

[Analysis] We note that if $\kappa = \hat{e}(P, P)^{abc}$, $\kappa M_\beta = \hat{e}(bP, cP)^a M_\beta = \hat{e}(Q, T)^a M_\beta$. Note also that $l_i aP = l_i aP - aQ + aQ = a(l_i P - Q) + aQ = aH_1(\mathtt{ID}_i^*) + aQ$ for $i = 1, \ldots, n$. Hence C^* is a valid ciphertext. On the other hand, if κ is uniform and independent in $\mathcal{F}$, so is κM_β. It is clear that from the construction above, B perfectly simulates the random oracle H_1 and the key private key extraction in Phase 3 and 5. Hence, we get $\Pr[B(P, aP, bP, cP, \hat{e}(P, P)^{abc}) = 1] = \Pr[\beta' = \beta]$, where $|\Pr[\beta' = \beta] - \frac{1}{2}| > \epsilon$, and $\Pr[B(P, aP, bP, cP, \gamma) = 1] = \Pr[\beta' = \beta] = \frac{1}{2}$, where γ is uniform in $\mathcal{F}$. Consequently, we get

$$|\Pr[B(P, aP, bP, cP, \hat{e}(P, P)^{abc}) = 1] - \Pr[B(P, aP, bP, cP, \gamma) = 1]| > \left|\left(\frac{1}{2} \pm \epsilon\right) - \frac{1}{2}\right| = \epsilon.$$

Note that $t' > t + q_{H_1} O(\tau_1)$, where τ_1 denotes the computing time for an exponentiation in $\mathcal{G}$. □

CCA Secure Scheme. In order to enhance security, we modify our scheme to provide (adaptive) chosen ciphertext security. Considering efficiency and simplicity, we employ the technique used in the REACT scheme proposed by Okamoto and Pointcheval' [20].

- $\mathsf{KeyGen}_{\mathrm{PKG}}$: Choose two groups $\mathcal{G} = \langle P \rangle$ and $\mathcal{F}$ of the same prime order q. Construct a bilinear pairing $\hat{e} : \mathcal{G} \times \mathcal{G} \to \mathcal{F}$. Choose $Q \in \mathcal{G}^*$ uniformly at random. Choose $s \in \mathbb{Z}_q^*$ uniformly at random and compute $T = sP$. Also, select hash functions $H_1 : \{0,1\}^* \to \mathcal{G}$, $H_2 : \mathcal{F} \to \{0,1\}^{k_1}$, and $H_3 : \mathcal{G} \times \cdots \times \mathcal{G} \times \mathcal{F} \times \{0,1\}^{k_1} \to \{0,1\}^{k_2}$. Return $cp_{\mathrm{PKG}} = (q, \mathcal{G}, \mathcal{F}, \hat{e}, q, P, Q, T, H_1, H_2, H_3)$ and mk_{PKG}=$(q, \mathcal{G}, \mathcal{F}, \hat{e}, P, s)$ as PKG's common parameter and master key respectively.

- $\mathsf{Extract}(mk_{\mathrm{PKG}}, \mathtt{ID})$: Compute $S_{\mathtt{ID}} = sH_1(\mathtt{ID})$. Return $S_{\mathtt{ID}}$ as a private key associated with identity $\mathtt{ID}$.
- $\mathsf{Encrypt}(cp_{\mathrm{PKG}}, (\mathtt{ID}_1, \ldots, \mathtt{ID}_n), M)$ where $M \in \{0,1\}^{k_1}$: Choose $R \in \mathcal{F}$ and $r \in \mathbb{Z}_q^*$ at random and compute compute $C = (U, V_1, \ldots, V_n, W_1, W_2, \mathcal{L}, \sigma)$ such that

$$\begin{aligned}(U, V_1, \ldots, V_n, W_1, W_2, \mathcal{L}, \sigma) \\ = (rP, rH_1(\mathtt{ID}_1) + rQ \ldots, rH_1(\mathtt{ID}_n) + rQ, \hat{e}(Q,T)^r R, M \oplus H_2(R), \\ H_3(R, M, U, V_1, \ldots, V_n, W_1, W_2, \mathcal{L}))\end{aligned}$$

  Return C as a ciphertext. (Notice that the "tag" σ guarantees the integrity of entire sequence of a ciphertext.)
- $\mathsf{Decrypt}(cp_{\mathrm{PKG}}, S_{\mathtt{ID}_i}, C, \mathtt{ID}_i)$ for some $i \in [1, n]$: Parse C as $(U, V_1, \ldots, V_n, W_1, W_2, \mathcal{L}, \sigma)$. Using $\mathcal{L}$, find appropriate V_i. Then, subsequently compute $R = \frac{\hat{e}(U, S_{\mathtt{ID}_i})}{\hat{e}(T, V_i)} W_1$, $M = W_2 \oplus H_2(R)$, and $\sigma' = H_3(R, M, V_1, \ldots, V_n, W_1, W_2, \mathcal{L})$. If $\sigma' = \sigma$, return M as a plaintext and "*Reject*" otherwise.

Security Analysis. We prove that the hardness of the Gap-BDH problem (Definition 6) is sufficient for the above scheme to be IND-sMID-CCA secure in the random oracle model. (The proof is given in Appendix A).

Theorem 2. *The above scheme is* $(t, q_{H_1}, q_{H_2}, q_{H_3}, q_{ex}, q_d, \epsilon)$*-IND-sMID-CCA secure in the random oracle model assuming that the Gap-BDH problem is* (t', q_o, ϵ')*-intractable, where* $\epsilon' > \epsilon - \frac{q_d}{2^{k_2}}$ *and* $t' > t + (q_{H_1} + q_{ex})O(\tau_1) + q_d O(\tau_2) + (q_{H_2} + q_{H_3})O(1)$*,* $q_o = q_{H_2} + q_{H_3}$ *(*τ_1 *and* τ_2 *respectively denote the computing time for an exponentiation in* $\mathcal{G}$ *and a pairing* $\hat{e}$*).*

5 Discussions on Efficiency and Security of Our Scheme

Efficiency Gains. We compare the major computational overhead and transmission rate (the length of the ciphertext) of our scheme with those of the obvious construction of multi-receiver IBE that simply re-encrypt a message M n times using Boneh and Franklin's IBE scheme, which we call "n-sequential composition of BF-IBE". In this scheme, M is encrypted to $(r_1 P, M \oplus H_2(\hat{e}(H_1(\mathtt{ID}_1), T)^{r_1}))$, ..., $(r_n P, M \oplus H_2(\hat{e}(H_1(\mathtt{ID}_n), T)^{r_n}))$, where $r_1, \ldots, r_n \in \mathbb{Z}_q^*$ are uniformly chosen at random and $(s, T = (sP))$ is the PKG's master key and common parameter respectively. As one can see, it is clear that our scheme provides much better performance: To encrypt a message M, our scheme only needs one pairing computation (none if $\hat{e}(Q,T)$ is precomputed), n additions in group $\mathcal{G}$ (to compute $H_1(\mathtt{ID}_i) + Q$), $n+1$ scalar multiplications with elements from $\mathcal{G}$ (to compute rP and $r(H_1(\mathtt{ID}_i) + Q) = rH_1(\mathtt{ID}_i) + rQ$), and 1 exponentiation in group $\mathcal{F}$ (to compute $\hat{e}(Q,T)^r$). The transmission rate is $(n+1)l_1 + l_2$ where l_1 and l_2 denote the bit-length of the element in $\mathcal{G}$ and $\mathcal{F}$ respectively. On the other hand, the n-sequential composition of BF-IBE needs n pairing computations (to compute $\hat{e}(H_1(\mathtt{ID}_i), T)$, n scalar multiplications with elements in $\mathcal{G}$ (to compute $r_i P$), n exponentiations in group $\mathcal{F}$ (to compute $\hat{e}(H_1(\mathtt{ID}_i), T)^{r_i}$). The transmission rate of this scheme is $nl_1 + nl_3$ where l_3 denotes the bit-length of the message.

In the following table, we summarize the above comparison.

	Pairings	Add. in $\mathcal{G}$	Mult. in $\mathcal{G}$	Exp. in $\mathcal{F}$	Trans. Rate
Our scheme	1 (or 0)	n	$n+2$	1	$(n+1)l_1 + l_2$
n-seq. comp. of BF-IBE	n	0	n	n	$nl_1 + nl_3$

One might argue, however, that the randomness re-use technique [16, 3] can be employed to reduce the number of multiplications in group $\mathcal{G}$. This indeed helps, but the n pairings and n exponentiations in group $\mathcal{F}$ still cannot be removed.

Fully Adaptive Multi-ID Attack. We notice that our scheme can also be proven secure in the "fully adaptive multi-ID attack" model where the attacker adaptively chooses which identity to attack and outputs target multiple identities in the challenger phase after it sees public parameters (rather than ahead of time). Unfortunately, the reduction is not very tight in that it introduces q_{ex}^n factor, where n denotes the number of receivers. The difficulty in getting an efficient reduction for our scheme stems from the difficulty in simulating a target ciphertext while handling the random oracle and key extraction queries.

To get a feeling for this, we sketch a security proof for our scheme in the fully adaptive multi-ID attack model. Let B be a BDDH attacker which is given (P, aP, bP, cP, κ), where κ is either $\hat{e}(P,P)^{abc}$ or a random element in $\mathcal{F}$, as an instance. Let A be a CPA attacker for our scheme in the fully adaptive multi-ID attack model. B first sets $Q = bP$ and $T = cP$, which will serve as the PKG's public key. Upon receiving a query $\mathtt{ID}$ to the random oracle H_1, B generates a random coin δ such that $\Pr[\delta = 0] = \rho$ and responds to the query with lP, where $l \in \mathbb{Z}_q^*$ is chosen at random, if $\delta = 0$, and $lP - Q$ otherwise. B puts $(\mathtt{ID}, l, \delta)$ in $\mathsf{H_1List}$, and if the same query is asked, B searches this list to respond to it. Upon receiving a private key extraction query $\mathtt{ID}$, B runs the above algorithm for simulating H_1 to get $(\mathtt{ID}, l, \delta)$ and answers with lT if $\delta = 0$, and aborts the simulation otherwise. Upon receiving target multiple identities $(\mathtt{ID}_1^*, \ldots, \mathtt{ID}_n^*)$ and target plaintexts (M_0, M_1), B runs the above algorithm for simulating H_1 to get $(\mathtt{ID}_1^*, l_1, \delta_1), \ldots, (\mathtt{ID}_n^*, l_n, \delta_n)$. Unless $\delta_1 = \cdots = \delta_n = 1$, B aborts the simulation, otherwise, creates a target ciphertext as follows: $C^* = (aP, l_1 aP, \ldots, l_n aP, \kappa M_\beta)$ for a random $\beta \in \{0,1\}$. The rest of the simulation are the same as the proof of Theorem 1.

As long as B does not abort the game, A's view in the simulation is identical to the view in the real attack from the same argument given in the proof of Theorem 1. The probability that B does not abort the simulation is $\rho^{q_{ex}}(1-\rho)^n$, which is maximized at $1 - \frac{n}{q_{ex}+n}$. Consequently, this introduces q_{ex}^n factor in the reduction cost. In the selective multi-ID attack model, we do not have this problem as $H_1(\mathtt{ID}_i^*)$ values can be "programmed" *at the beginning.*

On the other hand, we notice that one can get an efficient reduction for the security of the n-sequential composition of BF-IBE in the fully adaptive multi-ID attack model, due to its structural property which results in more pairing computations.

Trading off between security and efficiency is subjective. However, as seen from the beginning of this section, the efficiency gain in our scheme is huge, especially when there are a large number of receivers. In the following section, we show this is indeed a merit when our scheme is applied to broadcast encryption.

6 Application to Public Key Broadcast Encryption Based on Subset-Cover Framework

Broadcast Encryption Based on the Subset-Cover Framework. "Broadcast encryption" [14] deals with the problem of one party transmitting data to a large group of receivers so that only qualified subsets can decrypt the data. There are a number of applications of such scheme, e.g. pay-TV applications, distribution of copyright material, streaming audio/video, and etc. Since its introduction [14], broadcast encryption has been extensively studied in the literature. However, in this paper, we only focus on the "stateless receiver" case for broadcast encryption in the *public key* setting [18]. (Note that "stateless receiver" means that each user is given a fixed set of keys that cannot be changed through the lifetime of the system).

In the symmetric setting of broadcast encryption, only the trusted designer of the system, which we refer to as "Center", can broadcast a message. On the other hand, in the public key setting, the Center publishes a short public key which enables any party to broadcast data. Formally, a generic broadcast encryption scheme in the public key setting can be defined as follows [11].

Definition 7 (Public Key Broadcast Encryption). A public key broadcast encryption scheme consists of the following algorithms.

- Center's key generation algorithm $\mathsf{KeyGen}_{\mathrm{CTR}}$: Providing possibly a revocation threshold z (the maximum number of users that can be revoked) as input, the Center runs this algorithm to generate the Center's private key and public key, denoted by sk_{CTR} and pk_{CTR} respectively.
- Registration algorithm Reg: Providing the Center's private key and an index i associated with a user as input, the Center runs this algorithm to generate the secret initialization data, denoted by sk_i, to be delivered to a new user when he subscribes to the system. We write $sk_i = \mathsf{Reg}(sk_{\mathrm{CTR}}, i)$.
- Encryption algorithm $\mathsf{Encrypt}$: Providing the Center's public key, a session key K, and a set $\mathcal{R}$ of revoked users (with $|\mathcal{R}| \leq z$ if a threshold was specified to the Center's key generation algorithm) as input, the sender runs this algorithm to generate a ciphertext C to be broadcast. We write $C = \mathsf{Encrypt}(pk_{\mathrm{CTR}}, K, \mathcal{R})$.
- Decryption algorithm $\mathsf{Decrypt}$: Providing the secret data sk_i of a user and c ciphertext C, the user runs this algorithm to generate a decryption D, which is either a certain plaintext or a "*Reject*" message. We write $D = \mathsf{Decrypt}(sk_i, C)$.

Subset-Cover Framework. In brief, the basic idea behind the "subset-cover" framework for broadcast encryption (in the symmetric setting) proposed by Naor, Naor, and Lotspiech [18] is to define a family $\mathcal{S}$ of subsets of the set $\mathcal{N}$ of users, and to assign a key to each subset. Note that all the users in the

subset have access to the assigned key. If the Center wants to broadcast a message to all the "non-revoked" users, it first determines a partition of $\mathcal{N}/\mathcal{R}$, where $\mathcal{R}$ denotes the set of "revoked" users, and then encrypts the session key used to masquerade the message with all the keys associated to the subsets in the partition, which are elements of $\mathcal{S}$.

In [18], two specific methods that realize the above subset-cover framework: The "Complete Subtree (CS)" method and "Subset Difference (SD)" method. Since our scheme is well applicable to the CS method, we review it in detail as follows. In the CS scheme, users are organized in a full binary tree, denoted by $\mathcal{T}$: For simplicity, assume that there are $N = 2^h$ users in the system. Then, associate each user to a leaf of the full binary tree $\mathcal{T}$ of height h. The Subset-Cover family $\mathcal{S}$ is now the collection of all the full *subtrees* of $\mathcal{T}$. That is, if v_i is a node in $\mathcal{T}$, $S_i \in \mathcal{S}$ is the set of all the leaves of the full subtree of $\mathcal{T}$ rooted at v_i. To associate a key with each element of $\mathcal{S}$, the Center simply assigns a random number k_i to each node v_i. k_i is then be used as encryption/decryption key for the subset S_i. Since each user needs to know the keys corresponding to all the subsets he/she belongs to, during the registration step, the Center gives the user all the keys k_i assigned to each node v_i in the path from the root to the leaf representing the user. Hence, each user is required to store $O(\log N)$ keys. Note that the Center needs to keep track of all these keys given to each user. However, it was suggested in [18] that the Center derive all the $2N-1$ keys from some short seed using a pseudo-random function.

A New Public Key Broadcast Encryption from Our Efficient Multi-receiver IBE Scheme. The CS method described above can also be realized in the public key setting as envisioned in [18]. Namely, one can assign a public key pk_i to each node v_i. However, as already mentioned in [18], this is very inefficient in that total $2N - 1$ public keys should explicitly be stored in some directory. To overcome this deficiency, the authors of [18] suggest that the IBE scheme be employed, which requires only $O(1)$ space. According to Dodis and Fazio [11], this can be explained as follows: First, assign an identifier $\mathtt{ID}(S_i)$ to each subset S_i of the family $\mathcal{S}$. As an example, assign each edge of the full binary tree $\mathcal{T}$ with 0 or 1 depending on whether the edge connects the node with its left or right child, and assign to the subset S_i rooted at v_i the bit-string obtained reading off all the labels in the path from the root down to v_i. Then, the Center runs the key generation algorithm of IBE scheme to generate public parameters and the description of the mapping used to assign an identifier to each subset. Namely, the Center plays the role of the PKG in the IBE scheme. For each subset $S_i \in \mathcal{S}$, the Center generates a private key associated with it by running the private key extraction algorithm of the IBE scheme with the identifier $\mathtt{ID}(S_i)$. The Center then distributes the private data needed to decrypt the broadcast ciphertext, as in the symmetric key setting. Now, when a party wants to broadcast a message, it encrypts the session key used to protect the message under the public keys $\mathtt{ID}(S_{j_i})$ relative to all the subsets that cover all the non-revoked users. Note that this party only needs to know the public key of the Center and the description of the mapping $\mathtt{ID}(\cdot)$.

As a concrete instantiation, Dodis and Fazio apply the simple sequential composition of Boneh and Franklin's [7] IBE scheme to realize the above. More precisely, one can encrypt a session key K as follows: $(r_1P, K \oplus H_2(\hat{e}(H(\mathtt{ID}(S_1)), T)^{r_1})), \ldots, (r_tP, K \oplus H_2(\hat{e}(H(\mathtt{ID}(S_t)), T)^{r_t}))$ where $S_1, \ldots, S_t$ denote the subsets that cover $\mathcal{N}/\mathcal{R}$ and $(s, T = (sP))$ is the Center's private and public key pair. Note that $t = \mu \log \frac{N}{\mu}$ where $\mu = |\mathcal{R}|$ and $N = |\mathcal{N}|$. Hence, at least t pairing computations are needed.

Our Proposal. In contrast, using our multi-receiver IBE scheme presented in Section 4, one can design a very efficient public key broadcast encryption scheme that realizes the CS mechanism. In this new scheme, a session key K is encrypted as follows:

$$(rP, rH_1(\mathtt{ID}(S_1)) + rQ \ldots, rH_1(\mathtt{ID}(S_t)) + rQ, \hat{e}(Q, T)^r K),$$

where $(P, Q, T(= sP))$ and s are the Center's public and private keys respectively, and $r \in \mathbb{Z}_q^*$ is uniformly chosen at random.

Note that in the above scheme, the length of the broadcast message remains the same as that of the original scheme of [18]. That is, $t = \mu \log \frac{N}{\mu}$. The main advantage of our scheme over those considered in [18, 11], however, is that it is computationally much more efficient as we just need to compute t additions in group $\mathcal{G}$ instead of t pairings. Note also that compared with the scheme based on the SD method, which is proposed in [11], our scheme turns out to be more efficient in that the hierarchical IBE scheme [15] adopted in [11] results in expansion of the length of the encryption proportional to the depth in the hierarchy and more pairing computations proportional to the number of subset covers.

The above scheme can also be extended to provide chosen ciphertext security using our CCA scheme proposed in Section 4. More precisely, the security of the this scheme is relative to the following notion, which is weaker than the (public key version of) security notion for broadcast encryption presented in [18] in a sense that the the attacker outputs a set of revoked user before it sees a public key *but stronger* in a sense that it provides adaptive chosen ciphertext security. Note that the security notion given in [18] only considers non-adaptive chosen ciphertext attack, sometimes referred to as "CCA1 [5]".

Definition 8 (IND-sREV-CCA). Let A denote an attacker. Consider the following game in which A interacts with the "Challenger":

Phase 1: A outputs a set of revoked users denoted by $\mathcal{R}$.
Phase 2: The Challenger runs the Center's key generation algorithm $\mathsf{KeyGen}_{\mathrm{CTR}}$ to generate a private and public key pair $(sk_{\mathrm{CTR}}, pk_{\mathrm{CTR}})$ of the Center. The Challenger gives cp_{CTR} to A while keeps sk_{CTR} secret from A.
Phase 3: A requests the private data relative to the revoked users. Upon receiving each request, the Challenger runs the registration algorithm $\mathsf{Reg}(sk_{\mathrm{CTR}}, i)$ to give A the private data relative to the revoked users. A also queries arbitrary ciphertexts to see any non-revoked users decrypt them. Upon receiving each decryption query, the Challenger runs $\mathsf{Decrypt}(sk_i, C)$ and give the resulting decryption to A.

Phase 4: A outputs a target session key pair (K_0, K_1). Upon receiving (K_0, K_1), the Challenger picks a coin $\beta \in \{0,1\}$ at random and returns a target ciphertext $C^* = \mathsf{Encrypt}(pk_{\mathrm{CTR}}, K_\beta, \mathcal{R})$.
Phase 5: A issues a number of decryption queries C as in Phase 3 with a restriction that $C \neq C^*$.
Phase 6: A outputs its guess $\beta' \in \{0,1\}$.

The reduction from IND-sMID-CCA (Definition 3) of our CCA-version of multi-receiver IBE scheme presented in Section 4 to IND-sREV-CCA of the public key broadcast scheme described above is almost obvious: When the attacker A for the above broadcast encryption scheme outputs the set $\mathcal{R}$ of revoked users, the attacker B for the multi-receiver IBE scheme computes subsets $S_1, \ldots, S_1$ that cover $\mathcal{N}/\mathcal{R}$ and then outputs $\mathtt{ID}_1(S_1), \ldots, \mathtt{ID}_t(S_1)$ as a target multiple identities. B then gives A the obtained PKG's common parameter as the Center's public key. B proceeds to answer A's queries in Phase 3 by querying its private key extraction and decryption oracles. When A outputs a target key pair (K_0, K_1) in Phase 4, B forwards it to its Challenger to get an encryption of K_0 or K_1 under the target multiple identities $\mathtt{ID}_1(S_1), \ldots, \mathtt{ID}_t(S_1)$. B gives this as a target ciphertext to A and proceeds to answer A's decryption queries, which are different from the target ciphertext. When A outputs $\beta' \in \{0,1\}$ in Phase 6, B returns it as its final guess.

7 Concluding Remarks

In this paper, we proposed provably secure multi-receiver IBE schemes that broadcast encrypted data with a high-level of efficiency. We also discussed how the proposed schemes can be used to enhance the efficiency of public key broadcast encryption schemes for stateless receivers, based on the subset-cover framework.

Acknowledgement

The authors are grateful to anonymous referees for their helpful comments.

References

1. O. Baudron, D. Pointcheval, and J. Stern, *Extended Notions of Security for Multicast Public Key Cryptosystems*, In ICALP 2000, LNCS 1853, pp. 499–511, Springer-Verlag, 2000.
2. M. Bellare, A. Boldyreva, and S. Micali, *Public-key Encryption in a Multi-User Setting: Security Proofs and Improvements*, In Eurocrypt 2000, LNCS 1807, pp. 259–274, Springer-Verlag, 2000.
3. M. Bellare, A. Boldyreva, and D. Pointcheval, *Multi-Recepient Encryption Schemes: Security Notions and Randomness Re-Use*, In PKC 2003, LNCS 2567, pp. 85–99, Springer-Verlag, 2003.

4. D. Boneh and X. Boyen, *Efficient Selective-ID Secure Identity Based Encryption Without Random Oracles*, In Eurocrypt 2004, LNCS 3027, pp. 223–238, Springer-Verlag, 2004.
5. M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, *Relations Among Notions of Security for Public-Key Encryption Schemes*, In Crypto '98, LNCS 1462, pp. 26–45, Springer-Verlag, 1998.
6. M. Bellare and P. Rogaway, *Random Oracles are Practical: A Paradigm for Designing Efficient Protocols*, In ACM CCCS '93, pp. 62–73, 1993.
7. D. Boneh and M. Franklin, *Identity-Based Encryption from the Weil Pairing*, Advances in Cryptology - In Crypto 2001, LNCS 2139, pp. 213–229, Springer-Verlag, 2001.
8. R. Canetti, S. Halevi, and J. Katz, *A Forward-Secure Public-Key Encryption Scheme*, Advances in Cryptology - In Eurocrypt 2003, LNCS 2656, pp. 255–271, Springer-Verlag, 2003.
9. L. Chen, K. Harrison, D. Soldera, and N. P. Smart: *Applications of Multiple Trust Authorities in Pairing Based Cryptosysems*, In InfraSec 2002, LNCS 2437, pp. 260–275, Springer-Verlag, 2002.
10. C. Cocks, *An Identity Based Encryption Scheme Based on Quadratic Residues*, In IMA 2001, LNCS 2260, pp. 360–363, Springer-Verlag, 2001.
11. Y. Dodis and N. Fazio, *Public Key Broadcast Encryption for Stateless Receivers*, In ACM-DRM, 2002.
12. Y. Dodis and N. Fazio, *Public Key Trace and Revoke Scheme Secure against Adaptive Chosen Ciphertext Attack*, In Public Key Cryptography 2003 (PKC 2003), LNCS 2567, pp. 100–115, Springer-Verlag 2002.
13. T. ElGamal: *A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms*, IEEE Transactions on Information Theory, Vol. 31, pp. 469–472, IEEE, 1985.
14. A. Fiat and M. Naor, *Broadcast Encryption*, In Crypto '94, LNCS 773, pp. 480–491, Springer-Verlag, 1994.
15. C. Gentry and A. Silverberg, *Hierarchical ID-Based Cryptography*, In Asiacrypt 2002, LNCS 2501, pp. 548–566, Springer-Verlag, 2002.
16. K. Kurosawa, *Multi-Recepient Public-Key Encryption with Shortened Ciphertext*, In PKC 2002, LNCS 2274, pp. 48–63, Springer-Verlag, 2002.
17. A. J. Menezes, T. Okamoto, and S. A. Vanstone: *Reducing Elliptic Curve Logarithms to a Finite Field*, IEEE Tran. on Info. Theory, Vol. 31, pp. 1639–1646, IEEE, 1993.
18. D. Naor, M. Naor, and J. Lotspiech, *Revocation and Tracing Schemes for Stateless Receivers*, In Crypto 2001, LNCS 2139, pp. 41-62, Springer-verlag, 2001.
19. T. Okamoto and D. Pointcheval, *The Gap-Problems: A New Class of Problems for the Security of Cryptographic Schemes*, In PKC 2001, LNCS 1992, pp. 104–118, Springer-Verlag, 2001.
20. T. Okamoto and D. Pointcheval, *REACT: Rapid Enhanced-security Asymmetric Cryptosystem Transform*, In CT-RSA 2001, LNCS 2020, pp. 159–174, Springer-Verlag, 2001.
21. N. P. Smart, *Access Control Using Pairing Based Cryptography*, In CT-RSA 2003, LNCS 2612, pp. 111–121, Springer-Verlag, 2003.

A Proof of Theorem 2

Proof. We first define a normal public key encryption (non-IBE) scheme, which we call "*Bilinear ElGamal*" as follows.

- KeyGen: Choose two groups $\mathcal{G} = \langle P \rangle$ and $\mathcal{F}$ of the same prime order q. Construct a bilinear pairing $\hat{e} : \mathcal{G} \times \mathcal{G} \to \mathcal{F}$. Choose $Q \in \mathcal{G}^*$ uniformly at random. Choose $s \in \mathbb{Z}_q^*$ uniformly at random and compute $T = sP$. Return $pk = (q, \mathcal{G}, \mathcal{F}, \hat{e}, P, Q, T)$ and $sk{=}(q, \mathcal{G}, \mathcal{F}, \hat{e}, P, T, s)$ as a public key and a private key key respectively.
- Encrypt(pk, M): Choose $r \in \mathbb{Z}_q^*$ at random and compute $C = (U, W)$ such that $(U, W) = (rP, \hat{e}(Q, T)^r M)$ for $M \in \mathcal{F}$. Return C as a ciphertext.
- Decrypt(sk, C): Parse C as (U, W), compute $M = W/\hat{e}(U, Q)^s$, and return M as a plaintext.

In [20], a security notion for public key encryption called "One-Way-ness under Plaintext Checking Attack (OW-PCA)" is defined. Informally, a public key encryption scheme is (t', q_o, ϵ')-OW-PCA secure if for any t'-time attacker B making q_o queries to the *Plaintext Checking (PC) oracle*, which, given a ciphertext-plaintext message pair (C, M), outputs 1 if C encrypts M and 0 otherwise, B's advantage that finds a pre-image of a given ciphertext is less than ϵ'.

It is easy to see that the above Bilinear ElGamal scheme is OW-PCA secure assuming that the Gap-BDH problem (Definition 6) is intractable: Taking a public key (P, Q, T), a ciphertext (U, W), and a certain plaintext M' as input, the PC oracle checks whether $(P, U, Q, T, W/M')$ is a Bilinear Diffie-Hellman tuple. Hence, the running time and advantage of the OW-PCA attacker is exactly the same as those of Gap-BDH attacker.

Now, assume that an attacker A breaks IND-sMID-CCA of the proposed scheme in Section 4 with probability greater than ϵ within time t making q_{H_1}, q_{H_2} and q_{H_3} random oracle queries and q_{ex} private key extraction queries and q_d decryption queries. We show that using this A, one can construct an OW-PCA attacker B for the Bilinear ElGamal Scheme.

Suppose that B is given $(q, \mathcal{G}, \mathcal{F}, \hat{e}, P, Q, T)$ as a public key, and $(U^*, W^*) = (r^*P, \hat{e}(Q, T)^{r^*} R^*)$ as a target ciphertext of the *Bilinear ElGamal* Scheme. Suppose also that B's makes q_o queries to the PCA oracle of the Bilinear ElGamal scheme within time t'. We denote B's winning probability by ϵ', which will be determined later. B can simulate the Challenger's execution of each phase of IND-sMID-CCA game for A as follows.

[Simulation of Phase 1] Suppose that A outputs target multiple identities $(\mathtt{ID}_1^*, \ldots, \mathtt{ID}_n^*)$.

[Simulation of Phase 2] B gives A $(q, \mathcal{G}, \mathcal{F}, \hat{e}, P, Q, T, H_1, H_2, H_3)$ as the PKG's common parameter, where H_1, H_2, and H_3 are random oracles controlled by B as follows.

Upon receiving a query $\mathtt{ID}_j$ to the random oracle H_1 for some $j \in [1, q_{H_1}]$:

- If $(\mathtt{ID}_j, l_j, L_j)$ exists in $\mathsf{H_1List}$, return L_j. Otherwise do the following:
  - If $\mathtt{ID}_j = \mathtt{ID}_i^*$ for some $i \in [1, n]$, compute $L_j = l_j P - Q$.
  - Else choose $l_j \in \mathbb{Z}_q^*$ uniformly at random and compute $L_j = l_j P$.
  - Put $(\mathtt{ID}_j, l_j, L_j)$ in $\mathsf{H_1List}$ and return L_j as answer.

Upon receiving a query R_j to the random oracle H_2 for some $j \in [1, q_{H_2}]$:

- If (R_j, K_j) exists in $\mathsf{H_2List}$, return L_j. Otherwise do the following:
  - Check whether (U^*, W^*) encrypts R_j using the PC oracle. If it is, return R_j and terminate the game. (In this case, B has achieved its goal as the pre-image of (U^*, W^*) has been found). Otherwise, do the following:
    - Pick $K_j \in \{0,1\}^{k_1}$ uniformly at random.
    - Put (R_j, K_j) in $\mathsf{H_2List}$ and return K_j as answer.

Upon receiving a query $(R_j, M_j, U_j, V_{j_1}, \ldots, V_{j_n}, W_{j_1}, W_{j_2}, \mathcal{L}_j)$ to the random oracle H_3 for some $j \in [1, q_{H_3}]$:

- If $((R_j, M_j, U_j, V_{j_1}, \ldots, V_{j_n}, W_{j_1}, W_{j_2}, \mathcal{L}_j), \sigma_j)$ exists in $\mathsf{H_3List}$, return σ_j. Otherwise do the following:
  - Check whether (U^*, W^*) encrypts R_j using the PC oracle. If it is, return R_j and terminate the game. (In this case, B has achieved its goal as the pre-image of (U^*, W^*) has been found). Otherwise, do the following:
    - Pick $\sigma_j \in \{0,1\}^{k_2}$ uniformly at random.
    - Put $((R_j, M_j, U_j, V_{j_1}, \ldots, V_{j_n}, W_{j_1}, W_{j_2}, \mathcal{L}_j), \sigma_j)$ in $\mathsf{H_3List}$ and return σ_j as answer.

[Simulation of Phase 3] B then answers A's queries in Phase 3 as follows.

Upon receiving a private key extraction query $\mathtt{ID}_j$ for some $j \in [1, q_{ex}]$ (By assumption, $\mathtt{ID}_j \neq \mathtt{ID}_i^*$ for $i = 1, \ldots, n$).:

- If $(\mathtt{ID}_j, l_j, L_j)$ exists in $\mathsf{H_1List}$, compute $S_{\mathtt{ID}_j} = l_j T$. Otherwise do the following:
  - Choose $l_j \in \mathbb{Z}_q^*$ uniformly at random and compute $S_{\mathtt{ID}_j} = l_j T$.
  - Put $(\mathtt{ID}_j, l_j, L_j)$ in $\mathsf{H_1List}$ and return $S_{\mathtt{ID}_j}$ as answer.

Upon receiving a decryption query $(C_j, \mathtt{ID}_i^*)$ for some $i \in [1, n]$ and $j \in [1, q_d]$, where $C_j = (U_j, V_{j_1}, \ldots, V_{j_n}, W_{j_1}, W_{j_2}, \mathcal{L}_j, \sigma_j)$:

- If $((R_j, M_j, U_j, V_{j_1}, \ldots, V_{j_n}, W_{j_1}, W_{j_2}, \mathcal{L}_j), \sigma_j)$ exists in $\mathsf{H_3List}$ do the following:
  - Compute $H_2(R_j)$ using the simulation of H_2 above and check whether $H_2(R_j) \oplus M_j = W_{j_2}$. If not, return "*Reject*", otherwise do the following:
    - Check whether (U_j, W_{j_1}) encrypts R_j using the PC oracle,
    - Check $\hat{e}(U_j, H_1(\mathtt{ID}_i^*) + Q) = \hat{e}(P, V_{j_i})$.
    - If *both* of the above equations hold, return M_j and "*Reject*" otherwise.
- Else return "*Reject*".

[Simulation of Phase 4] Using the target ciphertext $(U^*, W^*) = (r^*P, \hat{e}(Q,T)^{r^*} R^*)$ of the Bilinear ElGamal scheme, B creates a target ciphertext C^* as follows.

Upon receiving (M_0, M_1):

- Choose $\beta \in \{0,1\}$ at random and search $\mathsf{H_1List}$ to get l_i that corresponds to $\mathtt{ID}_i^*$ for $i = 1, \ldots, n$. Then, compute $l_i U^*$ for $i = 1, \ldots, n$.
- Choose $K^* \in \{0,1\}^{k_1}$ uniformly at random and set $K^* = H_2(R^*)$. Also, create a label $\mathcal{L}^*$.
- Choose $\sigma^* \in \{0,1\}^{k_2}$ uniformly at random and set

$$\sigma^* = H_3(R^*, M_\beta, U^*, l_1 U^*, \ldots, l_n U^*, W^*, K^* \oplus M_\beta, \mathcal{L}^*).$$

- Return $C^* = (U^*, l_1 U^*, \ldots, l_n U^*, W^*, K^* \oplus M_\beta, \mathcal{L}^*, \sigma^*)$ as a target ciphertext.

[Simulation of Phase 5] B answers A's random oracle, decryption and private key extraction queries as before. Note that, this time, if $(R^*, M_\beta, U^*, l_1U^*, \ldots, l_nU^*, W^*, K^* \oplus M_\beta, \mathcal{L}^*)$ is asked to the the random oracle H_3, the value σ^* created in Simulation of Phase 4 is returned. (The value R^* can be detected with the help of the PC oracle).

[Simulation of Phase 6] A outputs its guess β'. If $\beta' = \beta$, B outputs 1. Otherwise, it outputs 0.

[Analysis] Note first that the private keys associated with each $\mathtt{ID}_j (\neq \mathtt{ID}_i^*)$ created in Simulation of Phase 3 are identically distributed as those in the real attack since $S_{\mathtt{ID}_j} = l_jT = l_jsP = sl_jP = sH_1(\mathtt{ID}_j)$. The simulations of the random oracle H_2 and H_3 are also perfect *unless* R^* has been asked to one of the random oracles H_2 and H_3. However, if these event happen, B breaks the OW-PCA of the Bilinear ElGamal scheme.

Note also that the distribution of the simulated target ciphertext is identical to that of the target ciphertext in the real attack since $l_iU^* = l_ir^*P = l_ir^*P - r^*Q + r^*Q = r^*(l_iP - Q) + r^*Q = r^*H_1(\mathtt{ID}_i^*) + r^*Q$ for all $i = 1, \ldots, n$.

The simulation of the decryption oracle is nearly perfect but there are cases when a valid ciphertext is rejected since, in the simulation of decryption oracle, if $(R, M, U, V_1, \ldots, V_n, W_1, W_2, \mathcal{L})$ has not been queried to H_3, the ciphertext is rejected straight way. Note that this leads to two cases: 1) A uses the value σ^* which is a part of a target ciphertext as a part its decryption query; 2) A has guessed a right value for the output of H_3 without querying it. However, in the first case, since $(U^*, l_1U^*, \ldots, l_nU^*, W^*, K^* \oplus M_\beta, \mathcal{L}^*)$ as well as (R^*, M_β) is provided as input to H_3, the decryption query A would ask is the same as the target ciphertext which is not allowed to query. The second case may happen but with a negligible probability $1/2^{k_2}$.

Following the above discussion, if B does not correctly guess the output of H_3, the view of A in the simulation is identical to the view in the real attack. Hence, we have $\Pr[B(P, aP, bP, cP) = \hat{e}(P, P)^{abc}] = \Pr[\beta' = \beta | \neg \mathsf{GuessH_3}] - \frac{1}{2}|$, where $\mathsf{GuessH_3}$ denotes an event that A correctly guesses the output of H_3. In the mean time, by definition of A, we have $|\Pr[\beta' = \beta] - \frac{1}{2}| > \epsilon$. Consequently, we have $\left|\Pr[\beta' = \beta | \neg \mathsf{GuessH_3}] - \frac{1}{2}\right| > \left|\Pr[\beta' = \beta] - \Pr[\mathsf{GuessH_3}] - \frac{1}{2}\right| > \epsilon - \Pr[\mathsf{GuessH_3}]$.

Since A makes total q_d decryption queries during the attack $\Pr[\mathsf{GuessH_3}] \leq q_d/2^{k_2}$. Thus, we have $\epsilon' > \epsilon - \frac{q_d}{2^{k_2}}$. The running time t' and the number q_o of PC oracle queries of B are readily checked. □