Efficient ID-Based Signcryption Schemes for Multiple Receivers*

S. Sharmila Deva Selvi[1], S. Sree Vivek**,[1], Rahul Srinivasan[2], and Pandu Rangan Chandrasekaran**,[1]

[1] {sharmila,svivek}@cse.iitm.ernet.in, prangan@iitm.ac.in
Indian Institute of Technology Madras
Theoretical Computer Science Laboratory
Department of Computer Science and Engineering
Chennai, India

[2] rahul.srinivasan@iitb.ac.in
Indian Institute of Technology Bombay
Department of Computer Science and Engineering
Chennai, India

Abstract. This paper puts forward new efficient constructions for Multi-Receiver Signcryption in the Identity-based setting. We consider a scenario where a user wants to securely send a message to a dynamically changing subset of the receivers in such a way that non-members of the of this subset cannot learn the message. The obvious solution is to transmit an individually signcrypted message to every member of the subset. This requires a very long transmission (the number of receivers times the length of the message) and high computation cost. Another simple solution is to provide every possible subset of receivers with a key. This requires every user to store a huge number of keys. In this case, the storage efficiency is compromised. The goal of this paper is to provide solutions which are efficient in all three measures i.e. transmission length, storage of keys and computation at both ends. We propose three new schemes that achieve both confidentiality and authenticity simultaneously in this setting and are the most efficient schemes to date, in the parameters described above. The first construction achieves optimal computational and storage cost. The second construction achieves much lesser transmission length than the previous scheme (down to a ratio of one-third), while still maintaining optimal storage cost. The third scheme breaks the barrier of ciphertext length of linear order in the number of receivers, and achieves constant sized ciphertext, independent of the size of the receiver set. This is the first Multi-receiver Signcryption scheme to do so. We support all three schemes with security proofs under a precisely defined formal security model.

Keywords: Multiple Receivers, Signcryption, Identity-Based Cryptography, Provable Security.

1 Introduction

Two fundamental tools of Public Key Cryptography are privacy and authenticity, achieved through encryption and signatures respectively. Both primitives are used in a number of high level protocols. Signcryption, introduced by Zheng in 1997 [30], is a cryptographic primitive that offers confidentiality and unforgeability simultaneously similar to the sign-then-encrypt technique, but with lesser computational complexity and lower communication cost. The security notion for signcryption was first formally defined in 2002 by Baek et al. in [3].

The concept of an Identity based (ID-based) cryptosystem was introduced by Shamir [26] in 1984. The idea is that users within a system could use their online identifiers (combined with certain system-wide information) as their public keys. This greatly reduces the problems with

[*] An abridged version of this report has been submitted to INSCRYPT '08.

[**] Work supported by Project No. CSE/05-06/076/DITX/CPAN on Protocols for Secure Communication and Computation sponsored by Department of Information Technology, Government of India.

key management and provides a more convenient alternative to conventional public key infrastructure. Only in 2001 did first fully practical identity-based encryption (IBE) solution arise, using bilinear mappings over elliptic curves [9].

ID-based signcryption schemes achieve the functionality of signcryption with the added advantage that ID-based cryptography provides. In [21], Malone-Lee gave the first ID-based signcryption scheme. Later it was found that Malone-Lee's scheme was not semantically secure. Since then, quite a few ID-based signcryption schemes have been proposed ([20], [5], [13]). To date, some of the most efficient ID-based signcryption schemes are that of Chen et al. [13], and Barreto et al. [5]

1.1 Motivation

Assume that there are n receivers, numbered 1 to 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 directed to receiver i using pk_i for $i = 1$ to 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 Encryption* in literature. As one can easily see, any multi-receiver public key encryption scheme can be transformed into a natural broadcast encryption scheme

The objective of a multi-receiver ID-based signcryption scheme is to efficiently broadcast a single ciphertext to different receivers while achieving the security properties of authenticity and unforgeability. In practice, broadcasting a message to multiple users in a secure and authenticated manner is an important facility for a group of people who are jointly working on the same project to communicate with one another.

1.2 Related Work

Multi-receiver Encryption. The concept of multi-receiver public key encryption was independently formalized by Bellare, Boldyreva, and Micali [7], and Baudron, Pointcheval, and Stern [6]. 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. But this is inefficient in the sense that the process of encryption is performed n times. Later, Kurosawa proposed a technique called randomness re-use to improve the computational efficiency in multi-receiver public key encryption schemes.

Multi-receiver Identity-Based Encryption. Chen, Harrison, Soldera, and Smart [12] considered conjunction and disjunction of private keys associated with multiple identities in Boneh and Franklin's IBE scheme. Regarding conjunction, users possesing all the private keys associated with the identities that were used to encrypt a message can decrypt the ciphertext. Considering disjunction, a user who possesses one of the private keys associated with identities that were used to encrypt the message can decrypt the ciphertext. [12] and [27] show how Boneh and Franklin's IBE scheme can be modified to solve the conjunction and disjunction problems efficiently. However, these schemes are not supported by a formal security model and appropriate proofs. Later Baek et. al [2] came up with an efficient scheme and formally proved it secure in the random oracle model.

Multi-receiver ID-based Key Encapsulation. The notion of mKEM was introduced in [28]. Later, in [4], the notion of mKEM was extended to multi-receiver identity based key encapsulation (mID-KEM), i.e. mKEM in the identity-based setting. In [2] and [4], the ciphertext size grows with the number of receivers. In [11], Chatterjee and Sarkar achieved a controllable trade-off between the ciphertext size and the private key size: ciphertexts are of size $|S|/N$, and private keys are of size N where S is the set of receivers and N a parameter of the protocol (which also represents, in the security reduction, the maximum number of identities that the adversary is allowed to target). Thus they introduced the first mIDKEM protocols to achieve sub-linear ciphertext sizes. Very recently, Abdalla et al. proposed in [1] a generic construction that achieves ciphertexts of constant size, but private keys of size $O(n_{max}^2)$. Furukawa [24] and Delerablée [14] independently proposed an mID-KEM scheme which achieves constant size ciphertext at the cost of the public key size growing linearly in the number of receivers.

Multi-receiver ID-based Signcryption. In the multi-receiver identity-based setting, we are interested in the situation where there is not only a single sender to multiple receivers, but also multiple senders. In such cases, it is desirable to achieve confidentiality and authenticity simultaneously. To our knowledge, identity-based signcryption in the multi-receiver setting has not been much treated in the literature. Duan et al. [16] proposed the first mIBSC scheme. Yu et al.[29] also proposed a mIBSC scheme recently, but is not secure in the sense of unforgeablity [25].

1.3 Our Contribution

Following the above discussion, a natural question one can ask is how to design a multi-receiver identity-based signcryption scheme that achieves both confidentiality and authenticity, and broadcasts a message with a high-level of computational and storage efficiency and optimal transmission length while retaining security. In this paper, we introduce three efficient schemes to answer this question. The first construction, is an extension of the signcryption scheme proposed by Barreto et. al. [5]. The original scheme, when directly extended to multiple receivers with randomness re-use, does not provide confidentiality of messages, in the sense that an adversary can learn the contents of the encrypted ciphertext without possessing the secret keys of any of the receivers. This extended construction, presented in this paper, is computationally the most efficient scheme to date - it requires no pairing computations during signcryption, while the reverse process requires just two. No existing public-key encryption scheme for multiple parties achieves such a high level of computational and storage efficiency. But the only drawback of this scheme is that the ciphertext size is of the order of the number of receivers. The second construction improves on the previous one, by reducing the ciphertext size to the order of a third of the number of receivers, by using the clever technique of dividing the users into groups of three. This construction retains the storage efficiency of the previous scheme and still achieves a level of efficiency on the computation side, that has not been reached to date. It requires no pairing computations to signcrypt a message and just three to retrieve the message from the ciphertext, while ensuring both confidentiality and authenticity. It achieves the optimal trade off between ciphertext size and storage cost. Moreover, this scheme does not pose a restriction on the size of the set of receivers, that certain schemes do. Scaling up this grouping ultimately results in the third construction which achieves constant size ciphertext. This is the first such scheme to do so. But this is achieved at the cost of storage efficiency. The size of the public key grows as the maximal size of the subset of receivers in the group (which can be significantly less than the total number of people in the group). This construction, when converted to a Broadcast Encryption scheme [17], is comparable to the Identity-Based Broadcast Encryption (IBBE) schemes proposed by Furukawa [24] and Delerablée [14]. We also provide formal security notions for Multi-receiver Identity-Based Signcryption (mIBSC) schemes and formally prove all three constructions secure in the random oracle model by reducing their security to standard assumptions related to the Biliniear Diffie Hellman Problems.

Remark It is a common practice in group oriented protocols to ignore the part of the broadcast ciphertext that identifies the target subset of users. We distinguish between the set identification transmission and the message signcryption transmission. Our goal is the study of latter and their requirements. What is called ciphertext size usually refers to the size of the header that corresponds to the message signcryption alone.

Scheme	Storage Cost		Computational Cost - No. of pairings for (Signcryption, Designcryption)	Header Size[4]
	Public Key Size[3]	Private Key Size		
Duan and Cao [16]	$O(1)$	$O(1)$	(1,4)	$O(t)$
Yu et al.[29]	$O(1)$	$O(1)$	(1,3)	$O(t)$
Construction 1	$O(1)$	$O(1)$	(0,2)	$O(t)$
Construction 2	$O(1)$	$O(1)$	(0,3)	$O(t/3)$
Construction 3	$O(N)$	$O(1)$	(0,3)	$O(1)$

Table 1. Performance Comparison of existing mIBSC Schemes

2 Preliminaries

Let $\mathbb{G}_1$ be an additive cyclic group of prime order p, with generators P and Q, and $\mathbb{G}_2$ be a multiplicative cyclic group of the same order p.

2.1 Bilinear Pairing

A bilinear pairing is a map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ with the following properties.

- ***Bilinearity.*** For all $P, Q, R \in \mathbb{G}_1$,
  - $e(P + Q, R) = e(P, R)e(Q, R)$
  - $e(P, Q + R) = e(P, Q)e(P, R)$
  - $e(aP, bQ) = e(P, Q)^{ab}$
- ***Non-Degeneracy.*** There exist $P, Q \in \mathbb{G}_1$ such that $e(P, Q) \neq I_{\mathbb{G}_2}$, where $I_{\mathbb{G}_2}$ is the identity element of $\mathbb{G}_2$.
- ***Computability.*** There exists an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in \mathbb{G}_1$.

2.2 Computational Assumptions

In this section, we review the computational assumptions related to bilinear maps that are relevant to the protocol we discuss.

Let $\mathcal{B} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e(\cdot, \cdot))$ be a bilinear map group system such that $\mathbb{G}_1 = \mathbb{G}_2 = \mathbb{G}$. Let $G_0 \in \mathbb{G}$ be a generator of $\mathbb{G}$, and set $g = e(G_0, G_0) \in \mathbb{G}_T$.

l-Strong Diffie Hellman Problem ($l - SDHP$) The l-Strong Diffie-Hellman problem ($l - SDHP$) in the groups $\mathbb{G}$ consists of, given $G_0, sG_0, s^2G_0, \ldots, s^lG_0$, finding a pair $(c, \frac{1}{c+s}G_0)$ with $c \in \mathbb{Z}_p^*$.

Definition 1 *The advantage of any probabilistic polynomial time algorithm $\mathcal{A}$ in solving the $l-SDHP$ in $\mathbb{G}$ is defined as $Adv_{\mathcal{A}}^{l-SDHP} = Pr\left[\mathcal{A}(G_0, sG_0, s^2G_0, \ldots, s^lG_0) = (c, \frac{1}{c+s}G_0) \mid c \in \mathbb{Z}_p^*\right]$ The l-SDHP Assumption is that, for any probabilistic polynomial time algorithm $\mathcal{A}$, the advantage $Adv_{\mathcal{A}}^{l-SDHP}$ is negligibly small.*

The General Diffie-Hellman Exponent Assumption We make use of the generalization of the Diffie-Hellman exponent assumption due to Boneh, Boyen and Goh [8]. Let m, n be positive integers and $U, V \in \mathbb{F}_p[X_1, ..., X_n]^m$ be two m-tuples of n-variate polynomials over $\mathbb{F}_p$. Thus, U and V are just two sets containing m multivariate polynomials each. We write $U = (u_1, u_2, ..., u_m)$ and $V = (v_1, v_2, ..., v_m)$ as tuples of polynomials and impose that $u_1 =$

[1] N is the maximal size of the receiver set.

[2] t is the size of the receiver set.

$v_1 = 1$; that is, the constant polynomials 1. For a set Ω, a function $h : \mathbb{F}_p \rightarrow \Omega$ and vector $(x_1, ..., x_n) \in \mathbb{F}_p^n$, we write

$$h(U(x_1, ..., x_n)) = (h(u_1(x_1, ..., x_n)), ..., h(u_m(x_1, ..., x_n))) \in \Omega^m$$

We use a similar notation for the m-tuple V. Let $F \in \mathbb{F}_p[X_1, ..., X_n]$. It is said that F *depends* on (U, V), which we denote by $F \in \langle U, V \rangle$, when there exists a linear decomposition

$$F = \sum_{1 \leq i,j \leq m} a_{i,j} \cdot u_i \cdot u_j + \sum_{1 \leq i \leq m} b_i \cdot v_i,\ a_{i,j}, b_i \in \mathbb{Z}_p$$

Let U, V be as above and $F \in \mathbb{F}_p[X_1, ..., X_n]$. The (U, V, F)-General Diffie-Hellman Exponent problems are defined as follows.

Definition 2 *((U, V, f)-GDHE) : Given the tuple*

$$H(x_1, ..., x_n) = \Big([U(x_1, ..., x_n)]G_0,\ g^{V(x_1, ..., x_n)} \Big) \in \mathbb{G}^m \times \mathbb{G}_T^m$$

compute $g^{F(x_1, ..., x_n)}$.

Definition 3 *((U, V, F)-GDDHE). Given $H(x_1, ..., x_n) \in \mathbb{G}^m \times \mathbb{G}_T^m$ as above and $T \in \mathbb{G}_T$, decide whether $T = g^{F(x_1, ..., x_n)}$.*

Definition 4 *The advantage of any probabilistic polynomial time algorithm $\mathcal{A}$ in solving the $(U, V, F) - GDDHE$ problem in $\mathbb{G}$ is defined as*

$$Adv_{\mathcal{A}}^{(U,V,F)-GDDHE} = |Pr[\mathcal{A}(U, V, F, g^{F(x_1, ..., x_n)}) = 1] - Pr\left[\mathcal{A}(U, V, F, T) = 1\right]|$$

The (U, V, F)-GDDHE Assumption is that, for any probabilistic polynomial time algorithm $\mathcal{A}$, the advantage $Adv_{\mathcal{A}}^{(U,V,F)-GDDHE}$ is negligibly small.

Complexity Bound in Generic Bilinear Groups We state the following upper bound in the framework of the generic group model. We are given oracles to compute the induced group action on $\mathbb{G}, \mathbb{G}_T$, and an oracle to compute a non-degenerate bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$. We refer to $\mathbb{G}$ as a generic bilinear group. The following theorem gives an upper bound on the advantage of a generic algorithm in solving the decision $(U, V, F) - GDDHE$ problem.

Theorem 1. *Let $U, V \in \mathbb{F}_p[X_1, ..., X_n]$ be two m-tuples of n-variate polynomials over $\mathbb{F}_p$ and let $F \in \mathbb{F}_p[X_1, ..., X_n]$. Let d_U (resp. d_V, d_F) denote the maximal degree of elements of U (resp. of V, F) and pose $d = max(2d_U, d_V, d_F)$. If $F \notin \langle U, V \rangle$ then for any generic-model adversary $\mathcal{A}$ totalizing at most q queries to the oracles (group operations in $\mathbb{G}, \mathbb{G}_T$ and evaluations of e) which is given $H(x_1, ..., x_n)$ as input and tries to distinguish $g^{F(x_1, ..., x_n)}$ from a random value in $\mathbb{G}_T$, one has*

$$Adv(A) \leq \frac{(q + 2m + 2)^2 \cdot d}{2p}$$

We refer to [8] for a proof that $(U, V, F) - GDHE$ and $(U, V, F) - GDDHE$ have generic security when $F \notin \langle U, V \rangle$. In our constructions, the order of the groups (p) that we consider is exponential in the security parameter λ.

2.3 Multi-Receiver Identity-Based Signcryption($mIBSC$)

A generic $mIBSC$ for sending a single message to t users consists of the following probabilistic polynomial time algorithms,

- ***Setup***(k, N). Given a security parameter k and the size of the maximal set of receivers[5] N, the Private Key Generator (PKG) generates the public parameters $params$ and master secret key MSK of the system.
- ***Extract***(ID, MSK). Given an identity ID, the PKG computes the corresponding private key S_{ID}
- ***Signcrypt***$(m, ID_A, ID_1, ID_2, ...ID_t, S_A)$. To send a message m to $(ID_1, ID_2, ...ID_t)$, a user with identity ID_A runs this algorithm to obtain the signcrypted ciphertext σ.
- ***Designcrypt***$(\sigma, ID_A, ID_i, S_i)$. When a user with identity ID_i and private key S_i receives the signcrypted ciphertext σ and runs this algorithm to obtain either the plain text m or $\perp$ according as whether σ was a valid signcryption from identity ID_A to or not.

2.4 Security Model

The notion of semantic security of public key encryption was extended to identity-based signcryption scheme by Malone-Lee in [21]. We describe the security models for *confidentiality* and *unforgeability* below.

Confidentiality The standard notion of Confidentiality for $m\mathcal{IBSC}$ schemes is *Chosen Ciphertext Security (CCA)* and *Chosen Plaintext Security (CPA)* against Static Adversaries.

A multi-receiver ID-based signcryption scheme is semantically secure against chosen ciphertext attacks (IND-mIBSC-CCA) if no probabilistic polynomial time adversary $\mathcal{A}$ has a non-negligible advantage in the following game.

1. Setup : The challenger $\mathcal{C}$ runs the *Setup* algorithm to generate the master public key $params$ and the master secret key MSK. He gives $params$ to the adversary $\mathcal{A}$. The adversary $\mathcal{A}$ outputs the set of target identities $\mathcal{S}^* = \{ID_1^*, ID_2^*, \ldots, ID_t^*\}$.
2. In the first phase, $\mathcal{A}$ makes polynomially bounded number of queries to the following oracles.
   (a) **Extract Oracle** ($\mathcal{O}_{Extract}$) — $\mathcal{A}$ produces an identity ID and queries for the secret key of ID. The *Extract Oracle* returns S_{ID} to $\mathcal{A}$ provided $ID \notin \mathcal{S}^*$.
   (b) **Signcrypt Oracle** ($\mathcal{O}_{Signcrypt}$) — $\mathcal{A}$ produces a message m, sender identity ID_A and a list of receiver identities $ID_1, ID_2, \ldots, ID_t$. $\mathcal{C}$ computes the secret key S_A by using *Extract*(ID_A, MSK) and returns to the adversary $\mathcal{A}$, the signcrypted ciphertext σ by using *Signcrypt*$(m, ID_A, ID_1, ID_2, \ldots, ID_t, S_A)$.
   (c) **Designcrypt Oracle** ($\mathcal{O}_{Designcrypt}$) — $\mathcal{A}$ produces a sender identity ID_A, receiver identity ID_B and a signcryption σ. The challenger $\mathcal{C}$ computes the secret key S_B from *Extract*(ID_B, MSK), returning the result of *Designcrypt*$(\sigma, ID_A, ID_B, S_B)$ to $\mathcal{A}$. The result returned is $\perp$ if σ is an invalid signcrypted ciphertext from ID_A to ID_B.
3. $\mathcal{A}$ produces two messages m_0 and m_1 of equal length from the message space $\mathcal{M}$ and an arbitrary sender identity ID_A^*. The challenger $\mathcal{C}$ flips a coin, sampling a bit $b \leftarrow \{0, 1\}$ and computes $\sigma^* = Signcrypt\,(m_b, ID_A^*, ID_1^*, ID_2^*, \ldots, ID_t^*, S_A^*)$. σ^* is returned to $\mathcal{A}$ as challenge signcrypted ciphertext.
4. $\mathcal{A}$ is allowed to make polynomially bounded number of new queries as in Step 2 with the restrictions that it should not query the *Designcryption Oracle* for the designcryption of σ^* and the *Extract Oracle* for the secret keys of any of $\{ID_1^*, ID_2^*, \ldots, ID_t^*\}$.
5. At the end of this game, $\mathcal{A}$ outputs a bit b'. $\mathcal{A}$ wins the game if $b' = b$.

[5] This input is optional. Certain specific schemes may not need this input

We define the advantage of the adversary $\mathcal{A}$ as

$$Adv_{\mathcal{A}}^{mIBSC-CCA} = |Pr\left[b = b'\right] - \frac{1}{2}|$$

Note. We analogously define security against chosen plaintext attacks (IND-mIBSC-CPA) by preventing the adversary from issuing Designcryption Queries in the above game.

Unforgeability A signcryption scheme is existentially unforgeable under chosen message attack (EUF-mIBSC-CMA) if no probabilistic polynomial time adversary $\mathcal{A}$ has a non-negligible advantage in the following game.

1. The challenger $\mathcal{C}$ runs the *Setup* algorithm to generate the master public and private keys *params* and MSK respectively. $\mathcal{C}$ gives system public parameters *params* to $\mathcal{A}$. $\mathcal{A}$ outputs the target identity ID^* on which he would like to be challenged.
2. The adversary $\mathcal{A}$ makes polynomially bounded number of queries to the oracles as described in Step 2 of the confidentiality game with the constraint that no *Extract* query is made on ID^*.
3. Finally $\mathcal{A}$ produces a signcrypted ciphertext σ^* along with the receiver's identities $ID_1^*, ID_2^*, \ldots, ID_t^*$. $\mathcal{A}$ wins the game if
   - The result of $Designcrypt(\sigma^*, ID_A^*, ID_i^*)$ for some $1 \leq i \leq t$ results in a valid message m^*.
   - No query to $\mathcal{O}_{Signcrypt}$ involved m^*, ID_A^* and any set of receivers.

3 The Schemes

3.1 Construction 1 - $m\mathcal{I}BSC_1$

In this section, we present an mIBSC scheme that is an extension of the Signcryption scheme of Barreto et. al [5]. It requires no pairing computations during Signcryption and just two pairing computations during Designcryption. This is the computationally the most efficient mIBSC scheme to date. The public key and private key size are constant, independent of the size of the subset of the receivers, while the size of the ciphertext is equal to that of the receiver set.

$m\mathcal{I}BSC_1$ has the following algorithms.

- ***Setup***(λ) The security parameter of the scheme is λ. $\mathbb{G}_1$, $\mathbb{G}_2$ are two groups of prime order p, where $|p| = \lambda$. P and Q are generators of $\mathbb{G}_1$ and e is a bilinear map defined as $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. Let n_0, n_1 denote the number of bits required to represent an identity and a message. Three hash functions $H_1 : \{0,1\}^{n_0} \rightarrow \mathbb{Z}_p^*$, $H_2 : \{0,1\}^{n_1} \times \mathbb{G}_2 \rightarrow \mathbb{Z}_p^*$, $H_3 : \mathbb{G}_2 \rightarrow \{0,1\}^{n_1}$ are used. The PKG chooses $s \in_R \mathbb{Z}_p^*$ and computes $R = sP$ and $g = e(P, Q)$. The public parameters are

  $$params = \langle \mathbb{G}_1, \mathbb{G}_2, Q, sQ, g, e(\cdot,\cdot), H_1, H_2, H_3 \rangle.$$

  The Master Secret Key is

  $$MSK = \langle s, P \rangle.$$

- ***Extract***(ID, MSK) The public key and private key of identity ID are $H_1(ID)$ and $S_{ID} = \frac{1}{H_1(ID)+s}P$ respectively.
- ***Signcrypt***$(m, ID_A, ID_1, ID_2, \ldots, ID_t, S_A)$ Suppose A wants to signcrypt a message m to t receivers with identities $ID_1, ID_2, \ldots, ID_t$. User A does the following.
  1. Choose $r \in_R \mathbb{Z}_p^*$

2. Compute the following.
   (a) $\alpha = g^r$
   (b) $h = H_2(m, \alpha)$
   (c) $Z_A = (r + h) S_A$
   (d) $c = m \oplus H_3(\alpha)$
   (e) $y_i = r.H_1(ID_i) Q + r.sQ$, for $1 \leq i \leq t$.
3. The signcrypted ciphertext is $\sigma = \langle c, Z_A, y_1, y_2, \ldots, y_t, \mathcal{L}\rangle$, where $\mathcal{L}$ is the list of receivers who can decrypt the message. Here, y_i is meant for the receiver ID_i.

- ***Designcrypt***$(\sigma, ID_A, ID_i, S_i)$ A receiver with identity ID_i uses his secret key S_i to designcrypt $\sigma = \langle c, Z_A, y_1, y_2, \ldots, y_t, \mathcal{L}\rangle$ from ID_A as follows.
  1. Compute the following.
     (a) $\alpha' = e(S_i, y_i)$
     (b) $m = c \oplus H_3(\alpha')$
     (c) $h = H_2(m, \alpha')$
  2. If $\alpha' = e(Z_A, H_1(ID_A) Q + sQ) g^{-h}$, return m. Otherwise, return $\perp$.

Security Properties

Definition 5 *($(U_1, V_1, F_1) - GDDHE$). Let $\mathcal{B} = (p, \mathbb{G}_1, \mathbb{G}_2, e(,))$ be a bilinear map group system and let $f, g_1, g_2, \ldots, g_t$ be pairwise co prime polynomials with pairwise distinct roots, of orders l for f and 1 for g_i. Let P_0 and Q_0 be generators of $\mathbb{G}_1$. Given*

$$\begin{pmatrix} P_0, sP_0, \ldots, s^{l-1}P_0 & s.f(s)P_0 \\ Q_0, sQ_0, s^2Q_0, s^3Q_0 & \gamma.s.g_1(s)Q_0, \gamma.s.g_2(s)Q_0, \ldots, \gamma.s.g_t(s)Q_0 \end{pmatrix}$$

and $T \in \mathbb{G}_2$, solving the $(U_1, V_1, F_1) - GDDHE$ problem consists of deciding whether T is equal to $e(P_0, Q_0)^{\gamma \cdot f(s)}$ or is some random element of $\mathbb{G}_2$.

Corollary 1 *(Generic security of $(U_1, V_1, F_1) - GDDHE$). For any probabilistic algorithm $\mathcal{A}$ that totalizes of at most q queries to the oracles performing the group operations in $\mathbb{G}_1, \mathbb{G}_2$ and the bilinear map $e(\cdot, \cdot)$,*

$$Adv^{GDDHE}(U_1, V_1, F_1, \mathcal{A}) \leq \frac{(q + 2(l + t + 4) + 2)^2 \cdot (l + 1)}{p}$$

Proof. Refer Appendix G

Theorem 2. *Assume that an IND-mIBSC-CCA adversary $\mathcal{A}$ has an advantage ϵ against $m\mathcal{IBSC}_1$, asking atmost l extraction queries. Then there is an algorithm $\mathcal{R}$ to solve the $(U_1, V_1, F_1) - GDDHE$ problem with advantage*

$$\epsilon' \geq \epsilon/2$$

Proof. Refer Appendix A

Theorem 3. *Assume that an EUF-mIBSC-CMA adversary $\mathcal{A}$ making l extraction queries, q_{H_i} queries to random oracles H_i (i= 1,2,3) and q_{sc} signcryption queries, has an advantage $\epsilon \geq 10(q_{sc} + 1)(q_{sc} + q_{H_2})/2^k$ against $m\mathcal{IBSC}_1$. Then there is an algorithm $\mathcal{R}$ to solve the $l - SDHP$ with advantage*

$$\epsilon' \geq 1/9$$

Proof. Refer Appendix B

3.2 Construction 2 - $m\mathcal{IBSC}_2$

We present a scheme that improves upon the previous scheme in terms of the length of ciphertext. It brings down this parameter to one-third the size of the set of receivers by grouping the users into sets of three. The size of the public keys and private keys remains constant-sized.This scheme presents the optimal trade off between ciphertext size and key storage cost.

$m\mathcal{I}BSC_2$ has the following algorithms.

- ***Setup***(λ) The security parameter of the scheme is λ. $\mathbb{G}_1$, $\mathbb{G}_2$ are two groups of prime order p, where $|p| = \lambda$. P and Q are generators of $\mathbb{G}_1$ and e is a bilinear map defined as $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. Let n_0 and n_1 denote the number of bits required to represent an identity and a message respectively. Three hash functions $H_1 : \{0,1\}^{n_0} \to \mathbb{Z}_p^*$, $H_2 : \{0,1\}^{n_1} \times \mathbb{G}_2 \to \mathbb{Z}_p^*$, $H_3 : \mathbb{G}_2 \to \{0,1\}^{n_1}$ are used. The PKG chooses $s \in_R \mathbb{Z}_p^*$ and computes $R = sP$ and $g = e(P,Q)$. The public parameters are
$$params = \langle \mathbb{G}_1, \mathbb{G}_2, sP, Q, sQ, s^2Q, s^3Q, g, e(\cdot,\cdot), H_1, H_2, H_3 \rangle.$$
The Master Secret Key is
$$MSK = \langle s, P \rangle.$$
- ***Extract***(ID, MSK) The public key and private key of identity ID are $H_1(ID)$ and $S_{ID} = \frac{1}{H_1(ID)+s}P$ respectively.
- ***Signcrypt***[6]$(m, ID_A, ID_1, ID_2, \ldots, ID_t, S_A)$ Let $t = 3n$. Suppose A wants to signcrypt a message m to t receivers with identities $ID_1, ID_2, \ldots, ID_t$. User A divides the users into groups of 3, where the i^{th} group consists of identities ID_{a_i}, ID_{b_i} and ID_{c_i} for $1 \le i \le n$. ID_A does the following.
  1. Choose $r \in_R \mathbb{Z}_p^*$
  2. Compute the following.
     (a) $\alpha = g^r$
     (b) $X = -rR$
     (c) $h = H_2(m, \alpha)$
     (d) $Z_A = (r+h)\,S_A$
     (e) $c = m \oplus H_3(\alpha)$
     (f) $y_i = r\,(s + H_1(ID_{a_i}))\,(s + H_1(ID_{b_i}))\,(s + H_1(ID_{c_i}))\,Q$ for $1 \le i \le n$
  3. The signcrypted ciphertext is $\sigma = \langle c, X, Z_A, y_1, y_2, \ldots, y_n, \mathcal{L} \rangle$, where $\mathcal{L}$ is the list of receivers who can decrypt the message.
- ***Designcrypt***$(\sigma, ID_A, ID_i, S_i)$ A receiver with identity ID_i uses his secret key S_i to designcrypt $\sigma = \langle c, Z_A, y_1, y_2, \ldots, y_n, \mathcal{L} \rangle$ from ID_A as follows. Let ID_i, ID_j and ID_k be the identities of the members present in the l^{th} group (can be retreived from $\mathcal{L}$).
  1. Compute the following.
     (a) $\alpha' = \left[e(S_i, y_l) \cdot e(X, (s + H_1(ID_j)\,H_1(ID_k))\,Q)\right]^{\frac{1}{H_1(ID_j) H_1(ID_k)}}$
     (b) $m = c \oplus H_3(\alpha')$
     (c) $h = H_2(m, \alpha')$
  2. If $\alpha' = e(Z_A, (H_1(ID_A)\,Q + sQ))\,g^{-h}$, return m. Otherwise, return $\perp$.

Correctness. As we can see,

$$\begin{aligned}\alpha' &= \left[e(S_i, y_l) \cdot e(X, (s + H_1(ID_j)\,H_1(ID_k))\,Q)\right]^{\frac{1}{H_1(ID_j) H_1(ID_k)}} \\ &= g^{r \cdot \left[(s+H_1(ID_j))(s+H_1(ID_k)) - (s^2 + H_1(ID_j) \cdot H_1(ID_k) \cdot s)\right] / H_1(ID_j) \cdot H_1(ID_k)} \\ &= g^r\end{aligned}$$

[6] Here we assume that the number of receivers is a multiple of three. If this is not the case, the scheme can be adjusted accordingly.

Security Properties

Definition 6 *($(U_2, V_2, F_2)-GDDHE$). Let $\mathcal{B} = (p, \mathbb{G}_1, \mathbb{G}_2, e(,))$ be a bilinear map group system and $f, g_1, g_2, \ldots, g_n$ be pairwise co prime polynomials with pairwise distinct roots, of orders l for f and 3 for g_i. Let P_0 and Q_0 be generators of $\mathbb{G}_1$. Given*

$$\begin{pmatrix} P_0, sP_0, \ldots, s^{l-1}P_0 & s.f(s)P_0, s^2.f(s)P_0, s^3.f(s)P_0 & \gamma.s.f(s)P_0 \\ Q_0, sQ_0, \ldots, s^5Q_0 & \gamma.s.g_1(s)Q_0, \gamma.s.g_2(s)Q_0, \ldots, \gamma.s.g_n(s)Q_0 & \end{pmatrix}$$

and $T \in \mathbb{G}_2$, solving the $(U_2, V_2, F_2) - GDDHE$ problem consists of deciding whether T is equal to $e(P_0, Q_0)^{\gamma \cdot f(s)}$ or is some random element of $\mathbb{G}_2$.

Corollary 2 *(Generic security of $(U_2, V_2, F_2) - GDDHE$). For any probabilistic algorithm $\mathcal{A}$ that totalizes of at most q queries to the oracles performing the group operations in $\mathbb{G}_1, \mathbb{G}_2$ and the bilinear map $e(\cdot, \cdot)$,*

$$Adv^{GDDHE}(U_2, V_2, F_2, \mathcal{A}) \leq \frac{(q + 2(l + n + 10) + 2)^2 \cdot (l + 2)}{p}$$

Proof. Refer Appendix G

Theorem 4. *Assume that an IND-mIBSC-CCA adversary $\mathcal{A}$ has an advantage ϵ against $m\mathcal{I}BSC_2$, asking at most l extraction queries. Then there is an algorithm $\mathcal{R}$ to solve the $(U_2, V_2, F_2) - GDDHE$ problem with advantage*

$$\epsilon' \geq \epsilon/2$$

Proof. Refer Appendix C

Theorem 5. *Assume that an EUF-mIBSC-CMA adversary $\mathcal{A}$ making l extraction queries, q_{H_i} queries to random oracles H_i (i= 1,2,3) and q_{sc} signcryption queries, has an advantage $\epsilon \geq 10(q_{sc} + 1)(q_{sc} + q_{H_2})/2^k$ against $m\mathcal{I}BSC_2$. Then there is an algorithm $\mathcal{R}$ to solve the $(l + 3) - SDHP$ with advantage*

$$\epsilon' \geq 1/9$$

Proof. Refer Appendix D

3.3 Construction 3 - $m\mathcal{I}BSC_3$

In this section, we further improve upon the scheme presented in the previous section. We progressively increase the size of the groups and ultimately achieve constant-sized ciphertexts and private keys. The size of the public keys is that of the maximal subset of receivers.

$m\mathcal{I}BSC_3$ has the following algorithms.

- ***Setup***(λ, N) The security parameter of the scheme is λ and N is the maximal size of the set of receivers. $\mathbb{G}_1$, $\mathbb{G}_2$ are two groups of prime order p, where $|p| = \lambda$. P and Q are generators of $\mathbb{G}_1$ and e is a bilinear map defined as $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. Let n_0 and n_1 denote the number of bits required to represent an identity and a message respectively. Three hash functions $H_1 : \{0, 1\}^{n_0} \rightarrow \mathbb{Z}_p^*$, $H_2 : \{0, 1\}^{n_1} \times \mathbb{G}_2 \rightarrow \mathbb{Z}_p^*$, $H_3 : \mathbb{G}_2 \rightarrow \{0, 1\}^{n_1}$ are used. The PKG chooses $s \in_R \mathbb{Z}_p^*$ and computes $R = sP$ and $g = e(P, Q)$. The public parameters are

$$params = \langle \mathbb{G}_1, \mathbb{G}_2, sP, Q, sQ, s^2Q, \ldots, s^NQ, g, e(\cdot,\cdot), H_1, H_2, H_3 \rangle.$$

The Master Secret Key is

$$MSK = \langle s, P \rangle.$$

- ***Extract***(ID, MSK) The public key and private key of identity ID are $H_1\left(ID\right)$ and $S_{ID} = \frac{1}{H_1(ID)+s}P$ respectively.
- ***Signcrypt***$(m, ID_A, ID_1, ID_2, \ldots, ID_t, S_A)$ Suppose A wants to signcrypt a message m to t receivers with identities $ID_1, ID_2, \ldots, ID_t$. User A does the following.
  1. Choose $r \in_R \mathbb{Z}_p^*$
  2. Compute the following.
     (a) $\alpha = g^r$
     (b) $h = H_2\left(m, \alpha\right)$
     (c) $Z_A = \left(r + h\right) S_A$
     (d) $c = m \oplus H_3\left(\alpha\right)$
     (e) $y = \left[\prod_{i=1}^{t} H_1\left(ID_i\right)\right] rQ$
  3. The signcrypted ciphertext is $\sigma = \langle c, Z_A, y, \mathcal{L} \rangle$, where $\mathcal{L}$ is the list of receivers who can decrypt the message.
- ***Designcrypt***$(\sigma, ID_A, ID_i, S_i)$ A receiver with identity ID_i uses his secret key S_i to designcrypt $\sigma = \langle c, Z_A, y, \mathcal{L} \rangle$ from ID_A as follows.
  1. Compute the following.
     (a) $\alpha' = \left[e\left(S_i, y\right).e\left(X, \frac{1}{s}\left[\prod_{j=1,j\neq i}^{t}\left(s + H_1\left(ID_j\right)\right) - \prod_{j=1,j\neq i}^{t} H_1\left(ID_j\right)\right] Q\right)\right]^{\frac{1}{\prod_{j=1,j\neq i}^{t} H_1\left(ID_j\right)}}$
     (b) $m = c \oplus H_3\left(\alpha'\right)$
     (c) $h = H_2\left(m, \alpha'\right)$
  2. If $\alpha' = e\left(Z_A, \left(H_1\left(ID_A\right)Q + sQ\right)\right) g^{-h}$, return m. Otherwise, return $\perp$.

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

$$\begin{aligned}
\beta &= e\left(S_i, y\right).e(X, \frac{1}{s}[\prod_{j=1,j\neq i}^{t}\left(s + H_1\left(ID_j\right)\right) - \prod_{j=1,j\neq i}^{t} H_1\left(ID_j\right)]Q) \\
&= e\left(P, Q\right)^{r\cdot\left\{\prod_{j=1,j\neq i}^{t}[s+H_1(ID_j)] - \left[\prod_{j=1,j\neq i}^{t}(s+H_1(ID_j)) - \prod_{j=1,j\neq i}^{t} H_1(ID_j)\right]\right\}} \\
&= g^{r\cdot\prod_{j=1,j\neq i}^{t} H_1(ID_j)} \\
\alpha &= \beta^{\frac{1}{\prod_{j=1,j\neq i}^{t} H_1\left(ID_j\right)}}
\end{aligned}$$

Security Properties

Definition 7 *$((U_3, V_3, F_3) - GDDHE)$. Let $\mathcal{B} = (p, \mathbb{G}_1, \mathbb{G}_2, e(,))$ be a bilinear map group system and let f and g be two co prime polynomials with pairwise distinct roots, of respective orders l and t. Let P_0 and Q_0 be generators of $\mathbb{G}_1$. Given*

$$\begin{pmatrix} P_0, sP_0, \ldots, s^{l-1}P_0 & s.f(s)P_0, s^2.f(s)P_0, s^3.f(s)P_0 & \gamma.s.f(s)P_0 \\ Q_0, sQ_0, \ldots, s^{N+3}Q_0 & & \gamma.s.g(s)Q_0 \end{pmatrix}$$

and $T \in \mathbb{G}_2$, solving the $(U_3, V_3, F_3) - GDDHE$ problem consists of deciding whether T is equal to $e(P_0, Q_0)^{\gamma\cdot f(s)}$ or is some random element of $\mathbb{G}_2$.

Corollary 3 *(Generic security of $(U_3, V_3, F_3) - GDDHE$). For any probabilistic algorithm $\mathcal{A}$ that totalizes of at most q queries to the oracles performing the group operations in $\mathbb{G}_1, \mathbb{G}_2$ and the bilinear map $e(\cdot,\cdot)$,*

$$Adv^{GDDHE}(U_3, V_3, F_3, \mathcal{A}) \leq \frac{(q + 2(l + N + 9) + 2)^2 \cdot d}{2p}$$

with $d = 2 \cdot max(N + 3, l + 1)$.

Proof. Refer Appendix G

Theorem 6. *Assume that an IND-mIBSC-CCA adversary $\mathcal{A}$ has an advantage ϵ against $mIBSC_3$, asking at most l extraction queries. Then there is an algorithm $\mathcal{R}$ to solve the $(U_3, V_3, F_3) - GDDHE$ problem with advantage*

$$\epsilon' \geq \epsilon/2$$

Proof. Refer Appendix E

Theorem 7. *Assume that an EUF-mIBSC-CMA adversary $\mathcal{A}$ making l extraction queries, q_{H_i} queries to random oracles H_i (i= 1,2,3) and q_{sc} signcryption queries, has an advantage $\epsilon \geq 10(q_{sc} + 1)(q_{sc} + q_{H_2})/2^k$ against $mIBSC_3$. Then there is an algorithm $\mathcal{R}$ to solve the $(l + N) - SDHP$ with advantage*

$$\epsilon' \geq 1/9$$

Proof. Refer Appendix F

A Note on Chosen-ciphertext security As noted in [15], given a chosen-plaintext secure Multi-receiver encryption scheme, one can convert it into a scheme that is secure in a very strong sense - indistinguishability against chosen ciphertext attacks in the random oracle model. Applying the Fujisaki-Okamoto transform [18] [19] guarantees CCA security in the random oracle model assuming the one-wayness of the underlying encryption scheme. REACT[22] gives the same guarantee. Therefore CCA security can be realized at negligible cost in the Random Oracle model. However combining the encryption scheme with an ID-based strong-unforgeable signature scheme as proposed in [10] may allow to avoid random oracles when such a combination is possible.

4 Conclusion

We presented three new multi-receiver signcryption schemes, each achieving optimal efficiency in at least one of computation, storage and transmission cost. No scheme to date has achieved such a high level of efficiency in the parameters specified. We also formally prove the security of these schemes in the sense of confidentiality and unforgeability, based on the $l - SDHP$ and the $GDDHE$ assumptions.

To our knowledge, no public key multi-receiver encryption scheme is known to resist fully adaptive adversaries. We leave this as an open problem. Another interesting problem would be to design a scheme that is secure under weaker assumptions and achieves efficiency comparable to ours.

References

1. Michel Abdalla, Eike Kiltz, and Gregory Neven. Generalized key delegation for hierarchical identity-based encryption. In *ESORICS*, pages 139–154, 2007.
2. Joonsang Baek, Reihaneh Safavi-Naini, and Willy Susilo. Efficient multi-receiver identity-based encryption and its application to broadcast encryption. In *Public Key Cryptography*, pages 380–397, 2005.
3. Joonsang Baek, Ron Steinfeld, and Yuliang Zheng. Formal proofs for the security of signcryption. In *Public Key Cryptography*, pages 80–98, 2002.
4. Manuel Barbosa and Pooya Farshim. Efficient identity-based key encapsulation to multiple parties. In *IMA Int. Conf.*, pages 428–441, 2005.
5. Paulo S. L. M. Barreto, Benoît Libert, Noel McCullagh, and Jean-Jacques Quisquater. Efficient and provably-secure identity-based signatures and signcryption from bilinear maps. In *ASIACRYPT*, pages 515–532, 2005.
6. Olivier Baudron, David Pointcheval, and Jacques Stern. Extended notions of security for multicast public key cryptosystems. In *ICALP*, pages 499–511, 2000.
7. Mihir Bellare, Alexandra Boldyreva, and Silvio Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In *EUROCRYPT*, pages 259–274, 2000.
8. Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In *EUROCRYPT*, pages 440–456, 2005.
9. Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. In *CRYPTO*, pages 213–229, 2001.
10. Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In *EUROCRYPT*, pages 207–222, 2004.
11. Sanjit Chatterjee and Palash Sarkar. Multi-receiver identity-based key encapsulation with shortened ciphertext. In *INDOCRYPT*, pages 394–408, 2006.
12. L. Chen, Keith Harrison, David Soldera, and Nigel P. Smart. Applications of multiple trust authorities in pairing based cryptosystems. In *InfraSec*, pages 260–275, 2002.
13. Liqun Chen and John Malone-Lee. Improved identity-based signcryption. In *Public Key Cryptography*, pages 362–379, 2005.
14. Cécile Delerablée. Identity-based broadcast encryption with constant size ciphertexts and private keys. In *ASIACRYPT*, pages 200–215, 2007.
15. Cécile Delerablée, Pascal Paillier, and David Pointcheval. Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys. In *Pairing*, pages 39–59, 2007.
16. Shanshan Duan and Zhenfu Cao. Efficient and provably secure multi-receiver identity-based signcryption. In *ACISP*, pages 195–206, 2006.
17. Amos Fiat and Moni Naor. Broadcast encryption. In *CRYPTO*, pages 480–491, 1993.
18. Eiichiro Fujisaki and Tatsuaki Okamoto. How to enhance the security of public-key encryption at minimum cost. In *Public Key Cryptography*, pages 53–68, 1999.
19. Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *CRYPTO*, pages 537–554, 1999.
20. Benot Libert and Jean-Jacques Quisquater. New identity based signcryption schemes from pairings. Cryptology ePrint Archive, Report 2003/023, 2003.
21. John Malone-Lee. Identity-based signcryption. Cryptology ePrint Archive, Report 2002/098, 2002.
22. Tatsuaki Okamoto and David Pointcheval. React: Rapid enhanced-security asymmetric cryptosystem transform. In *CT-RSA*, pages 159–175, 2001.
23. David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *J. Cryptology*, 13(3):361–396, 2000.
24. Ryuichi Sakai and Jun Furukawa. Identity-based broadcast encryption. Cryptology ePrint Archive, Report 2007/217, 2007. http://eprint.iacr.org/.
25. S. Sharmila Deva Selvi, S. Sree Vivek, Ragavendran Gopalakrishnan, Naga Naresh Karuturi, and C. Pandu Rangan. Cryptanalysis of id-based signcryption scheme for multiple receivers. Cryptology ePrint Archive, Report 2008/238, 2008.
26. Adi Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO*, pages 47–53, 1984.
27. Nigel P. Smart. Access control using pairing based cryptography. In *CT-RSA*, pages 111–121, 2003.
28. Nigel P. Smart. Efficient key encapsulation to multiple parties. In *SCN*, pages 208–219, 2004.
29. Yong Yu, Bo Yang, Xinyi Huang, and Mingwu Zhang. Efficient identity-based signcryption scheme for multiple receivers. In *ATC*, pages 13–21, 2007.
30. Yuliang Zheng. Digital signcryption or how to achieve cost(signature & encryption) << cost(signature) + cost(encryption). In *CRYPTO*, pages 165–179, 1997.

A Proof of Theorem 2

Both the adversary and the challenger are given as input l the total number of extraction queries and q the total number of random oracle queries that can be issued by the adversary. Algorithm

$\mathcal{R}$ is given as input a group system $\mathcal{B} = (p, \mathbb{G}_1, \mathbb{G}_2, e(\cdot,\cdot))$, and a $(U_1, V_1, F_1)-GDDHE$ instance in $\mathcal{B}$. We thus have $f, g_1, g_2, g_3, \ldots, g_t$ which are pairwise co prime polynomials with pairwise distinct roots, of orders l for f and 1 for g_i , and

$$\begin{pmatrix} P_0, sP_0, \ldots, s^{l-1}P_0 & s.f(s)P_0 \\ Q_0, sQ_0, s^2Q_0, s^3Q_0 & \gamma.s.g_1(s)Q_0, \gamma.s.g_2(s)Q_0, \ldots, \gamma.s.g_t(s)Q_0 \end{pmatrix}$$

and $T \in \mathbb{G}_2$, which is either is equal to $e(P_0, Q_0)^{\gamma \cdot f(s)}$ or to some random element of $\mathbb{G}_2$

Notations.

- $f(X) = \prod_{i=1}^{l}(X + x_i)$
- $g_i(X) = (X + x_i)$ for $l + 1 \leq i \leq l + t$
- $f_i(x) = \frac{f(x)}{x+x_i}$ for $i \in [1, l]$, which is a polynomial of degree $l - 1$

Init Phase: The adversary A outputs a t-set $\mathcal{S}^* = \{ID_1^*, ..., ID_t^*\}$ of identities that he wants to attack.

Setup Phase: To generate the system parameters, $\mathcal{R}$ formally sets $P = f(s)P_0$ (i.e. without computing it) and sets

- $Q = Q_0$
- $R = s.f(s)P_0 = sP$
- $g = e(P_0, Q_0)^{f(s)} = e(P, Q)$

$\mathcal{R}$ then defines the Public Key PK as $\langle Q, sQ, R, g\rangle$. Note $\mathcal{R}$ cannot compute the value of P.

Query phase 1: At any time the adversary $\mathcal{A}$ can query the following random oracles. To respond to these queries, $\mathcal{R}$ maintains three lists $\mathcal{L}_{H_1}, \mathcal{L}_{H_2}, \mathcal{L}_{H_3}$.

1. H_1 Queries: The list $\mathcal{L}_{H_1}$ contains at the beginning: $(*, x_i)_{i=1}^{l}$ $(ID_i, x_i)_{i=l+1}^{l+t}$ (we choose to note $*$ an empty entry in $\mathcal{L}_{H_1}$). When the adversary issues a hash query on identity ID_i,
   - If ID_i already appears in the list $\mathcal{L}_{H_1}$, R responds with the corresponding x_i.
   - Otherwise, $\mathcal{R}$ picks an x_i for some $(*, x_i)$ in $\mathcal{L}_{H_1}$, returns $H(ID_i) = x_i$, and extends the list with (ID_i, x_i).
2. Extraction query (ID_i): The challenger runs Extract on $ID_i \notin \mathcal{S}^*$ and forwards the resulting private key to the adversary. To generate the keys,
   - If $\mathcal{A}$ has already issued a hash query on ID_i, then $\mathcal{R}$ uses the corresponding x_i to compute $$S_{ID_i} = f_i(s)P_0 = \frac{1}{s+x_i}P$$
   - Otherwise, $\mathcal{R}$ sets $H(ID_i) = x_i$, computes the corresponding S_{ID_i} exactly as above, and completes the list $\mathcal{L}_{H_1}$ for ID_i.
3. H_2 queries: To respond to these queries $\mathcal{R}$ maintains a list of tuples called the $\mathcal{L}_{H_2}$ list. Each entry in the list is a tuple of the form (m_i, α_i, h_i). Initially the list is empty. To respond to query (m_i, α_i) algorithm $\mathcal{R}$ does the following:
   - If the query (m_i, α_i) already appears in the list in a tuple (m_i, α_i, h_i) then respond with $H_2(m_i, \alpha_i) = h_i$.
   - Otherwise, $\mathcal{R}$ just picks a random $h_i \leftarrow \mathbb{Z}_p^*$ and adds the tuple (m_i, α_i, h_i) to the list
   - It responds to $\mathcal{A}$ with $H_2(m_i, \alpha_i) = h_i$.
4. H_3 queries: To respond to these queries $\mathcal{R}$ maintains a list of tuples called the $\mathcal{L}_{H_3}$ list. Each entry in the list is a tuple of the form (α_i, h_i). Initially the list is empty. To respond to query α_i algorithm $\mathcal{R}$ does the following:
   - If the query α_i already appears in the list in a tuple (α_i, h_i) then respond with $H_3(\alpha_i) = h_i$.
   - Otherwise, $\mathcal{R}$ just picks a random $h_i \leftarrow \{0, 1\}^n$ where n is the number of bits in a message and adds the tuple (α_i, h_i) to the list
   - It responds to A with $H_3(\alpha_i) = h_i$.
5. Signcryption Queries : Of the form $(m, ID_A, ID_1, ID_2, \ldots, ID_n)$ If $ID_A \notin \mathcal{S}^*$, $\mathcal{R}$ has the secret key of ID_A and can therefore proceed as in normal *Signcrypt* algorithm. Otherwise, he does the following
   - Picks $r \in_R \mathbb{Z}_p^*$ and set $Z_A = r.sP$
   - Picks $h \in_R \mathbb{Z}_p^*$
   - Computes $y_i = r\,(s(x_A + s) - h)\,(s + x_i)Q$ for $1 \leq i \leq n$
   - Computes $\alpha = e(Z_A, (s + x_A)Q).g^{-h \cdot r}$ and picks a random string V of length same as the message.

- Returns $\sigma = \langle m \oplus V, Z_A, y_1, y_2, \ldots, y_n \rangle$ and enters the tuples $(m, \alpha, h \cdot r)$ and (α, V) in $\mathcal{L}_2$ and $\mathcal{L}_3$ respectively.

As one can see, σ will pass off as a valid ciphertext because

$$\begin{aligned} e(Z_A, (s + x_A)Q).g^{-h \cdot r} &= g^{r(s(x_A+s)-h)} \\ &= e(S_i, y_i) \end{aligned}$$

6. Designcryption Queries : Queries of the form (σ, ID_A, ID_i). $\mathcal{R}$ retrieves (c, Z_A, y_i) from σ and searches $\mathcal{L}_2$ for an entry of the form (m_j, α_j, h_j) that satisfies the following condition

$$\alpha_j = e(Z_A, H_1(ID_A)Q + sQ) \cdot g^{-h_j}$$

If such an entry is present, $\mathcal{R}$ returns m_j. Otherwise, he returns $\perp$.

We note that if σ is a valid ciphertext, then h_j is the correct value of $H_2(m_j, \alpha_j)$, for some (m_j, α_j). If $\mathcal{A}$ has queried the H_2 oracle with these values, then an entry of the form (m_j, α_j, h_j) will be present in $\mathcal{L}_2$, which $\mathcal{R}$ retrieves. The only other case in which $\mathcal{A}$ can produce a valid ciphertext is by correctly guessing the hash value of (m_j, α_j) without querying it. In a perfect simulation, this ciphertext using the correct guessed value should pass of as a valid one. But in our simulation, this does not happen, and we return $\perp$. However we note that this event occurs only with a probability of $1/p$ which is of the order of $1/2^k$, which is negligible in the security parameter k.

Challenge Phase: When $\mathcal{A}$ decides that Query phase 1 is over, he gives two messages m_0 and m_1 and a sender's identity ID_A, algorithm $\mathcal{R}$ sets $\alpha = T$, picks random Z_A, c and responds with the challenge ciphertext

$\sigma^* = \langle c, Z_A, y_1, y_2, \ldots, y_t\mathcal{L} \rangle$ where $y_i = \gamma.s.g_i(s)Q_0$. Note that if $T = g^\gamma$, then y_i is a valid encryption of $\alpha = g^\gamma$, although σ^* may not be a valid ciphertext.

Query phase 2: The adversary continues to issue queries with the constraint that no extraction query is made on ID_i for $ID_i \in \mathcal{S}^*$

Guess Phase: Finally, the adversary A outputs a guess b

$\mathcal{R}$ ignores the answer and searches $\mathcal{L}_{H_3}$ for an entry of the form $(T, *)$. If present, he outputs 1 (indicating that $T = g^\gamma$). Otherwise, he outputs 0.

We note that if y_i is a valid encryption of T, then an adversary with a non-negligible advantage in the above game must have issued a H_3 query on T, in which case an entry of the form $(T, *)$ will be present in $\mathcal{L}_{H_3}$.

$$\begin{aligned} Adv_{\mathcal{R}}^{GDDHE}(U_1, V_1, F_1) &= Pr\left[b = b' | real\right] - Pr\left[b = b' | random\right] \\ &= \frac{1}{2} \cdot Adv_{\mathcal{A}}^{mIBSC-CCA} \end{aligned}$$

B Proof of Theorem 3

Let l be the maximum number of extraction queries that can be queried by the adversary $\mathcal{A}$. Algorithm $\mathcal{R}$ takes as input $(Q, sQ, s^2Q, \ldots, s^lQ)$ and aims to find a pair $(c, \frac{1}{c+s}Q)$. In a setup phase, it builds a generator $G \in \mathbb{G}_1$ such that it knows $l-1$ pairs $(x_i, \frac{1}{x_i+s}G)$ for $x_1, \ldots, x_{l-1} \in_R \mathbb{Z}_p^*$. To do so,

- It picks $\beta \in_R \mathbb{Z}_p^*$ and sets $P = \beta Q$
- It picks $x_1, x_2, \ldots, x_{l-1} \in_R \mathbb{Z}_p^*$ and expands $f(z) = \prod_{i=1}^{l-1}(z + x_i)$ to obtain $c_0, c_1, \ldots, c_{l-1} \in \mathbb{Z}_p^*$ so that $f(z) = \sum_{i=0}^{l-1} c_i z^i$.
- It sets generators $H = \sum_{i=0}^{l-1} c^i(s^iQ) = f(s)Q$ and $G = \beta H = f(s)P$. It computes $sH = \sum_{i=1}^{l} c_{i-1}(s^iQ)$ and $g = e(G, H)$ and makes $\langle H, sH, g \rangle$ public.

- For $1 \leq i \leq l-1$, $\mathcal{R}$ expands $f_i(z) = \frac{f(z)}{(z+x_i)} = \sum_{i=0}^{l-2} d_i z^i$ and $\frac{1}{x_i+s}G = \beta \cdot f_i(s)P$

$\mathcal{A}$ inputs ID^* on which it would like to be challenged and a set of receivers $ID_1^*, ID_2^*, \ldots, ID_t^*$. $\mathcal{R}$ is then ready to answer $\mathcal{A}$'s queries along the course of the game. It first initializes a counter i to 1. For simplicity, we assume that queries to H_1 are distinct, and that any query involving an identifier ID is preceded by the random oracle query $H_1(ID)$.

1. H_1 queries on an identity ID : $\mathcal{R}$ returns a random $x^* \in_R \mathbb{Z}_p^*$ if $ID = ID^*$. Otherwise, $\mathcal{R}$ answers $x = x_i$ and increments i. $\mathcal{R}$ stores (ID, x) in a list $\mathcal{L}_{H_1}$.
2. H_2 queries: To respond to these queries $\mathcal{R}$ maintains a list of tuples called the $\mathcal{L}_{H_2}$ list. Each entry in the list is a tuple of the form (m_i, α_i, h_i). Initially the list is empty. To respond to query (m_i, α_i) algorithm $\mathcal{R}$ does the following:
   - If the query (m_i, α_i) already appears in the list in a tuple (m_i, α_i, h_i) then respond with $H_2(m_i, \alpha_i) = h_i$.
   - Otherwise, $\mathcal{R}$ just picks a random $h_i \leftarrow \mathbb{Z}_p^*$ and adds the tuple (m_i, α_i, h_i) to the list
   - It responds to $\mathcal{A}$ with $H_2(m_i, \alpha_i) = h_i$.
3. H_3 queries: To respond to these queries $\mathcal{R}$ maintains a list of tuples called the $\mathcal{L}_{H_3}$ list. Each entry in the list is a tuple of the form (α_i, h_i). Initially the list is empty. To respond to query α_i algorithm $\mathcal{R}$ does the following:
   - If the query α_i already appears in the list in a tuple (α_i, h_i) then respond with $H_3(\alpha_i) = h_i$.
   - Otherwise, $\mathcal{R}$ just picks a random $h_i \leftarrow \{0,1\}^n$ where n is the number of bits in a message and adds the tuple (α_i, h_i) to the list
   - It responds to $\mathcal{A}$ with $H_3(\alpha_i) = h_i$.
4. Key extraction queries on $ID \neq ID^*$: $\mathcal{R}$ recovers the matching pair (ID, x) from $\mathcal{L}_1$ and returns the previously computed $\frac{1}{s+x}G$. Note : No extraction query on ID^* can be made.
5. Signcryption query on $(M, ID_A, ID_1, ID_2, \ldots, ID_n)$: If $ID_A \neq ID^*$, $\mathcal{R}$ has the secret key corresponding to ID_A and can proceed normally as in the *Signcrypt* algorithm. Else $\mathcal{R}$ does the following
   - Picks $r, h \leftarrow \mathbb{Z}_p^*$ and a random string V of length equal to that of the message.
   - Computes $Z_A = r \cdot \frac{1}{\prod_{i=1}^{n}(x_i+s)}G$
   - Computes $y_i = \left(r \cdot \frac{x^*+s}{\prod_{j=1,j\neq i}^{n}(x_j+s)} - h \cdot (x_i + s)\right) H$ for $1 \leq i \leq n$
   - Computes $\alpha = e(Z_A, (x^* + s)H) \cdot g^{-h}$
   - Adds the tuples (h, M, α) in $\mathcal{L}_2$ and (V, α) in $\mathcal{L}_3$
   - Returns the ciphertext $\langle M \oplus V, Z_A, y_1, y_2, \ldots, y_n \rangle$.

   As one can see, the ciphertext pass off as a valid one since

$$\begin{aligned} e(Z_A, (x^* + s)H) \cdot g^{-h} &= g^{r \cdot (x^*+s)/e(Z_A,(x^*+s)H)-h} \\ &= e(y_i, S_i) \end{aligned}$$

6. Designcryption Query on (σ, ID_A, ID_i): $\mathcal{R}$ retrieves (c, Z_A, y_i) from σ and searches $\mathcal{L}_2$ for an entry of the form (m_j, α_j, h_j) thaqt satisfies the following condition

$$\alpha_j = e(Z_A, H_1(ID_A)Q + sQ) \cdot g^{-h_j}$$

   If such an entry is present, $\mathcal{R}$ returns m_j. Otherwise, he returns $\perp$.

   We note that if σ is a valid ciphertext, then h_j is the correct value of $H_2(m_j, \alpha_j)$, for some (m_j, α_j). If $\mathcal{A}$ has queried the H_2 oracle with these values, then an entry of the form (m_j, α_j, h_j) will be present in $\mathcal{L}_2$, which $\mathcal{R}$ retrieves. The only other case in which $\mathcal{A}$ can produce a valid ciphertext is by correctly guessing the hash value of (m_j, α_j) without querying it. In a perfect simulation, this ciphertext using the correct guessed value should pass of as a valid one. But in our simulation, this does not happen, and we return $\perp$. However we note that this event occurs only with a probability of $1/p$ which is of the order of $1/2^k$, which is negligible in the security parameter k.

We are ready to apply the forking lemma that essentially says the following: consider a scheme producing signatures of the form (M, α, h, Z_A), where each of α, h, Z_A corresponds to one of the three moves of a honest-verifier zero-knowledge protocol. In our setting, from a forger $\mathcal{A}$, we build an algorithm $\mathcal{A}'$ that replays $\mathcal{A}$ a sufficient number of times to obtain two suitable forgeries $(M^*, \alpha, h_1, Z_1), (M^*, \alpha, h_2, Z_2)$ on ID^*. The reduction then works as follows. The simulator $\mathcal{R}$ runs $\mathcal{A}$ to obtain two forgeries $(M^*, \alpha, h_1, Z_1), (M^*, \alpha, h_2, Z_2)$ for the same message M^* and commitment α. At this stage, $\mathcal{R}$ recovers the pair (ID^*, x^*) from list $\mathcal{L}_1$. If both forgeries satisfy the verification equation, we obtain the relations

$$e(Z_1, Q_{ID^*})e(G, H)^{-h_1} = e(Z_2, Q_{ID^*})e(G, H)^{-h_2}$$

with $Q_{ID^*} = H_1(ID^*)H + sH = (x^* + s)H$. Then, it comes that $e((h_1 - h_2)^{-1}(Z_1 - Z_2), Q_{ID^*}) = e(G, H)$ and hence $T^* = (h_1 - h_2)^{-1}(Z_1 - Z_2) = \frac{1}{w^*+s}G$ From T^*, $\mathcal{R}$ first obtains $a_{-1}, a_0, \ldots, a_{l-2}$ for which $\frac{f(z)}{(z+x^*)} = \frac{a_{-1}}{(z+x^*)} + \sum_{i=0}^{l-2} a_i z^i$ and eventually computes

$$\sigma^* = \frac{1}{a_{-1}} \left[T^* - \sum_{i=0}^{l-2} a_i s^i P \right] = \frac{1}{x^* + s} P$$

and $\frac{1}{x^*+s}Q = \beta^{-1} \cdot \sigma^*$ before returning the pair $(x^*, \frac{1}{x^*+s}Q)$ as a result.

We note as in [23], if $Adv_{\mathcal{A}}^{m\mathcal{IBSC}_1} \geq 10(q_{sc} + 1)(q_{sc} + q_{H_2})/2^k$, where l extraction queries, q_{H_i} queries to random oracles H_i (i= 1,2,3) and q_{sc} signcryption queries are made, then

$$Adv_{\mathcal{R}}^{l-SDHP} \geq 1/9$$

C Proof of Theorem 4

This proof is similar to that of Theorem 2, with the following modifications.

- Both the adversary and the challenger are given as input l the total number of extraction queries and q the total number of random oracle queries that can be issued by the adversary. Algorithm $\mathcal{R}$ is given as input a group system $\mathcal{B} = (p, \mathbb{G}_1, \mathbb{G}_2, e(,))$, and a $(U_2, V_2, F_2) - GDDHE$ instance in $\mathcal{B}$. We thus have $f, g_1, g_2, g_3, \ldots, g_n$ which are pairwise co prime polynomials with pairwise distinct roots, of orders q for f and 3 for g_i , and

$$\begin{pmatrix} P_0, sP_0, \ldots, s^{t-1}P_0 & s.f(s)P_0, s^2.f(s)P_0, s^3.f(s)P_0 & r.s.f(s)P_0 \\ Q_0, sQ_0, \ldots, s^5Q_0 & \gamma.s.g_1(s).Q_0, \gamma.s.g_2(s)Q_0, \ldots, \gamma.s.g_n(s)Q_0 & \end{pmatrix}$$

  and $T \in \mathbb{G}_2$, which is either is equal to $e(P_0, Q_0)^{\gamma \cdot f(s)}$ or to some random element of $\mathbb{G}_2$ Notations.
  - $f(X) = \prod_{i=1}^{l}(X + x_i)$
  - $g_i(X) = \prod_{j=l+3i-2}^{l+3i}(X + x_j)$ for $1 \leq i \leq n$
  - $f_i(x) = \frac{f(x)}{x+x_i}$ for $i \in [1, l]$, which is a polynomial of degree $l - 1$
- Init Phase: The adversary A outputs a t-set $\mathcal{S}^* = \{ID_1^*, ..., ID_t^*\}$ of identities that he wants to attack, where $t = 3n$.
- Setup Phase: To generate the system parameters, $\mathcal{R}$ formally sets $P = f(s)P_0$ (i.e. without computing it) and sets
  - $Q = Q_0$
  - $R = s.f(s)P_0 = sP$
  - $g = e(P_0, Q_0)^{f(s)} = e(P, Q)$

  $\mathcal{R}$ then defines the Public Key PK as $\langle Q, sQ, s^2Q, s^3Q, R, g\rangle$. Note $\mathcal{R}$ cannot compute the value of P.

- $\mathcal{R}$'s response to Signcryption Queries : Of the form $(m, ID_A, ID_1, ID_2, \ldots, ID_{3w})$ If $ID_A \notin \mathcal{S}^*$, $\mathcal{R}$ has the secret key of ID_A and can proceed as in normal *Signcrypt* algorithm. Otherwise, he does the following
  - Picks $r \in_R \mathbb{Z}_p^*$ and set $Z_A = r.sP$
  - Picks $h \in_R \mathbb{Z}_p^*$
  - Computes $y_i = r\left(s(x_A + s) - h\right)(x_i + s)(x_j + s)(x_k + s)Q$, where ID_j and ID_k are the identifiers grouped with ID_i.
  - Computes $X = \left(s(s + x_A)\right) sP$
  - Computes $\alpha = e(Z_A, (s + x_A)Q).g^{-h \cdot r}$ and picks a random string V of length same as the message
  - Returns $\langle m \oplus V, Z_A, X, y_1, y_2, \ldots, y_w\rangle$ and enters the tuples $(m, \alpha, h \cdot r)$ and (α, V) in $\mathcal{L}_2$ and $\mathcal{L}_3$ respectively.

  As one can see, the returned ciphertext will pass off as a valid one since

$$\begin{aligned} e(Z_A, (s + x_A)Q).g^{-h \cdot r} &= g^{r(s(s+x_A)-h)} \\ &= \left[e\left(S_i, y_l\right) \cdot e\left(X, \left(s + H_1\left(ID_j\right) H_1\left(ID_k\right)\right) Q\right)\right]^{\frac{1}{H_1\left(ID_j\right)H_1\left(ID_k\right)}} \end{aligned}$$

- Challenge Phase: When $\mathcal{A}$ decides that phase 1 is over, he gives two messages m_0 and m_1 and a sender's identity ID_A, algorithm $\mathcal{R}$ sets $\alpha = T$, picks random Z_A, c and responds with the challenge ciphertext
  $\sigma^* = \langle X, Z_A, y_1, y_2, \ldots, y_n\mathcal{L}\rangle$ where $X = \gamma.s.f(s)P_0$ and $y_i = \gamma.s.g_i(s)Q_0$ Note that if $T = g^\gamma$, then (X, y_i) is a valid encryption of $\alpha = g^\gamma$, although σ^* may not be valid ciphertext.

Guess Phase: Finally, the adversary A outputs a guess b

$\mathcal{R}$ ignores the answer and searches $\mathcal{L}_{H_3}$ for an entry of the form $(T, *)$. If present, he outputs 1 (indicating that $T = g^\gamma$). Otherwise, he outputs 0.

We note that if (X, y_i) is a valid encryption of T, then an adversary with a non-negligible advantage in the above game must have issued a H_3 query on T, in which case an entry of the form $(T, *)$ will be present in $\mathcal{L}_{H_3}$.

$$\begin{aligned} Adv_{\mathcal{R}}^{GDDHE}(U_2, V_2, F_2) &= Pr\left[b = b' | real\right] - Pr\left[b = b' | random\right] \\ &= \frac{1}{2} \cdot Adv_{\mathcal{A}}^{mIBSC-CCA} \end{aligned}$$

D Proof of Theorem 5

This proof is similar to that of Theorem 3, with the following modifications

- Let l be the maximum number of extraction queries that can be queried by the adversary $\mathcal{A}$. Algorithm $\mathcal{R}$ takes as input $(Q, sQ, s^2Q, \ldots, s^{l+3}Q)$ and aims to find a pair $(c, \frac{1}{c+s}Q)$. In a setup phase, it builds a generator $G \in \mathbb{G}_1$, as in the proof of Theorem 3, such that it knows $l - 1$ pairs $(x_i, \frac{1}{x_i+s}G)$ for $x_1, \ldots, x_{l-1} \in_R \mathbb{Z}_p^*$, computes sG, sH, s^2H, s^3H and s^4H and makes $\langle sG, H, sH, s^2H, s^3H, g = e(G, H)\rangle$ public.
- Response to a Signcryption query on $(M, ID_A, ID_1, ID_2, \ldots, ID_n)$: If $ID_A \neq ID^*$, procced normally as in the *Signcrypt* algorithm. Else $\mathcal{R}$ does the following
  - Picks $r, h \leftarrow \mathbb{Z}_p^*$ and a random string V of length same as the message
  - Computes $Z_A = rG$
  - Computes $y_i = r(x^* + s - h)(x_i + s)(x_j + s)(x_k + s)H$ for $1 \leq i \leq \frac{n}{3}$
  - Computes $X = rs(x^* + s - h)G$
  - Computes $\alpha = e(Z_A, (x^* + s)H)g^{-h \cdot r}$
  - Adds the tuple $(h \cdot r, M, \alpha)$ in $\mathcal{L}_2$ and (V, α) in $\mathcal{L}_3$
  - Returns the ciphertext $\langle M \oplus V, Z_A, y_1, y_2, \ldots, y_{\frac{n}{3}}\rangle$.

One can see that the above ciphertext will pass off as a valid ciphertext since

$$\begin{aligned} e(Z_A, (x^* + s)H)g^{-h\cdot r} &= g^{r\cdot(x^*+s-h)} \\ &= \left[e\left(S_i, y_l\right) \cdot e\left(X, \left(s + H_1\left(ID_j\right) H_1\left(ID_k\right)\right) H\right)\right]^{\frac{1}{H_1\left(ID_j\right)H_1\left(ID_k\right)}} \end{aligned}$$

As in the proof of Theorem 3, if $Adv_{\mathcal{A}}^{m\mathcal{I}BSC_2} \geq 10(q_{sc} + 1)(q_{sc} + q_{H_2})/2^k$, where l extraction queries, q_{H_i} queries to random oracles H_i (i= 1,2,3) and q_{sc} signcryption queries are made, then

$$Adv_{\mathcal{R}}^{(l+3)-SDHP} \geq 1/9$$

E Proof of Theorem 6

This proof is similar to that of Theorem 2, with the following modifications

- Both the adversary and the challenger are given as input N, the maximal size of a set of included users $\mathcal{S}$, and l the total number of extraction queries and q the total number of random oracle queries that can be issued by the adversary. Algorithm $\mathcal{R}$ is given as input a group system $\mathcal{B} = (p, \mathbb{G}_1, \mathbb{G}_2, e(,))$, and a $(U_3, V_3, F_3) - GDDHE$ instance in $\mathcal{B}$. We thus have f and g, two co prime polynomials with pairwise distinct roots, of respective orders l and t respectively , and

  $$\begin{pmatrix} P_0, sP_0, \ldots, s^{l-1}P_0 & s.f(s)P_0, s^2.f(s)P_0, s^3.f(s)P_0 & \gamma.s.f(s)P_0 \\ Q_0, sQ_0, \ldots, s^{N+2}Q_0 & & \gamma.s.g(s)Q_0 \end{pmatrix}$$

  and $T \in \mathbb{G}_2$, which is either is equal to $e(P_0, Q_0)^{\gamma\cdot f(s)}$ or to some random element of $\mathbb{G}_2$ Notations.
  - $f(X) = \prod_{i=1}^{l}(X + x_i)$
  - $g(X) = \prod_{i=l+1}^{l+t}(X + x_i)$
  - $f_i(x) = \frac{f(x)}{x+x_i}$ for $i \in [1, l]$, which is a polynomial of degree $l - 1$
- Init Phase: The adversary A outputs a t-set $\mathcal{S}^* = \{ID_1^*, ..., ID_t^*\}$ of identities that he wants to attack.
- Setup Phase: To generate the system parameters, $\mathcal{R}$ formally sets $P = f(s)P_0$ (i.e. without computing it) and sets
  - $Q = Q_0$
  - $R = s.f(s)P_0 = sP$
  - $g = e(P_0, Q_0)^{f(s)} = e(P, Q)$

  $\mathcal{R}$ then defines the Public Key PK as $Q, sQ, s^2Q, \ldots, s^NQ, R, g$. Note $\mathcal{R}$ cannot compute the value of P.
- $\mathcal{R}$'s response to Signcryption Queries : Of the form $(m, ID_A, ID_1, ID_2, \ldots, ID_n)$ If $ID_A \notin \mathcal{S}^*$, $\mathcal{R}$ proceeds as in normal *Signcrypt* algorithm. Otherwise, he does the following
  - Pick $r \in_R \mathbb{Z}_p^*$ and set $Z_A = r.sP$
  - Pick $h \in_R \mathbb{Z}_p^*$
  - Compute $y = r\left(s(x_A + s) - h\right)\prod_{i=1}^{n}(s + x_i)Q$ and $X = r.\left(s(s + x_A) - h\right)sP$
  - Computes $\alpha = e(Z_A, (s + x_A)Q).g^{-h\cdot r}$ and picks a random string V of length same as the message
  - Returns $\langle m \oplus V, Z_A, X, y\rangle$ and enters the tuples $(m, \alpha, h \cdot r)$ and (α, V) in $\mathcal{L}_2$ and $\mathcal{L}_3$ respectively.

  As one can see, the returned ciphertext will pass off as a valid one since

  $$\begin{aligned} e(Z_A, (s + x_A)Q).g^{-h\cdot r} &= g^{r\cdot(s(s+x_A)-h)} \\ &= e\left(S_i, y\right).e\left(X, \frac{1}{s}\left[\prod_{j=1, j\neq i}^{n}\left(s + H_1\left(ID_j\right)\right) - \prod_{j=1, j\neq i}^{n} H_1\left(ID_j\right)\right] Q\right) \end{aligned}$$

- Challenge Phase: When $\mathcal{A}$ decides that phase 1 is over, he gives two messages m_0 and m_1 and a sender's identity ID_A, algorithm $\mathcal{R}$ sets $\alpha = T$, picks random Z_A, c and responds with the challenge ciphertext $\sigma^* = \langle X, Z_A, y, \mathcal{L} \rangle$ where $X = r.s.f(s)P_0$, $y = \gamma.s.g(s)Q_0$. Note that if $T = g^\gamma$, then (X, y) is a valid encryption of $\alpha = g^\gamma$, although σ^* may not be a valid ciphertext.

Guess Phase: Finally, the adversary A outputs a guess b

$\mathcal{R}$ ignores the answer and searches $\mathcal{L}_{H_3}$ for an entry of the form $(T, *)$. If present, he outputs 1 (indicating that $T = g^\gamma$). Otherwise, he outputs 0.

We note that if (X, y) is a valid encryption of T, then an adversary with a non-negligible advantage in the above game must have issued a H_3 query on T, in which case an entry of the form $(T, *)$ will be present in $\mathcal{L}_{H_3}$.

$$\begin{aligned} Adv_{\mathcal{R}}^{GDDHE}(U_3, V_3, F_3) &= Pr\left[b = b' | real\right] - Pr\left[b = b' | random\right] \\ &= \frac{1}{2} \cdot Adv_{\mathcal{A}}^{mIBSC-CCA} \end{aligned}$$

F Proof of Theorem 7

This proof is similar to that of Theorem 3, with the following modifications.

- Let l be the maximum number of extraction queries that can be queried by the adversary $\mathcal{A}$ and N be the maximal size of the receiver set. Algorithm $\mathcal{R}$ takes as input $(Q, sQ, s^2Q, \ldots, s^{l+N}Q)$ and aims to find a pair $(c, \frac{1}{c+s}Q)$. In a setup phase, it builds a generator $G \in \mathbb{G}_1$ as in the proof of Theorem 3, such that it knows $l-1$ pairs $(x_i, \frac{1}{x_i+s}G)$ for $x_1, \ldots, x_{l-1} \in_R \mathbb{Z}_p^*$. He makes $\langle sG, H, sH, s^2H, s^3H, \ldots, s^NH, g = e(G, H) \rangle$ public.
- Response to a Signcryption query on $(M, ID_A, ID_1, ID_2, \ldots, ID_n)$: If $ID_A \neq ID^*$, proceed normally as in the *Signcrypt* algorithm. Else $\mathcal{R}$ does the following
  - Picks $r, h \leftarrow \mathbb{Z}_p^*$ and a random string V of length equal to that of the message.
  - Computes $Z_A = r.G$
  - Computes $y = r(x^* + s - h) \prod_{i=1}^{n} (x_i + s)H$
  - Computes $X = rs(x^* + s - h)G$
  - Computes $\alpha = e(Z_A, (x^* + s)H)g^{-h \cdot r}$
  - Adds the tuple $(h \cdot r, M, \alpha)$ in $\mathcal{L}_2$ and (V, α) in $\mathcal{L}_3$
  - Returns the ciphertext $\langle M \oplus V, Z_A, y \rangle$.

  As one can see, the returned ciphertext will pass off as a valid one as

$$\begin{aligned} e(Z_A, (s + x_A)H).g^{-h \cdot r} &= g^{r \cdot (s + x^* - h)} \\ &= e\left(S_i, y\right).e\left(X, \frac{1}{s}\left[\prod_{j=1, j \neq i}^{n} \left(s + H_1\left(ID_j\right)\right) - \prod_{j=1, j \neq i}^{n} H_1\left(ID_j\right)\right] H\right) \end{aligned}$$

As in the proof of Theorem 3, if $Adv_{\mathcal{A}}^{m\mathcal{I}BSC_3} \geq 10(q_{sc} + 1)(q_{sc} + q_{H_2})/2^k$, where l extraction queries, q_{H_i} queries to random oracles H_i (i= 1,2,3) and q_{sc} signcryption queries are made, then

$$Adv_{\mathcal{R}}^{(l+N)-SDHP} \geq 1/9$$

G Intractability of $(U_i, V_i, F_i) - GDDHE$

In this section, we prove the intractability of distinguishing the two distributions involved in the $(U_i, V_i, F_i) - GDDHE$ problems in the proofs of Theorems 2, 4 and 6.

In order to prove Corollaries 1, 2 and 3, we need to prove the intractability of $(U_i, V_i, F_i) - GDDHE$ problem for $i = 1, 2, 3$ and then subsequently use the result of Theorem 1. We consider the case when $\mathbb{G}_1 = \mathbb{G}_2 = \mathbb{G}$ and thus pose $Q_0 = \beta P_0$ Our problem can be reformulated as $(U, V, F) - GDHE$ where

$$U = \begin{pmatrix} 1, s, s^2, \ldots, s^{l-1}, & s.f(s), s^2.f(s), s^3.f(s), \gamma.s.f(s) \\ \beta, s.\beta, s^2.\beta. \ldots, s^{N+2}.\beta, & \gamma.\beta.g_1(s), \gamma.\beta.g_2(s), \gamma.\beta.g_3(s), \ldots, \gamma.\beta.g_k(s) \end{pmatrix}$$

$$V = 1$$

$$F = \gamma.\beta.f(s)$$

We have $k = 1, 2$ or 3 and $deg(g_i) = 1, 3$ or t for Corollaries 1,2 and 3 respectively. Degree of f is l. We have to show that F is independent of (U, V), i.e. that no coefficients $\{a_{i,j}\}_{i,j=1}^{n}$ and b_1 exist such that $F = \sum_{i,j=1}^{n} a_{i,j} u_i u_j + b_1 v_1$ where the polynomials u_i and v_1 are the one listed in U and V above. By making all possible products of two polynomials from U which are multiples of $\gamma.\beta$, we want to prove that no linear combination among the polynomials from the list R below leads to F:

$$R = \begin{pmatrix} \gamma.\beta.s.f(s), \gamma.\beta.s^2.f(s), \gamma.\beta.s^3.f(s), \ldots, \gamma.\beta.s^{N+3}.f(s), \\ \gamma.\beta.g_1(s), \gamma.\beta.s.g_1(s), \ldots, \gamma.\beta.s^{l-1}.g_1(s) \\ \gamma.\beta.g_2(s), \gamma.\beta.s.g_2(s), \ldots, \gamma.\beta.s^{l-1}.g_2(s) \\ \ldots \\ \ldots \\ \ldots \\ \gamma.\beta.g_k(s), \gamma.\beta.s.g_k(s), \ldots, \gamma.\beta.s^{l-1}.g_k(s) \\ \gamma.\beta.s.f(s).g_1(s), \gamma.\beta.s.f(s).g_2(s), \ldots, \gamma.\beta.s.f(s).g_k(s) \\ \gamma.\beta.s^2.f(s).g_1(s), \gamma.\beta.s^2.f(s).g_2(s), \ldots, \gamma.\beta.s^2.f(s).g_k(s) \\ \gamma.\beta.s^3.f(s).g_1(s), \gamma.\beta.s^3.f(s).g_2(s), \ldots, \gamma.\beta.s^3.f(s).g_k(s) \end{pmatrix}$$

Note that the every polynomial on the last three lines can be written as

$$\gamma.\beta.s^j.f(s).g_i(s) = \sum_{i=0}^{i=deg(g_i)} c_i.\gamma.\beta.s^{i+j} f(s)$$

for $j = 1, 2, 3$ and thus as a linear combination of the polynomials from the first line. We therefore simplify the task, by finding a linear combination of the elements of the list R' below, which leads to $f(s)$

$$R' = \begin{pmatrix} s.f(s), s^2.f(s), \ldots, s^{N+3}.f(s), \\ g_1(s), s.g_1(s), \ldots, s^{l-1}g_1(s) \\ g_2(s), s.g_2(s), \ldots, s^{l-1}g_2(s) \\ \ldots \\ \ldots \\ \ldots \\ g_k(s), s.g_k(s), \ldots, s^{l-1}g_k(s) \end{pmatrix}$$

Any linear combination can be written as

$$f(s) = A(s).f(s) + B_1(s)g_1(s) + B_2(s)g_2(s) + \ldots + B_k(s)g_k(s)$$

where A and B are polynomials such that $A(0) = 0$, $deg(A) \leq N + 3$ and $deg(B) \leq l - 1$. Since f and g_i are co prime by assumption, we must have f/B_i. Since $deg(f) = l$ and $deg(B_i) \leq l - 1$ this implies $B_i = 0$ for $1 \leq i \leq k$. Hence $A = 1$ which contradicts $A(0) = 0$. Therefore

$$F_i \notin \langle U_i, V_i \rangle \text{ for i = 1,2,3}$$