Hindawi
Security and Communication Networks
Volume 2022, Article ID 8653453, 24 pages
https://doi.org/10.1155/2022/8653453

Research Article

IBE-Signal: Reshaping Signal into a MITM-Attack-Resistant Protocol

Shiqi Liu,[1,2] **Yan Shao**,[1] **Hanbo Luo**,[1] **and Hong Di**[1]

[1]*School of Cyber Science and Engineering, University of International Relations, Beijing 100091, China*
[2]*School of Cyber Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China*

Correspondence should be addressed to Hong Di; di_hong@163.com

Received 18 November 2021; Revised 13 April 2022; Accepted 27 May 2022; Published 21 July 2022

Academic Editor: Zhili Zhou

The Signal Protocol is one of the most popular privacy protocols today for protecting Internet chats and supports end-to-end encryption. Nevertheless, despite its many advantages, the Signal Protocol is not resistant to Man-In-The-Middle (MITM) attacks because a malicious server can distribute the forged identity-based public keys during the user registration phase. To address this problem, we proposed the IBE-Signal scheme that replaced the Extended Triple Diffie–Hellman (X3DH) key agreement protocol with enhanced Identity-Based Encryption (IBE). Specifically, the adoption of verifiable parameter initialization ensures the authenticity of system parameters. At the same time, the Identity-Based Signature (IBS) enables our scheme to support mutual authentication. Moreover, we proposed a distributed key generation mechanism that served as a risk decentralization to mitigate IBE's key escrow problem. Besides, the proposed revocable IBE scheme is used for the revocation problem. Notably, the IND-ID-CPA security of the IBE-Signal scheme is proven under the random oracle model. Compared with the existing schemes, our scheme provided new security features of mutual authentication, perfect forward secrecy, post-compromise security, and key revocation. Experiments showed that the computational overhead is lower than that of other schemes when the Cloud Privacy Centers (CPCs) number is less than 8.

1. Introduction

Revelations of mass surveillance of communications have made consumers more privacy-aware. Scientists and developers have proposed security techniques for end-users even if they do not trust the service providers fully [1]. The Signal Protocol is a cryptographic protocol that secures the text messages of billions of people. It provides end-to-end encryption and is applied by secure communication tools such as WhatsApp, Facebook Messenger, and Microsoft Skype [2, 3]. The information of applications with the Signal Protocol and their monthly active users are shown in Table 1.

The Signal Protocol consists of two main sub-algorithms, X3DH and Double Ratchet. X3DH generates many sets of ephemeral key pairs for each user in addition to the permanent keys. Then a shared key is created for users by combining ephemeral and permanent key pairs. The Double Ratchet algorithm contains two ratchets [4], the Diffie–Hellman ratchet and the Symmetric-key ratchet. The Symmetric-key ratchet uses the key derivation function (KDF) chain to derive new keys. It offers a previous-key-based key to encrypt each message and attempt to provide Perfect Forward Secrecy (PFS). Even if an attacker cracks the key of one message, the keys of previous messages cannot be derived inversely. The included Diffie–Hellman ratchet aims at ensuring Post-compromise Security (PCS). Therefore, the Double Ratchet algorithm satisfies both PFS and PCS. Neither the keys generated before the key K nor after can be calculated when K is compromised.

Although having PFS and PCS, Signal Protocol cannot resist MITM attacks due to X3DH's key distribution problem. Since there is no public key authentication in X3DH, the server can be an attacker against the system itself or a collaborator of the adversary.

Table 1: The information of applications and their monthly active users.

Software using the signal protocol	Statistical deadline	Monthly active users
Signal [5]	2021	40,000,000
WhatsApp [6]	2021	2,000,000,000
Facebook messenger [7]	2020	2,770,000,000
Skype [8]	2019	300,000,000

The developers did consider the shortcomings of Signal Protocol to protect against MITM attacks and made some refinements. So each chat has a unique safety number [9] that enables one to verify the security of messages and specific contacts. As shown in Figure 1, the safety number is a hash of identity generated locally by both communicators, usually in a QR code. If a safety number has been marked as verified, any change can be manually approved before sending a new message [9]. However, a better plan to prevent MITM attacks should be in advance, rather than doing detection when an attack has approached.

It is unreasonable to believe that a server will duly alert when it is a potential attacker. Only the user actively verifies the safety number through a third-party channel is reliable. Schröder et al. [10] found that most users could not successfully verify each other's safety numbers due to usability problems and incomplete mental models. Worsely, the Signal Protocol is not secure under a threat model with an untrusted server since much of the Signal Protocol's functionality is on the server trusted premise.

The existence of MITM attacks is essentially a lack of public-key authentication. We are considering replacing X3DH with Identity-Based Encryption (IBE) for improvement. IBE scheme is a public-key cryptosystem where any string representing the identity is a valid public key [11]. A user can generate a public key from a known unique identifier, such as a phone number. Then a trusted third-party server calculates the corresponding private key from the public key. This process eliminates the need to distribute the public key before exchanging encrypted data. The sender can generate the public key and encrypt the data simply by using the receiver's unique identifier. Correspondingly, the receiver can generate the private key with the help of a trusted third-party server, the Private Key Generator (PKG).

Blazy et al. [12] introduced IBS into the Signal Protocol to make it resistant to MITM attacks, but the problems [13–16] inherited in IBS/IBE are yet to be solved:

The Authenticity of System Parameters Problem. In the BF-IBE scheme, the attacker can generate the public system parameters by himself to make the users mistakenly believe that the parameters are correct.

One-Way Authentication Problem. Instant messaging protocols generally support mutual authentication, but BF-IBE only authenticates the recipient's identity.

Key Escrow Problem. The PKG generates the users' private keys with the master key. For this reason, identity theft might occur since the PKG is available to sign any message with the users' identities. Users may also slander the PKG for misusing their private keys for signing, which threatens the undeniability of the signature.

Public Key Revocation Problem. A user can generate a new public-private key pair when his key is at risk or expire. However, the BF-IBE scheme is unavailable to revoke the user's public key.

1.1. Our Contributions. Our advanced Signal Protocol has the following security features:

Resist MITM attack. Our IBE-Signal scheme is resistant to MITM attacks by leveraging the IBE scheme. Furthermore, we fixed the problems inherited in IBS/IBE.

The Authenticity of System Parameters. By introducing a verifiable parameter initialization technique, the public validation of parameters is guaranteed to ensure the correctness of parameters transmitted by the Key Generation Center (KGC) and Cloud Privacy Centers (CPCs).

Mutual Authentication. Our IBE-Signal scheme supports mutual authentication by introducing an Identity-Based Signature (IBS).

Secure Key Escrow. We used distributed key generation to solve the key escrow problem, inspired by the ESKI-IBE scheme proposed by Kumar et al. [17]. Moreover, we also used the KGC and the CPCs instead of one PKG. Compared with [17], we reduced the computational overhead since there is no need to negotiate parameters after an initial session.

Resist User Slandering. The KGC and the CPCs own the master key share separately, which disenables an attacker from obtaining the complete key. Therefore, an attacker cannot decrypt the users' ciphertexts or forge the users' signatures, and users cannot defame the server for misusing their private key.

Secure Key Issuing. The blind signature enables the KGC or the CPCs to issue the private key without seeing the actual information sent by the user.

Key Revocation. We achieved efficient public key revocation by adopting Revocable IBE. This scheme proposed by Boldyreva et al. [18] is built on the Fuzzy IBE scheme and binary tree data structure.

Perfect Forward Secrecy and Post-compromise Security. Our scheme still retains the PFS and PCS of the Signal Protocol by introducing the concepts of session and connection in SSL protocol into our scheme and integrating the Double Ratchet algorithm into the *CONNECTION* part.

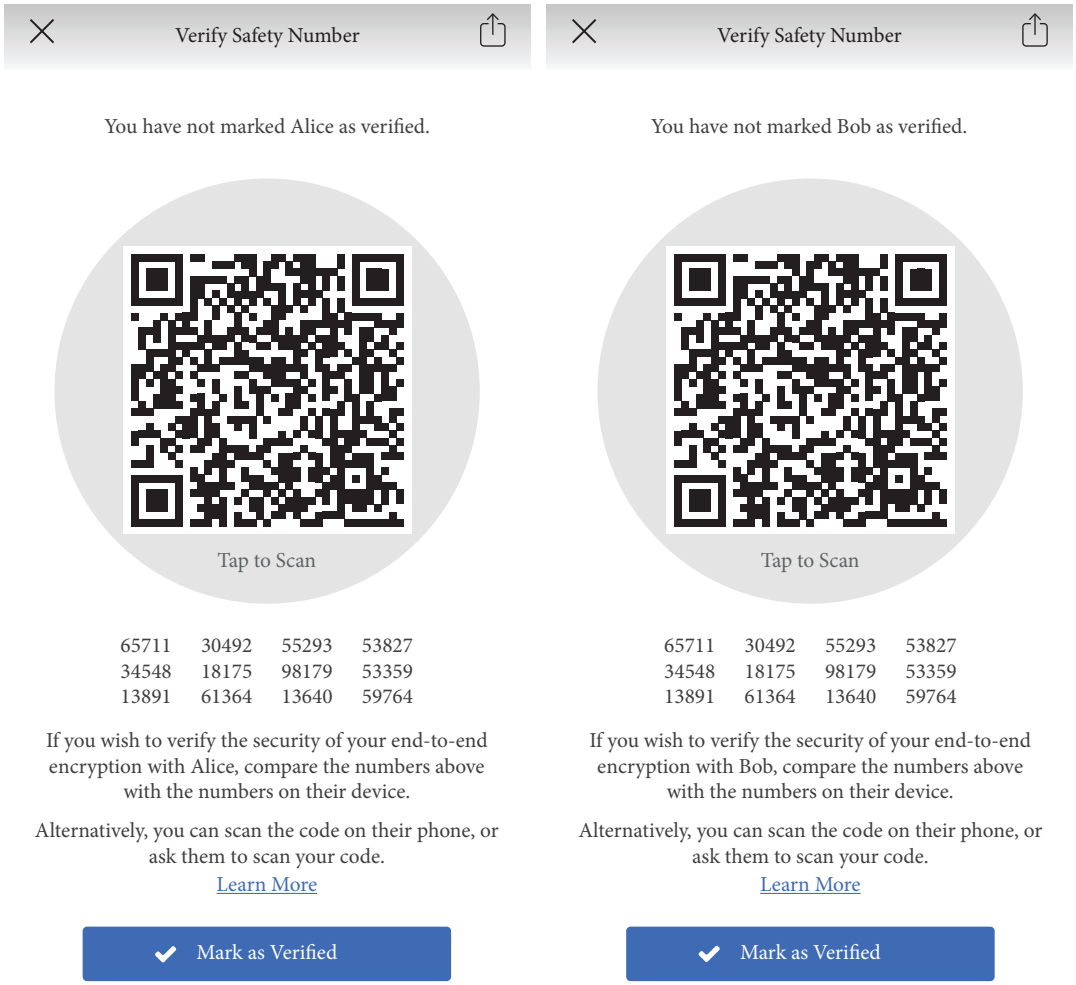

Figure 1: Safety number in the signal application.

IND-ID-CPA. We proved the IND-ID-CPA security of the IBE-Signal scheme under the random oracle model and the Bilinear Diffie–Hellman (BDH) assumption. With more powerful attackers, we gained a more compact reduction than [17] in a challenger-adversary game.

1.2. Related Work. The security proof of the Signal Protocol has always been a research hotspot. Cohn-Gordon et al. [19] analyzed the security of the Signal Protocol and other authenticated key exchange protocols. Van Dam [20] provided an automated analysis of the Signal Protocol. Frosch et al. [21] first proved that TextSecure (the predecessor of the Signal) achieves most of its claimed security goals if the key registration is secure. Alwen et al. [22] gave formal proof of the perfect forward secrecy and post-compromise security of the Signal Protocol. Cohn-Gordon et al. [1] analyzed the security of multi-stage key agreement protocols based on the Double Ratchet algorithm for the first time. In addition, Kobeissi et al. [23] verified the security of end-to-end cryptographic protocols like the Signal Protocol with automated tools.

The Signal Protocol cannot resist MITM attacks when the key registration is insecure. There are many ways to resist MITM attacks. The traditional methods are Digital Signatures and Message Authentication Codes. Rivest and Shamir [24] proposed the interlock protocol to expose the eavesdropper of full-duplex communication. However, the interlock protocol increases the communication cost and is not suitable for half-duplex communication. Khader and Lai [25] proposed a method to resist MITM attacks in the Diffie–Hellman Key Exchange Protocol with the Geffe generator.

The MITM attacks are due to the lack of public-key authentication, traditionally implemented by distributing public key certificates through the Public Key Infrastructure (PKI). In 1984, Shamir [15] proposed the IBE scheme to simplify the public key certificate management in the PKI. Boneh and Franklin [16] implemented a practical and functional IBE scheme, the BF-IBE scheme, based on bilinear pairings on elliptic curves in 2001. Goyal [26] introduced the Accountable Authority Identity-Based Encryption (A-IBE) scheme to reduce trust in the PKG as an effective solution for the key escrow problem. The A-IBE scheme was matured into the Black Box AIBE scheme by Goyal et al. [27]. Later, Garg et al. [28, 29] proposed the

Registration-Based Encryption (RBE) scheme, aggregating the KGC and compactly compressing all users' public keys into the master public key.

1.3. Organization. To begin, in Section 2, we presented helpful preliminary information. Then, in Section 3, we detailed the construction of our approach. In Section 4, we analyzed the security features of the scheme. In Section 5, the IND-ID-CPA security of our scheme was proved. In Section 6, we analyzed the scheme's performance and compared it with other schemes in performance and security. Finally, we concluded with comments on our work and limitations for future work in Section 7.

2. Preliminaries

We briefly introduced the background knowledge needed to read this paper, including Symmetric Bilinear Function, BDH Problem, X3DH Algorithm, BF-IBE Scheme, Hess's IBS Scheme, and RIBE Scheme.

2.1. Notation. We defined a secure symmetric encryption algorithm as Enc. For $\mathrm{Enc}(M_1, M_2, M_3, K)$, K is the key, and M_1, M_2, M_3 are the messages to be encrypted. The symmetric decryption algorithm Dec and the signature algorithm Sign are given similar forms.

2.2. Symmetric Bilinear Function. Let q be a large prime, G_1 is an additive group, and G_2 is a multiplicative group of the same order q. Given a symmetric bilinear function from G_1 to G_2 as e: $G_1 \times G_1 \longrightarrow G_2$ and e satisfies the following properties:

Bilinearity: $\forall P, Q \in G_1$ and $a, b \in \mathbb{Z}$, such that $e([a]P, [b]Q) = e(P, Q)^{ab}$.

Nondegeneracy: $\forall P \in G_1$, such that $e(P, P) \neq 1$, that means the generator of G_2 cannot be the identity element.

Computability: $\forall P, Q \in G_1$, there exists an effective algorithm to compute $e(P, Q)$.

Symmetry: $\forall P, Q \in G_1$, such that $e(P, Q) = e(Q, P)$.

2.3. BDH Problem. Given $(P, [a]P, [b]P, [c]P)(a, b, c \in \mathbb{Z}_q^*)$, compute $e(P, P)^{abc} \in G_2$, where e is a bilinear function, P is the generator on G_1, and G_1, G_2 are two groups of the same order q[30]. Assume the algorithm A is used to solve the BDH problem:

$$\Pr\left[A(P, [a]P, [b]P, [c]P) = e(P, P)^{abc}\right] \geq \varepsilon. \tag{1}$$

2.4. X3DH Algorithm. X3DH [31] is a tripartite key negotiation protocol where the two communicating parties go through a server to achieve asynchronous encrypted communication.

Figure 2 shows the specific design of the protocol. First, Bob registers his public key bundle to the server. When Alice wants to establish communication with Bob, she retrieves Bob's public key bundle from the server. Then, Alice verifies the prekey signature and performs four Diffie–Hellman key exchanges in the order shown in Figure 3. Finally, she generates the shared key SK. Bob similarly generates SK, decrypts the message M_1 and Alice's safety number SN, and encrypts the message M_2 and the locally calculated safety number SN. The server forwards it to Alice.

Figure 4 shows an example when the protocol is subjected to a MITM attack by a malicious server Trudy. Both parties in the protocol can get the safety number forwarded to each other by the server. However, it is still impossible to determine whether there is a MITM attack based on direct comparison results. Through the original channel comparison, the safety number will still be tampered with by the man in the middle.

2.5. BF-IBE Scheme. The public key of the BF-IBE [16] scheme is determined by the identity, and a trusted third-party PKG generates the private key.

Setup: Suppose k is the security parameter, q is a large prime of k-bits, G_1 is an additive group, and G_2 is a multiplicative group where the order of both groups is q, e: $G_1 \times G_1 \longrightarrow G_2$ is a bilinear mapping. P is a generator of G_1, H_1: $\{0,1\}^* \longrightarrow G_1^*$ and H_2: $G_2^* \longrightarrow \{0,1\}^n$ are two hash functions where n is the length of the message to be encrypted. The PKG picks $s \in \mathbb{Z}_q^*$ as the master key and computes the public key $P_{pub} = [s]P$. PKG keeps the public system parameter $\mathrm{param} = \{q, G_1, G_2, e, n, P, P_{\mathrm{pub}}, H_1, H_2\}$ and the secret s.

Key generation: The public key corresponding to $ID \in \{0,1\}^*$ is $Q_{\mathrm{ID}} = H_1(\mathrm{ID})$. The PKG computes the private key $S_{ID} = [s]Q_{ID}$.

Encryption: The public key of the receiver is Q_{ID}. To encrypt $M \in \{0,1\}^n$, calculate $r \xleftarrow{R} \mathbb{Z}_q^*$, $g = e(Q_{\mathrm{ID}}, P_{\mathrm{pub}}) \in G_2^*$, $C = ([r]P, M \oplus H_2(g^r))$.

Decryption: For the given ciphertext $C = \{U, V\}$ and the private key S_{ID}, the receiver can decrypt C as $M = V \oplus H_2(e(S_{\mathrm{ID}}, U))$.

2.6. Hess's IBS Scheme. The signature scheme used in this paper is Hess's IBS scheme [32], and Hess gives the security proof of their scheme under the random oracle model. The scheme reduces the number of bilinear pair operations and outperforms Shamir's scheme [33] and Jae Cha's scheme [34] in terms of efficiency.

Setup: Suppose q is a large prime, G_1 is an additive group, and G_2 is a multiplicative group where the order of both groups is q, e: $G_1 \times G_1 \longrightarrow G_2$ is a bilinear mapping. P is a generator of G_1, H_1: $\{0,1\}^* \longrightarrow G_1^*$ and H_2: $\{0,1\}^* \times G_2^* \longrightarrow \mathbb{Z}_q^*$ are two hash functions. The PKG picks $s \in \mathbb{Z}_q^*$ as the master key and computes the public key $P_{pub} = [s]P$. The PKG keeps the system parameter $\mathrm{param} = \{q, G_1, G_2, e, P, P_{\mathrm{pub}}, H_1, H_2\}$ public and s secret.

Key generation: After the identity verification passes, the PKG computes the public key $Q = H_1(ID)$ based

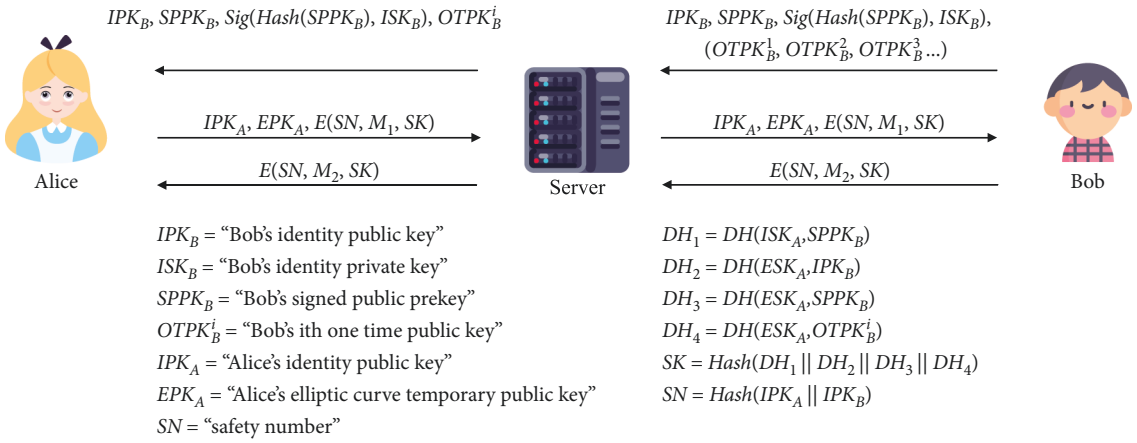

Figure 2: The X3DH.

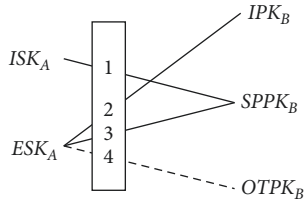

Figure 3: The SK generation process.

Figure 4: A MITM attack in X3DH.

on the ID and the corresponding private key $D = [s]H_1(ID) = [s]Q$.

Sign: m is the message that needs to be signed, the signer randomly selects $k \in \mathbb{Z}_q^*$ and $P_1 \in G_1^*$, computes $r = e(P_1, P)^k$, $V = H_2(m, r)$, and $U = [V]D + [k]P_1$. $\sigma = (U, V)$ is the signature for m.

Verify: After receiving the signature $\sigma = (U, V)$, the verifier computes $r = e(U, P)e(Q, -P_{pub})^V$ where $Q = H_1(ID)$. When $H_2(m, r) = V$ holds, the signature is valid.

2.7. RIBE Scheme. To make the IBE algorithm support efficient public key revocation, Boldyreva [18] et al. proposed the RIBE scheme. Each user corresponds to two attributes: identity and validity time of the public key, and therefore corresponds to two keys in RIBE: the *private key* and the *key update*. The core of RIBE is that only having two private keys can decrypt the message, while the PKG cannot update the *key update* to the user whose public key has been revoked. We only present the basic definition of RIBE here. For more details, please consult [18].

Setup: $S(1^\kappa, n) \longrightarrow (pk, mk, rl, st)$: n is the number of users, pk is the public parameters, mk is the master key, rl is the revocation list, and st is the binary tree representing states.

Private key generation: $SK(pk, mk, \omega, st) \longrightarrow (sk_\omega, st)$: ω is the identity, sk_ω is the private key, and the state st is updated by this function.

Key update generation: $KU(pk, mk, t, rl, st) \longrightarrow ku_t$: t is the validity time and ku_t is the key update.

Decryption key generation: $DK(sk_\omega, ku_t) \longrightarrow dk_{\omega,t}$ *or* $\perp$: If the identity ω is revoked, then the function outputs $\perp$; else outputs the decryption key $dk_{\omega,t}$.

Encryption: $E(pk, \omega, t, m) \longrightarrow c$: t is when the encryption occurs, m is the plaintext, and c is the ciphertext.

Decryption: $D(dk_{\omega,t}, c) \longrightarrow m$ *or* $\perp$: If the ciphertext c is invalid, the function outputs $\perp$; else outputs the plaintext m.

Revocation: $R(\omega, t, rl, st) \longrightarrow rl$: ω is the identity to be revoked, t is the revocation time, and rl is updated by this function.

3. Scheme Definition

We first define the threat model of our IBE-Signal scheme, then introduce the framework of the whole scheme, and finally present the scheme's construction in two parts.

3.1. Threat Model. As shown in Figure 5, our model uses one KGC, multiple CPCs, and one Key Authority to replace the single PKG in the BF-IBE model shown in Figure 6. The

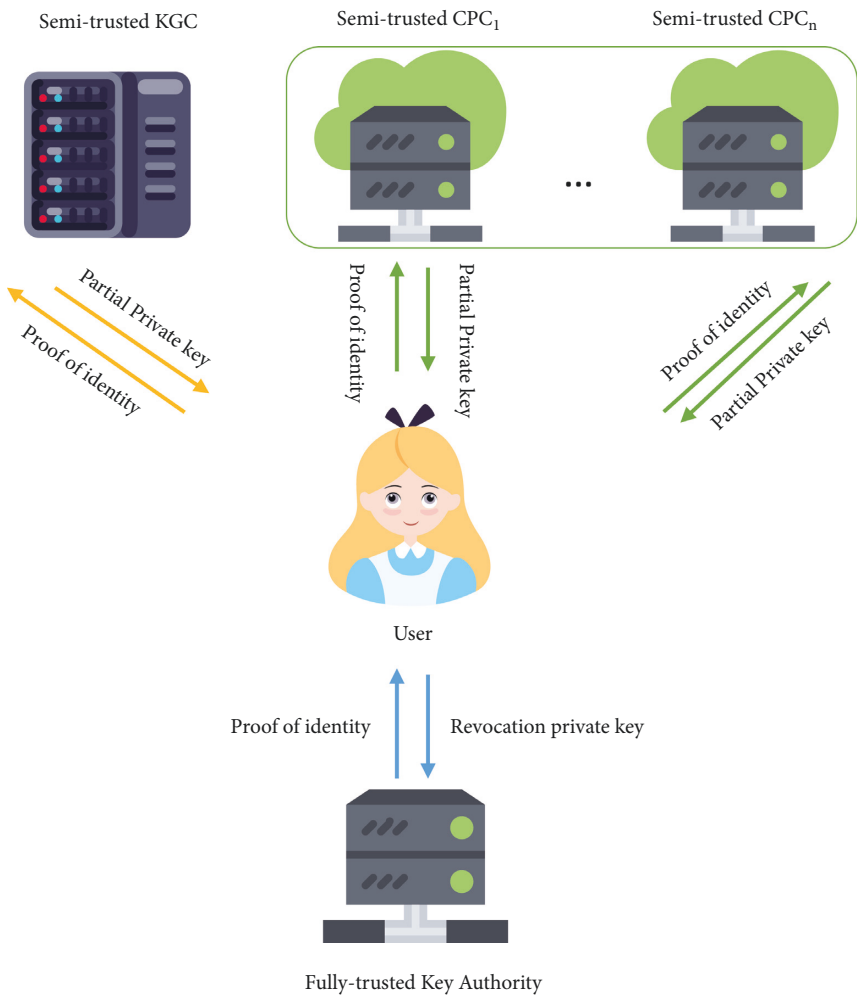

Figure 5: The IBE-Signal scheme's threat model.

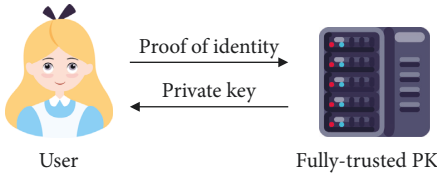

Figure 6: The BF-IBE scheme's threat model.

KGC sets most system parameters, and the CPCs also have the private key share for risk reduction. The Key Authority is fully trusted and only responsible for public key revocation. In addition, our public key revocation feature is optional, and if the user turns it off, the Key Authority will no longer serve him.

We assume that an ambitious adversary can collude with the KGC and n-1 CPCs. In other words, at least one CPC in our scheme is credible. Corrupt KGC/n-1 CPCs can assign the partial private key to any identity submitted by the adversary. The adversary can launch a chosen-plaintext attack and can finally select an ID^* to challenge. We allow the adversary to retrieve the partial private key from the KGC and n-1 partial private keys from the CPCs in this case. Before and after the challenge, the adversary can choose a random ID to query where $ID \neq ID^*$. Furthermore, we allow the adversary to retrieve the partial private key from the KGC and n partial private keys from the CPCs in this case. This process can be repeated polynomial bounded times.

Even with these capabilities, the adversary has a negligible advantage against our IBE-Signal scheme.

3.2. The Framework of the IBE-Signal Scheme. Like the SSL protocol [35], our IBE-Signal scheme contains parts of the *SESSION* and the *CONNECTION*. The *SESSION* part refers to a collection of parameters and encryption keys generated through a handshake between two communicating parties. In contrast, the *CONNECTION* part refers to a subsequent session established after the *SESSION* part using the shared key *RK*. The *SESSION* part is more overhead than the *CONNECTION* part for the server. Once the *SESSION* part is established, a user can use a *SESSION* part context to create the *CONNECTION* part for the next time. This reduces the communication overhead of the protocol. The *SESSION* part consists of eight Probabilistic Polynomial-Time algorithms and can be described as $SESSION == (S, EKG, SI, E, DKG, D, V, R)$. The *CONNECTION* part consists of four

Probabilistic Polynomial-Time algorithms and can be described merely as $CONNECTION == (SKG, E, RKG, D)$.

The *SESSION* part is shown in Figure 7, and the *CONNECTION* part is shown in Figure 8.

3.3. Scheme Construction. We present the scheme's construction in two parts, the *SESSION* and the *CONNECTION*. Since the scheme uses many symbols, Table 2 provides a table of characters for the reader's convenience.

3.3.1. SESSION Part. Let k be the secure parameter. G is the algorithm to generate parameters of BDH, including a prime q, an additive group G_1, a multiplicative group G_2 of the same order q, and a bilinear function e: $G_1 \times G_1 \longrightarrow G_2$.

Set up $S(1^k) \longrightarrow (\text{params}, S_i, a, rl, st)$, where $i = 0, 1, \ldots, n$. Including the following three algorithms, as shown in Figure 9:

(1) *KGC Setup* $(1^k) \longrightarrow (\text{param}_1, S_0)$. This algorithm is run by the KGC:

$$\begin{aligned} &(q, G_1, G_2, e) \leftarrow G, P \overset{R}{\leftarrow} G_1, S_0 \overset{R}{\leftarrow} \mathbb{Z}_q^*, Y_0 = P_0 = [S_0]P, \\ &H_1\colon \{0,1\}^* \longrightarrow G_1^*, H_2\colon G_2 \longrightarrow \{0,1\}^k, H_3\colon \{0,1\}^k \longrightarrow \{0,1\}^k, \\ &H_4\colon \{0,1\}^* \longrightarrow \mathbb{Z}_q^*, H_5\colon \{0,1\}^* \times G_2^* \longrightarrow \mathbb{Z}_q^*, \\ &H_6\colon \{0,1\}^k \times \{0,1\}^k \longrightarrow \{0,1\}^k, \text{KDF}\colon G_1 \times \{0,1\}^k \longrightarrow \{0,1\}^k, \end{aligned} \tag{2}$$

where P is a generator of group G_1, S_0 is the private key of the KGC, P_0 is the public key of the KGC, and Y_0 is the shared public key of the KGC. $H_1, H_2, H_3, H_4, H_5, H_6, KDF$ are secure hash functions, k represents the length of the message key. The system parameter of the KGC $\text{param}_1 = (q, G_1, G_2, e, k, P, P_0, Y_0, H_1, H_2, H_3, H_4, H_5, H_6, \text{KDF})$ is public, while the private key of the KGC S_0 is private.

(2) *CPC Setup* $(\text{param}_1) \longrightarrow (\text{param}_2, S_i)$, where $i = 1, 2, \ldots, n$. This algorithm is run by the CPCs:
Each CPC_i chooses $S_i \in \mathbb{Z}_q^*$ as the private key and computes $P_i = [S_i]P$ as the public key, where $i = 1, 2, \ldots, n$, n is the number of the CPCs.
Each CPC_i computes $Y_i = [S_i]Y_{i-1}$ as the shared public key. Specially, we define Y_n as Y. For example, the CPC_1 can compute Y_1 with the private key S_1 and the shared public key Y_0 of the KGC. We use this chain from the KGC to the CPC_n to compute the shared public key Y_i of each entity. It is important to emphasize that when the $\text{CPC}_{\text{i-1}}$ sends $\{Y_0, \ldots, Y_{i-1}\}$ to the CPC_i, then the CPC_i must verify the correctness of $\{Y_0, \ldots, Y_{i-1}\}$. Take the CPC_3 as an example. The CPC_3 verifies Y_2 by computing $e(Y_0, P_1)? = e(Y_1, P)$ and $e(Y_1, P_2)? = e(Y_2, P)$. System parameter of the CPC_i $param_2 = (P_i, Y_i)$ is public while the private key of the CPC_i S_i is private, where $i = 1, 2, \ldots, n$.

(3) *Key Authority Setup* $(\text{param}_1, \text{nu}) \longrightarrow (\text{param}_3, a, rl, \text{st})$. This algorithm is run by the Key Authority:

$$g = P, a \overset{R}{\leftarrow} \mathbb{Z}_p^*, g_1 = [a]g,\ g_2, h_1, h_2, h_3 \overset{R}{\leftarrow} G_1^*, \tag{3}$$

where nu is the number of users, rl is the initially empty revocation list, and st is the perfect binary tree with nu (nu is even) or $nu + 1$ (nu is odd) leaf nodes representing states. Let τ be the minimum time interval. System parameter of the Key Authority $\text{param}_3 = (g, g_1, g_2, h_1, h_2, h_3, \tau)$ is public, while the private key of the Key Authority a is private. In addition, the following two operations are defined just as [18]. For $x, i \in \mathbb{Z}$, set $J \subseteq \mathbb{Z}$, the *Lagrange coefficient* $\Delta_{i,J}(x)$ is defined as follows:

$$\Delta_{i,J}(x) = \prod_{j \in J, j \neq i} \left(\frac{x - j}{i - j} \right). \tag{4}$$

For $x \in \mathbb{Z}$, $J \subseteq \mathbb{Z}$, $g, h_1, \ldots, h_{|J|} \in G_1$,

$$H_{g,J,h_1,\ldots,h_{|J|}}(x) = \left([x^2]g \right) \prod_{i=1}^{|J|} \left(\left[\Delta_{i,J}(x) \right] h_i \right). \tag{5}$$

In the *Setup* step, the final public parameter is $\text{params} = (\text{param}_1, \text{param}_2, \text{param}_3)$.

Encryption Key Generation $EKG(\text{params}, \text{ID}_B, \text{time}) \longrightarrow (r, \text{RK}, ek, U, rc)$, including the following two algorithms:

(1) *Encryption RootInput Generation* $(\text{params}, ID_B) \longrightarrow (r, U, \text{RootInput})$. This algorithm is run by the sender:
Parse params as $(H_1, H_2, P, Y, q, G_1, e)$.

$$\begin{aligned} &Q_{ID_B} = H_1(ID_B) \in G_1^*, \\ &r \overset{R}{\leftarrow} \mathbb{Z}_q^*, U = [r]P, \text{RootInput} = H_2\left(\left(e\left(Q_{ID_B}, Y \right) \right)^r \right), \end{aligned} \tag{6}$$

where ID_B is the receiver's identity, *RootInput* is the input of the root chain in the Double Ratchet algorithm.

(2) *Encryption Revocation Key Generation* $(\text{params}, \text{ID}_B, \text{time}) \longrightarrow (rk, rc)$. This algorithm is run by the sender:

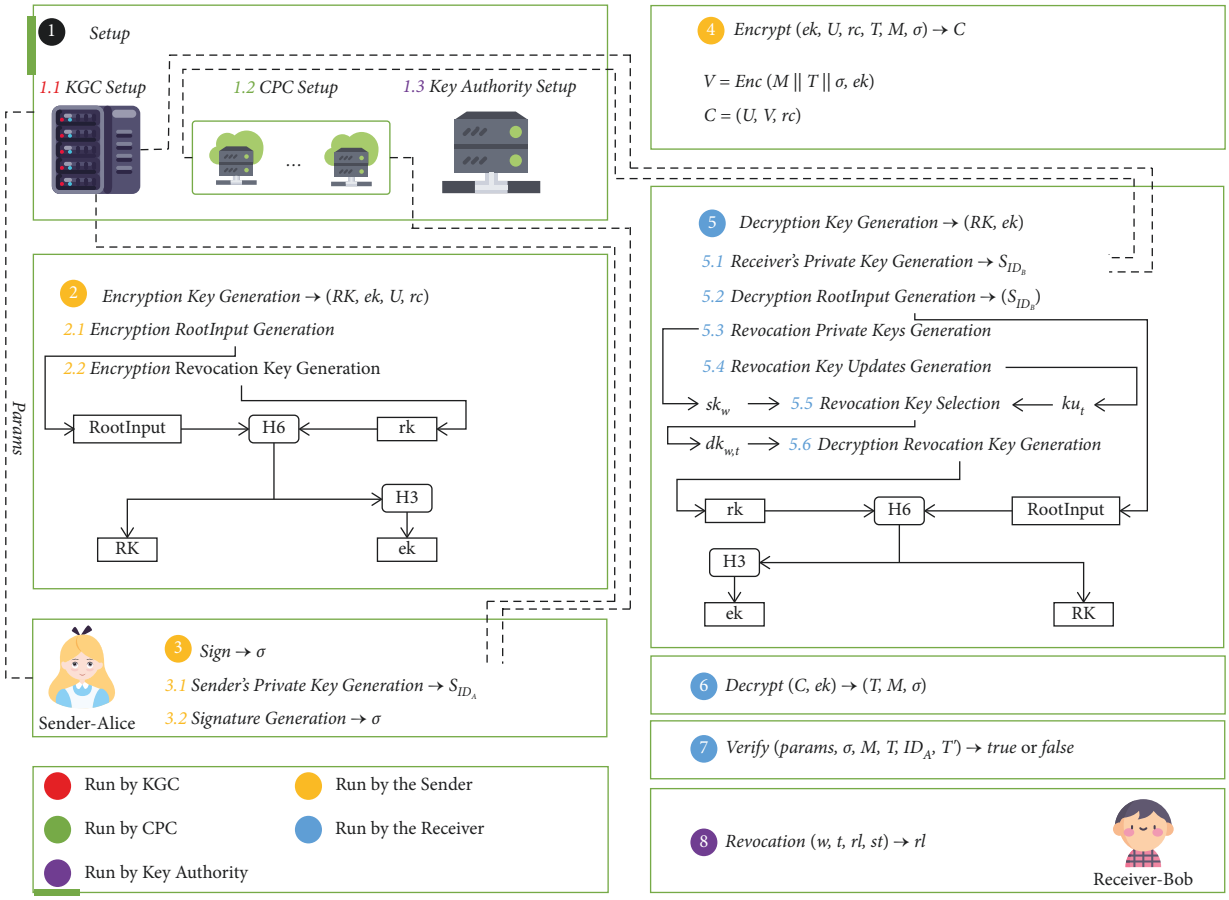

Figure 7: The *SESSION* part of the IBE-Signal scheme.

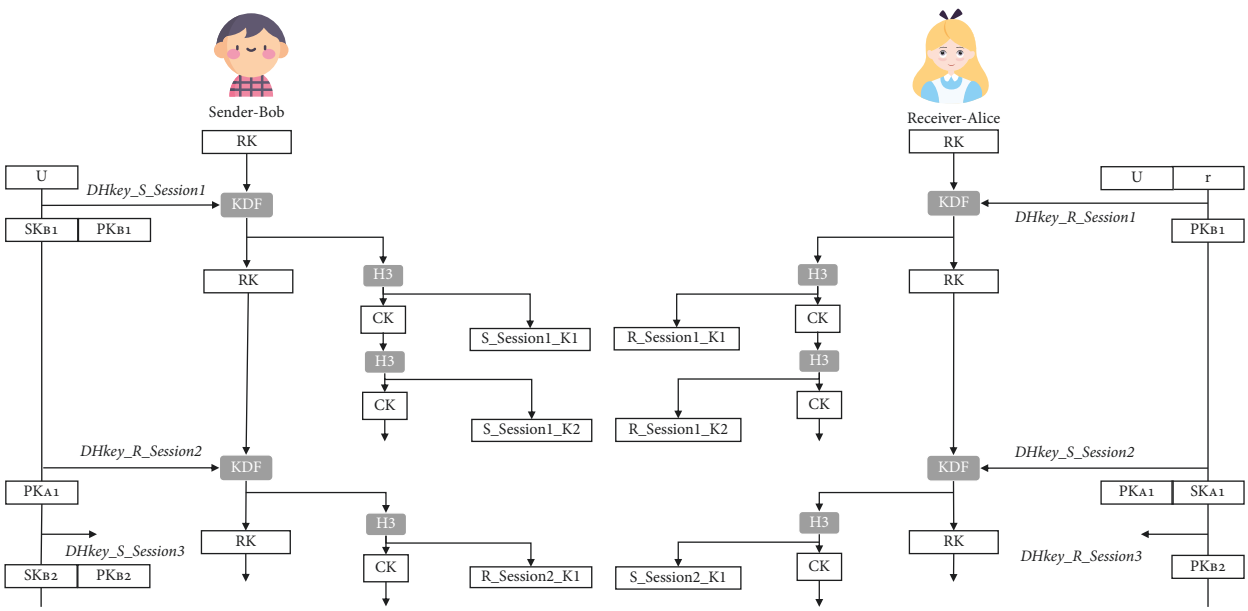

Figure 8: The *CONNECTION* part of the IBE-Signal scheme.

Table 2: Abbreviations and notations.

Abbreviations/notations	Meaning
k	Secure parameter
q	Big prime
G_1	Additive group on an elliptic curve of the order q
G_2	Multiplicative group on an elliptic curve of the order q
e: $G_1 \times G_1 \longrightarrow G_2$	Bilinear function
P, g, g_2, h_1, h_2, h_3	Generator of the group G_1
S_0, P_0, Y_0	The private key, the public key, and the shared public key of the KGC
$S_i, P_i, Y_i\,(i = 1, 2, ..., n)$	The private key, the public key, and the shared public key of the i^{th} CPC
Y	The shared public key of the $\mathrm{CPC_n}$ that equals to Y_n
H_1: $\{0,1\}^* \longrightarrow G_1^*$	A secure hash function that takes a string of any length as input and gives an element on G_1
H_2: $G_2 \longrightarrow \{0,1\}^k$	A secure hash function that takes an element on G_2 and gives a string of length k
H_3: $\{0,1\}^k \longrightarrow \{0,1\}^k$	A secure hash function that takes a string of length k and gives a string of length k
H_4: $\{0,1\}^* \longrightarrow \mathbb{Z}_q^*$	A secure hash function that takes a string of any length and gives an element on $\mathbb{Z}_q^*$
H_5: $\{0,1\}^* \times G_2^* \longrightarrow \mathbb{Z}_q^*$	A secure hash function that takes a string of any length and an element on G_2 and gives an element on $\mathbb{Z}_q^*$
H_6: $\{0,1\}^k \times \{0,1\}^k \longrightarrow \{0,1\}^k$	A secure hash function that takes two strings of length k and gives a string of length k
KDF: $G_1 \times \{0,1\}^k \longrightarrow \{0,1\}^k$	Secure key derivation function that takes an element on G_1 and a string of length k and gives a string of length k
nu	Number of users
rl	Initially empty revocation list
st	A binary tree with at least nu leaf nodes representing states
τ	Minimum time interval
a	The private key of key authority
g_1	Part of the system parameter of key authority and $g_1 = [a]g$
$\mathrm{param_1}$	System parameter of the KGC
$\mathrm{param_2}$	System parameter of the CPCs
$\mathrm{param_3}$	System parameter of key authority
params	The final public parameter in the setup step
$ID_B \in \{0,1\}^*$	Identity of the receiver
$ID_A \in \{0,1\}^*$	Identity of the sender
$Q_{ID_B} \in G_1^*$	Mapping of ID_B on G_1
$Q_{ID_A} \in G_1^*$	Mapping of ID_A on G_1
$S_{ID_B} \in G_1^*$	The private key of the receiver
$S_{ID_A} \in G_1^*$	The private key of the sender
ω	Mapping of ID_B on $\mathbb{Z}_q^*$
time	The time when the encryption occurs
t	Mapping of time on $\mathbb{Z}_q^*$
RootInput	The input of the root chain in the double ratchet algorithm
rk	Revocation key
RK	The input of the root chain in the *CONNECTION* part of the IBE-signal scheme
ek	Message key for the encryption of the first message
x	Blinding factor in the private key issuance process
T	Timestamp used to prevent replay attacks
T'	Receiver's local time
M	The message that needs to be signed
σ	Signature
sk_ω	Private keys for revocation
ku_t	Key updates for revocation
$dk_{\omega,t}$	Revocation key selected from sk_ω and ku_t
PK_{Bi}, SK_{Bi}	Bob's public and private keys in Diffie–Hellman key exchange in *CONNECTION* part
$\mathrm{PK_{Ai}}, \mathrm{SK_{Ai}}$	Alice's public and private keys in Diffie–Hellman key exchange in *CONNECTION* part
$S\_\mathrm{Session}_i\_K_j$	The j^{th} message key of the sender in the i^{th} session
$R\_\mathrm{Session}_i\_K_j$	The j^{th} message key of the receiver in the i^{th} session

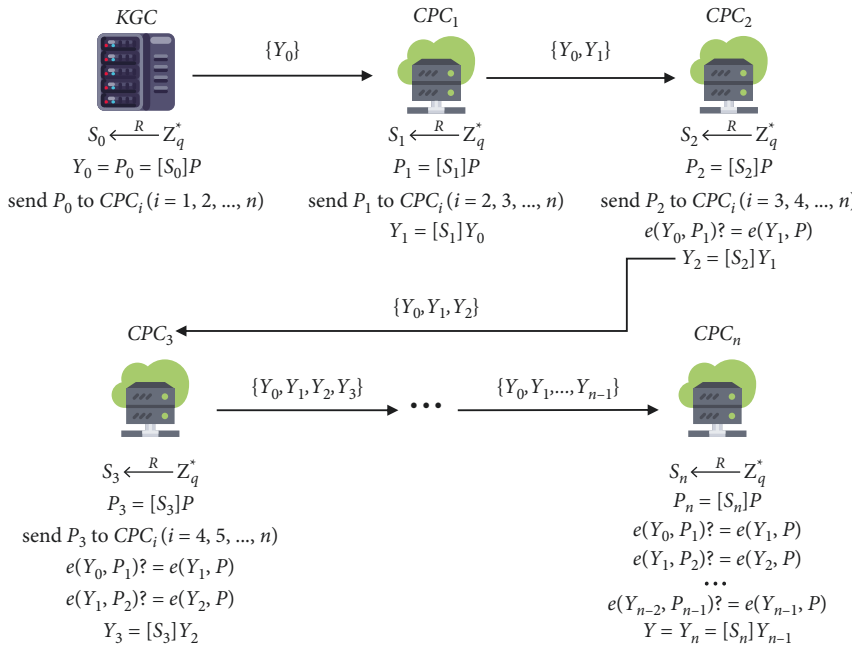

Figure 9: *KGC Setup and CPC Setup of the IBE-Signal scheme.*

Parse params as $(H_3, H_4, H_6, q, k, g, g_1, g_2, h_1, h_2, h_3, e)$.

$$\omega = H_4(ID_B) \in Z_q^*, t = H_4(\text{time}) \in Z_q^*, rk \xleftarrow{R} \{0,1\}^k, z \xleftarrow{R} Z_q^*,$$
$$c_1 = rk \cdot (e(g_1, g_2))^z, c_2 = [z]g,$$
$$c_\omega = [z]H_{g_2,J,h_1,h_2,h_3}(\omega), c_t = [z]H_{g_2,J,h_1,h_2,h_3}(t), \tag{7}$$

where time is the time when the encryption occurs and rk is the revocation key. Let J be $\{1, 2, 3\}$ and rc be $(\omega, t, c_\omega, c_t, c_1, c_2)$.

In the *Encryption Key Generation* step, the final encryption key ek:

$$RK = H_6(\text{RootInput}, rk), ek = H_3(RK). \tag{8}$$

Sign $SI(\text{params}, S_i, ID_A, T, M) \longrightarrow \sigma$, including the following two algorithms:

(1) *Sender's Private Key Generation* $(\text{params}, S_i, ID_A) \longrightarrow S_{ID_A}$, where $i = 0, 1, \ldots, n$. The sender runs this algorithm and interacts with the KGC and the CPCs, as shown in Figure 10.

Parse params as (q, e, P, Y, H_1).

The sender does the following steps: $x \xleftarrow{R} Z_q^*$, $X = [x]P$, $Q_{ID_A} = H_1(ID_A)$, $D_{ID_A} = [x]Q_{ID_A}$, where ID_A is the sender's identity and x is the blinding factor. We use the blind signature to protect the sender's partial private key from being obtained by an attacker or server. The sender sends (ID_A, D_{ID_A}, X) to the KGC. It should be noted that the sender must go through the corresponding identity authentication process, such as password authentication, to get the signed private key, whether interacting with the KGC or the CPCs.

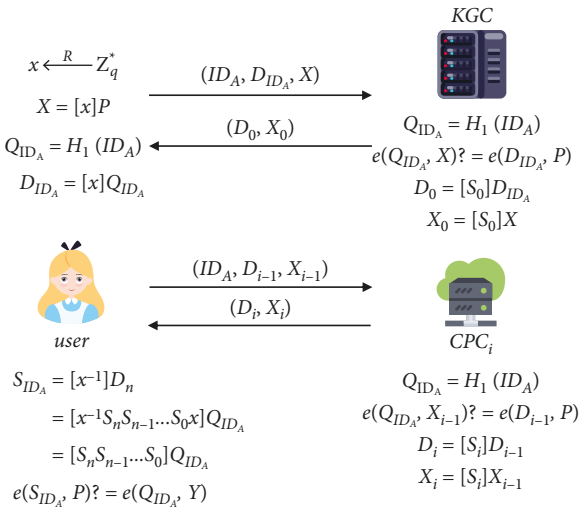

Figure 10: *Sender's Private Key Generation of the IBESignal scheme.*

Since this is not our focus, this paper has no specific design.

The KGC and the CPCs do the following steps: the KGC computes $Q_{ID_A} = H_1(ID_A)$ and verifies the correctness of $e(Q_{ID_A}, X)? = e(D_{ID_A}, P)$. This ensures that the KGC is issuing the partial private key to ID_A. The KGC computes $D_0 = [S_0]D_{ID_A}$, $X_0 = [S_0]X$ and sends them to the sender. The sender sends (ID_A, D_{i-1}, X_{i-1}) to the CPCi, the CPCi computes $Q_{ID_A} = H_1(ID_A)$ and verifies the correctness of $e(Q_{ID_A}, X_{i-1})? = e(D_{i-1}, P)$, then calculates $D_i = [S_i]D_{i-1}$, $X_i = [S_i]X_{i-1}$ and sends them to the sender where $i = 1, 2, \ldots, n$. After the iterations, the sender can get $D_n = [S_nS_{n-1}...S_0x]Q_{ID_A}$.

The sender extracts the private key by computing $S_{ID_A} = [x^{-1}]D_n$, then verifies the correctness of $e(S_{ID_A}, P)? = e(Q_{ID_A}, Y)$.

(2) *Signature Generation* $(params, T, M, S_{ID_A}) \longrightarrow \sigma$. This algorithm is run by the sender:
Parse params as (q, e, G_1, P, H_5).

$$b \xleftarrow{R} \mathbb{Z}_q^*, O \xleftarrow{R} G_1^*, R = (e(O,P))^b,$$
$$y = H_5(M \| T, R), F = [y]S_{ID_A} + [b]O, \sigma = (F, y), \tag{9}$$

where T is the timestamp used to prevent replay attacks, M is the message that needs to be signed, and σ is the signature. In the *SESSION* part, M can be a fixed format message shown in Figure 11, such as "Hi! I'm Alice." When Alice friends Bob, a notification is automatically sent when the initial session is established.

Encrypt $E(ek, U, rc, T, M, \sigma) \longrightarrow C$. This algorithm is run by the sender:

Let *Enc* be a secure symmetric encryption algorithm.

$$V = \mathrm{Enc}(M \| T \| \sigma, ek), C = (U, V, rc), \tag{10}$$

where C is the ciphertext sent to the receiver.

Decryption Key Generation DKG $(\mathrm{params}, a, S_i, ID_B, \mathrm{time}, C, st, rl) \longrightarrow (RK, ek)$, including the following six algorithms:

(1) *Receiver's Private Key Generation* $(\mathrm{params}, S_i, ID_B) \longrightarrow S_{ID_B}$. The receiver runs this algorithm and interacts with the KGC and the CPCs: The function is like *Sender's Private Key Generation*, so we will not go over it here.

(2) *Decryption RootInput Generation* $(\mathrm{params}, S_{ID_B}, C) \longrightarrow \mathrm{RootInput}$. This algorithm is run by the receiver:
Parse params as (H_2, e) and C as (U, V, rc).

$$\mathrm{RootInput} = H_2\left(e\left(S_{ID_B}, U\right)\right). \tag{11}$$

(3) *Revocation Private Keys Generation* $(\mathrm{params}, a, ID_B, st) \longrightarrow (sk_\omega, st)$. This algorithm is run by the Key Authority:
Parse *params* as $(q, H_4, g, g_2, h_1, h_2, h_3)$.

$$\omega = H_4(ID_B) \in \mathbb{Z}_q^*. \tag{12}$$

Select an empty leaf node v from st and store ω in it. In addition, the *Path* function is defined as follows: the set of all nodes on the path from the leaf node to the root node, including that leaf node and the root node.

(i) $\forall i \in \mathrm{Path}(v)$
(ii) *if* a_i *is undefined*,
(iii) *then* $a_i \xleftarrow{R} \mathbb{Z}_{q_R}^*$ *and store* a_i *in node* i,
(iv) *compute* $r_i \xleftarrow{R} \mathbb{Z}_q^*, d_i = [r_i]g$,
(v) $D_i = ([a_i\omega + a]g_2) + ([r_i]H_{g_2,J,h_1,h_2,h_3}(\omega))$

After the above code is executed, we get the *private keys* for revocation $sk_\omega = \{(i, d_i, D_i)\}_{i \in \mathrm{Path}(v)}$ and the updated state tree st.

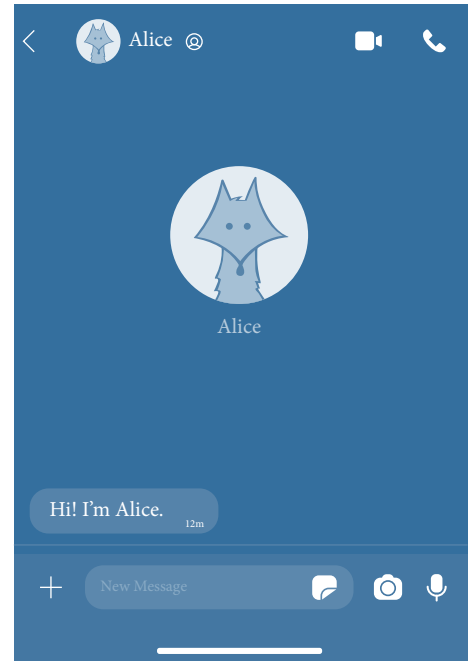

FIGURE 11: *In the signal application, messages are delivered when a session is established.*

(4) *Revocation Key Updates Generation* $(\mathrm{params}, a, \mathrm{time}, rl, st) \longrightarrow ku_t$. This algorithm is run by the Key Authority:
Parse params as (g, g_2, h_1, h_2, h_3).

$$t = H_4(\mathrm{time}) \in \mathbb{Z}_q^*. \tag{13}$$

The *KUNodes* function is the same as in [18] and is shown in Figure 12:
$\mathrm{KUNodes}(st, rl, t)$:

(i) $X, Y \leftarrow \varnothing$,
(ii) $\forall (v_i, t_i) \in rl$
(iii) *if* $t_i \le t$ *then add* $\mathrm{Path}(v_i)$ *to* X
(iv) $\forall x \in X$
(v) *if* $x_l \notin X$ *then add* x_l *to* Y
(vi) *if* $x_r \notin X$ *then add* x_r *to* Y
(vii) *if* $Y = \varnothing$ *then add* root *to* Y
(viii) *return* Y

The following code is executed for the nodes that are output by the *KUNodes* function.

(i) $\forall j \in \mathrm{KUNodes}(st, rl, t)$
(ii) *compute* $r_j \xleftarrow{R} \mathbb{Z}_q^*, e_j = [r_j]g$,
(iii) $E_j = ([a_j t + a]g_2) + ([r_j]H_{g_2,J,h_1,h_2,h_3}(t))$

Then we get the set of the *key updates* for revocation $ku_t = \{(j, e_j, E_j)\}_{j \in KUNodes(st,rl,t)}$.

(5) *Revocation Key Selection* $(sk_\omega, ku_t) \longrightarrow dk_{\omega,t}$ *or* $\perp$. This algorithm is run by the receiver:
Parse sk_ω as $\{(i, d_i, D_i)\}_{i \in \mathrm{Path}(v)}$ and ku_t as $\{(j, e_j, E_j)\}_{j \in \mathrm{KUNodes}(st,rl,t)}$.

(i) $\forall (i, d_i, D_i) \in sk_\omega, (j, e_j, E_j) \in ku_t$
(ii) *if* $i = j$ *then* $dk_{\omega,t} = (d_i, D_i, e_j, E_j)$
(iii) *else* $dk_{\omega,t} = \perp$

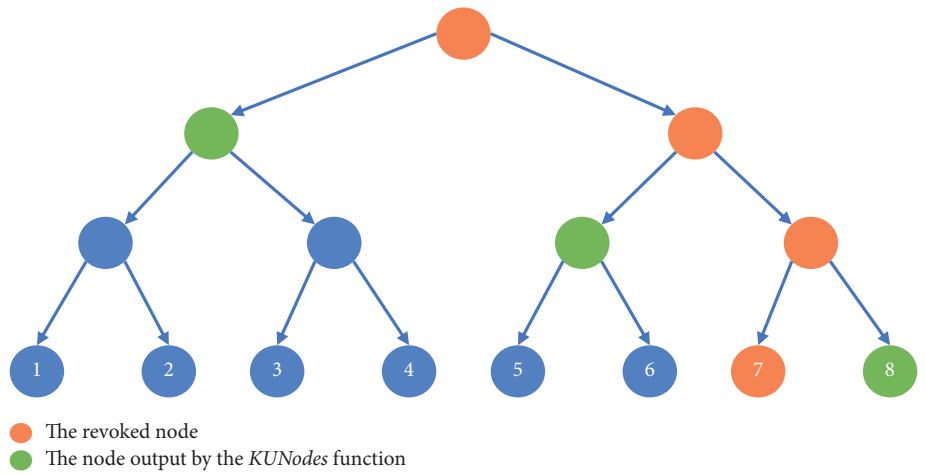

Figure 12: The *KUNodes* tree when node #7 is revoked.

(iv) *return* $dk_{\omega,t}$.

(6) *Decryption Revocation Key Generation* $(dk_{\omega,t}, C) \longrightarrow rk$. This algorithm is run by the receiver:

Parse $dk_{\omega,t}$ as (d_i, D_i, e_i, E_i), C as (U, V, rc) and rc as $(\omega, t, c_\omega, c_t, c_1, c_2)$.

$$rk = c_1\left(\frac{e(d_i, c_\omega)}{e(D_i, c_2)}\right)^{t/t-\omega}\left(\frac{e(e_i, c_t)}{e(E_i, c_2)}\right)^{\omega/\omega-t}. \tag{14}$$

In the *Decryption Key Generation* step, the final decryption key:

$$\mathrm{RK} = H_6(\mathrm{RootInput}, \mathrm{rk}), \mathrm{ek} = H_3(\mathrm{RK}). \tag{15}$$

Decrypt $D(C, ek) \longrightarrow (T, M, \sigma)$. This algorithm is run by the receiver:

Parse C as (U, V, rc), V as $\mathrm{Enc}(M\|T\|\sigma, ek)$. Let *Dec* be a secure symmetric decryption algorithm.

$$M||T||\sigma = Dec(V, ek). \tag{16}$$

Verify $V(\mathrm{params}, \sigma, M, T, ID_A, T') \longrightarrow$ true or false. This algorithm is run by the receiver:

Parse params as $(e, P, Y, H_1, H_5, \tau)$, σ as (F, y). Let T' be the receiver's local time.

$$Q_{ID_A} = H_1(ID_A), R = e(F, P)\left(e\left(Q_{ID_A}, -Y\right)\right)^y. \tag{17}$$

(i) *if* $H_5(M\|T, R) == y \&\& |T' - T| \le \tau$ *then return* true

(ii) *else return* false.

Revocation $R(\omega, t, rl, st) \longrightarrow rl$. This algorithm is run by the Key Authority:

(i) $\forall v \in \{leaf\ nodes\ in\ st\ with\ \omega\}$

(ii) *add* (v, t) *to* rl

(iii) *return* rl.

3.3.2. CONNECTION Part. In the *CONNECTION* part, we show how the first two messages are sent and received in the first session.

Sender Key Generation $SKG(\mathrm{params}, C, RK) \longrightarrow (S\_Session1\_K1, S\_Session1\_K2, PK_{B1}, RK)$:

Parse params as $(q, P, \mathrm{KDF}, H_3)$, C as (U, V, rc), U as $[r]P$,

$$\begin{aligned} SK_{B1} &\xleftarrow{R} \mathbb{Z}_q^*, PK_{B1} = [SK_{B1}]P, \\ \mathrm{DHkey\_S\_Session1} &= [SK_{B1}]U = [rSK_{B1}]P, \\ RK &= \mathrm{KDF}(\mathrm{DHkey\_S\_Session1}, RK), \\ S\_Session1\_K1 &= H_3(RK), \\ S\_Session1\_K2 &= H_3(S\_Session1\_K1). \end{aligned} \tag{18}$$

Encrypt $E(M_1, M_2, S\_Session1\_K1, S\_Session1\_K2, PK_{B1}) \longrightarrow (C_1, C_2)$:

Let *Enc* be a secure symmetric encryption algorithm.

$$\begin{aligned} C_1 &= (PK_{B1}, \mathrm{Enc}(M_1, S\_Session1\_K1)), \\ C_2 &= (PK_{B1}, \mathrm{Enc}(M_2, S\_Session1\_K2)). \end{aligned} \tag{19}$$

Receiver Key Generation $RKG(C_1, r, RK) \longrightarrow (R\_Session1\_K1, R\_Session1\_K2, RK)$:

Parse C_1 as $(PK_{B1}, \mathrm{Enc}(M_1, S\_Session1\_K1))$,

$$\begin{aligned} \mathrm{DHkey\_R\_Session1} &= [r]PK_{B1} = [rSK_{B1}]P, \\ RK &= \mathrm{KDF}(\mathrm{DHkey\_R\_Session1}, RK), \\ R\_Session1\_K1 &= H_3(RK), \\ R\_Session1\_K2 &= H_3(R\_Session1\_K1). \end{aligned} \tag{20}$$

Decrypt $D(C_1, C_2, R\_Session1\_K1, R\_Session1\_K2) \longrightarrow (M_1, M_2)$:

Let *Dec* be a secure symmetric encryption algorithm.

$$\begin{aligned} M_1 &= \mathrm{Dec}(\mathrm{Enc}(M_1, S\_Session1\_K1), R\_Session1\_K1), \\ M_2 &= \mathrm{Dec}(\mathrm{Enc}(M_2, S\_Session1\_K2), R\_Session1\_K2). \end{aligned} \tag{21}$$

4. Security Features' Analysis

This paper selects two features to analyze in detail: resisting MITM attacks and supporting public key revocation.

4.1. Our IBE-Signal Scheme Can Resist MITM Attack. We discussed that MITM attacks cannot be implemented in both the *SESSION* part and the *CONNECTION* part.

In the *SESSION* part, mutual authentication exists.

To indicate the existence of mutual authentication in the *SESSION* part, we show a simplified version of our IBE-Signal protocol in Figure 13. Based on the characteristics of the IBE scheme, Alice can get Bob's public key Q_{ID_B} directly from Bob's ID and then encrypt the message M by the public key Q_{ID_B}. $[r]P$ is a random challenge to Bob's identity. The only way to get $S\_Session1\_K1$ is to have Bob's private key, which encrypts message M_1 as a response to the challenge $[r]P$. This completes the authentication of Bob and is resistant to replay attacks. Authentication of Alice is done by signature and timestamp and prevents replay attacks. A MITM attack is not possible when mutual authentication is established.

In the *CONNECTION* part, launching a MITM attack is ineffective.

Since we use the Diffie–Hellman ratchet in the *CONNECTION* part, an attacker can launch a MITM attack to hijack the session, as shown in Figure 14. However, both parties have a negotiated *RK* in the *SESSION* part,

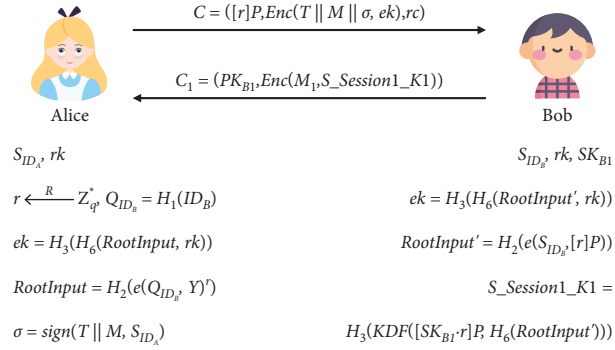

Figure 13: The simplified version of our IBE-Signal scheme.

while the attacker has no *RK*. The attacker cannot calculate *K1* and *K2* and naturally cannot send the message $(PK_{T1}, \mathrm{Enc}(M_1, K2))$.

4.2. Our IBE-Signal Scheme Supports Public Key Revocation. Once the user's public key expires, the Key Authority will add it to *rl*, and the *KUNodes* function will output a new point set. As shown in Figure 12, the set consisting of red dots is $\{(i, d_i, D_i)\}_{i \in \mathrm{Path}(v)}$ and the set consisting of green dots is $\{(j, e_j, E_j)\}_{j \in KU\mathrm{Nodes}(st,rl,t)}$. The two sets have no intersection and thus cannot generate the revocation key $dk_{\omega,t}$.

Assume that the encryption time is t and the expiration time is t'. Formula 22 shows that the attacker cannot get rk using the old revocation key $dk_{\omega,t}$, so the initial session cannot be established.

$$
\begin{aligned}
c_1\left(\frac{e(d_i,c_\omega)}{e(D_i,c_2)}\right)^{t'/t'-\omega}\left(\frac{e(e_i,c_t)}{e(E_i,c_2)}\right)^{\omega/\omega-t'} &= rk\cdot(e(g_1,g_2))^z\left(\frac{e\left([r_i]g,[z]H_{g_2,J,h_1,h_2,h_3}(\omega)\right)}{e\left(([a_i\omega+a]g_2)+\left([r_i]H_{g_2,J,h_1,h_2,h_3}(\omega)\right),[z]g\right)}\right)^{t'/t'-\omega}\\
&\quad\cdot\left(\frac{e\left([r_i]g,[z]H_{g_2,J,h_1,h_2,h_3}(t)\right)}{e\left(([a_it'+a]g_2)+\left([r_i]H_{g_2,J,h_1,h_2,h_3}(t')\right),[z]g\right)}\right)^{\omega/\omega-t'}\\
&= rk\cdot(e(g_1,g_2))^z\left(\frac{e\left([r_i]g,[z]H_{g_2,J,h_1,h_2,h_3}(\omega)\right)}{e(([a_i\omega+a]g_2),[z]g)\cdot e\left(\left([r_i]H_{g_2,J,h_1,h_2,h_3}(\omega)\right),[z]g\right)}\right)^{t'/t'-\omega}\\
&\quad\cdot\left(\frac{e\left([r_i]g,[z]H_{g_2,J,h_1,h_2,h_3}(t)\right)}{e(([a_it'+a]g_2),[z]g)\cdot e\left(\left([r_i]H_{g_2,J,h_1,h_2,h_3}(t')\right),[z]g\right)}\right)^{\omega/\omega-t'}\\
&= rk\cdot(e(g_1,g_2))^z\left(\frac{1}{e(([a_i\omega+a]g_2),[z]g)}\right)^{t'/t'-\omega}\\
&\quad\cdot\left(\frac{e\left([r_i]g,[z]H_{g_2,J,h_1,h_2,h_3}(t)\right)}{e(([a_it'+a]g_2),[z]g)\cdot e\left(\left([r_i]H_{g_2,J,h_1,h_2,h_3}(t')\right),[z]g\right)}\right)^{\omega/\omega-t'}\\
&\neq rk.
\end{aligned}
\tag{22}
$$

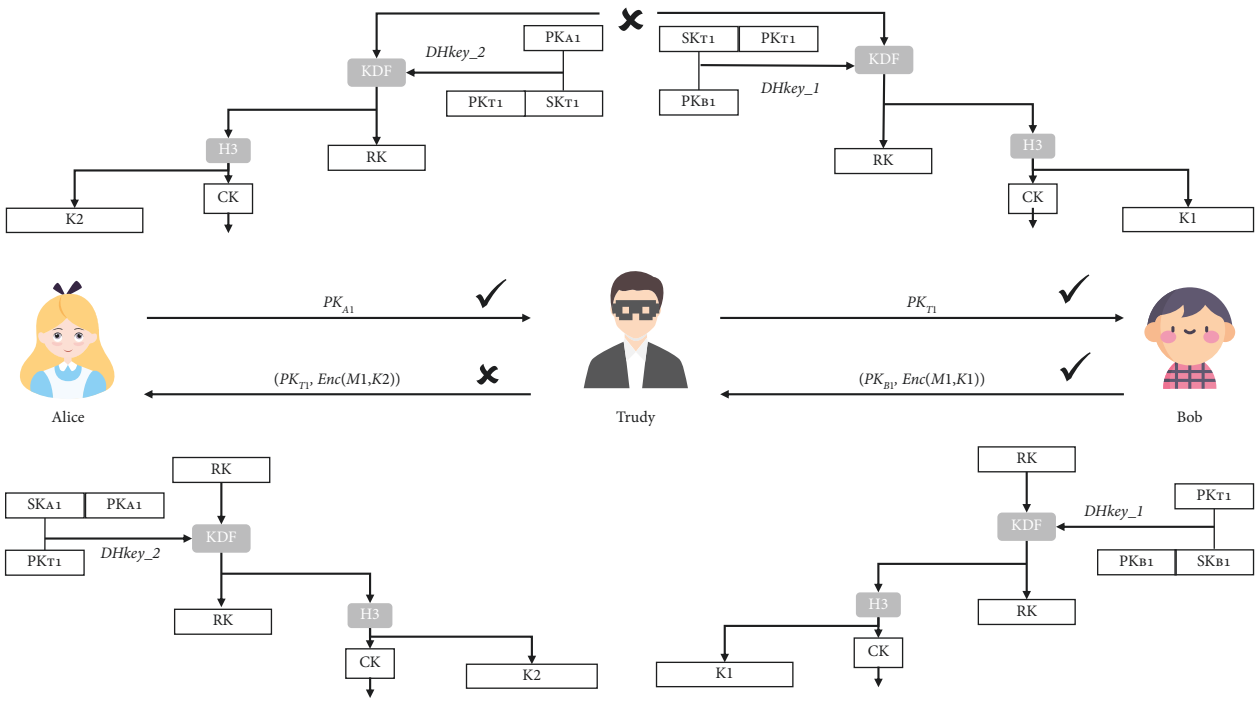

Figure 14: A MITM attack in the *CONNECTION* part.

5. Security Proof

We first prove the correctness of the scheme, then introduce the security model, and subsequently show that the scheme is IND-ID-CPA secure.

Theorem 1. *Our IBE-Signal scheme satisfied the correctness property.*

Proof. To facilitate the proof, we distinguish the corresponding recipient symbols by adding "′", e.g. *RootInput* and *RootInput*′. □

Lemma 1. *The SESSION part satisfied the correctness property.*

Proof

$$\begin{aligned}\text{RootInput}' &= H_2\left(e\left(S_{ID_B}, U\right)\right) = H_2\left(e\left(\left[S_0 \ldots S_n\right]Q_{ID_B}, [r]P\right)\right)\\ &= H_2\left(e\left(Q_{ID_B}, \left[S_0 \ldots S_n\right]P\right)^r\right) = H_2\left(e\left(Q_{ID_B}, Y\right)^r\right)\\ &= \text{RootInput}\end{aligned}$$

$$rk' = c_1\left(\frac{e\left(d_i, c_\omega\right)}{e\left(D_i, c_2\right)}\right)^{t/t-\omega}\left(\frac{e\left(e_i, c_t\right)}{e\left(E_i, c_2\right)}\right)^{\omega/\omega-t}$$

$$= rk \cdot e\left(g_1, g_2\right)^z \cdot \left(\frac{e\left([r_i]g, [z]H_{g_2,J,h_1,h_2,h_3}(\omega)\right)}{e\left(\left([a_i\omega + a]g_2\right) + \left([r_i]H_{g_2,J,h_1,h_2,h_3}(\omega)\right), [z]g\right)}\right)^{t/t-\omega}$$

$$\cdot \left(\frac{e\left([r_i]g, [z]H_{g_2,J,h_1,h_2,h_3}(t)\right)}{e\left(\left([a_it + a]g_2\right) + \left([r_i]H_{g_2,J,h_1,h_2,h_3}(t)\right), [z]g\right)}\right)^{\omega/\omega-t}$$

$$
\begin{aligned}
&= rk \cdot e(g_1, g_2)^z \cdot \left(\frac{e\left([r_i]g, [z]H_{g_2,J,h_1,h_2,h_3}(\omega)\right)}{e(([a_i\omega + a]g_2), [z]g) \cdot e\left(\left([r]_i H_{g_2,J,h_1,h_2,h_3}(\omega)\right), [z]g\right)} \right)^{t/t-\omega} \\
&\cdot \left(\frac{e\left([r_i]g, [z]H_{g_2,J,h_1,h_2,h_3}(t)\right)}{e(([a_i t + a]g_2), [z]g) \cdot e\left(\left([r_i]H_{g_2,J,h_1,h_2,h_3}(t)\right), [z]g\right)} \right)^{\omega/\omega-t} \\
&= rk \cdot e(g_1, g_2)^z \cdot \left(\frac{1}{e(([a_i\omega + a]g_2), [z]g)} \right)^{t/t-\omega} \cdot \left(\frac{1}{e(([a_i t + a]g_2), [z]g)} \right)^{\omega/\omega-t} \\
&= rk \cdot e(g_1, g_2)^z \cdot \left(\frac{1}{e([a_i\omega]g_2, [z]g) \cdot e([a]g_2, [z]g)} \right)^{t/t-\omega} \\
&\quad \cdot \left(\frac{1}{e([a_i t]g_2, [z]g) \cdot e([a]g_2, [z]g)} \right)^{\omega/\omega-t} \\
&= rk \cdot e(g_1, g_2)^z \cdot \left(\frac{1}{e([a_i]g_2, [z]g)^{\omega t/t-\omega} \cdot e([a]g_2, [z]g)^{t/t-\omega}} \right) \\
&\quad \cdot \left(\frac{1}{e([a_i]g_2, [z]g)^{t\omega/\omega-t} \cdot e([a]g_2, [z]g)^{\omega/\omega-t}} \right) \\
&= rk \cdot e(g_1, g_2)^z \cdot \left(\frac{1}{e([a_i]g_2, [z]g)^{\omega t/t-\omega} \cdot e([a_i]g_2, [z]g)^{t\omega/\omega-t}} \right) \\
&\quad \cdot \left(\frac{1}{e([a]g_2, [z]g)^{t/t-\omega} \cdot e([a]g_2, [z]g)^{\omega/\omega-t}} \right) \\
&= rk \cdot e(g_1, g_2)^z \cdot \left(\frac{1}{e([a]g_2, [z]g)} \right) = rk \cdot e(g_1, g_2)^z \cdot \left(\frac{1}{e(g_2, [a]g)^z} \right) \\
&= rk \cdot e(g_1, g_2)^z \cdot \left(\frac{1}{e(g_2, g_1)^z} \right) = rk
\end{aligned}
$$

$$
\begin{aligned}
M\|T\|\sigma &= \mathrm{Dec}(V, ek') \\
&= \mathrm{Dec}\left(\mathrm{Enc}(M\|T\|\sigma, ek), H_3(H_6(\mathrm{RootInput}', rk'))\right) \\
&= \mathrm{Dec}\left(\mathrm{Enc}(M\|T\|\sigma, ek), H_3(H_6(\mathrm{RootInput}, rk))\right) \\
&= \mathrm{Dec}(\mathrm{Enc}(M\|T\|\sigma, ek), ek) \\
&= M\|T\|\sigma.
\end{aligned} \tag{23}
$$

□

Lemma 2. *The CONNECTION part satisfied the correctness property.*

Proof

$$
\begin{aligned}
M_1 &= \mathrm{Dec}\left(\mathrm{Enc}(M_1, S\_\mathrm{Session1}\_K1), R\_\mathrm{Session1}\_K1\right) \\
&= \mathrm{Dec}\left(\mathrm{Enc}(M_1, S\_\mathrm{Session1}\_K1), H_3(\mathrm{KDF}(\mathrm{DHkey}\_R\_\mathrm{Session1}, RK))\right) \\
&= \mathrm{Dec}\left(\mathrm{Enc}(M_1, S\_\mathrm{Session1}\_K1), H_3(\mathrm{KDF}([r]PK_{B1}, RK))\right) \\
&= \mathrm{Dec}\left(\mathrm{Enc}(M_1, S\_\mathrm{Session1}\_K1), H_3(\mathrm{KDF}([rSK_{B1}]P, RK))\right) \\
&= \mathrm{Dec}\left(\mathrm{Enc}(M_1, S\_\mathrm{Session1}\_K1), H_3(\mathrm{KDF}([SK_{B1}]([r]P), RK))\right) \\
&= \mathrm{Dec}\left(\mathrm{Enc}(M_1, S\_\mathrm{Session1}\_K1), H_3(\mathrm{KDF}([SK_{B1}]U, RK))\right) \\
&= \mathrm{Dec}\left(\mathrm{Enc}(M_1, S\_\mathrm{Session1}\_K1), H_3(\mathrm{KDF}(\mathrm{DHkey}\_S\_\mathrm{Session1}, RK))\right) \\
&= \mathrm{Dec}\left(\mathrm{Enc}(M_1, S\_\mathrm{Session1}\_K1), S\_\mathrm{Session1}\_K1\right) \\
&= M_1.
\end{aligned} \tag{24}
$$

□

Lemma 3. *The Signe&Verify part satisfied the correctness property.*

Proof

$$\begin{aligned} y' &= H_5(M\|T, R') \\ &= H_5\left(M\|T, e(F,P)e\left(Q_{ID_A}, -Y\right)^y\right) \\ &= H_5\left(M\|T, e\left([y]S_{ID_A} + [b]O, P\right)e\left(Q_{ID_A}, -Y\right)^y\right) \\ &= H_5\left(M\|T, e\left([y]S_{ID_A}, P\right)e([b]O, P)e\left(Q_{ID_A}, -Y\right)^y\right) \\ &= H_5\left(M\|T, e\left([S_0 \ldots S_n]Q_{ID_A}, P\right)^y e(O,P)^b e\left(Q_{ID_A}, -Y\right)^y\right) \\ &= H_5\left(M\|T, e\left(Q_{ID_A}, Y\right)^y e(O,P)^b e\left(Q_{ID_A}, -Y\right)^y\right) \\ &= H_5\left(M\|T, e(O,P)^b\right) \\ &= H_5(M\|T, R) \\ &= y. \end{aligned} \tag{25}$$

□

5.1. Security Model. It should be noted that the proof of the IND-sRID-CPA security of the RIBE scheme is given in [18], so the revocation key rk is provided directly by the challenger in the *Setup* phase of our game. The next game describes the security model of our scheme:

Setup. The challenger inputs a secure parameter κ and outputs public parameters *params*, revocation key rk, and private keys a and S_i where $i = 0, 1, 2, \ldots, n$. The challenger sends *params* and rk to the adversary. The challenger controls the following queries and can be queried by the adversary.

H_1 *queries*: Given identity ID, the challenger runs these queries and gives Q_{ID}.
H_2 *queries*: Given $F_j \in G_2$, the challenger runs these queries and gives $H_j \in \{0,1\}^\kappa$.

Phase 1. The adversary chooses an ID and sends queries of partial private keys of the ID. The challenger generates the partial private keys $\left\{S_{ID_i} = [\prod_{h=0}^{i} S_h]Q_{ID}\right\}_{i=0,1,\ldots,n.}$ and then sends them to the adversary.

Challenge. The adversary outputs two plaintexts M_0, M_1 of equal length and the identity ID^* intentionally to challenge. The only restriction is that ID^* does not appear in any query in *Phase 1*. The adversary sends M_0, M_1 and ID^* to the challenger. The challenger randomly chooses k as the honest CPC's serial number and generates the partial private key $S_{ID_i^*} = [\prod_{h=0, h\neq k}^{i} S_h]Q_{ID^*}$. The challenger randomly chooses a bit $\beta \xleftarrow{R} \{0,1\}$ and computes $C^* = \mathrm{Enc}(M_\beta, S_{ID^*})$ where $S_{ID^*} = [\prod_{h=0}^{n} S_h]Q_{ID^*}$ then sends $\left\{S_{ID_i^*}\right\}_{i=0,1,\ldots,n,i\neq k}$ and C^* to the adversary.

Phase 2. The adversary sends queries of partial private keys of $ID\,(ID \neq ID^*)$ to the challenger. The challenger responds in the way of *Phase 1*.

Guess. The adversary outputs his guess $\beta' \in \{0,1\}$. If $\beta' = \beta$, the adversary attacks the IBE-Signal scheme successfully.

The adversary's advantage is defined as the function of the safe parameter κ:

$$Adv_{A,\mathrm{IBE-Signal}}^{\mathrm{IND-ID-CPA}}(\kappa) = \left|\Pr[\beta' = \beta] - \frac{1}{2}\right|. \tag{26}$$

Theorem 2. *Assume H1 and H2 are random oracle models, and the BDH problem is infeasible to solve; our IBE-Signal scheme is semantically secure against IND-ID-CPA attack.*

Proof. Expressly, assume the adversary A can run at most $q_E > 0$ partial private key queries and $q_{H_2} > 0$ H_2 queries. Assume A can break our scheme in polynomial time with probability $\varepsilon(\kappa)$; there must be an adversary B that can solve the BDH problem with advantage at least $\varepsilon'(\kappa)$, where $\varepsilon'(\kappa) \geq 2\varepsilon(\kappa)/e(1+q_E)q_{H_2}$, e is the base of the natural logarithm.

Theorem 2 reduces our IBE-Signal scheme to the BDH problem. To prove this reduction, firstly, we reduce the IBE-Signal scheme to a nonidentity-based encryption scheme BasicPub. Then, we reduce the BasicPub scheme to the BDH problem. The transitivity of the reduction is noticeable. □

Lemma 4. *Assume H_1 is a random oracle model from $\{0,1\}^*$ to G_1^* and the adversary A attacks the IBE-Signal scheme with advantage $\varepsilon(\kappa)$ in the IND-ID-CPA game. Assume A can run at most $q_E > 0$ partial private key queries. There must be an adversary B that can attack BasicPub successfully with an advantage $\varepsilon(\kappa)/e(1+q_E)$ in the IND-CPA game.*

Proof. The challenger first constructs the BasicPub scheme. B attacks the BasicPub scheme by taking A as a subroutine.

4.1. Establish the BasicPub scheme.

The challenger generates public parameters of the BasicPub scheme.

$$K_{\mathrm{pub}} = \begin{pmatrix} q, G_1, G_2, e, \kappa, k, P, P_i, Y_i, Q_{ID}, H_2, H_3, H_4, \\ H_5, H_6, \mathrm{KDF}, g, g_1, g_2, h_1, h_2, h_3, \tau, rk \end{pmatrix}. \tag{27}$$

And keeps the key a and the partial key S_i private, where $i = 0, 1, 2, \ldots, n$. The challenger sends K_{pub} to the adversary B.

4.2. B partial private key queries.

The challenger randomly chooses k as the honest CPC's serial number, generates the partial private keys $\left\{S_{ID_i} = [\prod_{h=0, h\neq k}^{i} S_h]Q_{ID}\right\}_{i=0,1,\ldots,n, i\neq k}$, and then sends them to B.

the following 4.3–4.8 steps, B simulates A's challenger to play IND-ID-CPA game with A.

4.3. Establish the IBE-Signal scheme.

B sends the IBE-Signal scheme's public parameters

$$K_{\text{pub}}' = \begin{pmatrix} q, G_1, G_2, e, \kappa, k, P, P_i, Y_i, H_1, H_2, H_3, H_4, \\ H_5, H_6, \text{KDF}, g, g_1, g_2, h_1, h_2, h_3, \tau, rk \end{pmatrix}, \tag{28}$$

to A. Since K_{pub} of the BasicPub scheme does not contain H_1, B needs to construct a 4-tuple $H_1^{list}(ID_j, Q_{ID_j}, b_j, \text{coin})$ to simulate A's challenger.

4.4. H_1 queries.

(i) *If A queries the H_1 values of ID_j, B will responds as follows*:
(ii) *If ID_j* already exists *in H_1^{list}, B responds* $Q_{ID_j} \in G_1^*$.
(iii) *Else, B randomly chooses* $\text{coin} \xleftarrow{R} \{0, 1\}$
(iv) *and sets* $\Pr[\text{coin} = 0] = \delta$, *then B chooses* $b_j \xleftarrow{R} \mathbb{Z}_q^*$,
(v) *If* coin = 0, *computes* $Q_{ID_j} = [b_j]Q_{ID} \in G_1^*$.
(vi) *Else, computes* $Q_{ID_j} = [b_j]P \in G_1^*$.
(vii) *B adds* $(ID_j, Q_{ID_j}, b_j, \text{coin})$ *to* H_1^{list} *and responds* Q_{ID_j} *to A.*

where coin = 0 represents that B thinks A will challenge ID^* in this query.

4.5. A partial private key queries – Phase I.

This step and step 4.7 can run at most q_E times in total. Assume A initiates a query on ID_j to B and each round of the query is independent.

(i) *B takes* $(ID_j, Q_{ID_j}, b_j, \text{coin})$ *from* H_1^{list}.
(ii) *If* coin = 0, *B reports an error and exits.*
(iii) *Else, B computes* $S_{ID_{ij}} = [b_j]Y_i = [b_j][\prod_{h=0}^{i} S_h]P = [\prod_{h=0}^{i} S_h][b_j]P = [\prod_{h=0}^{i} S_h]Q_{ID_j}$,
(iv) *and sends* $\left\{S_{ID_{ij}}\right\}_{i=0,1,\ldots,n.}$ *to A as the partial private key.*

4.6. challenge.

Assume A challenges (ID^*, M_0, M_1). B takes $(ID_j, Q_{ID_j}, b_j, coin)$ from H_1^{list} so that $ID_j = ID^*$.

(i) *If* coin = 1, *reports an error and exits*
(ii) *Else, computes* $S_{ID_{ij}} = [b_j]S_{ID_i} = [b_j] \quad [\prod_{h=0,h\neq k}^{i} S_h]Q_{ID}$
(iii) $= [\prod_{h=0,h\neq k}^{i} S_h]([b_j]Q_{ID}) = [\prod_{h=0,h\neq k}^{i} S_h]Q_{ID_j}$
(iv) *and sends* $\left\{S_{ID_{ij}}\right\}_{i=0,1,\ldots,n,i\neq k}$ *to A*
(v) *B sends M_0, M_1 to the challenger.*
(vi) *The challenger chooses* $\beta \xleftarrow{R} \{0, 1\}$ *and encrypts* M_β,
(vii) *then sends* $C^* = (U, V, rc)$ *to B.*
(viii) *B computes* $C^* = ([b_j^{-1}]U, V, rc)$ *and sends it to A as his response*

$$\begin{aligned} & e\left(S_{ID^*}, \left[b_j^{-1}\right]U\right) \\ &= e\left(\left[S_n S_{n-1} \ldots S_0 b_j\right]Q_{I\,D}, \left[b_j^{-1} r\right]P\right) \\ &= e\left(Q_{ID}, [S_n S_{n-1} \ldots S_0]P\right)^r \\ &= e\left(Q_{ID}, Y\right)^r. \end{aligned} \tag{29}$$

4.7. A partial private key queries–Phase II.

Phase II is the same as Phase I.

4.8. Guess.

A outputs β' as his guess and sends it to B. B sends β' to the challenger as his guess. □

Assertion 1. In the above reduction process, if B does not exist, the simulation of B is complete.

From Assertion 1, A's advantage in a simulated attack is equal to that in an actual attack, at least $\varepsilon(\kappa)$. If A attacks the IBE-Signal scheme successfully in step *4.8*, B can attack the BasicPub scheme successfully.

Since the probability of B without interruption is $(1-\delta)^{q_E}$ in steps *4.5*, *4.7*, and δ in step *4.6*. The advantage $\text{Adv}_{\text{BasicPub},B}^{\text{IND-CPA}}(\kappa)$ is at least $(1-\delta)^{q_E} \cdot \delta \cdot \varepsilon(\kappa)$. Consider $\text{Adv}_{\text{BasicPub},B}^{\text{IND-CPA}}(\kappa)$ is the function of δ. It can be calculated that when $\delta = 1/(q_E + 1)$, $\text{Adv}_{\text{BasicPub},B}^{\text{IND-CPA}}(\kappa)$ reaches the maximum.

$$\begin{aligned} \text{Adv}_{\text{BasicPub},B}^{\text{IND-CPA}}(\kappa)_{\max} &= \left(1 - \frac{1}{q_E + 1}\right)^{q_E} \frac{1}{q_E + 1} \varepsilon(\kappa) \\ &> \left(\lim_{q_E \longrightarrow \infty} \left(1 - \frac{1}{q_E + 1}\right)^{q_E}\right) \frac{\varepsilon(\kappa)}{q_E + 1} \\ &= \frac{\varepsilon(\kappa)}{e(q_E + 1)}. \end{aligned} \tag{30}$$

Therefore, the advantage of B is at least $\varepsilon(\kappa)/e(1 + q_E)$.

Lemma 5. *Assume H_2 is a random oracle model from G_2 to $\{0,1\}^\kappa$. The adversary A attacks the BasicPub scheme with an advantage $\varepsilon(\kappa)$ and A can run at most $q_{H_2} > 0 H_2$ queries. There must be an adversary B that can solve the BDH problem on G with an advantage $2\varepsilon(\kappa)/q_{H_2}$.*

Proof. B already knows $(P, [a]P, [b]P, [c]P)$ and intends to compute $e(P,P)^{\text{abc}} \in G_2$ through the attack of A on the BasicPub scheme.

5.1. B generates public parameters of the BasicPub scheme

$$K_{\text{pub}} = \begin{pmatrix} q, G_1, G_2, e, \kappa, k, P, P_i, Y_i, Q_{ID}, H_2, H_3, H_4, \\ H_5, H_6, \text{KDF}, g, g_1, g_2, h_1, h_2, h_3, \tau, rk \end{pmatrix}. \tag{31}$$

Some parameters of K_{pub} are generated as follows: Let Q_{ID} be $[b]P$. B randomly chooses k as the honest CPC's serial number and generates $S_i\,(i = 0, 1, \ldots, n, i \neq k)$, then computes $P_k = [a]P$, $P_i = [S_i]P\,(i = 0, 1, \ldots, n, i \neq k)$, $Y_i = [\prod_{h=0}^{i} S_h]P\,(i = 0, 1, 2, \ldots, k-1)$, $Y_i = [(\prod_{h=0,h\neq k}^{i} S_h)]([a]P) = [(\prod_{h=0,h\neq k}^{i} S_h) \cdot a]P\,(i = k, k+1, \ldots, n)$, and $Y = Y_n = [(\prod_{h=0,h\neq k}^{n} S_h) \cdot a]P$.

5.2. H_2 queries.

B constructs a 2-tuple $H_2^{\text{list}}(F_j, H_j)$. A can issue H_2 queries at any time and at most q_{H_2} times.

Table 3: Computational overhead of operations.

Notations	Definition	In T_M	Time (ms)
T_M	Modular multiplication	1.00 T_M	≈ 0.13
T_I	Modular inversion	11.60 T_M	≈ 1.51
T_A	Two elliptic curve points addition	0.12 T_M	≈ 0.02
T_S	Elliptic curve scalar point multiplication	29.00 T_M	≈ 3.77
T_E	Exponentiation	240.00 T_M	≈ 31.2
$T_{H\alpha}$	One-to-one hash mapping	29.00 T_M	≈ 3.77
$T_{H\beta}$	Two-to-one hash mapping	120.23 T_M	≈ 15.63
T_P	Bilinear pairing	87.00 T_M	≈ 11.31
$T_{\text{AES-128}}$	Encryption or decryption of the 1 KB size message using AES-128 at CTR mode	26.77 T_M	≈ 3.48

(i) *If* F_j already exists *in* H_2^{list}, B responds $H_2(F_j) = H_j$.
(ii) *Else, randomly chooses* $H_j \xleftarrow{R} \{0,1\}^{\kappa}$, B responds $H_2(F_j) = H_j$
(iii) *and add* (F_j, H_j) *into* H_2^{list}.

5.3. Partial private key queries.

B generates the partial private keys $\left\{S_{ID_i} = [\prod_{h=0,h\neq k}^{i} S_h]Q_{ID}\right\}_{i=0,1,\ldots,n,i\neq k}$, then sends them to A.

5.4. Challenge.

A outputs M_0, M_1 to challenge. B randomly chooses $\beta \xleftarrow{R} \{0,1\}$ and sends $C^* = (U, V, rc)$ to A. Let U be $[(\prod_{h=0,h\neq k}^{n} S_h^{-1}) \cdot c]P$. To decrypt C^*, A should compute:

$$\begin{aligned} H_2(S_{ID}, U) &= H_2\left(\left[\left(\prod_{h=0,h\neq k}^{n} S_h\right)\cdot a\right]Q_{ID}, \left[\left(\prod_{h=0,h\neq k}^{n} S_h^{-1}\right)\cdot c\right]P\right) \\ &= H_2\left(\left[\left(\prod_{h=0,h\neq k}^{n} S_h\right)\cdot a\cdot b\right]P, \left[\left(\prod_{h=0,h\neq k}^{n} S_h^{-1}\right)\cdot c\right]P\right) \\ &= H_2([ab]P, [c]P) \\ &= H_2(P,P)^{\text{abc}}. \end{aligned} \tag{32}$$

A issues the H_2 queries before decrypting C^* and the BDH problem's solution will be embedded in H_2^{list}.

5.5. Guess.

A outputs $\beta' \in \{0,1\}$. Meanwhile, B randomly chooses (F_j, H_j) from H_2^{list} and let F_j be the solution to the BDH problem. □

Assertion 2. In the above simulation process, the simulation of B is complete.

Assertion 3. Let Λ denotes the event: in the above simulation, A has issued the query $H_2(e(S_{ID}, U))$. Then, $\Pr[\Lambda] \geq 2\varepsilon$.

Table 4: Computational overhead of the subalgorithms in our scheme IBE-Signal for $n = 10$, the number of the CPCs and $nu = 1,000,000,000$, and the number of users.

Phases			Operations	Time(ms)
SESSION	Setup	KGC setup	T_S	3.77
		CPC setup	$n(n-1)/2T_P + nT_S$	546.65
		Key authority setup	T_S	3.77
	Encryption key generation	Encryption RootInput generation	$2T_{H\alpha} + T_S + T_P + T_E$	53.82
		Encryption revocation key generation	$3T_{H\alpha} + T_P + T_E+$, $12T_S + 6T_M + T_{H\beta}$.	115.47
	Sign	Sender's private key generation	$T_I + (2n+5)T_S+$, $(n+2)T_{H\alpha} + (n+2)T_P$	276.72
		Signature generation	$T_P + T_E + T_{H\beta} + 2T_S + T_A$	65.7
	Encrypt	Encrypt	$T_{\text{AES-128}}$	3.48
	Decryption key generation	Receiver's private key generation	$T_I + (2n+5)T_S+$, $(n+2)T_{H\alpha} + (n+2)T_P$	276.72
		Decryption RootInput generation	$T_P + T_{H\alpha}$	15.08
		Revocation private keys generation	$\lceil \log_2^{nu} + 1 \rceil (7T_S + 3T_M) + T_{H\alpha}$	833.95
		Revocation key updates generation	$\lceil \log_2^{nu} \rceil (7T_S + 3T_M) + T_{H\alpha}$	807.17
		Revocation key selection	0	0
		Decryption revocation key generation	$2T_M + 2T_I + 4T_P$, $+2T_E + T_{H\alpha} + T_{H\beta}$	130.32
	Decrypt	Decrypt	$T_{\text{AES-128}}$	3.48
	Verify	Verify	$T_{H\alpha} + 2T_P + T_M + T_E + T_{H\beta}$	73.35
	Revocation	Revocation	0	0
SESSION total	$(12 + 6\lceil \log_2^{nu} \rceil) T_M + 4T_I + T_A + (34 + 5n + 14\lceil \log_2^{nu} \rceil)T_S+$, $6T_E + (2n+14) T_{H\alpha} + 4T_{H\beta} + (n(n+3)/2 + 14) T_P + 2T_{\text{AES-128}}$		3206.45	
CONNECTION	Sender key generation		$2T_S + T_{H\alpha} + T_{H\beta}$ *or* $T_{H\alpha}$	26.94
	Encrypt		$T_{AES-128}$	3.48
	Receiver key generation		$T_S + T_{H\alpha} + T_{H\beta}$ *or* $T_{H\alpha}$	23.17
	Decrypt		$T_{\text{AES-128}}$	3.48
CONNECTION total	$3T_s + 2T_{H\alpha} + 2T_{H\beta} + 2T_{\text{AES-128}}$		57.07	
Total	$(12 + 6\lceil \log_2^{nu} \rceil)T_M + 4T_I + T_A + (37 + 5n + 14\lceil \log_2^{nu} \rceil)T_S+$, $6T_E + (2n+16)T_{H\alpha} + 6T_{H\beta} + (n(n+3)/2 + 14)T_P + 4T_{\text{AES-128}}$		3263.52	

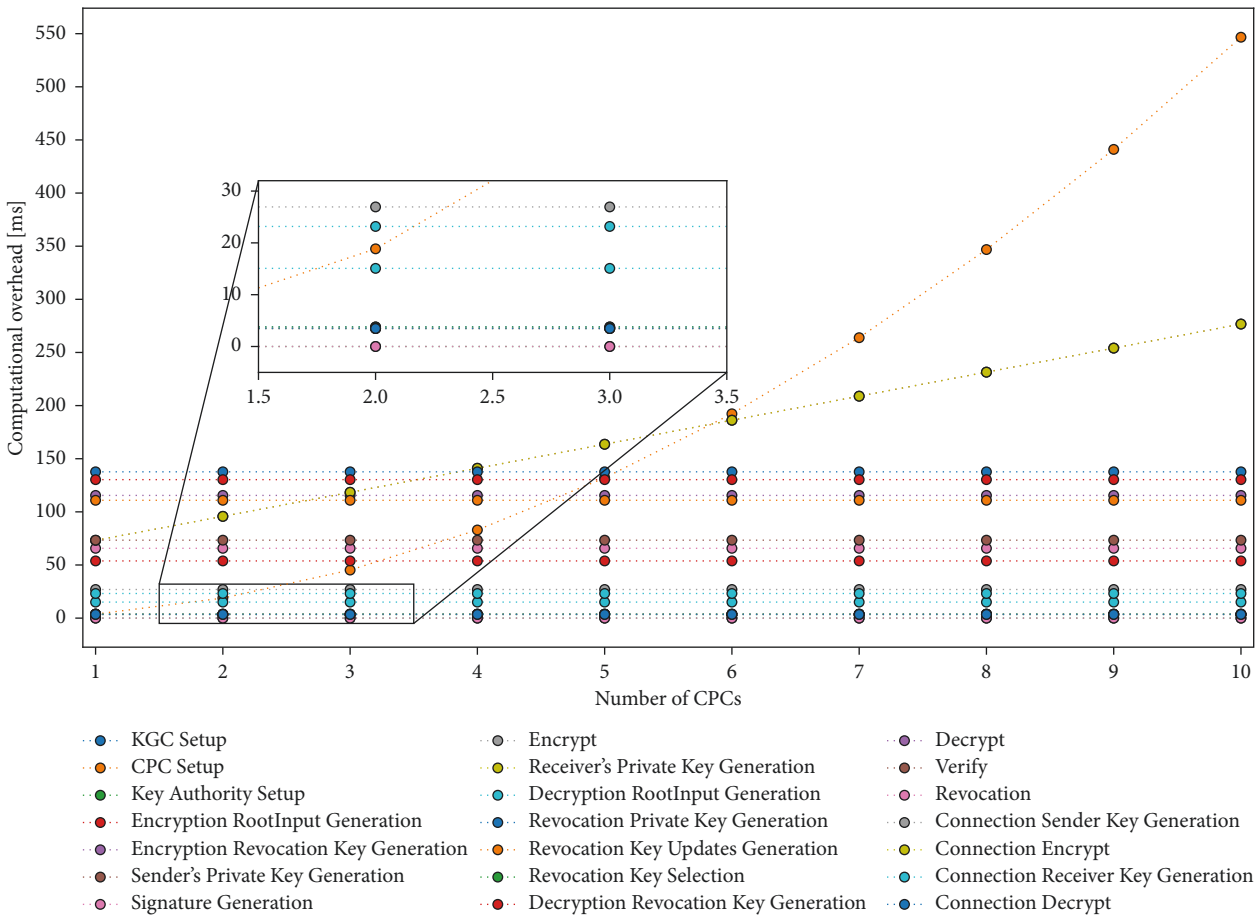

Figure 15: The computational overhead of each subalgorithm of the IBE-Signal scheme.

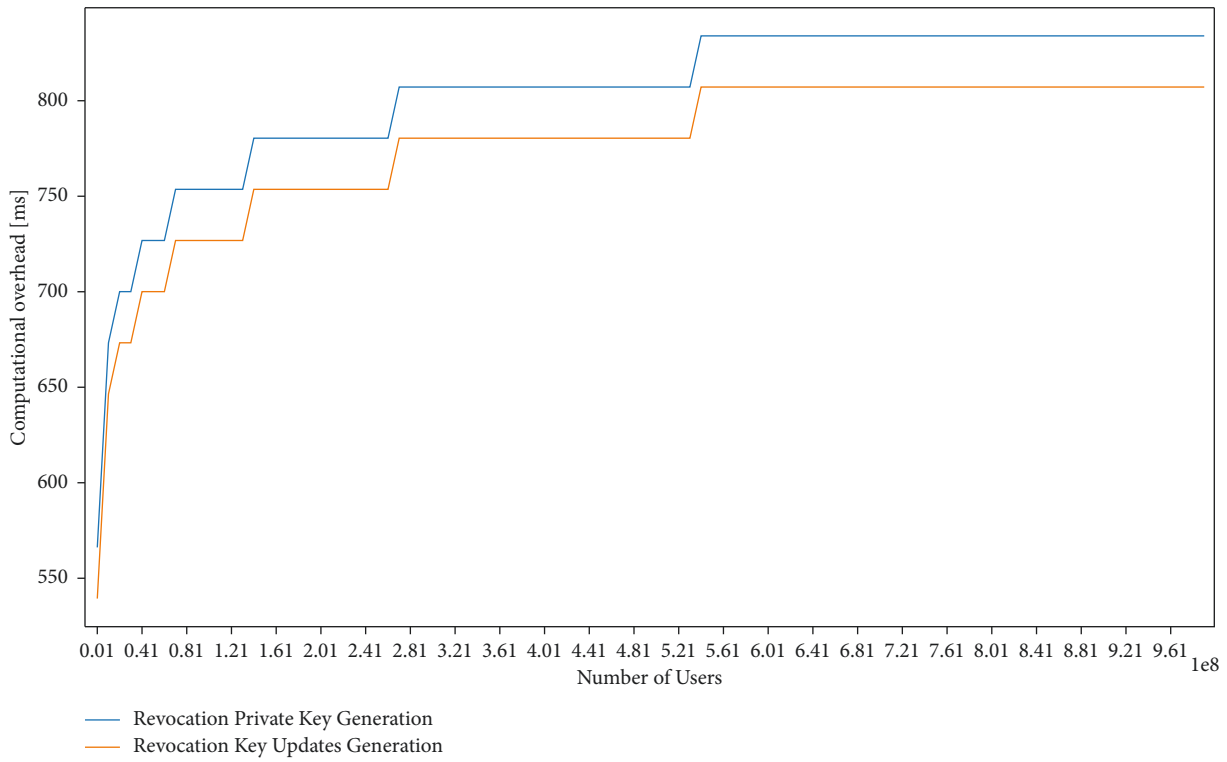

Figure 16: The computational overhead of *Revocation Private Keys Generation* and *Revocation Key Updates Generation*.

Table 5: The computational overhead of our scheme and other schemes.

Schemes	Setup and key extraction	Time
Kumar and Chand [17]	$(4n+6)(T_P+T_S)+T_{H\alpha}$	$\approx 60.32n+94.25$
Chen et al. [41]	$(3+n)T_S+(2+n)T_A$	$\approx 3.79n+11.35$
Chen et al. [42]	$(4n+7)(T_S+T_A)$	$\approx 15.16n+26.53$
Li et al. [43]	$(2+n)T_S+nT_A+T_P$	$\approx 3.79n+18.85$
Our scheme.	$T_I+(3n+7)T_S+(n+2)T_{H\alpha}+(n^2+n+4)/2T_P$	$\approx 5.655n^2+20.735n+58.06$

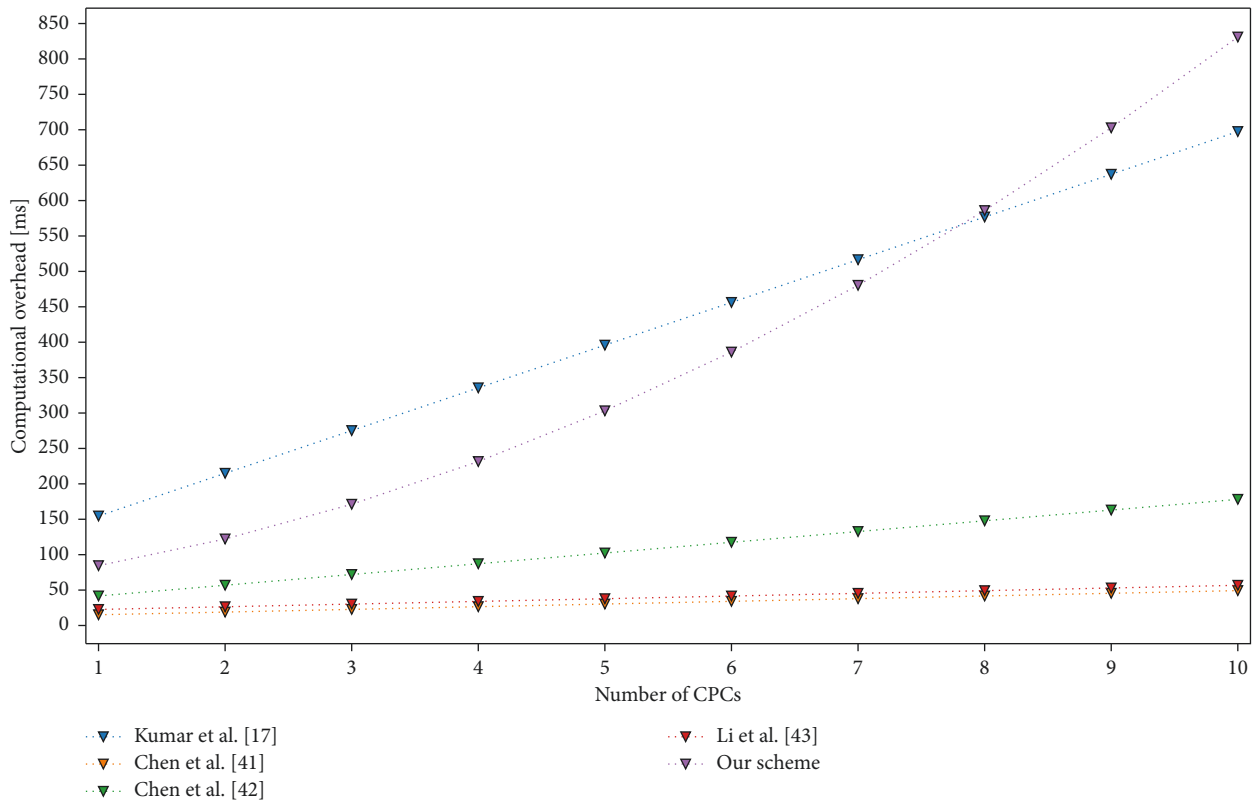

Figure 17: Comparison of the computational overhead with other schemes.

Table 6: Comparison of the security features between our scheme and other schemes.

Schemes	RKEP	RSKIP	RUSP	SMA	SPFS	SPCS	SKR
Kumar and Chand [17]	✓	✓	✓	✗	✗	✗	✗
Chen et al. [41]	✓	✗	✗	✗	✗	✗	✗
Chen et al. [42]	✓	✓	✗	✗	✗	✗	✗
Li et al. [43]	✓	✗	✗	✗	✗	✗	✗
Boldyreva et al. [18]	✗	✗	✗	✗	✗	✗	✓
Our scheme	✓	✓	✓	✓	✓	✓	✓

Proof. Obviously, $\Pr[\beta' = \beta|\Lambda] = 1/2$. We know $|\Pr[\beta' = \beta] - 1/2| \geq \varepsilon$. So,

$$\begin{aligned}
&\Pr[\beta' = \beta],\\
&= \Pr[\beta' = \beta|\Lambda]\Pr[\Lambda] + \Pr[\beta' = \beta|\Lambda]\Pr[\Lambda]\\
&\leq \Pr[\beta' = \beta|\Lambda]\Pr[\Lambda] + \Pr[\Lambda] = \frac{1}{2}\Pr[\Lambda] + \Pr[\Lambda]\\
&= \frac{1}{2}(1 - \Pr[\Lambda]) + \Pr[\Lambda] = \frac{1}{2} + \frac{1}{2}\Pr[\Lambda]\\
\Pr[\beta' = \beta] &\geq \Pr[\beta' = \beta|\Lambda]\Pr[\Lambda]\\
&= \frac{1}{2}\Pr[\Lambda]\\
&= \frac{1}{2}(1 - \Pr[\Lambda])\\
&= \frac{1}{2} - \frac{1}{2}\Pr[\Lambda]\\
\varepsilon \leq \left|\Pr[\beta' = \beta] - \frac{1}{2}\right| &\leq \frac{1}{2}\Pr[\Lambda]\\
\Pr[\Lambda] &\geq 2\varepsilon.
\end{aligned} \tag{33}$$

From Assertion 2 and Assertion 3, B solves the BDH problem with the probability at least $2\varepsilon/q_{H_2}$.

From Lemma 4, there is an adversary B' that attacks the BasicPub scheme with the advantage of at least $\varepsilon_1 = \varepsilon(\kappa)/e(1 + q_E)$. From Lemma 5, there is an adversary B that solves the BDH problem on G with the advantage of at least $2\varepsilon_1/q_{H_2} = 2\varepsilon(\kappa)/e(1 + q_E)q_{H_2}$. □

6. Performance Evaluation

We mainly evaluated the performance of our scheme from the computational overhead and the security features. To facilitate the calculation of overhead, we implemented it by calculating and combining cryptographic operations, which is also adopted in [17, 36–39]. Moreover, the implementation of the IBE-Signal scheme was based on OpenSSL v1.1.1f and the Type-A curve of the PBC library and was performed on a Dell Inspiron 7580 with Intel Core i7-8565U CPU (1.80 GHz) and 16 GB memory, running Ubuntu 64-bit Linux (v20.04.4 LTS). For the pairing-based scheme, to achieve the 1,024-bit RSA level security, we used the Tate pairing defined over the supersingular elliptic curve E/F_P: $y^2 = x^3 + x$ with embedding degree 2, where q is a 160-bit Solinas prime $q = 2^{159} + 2^{17} + 1$ and p a 512-bit prime satisfying $p + 1 = 12qr$. For the ECC-based schemes, to achieve the same security level, we employed the ECC group on Koblitz elliptic curve $y^2 = x^3 + ax^2 + b$ defined on $\mathbb{F}_{2^{163}}$ with $a = 1$ and b a 163-bit random prime.

6.1. The Computational Overhead of Our Scheme. Table 3 shows the time taken for the cryptographic operations involved in this paper. We define the overhead of modular multiplication as T_M, the overhead of modular inversion as T_I, the overhead of two elliptic curve points addition as T_A, the overhead of elliptic curve scalar point multiplication as T_S, the overhead of exponentiation as T_E, the overhead of one-to-one hash mapping as $T_{H\alpha}$, the overhead of two-to-one hash mapping as $T_{H\beta}$, the overhead of bilinear pairing as T_P, and the overhead of encryption or decryption of the 1 KB size message using AES-128 at CTR mode as $T_{\text{AES}-128}$. From [17], we have obtained $T_I = 11.60T_M$ and $T_{H\alpha} = 29.00T_M$. Then, we calculated the overheads of $T_M = 0.13ms$, $T_{H\beta} = 15.63ms$, and $T_{AES-128} = 3.48ms$ separately. In [40], $T_A = 0.12T_M$, $T_S = 29.00T_M$, and $T_E = 240.00T_M$ have been given. It has been given in [39] that the overhead of bilinear pairing $T_P = 3.00T_S = 87.00T_M$.

As shown in Table 4, we calculated the number of operations and the computational overhead of each subalgorithm of the IBE-Signal scheme according to Table 3. Among them, the CPCs are selected as ten, and the number of users is selected as 1 billion. The overhead of the *SESSION* part of our scheme is 3206.45 ms. However, the *SESSION* part only runs when the two opposing parties communicate for the first time, so a computational overhead of 3 seconds is acceptable. After the *SESSION* part is established, the *CONNECTION* part takes only 57.07 ms. Further, if the communication parties are not in the first message in each round of session, the computational overhead can be reduced to 14.5 ms.

The number of CPCs n is essential in determining the system's computational overhead. Theoretically, the more the number of the CPCs, the lower the risk of the key escrow problem, but the computational overhead also increases. In Figure 15, we show the relationship between the computational overhead of each subalgorithm and n from 1 to 10. Among them, the computational overhead of three subalgorithms *CPC Setup*, *Sender's Private Key Generation*, and *Receiver's Private Key Generation* increases as n increases. In addition to the number of the CPCs, the computational overhead of the two subalgorithms *Revocation Private Keys Generation* and *Revocation Key Updates Generation* is related to the number of users *nu*. We give the overhead trend as *nu* grows in Figure 16, where *nu* grows from 1 million to 1 billion. Considering the monthly active users in Table 1, we set the number of users to 1 billion. Furthermore, the algorithm's time complexity is $O(\log N)$ due to a binary tree structure. This results in a computational overhead of only about 800 ms, even with a billion users.

6.2. Comparison of the Computational Overhead with Other Schemes. In Table 5 and Figure 17, we compare the computational overhead between our scheme and other schemes. Moreover, the comparison mainly focuses on the two stages of Setup and Key Extraction. As shown in Figure 17, the *SESSION* part overhead of our scheme is lower than the Kumar et al. scheme [17] when the number of the CPCs is less than 8. Even if there are less than 8 CPCs, they can already provide sufficient private key distribution protection. Moreover, the *CONNECTION* part of our scheme is a

constant function for n. Our scheme runs the *CONNECTION* part in most cases, so our scheme is still competitive in computational overhead.

6.3. Comparison of the Security Features with Other Schemes. Table 6 compares our scheme with other schemes in terms of security features. We define Resilient to the Key Escrow Problem as RKEP, Resilient to the Secure Key Issuing Problem as RSKIP, Resilient to the User Slandering Problem as RUSP, Support Mutual Authentication as SMA, Support Perfect Forward Secrecy as SPFS, Support Post-compromise Security as SPCS, and Support Key Revocation as SKR. The KGC cannot directly generate private keys for users in our scheme, so the scheme has the RKEP feature. The blind signature enables the KGC or the CPCs to issue the private key without seeing the actual information sent by the user, which provides the RSKIP feature for the scheme. The CPCs have a master key slice, which spreads the risk of private key distribution and provides the RUSP feature. Furthermore, we provide the SMA feature by introducing IBS. Since our *CONNECTION* part is based on the Double Ratchet algorithm, we provide both the SPFS feature and the SPCS feature. Finally, we introduce the Revocable IBE to provide the SKR feature.

7. Conclusion

We have proposed the IBE-Signal scheme to reshape the Signal into a MITM-attack-resistant protocol. Our scheme provides more security features (such as Perfect Forward Secrecy, Post-compromise Security, and Key Revocation) than other schemes. Furthermore, experiments show that our scheme has less computational overhead when the number of the CPCs is less than 8. Moreover, we proved that our scheme is IND-ID-CPA secure under the random oracle model, even if only one CPC is credible. We increased the attackers' capabilities in the proofs: some partial private keys of ID^* can still be queried in the challenge phase. In the future, we will transform the IND-ID-CPA secure scheme into IND-ID-CCA secure scheme using the Fujisaki-Okamoto method [44].

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare no known conflicts of interest or personal relationships that could have appeared to influence the work reported in this study.

Acknowledgments

This work was supported by Research Funds for NSD Construction, University of International Relations (2020GA04) and National Natural Science Foundation of China (62102113).

References

[1] K. Cohn-Gordon, C. Cremers, B. Dowling, L. Garratt, and D. Stebila, "A formal security analysis of the signal messaging protocol," *Journal of Cryptology*, vol. 33, no. 4, pp. 1914–1983, 2020.

[2] K. Ermoshina, F. Musiani, and H. Halpin, "End-to-end encrypted messaging protocols: an overview," in *Internet Science*, pp. 244–254, Springer, New York City, USA, 2016.

[3] N. Unger, "SoK: secure messaging," in *Proceedings of the 2015 IEEE Symposium on Security and Privacy*, pp. 232–249, IEEE, San jose, CA, USA, May, 2015.

[4] M. Moxie and P. Trevor, "The double ratchet algorithm," 2016, https://signal.org/docs/specifications/doubleratchet/.

[5] D. Curry, "Signal revenue & usage statistics," 2022, https://www.businessofapps.com/data/signal-statistics/.

[6] "WhatsApp - statistics & facts," 2022, https://www.statista.com/topics/2018/whatsapp/.

[7] "Facebook messenger - statistics & facts," 2022, https://www.statista.com/topics/4625/facebook-messenger/.

[8] SkypeStatistics, "Skype Statistics, User Counts, Facts & News," 2022, https://expandedramblings.com/index.php/skype-statistics/.

[9] Supportsignalorg, "What is a safety number and why do I see that it changed?," 2021, https://support.signal.org/hc/en-us/articles/360007060632-What-is-a-safety-number-and-why-do-I-see-that-it-changed.

[10] S. Schröder, M. Huber, D. Wind, and C. Rottermanner, "When SIGNAL hits the fan: on the usability and security of state-of-the-art secure mobile messaging," in *Proceedings of the European Workshop on Usable Security*, pp. 1–7, IEEE, Darmstadt, Germany, July, 2016.

[11] D. Boneh, M. Franklin, B. Lynn, M. Pauker, R. Kacker, and G. Tsudik, "IBE secure E-mail," 2002, https://crypto.stanford.edu/ibe/#technical.

[12] O. Blazy, A. Bossuat, X. Bultel, P.-A. Fouque, C. Onete, and E. Pagnin, "SAID: Reshaping Signal into an Identity-Based Asynchronous Messaging Protocol with Authenticated Ratcheting," in *Proceedings of the 2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 294–309, IEEE, Stockholm, Sweden, June 2019.

[13] K. G. Paterson, "Cryptography from pairings: a snapshot of current research," *Information Security Technical Report*, vol. 7, no. 3, pp. 41–54, 2002.

[14] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Florida, USA, 2018.

[15] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques*, pp. 47–53, Springer, Santa Barbara, CA, USA, Auguest 1984.

[16] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Proceedings of the Annual International Cryptology Conference*, pp. 213–229, Springer, Santa Barbara, California, USA, Auguest 2001.

[17] M. Kumar and S. Chand, "ESKI-IBE: efficient and secure key issuing identity-based encryption with cloud privacy centers," *Multimedia Tools and Applications*, vol. 78, no. 14, pp. 19753–19786, 2019.

[18] A. Boldyreva, V. Goyal, and V. Kumar, "Identity-based encryption with efficient revocation," in *Proceedings of the 15th ACM Conference on Computer and Communications Security*, pp. 417–426, Alexandria,Virginia, USA, October 2008.

[19] K. Cohn-Gordon, C. Cremers, and L. Garratt, "On post-compromise security," in *Proceedings of the 2016 IEEE 29th Computer Security Foundations Symposium*, pp. 164–178, IEEE, Lisbon, Portugal, July 2016.

[20] D. Van Dam, "Analysing the signal protocol," Available: https://www.ru.nl/publish/pages/769526/z00b_2019_thesis_dion_van_dam_2019_eerder.pdf.

[21] T. Frosch, C. Mainka, C. Bader, F. Bergsma, J. Schwenk, and T. Holz, "How Secure Is TextSecure?" in *Proceedings of the 2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 457–472, IEEE, Saarbruecken, Germany, March 2016.

[22] J. Alwen, S. Coretti, and Y. Dodis, "The double ratchet: security notions, proofs, and modularization for the signal protocol," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 129–158, Springer, Darmstadt, Germany, May 2019.

[23] N. Kobeissi, K. Bhargavan, and B. Blanchet, "Automated Verification for Secure Messaging Protocols and Their Implementations: A Symbolic and Computational Approach," in *Proceedings of the 2017 IEEE European symposium on security and privacy (EuroS&P)*, pp. 435–450, IEEE, Paris, France, April 2017.

[24] R. L. Rivest and A. Shamir, "How to expose an eavesdropper," *Communications of the ACM*, vol. 27, no. 4, pp. 393-394, 1984.

[25] A. S. Khader and D. Lai, "Preventing man-in-the-middle attack in Diffie-Hellman key exchange protocol," in *Proceedings of the 2015 22nd International Conference on Telecommunications (ICT)*, pp. 204–208, IEEE, Sydney, NSW, Australia, April 2015.

[26] V. Goyal, "Reducing trust in the PKG in identity based cryptosystems," in *Proceedings of the Annual International Cryptology Conference*, pp. 430–447, Springer, Santa Barbara, CA, USA, Auguest 2007.

[27] V. Goyal, S. Lu, A. Sahai, and B. Waters, "Black-box accountable authority identity-based encryption," in *Proceedings of the 15th ACM Conference on Computer and Communications Security*, pp. 427–436, Alexandria Virginia, USA, October 2008.

[28] S. Garg, M. Hajiabadi, M. Mahmoody, and A. Rahimi, "Registration-based encryption: removing private-key generator from IBE," in *Proceedings of the Theory of Cryptography Conference*, pp. 689–718, Springer, Panaji, India, November 2018.

[29] S. Garg, M. Hajiabadi, M. Mahmoody, A. Rahimi, and S. Sekar, "Registration-based encryption from standard assumptions," in *Proceedings of the IACR International Workshop on Public Key Cryptography*, pp. 63–93, Springer, Beijing,China, April 2019.

[30] A. Joux, "The Weil and Tate pairings as building blocks for public key cryptosystems," in *Proceedings of the International Algorithmic Number Theory Symposium*, pp. 20–32, Springer, Sydney, Australia, July 2002.

[31] M. Moxie and P. Trevor, "The X3DH key agreement protocol," 2016, https://signal.org/docs/specifications/x3dh/.

[32] F. Hess, "Efficient identity based signature schemes based on pairings," in *Proceedings of the International Workshop on Selected Areas in Cryptography*, pp. 310–324, Springer, Newfoundland, Canada, Auguest 2002.

[33] A. Fiat and A. Shamir, "How to prove yourself: practical solutions to identification and signature problems," in *Proceedings of the Conference on the Theory and Application of Cryptographic Techniques*, pp. 186–194, Springer, Santa Barbara,California, USA, January 1987.

[34] J. C. Choon and J. Hee Cheon, "An identity-based signature from gap Diffie-Hellman groups," in *Proceedings of the International Workshop on Public Key Cryptography*, pp. 18–30, Springer, Miami, FL, USA, January 2003.

[35] T. Dierks and C. Allen, *Rfc2246: The TLS Protocol Version 1.0*, RFC, Ed., , 1999.

[36] H. Debiao, C. Jianhua, and H. Jin, "An ID-based proxy signature schemes without bilinear pairings," *annals of telecommunications - annales des télécommunications*, vol. 66, no. 11-12, pp. 657–662, 2011.

[37] P. S. Barreto, B. Lynn, and M. Scott, "On the selection of pairing-friendly groups," in *Proceedings of the International Workshop on Selected Areas in Cryptography*, pp. 17–25, Springer, Ottawa, Canada, Auguest 2003.

[38] S. H. Islam and G. P. Biswas, "A pairing-free identity-based authenticated group key agreement protocol for imbalanced mobile networks," *annals of telecommunications - annales des télécommunications*, vol. 67, no. 11-12, pp. 547–558, 2012.

[39] X. Cao, W. Kou, and X. Du, "A pairing-free identity-based authenticated key agreement protocol with minimal message exchanges," *Information Sciences*, vol. 180, no. 15, pp. 2895–2903, 2010.

[40] N. Koblitz, A. Menezes, and S. Vanstone, "The state of elliptic curve cryptography," *Designs, Codes and Cryptography*, vol. 19, no. 2/3, pp. 173–193, 2000.

[41] P. Chen, X. Wang, B. Zhao, J. Su, and I. You, "Removing key escrow from the LW-HIBE scheme," in *Proceedings of the International Conference on Algorithms and Architectures for Parallel Processing*, pp. 593–605, Springer, Zhangjiajie, China, November 2015.

[42] P. Chen, X. Wang, and J. Su, "T-HIBE: a trustworthy HIBE scheme for the OSN privacy protection," in *Proceedings of the 2015 International Symposium on Security and Privacy in Social Networks and Big Data (SocialSec)*, pp. 72–79, IEEE, Hangzhou, China, November 2015.

[43] Y. Li, F. Qi, and Z. Tang, "An efficient hierarchical identity-based encryption scheme for the key escrow," in *Proceedings of the International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*, pp. 108–120, Springer, Guangzhou, China, December 2017.

[44] E. Fujisaki and T. Okamoto, "Secure integration of asymmetric and symmetric encryption schemes," in *Proceedings of the Annual International Cryptology Conference*, pp. 537–554, Springer, Santa Barbara, California, USA, Auguest 1999.