Identity-Based Higncryption

May 16, 2019

Abstract

After two decades of research on signcryption, recently a new cryptographic primitive, named higncryption, was proposed at ACM CCS'16. Higncryption can be viewed as privacy-enhanced signcryption, which integrates public key encryption, digital signature and identity concealment (which is not achieved in signcryption) into a monolithic primitive. Here, briefly speaking, identity concealment means that the transcript of protocol runs should not leak participants' identity information.

In this work, we propose the first identity-based higncryption (IBHigncryption), with motivational applications to 5G communications. We present the formal security model of IBHigncryption, and the detailed security proofs for the proposed scheme. The most impressive feature of IBHigncryption, among others (including the desirable properties it offers, such as forward ID-privacy, receiver deniability, and x-security), is its simplicity and efficiency, which might be somewhat surprising in retrospect. The proposed IBHigncryption scheme is essentially as efficient as the fundamental CCA-secure Boneh-Franklin IBE scheme [16], while offering entity authentication and identity concealment simultaneously. Compared to the identity-based signcryption scheme [9], which is adopted in the IEEE P1363.3 standard, our IBHigncryption scheme is much simpler, and has significant efficiency advantage in total (particularly on the receiver side). Besides, our IBHigncryption enjoys forward ID-privacy, receiver deniability and x-security simultaneously, while the IEEE 1363.3 standard of ID-based signcryption satisfies none of them. In addition, our IBHigncryption has a much simpler setup stage with smaller public parameters, which in particular does not have the traditional master public key.

1 Introduction

Identity-based cryptography (ID-based) was proposed by Shamir in 1984 [48], with the motivation to simplify certificate management in traditional public-key cryptography. In an ID-based cryptosystem, the identity of a user acts as its public key, so the certificate issuance and management problem

is simplified in an ID-based system. In general, ID-based cryptography includes identity-based signature (IBS), identity-based encryption (IBE), etc. Though ID-based signature schemes appeared much earlier [48, 26, 25]. However, the first practical and fully functional identity-based encryption scheme was only proposed by Boneh and Franklin [16] in 2001 based on bilinear maps. The Boneh-Franklin's IBE scheme is further standardized with ISO/IEC 18033-5 and IETF RFC 5091 [18], and is now widely deployed (e.g., in HPE Secure Data by Voltage security [4]).[1]

The concept of signcryption was proposed by Zheng [51]. It enables the sender to send an encrypted message such that only the intended receiver can decrypt it, and meanwhile, the intended receiver has the ability to authenticate that the message is indeed from the specified sender. It provides a more economical and safer way to integrate encryption and signature, compared to the sequential composition of them. Since its introduction, research and development (including international standardizations) of signcryption have been vigorous. For example, a list of public-key signcryption schemes was standardized with ISO 29150, and a pairing-based ID-based signcryption scheme [9] was adopted as IEEE P1363.3 standard.

With signcryption, the sender's identity information has to be exposed; otherwise, the ciphertext cannot be decrypted and the authentication cannot be verified. However, identity is a fundamental privacy concern. Identity confidentiality is now mandated by a list of prominent standards such as TLS1.3 [44], EMV [19], QUIC [46], and the 5G telecommunication standard [2] by 3GPP, etc, and is enforced by General Data Protection Regulation (GDPR) of EU. Under this motivation, a new cryptographic primitive called identity-hiding signcryption (higncryption, for short) was introduced in [50]. Higncryption can be viewed as a novel monolithic integration of public key encryption, digital signature, and identity concealment. Here, identity concealment means that the transcript of protocol runs should not leak participants' identity information. Moreover, a higncryption scheme satisfies the following features simultaneously:

- Forward ID-privacy, which means that player's ID-privacy preserves even when its static secret key is compromised.
- Receiver deniability [32], in the sense that the session transcript can be simulated from the public parameters and the receiver's secret-key.
- x-security [32], in the sense that the leakage of some critical intermediate randomness (specifically, DH-exponent x) does not cause the exposure of the sender's static secret key or the pre-shared secret (from which session-key is derived).

[1] The HPE IBE (including BF01 [16] and BB1 [15]) technology developed by Voltage provides plug-ins for Outlook, Pine, Hotmail, Yahoo, etc, and is reported to be used by over 200 million users and more than 1,000 enterprises worldwide.

We note that the work in [50] only considered higncryption in the traditional public-key setting. In this work, we study identity-based highcryption and its applications.

1.1 Motivational Application for 5G

5G is the fifth generation of cellular mobile communication, which succeeds the 4G (LTE/WiMax), 3G (UMTS) and 2G (GSM) systems. 5G performance targets include high data rate, reduced latency, and massive device connectivity (for low-power sensors and smart devices), which are far beyond the levels 4G technologies can achieve. Among the services 5G supported, mission critical services and communications require ultra reliability and virtual zero latency. The platform for mission critical (MC) communications and MC services has been a key priority of 3GPP in recent years, and is expected to evolve further in the future [37]. In June 2018, 3GPP has identified the following essential requirements related to user privacy [1, 35] for 5G communications.

- User identity confidentiality: The permanent identity of a user to whom a service is delivered cannot be eavesdropped on the radio access link.
- User untraceability: An intruder cannot deduce whether different services are delivered to the same user by eavesdropping on the radio access link.
- User location confidentiality: The presence or the arrival of a user in a certain area cannot be determined by eavesdropping on the radio access link.

At the heart of the security architecture, specified by 3GPP [2] for 5G mission critical communications and services, is an identity-based authenticated key transport (IB-AKT) protocol inherited from 4G, which is the identity-based version of Multimedia Internet KEYing (MIKEY) specified in IETF RFC 3830 [5]. This IB-AKT protocol involves the *sequential* composition of an identity-based encryption scheme (specifically, SAKKE specified in IETF RFC 6508 [31] and 6509 [30]) and an identity-based signature scheme (specifically, ECCSI specified in IETF RFC 6507 [29]). In MIKEY-SAKKE, the user's identity ID takes the form of a constrained telephone URI (universal resource identifier), in front of which there is a monthly-updated time stamp for periodically refreshing the key of the user. It also provides a simple mechanism for masking identity; Briefly speaking, for MIKEY-SAKKE with identity masking [3], a user's URI is replaced by $UID = H(S)$, where H is the SHA-256 hash function and S is some information related to the identifiers of the user and the key management server (KMS). Further, UID

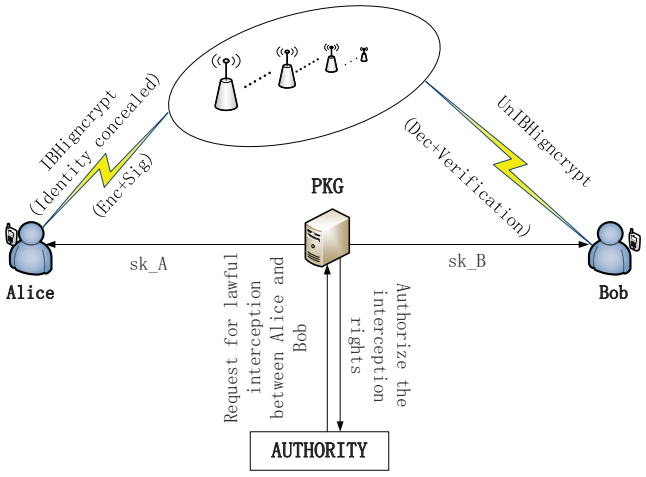

Figure 1: IBHigncryption's Application in 4G-LTE

shall be used as the identifier within MIKEY-SAKKE with identity masking. Clearly, MIKEY-SAKKE does not satisfy the above requirements on identity privacy mandated by 5G now.

Considering that the *sequential* composition of an identity-based encryption scheme and an identity-based signature scheme is less efficient, identity-based signcryption may be a promising candidate for mission critical services. We note that there already exists IEEE P1363.3 standard for ID-based signcryption [9]. However, as mentioned ahead, the sender's identity has to be exposed [9]. In this sense, ID-based identity-concealed signcryption (IBHigncryption) takes place. Moreover, for enhancing privacy and strengthening security, forward ID-privacy, receiver deniability, and x-security are all desirable in such settings.

Figure 1 illustrates the application of IBHigncrypt in MIKEY-based mission critical communications. If Alice (the session initiator) wants to make a private call to Bob (the session receiver), she IBHigncrypts her request and her identity using her private key generated by the public key generator (PKG) on her public identity, and then sends it to Bob via internet or wireless channel. On receiving Alice's request, Bob UnIBHigncrypts the ciphertext, and gets Alice's request and her identity information. By verifying the message decrypted (which is equivalent to the verification of Alice's signature), Bob can determine whether the request is indeed from Alice. Based on the verification, Bob can choose whether he accepts the session or not. Meanwhile, if there is an authority who needs to intercept the communications between Alice and Bob, it contacts PKG to request the private key of Bob, with which the authority can inspect the communications lawfully.

1.2 Contribution

In this work, we propose the first identity-based higncryption (IBHigncryption, for short). We present the formal security model of IBHigncryption, and the detailed security proofs for the proposed scheme. The most impressive feature of IBHigncryption, among others (including the desirable properties it offers, such as forward ID-privacy, receiver deniability, and x-security), is its simplicity and efficiency, which might be somewhat surprising in retrospect. The proposed IBHigncryption scheme is essentially as efficient as the fundamental CCA-secure Boneh-Franklin IBE scheme [16], while offering entity authentication and identity concealment simultaneously. Compared to the identity-based signcryption scheme [9], which is adopted in the IEEE P1363.3 standard, our IBHigncryption scheme is much simpler, and has significant efficiency advantage in total (particularly on the receiver side). Besides, our IBHigncryption enjoys forward ID-privacy, receiver deniability and x-security simultaneously, while the IEEE 1363.3 standard of ID-based signcryption satisfies none of them.

In addition, our IBHigncryption has a much simpler setup stage with smaller public parameters, which in particular *does not need to generate the traditional master public key*. The much simpler setup stage of IBHigncryption, particularly waiving the master public key, brings the following advantages:

- The computational and space complexity for generating and storing the system parameters is reduced.
- The attack vector (for recovering the master secret key) is decreased, e.g., for some mission critical applications.
- It eases deployment and compatibility with existing ID-based cryptosystems. Specifically, when deploying our IBHigncryption scheme in reality with other existing identity-based cryptosystems, the system parameters and particularly the master public key can remain unchanged.

We implement the IBHigncryption scheme for pairings of Type 1 and 3, where the codes are (anonymously) available from https://github.com/IBHigncryption2018/IBHigncryption. The implementations use the PBC (pairing-based cryptography) library of Stanford University http://crypto.stanford.edu/pbc, and the underlying authenticated encryption is implemented with AES-GCM-256.

2 Preliminaries

If S is a finite set, $|S|$ is its cardinality, and $x \leftarrow S$ is the operation of picking an element uniformly at random from S. If S denotes a probability

distribution, $x \leftarrow S$ is the operation of picking an element according to S. We overload the notion for probabilistic or stateful algorithms, where $V \leftarrow \mathsf{Alg}$ means that algorithm Alg runs and outputs value V. A string or value α means a binary number, and $|\alpha|$ denotes its length. Let $a := b$ denote a simple assignment statement, which means assigning b to a, and $x \| y$ be the concatenation of two elements $x, y \in \{0,1\}^*$.

2.1 Authenticated Encryption

Briefly speaking, an *authenticated encryption* (AE) scheme transforms a message M and a public header information H (e.g., a packet header, an IP address, some predetermined nonce or initial vector) into a ciphertext C in such a way that C provides both privacy (of M) and authenticity (of C and H) [11, 12, 45, 36]. In practice, when AE is used within cryptographic systems, the associated data H is usually implicitly determined from the context (e.g., the hash of the transcript of the protocol run or some predetermined states).

Let $\mathsf{SE} = (\mathsf{K}_{se}, \mathsf{Enc}, \mathsf{Dec})$ be a symmetric encryption scheme. The probabilistic polynomial-time (PPT) algorithm K_{se} takes the security parameter κ as input and samples a key K from a finite and non-empty set $\mathcal{K} \bigcap \{0,1\}^\kappa$. For presentation simplicity, we assume $K \leftarrow \mathcal{K} = \{0,1\}^\kappa$. The polynomial-time (randomized or stateful)[2] encryption algorithm $\mathsf{Enc} : \mathcal{K} \times \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^* \cup \{\perp\}$, and the (deterministic) polynomial-time decryption algorithm $\mathsf{Dec} : \mathcal{K} \times \{0,1\}^* \rightarrow \{0,1\}^* \cup \{\perp\}$ satisfy: for any $K \leftarrow \mathcal{K}$, any associated data $H \in \{0,1\}^*$ and any message $M \in \{0,1\}^*$, if $\mathsf{Enc}_K(H, M)$ outputs $C \neq \perp$, $\mathsf{Dec}_K(C)$ always outputs M. *Here, for presentation simplicity, we assume that the ciphertext C bears the associated data H in plain.*

Let $\mathcal{A}$ be an adversary. Table 1 describes the security game for authenticated encryption. We define the advantage of $\mathcal{A}$ to be

$$\mathbf{Adv}^{\mathsf{AE}}_{\mathsf{SE}}(\mathcal{A}) = \left| 2 \cdot \Pr[\mathrm{AE}^{\mathcal{A}}_{\mathsf{SE}} \ \ \textit{returns} \ \ \mathsf{true}] - 1 \right|.$$

We say that the SE scheme is AE-secure, if for any sufficiently large κ, the advantage of any probabilistic polynomial-time (PPT) algorithm adversary is negligible. We say the SE scheme is (t_{AE}, ϵ_{AE})-secure, if for any sufficiently large κ and any PPT adversary $\mathcal{A}$ of running time t, $\mathbf{Adv}^{\mathsf{AE}}_{\mathsf{SE}}(\mathcal{A}) < \epsilon_{AE}$.

The above AE definition is based on that given in [11, 12], but with the public header data H explicitly taken into account. The definition of *authenticated encryption with associated data* (AEAD) given in [36] is stronger than ours in that: (1) it is length-hiding; and (2) both the encryption and the decryption algorithms are stateful.

[2] If randomized, it flips coins anew on each invocation. If stateful, it uses and then updates a state that is maintained across invocations.

main $\mathrm{AE}^{\mathcal{A}}_{\mathsf{SE}}$:	**proc.** $\mathsf{Enc}(H, M_0, M_1)$:	**proc.** $\mathsf{Dec}(C')$:
$K \leftarrow \mathcal{K}_{\mathrm{se}}$	If $\lvert M_0\rvert \neq \lvert M_1\rvert$, Ret $\perp$	If $\sigma = 1 \wedge C' \notin \mathcal{C}$
$\sigma \leftarrow \{0,1\}$	$C_0 \leftarrow \mathsf{Enc}_K(H, M_0)$	Ret $\mathsf{Dec}_K(C')$
$\sigma' = \mathcal{A}^{\mathbf{Enc},\mathbf{Dec}}$	$C_1 \leftarrow \mathsf{Enc}_K(H, M_1)$	Ret $\perp$
Ret $(\sigma' = \sigma)$	If $C_0 = \perp$ or $C_1 = \perp$	
	Ret $\perp$	
	$\mathcal{C} \xleftarrow{\cup} C_\sigma$; Ret C_σ	

Table 1: AE security game

The above AE security is quite strong. In particular, it means that, after adaptively seeing a polynomial number of ciphertexts, an efficient adversary is unable to generate a new valid ciphertext in the sense that its decryption is not "$\perp$". Also, for two independent keys $K, K' \leftarrow \mathcal{K}$ and any message M and any header information H, $\Pr[\mathsf{Dec}_{K'}(\mathsf{Enc}_K(H, M)) \neq \perp]$ is negligible.

3 Bilinear Pairings, and Hard Problems

Definition 1 (Bilinear Paring [47, 16]) *Let $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_\mathsf{T}$ be three multiplicative groups of the same prime order q, and let g_1, g_2 be generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. Assume that the discrete logarithm problems in $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_\mathsf{T}$ are intractable. We say that $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_\mathsf{T}$ is an admissible bilinear pairing, if it satisfies the following properties:*

1. *Bilinear: For all $a,\ b \leftarrow \mathbb{Z}_q^*, \hat{g}_1 \leftarrow \mathbb{G}_1, \hat{g}_2 \leftarrow \mathbb{G}_2$, $e(\hat{g_1}^a, \hat{g_2}^b) = e(\hat{g}_1, \hat{g}_2)^{ab}$.*
2. *Non-degenerate: For each $\hat{g}_1 \in \mathbb{G}_1/\{1\}$, there exists $\hat{g}_2 \in \mathbb{G}_2$, such that $e(\hat{g}_1, \hat{g}_2) \neq 1$.*
3. *Computable: For all $\hat{g}_1 \leftarrow \mathbb{G}_1, \hat{g}_2 \leftarrow \mathbb{G}_2$, $e(\hat{g}_1, \hat{g}_2)$ is efficiently computable.*

Bilinear pairings are powerful mathematical tools for numerous cryptographic applications (e.g., [16, 17, 15, 9, 41, 22, 34, 10, 24, 33, 38, 14]). Generally, there are three types of bilinear pairing [28, 49, 20, 21, 43]:

Type 1: $\mathbb{G}_1 = \mathbb{G}_2$, it is also called symmetric bilinear pairing.

Type 2: There is an efficiently computable isomorphism either from $\mathbb{G}_1$ to $\mathbb{G}_2$ or from $\mathbb{G}_2$ to $\mathbb{G}_1$.

Type 3: There exists no efficiently computable isomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$.

A brief history of pairings is presented in [7]. In recent years, much progress on number field sieve (NFS) has been made against pairing-friendly curves, which imposes new estimation of the security of parings. The reader is referred to [8] for updated key size estimation of some popular pairing-friendly curves (e.g., BN, BLS, KSS).

The computationally *intractable* problems considered in this work are defined as follows, which are described w.r.t. Type 1 pairings for presentation simplicity. Let $\mathbb{G}_1, \mathbb{G}_\mathsf{T}$ be two multiplicative groups of the same prime order q, g be a generator of $\mathbb{G}_1$, $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_\mathsf{T}$ be an admissible symmetric bilinear pairing.

Definition 2 (Bilinear Diffie-Hellman (BDH)) *The bilinear Diffie-Hellman (BDH) problem [39] in $\langle \mathbb{G}_1, \mathbb{G}_\mathsf{T}, e \rangle$ is to compute $e(g,g)^{abc} \in \mathbb{G}_\mathsf{T}$, given $(g, g^a, g^b, g^c) \in \mathbb{G}_1^4$, where $a, b, c \leftarrow \mathbb{Z}_q^*$. The BDH assumption says that no PPT algorithm can solve the BDH problem with non-negligible probability.*

Definition 3 (Square Bilinear Diffie-Hellman (SBDH)) *The square bilinear Diffie-Hellman (SBDH) problem in $\langle \mathbb{G}_1, \mathbb{G}_\mathsf{T}, e \rangle$ is to compute $e(g,g)^{a^2 b} \in \mathbb{G}_\mathsf{T}$, given $(g, g^a, g^b) \in \mathbb{G}_1^3$, where $a, b \leftarrow \mathbb{Z}_q^*$. The SBDH assumption says that no PPT algorithm can solve the SBDH problem with non-negligible probability.*

Below, we show that the SBDH assumption is equivalent to the BDH assumption. To the best of our knowledge, the equivalence between the two problems is first proved in this work, which might be of independent interest.

Proposition 1 *Let $x, y, z \leftarrow \mathbb{Z}_q^*$. Then the statistical distance between $x + y \pmod q$ and z is just $\frac{1}{q-1}$.*

Proof 1 *For presentation simplicity, we omit the modular arithmetic. Firstly, we consider the distribution of $x + y$. There are two cases to consider. For any $\alpha \in Z_q$, (1) if $\alpha = 0$, then $\Pr[x + y = 0 | x, y \leftarrow \mathbb{Z}_q^*] = \frac{1}{q-1}$; (2) if $\alpha \neq 0$, then $\Pr[x + y = \alpha | x, y \leftarrow \mathbb{Z}_q^*] = (1 - \frac{1}{q-1}) \cdot \frac{1}{q-1} = \frac{q-2}{(q-1)^2}$. Therefore the statistical distance between $x + y \pmod q$ and z is:*

$$
\begin{aligned}
\Delta(x+y,z) &= \frac{1}{2}\sum_{\alpha}|\Pr[x+y=\alpha]-\Pr[z=\alpha]| \\
&= \frac{1}{2}|\Pr[x+y=0]-\Pr[z=0]| \\
&+ \frac{1}{2}\sum_{\alpha=1}^{q-1}|\Pr[x+y=\alpha]-\Pr[z=\alpha]| \\
&= \frac{1}{2}\cdot\frac{1}{q-1}+\frac{1}{2}\cdot\sum_{\alpha=1}^{q-1}\left|\frac{q-2}{(q-1)^2}-\frac{1}{q-1}\right| \\
&= \frac{1}{q-1}
\end{aligned}
$$

Theorem 1 *The* BDH *assumption and the* SBDH *assumption are equivalent.*

Proof 2 BDH $\Longrightarrow$ SBDH*:*

Suppose that there is an oracle $\mathcal{O}_1$, which, on input $(g, g^a, g^b, g^c) \in \mathbb{G}_1^4$, outputs $e(g,g)^{abc} \in \mathbb{G}_\mathsf{T}$ with non-negligible probability. Then, there must exist a PPT algorithm $\mathcal{A}_1$, which, on input $(g, g^a, g^b) \in \mathbb{G}_1^3$, outputs $e(g,g)^{a^2b} \in \mathbb{G}_\mathsf{T}$ with the same probability. The algorithm $\mathcal{A}_1$ chooses $t_1, t_2 \leftarrow \mathbb{Z}_q^*$, and computes $u_1 = (g^a)^{t_1} = g^{at_1}$, $u_2 = (g^a)^{t_2} = g^{at_2}$. Therefore, $\mathcal{A}_1$ is able to compute $v = \mathcal{O}_1(g, u_1, u_2, g^b) = e(g,g)^{a^2bt_1t_2}$. It follows that $e(g,g)^{a^2b}$ can be computed from v, t_1, t_2 immediately with the same advantage.

SBDH $\Longrightarrow$ BDH*:*

Suppose that there is an oracle $\mathcal{O}_2$, which, on input $(g,\ g^a,\ g^b) \in \mathbb{G}_1^3$, outputs $e(g,g)^{a^2b} \in \mathbb{G}_\mathsf{T}$ with non-negligible probability ϵ, where $a, b, c \leftarrow Z_q^*$. Then, we show there exists a PPT algorithm $\mathcal{A}_2$, which, on input $(g, g^a, g^b, g^c) \in \mathbb{G}_1^4$, outputs $e(g,g)^{abc} \in \mathbb{G}_\mathsf{T}$ also with non-negligible probability. The algorithm $\mathcal{A}_2$ chooses $r, s, t \leftarrow \mathbb{Z}_q^*$, and by querying the oracle $\mathcal{O}_2$ gets the following values with probability ϵ^2: $u_1 = \mathcal{O}_2(g, (g^a)^r, (g^c)^t) = e(g,g)^{a^2cr^2t}$, and $u_2 = \mathcal{O}_2(g, (g^b)^s, (g^c)^t) = e(g,g)^{b^2cs^2t}$. Finally, $\mathcal{A}_2$ gets $v = \mathcal{O}_2(g, (g^a)^r \cdot (g^b)^s, (g^c)^t) = e(g,g)^{(ar+bs)^2 \cdot ct} = e(g,g)^{a^2cr^2t+b^2cs^2t+2abcrst}$ from which $e(g,g)^{abc}$ can be computed as r, s, t are known already, with probability at least $\epsilon(1-\frac{1}{q-1})$ according to Proposition 1; Specifically, the statistical distance between $ar+bs$ and the uniform distribution over Z_q^* is $\frac{1}{q-1}$. We conclude that, with probability at least $\epsilon^3(1-\frac{1}{q-1})$, $\mathcal{A}_2$ can solve the BDH problem.

Definition 4 (Gap Bilinear Diffie-Hellman (Gap-BDH)) *The gap bilinear Diffie-Hellman (*Gap-BDH*) problem [39, 6] is to compute $e(g,g)^{abc} \in \mathbb{G}_\mathsf{T}$, given $(g, g^a, g^b, g^c) \in \mathbb{G}_1^4$, where $a, b, c \leftarrow \mathbb{Z}_q^*$, but with the help of a decisional bilinear Diffie-Hellman (*DBDH*) oracle for $\mathbb{G}_1 = \langle g \rangle$ and $\mathbb{G}_\mathsf{T}$. Here, on arbitrary input $(A = g^a, B = g^b, C = g^c, T) \in \mathbb{G}_1^3 \times \mathbb{G}_\mathsf{T}$, the* DBDH *oracle outputs*

1 if and only if $T = e(g,g)^{abc}$. *The* Gap-BDH *assumption says that no PPT algorithm can solve the* Gap-BDH *problem with non-negligible probability.*

Definition 5 (Gap Square Bilinear Diffie-Hellman) *The gap square bilinear Diffie-Hellman (*Gap-SBDH*) problem is to compute* $e(g,g)^{a^2b} \in \mathbb{G}_{\mathsf{T}}$, *given* $(g, g^a, g^b) \in \mathbb{G}_1^3$, *where* $a, b \leftarrow \mathbb{Z}_q^*$, *but with the help of a decisional bilinear Diffie-Hellman (*DBDH*) oracle for* $\mathbb{G}_1 = \langle g \rangle$ *and* $\mathbb{G}_{\mathsf{T}}$. *Here, on arbitrary input* $(A' = g^{a'}, B' = g^{b'}, C' = g^{c'}, T) \in \mathbb{G}_1^3 \times \mathbb{G}_{\mathsf{T}}$, *the* DBDH *oracle outputs 1 if and only if* $T = e(g,g)^{a'b'c'}$. *The* Gap-SBDH *assumption says that no PPT algorithm can solve the* Gap-SBDH *problem with non-negligible probability.*

Clearly, by Theorem 1, the Gap-BDH assumption and the Gap-SBDH assumption are equivalent.

4 Identity-Based Higncryption: Definition and Security Model

4.1 Definition of IBHigncryption

In an identity-based higncryption scheme (IBHigncryption), denoted IBHC, there is a private key generator (PKG) who is responsible for the generation of private keys for the users in the system. The PKG computes the private key for each user using its master secret key on the user's public identity. Next, we give the formal definition of an IBHigncryption.

Definition 6 (IBHigncryption**)** *An* IBHigncryption *scheme* IBHC *with associated data, consists of the following four polynomial-time algorithms:* Setup, KeyGen, IBHigncrypt, *and* UnIBHigncrypt.

- $\mathsf{Setup}(1^\kappa) \rightarrow (\mathsf{par}, \mathsf{msk})$*: The algorithm is run by the PKG. On input of the security parameter* κ, *it outputs the system's common parameters* par *and the master secret key* msk. *Finally, the PKG outputs* par, *and it keeps the master secret key* msk *in private. We assume that the security parameter and an admissible identity space* $\mathcal{ID}$ *are always (implicitly) encoded in* par.

- $\mathsf{KeyGen}(\mathsf{par}, \mathsf{msk}, \mathsf{ID}) \rightarrow sk$*: On input of the system's public parameters* par, *the master secret key* msk *of the PKG, and a user's identity* ID, *the PKG computes and outputs the private key* sk *of* ID *using* msk *if* $\mathsf{ID} \in \mathcal{ID}$. *The public identity and its private key are for algorithm* IBHigncrypt *and algorithm* UnIBHigncrypt *respectively.*

- $\mathsf{IBHigncrypt}(\mathsf{par}, sk_s, \mathsf{ID}_s, \mathsf{ID}_r, H, M) \rightarrow (C, \perp)$*: It is a* PPT *algorithm. On input of the system's public parameters* par, *a sender's private*

key sk_s, and his public identity $\mathsf{ID}_s \in \mathcal{ID}$, a receiver's public identity $\mathsf{ID}_r \in \mathcal{ID}$, a message $M \in \{0,1\}^*$ and its associated data $H \in \{0,1\}^*$ to be IBHigncrypted*, it outputs an* IBHigncryptext *$C \in \{0,1\}^*$, or $\perp$ indicating* IBHigncrypt*'s failure. The associated data H, if there is any, appears in clear in the* IBHigncryptext *C, when $C \neq \perp$.*

- $\mathsf{UnIBHigncrypt}(\mathsf{par}, sk_r, \mathsf{ID}_r, C) \rightarrow ((\mathsf{ID}_s, M), \perp)$*: It is a deterministic algorithm. On input of the system's public parameters* par*, the receiver's private key sk_r, the receiver's public identity $\mathsf{ID}_r \in \mathcal{ID}$, and an* IBHigncryptext *C, it outputs (ID_s, M) if the verification is successful, or $\perp$ indicating an error, where $\mathsf{ID}_s \in \mathcal{ID}$ is the sender's public identity, and $M \in \{0,1\}^*$ is the message* IBHigncrypted *by ID_s. It is different from the traditional identity-based signcryption in that* UnIBHigncrypt *does not need to take the sender's public identity ID_s as input.*

Definition 7 (correctness) *We say an* IBHigncryption *scheme* IBHC *is* correct*, if for any sufficiently large security parameter κ, any key pairs (ID_s, sk_s), and (ID_r, sk_r), where sk_s and sk_r are output by* KeyGen *on ID_s and ID_r respectively, it holds that* UnIBHigncrypt*$(\mathsf{par}, sk_r, \mathsf{ID}_r,$* IBHigncrypt*$(\mathsf{par}, sk_s, \mathsf{ID}_s, \mathsf{ID}_r, H, M)) = (\mathsf{ID}_s, M)$ for any $H, M \in \{0,1\}^*$ such that $\mathsf{IBHigncrypt}(\mathsf{par}, sk_s, \mathsf{ID}_s, \mathsf{ID}_r, H, M) \neq \perp$.*

Definition 8 (receiver deniability) *We say that an* IBHigncryption *scheme* IBHC *has receiver deniability, if the same* IBHigncryptext *can be generated either by the sender or the receiver. Specifically, there exists a PPT algorithm $\mathsf{IBHigncrypt}'(\mathsf{par}, sk_r, \mathsf{ID}_s, \mathsf{ID}_r, H, M) \rightarrow (C, \perp)$, satisfying: the output of $\mathsf{IBHigncrypt}'(\mathsf{par}, sk_r, \mathsf{ID}_s, \mathsf{ID}_r, H, M)$ has the same distribution as that of $\mathsf{IBHigncrypt}(\mathsf{par}, sk_s, \mathsf{ID}_s, \mathsf{ID}_r, H, M)$, for any security parameter κ, any $H, M \in \{0,1\}^*$, and any key pairs (ID_s, sk_s) and (ID_r, sk_r) where sk_s and sk_r are output by* KeyGen *on ID_s and ID_r respectively.*

Remark 1 *Deniability has always been a central privacy concern in personal and business communications, with off-the-record communication serving as an essential social and political tool [42]. Given that many of these interactions now happen over digital media (e.g., email, instant messaging, web transactions, virtual private networks), it is critically important to provide these communications with "off-the-record" or deniability capability to protocol participants.*[3] *For these applications, we may only concern about the authentication of the communication, and less care about the non-repudiation of the communication.*

[3]Needless to say, there are special applications where non-repudiable communication is essential. But this is not the case for most of our nowaday communications over Internet, where deniable authentication is much more desirable than non-repudiable one [42].

4.2 Security Model for IBHigncryption

We focus on the security model for IBHigncryption in the multi-user environment, where each user possesses a single key pair for both IBHigncrypt and UnIBHigncrypt, and the sender can IBHigncrypt messages to itself. Our security model is stronger than that of an identity-based signcryption, since it allows the adversaries to access more oracles.

The private keys of all the users in the system are generated by the challenger by running the specified key generation algorithm KeyGen. All the users' public identities are given to the adversary initially. Throughout this work, denote by ID_i, the public identity of user i, and denote by ID_s (resp., ID_r) the public identity of the sender (resp., the receiver). For presentation simplicity, throughout this work we assume that all the users in the system have public identity information of equal length. But our security model and protocol construction can be extended to the general case of different lengths of identities, by incorporating length-hiding authenticated encryption [40] in the underlying security model and protocol construction.

The security of an IBHigncryption includes two parts: outsider unforgeability (OU) and insider confidentiality (IC). In order to formally define the above security, we introduce two types of adversaries in our system, one is called OU-adversary, $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$, and the other is called IC-adversary, $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$. The goal of an $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$ is to forge a valid IBHigncryptext on behalf of an uncorrupted sender ID_{s^*} to an uncorrupted receiver ID_{r^*}, where ID_{s^*} may be equal to ID_{r^*}. The goal of an $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$ adversary is to break the confidentiality of the message or the privacy of the sender's identity for any IBHigncryptext from any (even corrupted) sender to any uncorrupted receiver, even if $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$ is allowed to corrupt the sender and to expose the intermediate randomness used for generating other IBHigncryptexts. Likewise, here the sender may be equal to the receiver. The terminology "insider" (resp., "outsider"), which is traditional in this literature, refers to the situation that the target sender can (resp., cannot) be corrupted.

Now, we describe the oracles to which $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$ or $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$ gets access in our security model for IBHigncryption.

- HO Oracle : This oracle is used to respond to the IBHigncrypt queries made by an adversary, including $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$ or $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$. On input $(\mathsf{ID}_s, \mathsf{ID}_r, H, M)$ by an adversary, where $\mathsf{ID}_r \in \mathcal{ID}$ may be equal to $\mathsf{ID}_s \in \mathcal{ID}$, and $H, M \in \{0,1\}^*$, this oracle returns $C = \mathsf{IBHigncrypt}(\mathsf{par}, sk_s, \mathsf{ID}_s, \mathsf{ID}_r, H, M)$ to the adversary. In order to respond to some EXO queries against C by the adversary, the HO Oracle needs to store some specified offline-computable intermediate randomness (which is used in generating C) into an initially empty table $\mathsf{ST_C}$ privately.

- UHO Oracle: This oracle is used to respond to the UnIBHigncrypt queries made by an adversary. On input (ID_r, C) by an adversary, this

oracle returns $\mathsf{UnIBHigncrypt}(\mathsf{par}, sk_r, \mathsf{ID}_r, C)$ to the adversary, where sk_r is the private key of the receiver $\mathsf{ID}_r \in \mathcal{ID}$.

- EXO Oracle: This oracle is used to respond to the intermediate randomness used in generating an IBHigncryptext of an earlier HO query. It is an additional oracle in our security model that makes our security stronger than the traditional security for signcryption; This feature is considered and named as x-security in [32]. On input an IBHigncryptext C, this oracle returns the value (i.e., the offline-computable intermediate randomness used in generating C) stored in the table $\mathsf{ST_C}$, if $C \neq \perp$ and C was an output of an earlier HO query. If there is no such a record in $\mathsf{ST_C}$, this oracle returns $\perp$ to the adversary.

- CORRUPT Oracle: This oracle is used to respond to the private key queries for any user in the system. On input a user's identity $\mathsf{ID}_i \in \mathcal{ID}$, this oracle returns the private key $sk_i = \mathsf{KeyGen}(\mathsf{par}, \mathsf{msk}, \mathsf{ID}_i)$, and ID_i is then marked as a corrupted user. Denote by $\mathsf{S_{corr}}$ the set of corrupted users in the system, which is initially empty. This oracle updates $\mathsf{S_{corr}}$ with $\mathsf{S_{corr}} := \mathsf{S_{corr}} \bigcup \{\mathsf{ID}_i\}$ whenever the private key of ID_i is returned to the adversary.

Next, we describe the security games for insider confidentiality (IC) and outsider unforgeability (OU).

Definition 9 (Insider Confidentiality (IC)) *Let $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$ be an IC-adversary against* IBHC. *We consider the following game, denoted by $\mathsf{GAME}^{\mathcal{A}^{\mathsf{IC}}}_{\mathsf{IBHC}}$, in which an adversary $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$ interacts with a challenger $\mathcal{C}$.*

- Setup*: The challenger $\mathcal{C}$ runs* Setup *to generate the system public parameters* par *and a master secret key* msk*. The challenger returns* par *to the adversary $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$, and keeps the* msk *secretly for itself.*

- Phase 1*: In this phase, $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$ issues any polynomial number of queries, including* HO, UHO, EXO, *and* CORRUPT.

- Challenge*: At the end of* phase 1, *$\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$ selects in the identity space $\mathcal{ID}$ two different target senders, $\mathsf{ID}_{s_0^*}$ and $\mathsf{ID}_{s_1^*}$, and an uncorrupted target receiver ID_{r^*}, a pair of messages (M_0^*, M_1^*) of equal length from the message space, and associated data H^*. $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$ submits (M_0^*, M_1^*), H^*, and $(\mathsf{ID}_{s_0^*}, \mathsf{ID}_{s_1^*}, \mathsf{ID}_{r^*})$ to the challenger $\mathcal{C}$. The challenger $\mathcal{C}$ chooses $\sigma \leftarrow \{0,1\}$, and gives the challenge* IBHigncryptext

$$C^* = \mathsf{IBHigncrypt}(\mathsf{par}, sk_{s_\sigma^*}, \mathsf{ID}_{s_\sigma^*}, \mathsf{ID}_{r^*}, H^*, M_\sigma^*)$$

to the adversary $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$. Here, we stress that there is no restriction on selecting the target senders $\mathsf{ID}_{s_0^*}$ and $\mathsf{ID}_{s_1^*}$. It implies that both

target senders can be corrupted, which captures forward ID*-privacy; And either one of the target senders can be the target receiver (i.e., it may be the case that* $\mathsf{ID}_{s^*_\sigma} = \mathsf{ID}_{r^*}$*).*

- Phase 2*:* $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$ *continues to make queries as in* phase 1 *with the following restrictions:*
  1. $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$ *is not allowed to issue* $\mathsf{CORRUPT}(\mathsf{ID}_{r^*})$.
  2. $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$ *is not allowed to issue* $\mathsf{UHO}(\mathsf{ID}_{r^*}, C^*)$.
  3. $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$ *is not allowed to issue* $\mathsf{EXO}(C^*)$.
- Guess*: Finally,* $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$ *outputs* $\sigma' \in \{0,1\}$ *as his guess of the random bit* σ*.* $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$ *wins the game if* $\sigma' = \sigma$*.*

With respect to the above security game $\mathsf{GAME}^{\mathcal{A}^{\mathsf{IC}}}_{\mathsf{IBHC}}$*, we define the advantage of an* $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$ *adversary in* $\mathsf{GAME}^{\mathcal{A}^{\mathsf{IC}}}_{\mathsf{IBHC}}$ *as:*

$$\mathsf{Adv}^{\mathcal{A}^{\mathsf{IC}}}_{\mathsf{IBHC}} = |2 \cdot \mathsf{Pr}[\sigma' = \sigma] - 1|.$$

We say that an IBHigncryption *scheme* IBHC *has insider confidentiality, if for any* PPT *adversary* $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$*, its advantage* $\mathsf{Adv}^{\mathcal{A}^{\mathsf{IC}}}_{\mathsf{IBHC}}$ *is negligible for any sufficiently large security parameter.*

Definition 10 (Outsider Unforgeability (OU)) *Let* $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$ *be an OU-adversary against* IBHC*. We consider the following game, denoted by* $\mathsf{GAME}^{\mathcal{A}^{\mathsf{OU}}}_{\mathsf{IBHC}}$*, in which an adversary* $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$ *interacts with a challenger* $\mathcal{C}$*.*

- Phase 1*: The challenger* $\mathcal{C}$ *runs* Setup *to generate the system public parameters* par *and a master secret key* msk*. The challenger returns* par *to the adversary* $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$*, and keeps the* msk *for itself in private.*
- Phase 2*: In this phase,* $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$ *issues any polynomial number of queries, including* HO*,* UHO*,* EXO*, and* CORRUPT*.*
- Phase 3*: In this* phase*,* $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$ *outputs* (ID_{r^*}, C^*) *as its forgery, where* $\mathsf{ID}_{r^*} \notin \mathsf{S}_{\mathsf{corr}}$ *and the associated data contained in* C^* *in clear is denoted by* H^**. We say the forgery* (ID_{r^*}, C^*) *is a valid* IBHigncryptext *created by an uncorrupted sender* $\mathsf{ID}_{s^*} \in \mathcal{ID}$ *for an uncorrupted receiver* $\mathsf{ID}_{r^*} \in \mathcal{ID}$ *if and only if the following conditions hold simultaneously:*
  1. $\mathsf{UnIBHigncrypt}(sk_{r^*}, \mathsf{ID}_{r^*}, C^*) = (\mathsf{ID}_{s^*}, M^*)$*, where* $\mathsf{ID}_{s^*} \in \mathcal{ID} \setminus \mathsf{S}_{\mathsf{corr}}$*,* $M^* \in \{0,1\}^*$*, and* ID_{s^*} *may be equal to* ID_{r^*}*.*
  2. $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$ *is not allowed to issue* CORRUPT *queries on* ID_{s^*} *or* ID_{r^*}*.*

3. $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$ *is allowed to issue* $\mathsf{HO}(\mathsf{ID}_{s'}, \mathsf{ID}_{r'}, H', M')$ *for any* $(\mathsf{ID}_{s'}, \mathsf{ID}_{r'}, H', M') \neq (\mathsf{ID}_{s^*}, \mathsf{ID}_{r^*}, H^*, M^*)$. *In particular,* $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$ *can make an* HO *query on* $(\mathsf{ID}_{s^*}, \mathsf{ID}_{r^*}, H', M^*)$, *where* $H' \neq H^*$. *It can even make the query* $\mathsf{HO}(\mathsf{ID}_{s^*}, \mathsf{ID}_{r^*}, H^*, M^*)$, *as long as the output returned is not equal to* C^*.

Let $\mathsf{Adv}^{\mathcal{A}^{\mathsf{OU}}}_{\mathsf{IBHC}}$ *denote the advantage that* $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$ *outputs a valid forgery in the above security game* $\mathsf{GAME}^{\mathcal{A}^{\mathsf{OU}}}_{\mathsf{IBHC}}$. *We say an* IBHigncryption *scheme* IBHC *has outsider unforgeability, if for any* PPT *adversary* $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$, *its advantage* $\mathsf{Adv}^{\mathcal{A}^{\mathsf{OU}}}_{\mathsf{IBHC}}$ *is negligible for any sufficiently large security parameter.*

Remark 2 *Note that the above definition of outsider unforgeability implies the x-security considered and named in [32]. Specifically, getting access to the oracle* EXO in an arbitrary way *does not allow the adversary to forge* IBHigncryptext *(in particular, to recover the secret key of any uncorrupted user).*

5 IBHigncryption: Construction and Discussion

For presentation simplicity, below we only present the construction of IBHigncryption based on bilinear pairings of Type 1. The extensions to Type 2 and 3 pairings are straightforward, and are presented in Appendix A.

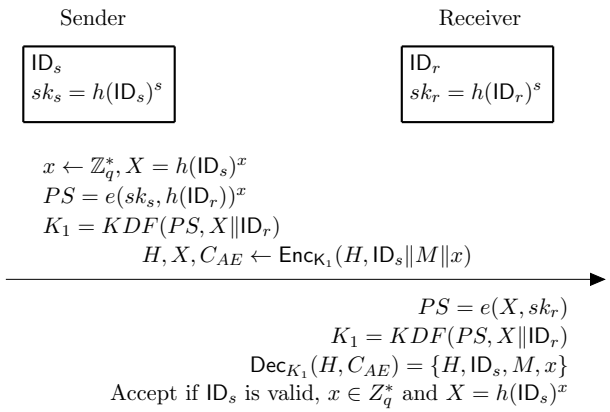

Figure 2: Protocol Structure of IBHigncryption

Our IBHigncryption scheme consists of the following four algorithms:

- $\mathsf{Setup}(1^\kappa)$: The algorithm is run by the PKG in order to produce the system's public parameters and the master secret key. On input of the security parameter κ, it chooses two multiplicative bilinear map groups $\mathbb{G}_1 = \langle g \rangle$ and $\mathbb{G}_\mathsf{T}$ of the same prime order q such that the discrete

logarithm problems in both $\mathbb{G}_1$ and $\mathbb{G}_\mathsf{T}$ are intractable. The algorithm constructs a bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_\mathsf{T}$, and chooses $s \leftarrow \mathbb{Z}_q^*$. Additionally, it selects a one-way collision-resistant cryptographic hash function, $h : \{0,1\}^* \to \mathbb{G}_1$. Finally, the algorithm outputs the public parameters $\mathsf{par} = (q, \mathbb{G}_1, \mathbb{G}_\mathsf{T}, e, g, h)$, and the PKG's master secret key $\mathsf{msk} = s$. The PKG makes par public to the users in the system, but keeps msk secret for itself. Note that the setup stage is much simpler, where in particular no modular exponentiation is performed in order to generate a traditional master public key as in [16] and [9]. For presentation simplicity, we assume the admissible identity space $\mathcal{ID} = \{0,1\}^*$.

- $\mathsf{KeyGen}(\mathsf{par}, \mathsf{msk}, \mathsf{ID})$: On input of the system's public parameters par, the master secret key msk of PKG, and a user's identity $\mathsf{ID} \in \{0,1\}^*$, the PKG computes $sk = h(\mathsf{ID})^{\mathsf{msk}} = h(\mathsf{ID})^s$, and outputs sk_{ID} as the private key associated with identity ID.

- $\mathsf{IBHigncrypt}(\mathsf{par}, sk_s, \mathsf{ID}_s, \mathsf{ID}_r, H, M)$: Let $\mathsf{SE} = (\mathsf{K_{se}}, \mathsf{Enc}, \mathsf{Dec})$ be an authenticated encryption (AE) scheme as defined in Section 2.1, $M \in \{0,1\}^*$ be the message to be $\mathsf{IBHigncrypted}$ with associated data $H \in \{0,1\}^*$, and $\mathsf{KDF} : \mathbb{G}_\mathsf{T} \times \{0,1\}^* \to \mathcal{K}$ be a key derivation function that is modelled to be a random oracle [13], where $\mathcal{K}$ is the key space of $\mathsf{K_{se}}$. For presentation simplicity, we denote by ID_s the sender's public identity whose private key is $sk_s = h(\mathsf{ID}_s)^s$, and by ID_r the receiver's public identity whose private key is $sk_r = h(\mathsf{ID}_r)^s$.

  To IBHigncrypt a message $M \leftarrow \{0,1\}^*$ with the sender's identity ID_s concealed, the sender ID_s runs the following steps: (1) selects $x \leftarrow \mathbb{Z}_q^*$, and computes $X = h(\mathsf{ID}_s)^x \in \mathbb{G}_1$; (2) computes the pre-shared secret $PS = e(sk_s, h(\mathsf{ID}_r))^x \in \mathbb{G}_\mathsf{T}$; (3) derives the AE key $K_1 = KDF(PS, X \| \mathsf{ID}_r) \in \mathcal{K}$; (4) computes $C_{AE} \leftarrow \mathsf{Enc}_{\mathsf{K}_1}(H, \mathsf{ID}_s \| M \| x)$; and finally (5) sends the $\mathsf{IBHigncryptext}$ $C = (H, X, C_{AE})$ to the receiver ID_r.

- $\mathsf{UnIBHigncrypt}(\mathsf{par}, sk_r, \mathsf{ID}_r, C)$: On receiving $C = (H,\ X,\ C_{AE})$, the receiver ID_r with private key sk_r does the following: (1) computes the pre-shared secret $PS = e(X, sk_r) \in \mathbb{G}_\mathsf{T}$, and derives the key $K_1 = KDF(PS, X \| \mathsf{ID}_r) \in \mathcal{K}$; (2) runs $\mathsf{Dec}_{K_1}(H, C_{AE})$. If $\mathsf{Dec}_{K_1}(H, C_{AE})$ returns $\perp$, it aborts; Otherwise, the receiver gets $\{\mathsf{ID}_s, M, x\}$, and outputs (ID_s, M) if $\mathsf{ID}_s \in \mathcal{ID}$, $x \in Z_q^*$, and $X = h(\mathsf{ID}_s)^x$. Otherwise, it outputs "$\perp$" and aborts.

Remark 3 *The correctness and the property of receiver deniability of the above* $\mathsf{IBHigncryption}$ *are straightforward. It also enjoys x-security and forward ID-privacy, which are implied by the formal analyses of outsider unforgeability and insider confidentiality to be given in Section 6.*

		IBHigncryption	BF-IBE [16]
par		$(q, \mathbb{G}_1, \mathbb{G}_\mathsf{T}, e, g, h)$	$(q, \mathbb{G}_1, \mathbb{G}_\mathsf{T}, e, n, g, P_{\mathsf{pub}}, h_1, h_2, h_3, h_4)$
efficiency	Setup	-	1 E
	KeyGen	1 E + 1 H_2	1 E + 1 H_2
	Sender	2 E + 1 P + 2 H_2 + 1 Enc	2 E + 1 P + 1 H_2 + 3 H_1
	Receiver	1 E + 1 P + 1 H_2 + 1 Dec	1 E + 1 P + 3 H_1
message space		$\{0,1\}^*$	$\{0,1\}^n$
assumption		Gap-SBDH	BDH

Table 2: Brief comparison between IBHigncryption and CCA-secure BF-IBE

		IBHigncryption	IEEE P1363.3 [9]
par		$(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\mathsf{T}, g_1, g_2, e, \psi, h)$	$(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\mathsf{T}, g_1, g_2, g, Q_{\mathsf{pub}}, e, \psi, h_1, h_2, h_3)$
efficiency	Setup	1 ψ	1 E + 1 P + 1 ψ
	KeyGen	1 E + 1 H_2	1 E + 1 INV + 1 H_1 + 1 A
	Sender	2 E + 1 P + 2 H_2 + 1 ψ + 1 Enc	4 E +2 ψ + 3 H_1 + 1 M +1 A
	Receiver	1 E + 1 P + 1 H_2 + 1 ψ + 1 Dec	2 E + 2 P + 3 H_1 + 1 M_T +1 M+ 1 A
message space		$\{0,1\}^*$	$\{0,1\}^n$
forward ID-privacy		✓	×
x-security		✓	×
receiver deniability		✓	×
consider $\mathsf{ID}_s = \mathsf{ID}_r$		✓	×
assumption		Gap-SBDH	q-BDHIP

Table 3: Brief comparison between IBHigncryption and IEEE P1363.3

5.1 Comparison and Discussion

In this section, we briefly compare our IBHigncryption scheme with the CCA-secure Boneh-Franklin IBE [16] (referred to as BF-IBE), and the IEEE P1363.3 standard of ID-based signcryption [9] (referred to as IEEE P1363.3 for simplicity). The schemes of BF-IBE and IEEE P1363.3 are reviewed in Appendix B and C, respectively.

The comparisons between our IBHigncryption scheme based on symmetric bilinear pairings of Type 1 and BF-IBE [16], and our IBHigncryption scheme based on asymmetric bilinear pairings of Type 2 and the IEEE P1363.3 standard [9], are briefly summarized in Table 2 and Table 3 respectively. Therein, $\perp$ denotes "unapplicable", "-" denotes no exponentiation operation, "E" denotes modular exponentiation, "P" denotes paring, "H_1" denotes a plain hashing, "H_2" denotes a hashing onto the bilinear group, "A" denotes modular addition, "M" (resp., M_T) denotes modular multiplication in G_1 or G_2 (resp., G_T), "INV" denotes modular inversion, and ψ denotes isomorphism. Note that modular inverse is a relatively expensive operation, which is typically performed by the extended Euclid algorithm.

In comparison with BF-IBE [16] and IEEE P1363.3 [9]), IBHigncryption has a much simpler setup stage. Specifically, the setup stage of our IBHigncryption has much smaller public parameters, and actually does not need to perform

exponentiation to generate the master public key (corresponding to P_{pub} in BF-IBE, and Q_{pub} in IEEE P1363.3). The much simpler setup stage of IBHigncryption, particularly waiving the master public key, brings the following advantages:

- The computational and space complexity for generating and storing the system parameters is reduced.
- The attack vector (for recovering the master secret key) is decreased, e.g., for some mission critical applications.
- It eases deployment and compatibility with existing identity-based cryptosystems. Specifically, when deploying our IBHigncryption scheme in reality with other existing identity-based cryptosystems, the system parameters and particularly the master public key can remain unchanged.

For IEEE P1363.3 [9], if the secret x is exposed one can compute from the corresponding signcryptext the following values: the message M being signcrypted, and more importantly the secret key value $\psi(sk_{\mathsf{ID_A}})$ which then allows the attacker to impersonate the sender in an arbitrary way. This shows that IEEE P1363.3 lacks the x-security (specifically, cannot be outsider unforgeable when getting access to the EXO oracle is allowed). We also note that the provable security of IEEE P1363.3 [9] does not consider the case of $\mathsf{ID}_s = \mathsf{ID}_r$.

For computational efficiency, briefly speaking, our IBHigncryption is essentially as efficient as BF-IBE [16], while providing the functionalities of encryption, authentication, and ID-privacy simultaneously and with a much simpler setup stage. In other words, compared with BF-IBE, the functionalities of authentication and ID-privacy are gotten almost for free with IBHigncryption. In comparison with IEEE P1363.3 [9], besides the extra properties of forward ID-privacy, x-security, receiver deniability, IBHigncryption is also computationally more efficient in total. Note that the plaintext spaces for BF-IBE and IEEE P1363.3 are pre-specified to be $\{0,1\}^n$. If one employs the hybrid encryption approach to encrypt messages of arbitrary length with BF-IBE or IEEE P1363.3, it also needs to employ some appropriate symmetric-key encryption scheme in reality.

6 Security Analysis of IBHigncryption

Due to space limitation, we focus on the security proof of our IBHigncryption construction with symmetric bilinear groups. The extension to the asymmetric bilinear groups is straightforward. In the following security analysis, KDF and the hash function h are modelled as random oracles (RO).

Theorem 2 *The* IBHigncryption *scheme presented in Fig. 2 is outsider unforgeable in the random oracle model, under the AE security and the* Gap-SBDH *assumption.*

Theorem 3 *The* IBHigncryption *scheme presented in Fig. 2 has insider confidentiality in the random oracle model, under the AE security and the* Gap-SBDH *assumption.*

6.1 Proof of Outsider Unforgeability

In this section, we prove Theorem 2 in detail.

At first, the challenger $\mathcal{C}$ takes a tuple $(g, g^a, g^c) \in \mathbb{G}_1^3$ and a paring $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_\mathsf{T}$ as its inputs, where g is a generator of $\mathbb{G}_1$ and $a, c \leftarrow Z_q^*$ that are actually unknown to $\mathcal{C}$. The goal of $\mathcal{C}$ is to compute $T = e(g, g)^{a^2 c} \in \mathbb{G}_\mathsf{T}$ with the help of a DBDH oracle denoted $\mathcal{O}_{\mathsf{DBDH}}$, i.e., to solve the gap square bilinear Diffie-Hellman (Gap-SBDH) problem as defined in Section 3. Towards this goal, the challenger $\mathcal{C}$ runs the adversary $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$ who is assumed to break the outsider unforgeability of IBHC with non-negligible probability. During the simulation, the challenger $\mathcal{C}$ maintains four tables $\mathsf{T_h}, \mathsf{S_{corr}}, \mathsf{K_{KDF}}$, and $\mathsf{ST_C}$. They are all initialized to be empty.

Phase 1: $\mathcal{C}$ sets the public parameters $\mathsf{par} = (q, \mathbb{G}_1, \mathbb{G}_\mathsf{T}, e, g, h)$, where q is the prime order of $\mathbb{G}_1$ and $\mathbb{G}_\mathsf{T}$, and $h : \{0,1\}^* \to \mathbb{G}_1$ is a collision-resistant cryptographic hash function that is modelled as a random oracle. The challenger $\mathcal{C}$ defines the master secret key $\mathsf{msk} = c$ (note that a, c are unknown to $\mathcal{C}$). Finally, $\mathcal{C}$ gives par to the adversary $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$.

Hash Query on $h : \{0,1\}^* \to \mathbb{G}_1$: On input of a user's identity ID_i, the challenger $\mathcal{C}$ chooses a random $y_i \leftarrow \mathbb{Z}_q^*$. Using the techniques of Coron [23], $\mathcal{C}$ flips a biased coin $b_i \in \{0,1\}$ satisfying $b_i = 1$ with probability γ and 0 otherwise. If $b_i = 1$, $\mathcal{C}$ sets $h(\mathsf{ID}_i) = g^{y_i}$. Otherwise (i.e., $b_i = 0$), $\mathcal{C}$ sets $h(\mathsf{ID}_i) = (g^a)^{y_i}$. The challenger returns $h(\mathsf{ID}_i)$ to $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$, and stores $(\mathsf{ID}_i, b_i, y_i, h(\mathsf{ID}_i))$ into the table $\mathsf{T_h}$.

Phase 2: $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$ issues a number of queries adaptively, including HO, UHO, EXO, and CORRUPT. With respect to each kind of queries, the challenger $\mathcal{C}$ responds to $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$ as following:

- CORRUPT Query:

  For a CORRUPT query on user ID_i, $\mathcal{C}$ first visits table $\mathsf{T_h}$. If $b_i = 1$, $\mathcal{C}$ returns $sk_i = h(\mathsf{ID}_i)^c = (g^c)^{y_i}$, and sets $\mathsf{S_{corr}} := \mathsf{S_{corr}} \bigcup \{\mathsf{ID}_i\}$. Otherwise, $\mathcal{C}$ aborts.

- HO Query:

  For an HO query on $(\mathsf{ID}_s, \mathsf{ID}_r, H, M)$, $\mathcal{C}$ first visits table $\mathsf{T_h}$, and gets the entries corresponding to ID_s and ID_r, i.e., $(\mathsf{ID}_s, b_s, y_s, h(\mathsf{ID}_s))$ and $(\mathsf{ID}_r, b_r, y_r, h(\mathsf{ID}_r))$. We further consider the following cases:

1. $b_s = 1$

---

the challenger $\mathcal{C}$ selects $x \leftarrow \mathbb{Z}_q^*$;
sets $X = h(\mathsf{ID}_s)^x = (g^{y_s})^x$;
if $b_r = 1$
$\mathcal{C}$ computes
$PS = e(sk_s, h(\mathsf{ID}_r))^x = e((g^c)^{y_s}, g^{y_r})^x$;
$K_1 = KDF(PS, X \| \mathsf{ID}_r)$;
else
$\mathcal{C}$ computes
$PS = e(sk_s, h(\mathsf{ID}_r))^x = e((g^c)^{y_s}, (g^a)^{y_r})^x$;
$K_1 = KDF(PS, X \| \mathsf{ID}_r)$;
endif
stores the tuple $((PS, X \| \mathsf{ID}_r), K_1)$ into $\mathsf{K}_{\mathsf{KDF}}$;
computes $C_{\mathsf{AE}} \leftarrow \mathsf{Enc}_{K_1}(H, \mathsf{ID}_s \| M \| x)$;
returns $C = (H, X, C_{\mathsf{AE}})$ to $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$;
stores the tuple (C, x) into the table ST_{C}.

---

2. $b_s = 0$

---

the challenger $\mathcal{C}$ selects $x \leftarrow \mathbb{Z}_q^*$;
sets $X = h(\mathsf{ID}_s)^x = (g^a)^{x \cdot y_s}$;
if $b_r = 1$
$\mathcal{C}$ computes
$PS = e(sk_s, h(\mathsf{ID}_r))^x = e((g^c)^{y_s}, (g^a)^{y_r})^x$;
$K_1 = KDF(PS, X \| \mathsf{ID}_r)$;
$\mathcal{C}$ stores the tuple $((PS, X \| \mathsf{ID}_r), K_1)$ into $\mathsf{K}_{\mathsf{KDF}}$;
else
$\mathcal{C}$ sets K_1 to be a string taken uniformly at random from $\mathcal{K}$ of AE;
$\mathcal{C}$ stores the tuple $((\star, X \| \mathsf{ID}_r), K_1)$ into $\mathsf{K}_{\mathsf{KDF}}$;
endif
computes $C_{\mathsf{AE}} \leftarrow \mathsf{Enc}_{K_1}(H, \mathsf{ID}_s \| M \| x)$;
returns $C = (H, X, C_{\mathsf{AE}})$ to $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$;
stores the tuple (C, x) into the table ST_{C}.

Note that for the above HO queries, if $b_s \neq 0$ or $b_r \neq 0$, the simulation of $\mathcal{C}$ is perfect by the properties of $\mathsf{IBHigncryption}$ (in particular, the receiver deniability for the case of $b_s = 0 \wedge b_r = 1$). However, if $b_s = b_r = 0$, the challenger $\mathcal{C}$ cannot compute the pre-shared secret:

$$PS = e(sk_s, h(\mathsf{ID}_r))^x = \mathsf{BDH}(X, h(\mathsf{ID}_r), g^c),$$

and consequently $KDF(PS, X\|\mathsf{ID}_r)$. Whenever $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$ makes an oracle of the form $KDF(PS', X'||\mathsf{ID}'_r)$ for some ID'_r with $b'_r = 0$, such that the random oracle KDF has not been defined over $(PS', X'||\mathsf{ID}'_r)$ but $(X'||\mathsf{ID}'_r = X||\mathsf{ID}_r)$ for some entry $((\star, X\|\mathsf{ID}_r), K_1)$ in the table $\mathsf{K}_{\mathsf{KDF}}$, $\mathcal{C}$ does the following to ensure simulation consistency. The challenger $\mathcal{C}$ first makes a DBDH oracle query $\mathcal{O}_{\mathsf{DBDH}}(X', h(\mathsf{ID}'_r), g^c, PS')$, and acts as follows:

- If $\mathcal{O}_{\mathsf{DBDH}}$ answers 1, $\mathcal{C}$ returns to $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$ the value K_1 already stored in $((\star, X\|\mathsf{ID}_r), K_1)$; meanwhile, $\mathcal{C}$ updates the table $\mathsf{K}_{\mathsf{KDF}}$ by replacing $\star$ in $((\star, X\|\mathsf{ID}_r), K_1)$ with PS'.
- If $\mathcal{O}_{\mathsf{DBDH}}$ returns 0, $\mathcal{C}$ takes K'_1 uniformly at random from $\mathcal{K}$ of AE, returns K'_1 to $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$, and stores $((PS', X'\|\mathsf{ID}'_r), K'_1)$ into $\mathsf{K}_{\mathsf{KDF}}$.

- EXO Query:

  For an EXO query on C, the challenger $\mathcal{C}$ first visits the table ST_{C}. If there is an entry in the table, $\mathcal{C}$ returns the corresponding x to the adversary. Otherwise, $\mathcal{C}$ returns $\perp$ to the adversary.

- UHO Query:

  For a UHO query on $(\mathsf{ID}_r, C = (H, X, C_{AE}))$: If ID_r's corresponding value $b_r = 1$, $\mathcal{C}$ can perfectly simulate the game. Therefore, we only consider the case when $b_r = 0$. $\mathcal{C}$ first checks whether C was ever output by $\mathsf{HO}(\mathsf{ID}_s, \mathsf{ID}_r, H, M)$ for some ID_s and M, and returns (ID_s, M) if so; Otherwise, for each KDF oracle query of the form $KDF(PS, X\|\mathsf{ID}_r)$ made by $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$, $\mathcal{C}$ checks if $PS = \mathsf{BDH}(X, h(\mathsf{ID}_r), g^c)$ with the aid of the DBDH oracle $\mathcal{O}_{\mathsf{DBDH}}$. If so, $\mathcal{C}$ gets $K_1 = KDF(PS, X\|\mathsf{ID}_r)$, and uses K_1 to decrypt C_{AE} and returns the result to $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$. In all the other cases, $\mathcal{C}$ simply returns "$\perp$" to $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$.

  Let $\mathsf{Event}_{\mathsf{F}}$ be the event that on the query of $\mathsf{UHO}(\mathsf{ID}_r, C = (H, X, C_{AE}))$ by $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$, where the corresponding value $b_r = 0$, $\mathcal{C}$ returns $\perp$ while C is actually a valid IBHigncryptext. Conditioned on $\mathsf{Event}_{\mathsf{F}}$ does not occur, the simulation for UHO is perfect. Below, we show that $\mathsf{Event}_{\mathsf{F}}$ can occur with at most negligible probability by the AE security.

  Suppose that the $\mathsf{Event}_{\mathsf{F}}$ event occurs w.r.t. $\mathsf{UHO}(\mathsf{ID}_r, C = (H, X, C_{AE}))$, where ID_r is the receiver whose corresponding value $b_r = 0$. For presentation simplicity, we refer to such a query as "failed UHO-query". We have the following observations:

  Fact-1: C was not the output of $\mathsf{HO}(\mathsf{ID}_s, \mathsf{ID}_r, H, M)$ for the given H and ID_r whose corresponding value $b_r = 0$, and for arbitrary ID_s and M.

Fact-2: $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$ didn't query $KDF(PS, X\|\mathsf{ID}_r)$ for $PS = \mathsf{BDH}(X, h(\mathsf{ID}_r), g^c)$.

Fact-3: (H, C_{AE}) is a valid AE ciphertext w.r.t. $K_1 = KDF(PS = \mathsf{BDH}(X,\ h(\mathsf{ID}_r),\ g^c),\ X\|\mathsf{ID}_r)$.

$\mathsf{Event_F}$ can be further divided into the following two cases:

1. K_1 was set by $\mathcal{C}$ uniformly at random for a query of the form $\mathsf{HO}(\mathsf{ID}'_s, \mathsf{ID}_r, H', M')$ when $b_s = b_r = 0$. Suppose that the $\mathsf{IBHigncryptext}$ output, when dealing with this HO query, is $C' = (H',\ X',\ C'_{\mathsf{AE}})$, where $C'_{\mathsf{AE}} \leftarrow \mathsf{Enc}_{K_1}(H',\ \mathsf{ID}'_s\|M'\|x')$.
   It means that $K_1 = KDF(PS = \mathsf{BDH}(X, h(\mathsf{ID}_r), g^c), X\|\mathsf{ID}_r) = KDF(PS' = \mathsf{BDH}(X', h(\mathsf{ID}_r), g^c), X'\|\mathsf{ID}_r)$. As KDF is a random oracle, with probability at least $1 - \frac{q^2_{kdf}}{2|\mathcal{K}|}$ we have $X' = X$, where q_{kdf} is the number of oracle queries made by $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$ to KDF. Then, by Fact-1, we have $(H', C'_{AE}) \neq (H, C_{AE})$, where (H, C_{AE}) is in the $\mathsf{failed\ UHO\text{-}query}$. It means that $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$ has output a new valid AE ciphertext (H', C'_{AE}) with respect to K_1. Assume that the underlying AE scheme is (t, ϵ_{AE}) secure, where t is the running time of $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$. Consequently, the event $\mathsf{Event_F}$ can occur with at most negligible probability (specifically, ϵ_{AE}) by the AE security. In total, the event $\mathsf{Event_F}$ can occur with probability at most $\epsilon_{AE} + \frac{q^2_{kdf}}{2|\mathcal{K}|}$ in this case.
2. Otherwise, K_1 was neither set by $\mathcal{C}$ nor ever defined for the KDF oracle. Hence, the event $\mathsf{Event_F}$ can also occur with probability at most ϵ_{AE} in this case by the AE security.

Now, we conclude that the $\mathsf{Event_F}$ event can occur with probability $P_{fail} \leq \epsilon_{AE} + \frac{q^2_{kdf}}{|\mathcal{K}|}$. And conditioned on that $\mathsf{Event_F}$ does not occur, and on that the challenger $\mathcal{C}$ does not abort in handling the $\mathsf{CORRUPT}$ queries, the view of $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$ in the simulation is the same as that in its real attack experiment.

$\mathsf{Phase\ 3}$: $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$ outputs $(\mathsf{ID}_{r^*}, C^* = (H^*, X^*, C^*_{AE}))$ as its forgery. If the forgery (ID_{r^*}, C^*) is a valid $\mathsf{IBHigncryptext}$, it must satisfy the following conditions simultaneously:

1. $\mathsf{UnIBHigncrypt}(sk_{r^*}, \mathsf{ID}_{r^*}, C^*) = (\mathsf{ID}_{s^*}, M^*, x^*)$, where $x \in Z^*_q$ and $X^* = h(\mathsf{ID}_{s^*})^{x^*}$.
2. ID_{s^*} or ID_{r^*} are both uncorrupted.
3. If there is any $\mathsf{HO}(\mathsf{ID}_{s^*}, \mathsf{ID}_{r^*}, H^*, M^*)$ query made by $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$ in $\mathsf{Phase\ 2}$, C^* must not be the output of that query.

Here, in order to solve the Gap-SBDH problem, we additionally require that for the uncorrupted target users ID_{s^*} and ID_{r^*} their corresponding values be $b_{s^*} = b_{r^*} = 0$; Otherwise, $\mathcal{C}$ aborts. By the AE security, except for some negligible probability at most ϵ_{AE}, $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$ must have made a KDF query on $(PS^*, X^* \| \mathsf{ID}_{r^*})$, where X^* may be generated by the adversary itself. By looking up the table $\mathsf{K}_{\mathsf{KDF}}$, $\mathcal{C}$ gets K_1^* corresponding to $(PS^*, X^* \| \mathsf{ID}_{r^*})$. Then, it UnIBHigncrypts C^* by using K_1^* to get $(\mathsf{ID}_{s^*}, M^*, x^*)$. Finally, $\mathcal{C}$ computes $e(g,g)^{a^2c} = (PS^*)^{\frac{1}{y_{s^*} y_{r^*} x^*}} = e(X^*, sk_{r^*})^{\frac{1}{y_{s^*} y_{r^*} x^*}} = e(h(\mathsf{ID}_{s^*})^{x^*}, h(\mathsf{ID}_{r^*})^c)^{\frac{1}{y_{s^*} y_{r^*} x^*}} = e((g^a)^{y_{s^*} x^*}, (g^a)^{y_{r^*}})^{\frac{c}{y_{s^*} y_{r^*} x^*}}$.

Remark 4 *For the case that the target sender and the target receiver are the same, we denote by* ID_* *the target user. In this case,* $h(\mathsf{ID}_{s^*}) = h(\mathsf{ID}_{r^*}) = h(\mathsf{ID}_*) = (g^a)^{y_*}$, $PS^* = e(sk_*, h(\mathsf{ID}_*))^{x^*} = e(g,g)^{a^2 c y_*^2 x^*}$. *It is obvious that the security can be reduced to the* Gap-SBDH *assumption: on input* $(g, g^a, g^c) \in \mathbb{G}_1^3$, *the challenger* $\mathcal{C}$ *computes* $e(g,g)^{a^2c} = (PS^*)^{\frac{1}{y_*^2 x^*}}$.

Now, we calculate the probability that the challenger $\mathcal{C}$ aborts because of simulation failure in dealing with oracle queries to CORRUPT or of the unexpected case $b_{s^*} = 1 \vee b_{r^*} = 1$. Denote by q_{corr} the number of queries made by $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$ to oracle CORRUPT. The total probability that $\mathcal{C}$ does not abort is $(1-\gamma)^2 \gamma^{q_{corr}}$, which is maximized to be $P_{\neg abort} = \frac{4}{(2+q_{corr})^2 e^2}$ at $\gamma = \frac{q_{corr}}{q_{corr}+2}$.

Remark 5 *For the case of* $\mathsf{ID}_{r^*} = \mathsf{ID}_{s^*}$, *the probability that* $\mathcal{C}$ *does not abort is* $(1-\gamma)\gamma^{q_{corr}}$, *which is maximized to be* $P_{\neg abort} = \frac{1}{(1+q_{corr})e}$ *at* $\gamma = \frac{q_{corr}}{q_{corr}+1}$.

Suppose that the adversary $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$'s running time is t, and can break the outsider unforgeability of IBHC with non-negligible probability ϵ. Then, the challenger $\mathcal{C}$ can solve the Gap-SBDH problem with non-negligible probability at least $(1 - P_{fail}) \cdot (1 - \epsilon_{AE}) \cdot P_{\neg abort} \cdot \epsilon$; If t is polynomial time, so is the running time of $\mathcal{C}$. Up to now, we finish the proof of outsider unforgeability.

6.2 Proof of Insider Confidentiality

In this section, we prove Theorem 3 in detail.

At first, the challenger $\mathcal{C}$ takes a tuple $(\mathbb{G}_1 = g, g^a, g^c) \in \mathbb{G}_1^3$ and a paring $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_{\mathsf{T}}$ as its inputs, where g is a generator of $\mathbb{G}_1$ and $a, c \leftarrow \mathbb{Z}_q^*$ are actually unknown to $\mathcal{C}$. The goal of $\mathcal{C}$ is to compute $T = e(g,g)^{a^2c} \in \mathbb{G}_{\mathsf{T}}$ with the help of a DBDH oracle denoted by $\mathcal{O}_{\mathsf{DBDH}}$, i.e., to solve the Gap-SBDH problem as defined in Section 3. Towards this goal, the challenger $\mathcal{C}$ runs the adversary $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$ who is assumed to break the insider confidentiality of IBHC with non-negligible probability. During the simulation, the challenger $\mathcal{C}$ maintains four tables $\mathsf{T_h}, \mathsf{S_{corr}}, \mathsf{K_{KDF}}$, and $\mathsf{ST_C}$. They are all initialized to be empty.

Setup: The challenger $\mathcal{C}$ sets the public parameters $\mathsf{par} = (q, \mathbb{G}_1, \mathbb{G}_\mathsf{T}, e, g, h)$, where q is the prime order of $\mathbb{G}_1$ and $\mathbb{G}_\mathsf{T}$, and $h : \{0,1\}^* \to \mathbb{G}_1$ is a collision-resistant cryptographic hash function that is modelled as a random oracle. The challenger $\mathcal{C}$ defines the master secret key $\mathsf{msk} = c$ (note that both a and c are unknown to $\mathcal{C}$). Finally, $\mathcal{C}$ gives par to the adversary $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$.
Hash Query on $h : \{0,1\}^* \to \mathbb{G}_1$: On input of a user's identity ID_i, the challenger $\mathcal{C}$ chooses a random $y_i \leftarrow \mathbb{Z}_q^*$. Using the techniques of Coron [23], $\mathcal{C}$ flips a biased coin $b_i \in \{0,1\}$ satisfying $b_i = 1$ with probability γ and 0 otherwise. If $b_i = 1$, $\mathcal{C}$ sets $h(\mathsf{ID}_i) = g^{y_i}$; Otherwise, $\mathcal{C}$ sets $h(\mathsf{ID}_i) = (g^a)^{y_i}$. The challenger returns $h(\mathsf{ID}_i)$ to $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$, and stores $(\mathsf{ID}_i, b_i, y_i, h(\mathsf{ID}_i))$ into the table $\mathsf{T_h}$.
Phase 1: $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$ issues a number of queries adaptively, including CORRUPT, HO, EXO, and UHO. With respect to each kind of the queries, the challenger $\mathcal{C}$ responds to $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$ as following:

- CORRUPT Query:

  For a CORRUPT query on user ID_i, $\mathcal{C}$ first visits table $\mathsf{T_h}$. If $b_i = 1$, $\mathcal{C}$ returns $sk_i = h(\mathsf{ID}_i)^c = (g^c)^{y_i}$, and sets $\mathsf{S_{corr}} := \mathsf{S_{corr}} \bigcup \{\mathsf{ID}_i\}$. Otherwise, $\mathcal{C}$ aborts.

- HO Query:

  For an HO query on $(\mathsf{ID}_s, \mathsf{ID}_r, H, M)$, $\mathcal{C}$ first visits table $\mathsf{T_h}$, and gets the entries corresponding to ID_s and ID_r, i.e., $(\mathsf{ID}_s, b_s, y_s, h(\mathsf{ID}_s))$ and $(\mathsf{ID}_r, b_r, y_r, h(\mathsf{ID}_r))$. We further consider the following cases:

  1. $b_s = 1$

     ---

     the challenger $\mathcal{C}$ selects $x \leftarrow \mathbb{Z}_q^*$;
     sets $X = h(\mathsf{ID}_s)^x = (g^{y_s})^x$;
     if $b_r = 1$
     $\mathcal{C}$ computes
     $PS = e(sk_s, h(\mathsf{ID}_r))^x = e((g^c)^{y_s}, g^{y_r})^x$;
     $K_1 = KDF(PS, X \| \mathsf{ID}_r)$;
     else
     $\mathcal{C}$ computes
     $PS = e(sk_s, h(\mathsf{ID}_r))^x = e((g^c)^{y_s}, (g^a)^{y_r})^x$;
     $K_1 = KDF(PS, X \| \mathsf{ID}_r)$;
     endif
     stores the tuple $((PS, X \| \mathsf{ID}_r), K_1))$ into $\mathsf{K_{KDF}}$;
     computes $C_{\mathsf{AE}} \leftarrow \mathsf{Enc}_{K_1}(H, \mathsf{ID}_s \| M \| x)$;
     returns $C = (H, X, C_{\mathsf{AE}})$ to $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$;
     stores the tuple (C, x) into the table $\mathsf{ST_C}$.

     ---

2. $b_s = 0$

---

the challenger $\mathcal{C}$ selects $x \leftarrow \mathbb{Z}_q^*$;
sets $X = h(\mathsf{ID}_s)^x = (g^a)^{x \cdot y_s}$;
if $b_r = 1$
$\mathcal{C}$ computes
$PS = e(sk_s, h(\mathsf{ID}_r))^x = e((g^c)^{y_s}, (g^a)^{y_r})^x$;
$K_1 = KDF(PS, X \| \mathsf{ID}_r)$;
stores the tuple $((PS, X \| \mathsf{ID}_r), K_1))$ into $\mathsf{K_{KDF}}$;
else
$\mathcal{C}$ sets K_1 to be a string taken uniformly at random from $\mathcal{K}$ of AEAD;
stores the tuple $((\star, X \| \mathsf{ID}_r), K_1)$ into $\mathsf{K_{KDF}}$;
endif
computes $C_{\mathsf{AE}} \leftarrow \mathsf{Enc}_{K_1}(H, \mathsf{ID}_s \| M \| x)$;
returns $C = (H, X, C_{\mathsf{AE}})$ to $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$;
stores the tuple (C, x) into the table $\mathsf{ST_C}$.

Note that for the above HO queries, if $b_s \neq 0$ or $b_r \neq 0$, the simulation of $\mathcal{C}$ is perfect by the properties of IBHigncryption. However, if $b_s = b_r = 0$, the challenger $\mathcal{C}$ cannot compute the pre-shared secret:

$$PS = e(sk_s, h(\mathsf{ID}_r))^x = \mathsf{BDH}(X, h(\mathsf{ID}_r), g^c),$$

and consequently $KDF(PS, X \| \mathsf{ID}_r)$. Whenever $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$ makes an oracle of the form $KDF(PS', X' || \mathsf{ID}'_r)$ for some ID'_r with $b'_r = 0$, such that KDF has not been defined over $(PS', X' || \mathsf{ID}'_r)$ but $(X' || \mathsf{ID}'_r = X || \mathsf{ID}_r)$ for some entry $((\star, X \| \mathsf{ID}_r), K_1)$ in $\mathsf{K_{KDF}}$, $\mathcal{C}$ does the following to ensure simulation consistency. It first makes a DBDH oracle query $\mathcal{O}_{\mathsf{DBDH}}(X', h(\mathsf{ID}'_r), g^c, PS')$, and acts as follows:

- If $\mathcal{O}_{\mathsf{DBDH}}$ answers 1, $\mathcal{C}$ returns to $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$ the value K_1 already stored in $((\star, X \| \mathsf{ID}_r), K_1)$; meanwhile, $\mathcal{C}$ updates the table $\mathsf{K_{KDF}}$ by replacing $\star$ with PS'.
- If $\mathcal{O}_{\mathsf{DBDH}}$ returns 0, $\mathcal{C}$ takes K'_1 uniformly at random from $\mathcal{K}$ of AE, returns K'_1 to $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$, and stores $((PS', X' \| \mathsf{ID}'_r), K'_1)$ into $\mathsf{K_{KDF}}$.

- EXO Query:

  For an EXO query on C, the challenger $\mathcal{C}$ visits the table $\mathsf{ST_C}$. If there is an entry in the table, $\mathcal{C}$ returns the corresponding x to the adversary. Otherwise, $\mathcal{C}$ returns $\perp$ to the adversary.

- UHO Query:

  For a UHO query on $(\mathsf{ID}_r, C = (H, X, C_{AE}))$: If ID_r's corresponding value $b_r = 1$, $\mathcal{C}$ can perfectly handle this query. Therefore, we only consider the case when $b_r = 0$. In this case, the challenger $\mathcal{C}$ does what he does in the proof of outsider unforgeability with respect to $b_r = 0$. The simulation analysis is also identical to that in the proof of Theorem 2. In particular, denote by P_{fail} the probability that the $\mathsf{Event_F}$ event occurs in the simulation of UHO queries. As shown in the proof of Theorem 2, conditioned on that $\mathsf{Event_F}$ does not occur, and on that the challenger $\mathcal{C}$ does not abort in handling the CORRUPT queries, up to now the view of $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$ in the simulation is the same as that in its real attack experiment.

  Challenge: At the end of phase 1, $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$ selects two target senders $\mathsf{ID}_{s_0^*}$, $\mathsf{ID}_{s_1^*}$, and a target receiver $\mathsf{ID}_{r^*} \in \{0,1\}^*$, a pair of messages (M_0^*, M_1^*) of equal length from $\{0,1\}^*$, and the associated data $H^* \in \{0,1\}^*$. $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$ submits $(M_0^*, M_1^*), H^*$, and $(\mathsf{ID}_{s_0^*}, \mathsf{ID}_{s_1^*}, \mathsf{ID}_{r^*})$ to the challenger $\mathcal{C}$, where $\mathsf{ID}_{r^*} \notin \mathsf{S_{corr}}$. If $b_{r^*} = 1$, the challenger $\mathcal{C}$ aborts; Otherwise, $\mathcal{C}$ does the following:

  1. Choose $\sigma \leftarrow \{0,1\}$. Here, $\mathsf{ID}_{s_\sigma^*}$ may be equal to ID_{r^*};
  2. If $b_{s_\sigma^*} = 0$, $\mathcal{C}$ chooses $x^* \leftarrow \mathbb{Z}_q^*$, and computes $X^* = h(\mathsf{ID}_{s_\sigma^*})^{x^*} = (g^a)^{y_{s_\sigma^*} x^*}$;
  3. Otherwise (i.e., $b_{s_\sigma^*} = 1$), $\mathcal{C}$ chooses $x^* \leftarrow \mathbb{Z}_q^*$, and sets $\bar{x}^* = x^* a$ (which is actually unknown to $\mathcal{C}$), and computes $X^* = h(\mathsf{ID}_{s_\sigma^*})^{\bar{x}^*} = (g^a)^{y_{s_\sigma^*} x^*}$;
  4. Check whether there is a record $((\star, X^* \| \mathsf{ID}_{r^*}), K_1^*)$ in the table $\mathsf{K_{KDF}}$ for arbitrary $K_1^* \in \mathcal{K}$. If yes, it outputs "collision" and aborts, which is referred to as the "collision" event. As X^* is distributed uniformly at random over $\mathbb{G}_1 \setminus \{1_{\mathbb{G}_1}\}$, where $1_{\mathbb{G}_1}$ is the identity element of $\mathbb{G}_1$, the "collision" event occurs with probability $P_{collision} \leq \frac{q_{kdf}^2}{2|\mathbb{G}_1 \setminus \{1_{\mathbb{G}_1}\}|} = \frac{q_{kdf}^2}{2(q-1)}$. Otherwise, the challenger chooses K_1 uniformly at random from the key space $\mathcal{K}$ of AE, and stores the tuple $((\star, X^* \| \mathsf{ID}_{r^*}), K_1^*)$ into the table $\mathsf{K_{KDF}}$;
  5. If $b_{s_\sigma^*} = 0$, $\mathcal{C}$ computes $C_{AE}^* = \mathsf{Enc}_{K_1}(H^*, \mathsf{ID}_{s_\sigma^*} \| M_\sigma^* \| x^*)$; otherwise, $\mathcal{C}$ selects $\hat{x}^* \leftarrow \mathbb{Z}_q^*$, and computes $C_{AE}^* = \mathsf{Enc}_{K_1}(H^*, \mathsf{ID}_{s_\sigma^*} \| M_\sigma^* \| \hat{x}^*)$;
  6. Return (H^*, X^*, C_{AE}^*) to $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$ as the challenge IBHigncryptext. From this point on, whenever $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$ makes an query of the form $KDF(PS^*, X^* \| \mathsf{ID}_{r^*})$ for arbitrary $PS^* \in \mathbb{G}_\mathsf{T}$, the challenger checks with its $\mathcal{O}_{\mathsf{DBDH}}$ oracle whether $PS^* = DBDH(X^*, h(\mathsf{ID}_{r^*}), g^c)$. If $\mathcal{O}_{\mathsf{DBDH}}(X^*, h(\mathsf{ID}_{r^*}), g^c, PS^*)$ returns "1", then $PS^* = e(X^*, sk_{r^*}) = e(g^{a y_{s_\sigma^*} x^*}, g^{acy_{r^*}})$. In this case, $\mathcal{C}$ returns the pre-defined

key K_1^* to $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$; Meanwhile, $\mathcal{C}$ replaces the "$\star$" in $((\star, X^* \| \mathsf{ID}_{r^*}), K_1)$ with PS^* in the table $\mathsf{K}_{\mathsf{KDF}}$.

Phase 2: $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$ continues to make queries as in phase 1 with the following restrictions:

1. $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$ is not allowed to issue a UHO query with the form $\mathsf{UHO}(\mathsf{ID}_{r^*}, C^*)$.
2. $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$ is not allowed to issue an EXO query on C^*.
3. $\mathsf{CORRUPT}(\mathsf{ID}_{r^*})$ is not allowed. But $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$ is allowed to issue $\mathsf{CORRUPT}(\mathsf{ID}_i)$ for any $\mathsf{ID}_i \neq \mathsf{ID}_{r^*}$.

Guess: Finally, $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$ outputs $\sigma' \in \{0, 1\}$ as its guess of the random bit σ.

Suppose that the adversary $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$ outputs $\sigma' = \sigma$ with non-negligible probability ε over $\frac{1}{2}$. By the AE security, in the random oracle model it must have made the KDF query on $(PS^*, X^* \| \mathsf{ID}_{r^*})$; Consequently, $\mathcal{C}$ gets $PS^* = e(X^*, sk_{r^*}) = \mathsf{BDH}(X^*, h(\mathsf{ID}_{r^*}), g^c) = e(g, g)^{a^2 c y_{s_\sigma^*} y_{r^*} x^*}$, from which it computes $e(g, g)^{a^2 c} = (PS^*)^{\frac{1}{y_{s_\sigma^*} y_{r^*} x^*}}$.

Remark 6 *Consider the case that the target sender, chosen by $\mathcal{C}$ for generating the challenge* IBHigncryptext*, is identical to the target receiver. For this case, we have $h(\mathsf{ID}_{s_\sigma^*}) = h(\mathsf{ID}_{r^*}) = (g^a)^{y_{r^*}}$. It is obvious that the security is also reduced to the* Gap-SBDH *assumption, where $PS^* = e(g, g)^{a^2 c y_{r^*}^2 x^*}$.*

Now, we analyze the probability that the challenger $\mathcal{C}$ aborts due to oracle queries to $\mathsf{CORRUPT}$ or to the unexpected value of b_{r^*}. Suppose that $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$ makes q_{corr} $\mathsf{CORRUPT}$ oracles, the probability $\mathcal{C}$ does not abort is $\gamma^{q_{corr}}$. Also note that when the adversary $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$ submits $(M_0^*, M_1^*), H^*$, and $(\mathsf{ID}_{s_0^*}, \mathsf{ID}_{s_1^*}, \mathsf{ID}_{r^*})$, $\mathcal{C}$ aborts if $b_{r^*} = 1$. So, the total probability that $\mathcal{C}$ does not abort is $(1 - \gamma)\gamma^{q_{corr}}$, which is maximized to be $P_{\neg abort} = \frac{1}{e(1+q_{corr})}$ at $\gamma = \frac{q_{corr}}{q_{corr}+1}$. This probability is independent of whether $\mathsf{ID}_{s_\sigma^*} = \mathsf{ID}_{r^*}$ or not.

Suppose that the adversary $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$'s running time is t, and can break the insider confidentiality of IBHC with non-negligible probability ϵ over $\frac{1}{2}$. Then, the challenger $\mathcal{C}$ can solve the $\mathsf{Gap\text{-}SBDH}$ problem with non-negligible probability at least $(1-P_{fail})\cdot(1-P_{collision})\cdot(1-\epsilon_{AE})\cdot P_{\neg abort}\cdot\epsilon$ (where P_{fail} and ϵ_{AE} are defined as in the proof of Theorem 2). If t is polynomial time, so is the running time of $\mathcal{C}$. This finishes the proof of insider confidentiality.

7 Identity-based Identity-Concealed Authenticaketed Key-Exchange(IB-CAKE)

Authentication Key Exchange (AKE), especially Diffie-Hellman (DH), plays an important role in modern cryptography and serves as a bridge between public-key cryptography and symmetric cryptography, as well as

the core mechanism of the network security protocol. Compared with the key exchange protocol under the traditional public-key cryptosystem, the identity-based key exchange protocol uses the identity of a user as its public key so that the management and distribution of public key certificates are simplified. However, the existing secure identity-based key agreement protocols need to transmit the user's identity and public key information publicly, and are not efficient enough. In the era of mobile internet, the computing and storage capabilities of devices are limited, in many applications, the user's identity is often considered to be sensitive information which should be protected during communications. With this explanation, designing of an efficient identity-based identity hiding key agreement protocol has important theoretical and practical significance. In this section, we will construct efficient identity-based identity hiding authenticated key agreement protocols in three types of bilinear groups.

Let n be a secure parameter, $\mathbb{G}_1$ and $\mathbb{G}_\mathsf{T}$ be two multiplicative bilinear map groups of the same prime order q such that the discrete logarithm problems in $\mathbb{G}_1$ and $\mathbb{G}_\mathsf{T}$ are intractable, and $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_\mathsf{T}$ be a bilinear pairing over $\mathbb{G}_1$ and $\mathbb{G}_\mathsf{T}$. Denote by $1_{\mathbb{G}_1}$ and $1_{\mathbb{G}_\mathsf{T}}$ the identity elements of $\mathbb{G}_1$ and $\mathbb{G}_\mathsf{T}$, by $\mathbb{G}_1/1_{\mathbb{G}_\mathsf{T}}$ the set of elements of $\mathbb{G}_1$ except $1_{\mathbb{G}_\mathsf{T}}$. Let $\mathsf{SE} = (\mathsf{K_{se}}, \mathsf{E}, \mathsf{D})$ be an authenticated encryption with associated data (AEAD) scheme [45], $h : \{0,1\}^* \to \mathbb{G}_1$ be a one-way collision-resistant cryptographic hash function, and $\mathsf{KDF} : \{0,1\}^* \to \{0,1\}^{p(n)}$ be a key derivation function, where $p(n)$ is a polynomial of n. For presentation simplicity, we denote by Alice the anonymous session initiator, whose public identity and private key are ID_A and $SK_A = (h(\mathsf{ID}_A))^{msk}$, and by Bob the session responder, whose public identity and private key are ID_B and $SK_A = (h(\mathsf{ID}_B))^{msk}$, where msk is the master secret key of PKG. The protocol structure of IB-CAKE using type-I pairing is depicted in Fig. 3

We note that the above IB-CAKE is constructed in the symmetric pairing (type-I) setting, where the bilinear map $\hat{e}$ is defined over $\mathbb{G}_1$ and $\mathbb{G}_T$, i.e., $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$. In practice, using asymmetric bilinear groups (type-II and type-III) is most practical for paring implementations, where $\hat{e}$ is defined as $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$.

Similar to our construction of IBHigncryption, an additional efficient publicly computable isomorphism ψ is required for our IB-CAKE protocol with type-II bilinear pairing. The isomorphism ψ is for the purpose of mapping an element from $\mathbb{G}_1$ to $\mathbb{G}_2$. For the construction of our IB-CAKE protocol with type-III bilinear pairing, the private key sk of any user ID is replaced by a pair of key (sk^I, sk^R), where sk^I is used when the user is an initiator in a session, and sk^R is used when the user is a responder in a session. These two protocols are depicted in Fig. 4 and Fig. 5, respectively.

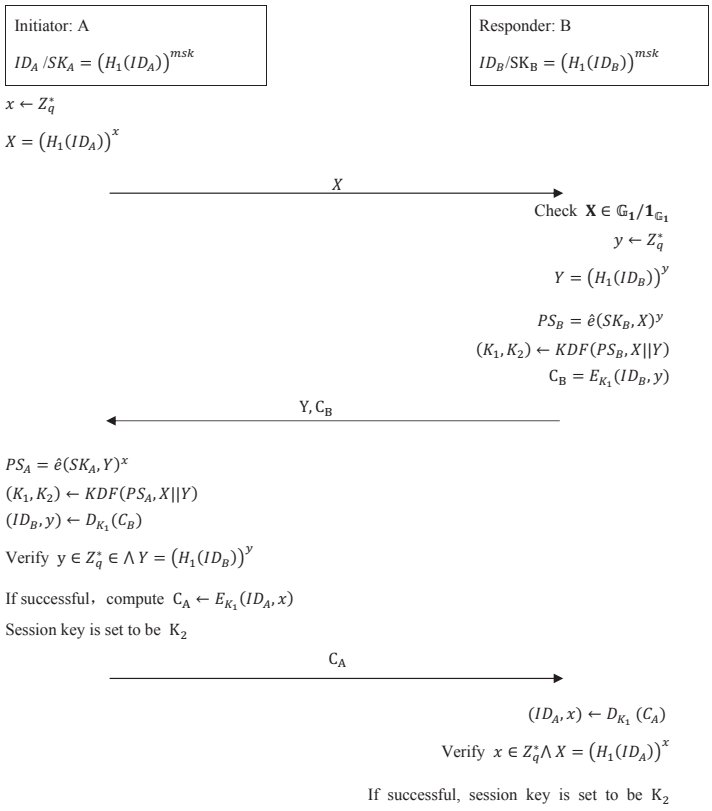

Figure 3: Construction of IB-CAKE with Type-I Bilinear Mapping

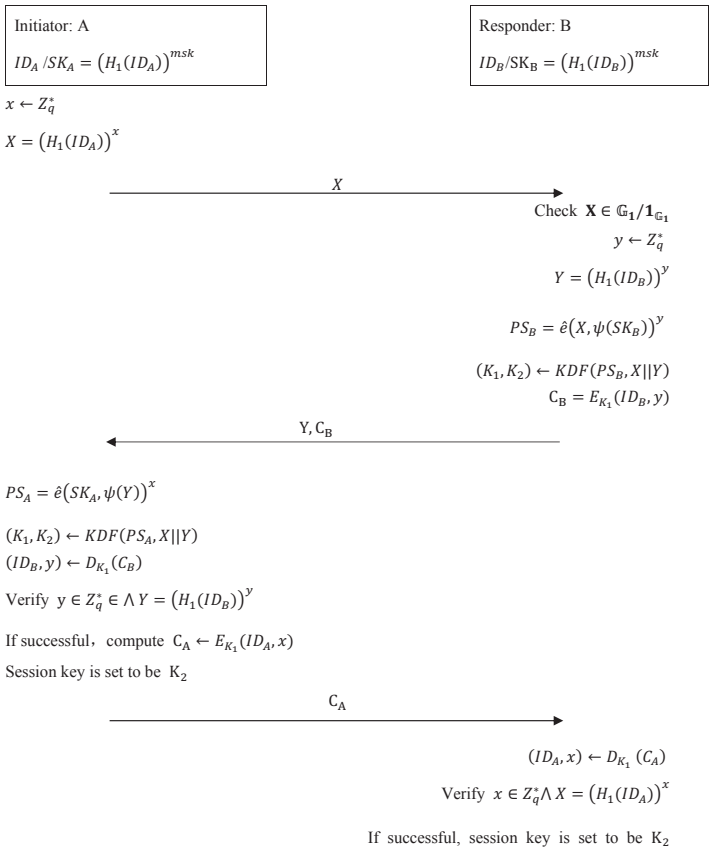

Figure 4: Construction of IB-CAKE with Type-II Bilinear Mapping

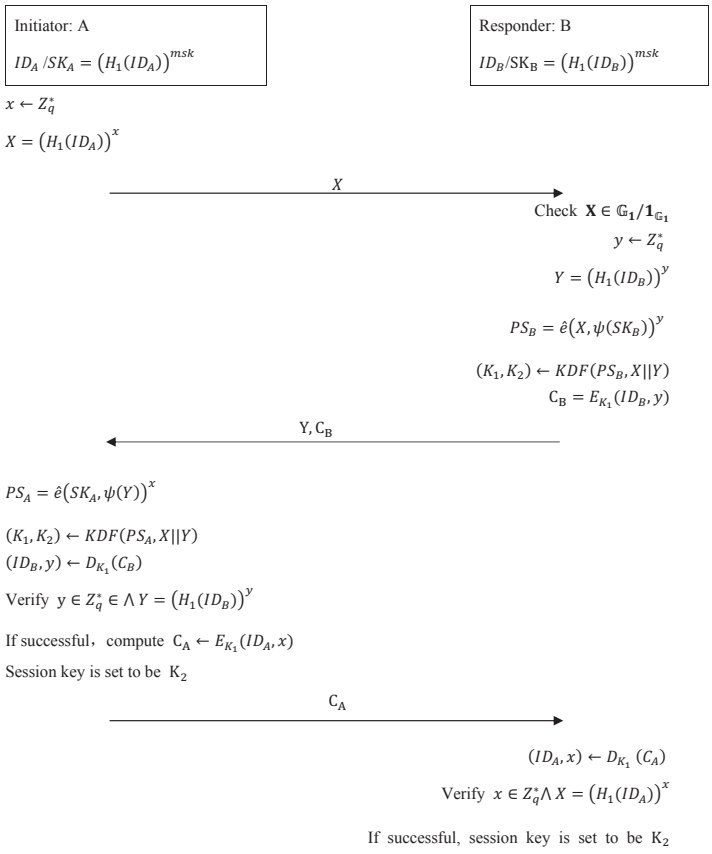

Figure 5: Construction of IB-CAKE with Type-III Bilinear Mapping

References

[1] 3GPP TS 33.180 v15.3.0 (2018-09),3rd Generation Partnership Project: 3G Security; Security Architecture (3GPP TS 33.102 Version 15.0.0 Release 15)

[2] 3GPP TS 33.180 v15.3.0 (2018-09),3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Security of the mission critical service; (Release 15)

[3] 3GPP TS 33.220 v15.3.0 (2018-09),3rd Generation Partnership Project; Techincal Specification Group Services and System Aspect; Generic authentication Architecture (GAA); Generic Bootstrapping Architecture (GBA) (Release 15)

[4] Voltage identity-based encryption–information encryption for email, files, documents and databases, https://www.voltage.com/technology/data-encryption/identity-based-encryption/

[5] Arkko, J., Carrara, E., Lindholm, F., Naslund, M., Norrman, K.: Mikey: Multimedia internet keying. RFC **3830**, pp. 1–66 (2004)

[6] Baek, J., Safavi-Naini, R., Susilo, W.: Efficient multi-receiver identity-based encryption and its application to broadcast encryption. In: Public Key Cryptography - PKC 2005, pp. 380–397 (2005)

[7] Barbulescu, R.: A brief history of pairings. In: Arithmetic of Finite Fields - 6th International Workshop, WAIFI 2016, pp. 3–17 (2016)

[8] Barbulescu1, R., Duquesne, S.: Updating key size estimations for pairings. Journal of Cryptology, to appear

[9] Barreto, P.S.L.M., Libert, B., McCullagh, N., Quisquater, J.: Efficient and provably-secure identity-based signatures and signcryption from bilinear maps. In: Advances in Cryptology - ASIACRYPT 2005, pp. 515–532 (2005)

[10] Barthe, G., Fagerholm, E., Fiore, D., Scedrov, A., Schmidt, B., Tibouchi, M.: Strongly-optimal structure preserving signatures from type II pairings: Synthesis and lower bounds. In: Katz, J. (ed.) Public-Key Cryptography - PKC 2015 , volume 9020 of LNCS, pp. 355–376. Springer (2015)

[11] Bellare, M., Namprempre, C.: Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In: Okamoto, T. (ed.) Advances in Cryptology - ASIACRYPT 2000, volume 1976 of LNCS, pp. 531–545. Springer (2000)

[12] Bellare, M., Namprempre, C.: Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. J. Cryptology **21**(4), 469–491 (2008)

[13] M. Bellare, and P. Rogaway. Random Oracles Are Practical: A Paradigm for Designing Efficient Protocols. *ACM CCS* 1993: 62-73.

[14] Blazy, O., Germouty, P., Phan, D.H.: Downgradable identity-based encryption and applications. In: Matsui, M. (ed.) Topics in Cryptology - CT-RSA 2019 ,volume 11405 of LNCS, pp. 44–61. Springer (2019)

[15] Boneh, D., Boyen, X.: Efficient selective-id secure identity based encryption without random oracles. In: Proceedings of Eurocrypt 2004, volume 3027 of LNCS, pp. 223–238. Springer-Verlag (2004)

[16] Boneh, D., Franklin, M.K.: Identity-based encryption from the weil pairing. In: Advances in Cryptology - CRYPTO 2001, pp. 213–229 (2001)

[17] Boneh, D., Lynn, B., Shacham, H.: Short signatures from the weil pairing. In: Boyd, C. (ed.) Advances in Cryptology - ASIACRYPT 2001, volume 2248 of LNCS, pp. 514–532. Springer (2001)

[18] Boyen, X., Martin, L.: Identity-based cryptography standard (IBCS) #1: Supersingular curve implementations of the BF and BB1 cryptosystems. RFC **5091**, 1–63 (2007)

[19] Brzuska, C., Smart, N.P., Warinschi, B., Watson, G.J.: An analysis of the EMV channel establishment protocol. In: Sadeghi, A., Gligor, V.D., Yung, M. (eds.) 2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13 , pp. 373–386. ACM (2013)

[20] Chatterjee, S., Menezes, A.: On cryptographic protocols employing asymmetric pairings - the role of psi revisited. IACR Cryptology ePrint Archive **2009**, 480 (2009) http://eprint.iacr.org/2009/480

[21] Chatterjee, S., Menezes, A.: On cryptographic protocols employing asymmetric pairings - the role of Ψ revisited. Discrete Applied Mathematics **159**(13), 1311–1322 (2011)

[22] Chen, J., Lim, H.W., Ling, S., Wang, H., Wee, H.: Shorter IBE and signatures via asymmetric pairings. In: Abdalla, M., Lange, T. (eds.) Pairing-Based Cryptography - Pairing 2012 , volume 7708 of LNCS, pp. 122–140. Springer (2012)

[23] Cowell, S.R., Beiu, V., Daus, L., Poulin, P.: On the exact reliability enhancements of small hammock networks. IEEE Access **6**, 25411–25426 (2018)

[24] Fauzi, P., Lipmaa, H., Siim, J., Zajac, M.: An efficient pairing-based shuffle argument. In: Takagi, T., Peyrin, T. (eds.) Advances in Cryptology - ASIACRYPT 2017 , volume 10625 of LNCS, pp. 97–127. Springer (2017)

[25] Feige, U., Fiat, A., Shamir, A.: Zero-knowledge proofs of identity. Journal of Cryptology **1**(2), 77–94 (1988)

[26] Fiat, A., Shamir, A.: How to prove yourself: practical solutions to identification and signature problems. In: Proceedings on Advances in cryptology—CRYPTO '86. pp. 186–194 (1986)

[27] Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. Journal of Cryptology **26**(1), 80–101 (2013)

[28] Galbraith, S.D., Paterson, K.G., Smart, N.P.: Pairings for cryptographers. Discrete Applied Mathematics **156**(16), 3113–3121 (2008)

[29] Groves, M.: Elliptic curve-based certificateless signatures for identity-based encryption (ECCSI). RFC **6507**, 1–17 (2012)

[30] Groves, M.: MIKEY-SAKKE: sakai-kasahara key encryption in multimedia internet keying (MIKEY). RFC **6509**, 1–21 (2012)

[31] Groves, M.: Sakai-kasahara key encryption (SAKKE). RFC **6508**, 1–21 (2012)

[32] Halevi, S., Krawczyk, H.: One-pass HMQV and asymmetric key-wrapping. In: Public Key Cryptography - PKC 2011 , pp. 317–334 (2011)

[33] Hofheinz, D., Jia, D., Pan, J.: Identity-based encryption tightly secure under chosen-ciphertext attacks. In: Peyrin, T., Galbraith, S.D. (eds.) Advances in Cryptology - ASIACRYPT 2018 , volume 11273 of LNCS, pp. 190–220. Springer (2018)

[34] Ishida, Y., Watanabe, Y., Shikata, J.: Constructions of cca-secure revocable identity-based encryption. In: Foo, E., Stebila, D. (eds.) Information Security and Privacy - 20th Australasian Conference, ACISP 2015, volume 9144 of LNCS, pp. 174–191. Springer (2015)

[35] Khan, H., Dowling, B., Martin, K.M.: Identity confidentiality in 5g mobile telephony systems. IACR Cryptology ePrint Archive **2018**, 876 (2018), https://eprint.iacr.org/2018/876

[36] Krawczyk, H., Paterson, K.G., Wee, H.: On the security of the TLS protocol: A systematic analysis. In: Canetti, R., Garay, J.A. (eds.) Advances in Cryptology - CRYPTO 2013 , volume 8042 of LNCS, pp. 429–448. Springer (2013)

[37] Lair, Y., Mayer, G.: Mission critical services in 3GPP. IEEE Spectrum, October **6507**, 1–195 (2018)

[38] Langrehr, R., Pan, J.: Tightly secure hierarchical identity-based encryption. In: PKC 2019, to appear

[39] Libert, B., Quisquater, J.: Identity based undeniable signatures. In: Okamoto, T. (ed.) Topics in Cryptology - CT-RSA 2004, , volume 2964 of LNCS, pp. 112–125. Springer (2004)

[40] Paterson, K.G., Ristenpart, T., Shrimpton, T.: Tag size does matter: Attacks and proofs for the TLS record protocol. In: Lee, D.H., Wang, X. (eds.) Advances in Cryptology - ASIACRYPT 2011 , volume 7073 of LNCS, pp. 372–389. Springer (2011)

[41] Paterson, K.G., Srinivasan, S.: On the relations between non-interactive key distribution, identity-based encryption and trapdoor discrete log groups. Des. Codes Cryptography **52**(2), 219–241 (2009)

[42] Raimondo, M.D., Gennaro, R., Krawczyk, H.: Deniable authentication and key exchange. In: Juels, A., Wright, R.N., di Vimercati, S.D.C. (eds.) Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006, pp. 400–409. ACM (2006)

[43] Ramanna, S.C., Chatterjee, S., Sarkar, P.: Variants of waters' dual system primitives using asymmetric pairings - (extended abstract). In: Fischlin, M., Buchmann, J.A., Manulis, M. (eds.) Public Key Cryptography - PKC 2012 , volume 7293 of LNCS, pp. 298–315. Springer (2012)

[44] Rescorla, E.: The transport layer security (TLS) protocol version 1.3, draft-12. https://tools.ietf.org/html/draft-ietf-tls-tls-12

[45] Rogaway, P.: Authenticated-encryption with associated-data. In: Proceedings of the 9th ACM Conference on Computer and Communications Security, CCS 2002 , pp. 98–107 (2002)

[46] Roskind, J.: Quick UDP internet connections: Multiplexed stream transport over UDP. https://tools.ietf.org/html/draft-ietf-tls-tls-12 **1**(2), 77–94 (2012)

[47] Sakai, R., Ohgishi, K., Kasahara, M.: Cryptosystem based on pairings. In: Symposium on Cryptography and Information Security(SCIS), pp. 26–28 (2000)

[48] Shamir, A.: Identity-based cryptosystems and signature schemes. In: Advances in Cryptology, Proceedings of CRYPTO '84 , pp. 47–53 (1984)

[49] Zhao, C., Zhang, F.: Research and development on efficient pairing computations. Journal of Software **20**(11), 3001–3009 (2009)

[50] Zhao, Y.: Identity-concealed authenticated encryption and key exchange. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security , pp. 1464–1479 (2016)

[51] Zheng, Y.: Digital signcryption or how to achieve cost(signature & encryption) << cost(signature) + cost(encryption). In: Advances in Cryptology - CRYPTO '97 , pp. 165–179 (1997)

A IBHigncryption Constructions with Asymmetric Bilinear Pairings

In this part, we describe our IBHigncryption constructions based on bilinear pairings of Type 2 and Type 3, respectively.

A.1 Construction with Bilinear Pairings of Type 2

The construction of our IBHigncryption in this section, as well as the IEEE P1363.3 standard [9] for ID-Based signcryption, is based on asymmetric bilinear parings of Type 2. The extension of our IBHigncryption construction to the Type 2 bilinear parings is straightforward, which is described below from scratch for ease of reference.

- $\mathsf{Setup}(1^\kappa)$: On input of the security parameter κ, the algorithm chooses three multiplicative bilinear map groups $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_\mathsf{T}$ of the same prime order q, generators $g_1 \in \mathbb{G}_1$, $g_2 = \psi(g_1) \in \mathbb{G}_2$, and a bilinear paring $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_\mathsf{T}$ such that the discrete logarithm problems in $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_\mathsf{T}$ are intractable, where $\psi : \mathbb{G}_1 \to \mathbb{G}_2$ is an efficient, publicly computable isomorphism. The algorithm chooses a master secret key $s \leftarrow \mathbb{Z}_q^*$. Additionally, it selects a one-way collision-resistant cryptographic hash function, $h : \{0,1\}^* \to \mathbb{G}_1$. Finally, the algorithm outputs the public parameters $\mathsf{par} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\mathsf{T}, e, g_1, g_2, \psi, h)$, and the PKG's master secret key $\mathsf{msk} = s$. The PKG makes par public to the users in the system, but keeps msk secret for itself.

- $\mathsf{KeyGen}(\mathsf{par}, \mathsf{msk}, \mathsf{ID})$: On input of the system's public parameters par, the master secret key msk of the PKG, and a user's identity $\mathsf{ID} \in \{0,1\}^*$, the PKG computes $sk = h(\mathsf{ID})^{\mathsf{msk}} = h(\mathsf{ID})^s$, and outputs sk as the private key associated with identity ID.

- $\mathsf{IBHigncrypt}(\mathsf{par}, sk_s, \mathsf{ID}_s, \mathsf{ID}_r, H, M)$: Let $\mathsf{SE} = (\mathsf{K_{se}}, \mathsf{Enc}, \mathsf{Dec})$ be an authenticated encryption scheme, $M \in \{0,1\}^*$ be the message to be IBHigncrypted with associated data $H \in \{0,1\}^*$, and $\mathsf{KDF} : \mathbb{G}_\mathsf{T} \times$

$\{0,1\}^* \rightarrow \{0,1\}^*$ be a key derivation function, where $\mathcal{K}$ is the key space of $\mathsf{K_{se}}$. For presentation simplicity, we denote by ID_s the sender's public identity whose private key is $sk_s = h(\mathsf{ID}_s)^s$, and by ID_r the receiver's public identity whose private key is $sk_r = h(\mathsf{ID}_r)^s$.

To IBHigncrypt a message $M \leftarrow \{0,1\}^*$ with the sender's identity ID_s concealed, the sender: (1) selects $x \leftarrow \mathbb{Z}_q^*$, and computes $X = h(\mathsf{ID}_s)^x \in \mathbb{G}_1$; (2) computes the pre-shared secret $PS = e(sk_s, \psi(h(\mathsf{ID}_r)))^x$; (3) derives $K_1 = KDF(PS, X \| \mathsf{ID}_r) \in \mathcal{K}$; (4) computes $C_{AE} \leftarrow \mathsf{Enc}_{\mathsf{K}_1}(H, \mathsf{ID}_s \| M \| x)$; and finally (5) sends the $\mathsf{IBHigncryptext}$ $C = (H, X, C_{AE})$ to the receiver ID_r.

- $\mathsf{UnIBHigncrypt}(\mathsf{par}, sk_r, \mathsf{ID}_r, C)$: Upon receiving $C = (H,\ X,\ C_{AE})$, the receiver: (1) computes the pre-shared secret $PS = e(X, \psi(sk_r)) \in \mathbb{G}_\mathsf{T}$, and derives the key $K_1 = KDF(PS, X \| \mathsf{ID}_r) \in \mathcal{K}$; (2) runs $\mathsf{Dec}_{K_1}(H, C_{AE})$. If $\mathsf{Dec}_{K_1}(H, C_{AE})$ returns $\perp$, it aborts; Otherwise, the receiver gets $\{\mathsf{ID}_s, M, x\}$, and outputs (ID_s, M) if $x \in Z_q^*$ and $X = h(\mathsf{ID}_s)^x$; Otherwise, it outputs "$\perp$" and aborts.

A.2 Construction with Bilinear Pairings of Type 3

The construction of our $\mathsf{IBHigncryption}$ in this subsection is based on the bilinear parings of Type 3.

- $\mathsf{Setup}(1^\kappa)$: On input of the security parameter κ, the algorithm chooses three multiplicative bilinear map groups $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_\mathsf{T}$ of the same prime order q, generators $g_1 \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$, and a bilinear paring $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_\mathsf{T}$ such that the discrete logarithm problems in $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_\mathsf{T}$ are intractable. The algorithm chooses a master secret key $s \leftarrow \mathbb{Z}_q^*$. Additionally, it selects two one-way collision-resistant cryptographic hash functions, $h_1 : \{0,1\}^* \rightarrow \mathbb{G}_1$, and $h_2 : \{0,1\}^* \rightarrow \mathbb{G}_2$. Finally, the algorithm outputs the public parameters $\mathsf{par} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\mathsf{T}, e, g_1, g_2, h_1, h_2)$, and the PKG's master secret key $\mathsf{msk} = s$. The PKG makes par public to the users in the system, but keeps msk secret for itself.

- $\mathsf{KeyGen}(\mathsf{par}, \mathsf{msk}, \mathsf{ID})$: On input of the system's public parameters par, and a user's identity $\mathsf{ID} \in \{0,1\}^*$, the PKG computes $sk = (sk_1, sk_2) = (h_1(\mathsf{ID})^s, h_2(\mathsf{ID})^s)$, and outputs sk as the private key associated with identity ID.

- $\mathsf{IBHigncrypt}(\mathsf{par}, sk_s = (sk_{s_1}, sk_{s_2}), \mathsf{ID}_s, \mathsf{ID}_r, H, M)$: Let $\mathsf{SE} = (\mathsf{K_{se}}, \mathsf{Enc}, \mathsf{Dec})$ be an authenticated encryption scheme, $M \in \{0,1\}^*$ be the message to be $\mathsf{IBHigncrypted}$ with associated data $H \in \{0,1\}^*$, and $\mathsf{KDF} : \mathbb{G}_\mathsf{T} \times \{0,1\}^* \rightarrow \{0,1\}^*$ be a key derivation function, where $\mathcal{K}$ is the key space of $\mathsf{K_{se}}$. For presentation simplicity, we denote by ID_s

the sender's public identity whose private key is $sk_s = (sk_{s_1}, sk_{s_2}) = (h_1(\mathsf{ID}_s)^s, h_2(\mathsf{ID}_s)^s)$, and by ID_r the receiver's public identity whose private key is $sk_r = (sk_{r_1}, sk_{r_2}) = (h_1(\mathsf{ID}_r)^s, h_2(\mathsf{ID}_r)^s)$.

To IBHigncrypt a message $M \leftarrow \{0,1\}^*$ with the sender's identity ID_s concealed, the sender: (1) selects $x \leftarrow \mathbb{Z}_q^*$, and computes $X = h_1(\mathsf{ID}_s)^x \in \mathbb{G}_1$; (2) computes the pre-shared secret $PS = e(sk_{s_1}, h_2(\mathsf{ID}_r))^x$; (3) derives $K_1 = KDF(PS, X \| \mathsf{ID}_r) \in \mathcal{K}$; (4) computes $C_{AE} \leftarrow \mathsf{Enc}_{\mathsf{K}_1}(H, \mathsf{ID}_s \| M \| x)$; and finally (5) sends the $\mathsf{IBHigncryptext}$ $C = (H, X, C_{AE})$ to the receiver ID_r.

- $\mathsf{UnIBHigncrypt}(\mathsf{par}, sk_r = (sk_{r_1}, sk_{r_2}), \mathsf{ID}_r, C)$: On receiving $C = (H, X, C_{AE})$, the receiver: (1) computes the pre-shared secret $PS = e(X, sk_{r_2}) \in \mathbb{G}_\mathsf{T}$, and derives the key $K_1 = KDF(PS, X \| \mathsf{ID}_r) \in \mathcal{K}$; (2) runs $\mathsf{Dec}_{K_1}(H, C_{AE})$. If $\mathsf{Dec}_{K_1}(H, C_{AE})$ returns $\perp$, it aborts; Otherwise, the receiver gets $\{\mathsf{ID}_s, M, x\}$, and outputs (ID_s, M) if $x \in Z_q^*$ and $X = h_1(\mathsf{ID}_s)^x$; Otherwise, it outputs "$\perp$" and aborts.

Remark 7 *For presentation simplicity, the above Type 3 pairing based implementation of* $\mathsf{IBHigncryption}$ *is described w.r.t. a pair of secret keys* (sk_1, sk_2) *for each user in the system. But from the protocol description, it is clear that: if a user only performs the role of sender (resp., receiver), it only needs a single secret key* sk_1 *(resp.,* sk_2*).*

B CCA-Secure Boneh-Franklin IBE

The identity-based encryption from Weil paring [16] (referred to as BF-IBE for simplicity) is the first practical identity-based encryption from pairing. In [16], both a CPA-secure IBE, and a CCA-secure IBE via the Fujisaki-Okamoto transformation [27], are proposed. Below, we briefly review the CCA-secure BF-IBE construction.

The CCA-secure BF-IBE scheme consists of the following four algorithms:

- Setup: Given a security parameter $\kappa \in \mathbb{Z}^+$, this algorithm: (1) generates a prime q, two bilinear map groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of order q, and an admissible bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$; (2) chooses a random generator $g \in \mathbb{G}_1$; (3) picks $s \leftarrow \mathbb{Z}_q^*$ and sets the master public key $P_{\mathsf{pub}} = g^s$; (4) chooses a cryptographic hash function $h_1 : \{0,1\}^* \rightarrow \mathbb{G}_1$, and three cryptographic hash functions $h_2 : \mathbb{G}_2 \rightarrow \{0,1\}^n, h_3 : \{0,1\}^n \times \{0,1\}^n \rightarrow \mathbb{Z}_q^*$, and $h_4 : \{0,1\}^n \rightarrow \{0,1\}^n$ for some n. The message space is $\mathcal{M} = \{0,1\}^n$, and the ciphertext space is $\mathcal{C} = \mathbb{G}_1 \times \{0,1\}^n \times \{0,1\}^n$. The system parameters are

$$\mathsf{par} = (q, \mathbb{G}_1, \mathbb{G}_2, e, n, g, P_{\mathsf{pub}}, h_1, h_2, h_3, h_4),$$

and the master secret key is $s \in \mathbb{Z}_q^*$.

- KeyGen: For a given string $\mathsf{ID} \in \{0,1\}^*$, this algorithm: (1) computes $Q_{\mathsf{ID}} = h_1(\mathsf{ID}) \in \mathbb{G}_1$, and (2) sets the private key $sk_{\mathsf{ID}} = Q_{\mathsf{ID}}^s$, where $s \in Z_q^*$ is the master secret key.

- Enc: To encrypt a message $M \in \{0,1\}^n$ under the public key ID, this algorithm: (1) computes $Q_{\mathsf{ID}} = h_1(\mathsf{ID}) \in \mathbb{G}_1$; (2) chooses a random $\sigma \leftarrow \{0,1\}^n$; (3) sets $r = h_3(\sigma, M)$; and (4) sets the ciphertext as:

$$C = (g^r, \sigma \oplus h_2(g_{\mathsf{ID}}^r), M \oplus h_4(\sigma)),$$

  where $g_{\mathsf{ID}} = e(Q_{\mathsf{ID}}, P_{pub}) \in \mathbb{G}_2$.

- Dec: Let $C = (U, V, W)$ be a ciphertext encrypted using the public key ID. If $U \notin \mathbb{G}_1$, this algorithm rejects the ciphertext; Otherwise, it decrypts C using the private $sk_{\mathsf{ID}} \in \mathbb{G}_1$:

  1. compute $V \oplus h_2(e(sk_{\mathsf{ID}}, U)) = \sigma$;
  2. compute $W \oplus h_4(\sigma) = M$;
  3. set $r = h_3(\sigma, M)$. Test whether $U = g^r$. If not, the algorithm rejects the ciphertext;
  4. Otherwise, the algorithm outputs M as the decryption of C.

C IEEE P1363.3 ID-Based Signcryption

The identity-based signcryption from Type 2 bilinear maps [9], adopted as IEEE P1363 standard, consists of the following algorithms.

- Setup: Given a security parameter κ, the PKG chooses bilinear map groups $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_{\mathsf{T}})$ of prime order $q > 2^\kappa$, an admissible bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_{\mathsf{T}}$; and generators $g_2 \in \mathbb{G}_2, g_1 = \psi(g_2) \in \mathbb{G}_1, g = e(g_1, g_2) \in \mathbb{G}_{\mathsf{T}}$, where $\psi : \mathbb{G}_2 \to \mathbb{G}_1$ is an efficient, publicly computable (but not necessarily invertible) isomorphism such that $\psi(g_2) = g_1$. It then chooses a master secret key $s \leftarrow \mathbb{Z}_q^*$, computes a system-wide master public key $Q_{\mathsf{pub}} = g_2^s \in \mathbb{G}_2$, and chooses hash functions $h_1 : \{0,1\}^* \to \mathbb{Z}_q^*$, $h_2 : \{0,1\}^* \times \mathbb{G}_{\mathsf{T}} \to \mathbb{Z}_q^*$, and $h_3 : \mathbb{G}_{\mathsf{T}} \to \{0,1\}^n$. The public parameters are

$$\mathsf{par} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_{\mathsf{T}}, g_1, g_2, g, Q_{\mathsf{pub}}, e, \psi, h_1, h_2, h_3),$$

  and the master secret key is $s \in \mathbb{Z}_q^*$.

- KeyGen: For a given string $\mathsf{ID} \in \{0,1\}^*$, this algorithm computes the private key $sk_{\mathsf{ID}} = g_2^{\frac{1}{h_1(\mathsf{ID})+s}} \in \mathbb{G}_2$.

- Sign/Encrypt: Given a message $M \in \{0,1\}^n$, a receiver's identity $\mathsf{ID_B}$ and a sender's private key $sk_{\mathsf{ID_A}}$, the algorithm:

  1. picks $x \leftarrow \mathbb{Z}_q^*$, computes $r = g^x$, and $C = M \oplus h_3(r) \in \{0,1\}^n$;
  2. sets $u = h_2(M, r) \in \mathbb{Z}_q^*$;
  3. computes $S = \psi(sk_{\mathsf{ID_A}})^{x+u}$;
  4. computes $T = (g_1^{h_1(\mathsf{ID_B})} \cdot \psi(Q_{\mathsf{pub}}))^x$.

  The ciphertext is $\sigma = (C, S, T) \in \{0,1\}^n \times \mathbb{G}_1 \times \mathbb{G}_1$.

- Decrypt/Verify: Give $\sigma = (C, S, T)$, and some sender's identity $\mathsf{ID_A}$, the receiver:

  1. computes $r = e(T, sk_{\mathsf{ID_B}})$, $M = C \oplus h_3(r)$, and $u = h_2(M, r)$;
  2. accepts the message if and only if $r = e(S, g_2^{h_1(\mathsf{ID_A})} \cdot Q_{\mathsf{pub}})g^{-u}$. If this condition holds, returns the message M and the signature $(u, S) \in \mathbb{Z}_q^* \times \mathbb{G}_1$.