Identity-Based Higncryption

Hongbing Wang and Yunlei Zhao
Fudan University
Email: ylzhao@fudan.edu.cn

March 10, 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, identity concealment means that the transcript of protocol run should not leak participants' identity information.

In this work, we propose the first identity-based higncryption (IBHigncryption, for short). We present formal security model for IBHigncryption, under which security proof of the proposed scheme is conducted. The most impressive feature of IBHigncryption, besides other desirable properties it offers, is its simplicity and efficiency, which might be somewhat surprising in retrospect. Our IBHigncryption has a much simpler setup stage with smaller public parameters and particularly no need of computing master public key. It is essentially as efficient as (if not more than) the fundamental CCA-secure Boneh-Franklin identity-based encryption scheme [8], and has significant efficiency advantage over the IEEE 1363.3 standard of identity-based signcryption [6].

1 Introduction

Identity-based cryptography (IBC) was proposed by Shamir in 1984 [29], with the motivation to simplify certificate management in traditional public-key cryptography. In an identity-based (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. ID-based signature schemes appear much earlier [15, 14], however, the first practical and fully functional identity-based encryption scheme was only proposed by Boneh and Franklin [8] in 2001 based on bilinear maps. The Boneh-Franklin's IBE scheme is further standardized with ISO/IEC

18033-5 and IETF RFC 5091 [9], and is now widely deployed (e.g., in HPE Secure Data by Voltage security [4]).[1]

The concept of signcryption was proposed by Zheng [31]. 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 sequential composition). 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 in ISO 29150, and a pairing-based ID-based signcryption scheme [6] was adopted as IEEE P1363.3 standard.

With signcryption, the sender's identity information has to be exposed, as otherwise, the ciphertext cannot be decrypted and the message cannot be verified. However, identity is a fundamental privacy concern, and identity confidentiality is now mandated by a list of prominent standards such as TLS1.3 [26], QUIC [28], EMV [10], and the 5G telecommunication standard [2] by 3GPP (the 3rd Generation Partnership Projec), etc. Under this motivation, Zhao [30] introduced a new cryptographic primitive called identity-hiding signcryption (higncryption, for short). 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 run 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 [20],[2] in the sense that the session transcript can be simulated from the public parameters and the receiver's secret-key.
- x-security [20], 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 secrecy (from which session-key is derived).

We note that the work in [30] only considered higncryption in the traditional public-key setting. In this work, we study identity-based highcryption and its applications.

[1]The HPE IBE (including BF01 [8] and BB1 [7]) 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.

[2]The formal definitions of receiver deniability and the following x-security are quite straightforward, and are referred to [20] for presentation simplicity.

1.1 Motivation and Application Scenarios

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), 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 [23]. In June 2018, 3GPP has identified the following essential requirements related to user privacy [1, 22] 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] 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 [21]. This IB-AKT protocol involves the *sequential* composition of an identity-based encryption scheme (i.e., SAKKE specified in IETF RFC 6508 [19] and 6509 [18]) and an identity-based signature scheme (i.e., ECCSI specified in IETF RFC 6507 [17]). In MIKEY-SAKKE, the user's identity ID takes the form of a constrained "tel" URI, in front of "tel" URI is a monthly-updated timestamp for refreshing the key of the user periodically. It also provides a mechanism with identity hiding, but this mechanism is too simple. Concretely, in MIKEY-SAKKE with identity hiding, a user's URI is replaced by its $UID = H(Key, S)$, which is generated by hashing the user's related strings [3]. Further, UID shall be used as the identifer within MIKEY-SAKKE with identity hiding. 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, signcryption may be a candidate for the service. We note that there already

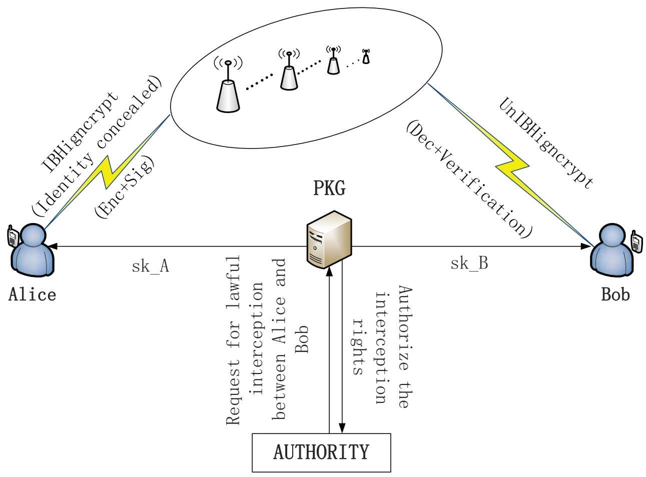


Figure 1: IBHigncryption's Application in 4G-LTE

has been IEEE P1363.3 standard for ID-based signcryption [6]. However, as mentioned ahead, the sender's identity information has to be exposed with signcryption. In this sense, ID-based identity-concealed signcryption takes place. Moreover, for enhancing privacy and strengthening security, forward ID-privacy, receiver deniability, and x-security are all desirable in such settings. This is just our motivation for developing ID-based identity-concealed signcryption (IBHigncryption).

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 (P-KG) 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 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. 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 session lawfully.

1.2 Our Contribution

In this work, we propose the first identity-based higncryption (IBHigncryption, for short). We present formal security model for IBHigncryption, under which the security proof of the proposed scheme is conducted. 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. Specifically, 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 implementation of our IBHigncryption is provided, with source code available from Github.

The proposed IBHigncryption scheme is essentially as efficient as (if not more than) the fundamental CCA-secure Boneh-Franklin IBE scheme [8], while offering entity authentication and identity concealment simultaneously. Compared to the identity-based signcryption scheme [6], which is adopted as IEEE P1363.3 standard, our generalized construction of IBHigncryption (when implemented on asymmetric bilinear groups) is much simpler, and has significant efficiency advantage in total (particularly on the receiver side). Besides, our generalized 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.

2 Preliminaries

2.1 Notations

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 $x\|y$ is the concatenation of two elements $x, y \in \{0,1\}^*$.

2.2 Bilinear Pairing

Bilinear pairings were first introduced by Weil in 1946 as a computationally efficient bilinear mapping on algebraic curves (i.e Weil pairings). It is a very important concept and tool in algebraic geometry, especially in algebraic curve theory. Bilinear pairings can be widely used in designing cryptographic protocols, for example, ID-based encryption, key exchange protocol, short signature, signature with special properties, attribute-based encryption(ABE), predicate encryption (PE), function encryption (FE), searchable encryption (SE), etc.

2.3 Definitions and Hard Problems

Definition 1 (Bilinear Paring) *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 efficient computable.*

Generally, there are three types of bilinear pairing [25]:

1. Type 1: $\mathbb{G}_1 = \mathbb{G}_2$, it is also called symmetric bilinear pairing.
2. Type 2: There is an efficiently computable isomorphism either from $\mathbb{G}_1$ to $\mathbb{G}_2$ or from $\mathbb{G}_2$ to $\mathbb{G}_1$.
3. Type 3: There is no efficiently computable isomorphisms between $\mathbb{G}_1$ and $\mathbb{G}_2$.

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. The computationally *intractable* problems considered in this work are defined as follows.

Definition 2 (Bilinear Diffie-Hellman (BDH)) *The bilinear Diffie-Hellman (BDH) problem [24] 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 $a, b, c \leftarrow \mathbb{Z}_q^*$.*

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 $e(g,g)^{a^2b} \in \mathbb{G}_\mathsf{T}$, given $(g, g^a, g^b) \in \mathbb{G}_1^3$, where $a, b \leftarrow \mathbb{Z}_q^*$.*

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

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

Proof 1 $\mathsf{BDH} \Longrightarrow \mathsf{SBDH}$*:*

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 an 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.*

$\mathsf{SBDH} \Longrightarrow \mathsf{BDH}$*:*

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. Then, there must exist an algorithm* $\mathcal{A}_2$*, which, on input* $(g, g^a, g^b, g^c) \in \mathbb{G}_1^4$*, outputs* $e(g,g)^{abc} \in \mathbb{G}_\mathsf{T}$ *with the same probability. The algorithm* $\mathcal{A}_2$ *chooses* $r, s, t \leftarrow \mathbb{Z}_q^*$*, and computes* $u_1 = \mathcal{O}_2(g, (g^a)^r, (g^c)^t) = e(g,g)^{a^2cr^2t}$*,* $u_2 = \mathcal{O}_2(g, (g^b)^s, (g^c)^t) = e(g,g)^{b^2cs^2t}$*. Finally,* $\mathcal{A}_2$ *computes* $v = \mathcal{O}_2(g, (g^a)^r \cdot (g^b)^s, (g^c)^t) = e(g,g)^{(ar+bs)^2 \cdot ct} = e(g,g)^{a^2cr^2t + b^2cs^2t + 2abcrst}$*. Since* r, s, t *are known already, it follows that* $e(g,g)^{abc}$ *can be computed from* r, s, t *immediately with the same advantage.*

Definition 4 (Gap Bilinear Diffie-Hellman (Gap-BDH)) *The gap bilinear Diffie-Hellman (*Gap-BDH*) problem [24, 5] 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* $T = e(g,g)^{abc}$*.*

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* $T = e(g,g)^{a'b'c'}$*.*

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

2.4 Authenticated Encryption

Briefly speaking, an *authenticated encryption with associated data* (AEAD) scheme transforms a message M and a public header information H (e.g., a packet header, an IP address) into a ciphertext C in such a way that C provides both privacy (of M) and authenticity (of C and H) [27]. In practice, when AEAD is used within cryptographic systems, the associated data is usually implicitly determined from the context (e.g., the hash of the transcript of protocol run or some pre-determined states).

main $\text{AEAD}_{\mathsf{SE}}^{\mathcal{A}}$:	**proc.** $\mathsf{Enc}(H, M_0, M_1)$:	**proc.** $\mathsf{Dec}(C')$:
$K \leftarrow \mathcal{K}_{\text{se}}$	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 $\mathsf{Dec}_K(C')$
$\sigma' = \mathcal{A}^{\mathbf{Enc},\mathbf{Dec}}$	$C_1 \leftarrow \mathsf{Enc}(H, M_1)$	Ret $\perp$
Ret $(\sigma' = \sigma)$	If $C_0 = \perp$ or $C_1 = \perp$	
	Ret $\perp$	
	$\mathcal{C} \overset{\cup}{\leftarrow} C_\sigma$; Ret C_σ	

Table 1: AEAD security game

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 encryption algorithm $\mathsf{Enc} : \mathcal{K} \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^* \cup \{\perp\}$ and the (deterministic) polynomial-time decryption algorithm $\mathsf{Dec} : \mathcal{K} \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^* \cup \{\perp\}$ 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, we assume the ciphertext C bears the associated data H in plain.

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

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

We say that the SE scheme is AEAD-secure, if for any sufficiently large κ, the advantage of any probabilistic polynomial-time (PPT) algorithm adversary is negligible.

The above AEAD 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 ID-based Higncryption: Definition and Security Model

3.1 Definition of IBHigncryption

In an identity-based identity-concealed signcryption scheme ($\mathsf{IBHigncryption}$) (denoted by 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 is 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. *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 ID_s, a receiver's public identity ID_r, a message M and its associated data H to be* IBHigncrypted, *it outputs an* IBHigncryptext *C, 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 ID_r, and an* IBHigncryptext *C, it outputs (ID_s, M) if the verification is successful, or $\perp$ indicating an error, where ID_s is the sender's public identity, and M 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.*

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 $\mathsf{UnIBHigncrypt}(\mathsf{par}, sk_r, \mathsf{ID}_r, \mathsf{IBHigncrypt}(\mathsf{par}, sk_s, \mathsf{ID}_s, \mathsf{ID}_r, H, M)) = (\mathsf{ID}_s, M)$ for any $H, M \in \{0,1\}^*$ such that $\mathsf{IBHigncrypt}(\mathsf{par}, sk_s, \mathsf{ID}_s, \mathsf{ID}_r, H, M) \neq \perp$.

3.2 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 $\mathsf{UnIBHigncrypt}$, and the sender can $\mathsf{IBHigncrypt}$ 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. 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 in the underlying security model and protocol construction.

The security of an $\mathsf{IBHigncryption}$ 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 $\mathsf{IBHigncryptext}$ 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 $\mathsf{IBHigncryptext}$ 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 $\mathsf{IBHigncryptexts}$. Likewise, here the sender may be equal to the receiver.

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

- HO Oracle : This oracle is used to respond to the $\mathsf{IBHigncrypt}$ 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 ID_r may be equal to ID_s, 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 ST_C privately.

- UHO Oracle: This oracle is used to respond to the $\mathsf{UnIBHigncrypt}$ queries made by an adversary, including $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$ or $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$. 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 ID_r.

- EXO Oracle: This oracle is used to respond to the intermediate randomness used in generating an $\mathsf{IBHigncryptext}$ of an earlier HO query.

It is an additional oracle in our security model which makes our model more stronger than the security model for signcryption, and describes the x-security property of an IBHC protocol. On input an $\mathsf{IBHigncryptext}$ 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 ID_i, this oracle returns the private key sk_i of the user ID_i, and ID_i is 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 game for outsider unforgeability and insider confidentiality.

Definition 7 (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 $\mathsf{ID}_{r^*} \notin \mathsf{S_{corr}}$ 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 ID_{s^*} for an uncorrupted receiver ID_{r^*} if and only if the following conditions hold simultaneously:*

  1. $\mathsf{UnIBHigncrypt}(sk_{r^*}, \mathsf{ID}_{r^*}, C^*) = (\mathsf{ID}_{s^*}, M'^* = (M^*, x^*)$*, where $\mathsf{ID}_{s^*} \notin \mathsf{S_{corr}}$, $x^* \neq 0$, 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 any $\mathsf{HO}(\mathsf{ID}_{s'}, \mathsf{ID}_{r'}, H', M')$ for $(\mathsf{ID}_{s'}, \mathsf{ID}_{r'}, H', M') \neq (\mathsf{ID}_{s^*}, \mathsf{ID}_{r^*}, H^*, M^*)$. 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^*. Moreover, $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$ is allowed to issue an $\mathsf{EXO}(C^*)$ to expose the intermediate randomness used in generating C^*.*

Definition 8 *Let* $\mathsf{Adv}_{\mathsf{IBHC}}^{\mathcal{A}^{\mathsf{OU}}}$ *denote the advantage that an* $\mathcal{A}_{\mathsf{IBHC}}^{\mathsf{OU}}$ *adversary outputs a valid forgery in the above security game* $\mathsf{GAME}_{\mathsf{IBHC}}^{\mathcal{A}^{\mathsf{OU}}}$*. We say that an* IBHigncryption *scheme* IBHC *has outsider unforgeability, if for any* PPT *adversary* $\mathcal{A}_{\mathsf{IBHC}}^{\mathsf{OU}}$*, its advantage* $\mathsf{Adv}_{\mathsf{IBHC}}^{\mathcal{A}^{\mathsf{OU}}}$ *is negligible for any sufficiently large security parameter.*

Definition 9 (Insider Confidentiality (IC)) *Let* $\mathcal{A}_{\mathsf{IBHC}}^{\mathsf{IC}}$ *be an IC-adversary against* IBHC*. We consider the following game, denoted by* $\mathsf{GAME}_{\mathsf{IBHC}}^{\mathcal{A}^{\mathsf{IC}}}$*, in which an adversary* $\mathcal{A}_{\mathsf{IBHC}}^{\mathsf{IC}}$ *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* $\mathcal{A}_{\mathsf{IBHC}}^{\mathsf{IC}}$*, and keeps the* msk *secretly for itself.*

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

- Challenge*: At the end of* phase 1*,* $\mathcal{A}_{\mathsf{IBHC}}^{\mathsf{IC}}$ *selects 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^**.* $\mathcal{A}_{\mathsf{IBHC}}^{\mathsf{IC}}$ *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 $\mathcal{A}_{\mathsf{IBHC}}^{\mathsf{IC}}$*. 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* $\mathsf{ID}_{s_\sigma^*} = \mathsf{ID}_{r^*}$*).*

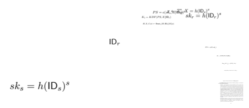
- Phase 2*:* $\mathcal{A}_{\mathsf{IBHC}}^{\mathsf{IC}}$ *continues to make queries as in* phase 1 *with the following restrictions:*

  1. $\mathcal{A}_{\mathsf{IBHC}}^{\mathsf{IC}}$ *is not allowed to issue* $\mathsf{CORRUPT}(\mathsf{ID}_{r^*})$*.*
  2. $\mathcal{A}_{\mathsf{IBHC}}^{\mathsf{IC}}$ *is not allowed to issue* $\mathsf{UHO}(\mathsf{ID}_{r^*}, C^*)$*.*
  3. $\mathcal{A}_{\mathsf{IBHC}}^{\mathsf{IC}}$ *is not allowed to issue* $\mathsf{EXO}(C^*)$*.*

- Guess*: Finally,* $\mathcal{A}_{\mathsf{IBHC}}^{\mathsf{IC}}$ *outputs* $\sigma' \in \{0,1\}$ *as his guess of the random bit* σ*.* $\mathcal{A}_{\mathsf{IBHC}}^{\mathsf{IC}}$ *wins the game if* $\sigma' = \sigma$*.*

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

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



Definition 10 *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.*

4 Construction of IBHigncryption

4.1 Construction with Symmetric Bilinear Pairing

- $\mathsf{IBHigncrypt}(\mathsf{par}, sk_s, \mathsf{ID}_s, \mathsf{ID}_r, H, M)$: Let $\mathsf{SE} = (\mathsf{K_{se}}, \mathsf{Enc}, \mathsf{Dec})$ be an authenticated encryption with associated data (AEAD) scheme [27], $M \in \{0,1\}^*$ be the message to be IBHigncrypted with associated data $H \in \{0,1\}^*$, and $\mathsf{KDF} : \mathbb{G}_\mathsf{T} \times \{0,1\}^* \to \{0,1\}^*$ be a key derivation function[3], 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, ID_s: (1) selects $x \leftarrow \mathbb{Z}_q^*$, and computes $X = h(\mathsf{ID}_s)^x \in \mathbb{G}_1$; (2) computes the pre-shared secrecy $PS = e(sk_s, h(\mathsf{ID}_r))^x \in \mathbb{G}_\mathsf{T}$; (3) derives 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 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: (1) computes the pre-shared secrecy $PS = e(X, sk_r) \in \mathbb{G}_\mathsf{T}$, 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 accepts (ID_s, M) if $X = h(\mathsf{ID}_s)^x$, and $x \neq 0$. Otherwise, it aborts.

Remark 1 *The correctness of the above* IBHigncryption *is straightforward. It enjoys x-security, because even if x is exposed, an adversary cannot compute the pre-shared secrecy PS without the private key of* ID_s *or* ID_r.

4.2 Generalized Construction with Asymmetric Bilinear Pairing

In this part, we describe our IBHigncryption constructions over bilinear pairing type 2 and type 3, respectively.

4.2.1 Construction with Bilinear Pairing of Type 2

The construction of our IBHigncryption in this section, as well as the IEEE P1363.3 standard [6] for ID-Based signcryption, is based on asymmetric bilinear paring type 2. The extension of our IBHigncryption construction to the bilinear paring type 2 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

[3] When implemented, KDF can be viewd as a random oracle. It also holds in our constructions with asymmetric bilinear pairing.

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, where $\psi : \mathbb{G}_1 \rightarrow \mathbb{G}_2$ 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, $h : \{0,1\}^* \rightarrow \mathbb{G}_1$. 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. Note that, here, the setup stage is also pretty simple, where in particular no modular exponentiation is performed in order to generate a traditional master public key.

- $\mathsf{KeyGen}(\mathsf{par}, \mathsf{msk}, \mathsf{ID})$: 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 with associated data (AEAD) scheme [27], $M \in \{0,1\}^*$ 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 $sk_s = h(\mathsf{ID}_s)^s$, 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 secrecy $PS = e(sk_s, \psi(h(\mathsf{ID}_r)))^x \in \mathbb{G}_\mathsf{T}$; (3) derives 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: (1) computes the pre-shared secrecy $PS = e(X, \psi(sk_r)) \in \mathbb{G}_\mathsf{T}$, 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 accepts (ID_s, M) if $X = h(\mathsf{ID}_s)^x$, and $x \neq 0$. Otherwise, it aborts.

4.2.2 Construction with Bilinear Pairing of Type 3

The construction of our $\mathsf{IBHigncryption}$ in this subsection is based on the bilinear paring type 3. In [11, 12], the authors argue that type 2 pairings are merely inefficient implementations of type 3 pairings, and appear to offer no benefit for protocols based on asymmetric pairings considering functionality, security, and performance. For this reason, we extend our IBHigncryption based on type 1 and type 2 pairing to type 3 pairing. This extension is easy to be implemented with little changes, and it is described as follows:

- $\mathsf{Setup}(1^\kappa)$: 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 pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_\mathsf{T}$ 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. Note that, here, the setup stage is also pretty simple, where in particular no modular exponentiation is performed in order to generate a traditional master public key.

- $\mathsf{KeyGen}(\mathsf{par}, \mathsf{msk} = s, \mathsf{ID})$: 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 with associated data (AEAD) scheme [27], $M \in \{0,1\}^*$ 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 $sk_s = (sk_{s_1}, sk_{s_2}) = (h_1(\mathsf{ID}_s)^s, h_2(\mathsf{ID}_s)^s)$, 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 secrecy $PS = e(sk_{s_1}, h_2(\mathsf{ID}_r))^x \in \mathbb{G}_\mathsf{T}$; (3) derives 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 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 ID_r: (1) computes the pre-shared secrecy $PS = e(X, sk_{r_2}) \in \mathbb{G}_\mathsf{T}$, 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 accepts (ID_s, M) if $X = h_1(\mathsf{ID}_s)^x$, and $x \neq 0$. Otherwise, it aborts.

Remark 2 *It is easy to observe that our construction based on Type 3 is efficient, since it is only at the cost of doubling the private key of each user, which needs one more exponent operation in $\mathbb{G}_2$. No matter comparing with BF01 [8], or IEEE P1363.3 standard [6], our constructions are comparable based u nder the same environment.*

5 Security Proof 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) which are controlled by the challenger.

Theorem 2 *The* IBHigncryption *scheme presented in Fig. 2 is outsider unforgeable in the random oracle model under the AEAD security and the* Gap-SBDH *assumption. Concretely, suppose that there exists a (t,ϵ)-adversary $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$ who can break outsider unforgeability of the* IBHigncryption *scheme with non-negligible advantage ϵ and running time t, then, there exists another (t',ϵ')-algorithm, which can solve the* Gap-SBDH *problem with non-negligible advantage $\epsilon' = \frac{4(1-1/q)\cdot\epsilon}{e^{(q_{corr}+2)\ln(q_{corr}+2)-q_{corr}\ln q_{corr}}}$ and running time $t' \leq t+(q_h+q_{kdf}+q_{dbdh})O(1)+q_{corr}\cdot t_e+q_{ho}(2t_e+1t_p+1t_{enc})+q_{uho}(1t_e+1t_p+1t_{dec})$, where $q_h, q_{kdf}, q_{corr}, q_{ho}$, q_{uho}, and q_{dbdh} are the adversary's query times to Hash, KDF, CORRUPT, HO, UHO, and DBDH oracles, and t_e, t_p, t_{enc} and t_{dec} represent the running time of an exponentiation, pairing, Enc, and Dec operation, respectively.*

Theorem 3 *The* IBHigncryption *scheme presented in Fig. 2 has insider confidentiality in the random oracle model under the AEAD security and the* Gap-SBDH *assumption. Concretely, suppose that there exists a (t,ϵ)-adversary $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$ who can break insider confidentiality of the* IBHigncryption *scheme with non-negligible advantage ϵ and running time t, then, there exists another (t',ϵ')-algorithm, which can solve the* Gap-SBDH *problem with non-negligible advantage $\epsilon' = \frac{(1-1/q)\cdot\epsilon}{e\cdot(q_{corr}+1)}$ and running time $t' \leq t + (q_h + q_{kdf} + q_{dbdh})O(1) + q_{corr}\cdot t_e + q_{ho}(2t_e + 1t_p + 1t_{enc}) + q_{uho}(1t_e + 1t_p + 1t_{dec})$, where $q_h, q_{kdf}, q_{corr}, q_{ho}$, q_{uho}, and q_{dbdh} are the adversary's query times to Hash, KDF, CORRUPT, HO, UHO, and DBDH oracles, and t_e, t_p, t_{enc} and t_{dec} represent the running time of an exponentiation, pairing, Enc, and Dec operation, respectively.*

5.1 Proof of Outsider Unforgeability

In this section, we prove Theorem 2 in detail.

At first, the challenger $\mathcal{C}$ accepts a tuple $(\mathbb{G}_1 = \langle g \rangle,\ g^a,\ g^c) \in \mathbb{G}_1^3$ and a paring $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_\mathsf{T}$ as inputs. 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 $\mathcal{O}_{\mathsf{DBDH}}$), which is regarded as the gap square bilinear Diffie-Hellman hard problem (Gap-SBDH) [24, 5], conditioned on that unforgeability of IBHC is broken with non-negligible probability by the adversary $\mathcal{A}_{\mathsf{IBHC}}^{\mathsf{OU}}$. The DBDH oracle $\mathcal{O}_{\mathsf{DBDH}}$ for $\mathbb{G}_1 = \langle g \rangle$ and $\mathbb{G}_\mathsf{T}$ on arbitrary input $(A' = g^{a'}, B' = g^{b'}, C = g^{c'}, Z) \in \mathbb{G}_1^3 \times \mathbb{G}_\mathsf{T}$, outputs 1 if and only if $Z = e(g,g)^{a'b'c'}$.

During the simulation, the challenger $\mathcal{C}$ maintains four tables $\mathsf{T_h}, \mathsf{K_{KDF}}, \mathsf{K_{DBDH}}$, and $\mathsf{ST_C}$. They are all initialized to be empty.

Phase 1: 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, which is modelled as a random oracle and controlled by $\mathcal{C}$ in our security proof. The challenger $\mathcal{C}$ defines the master secret key $\mathsf{msk} = c$, (where a, c are unknown to $\mathcal{C}$). Finally, $\mathcal{C}$ gives par to the adversary $\mathcal{A}_{\mathsf{IBHC}}^{\mathsf{OU}}$.

Hash Query on $h : \{0,1\}^* \to \mathbb{G}_1$:

On input of a user's identity ID_i, the challenger chooses a random $y_i \leftarrow \mathbb{Z}_q^*$. Using the techniques of Coron [13], $\mathcal{C}$ flips a biased coin $b_i \in \{0,1\}$ satisfying $b_i = 1$ with probability γ and 0 otherwise [13]. If $b_i = 1$, $\mathcal{C}$ sets $h(\mathsf{ID}_i) = g^{y_i}$. Otherwise, if $b_i = 0$, $\mathcal{C}$ sets $h(\mathsf{ID}_i) = (g^a)^{y_i}$. The challenger returns $h(\mathsf{ID}_i)$ to $\mathcal{A}_{\mathsf{IBHC}}^{\mathsf{OU}}$, and stores $(\mathsf{ID}_i, b_i, y_i, h(\mathsf{ID}_i))$ into the table $\mathsf{T_h}$.

Phase 2: $\mathcal{A}_{\mathsf{IBHC}}^{\mathsf{OU}}$ 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 $\mathcal{A}_{\mathsf{IBHC}}^{\mathsf{OU}}$ 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 $sk_i = h(\mathsf{ID}_i)^c = (g^c)^{y_i}$. Otherwise, $\mathcal{C}$ aborts. Let $\mathsf{S_{corr}}$ be the set of corrupted users in the system, which is initialized to be empty. On each CORRUPT query on ID_i, if the challenger $\mathcal{C}$ returns the private key of ID_i to the adversary , it sets $\mathsf{S_{corr}} := \mathsf{S_{corr}} \bigcup \{\mathsf{ID}_i\}$.

- HO Query:

  For an HO query on $(\mathsf{ID}_s, \mathsf{ID}_r, H, M)$, there is no restriction on H and M, which means that H can even be H^*, and M can even be M^*(here, H^* is the associated data in the adversary's forgery, and M^* is the message IBHigncrypted in the adversary's forgery). $\mathcal{C}$ first visits table $\mathsf{T_h}$, and get the values of 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
$PS = e(sk_s, h(\mathsf{ID}_r))^x = e((g^c)^{y_s}, (g^a)^{y_r})^x$;
$K_1 = KDF(PS, X \| \mathsf{ID}_r)$;
$\mathcal{C}$ stores the tuple $(X \| \mathsf{ID}_r, K_1)$ into $\mathsf{K}_{\mathsf{KDF}}$;
endif
$\mathcal{C}$ computes $C_{\mathsf{AE}} \leftarrow \mathsf{Enc}_{K_1}(H, \mathsf{ID}_s \| M \| x)$;
$\mathcal{C}$ returns $C = (H, X, C_{\mathsf{AE}})$ to $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$;
$\mathcal{C}$ 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
$PS = e(sk_s, h(\mathsf{ID}_r))^x = e((g^c)^{y_s}, (g^a)^{y_r})^x$;
$K_1 = KDF(PS, X \| \mathsf{ID}_r)$;
else
$\mathcal{C}$ sets K_1 to be a string taken uniformly at random from $\mathcal{K}$ of AEAD;
$\mathcal{C}$ stores the tuple $(X \| \mathsf{ID}_r, K_1)$ into $\mathsf{K}_{\mathsf{KDF}}$;
endif
$\mathcal{C}$ computes $C_{\mathsf{AE}} \leftarrow \mathsf{Enc}_{K_1}(H, \mathsf{ID}_s \| M \| x)$;
$\mathcal{C}$ returns $C = (H, X, C_{\mathsf{AE}})$ to $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$;
$\mathcal{C}$ stores the tuple (C, x) into the table ST_{C}.

- 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.

  Note that in the above HO queries, if $b_s \neq 0$, or $b_r \neq 0$, K_1 is derived based on the correctly computed PS, therefore, the simulation of $\mathcal{C}$ is perfect. If $b_s = b_r = 0$, though the challenger $\mathcal{C}$ cannot compute PS, X is computed correctly, and K_1 is set uniformly at random and can be used to correctly UnIBHigncrypt the output of the HO Query. Due to the fact that KDF is a random oracle, the simulation of $\mathcal{C}$ in this case is also perfect.

Also note that in the above cases, if $b_s = b_r = 0$, the challenger $\mathcal{C}$ cannot compute the pre-share secrecy

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

and consequently $KDF(PS,\ X\|\mathsf{ID}_r)$. In order to keep the consistency of the random oracle KDF, whenever the adversary $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$ makes an oracle of the form $KDF(PS', X\|\mathsf{ID}_r)$ for some ID_r whose corresponding value $b_r = 0$, based on the table $\mathsf{K_{KDF}}$ and $\mathsf{T_h}$, the challenger $\mathcal{C}$ checks whether $\mathcal{O}_{\mathsf{DBDH}}(X, h(\mathsf{ID}_r), g^c, PS')$ oracle returns 1, which implies $PS' = e(X, sk_r) = e(X, h(\mathsf{ID}_r)^c) = e(X,\ h(\mathsf{ID}_r))^c$; If yes, it returns the corresponding pre-shared key K_1 in the table $\mathsf{K_{KDF}}$ to the adversary, meanwhile, $\mathcal{C}$ stores the tuple $(X\|\mathsf{ID}_r, PS', K_1)$ into the table $\mathsf{K_{DBDH}}$.

So far, all the simulations for $\mathsf{CORRUPT}, \mathsf{HO}$, and EXO is perfect.

- UHO Query:

  For an 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 where $b_r = 0$. $\mathcal{C}$ first checks whether C was ever output by $\mathsf{HO}(\mathsf{ID}_s, \mathsf{ID}_r, H, M)$ for some $M \in \{0,1\}^*$ and ID_s, and outputs (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 there is a match in the table $\mathsf{K_{DBDH}}$. If so, $\mathcal{C}$ gets $K_1 = KDF(PS, X\|\mathsf{ID}_r)$, and uses K_1 to decrypt C_{AE}. The challenger $\mathcal{C}$ further verifies the decryption results. If the verification is successful, it returns the results to $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$; Otherwise, $\mathcal{C}$ returns $\perp$ indicating C is an invalid $\mathsf{IBHigncryptext}$ for user ID_r. Let $\mathsf{Event_F}$ be the event that on the query of the form $\mathsf{UHO}(\mathsf{ID}_r, C = (H, X, C_{AE}))$ by $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$, $\mathcal{C}$ returns $\perp$ while C is a valid $\mathsf{IBHigncryptext}$. On conditioned that the $\mathsf{Event_F}$ does not occur, the simulation for UHO is perfect. Below, we show that the $\mathsf{Event_F}$ can occur with at most negligible probability.

  Note that the $\mathsf{Event_F}$ has already ruled out the possibility that C was the output of $\mathsf{HO}(\mathsf{ID}_s, \mathsf{ID}_r, H, M)$ for some ID_r whose corresponding value $b_r = 0$, and for arbitrary ID_s and arbitrary (H, M). The other case, if $C = (H, X, C_{AE})$ is the output of $\mathsf{HO}(\mathsf{ID}_s, \mathsf{ID}_r, H, M)$ made by $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$ for ID_r whose corresponding value $b_r = 1$, (and arbitrary ID_s, H, M), the challenger can decrypt the message correctly, which implies that $\mathcal{C}$ will not output $\perp$ for a valid $\mathsf{IBHigncryptext}$.

  Therefore, when the $\mathsf{Event_F}$ event occurs with respect to $\mathsf{UHO}(\mathsf{ID}_r, C = (H, X, C_{AE}))$, where ID_r is the receiver whose corresponding value

$b_r = 0$, $\mathsf{Event_F}$ covers the following three cases with overwhelming probability: (1) C was never output by the HO oracle; (2) $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$ did not make the $KDF(PS, X\|\mathsf{ID}_r)$ query for $PS = \mathsf{BDH}(X, h(\mathsf{ID}_r), g^c)$; and (3) (H, C_{AE}) is a valid AEAD ciphertext with respect to $K_1 = KDF(PS = \mathsf{BDH}(X,\ h(\mathsf{ID}_r),\ g^c),\ X\|\mathsf{ID}_r)$. $\mathsf{Event_F}$ can be further divided into the following two cases which can occur with negligible probability:

1. K_1 was set by $\mathcal{C}$ uniformly at random for an HO query when that $b_s = b_r = 0$. It implies that by the KDF security, with overwhelming probability, X is a part of the output of HO queries when $b_s = b_r = 0$ generated by $\mathcal{C}$ for ID_r. Let (H', X, C'_{AE}) be the challenger's output when it deals with the query $\mathsf{HO}(\mathsf{ID}_s, \mathsf{ID}_r, H', M')$. Note that $(H',\ C'_{AE})$ is the only AEAD ciphertext output by $\mathcal{C}$ with respect to K_1. As we assume $C = (H, X, C_{AE})$ was never output by $\mathcal{C}$ in the above HO query, it means that $(H', C'_{AE}) \neq (H, C_{AE})$. It implies that the adversary $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$ has output a new valid AEAD ciphertext (H', C'_{AE}) with respect to K_1. It is obvious that this $\mathsf{Event_F}$ can occur with negligible probability by the AEAD security.
2. Otherwise, with overwhelming probability, K_1 was neither set by $\mathcal{C}$ nor ever defined for the KDF oracle. It can also be expected to occur with negligible probability by the AEAD security.

Then, we conclude that the $\mathsf{Event_F}$ event can occur with at most negligible probability, and consequently the view of $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$ in the simulation is indistinguishable from that in its real attack experiment.
Phase 3: $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$ outputs (ID_{r^*}, C^*) as its forgery and the associated data contained in C^* in plain is denoted H^*. If the forgery (ID_{r^*}, C^*) is a valid IBHigncryptext created by the uncorrupted user ID_{s^*} for the uncorrupted user ID_{r^*}, it must satisfy the following conditions simultaneously:

1. $\mathsf{UnIBHigncrypt}(sk_{r^*}, \mathsf{ID}_{r^*}, C^*) = (\mathsf{ID}_{s^*}, M^*)$, and $x^* \neq 0$.
2. If there is any $\mathsf{HO}(\mathsf{ID}_{s^*}, \mathsf{ID}_{r^*}, H^*, M^*)$ query by $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$ in Phase 2, then C^* must not be the output of $\mathsf{HO}(\mathsf{ID}_{s^*},\ \mathsf{ID}_{r^*},\ H^*,\ M^*)$.

Now, let $(\mathsf{ID}_{r^*}, C^* = (H^*, X^*, C^*_{AE}))$ be the successful forgery output by $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$, which satisfies the above two conditions. Here, we require that ID_{s^*} and ID_{r^*} are the uncorrupted users and corresponding values $b_{s^*} = b_{r^*} = 0$. From the above analysis showing $\mathsf{Event_F}$ occurs with negligible probability in the UHO simulation, by the AEAD security, for the adversary $\mathcal{A}^{\mathsf{OU}}_{\mathsf{IBHC}}$'s successful forgery $(\mathsf{ID}_{r^*}, C^* = (H^*, X^*, C^*_{AE}))$, it must have made a KDF query on $(PS^*, X^*\|\mathsf{ID}_{r^*})$ with non-negligible probability, where X^* may be generated by the adversary itself; Otherwise, $\mathsf{UnIBHigncrypt}(sk_{r^*}, \mathsf{ID}_{r^*}, C^*)$ returns $\perp$ with overwhelming probability in the random oracle model. By looking up the table $\mathsf{K}_{\mathsf{DBDH}}$, $\mathcal{C}$ gets K_1 and PS^* corresponding to $X^*\|\mathsf{ID}_{r^*}$. With the help of K_1, $\mathcal{C}$ $\mathsf{UnIBHigncrypts}$ C^*, and gets the corresponding x^* which is used to generate X^* by the adversary. $\mathcal{C}$ verifies whether $X^* = h(\mathsf{ID}_{s^*})^{x^*}$ (for a successful forgery, x^* must not be 0, and the verification must be successful), then, $\mathcal{C}$ 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 3 *For the case where the target sender and the target receiver are the same, we denote by* ID_* *the user. In this case,* $h(\mathsf{ID}_{s^*}) = h(\mathsf{ID}_{r^*}) = h(\mathsf{ID}_*) = (g^a)^{y_*}$, $PS^* = e(sk_*, h(\mathsf{ID}_*))^{x^*} = e(g,g)^{a^2cy_*^2x^*}$. *It is obvious that the security is based on the* $\mathsf{Gap\text{-}SBDH}$ *assumption, on input* $(g, g^a, g^c) \in \mathbb{G}_1^3$, *the challenger* $\mathcal{C}$ *can compute* $e(g,g)^{a^2c} = (PS^*)^{\frac{1}{y_*^2x^*}}$.

The observation here is that, for any pair $(\mathsf{ID}_s, \mathsf{ID}_r, x) \neq (\mathsf{ID}_{s'}, \mathsf{ID}_{r'}, x')$, the probability of $PS = PS'$, i.e., $\Pr[PS = PS'] = \frac{1}{q}$, where $PS = e(sk_s, h(\mathsf{ID}_r))^x, PS' = e(sk_{s'}, h(\mathsf{ID}_{r'}))^{x'}$, q is the prime order of $\mathbb{G}_1$ and $\mathbb{G}_\mathsf{T}$, and $x, x' \leftarrow \mathbb{Z}_q^*$. For an identity ID_i, if $b_i = 0$, $\mathcal{C}$ aborts when it deals with a $\mathsf{CORRUPT\ Query}$. When the adversary $\mathcal{A}_{\mathsf{IBHC}}^{\mathsf{OU}}$ outputs a forgery from ID_{s^*} to ID_{r^*}, $\mathcal{C}$ dose not abort if $b_{s^*} = b_{r^*} = 0$. Suppose that the adversary makes q_{corr} times of $\mathsf{CORRUPT\ Query}$. The total probability that $\mathcal{C}$ does not abort is $(1-\gamma)^2\gamma^{q_{corr}}$. Suppose that the adversary $\mathcal{A}_{\mathsf{IBHC}}^{\mathsf{OU}}$'s running time is polynomial time t, and can break outsider unforgeability of IBHC with non-negligible probability ϵ, then the challenger $\mathcal{C}$ can solve the $\mathsf{Gap\text{-}SBDH}$ hard problem with the probability $\frac{4(1-1/q)\cdot\epsilon}{e^{(q_{corr}+2)\ln(q_{corr}+2) - q_{corr}\ln q_{corr}}}$, and its running time $t' \leq t + (q_h + q_{kdf} + q_{dbdh})O(1) + q_{corr}\cdot t_e + q_{ho}(2t_e + 1t_p + 1t_{enc}) + q_{uho}(1t_e + 1t_p + 1t_{dec})$. Up to now, we finish the proof of outsider unforgeability.

5.2 Proof of Insider Confidentiality

In this section, we present the proof of Theorem 3.

At first, the challenger $\mathcal{C}$ accepts a tuple($\mathbb{G}_1 = \langle g \rangle$, 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 inputs. 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 $\mathcal{O}_{\mathsf{DBDH}}$, which is regarded as the gap square bilinear Diffie-Hellman hard problem ($\mathsf{Gap\text{-}SBDH}$), assuming that the confidentiality of the message or the privacy of the sender's identity of IBHC is broken with non-negligible probability by the adversary $\mathcal{A}_{\mathsf{IBHC}}^{\mathsf{IC}}$. The DBDH oracle for $\mathbb{G}_1 = \langle g \rangle$ and $\mathbb{G}_\mathsf{T}$ on arbitrary input $(A' = g^{a'}, B' = g^{b'}, C' = g^{c'}, Z) \in \mathbb{G}_1^3 \times \mathbb{G}_\mathsf{T}$, outputs 1 if and only if $Z = e(g,g)^{a'b'c'}$.

During the simulation, the challenger $\mathcal{C}$ also need to maintain four tables $\mathsf{T}_\mathsf{h}, \mathsf{K}_{\mathsf{KDF}}, \mathsf{K}_{\mathsf{DBDH}}$, and ST_C. Similarly, they are all initialized to be empty. The simulation is divided into the following five phases:

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, which is modelled as a random oracle and controlled by $\mathcal{C}$ in our security proof. The challenger $\mathcal{C}$ defines the master secret key $\mathsf{msk} = c$, which is unknown to $\mathcal{C}$. Finally, the challenger $\mathcal{C}$ 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 chooses a random $y_i \leftarrow \mathbb{Z}_q^*$. Using the techniques of Coron [13], $\mathcal{C}$ flips a biased coin $b_i \in \{0,1\}$ satisfying $b_i = 1$ with probability γ, and $b_i = 0$ with probability 1 - γ [13]. If $b_i = 1$, $\mathcal{C}$ sets $h(\mathsf{ID}_i) = g^{y_i}$. Otherwise, if $b_i = 0$, $\mathcal{C}$ sets $h(\mathsf{ID}_i) = (g^a)^{y_i}$. The challenger returns $h(\mathsf{ID}_i)$ to $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$, 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 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 $sk_i = h(\mathsf{ID}_i)^c = (g^c)^{y_i}$. Otherwise, $\mathcal{C}$ aborts. Let $\mathsf{S_{corr}}$ be the set of corrupted users in the system, which is initialized to be empty. On each CORRUPT query on ID_i, if the challenger $\mathcal{C}$ returns the private key of ID_i to the adversary , it sets $\mathsf{S_{corr}} := \mathsf{S_{corr}} \bigcup \{\mathsf{ID}_i\}$.

- HO Query:

  For an HO query on $(\mathsf{ID}_s, \mathsf{ID}_r, H, M)$, where there is no restriction on H and M, which means that H can even be H^*, and M can even be M^*, the challenger $\mathcal{C}$ performs:

  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
     $PS = e(sk_s, h(\mathsf{ID}_r))^x = e((g^c)^{y_s}, (g^a)^{y_r})^x$;
     $K_1 = KDF(PS, X \| \mathsf{ID}_r)$;
     $\mathcal{C}$ stores the tuple $(X \| \mathsf{ID}_r, K_1)$ into $\mathsf{K_{KDF}}$;
     endif

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

- EXO Query:

  For an EXO query on C, the challenger $\mathcal{C}$ first 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.

  We note that in an HO query, K_1 is derived based on the correctly computed PS as long as b_s or b_r equals to 1, therefore, the simulation of $\mathcal{C}$ is perfect. If both $b_s = b_r = 0$, the challenger $\mathcal{C}$ cannot compute PS. However, X is computed correctly, and K_1 is set uniformly at random and can be used to correctly UnIBHigncrypt the output of the HO Query. Due to the fact that KDF is a random oracle, the simulation of $\mathcal{C}$ in this case is also perfect.

  Also note that in the above cases, if $b_s = b_r = 0$, the challenger $\mathcal{C}$ cannot compute the pre-share secrecy

  $$PS = e(sk_s, h(\mathsf{ID}_r))^x = e(g^{acy_s}, (g^a)^{y_r})^x,$$

  and consequently $KDF(PS, X \| \mathsf{ID}_r)$. In order to keep the consistency of the random oracle KDF, whenever the adversary $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$ makes an oracle of the form $KDF(PS', X \| \mathsf{ID}_r)$ for some ID_r whose corresponding $b_r = 0$, based on the table $\mathsf{K_{KDF}}$ and $\mathsf{T_h}$, the challenger $\mathcal{C}$

checks whether $\mathcal{O}_{\mathsf{DBDH}}(X, h(\mathsf{ID}_r), g^c, PS')$ oracle returns 1, which implies $PS' = e(X, sk_r) = e(h(\mathsf{ID}_s)^x, h(\mathsf{ID}_r))^c = \mathsf{BDH}(X, h(\mathsf{ID}_r), g^c)$; If yes, it returns the pre-shared key K_1 to the adversary, meanwhile, the challenger $\mathcal{C}$ stores the tuple $(X\|\mathsf{ID}_r, PS', K_1)$ into the table $\mathsf{K}_{\mathsf{DBDH}}$.

So far, all the simulations for CORRUPT, HO, and EXO is perfect.

- UHO Query:

  For an UHO query on $(\mathsf{ID}_r, C = (H, X, C_{AE}))$: If $b_r = 1$, $\mathcal{C}$ can perfectly simulate the game. Therefore, we only consider the case where $b_r = 0$. In this case, the challenger $\mathcal{C}$ does what he dose in the proof of outsider unforgeability with respect to $b_r = 0$. The simulation analysis is also identical to the proof of Theorem 2.

  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 $\mathsf{ID}_{r^*} \notin \mathsf{S}_{\mathsf{corr}}$. If $b_{r^*} = 1$, the challenger $\mathcal{C}$ aborts; Otherwise $\mathcal{C}$: (1) chooses $\sigma \leftarrow \{0,1\}$ (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, if $b_{s_\sigma^*} = 1$, $\mathcal{C}$ sets $x^* = a$ (which is unknown to the challenger $\mathcal{C}$), and computes $X^* = h(\mathsf{ID}_{s_\sigma^*})^{x^*} = (g^a)^{y_{s_\sigma^*}}$;(4) checks whether there is a record $(X^*\|\mathsf{ID}_{r^*}, PS, K_1)$ in the table $\mathsf{K}_{\mathsf{DBDH}}$. If yes, it outputs "failure". Otherwise, the challenger chooses K_1 uniformly at random from the key space $\mathcal{K}$ of AEAD, and stores the tuple $(X^*\|\mathsf{ID}_{r^*}, K_1)$ into the table $\mathsf{K}_{\mathsf{KDF}}$; (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 $x^{*'} \leftarrow \mathbb{Z}_q^*$, and computes $C^*_{AE} = \mathsf{Enc}_{K_1}(H^*, \mathsf{ID}_{s_\sigma^*}\|M_\sigma^*\|x^{*'})$; (6) gives the challenge IBHigncryptext (H^*, X^*, C^*_{AE}) to $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$. From this point on, with the aid of its DBDH oracle $\mathcal{O}_{\mathsf{DBDH}}$ and based upon the table $\mathsf{K}_{\mathsf{KDF}}$ and T_{h}, whenever $\mathcal{C}$ finds that $\mathcal{A}^{\mathsf{IC}}_{\mathsf{IBHC}}$ makes an query of the form $KDF(PS^*, X^*\| \mathsf{ID}_{r^*})$, the challenger checks whether $\mathcal{O}_{\mathsf{DBDH}}(X^*, h(\mathsf{ID}_{r^*}), g^c, PS^*)$ oracle returns 1, which implies $PS^* = e(X^*, sk_{r^*}) = e(g^{ay_s x^*}, g^{acy_{r^*}})$ when $b_s = 0$, or $PS^* = e(X^*, sk_{r^*}) = e(g^{ay_s}, g^{acy_{r^*}})$ when $b_s = 1$; If yes, it returns the pre-shared key K_1 to the adversary, meanwhile, the $\mathcal{C}$ stores the tuple $(X^*\|\mathsf{ID}_{r^*}, PS^*, K_1)$ into the table $\mathsf{K}_{\mathsf{DBDH}}$.

  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 an 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^*, i.e., $\mathsf{EXO}(C^*)$ is not allowed.

3. $\mathcal{A}_{\mathsf{IBHC}}^{\mathsf{IC}}$ is allowed to issue a CORRUPT query on any identity $\mathsf{ID}_i \neq \mathsf{ID}_{r^*}$, i.e., only $\mathsf{CORRUPT}(\mathsf{ID}_{r^*})$ is not allowed.

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

Similar to the proof of outsider unforgeability, if $\sigma' = \sigma$, the adversary must have made a KDF query on $(PS^*, X^* \| \mathsf{ID}_{r^*})$ with non-negligible probability in the random oracle model, where $PS^* = e(X^*, sk_{r^*}) = \mathsf{BDH}(X^*, h(\mathsf{ID}_{r^*}), g^c) = e(g,g)^{a^2 c y_{s_\sigma^*} y_{r^*} x^*}$ when $b_{s_\sigma^*} = 0$, and $PS^* = e(X^*, sk_{r^*}) = \mathsf{BDH}(X^*, h(\mathsf{ID}_{r^*}), g^c) = e(g,g)^{a^2 c y_{s_\sigma^*} y_{r^*}}$ when $b_{s_\sigma^*} = 1$. Since $\mathcal{C}$ has recorded the value $PS^* = \mathsf{BDH}(X^*, h(\mathsf{ID}_{r^*}), g^c)$, it can compute $e(g,g)^{a^2 c} = (PS^*)^{\frac{1}{y_{s_\sigma^*} y_{r^*} x^*}}$ if $b_{s_\sigma^*} = 0$, or $e(g,g)^{a^2 c} = (PS^*)^{\frac{1}{y_{s_\sigma^*} y_{r^*}}}$ if $b_{s_\sigma^*} = 1$.

Remark 4 *For the case, one of the target senders is equal to the target receiver and chosen by $\mathcal{C}$ in generating the final challenge* IBHigncryptext*, w.l.g., denote by $\mathsf{ID}_{s_0^*}$ the sender, i.e., $\mathsf{ID}_{s_0^*} = \mathsf{ID}_{s_\sigma^*} = \mathsf{ID}_{r^*}$. In this case, $h(\mathsf{ID}_{s_\sigma^*}) = h(\mathsf{ID}_{r^*}) = (g^a)^{y_{r^*}}$. It is obvious that the security is based on the* Gap-SBDH *assumption on input $(g, g^a, g^c) \in \mathbb{G}_1^3$, where $PS^* = e(sk_{r^*}, h(\mathsf{ID}_{r^*}))^{x^*} = e(g,g)^{a^2 c y_{r^*}^2 x^*}$.*

Remark 5 *The probability analysis is similar to the proof of outsider unforgeability. Suppose that the an (adversary makes q_{corr} times of* CORRUPT Query*. The total probability that $\mathcal{C}$ does not abort is $(1-\gamma)\cdot\gamma^{q_{corr}}$. Suppose that the adversary $\mathcal{A}_{\mathsf{IBHC}}^{\mathsf{OU}}$'s running time is polynomial time t, and can break the insider confidentiality of* IBHC *with non-negligible probability ϵ, then the challenger $\mathcal{C}$ can solve the* Gap-SBDH *hard problem with the probability $\frac{(1-1/q)\cdot\epsilon}{e\cdot(q_{corr}+1)}$, and its running time $t' \leq t + (q_h + q_{kdf} + q_{dbdh})O(1) + q_{corr}\cdot t_e + q_{ho}(2t_e + 1t_p + 1t_{enc}) + q_{uho}(1t_e + 1t_p + 1t_{dec})$.*

Up to now, we finish the proof of insider confidentiality.

6 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

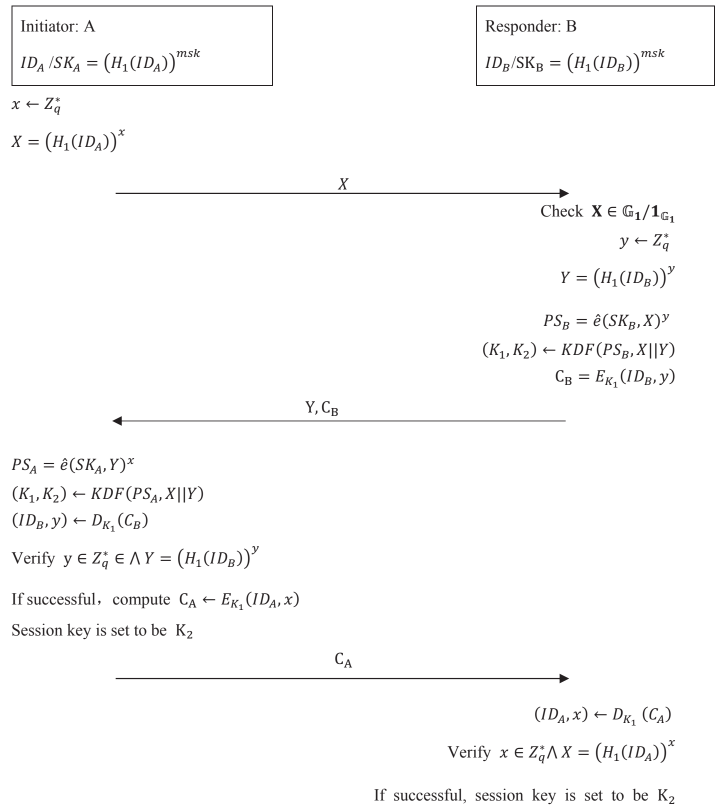


Figure 3: Construction of IB-CAKE with Type-I Bilinear Mapping

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 protocol 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 [27], $h : \{0,1\}^* \to \mathbb{G}_1$ 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

		IBHigncryption	BF-IBE [8]
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 + 4 H
	Receiver	1 E + 1 P + 1 H_2 + 1 Dec	1 E + 1 P + 1 H_2+ 3 H_1
message space		$\{0,1\}^*$	$\{0,1\}^n$
forward ID-privacy		✓	$\perp$
x-security		✓	$\times$
receiver deniability		✓	$\perp$
assumption		Gap-SBDH	BDH

Table 2: Comparison between IBHigncryption and CCA-secure BF01 [8]

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 is depicted in Fig. 3

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.

7 Comparison and Conclusion

In this section, we compare our IBHigncryption scheme with the CCA-secure Boneh-Franklin IBE [8] (referred to as BF-IBE) and the IEEE P1363.3 standard of ID-based signcryption [6] (referred to as IEEE P1363.3 for simplicity), which are reviewed in Appendix 8.1 and 8.2, respectively. We finally finish this work with some concluding remarks.

The comparisons between our IBHigncryption scheme based on symmetric bilinear pairing and BF-IBE [8], and our IBHigncryption scheme based on asymmetric bilinear pairing and the IEEE P1363.3 standard [6], are briefly summarized in Table 2 and Table 3 respectively. Therein, $\perp$ denotes "unapplicable", "-" denotes no exponentiation operation, "E" denotes modular

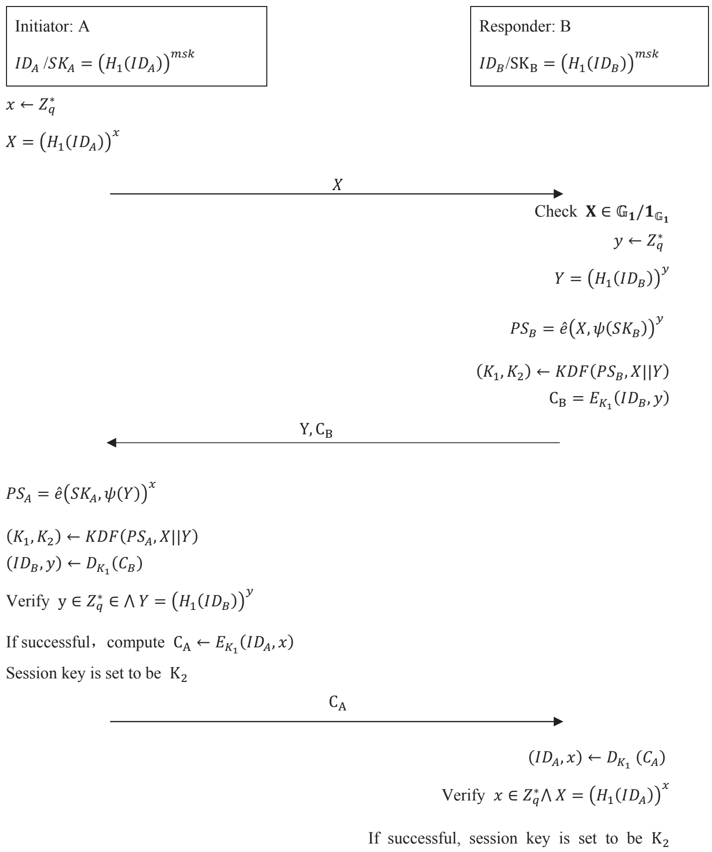


Figure 4: Construction of IB-CAKE with Type-II Bilinear Mapping

		IBHigncryption	IEEE P1363.3 [6]
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	$1\ \psi$	$1\ \mathrm{E} + 1\ \mathrm{P} + 1\ \psi$
	KeyGen	$1\ \mathrm{E} + 1\ \mathrm{H}_2$	$1\ \mathrm{E} + 1\ \mathrm{H}_1 + 1\ \mathrm{A} + 1\ \mathrm{INV}$
	Sender	$2\ \mathrm{E} + 1\ \mathrm{P} + 2\ \mathrm{H}_2 + 1\ \psi + 1\ \mathsf{Enc}$	$4\ \mathrm{E} + 3\ \mathrm{H}_1 + 1\ \mathrm{A} + 1\ \mathrm{M} + 2\ \psi$
	Receiver	$1\ \mathrm{E} + 1\ \mathrm{P} + 1\ \mathrm{H}_2 + 1\ \psi + 1\ \mathsf{Dec}$	$2\ \mathrm{E} + 2\ \mathrm{P} + 3\ \mathrm{H}_1 + 2\ \mathrm{M} + 1\ \mathrm{INV}$
message space		$\{0,1\}^*$	$\{0,1\}^n$
forward ID-privacy		✓	×
x-security		✓	×
receiver deniability		✓	×
assumption		Gap-SBDH	q-BDHIP

Table 3: Comparisons between IBHigncryption and IEEE P1363.3 Standard [6]

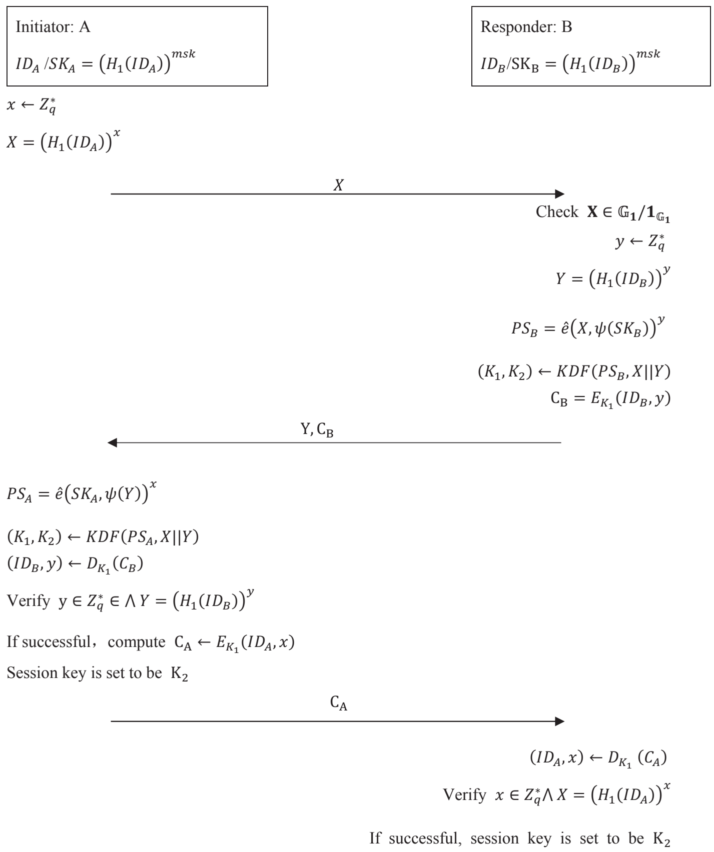


Figure 5: Construction of IB-CAKE with Type-III Bilinear Mapping

exponentiation, "P" denotes paring, "H_1" denotes a plain hashing, "H_2" denotes a hashing onto the bilinear group,"A" denotes modular addition, "M" denotes modular multiplication, "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.

Above all, our IBHigncryption is essentially as efficient as BF-IBE [8], but our IBHigncryption has a much simpler setup stage, with which our IBHigncryption can be more efficient than BF-IBE in total. 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 [8], and Q_{pub} in IEEE P1363.3 [6]). We remark that waiving the master public key P_{pub} brings advantages not only on space and computational complexity, but on security (e.g., to adversaries outside the system) as well. Meanwhile, our IBHigncryption offers entity authentication, ID-privacy, receiver-deniability and x-security simultaneously. 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 and IEEE P1363.3, it needs to employ some appropriate symmetric-key encryption scheme in reality. In this case, suppose that the same authenticated encryption is used in all the three schemes, IBHigncryption can even be slightly more efficient than BF-IBE [8] without considering the advantages of the much simpler setup stage.

Since IEEE 1363.3 is constructed over asymmetric bilinear group, in order to illustrate the advantage of our IBHigncryption, we generalize our IBHigncryption to asymmetric bilinear group in Subsection 4.2.1. The comparison results of these two schemes are summarized in Table 3. From the table, we can draw the conclusion that our IBHigncrypt scheme is much more efficient than IEEE P1363.3, especially on the receiver side. Additionally, IEEE P1363.3 does not consider the case that the sender/signer ID_s equals to the receiver/verifer ID_r in its security proof, but ours does.

In this work, we introduce the first identity-based higncryption (IBHigncryption). We present the formal definitions for IBHigncryption and its security model, and conduct detailed security proof. Our construction of IBHigncryption is conceptually simple and highly practical, and can exceed some fundamental and widely deployed standards in identity-based cryptography. In particular, it is well applicable to mission critical communication for 5G.

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] J. Baek, R. Safavi-Naini, and W. Susilo, "Efficient multi-receiver identity-based encryption and its application to broadcast encryption," in *Public Key Cryptography - PKC 2005, 8th International Workshop on Theory and Practice in Public Key Cryptography, Les Diablerets, Switzerland, January 23-26, 2005, Proceedings*, 2005, pp. 380–397.

[6] P. S. L. M. Barreto, B. Libert, N. McCullagh, and J. Quisquater, "Efficient and provably-secure identity-based signatures and signcryption from bilinear maps," in *Advances in Cryptology - ASIACRYPT 2005, 11th International Conference on the Theory and Application of Cryptology and Information Security, Chennai, India, December 4-8, 2005, Proceedings*, 2005, pp. 515–532.

[7] D. Boneh and X. Boyen, "Efficient selective-id secure identity based encryption without random oracles," in *Proceedings of Eurocrypt 2004, volume 3027 of LNCS*. Springer-Verlag, 2004, pp. 223–238.

[8] D. Boneh and M. K. Franklin, "Identity-based encryption from the weil pairing," in *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*, 2001, pp. 213–229.

[9] X. Boyen and L. Martin, "Identity-based cryptography standard (IBCS) #1: Supersingular curve implementations of the BF and BB1 cryptosystems," *RFC*, vol. 5091, pp. 1–63, 2007. [Online]. Available: https://doi.org/10.17487/RFC5091

[10] C. Brzuska, N. P. Smart, B. Warinschi, and G. J. Watson, "An analysis of the EMV channel establishment protocol," in *2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4-8, 2013*, A. Sadeghi, V. D. Gligor, and M. Yung, Eds. ACM, 2013, pp. 373–386. [Online]. Available: http://doi.acm.org/10.1145/2508859.2516748

[11] S. Chatterjee and A. Menezes, "On cryptographic protocols employing asymmetric pairings - the role of psi revisited," *IACR Cryptology ePrint Archive*, vol. 2009, p. 480, 2009. [Online]. Available: http://eprint.iacr.org/2009/480

[12] S. Chatterjee and A. Menezes, "On cryptographic protocols employing asymmetric pairings - the role of Ψ revisited," *Discrete Applied Mathematics*, vol. 159, no. 13, pp. 1311–1322, 2011. [Online]. Available: https://doi.org/10.1016/j.dam.2011.04.021

[13] S. R. Cowell, V. Beiu, L. Daus, and P. Poulin, "On the exact reliability enhancements of small hammock networks," *IEEE Access*, vol. 6, pp. 25 411–25 426, 2018. [Online]. Available: https://doi.org/10.1109/ACCESS.2018.2828036

[14] U. Feige, A. Fiat, and A. Shamir, "Zero-knowledge proofs of identity," *Journal of Cryptology*, vol. 1, no. 2, pp. 77–94, 1988.

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

[16] E. Fujisaki and T. Okamoto, "Secure integration of asymmetric and symmetric encryption schemes," *Journal of Cryptology*, vol. 26, no. 1, pp. 80–101, 2013.

[17] M. Groves, "Elliptic curve-based certificateless signatures for identity-based encryption (ECCSI)," *RFC*, vol. 6507, pp. 1–17, 2012. [Online]. Available: https://doi.org/10.17487/RFC6507

[18] M. Groves, "MIKEY-SAKKE: sakai-kasahara key encryption in multimedia internet keying (MIKEY)," *RFC*, vol. 6509, pp. 1–21, 2012. [Online]. Available: https://doi.org/10.17487/RFC6509

[19] M. Groves, "Sakai-kasahara key encryption (SAKKE)," *RFC*, vol. 6508, pp. 1–21, 2012. [Online]. Available: https://doi.org/10.17487/RFC6508

[20] S. Halevi and H. Krawczyk, "One-pass HMQV and asymmetric key-wrapping," in *Public Key Cryptography - PKC 2011 - 14th International Conference on Practice and Theory in Public Key Cryptography, Taormina, Italy, March 6-9, 2011. Proceedings*, 2011, pp. 317–334.

[21] F. L. M. N. J. Arkko, E. Carrara and K. Norrman, "Mikey: Multimedia internet keying," *RFC*, vol. 3830, pp. 1–66, 2004. [Online]. Available: https://doi.org/10.17487/RFC3830

[22] H. Khan, B. Dowling, and K. M. Martin, "Identity confidentiality in 5g mobile telephony systems," *IACR Cryptology ePrint Archive*, vol. 2018, p. 876, 2018. [Online]. Available: https://eprint.iacr.org/2018/876

[23] Y. Lair and G. Mayer, "Mission critical services in 3GPP."

[24] B. Libert and J. Quisquater, "Identity based undeniable signatures," in *Topics in Cryptology - CT-RSA 2004, The Cryptographers' Track at the RSA Conference 2004, San Francisco, CA, USA, February 23-27, 2004, Proceedings*, ser. Lecture Notes in Computer Science, T. Okamoto, Ed., vol. 2964. Springer, 2004, pp. 112–125. [Online]. Available: https://doi.org/10.1007/978-3-540-24660-2_9

[25] S. C. Ramanna, S. Chatterjee, and P. Sarkar, "Variants of waters' dual system primitives using asymmetric pairings - (extended abstract)," in *Public Key Cryptography - PKC 2012 - 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, Germany, May 21-23, 2012. Proceedings*, ser. Lecture Notes in Computer Science, M. Fischlin, J. A. Buchmann, and M. Manulis, Eds., vol. 7293. Springer, 2012, pp. 298–315. [Online]. Available: https://doi.org/10.1007/978-3-642-30057-8_18

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

[27] P. Rogaway, "Authenticated-encryption with associated-data," in *Proceedings of the 9th ACM Conference on Computer and Communications Security, CCS 2002, Washington, DC, USA, November 18-22, 2002*, 2002, pp. 98–107.

[28] J. Roskind, "Quick UDP internet connections: Multiplexed stream transport over UDP."

[29] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology, Proceedings of CRYPTO '84, Santa Barbara, California, USA, August 19-22, 1984, Proceedings*, 1984, pp. 47–53.

[30] Y. Zhao, "Identity-concealed authenticated encryption and key exchange," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, 2016, pp. 1464–1479.

[31] Y. Zheng, "Digital signcryption or how to achieve cost(signature & encryption) << cost(signature) + cost(encryption)," in *Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings*, 1997, pp. 165–179.

8 Appendices

8.1 CCA-Secure Boneh-Franklin IBE

Boneh's identity-based encryption from Weil paring [8] (referred to as BF-IBE for simplicity) is the first practical identity-based encryption from pairing. It is also regarded as one of the most efficient and most usable identity-based encryption scheme. In [8], the authors proposed a CPA-secure IBE, and a CCA-secure IBE via the Fujisaki-Okamoto transformation [16]. Below, we briefly recall the CCA-secure one.

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 $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$; (2) chooses a random generator $g \in \mathbb{G}_1$; (3) picks $s \leftarrow \mathbb{Z}_q^*$ and sets $P_{\mathsf{pub}} = g^s$; (4) chooses a cryptographic hash function $h_1 : \{0,1\}^* \to \mathbb{G}_1$, and three cryptographic hash functions $h_2 : \mathbb{G}_2 \to \{0,1\}^n, h_3 : \{0,1\}^n \times \{0,1\}^n \to \mathbb{Z}_q^*$, and $h_4 : \{0,1\}^n \to \{0,1\}^n$ 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 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 is the master 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.

8.2 IEEE P1363.3 Standard for ID-Based Signcryption

The identity-based signcryption from bilinear maps [6], 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 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 key is $s \in \mathbb{Z}_q^*$.

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

- $\mathsf{Sign/Encrypt}$: 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 $T = (g_1^{h_1(\mathsf{ID_B})} \cdot \psi(Q_\mathsf{pub}))^x = (g_1^{h_1(\mathsf{ID_B})+s})^x$.

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

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

  1. computes $r = e(T, S_{\mathsf{ID_B}})$, $M = C \oplus h_3(r)$, 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 $(u, S) \in \mathbb{Z}_q^* \times \mathbb{G}_1$.