Anonymous Key Issuing Protocol with Certified Identities in Identity-Based Encryption

Yanqing Yang(✉) and Jian Wang

Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
aurora2020_y@163.com

Abstract. Identity-based encryption can simplify the certificate management problem of traditional public key cryptosystem. But it also has an inherent drawback, called the key escrow problem. A malicious key generation center (KGC) can easily generate the user's private key to decrypt ciphertexts or perform illegal activities. In this paper, we propose an anonymous key issuing-based IBE protocol to solve the key escrow problem. The protocol separates the tasks of authentication and key generation, which are respectively implemented by two authorities. Identity-certifying authority (ICA) authenticates the user and collaborates with him to generate a pseudo-identity to hide the real one, and issues an evidence to prove its validity. KGC generates private keys for users based on their pseudo-identities. We prove the security based on recipient anonymity and ciphertext indistinguishability, with the assumption that KGC does not collude with ICA. Compared with Emura's work, we weaken the communication restriction between KGC and ICA by allowing KGC to access the identity list stored by ICA arbitrarily, and only restrict its query method during the key generation phase, making it accessible through the index. This lowers the security assumption and makes it more realistic. Meanwhile, we eliminate the need for secure channels using blinding technique, and only require authentication channels to ensure that messages are not being tampered with. Our protocol has higher security and utility.

Keywords: Identity-based encryption · Key escrow · Anonymous key issuing

1 Introduction

In 1984, Shamir [1] proposed the identity-based cryptosystem (IBC). Boneh and Franklin [2] applied Weil pairs to elliptic curves and proposed the first identity-based encryption scheme (IBE) in 2001, whose public key is a string related to user's identity and the private key is distributed by KGC after authentication. Therefore, there is no certificate management problem of traditional public key cryptography, and users only need to obtain the identity of recipient to encrypt

H. Duan et al. (Eds.): SecureComm 2023, LNICST 567, pp. 309–328, 2025.
https://doi.org/10.1007/978-3-031-64948-6_16

messages. The algorithm for generating private keys is public, and KGC holds the system master key for generating all users' private keys, which makes users' private keys and system security completely dependent on KGC. This leads to the key escrow problem of IBE.

In traditional IBE scheme, KGC not only checks the legitimacy of user identity, but also establishes a secure channel to communicate with legitimate users. KGC's right is too centralized, causing it to be overburdened, and once KGC is breached, it will reveal all user's private keys. A malicious KGC can directly use the identity of legitimate user to generate private keys, decrypt or sign messages. Despite the good usability of IBE, the key escrow problem caused by KGC still limits its development. Meanwhile, in the real communication environment, the need for secure channels to ensure the distribution of users' private keys also greatly limits the popular application of IBE. Although there are already protocols that provide secure communication to guarantee confidentiality and data integrity, such as the TLS protocol. But they need to negotiate a session key to encrypt and decrypt messages to ensure secure communication, incurring some computational overhead. Therefore, it is important to solve the key escrow problem in IBE scheme while eliminating the high need for secure channels.

To tackle this problem, many literatures give the corresponding solutions such as multiple trusted authorities [3–9], certificateless public key cryptography [10–12] and anonymous key issuing [13–16]. These schemes solve the key escrow problem to some extent, but still have several drawbacks. Section 2 reviews these schemes and discusses their properties in more detail.

The anonymous key issuing architecture combined with the blind key extraction scheme achieves identity anonymity and ciphertext indistinguishability, and solves the key escrow problem. It separates the two processes of authentication and key issuing, which are performed by ICA and KGC respectively. However, the existing schemes [13–16] require that KGC cannot communicate with ICA, which is not realistic. In practice, there is no way to restrict communication between arbitrary agencies in the network, unless the agency itself refuses. If this requirement is relaxed, i.e., KGC can communicate with ICA, these anonymous key issuing schemes cannot guarantee security. And in Emura's scheme [16], once ICA gets the key distributed to the user by KGC, it could use trapdoor information to recover the user's private key and threaten the system security.

Following Emura's work [16], we propose an anonymous key issuing-based IBE protocol with certified identity to solve the key escrow problem, which combines BF-IBE [2] with Bolydreva's blind signature scheme [17] and uses a one-time evidence to hide the connection between user's identity and private key. The contributions are summarized as follows:

- We improve the framework proposed by Emura, and allow ICA to communicate with KGC. We only restrict the way that KGC accesses the identity list stored by ICA during the key generation phase, allowing him to query by index, while not getting the user's real identity. At other times, KGC can query the identities of registered users in the system.
- We reset the composition of evidence, and allow user and ICA jointly generate the evidence to avoid passive attack of ICA. KGC verifies the pseudo-identity

based on the evidence and distributes a private key for the anonymous applicant through an authentication channel instead of a secure channel. In the end, only the user can recover the real private key using the trapdoor information sent by ICA through an authentication channel and his own secret.
– We consider the existence of malicious user, KGC and the honest but curious ICA, and prove the protocol's security under random oracle model. The analysis shows that our protocol has better security than existing schemes [13–16].

The rest of the paper is organized as follows. In Sect. 2, we review the related work on solving key escrow problem. Section 3 describes the relevant concepts, computational problems and algorithms in IBE, which are the basis of this paper. The system model and specific construction are given in Sect. 4. Section 5 and 6 demonstrate the security of our scheme, compare and analyze the advantages. Finally, we conclude the whole paper.

2 Related Work

The current solution to the key escrow problem is to decrease the trust dependence of single KGC, e.g., multiple trusted authorities [3–9], certificateless public key cryptography [10–12], anonymous key issuing protocol [13–16], etc.

2.1 Multiple Trusted Authorities

Such schemes distribute the right to generate private keys by KGC to multiple trusted authorities, such as the secret sharing-based scheme of Boneh and Franklin [2], and the use of multiple KPAs approach proposed by Lee. B et al. [6].

The secret sharing-based scheme, also called threshold key issuing scheme. In 2001, Boneh and Franklin [2] proposed to protect the master key using distributed KGCs, but they did not consider the secrecy of users' private keys in the system. This scheme uses Shamir's (t,n)-threshold secret sharing scheme, where the master key is secretly partitioned into n shares and distributed to n KGCs. Each KGC owns one share of master key and generates a piece of private key. The user's private key is obtained by cooperation of greater than or equal to t KGCs. Chen et al. [3] and Hess et al. [5] have proposed various improvement schemes, but all of them have certain problems:

– A large number of hash operations and other algebraic operations always make the computation less efficient.
– A secure channel needs to be established between KGCs and users to verify and distribute information, raising the requirements for network transmission efficiency, network bandwidth, and the response speed of KGCs.
– There is an inherent problem with Shamir's threshold secret sharing scheme, which is constructed on the basis that both the participants and sub-secret

sharers of scheme are trusted. If the secret distributor distributes false sub-keys, the master key cannot be recovered even if all sub-secret sharers cooperate together. Similarly, if the sub-secret sharer provides false sub-secrets, the master key cannot be recovered in the end.

In 2004, Lee et al. [6] proposed a secure key issuing protocol, where KGC distributes the private key and multiple key privacy authorities (KPAs) sequentially protect key privacy, so as to reduce the trust of single KGC. Although KPA has its own independently chosen master key, it is essentially the same as KGC in that it generates a partial private key for user. Multiple KPAs need to verify the correctness of partial public keys sequentially, which increases the time overhead of key generation. This scheme is similar to the threshold key issuing scheme, and they both make the structure of IBE scheme more complex.

With the development of network technology, some schemes based on Net-ID clustering approach [7] or cloud computing [8,9,12] have emerged, which put a large amount of exponential and bilinear pairings in the cloud. However, these schemes do not optimize the complex structure of multiple trusted authorities and still have huge communication overhead.

2.2 Certificateless Public Key Cryptography

Al-Riyami and Paterson [11] proposed certificateless public key cryptosystem, where the user needs to select some secret information to eliminate the key escrow problem. Suppose KGC has master key s_0 and public key $P_0 = s_0 P$ and the user with identity ID requests to issue a private key. Then the scheme runs as follows:

1. User selects a random secret value $x_{ID} \in Z_q^*$, calculates public key $P_{ID} = (X_{ID}, Y_{ID}) = (x_{ID}P, x_0 P)$. He sends public key P_{ID} and identity ID to KGC and requests him to issue a private key.
2. KGC verifies the user's identity, extracts a partial private key $D'_{ID} = s_0 Q_{ID}$, which $Q_{ID} = H_1(ID, P_{ID})$, and sends D'_{ID} to the user.
3. User sets his private key as $D_{ID} = x_{ID} D'_{ID}$.

Although the certificateless public key cryptosystem protects the secrecy of user's private key, both ID and P_{ID} are required to encrypt messages. Additional individual public keys detract one of the main merits of IBE since the size of public information grows linearly with the number of users in the system.

2.3 Anonymous Key Issuing Protocol

Sui [13] first proposed the idea of anonymous key issuing (AKI), where the responsibility of authentication and key issuing are separated to local registration authority (LRA) and KGC respectively. To help KGC authenticate users, LRA distributes a one-time password for each user. However, Sui requires that LRA sends the user's identity-password pair to KGC and his scheme can only provide identity anonymity to external adversaries.

In 2006, Cao [18] pointed out that Sui's scheme failed to address the adversary's attack on the run records and his authentication technique is insecure. According to Diffie's earlier suggestion [19], a secure authentication protocol must satisfy certain properties. The second property is called matching records of runs, which says that for a successful run of a mutual authentication protocol, if both of two principals have recorded every exchange, then their records of the run will match [19]. In Sui's scheme, each item of the user's run records does not match the corresponding item in KGC's run records. Thus this protocol does not reach the second property of Diffie's proposal for successful running of a mutual authentication. Cao does not use the AKI architecture, instead using multiple trusted authorities to distribute private keys for users. Here the details will not be discussed.

Chow [14] formally proposed a new secure notion of KGC anonymous ciphertext indistinguishability (ACI-KGC), an important step in addressing the key escrow problem in IBE. In his scheme, the AKI architecture is used to protect the confidentiality of user's identity and contains three participants, namely, user, identity-certifying authority (ICA), and KGC. ICA checks the user's identity and distributes a certificate for him. The certificate is a commitment to the identity, with the property of hiding and binding, which enables KGC to verify the commitment information through the certificate, while not obtaining the real identity of users. KGC holds the master key and performs a secure two-party computational protocol with the anonymous user to generate a private key. But secure two-party computational protocols are always complex, typically implement the confidentiality of their inputs via obfuscated circuits, and have low processing throughput and computational efficiency, even with hardware acceleration.

Wei [15] improved Chow's scheme by removing the interactive key generation protocol and reduced the interactive communication overhead. The scheme successfully achieves ACI-KGC security and solves the key escrow problem. However, observing the composition of user's private key, we can find it contains the keys of both ICA and KGC. In other words, ICA distributes partial private keys for users, which is different from the description of roles in Wei's scheme, where ICA only verifies the user's identity and distributes a proof.

Emura [16] proposed a blind IBE scheme with certified identities and achieved user anonymity and ciphertext indistinguishability. Users use the certificate provided by ICA to request a private key from KGC, without secure two-party computation. In this scheme, the evidence is a signature of a pseudo-identity, which is provided by ICA, and requires the signature algorithm to provide existential unforgeability. This enables that KGC can authenticate but cannot recover the real identity from the evidence, truncates the direct connection between user's identity and his private key.

However, the above works [13–16] assume that KGC cannot communicate with ICA and cannot access the identity list of users stored by ICA to guarantee user anonymity. In fact, this assumption is not valid to prevent the two organizations from communicating. Once the two organizations communicate, ICA can eavesdrop and even use the user's evidence to request, so as to obtain the

key distributed to user by KGC and learn the information about user's private keys, and the above schemes will not guarantee security.

By analyzing the above three technical routes to solve the key escrow problem, we can draw conclusions. Firstly, the scheme with multiple trusted authorities uses a secret sharing scheme that requires taking collusion into account, and no amount of improvement can eliminate the complex structure and huge communication overhead in such schemes. Secondly, certificateless public key encryption schemes provide only implicit authentication and require maintaining a public key list, thus losing the advantage of IBE. Therefore, this paper uses anonymous key issuing architecture to solve the key escrow problem, which has more research prospect.

3 Preliminaries

We introduce some preliminary knowledge related to the proposed protocol. Our scheme builds on BF-IBE and borrows the framework from Emura's work.

3.1 Notation Definition

The major notations used in this paper are listed in Table 1.

Table 1. Notation table

Notation	Description
ID	user's identity
λ	security parameter
$\mathcal{ID}$	identity space
$\mathcal{M}$	plaintext space
$\mathcal{C}$	ciphertext space
msk	system master key
P_{pub}	system public key
$\mathbb{G}, \mathbb{G}_T$	cycle group of order p
g	the generator of $\mathbb{G}$
Π_{Sig}	digital signature algorithm

3.2 Bilinear Pairings

Suppose two cyclic groups $\mathbb{G}$ and $\mathbb{G}_T$ of order p, where p is a large prime of λ-bit and g is a generator of $\mathbb{G}$. The discrete logarithm problem (DLP) in these groups is believed to be hard. A bilinear map is defined as $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ with the following properties:

1. Bilinear: $\forall g_1, g_2 \in G$ and $a, b \in Z_p^*$,

$$e(g_1^a, g_2^b) = e(g_1, g_2)^{ab} \tag{1}$$

2. Non-degeneracy: $e(g, g) \neq 1$ holds.
3. Computable: $\forall g_1, g_2 \in \mathbb{G}$, there exists an efficient algorithm to calculate $e(g_1, g_2)$.

Typically, the map e is derived from Weil and Tate pairings over finite fields.

3.3 Hard Problem Assumption

Here, we only present the Decision Bilinear Diffie-Hellman (DBDH) problem used in this paper: For any $a, b, c \in Z_p^*$, given (g, g^a, g^b, g^c, Z), decide whether $Z = e(g, g)^{abc}$.

3.4 Boneh-Franklin's IBE Scheme

BF-IBE scheme [2] consists of four algorithms, namely Setup, Key generation, Encryption and Decryption, which are described as follows:

1. $Setup(1^\lambda) \rightarrow (msk, params)$: KGC selects $msk = s \in Z_p^*$, and computes $P_{pub} = sP$. The system parameter $params = \{\mathbb{G}, \mathbb{G}_T, e, P, H, H_2, P_{pub}\}$.
2. $KeyDer(msk, ID) \rightarrow D_{ID}$: KGC authenticates the user and generates a private key $D_{ID} = sQ_{ID}$, where $Q_{ID} = H(ID)$.
3. $Encrypt(ID, M, params) \rightarrow C$: Sender randomly selects $r \in Z_p^*$, calculates $Q_{ID} = H(ID)$, $U = rP$ and $V = M \oplus H_2(e(Q_{ID}, P_{pub})^r)$ using recipient identity ID. The ciphertext is $C = (U, V)$.
4. $Decrypt(C, D_{ID}) \rightarrow M$: Receiver decrypts $V \oplus H_2(e(D_{ID}, U))$ using the private key D_{ID} and outputs the plaintext M.

3.5 Emura's General Framework

Emura [16] proposed a blind IBE with certified identity to solve the key escrow problem, in which users first authenticate themselves to ICA to obtain a certificate, which they later use to run an interactive protocol with KGC and construct a secret key as in standard IBE. The certificate is an evidence distributed by ICA to the user, which is different from the public-key certificate. In Fig. 1, we show the system architecture that was given in [16].

Emura's blind IBE scheme with certified identities consists of the following PPT algorithms:

1. $Setup(1^\lambda) \rightarrow (params)$: The system initialization uses the security parameters 1^λ as input and generates the public parameters $params$.
2. $KGC.KeyGen(params) \rightarrow (mpk, msk)$: KGC initializes the master public key mpk and the master private key msk.

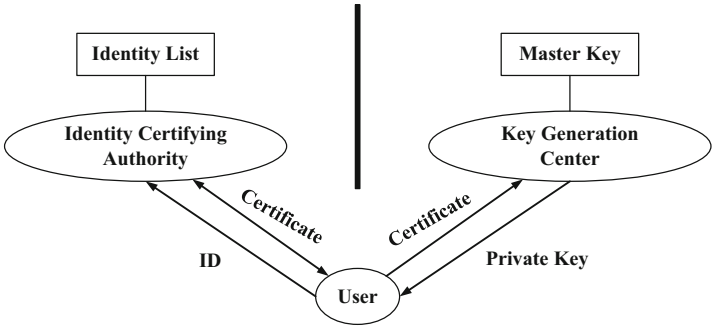

Fig. 1. Emura's system architecture

3. $ICA.KeyGen(params) \rightarrow (ik, vk)$: ICA runs the key generation algorithm and outputs a certificate issuing-verifying key pair (ik, vk).
4. $ICA.Cert(vk, ik, ID) \rightarrow (cert, td)$: ICA verifies the user's identity, distributes a certificate $cert$ and a trapdoor message td.
5. $\langle ObtainKey(mpk, ID, cert, td), IssueKey(mpk, msk, vk)\rangle \rightarrow (sk_{ID}, \epsilon)$: The user and KGC run $ObtainKey$ and $IssueKey$ algorithms and get the final private key sk_{ID} and an empty string ϵ respectively. During this process, the user sends the certificate $cert$ to KGC as first-round message M_{user}, KGC runs $IssueKey$ algorithm and returns a second-round message M_{KGC}, and finally the user locally outputs either sk_{ID}.
6. $IBE.Enc(mpk, ID, M) \rightarrow ct$: The user runs the *encryption* algorithm and gets the ciphertext ct as in standard IBE.
7. $IBE.Dec(mpk, sk_{ID}, ct) \rightarrow M$ or $\perp$: The user uses the private key sk_{ID} to decrypt and recover the message as in standard IBE.

4 The Anonymous Key Issuing-Based IBE Protocol

Based on Emura's work [16], we present an improved anonymous key issuing-based IBE protocol with certified identities to solve the key escrow problem. The scheme follows the AKI architecture, also combines BF-IBE scheme [2] and Boldyreva's blind signature scheme [17], and considers blind signatures as the private key of BF-IBE scheme, which inherently has anonymity. The scheme improved the evidence-issuing algorithm in Emura's scheme, so that it can successfully resist passive attacks launched by an honest but curious ICA. That is, even if ICA uses the user's evidence to request a key from KGC, it cannot learn any information about the user's private key.

4.1 System Architecture

The anonymous key issuing-based IBE protocol employs a non-colluding identity-certifying authority (ICA) and KGC. ICA is not just an authentication

authority for users' identities, but also hides identities through an encapsulation method, and generates evidence together with users to achieve user anonymity. The evidence is one-time, which is only used to authenticate the user's pseudo-identity and does not allow KGC to decrypt it. User uses the evidence to request a private key from KGC. Since KGC cannot know the identity contained in the evidence, it can only blindly generate a key for that user, i.e. blind key extraction.

ICA owns the identity list of system users. To ensure secure key issuing, we require that ICA and KGC do not collude. That is, ICA sets the access rights of the identity list so that during the key generation phase, KGC cannot access the real identity of the user; at other phases, KGC can access the identity list arbitrarily and obtain the registered users' identities. At the same time, ICA cannot get the master key of KGC.

Figure 2 gives the authentication and key issuing process of the anonymous key issuing-based IBE protocol. Noticing that, the scheme removes the restriction of communication between ICA and KGC in Fig. 1, i.e., KGC can access the identity list stored by ICA arbitrarily in the rest of phases, except in the key generation phase, where it only accesses the identity list through the index. Moreover, our protocol does not require secure channels, only the authentication channels between the user and ICA, and between the user and KGC to ensure that the transmitted information will not be tampered with.

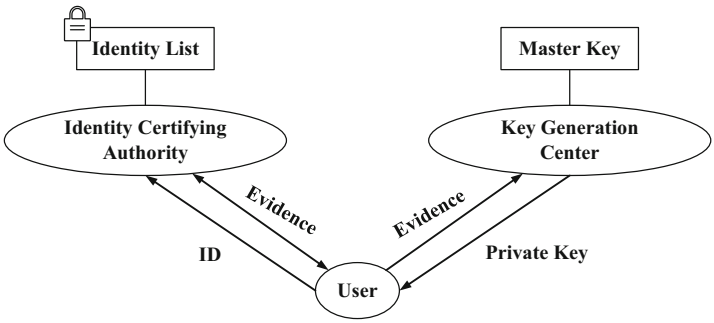

Fig. 2. Our system architecture

4.2 Security Model

The proposed scheme satisfies indistinguishability and anonymity against chosen plaintext attacks for user, KGC, and ICA, i.e., IND-ANON-CPA, IND-ANON-KGC, and IND-ANON-ICA security, respectively. The scheme is based on the DBDH problem under random oracle model. The formal security is defined as follows.

IND-ANON-CPA Security. The security is similar to that defined in the standard IBE scheme, which allows an adversary, i.e. a malicious participant, to query a private key of any user (except the challenge identity). The difference is that the adversary can access the verification oracle and obtain the evidence to access the key generation oracle. We allow the user to choose a signature algorithm Π_{Sig_user} of his own, so that the adversary can obtain the user's issuing key sk_{Sig_user}. IND-ANON-CPA security is defined as follows: adversary $\mathcal{A}$ adaptively makes at most q_c verification queries and q_s key queries to win the challenge game $Exp_{IBE,\mathcal{A}}^{IND-ANON-CPA}(\lambda)$ with at most probability ϵ as shown below.

1. $Setup(\lambda) \rightarrow (params)$: Challenger C runs the $Setup$ algorithm, holds master key s, and sends public key P_{pub} to $\mathcal{A}$. At the same time, C initializes an empty list of identities $IDList := \phi$.
2. Verification Query $\mathcal{A}^{CerO(ID_i)} \rightarrow (cert, y_{ID_i,1}, sk_{Sig\_user_i})$: $\mathcal{A}$ uses different identity ID_i $(i = 1, 2, \ldots, q_c)$ and issues corresponding verification query q_i. C runs the $CertifyID$ algorithm and responds to $\mathcal{A}$ with an evidence $cert$, a trapdoor massage $y_{ID_i,1}$ and the user's issuing key $sk_{Sig\_user_i}$. Finally C stores the query identity in $IDList$.
3. Key Query $\mathcal{A}^{SKO(cert)} \rightarrow M_{KGC}$: $\mathcal{A}$ submits $cert$ to the challenger. Then C runs the $IssueKey$ algorithm, computes message M_{KGC} and returns it to $\mathcal{A}$.
4. Challenge: $\mathcal{A}$ outputs an identity $ID^* \in \mathcal{ID}$ and a plaintext $M^* \in \mathcal{M}$ that he wants to challenge, but no verification query and key query on ID^* until then. C tosses a random coin $b \leftarrow \{0, 1\}$: if $b = 0$, he runs the *encryption* algorithm and returns the challenge ciphertext $C^* = Enc(P_{pub}, ID^*, M^*)$; otherwise he chooses a random ciphertext from the ciphertext space, i.e. $C^* \xleftarrow{Sample} \mathcal{C}$.
5. Guess: $\mathcal{A}$ outputs a guess $b' \leftarrow \{0, 1\}$ for b. If $b' = b$, it returns 1 and $\mathcal{A}$ wins the game, otherwise it returns 0. The advantage of $\mathcal{A}$ winning the game is

$$|Pr[Exp_{IBE,\mathcal{A}}^{IND-ANON-CPA}(\lambda) = 1] - \frac{1}{2}|$$

Definition 1. *An IBE scheme is (t, q_c, q_s, ϵ)-IND-ANON-CPA secure, if the adversary cannot win the challenge game with at least probability ϵ in polynomial time t.*

The above definition is mainly for CPA security, which can also be transformed to CCA security by adding decryption queries.

IND-ANON-KGC Security. This security is mainly defined for internal attacker (KGC) and allows adversary to obtain the master key s and some public parameters *params* at the beginning of the game.

Chow [14] proposed the security concept of anonymous ciphertext indistinguishability against KGC (ACI-KGC), whose security model provides an "embedded-identity encryption" query mechanism that allows an adversary (malicious KGC) to adaptively obtain multiple ciphertexts of the same unknown

identity ID^*. If the adversary can not identify the true identity of the recipient from ciphertext, it can only decrypt ciphertext by generating private keys of all users, which is impossible in polynomial time. The existence of ICA is not considered in Chow's security model, therefore the adversary can only access embedded identity encryption oracle, not key-generation oracle.

As described in [16], IND-ANON-KGC is a stricter and more practical security definition than ACI-KGC, allowing adversary to adaptively access two oracles, i.e., key-generation oracle and encryption oracle. The adversary gets the message M_{user} of an unknown-identity user from the key-generation oracle, and this identity is chosen randomly by challenger from $IDList$. For any plaintext message M and any identity ID in $IDList$, adversary can access the encryption oracle to get ciphertext. Since KGC does not have access to $IDList$, the adversary can only specify the index value to query.

IND-ANON-KGC security is defined as follows: adversary $\mathcal{A}$ wins the challenge game $Exp_{IBE,\mathcal{A}}^{IND-ANON-KGC}(\lambda)$ with at most probability ϵ after adaptively making at most q_I key issuing queries and q_E encryption queries.

1. $Setup(1^\lambda) \rightarrow (params)$: Challenger C runs the $Setup$ algorithm, and sends master key s and public key P_{pub} to $\mathcal{A}$. Then C initializes an empty identity list $IDList := \phi$ and a counter $count := 0$.
2. Key Issuing Query $\mathcal{A}^{IKO} \rightarrow M_{user}$: $\mathcal{A}$ makes a key issuing query. C randomly selects an identity $ID \in \mathcal{ID}$, calculates its evidence $cert$ and runs $ObtainKey$ algorithm to generate a one-round message M_{user} for $\mathcal{A}$. C stores ID in $IDList$ and updates $count := count + 1$.
3. Encryption Query $\mathcal{A}^{EnO(M,i)} \rightarrow C$: $\mathcal{A}$ submits an index i and a message $M \in \mathcal{M}$ to the challenger. C first checks whether the index value is satisfied $i < count$ and if so, C finds the i-th identity in $IDList$, runs the $encryption$ algorithm and returns the ciphertext $C = Enc(P_{pub}, ID_i, M)$. Otherwise, the challenger returns $\perp$, indicating that the query fails.
4. Challenge: $\mathcal{A}$ outputs an identity index i^* and a plaintext $M^* \in \mathcal{M}$ that he wants to challenge. C first checks whether index i^* is reasonable. If $i^* < count$, he finds the corresponding identity ID^* in $IDList$ and tosses a random coin $b \leftarrow \{0, 1\}$. If $b = 0$, C runs the $encryption$ algorithm and returns $C^* = Enc(P_{pub}, ID^*, M^*)$, but if $b = 1$, C selects a random ciphertext $C^* \xleftarrow{Sample} \mathcal{C}$. Otherwise, C reselects the index value.
5. Guess: $\mathcal{A}$ outputs a guess $b' \leftarrow \{0, 1\}$ for b. If $b' = b$, it returns 1 and $\mathcal{A}$ wins the game, otherwise it returns 0. The advantage of $\mathcal{A}$ winning the game is

$$|Pr[Exp_{IBE,\mathcal{A}}^{IND-ANON-KGC}(\lambda) = 1] - \frac{1}{2}|$$

Definition 2. *An IBE scheme is (t, q_I, q_E, ϵ)-IND-ANON-KGC secure, if the adversary cannot win the challenge game with at least probability ϵ in polynomial time t.*

IND-ANON-ICA Security. Since ICA is a third-party authentication authority introduced in standard IBE, we assume that he is not malicious and will not impersonate users, i.e. he will not generate an evidence alone to request a key issuing from KGC. Chow's work [14] considers a fully trusted ICA, while Emura [16] defines the security against a potentially malicious ICA that disallows arbitrary access to the key generation oracle (i.e., communication with KGC), which is unrealistic. In our security model, the honest but curious ICA has an evidence issuing-verifying key pair $(sk_{Sig\_ICA}, vk_{Sig\_ICA})$ and is able to adaptively access the evidence oracle and key generation oracle for arbitrary identities (except challenge identity ID^*). Here we only consider the case, where ICA launches a passive attack, eavesdrops on the key distributed by KGC for the user, and tries to learn information about the user's private key, instead of maliciously generating a key pair $(sk_{Sig\_ICA}, vk_{Sig\_ICA})$. The security against passive attack by ICA is explained in the security analysis.

IND-ANON-ICA security is defined as follows: adversary $\mathcal{A}$ wins the challenge game $Exp_{IBE,\mathcal{A}}^{IND-ANON-ICA}(\lambda)$ with at most probability ϵ after adaptively making at most q_c evidence queries and q_s key queries.

1. $Setup(1^\lambda) \rightarrow (params)$: Challenger C runs the $Setup$ algorithm, and sends public key P_{pub} and an evidence issuing-verifying key pair $(sk_{Sig\_ICA}, vk_{Sig\_ICA})$ to $\mathcal{A}$. Then C initializes an empty identity list $IDList := \phi$.
2. Evidence Query $\mathcal{A}^{CertO(ID)} \rightarrow cert$: $\mathcal{A}$ randomly selects identity $ID \in \mathcal{ID}$ and sends an evidence query. C stores ID in $IDList$, selects a trapdoor information $y_{ID,1} \in Z_p$, runs $GetCert$ algorithm to generate an evidence $cert$, and sends it with $y_{ID,1}$ to $\mathcal{A}$.
3. Key Query $\mathcal{A}^{SKO(cert)} \rightarrow M_{KGC}$: $\mathcal{A}$ submits $cert$ to the challenger. Then C runs $IssueKey$ algorithm, computes and returns a message M_{KGC} to $\mathcal{A}$.
4. Challenge: $\mathcal{A}$ outputs an identity $ID^* \in \mathcal{ID}$ and a plaintext $M^* \in \mathcal{M}$ to challenge. C tosses a random coin $b \leftarrow \{0,1\}$. If $b = 0$, C runs the $encryption$ algorithm and returns the challenge ciphertext $C^* = Enc(P_{pub}, ID^*, M^*)$; otherwise he returns a random ciphertext, i.e. $C^* \xleftarrow{Sample} \mathcal{C}$.
5. Guess: $\mathcal{A}$ outputs a guess $b' \leftarrow \{0,1\}$ for b. If $b' = b$, it returns 1 and $\mathcal{A}$ wins the game, otherwise it returns 0. The advantage of $\mathcal{A}$ winning the game is

$$|Pr[Exp_{IBE,\mathcal{A}}^{IND-ANON-ICA}(\lambda) = 1] - \frac{1}{2}|$$

Definition 3. *An IBE scheme is (t, q_c, q_s, ϵ)-IND-ANON-ICA secure, if the adversary cannot win the challenge game with at least probability ϵ in polynomial time t.*

4.3 The Proposed Protocol

The proposed anonymous key issuing-based IBE protocol contains the following PPT algorithms: Setup, Key Generation, Encryption and Decryption.

$\Pi_{Sig\_user} : (Sig\_user.KeyGen, Sig\_user.Sign, Sig\_user.Verify)$ and $\Pi_{Sig\_ICA} : (Sig\_ICA.KeyGen, Sig\_ICA.Sign, Sig\_ICA.Verify)$ are the digital signature algorithms chosen by user and ICA respectively. The algorithm must be able to provide EU-CMA security. The protocol is constructed as follows.

1. Setup: Given a security parameter λ, KGC performs the following steps (a-c), then ICA executes step (d).
   (a) Run the group generation algorithm and get the tuple $\{p, \mathbb{G}, \mathbb{G}_T, g, e\}$. Let p be a λ-bit prime number, g be a generator of $\mathbb{G}$ and $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ be a bilinear pairing, where $\mathbb{G}$ and $\mathbb{G}_T$ are two bilinear groups of order p.
   (b) Choose secure hash functions: $H : \{0,1\}^* \rightarrow \mathbb{G}$ for mapping user ID to $\mathbb{G}$, $H_2 : \mathbb{G}_T \rightarrow \{0,1\}^m$ (hash to the message space), where m is the plaintext length.
   (c) Randomly select system master key $s \leftarrow Z_p$ and calculate public key $P_{pub} = g^s$. The system parameter is $params = \{p, \mathbb{G}, \mathbb{G}_T, g, e, H, H_2, P_{pub}\}$.
   (d) Run the algorithm $\Pi_{Sig\_ICA}.KeyGen$ with $params$ as input, and obtain an evidence verifying key vk_{Sig_ICA} and an evidence issuing key sk_{Sig_ICA}.
2. Key Generation: Using the improved AKI protocol, the evidence is generated by ICA and user, and KGC distributes a private key for the user according to the evidence. The calculation process is mainly divided into two stages: evidence generation and key distribution.
   (a) Evidence Generation ($CertifyID \leftrightarrow GetCert$): ICA and user interactively run $CertifyID$ and $GetCert$, respectively, as shown in Fig. 3.
      i. User: Submit his identity ID to ICA for registration and verification.
      ii. ICA: Verify the user's identity. If the verification is successful, randomly select a trapdoor information $y_{ID,1} \in Z_p$ to generate a one-round pseudo-identity $u_{ID,1} = H(ID) \cdot g^{y_{ID,1}}$.

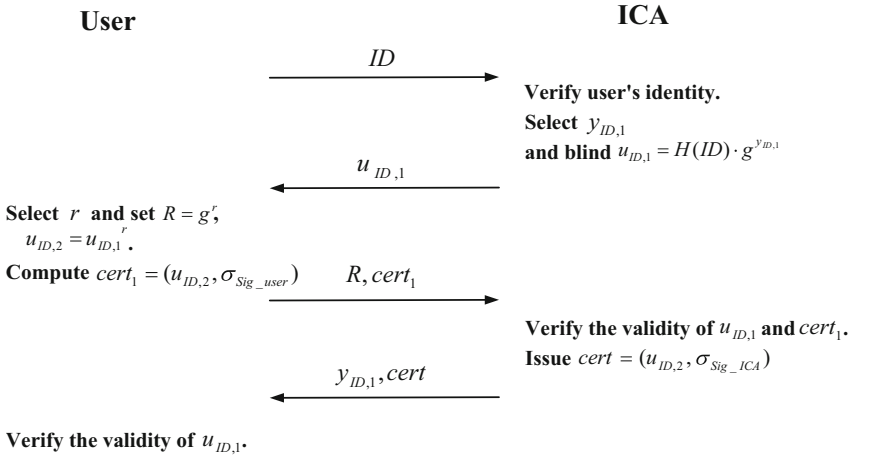

Fig. 3. Evidence Generation Process

iii. User: Randomly select a secret value $r \leftarrow Z_p^*$ and calculate the second-round pseudo-identity $u_{ID,2} = {u_{ID,1}}^r \in \mathbb{G}$ and $R = g^r$. In addition, select a signature algorithm Π_{Sig_user}, then obtain a signing key pair using $params$ to calculate $\sigma_{Sig\_user} \leftarrow \Pi_{Sig\_user}.Sign(sk_{Sig\_user}, u_{ID,2})$ and output the partial evidence $cert_1 = (u_{ID,2}, \sigma_{Sig\_user})$ as well as the verifying key vk_{Sig_user}. Then send the verification value R to ICA.

iv. ICA: Verify the second-round pseudo-identity $u_{ID,2}$ using the equation

$$e(u_{ID,2}, g) \stackrel{?}{=} e(u_{ID,1}, R) \tag{2}$$

to avoid the user misusing $u_{ID,1}$. If the verification fails, ICA does not issue an authentication evidence. Otherwise, ICA executes verification algorithm $\Pi_{Sig\_user}.Verify(vk_{Sig\_user}, u_{ID,2}, \sigma_{Sig\_user})$ and later calculates signature $\sigma_{Sig\_ICA} \leftarrow \Pi_{Sig\_ICA}.Sign(sk_{Sig\_ICA}, u_{ID,2})$. Then transmit the trapdoor information $y_{ID,1}$ and the evidence $cert = (u_{ID,2}, \sigma_{Sig\_ICA})$ to the user with an authentication channel.

(b) Key Distribution ($ObtainKey \leftrightarrow IssueKey$): The user and KGC run $ObtainKey$ and $IssueKey$ separately (see Fig. 4).

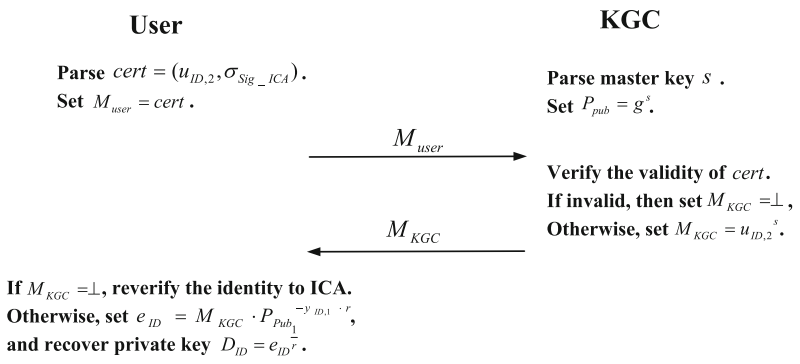

Fig. 4. Key Distribution Process

i. User: Set the one-round message $M_{user} = cert$ and request a private key from KGC, where $cert = (u_{ID,2}, \sigma_{Sig\_ICA})$.

ii. KGC: Execute the verification algorithm $\Pi_{Sig\_ICA}.Verify(vk_{Sig\_ICA}, u_{ID,2}, \sigma_{Sig\_ICA})$ to verify M_{user} using ICA's verifying key vk_{Sig_ICA}. If successful, issue a private key $y_{ID,2} = {u_{ID,2}}^s$ using master key s. Set message $M_{KGC} = y_{ID,2}$ and send to user using an authentication channel. If failing verification, set $M_{KGC} = \perp$.

iii. User: If the received message $M_{KGC} = \perp$, reverify his identity to ICA. Otherwise, based on the trapdoor information $y_{ID,1}$ transmitted by ICA and the secret value r chosen by himself, calculate $e_{ID} = M_{KGC} \cdot {P_{pub}}^{-y_{ID,1} \cdot r}$ and recover the real private key $D_{ID} = {e_{ID}}^{\frac{1}{r}} = H(ID)^s$.

(c) Encryption: For the plaintext M to be encrypted, the sender randomly selects $t \leftarrow Z_p^*$, calculates $U = g^t$ and $V = M \bigoplus H_2(e(Q_{ID}, P_{pub})^t)$. The ciphertext is $C = (U, V)$.
(d) Decryption: The receiver uses his own private key D_{ID} to decrypt the ciphertext C, i.e. $V \bigoplus H_2(e(D_{ID}, U)) = M$.

The protocol described above is still based on IBE, where the sender only knows the identity of recipient to encrypt the message without obtaining additional information.

4.4 Correctness Analysis

To verify the validity of the private key, user only needs to check whether the following equation holds against the system public key:

$$e(D_{ID}, g)? = e(Q_{ID}, P_{pub}) \tag{3}$$

where

$$\begin{aligned} D_{ID} = e_{ID}^{\frac{1}{r}} &= (y_{ID,2} \cdot {P_{pub}}^{-y_{ID,1} \cdot r})^{\frac{1}{r}} = ({u_{ID,2}}^s)^{\frac{1}{r}} \cdot {P_{pub}}^{-y_{ID,1}} \\ &= (H(ID)^{r \cdot s} \cdot g^{r \cdot s \cdot y_{ID,1}})^{\frac{1}{r}} \cdot g^{-y_{ID,1} \cdot s} = H(ID)^s \end{aligned} \tag{4}$$

5 Security Analysis

The anonymous key issuing-based IBE protocol solves the key escrow problem based on recipient anonymity and ciphertext indistinguishability. Anonymity ensures that KGC cannot identify the recipient identity from a given ciphertext to generate a private key for decryption. And ciphertext indistinguishability ensures that KGC cannot distinguish a ciphertext generated by IBE algorithm from a random ciphertext in the ciphertext space if the user's identity is already known to KGC. Therefore, under two security definitions, KGC cannot associate users with their ciphertexts. Also the use of AKI architecture makes KGC no longer have the identity list of registered users and cannot generate private keys of all users in polynomial time for brute force attack, thus better guarantees anonymity and ciphertext indistinguishability. Our protocol can maintain the same security level as Emura's scheme [16], while being secure against passive attacks from ICA. The formal security proof is as follows.

5.1 Anonymity and Indistinguishability

Theorem 1. *Assuming DBDH problem on the elliptic curve group is intractable and the underlying signature algorithm Π_{Sig_ICA} is EU-CMA secure, the IBE protocol with certified identities proposed in this paper is semantically secure against IND-ANON-CPA attack.*

Proof. As shown in Sect. 4.2, IND-ANON-CPA security is defined by a game between an adversary and a challenger. We set $\mathcal{A}$ as the adversary to break the security of IND-ANON-CPA, $\mathcal{B}$ as the PPT algorithm to break the DBDH problem. $\mathcal{B}$ interacts with adversary $\mathcal{A}$ as a challenger. Assume that $\mathcal{A}$ can make only one random oracle (random function H) query for the same input identity and that the corresponding ID has been queried by the random oracle before making the verification query and the key query.

In the initialization phase, challenger $\mathcal{B}$ runs *Setup* algorithm and sets the DBDH problem instance (g, g^a, g^b, g^c, Z), where g is the generator of $\mathbb{G}$. $\mathcal{B}$ sets system public key $P_{pub} = g^a$ and sends public parameters *params* to $\mathcal{A}$. Three empty lists are also initialized for security proofs, namely the identity list $IDList$, evidence list $EList$ and hash list $HList$. $\mathcal{B}$ tosses a random coin $\overline{c} \leftarrow \{0, 1\}$, assuming $Pr[\overline{c} = 1] = \alpha \in [0, 1]$, and then responds to the adversary's query in the following manner.

1. Random Oracle Query: $\mathcal{A}$ submits ID to query the random oracle, and $\mathcal{B}$ queries the list $HList$.
   (a) If $HList$ already contains a tuple $(ID, u_{ID}, \overline{b}, \overline{c})$, output $u_{ID} \in \mathbb{G}$.
   (b) Otherwise, $\mathcal{B}$ randomly selects $\overline{b} \leftarrow Z_p$. If $\overline{c} = 0$, compute $u_{ID} = g^{\overline{b}}$; otherwise compute $u_{ID} = (g^b)^{\overline{b}}$. Then $\mathcal{B}$ updates $HList$, adds a new tuple $(ID, u_{ID}, \overline{b}, \overline{c})$ and outputs u_{ID} to $\mathcal{A}$.
2. Verification Query: $\mathcal{A}$ uses identity ID for verification query. $\mathcal{B}$ randomly selects $y_{ID,1} \leftarrow Z_p$ and computes one-round pseudo-identity $u_{ID,1} = u_{ID} \cdot g^{y_{ID,1}}$. To avoid a one-to-many situation, the challenger uses the same trapdoor information for same ID. Then $\mathcal{B}$ selects a secret value $r \leftarrow Z_p^*$ and a signature algorithm Π_{Sig_user}, calculates $u_{ID,2} = {u_{ID,1}}^r$, $\sigma_{Sig\_user} \leftarrow \Pi_{Sig\_user}.Sign(sk_{Sig\_user}, u_{ID,2})$ and $\sigma_{Sig\_ICA} \leftarrow \Pi_{Sig\_ICA}.Sign(sk_{Sig\_ICA}, u_{ID,2})$. Finally, $\mathcal{B}$ returns trapdoor message $y_{ID,1}$, the evidence $cert = (u_{ID,2}, \sigma_{Sig\_ICA})$ and its issuing key sk_{Sig_user} to $\mathcal{A}$, and updates $IDList \leftarrow IDList \bigcup \{ID\}$, $EList \leftarrow EList \bigcup \{(cert, y_{ID,1}, ID, r, sk_{Sig\_user})\}$.
3. Key Query: $\mathcal{A}$ uses $cert = (u_{ID,2}, \sigma_{Sig\_ICA})$ to perform a key query. $\mathcal{B}$ executes $\Pi_{Sig\_ICA}.Verify(vk_{Sig\_ICA}, u_{ID,2}, \sigma_{Sig\_ICA})$ to verify the validity of pseudo-identity, if invalid, set $M_{KGC} = \perp$. Otherwise, $\mathcal{B}$ queries $EList$ and $HList$, recovers the tuple $(cert, y_{ID,1}, ID, r, sk_{Sig\_user})$ and $(ID, u_{ID}, \overline{b}, \overline{c})$. If $\overline{c} = 0$, $\mathcal{B}$ sends message $M_{KGC} = R^{(y_{ID,1} + \overline{b})}$ to $\mathcal{A}$, otherwise $\mathcal{B}$ terminates the game and forces $\mathcal{A}$ output the guessed value $\overline{c}'$.
4. Challenge: $\mathcal{A}$ submits identity ID^* and message M^* that he wants to challenge, and $\mathcal{B}$ queries $HList$ to get the tuple $(ID^*, u_{ID^*}, \overline{b^*}, \overline{c^*})$. If $\overline{c^*} = 1$, $\mathcal{B}$ sets $c_0^* = g^c$, $c_1^* = M^* \cdot Z^{\overline{b^*}}$; if $\overline{c^*} = 0$, $\mathcal{B}$ terminates the game and forces $\mathcal{A}$ output the guess value $\overline{c}'$.

According to the above analysis, if $Z = e(g,g)^{abc}$ and the game is not terminated, $c_1^* = M^* \cdot (e(g,g)^{abc})^{\overline{b^*}} = M^* \cdot e(g^{b \cdot \overline{b^*}}, g^a)^c = M^* \cdot e(H(ID^*), P_{pub})^c$. Assume that $\mathcal{A}$ makes h random oracle queries, the probability that $\mathcal{B}$ does not terminate the game during the queries is $\alpha(1-\alpha)^h$. Maximizing the probability value $\alpha = \frac{1}{1+h}$, the probability that $\mathcal{B}$ does not terminate the game at this point is at least $\frac{1}{\widehat{e}(1+h)}$, where $\widehat{e}$ is the base of the natural logarithm. Thus, assuming the DBDH problem is intractable, there is no adversary can breach the IND-ANON-CPA security with a non-negligible probability.

The detailed proofs of Theorem 2 and Theorem 3 are similar to Theorem 1. We only briefly explained here.

Theorem 2. *Assuming DBDH problem on the elliptic curve group is intractable, the IBE protocol with certified identities proposed in this paper is semantically secure against IND-ANON-KGC attack under the random oracle model.*

Proof. IND-ANON-KGC security is defined by a game between an adversary and a challenger. We set $\mathcal{A}$ as the adversary to break the IND-ANON-KGC security, $\mathcal{B}$ as the PPT algorithm to break the DBDH problem. $\mathcal{B}$ interact with adversary $\mathcal{A}$ as a challenger.

The difference with other security proofs is that at the beginning of the game, $\mathcal{B}$ needs to select a random index value I of $IDList$ and keep it secret. When $\mathcal{A}$ submits identity index i^* and message M^* that he wants to challenge, $\mathcal{B}$ first checks whether $i^* = I$ is satisfied, and if so terminates the game and forces $\mathcal{A}$ output a guessed value of the random coin, otherwise the game continues as before. If algorithm $\mathcal{B}$ can solve the DBDH problem with a non-negligible advantage ϵ using the given tuple, then $\mathcal{A}$ can compute the user's private key, decrypt ciphertext and win the challenge game. Moreover, under the random oracle, ID is theoretically hidden information to $\mathcal{A}$. It is simply impossible to learn the real user's identity and to generate all users' private keys to decrypt the challenger's encrypted ciphertext in polynomial time.

Theorem 3. *Assuming DBDH problem on the elliptic curve group is intractable, the IBE protocol with certified identities proposed in this paper is semantically secure against IND-ANON-ICA attack under the random oracle model.*

5.2 Security Against Passive Attack by ICA

We assume that ICA and KGC are not in collusion and can communicate freely. ICA verifies the user identity and issues an evidence, which guarantees to KGC that the user with the evidence is legitimate.

In the previous scheme [16], ICA selects a trapdoor information, provides the user with a pseudo-identity and signs it, and sends the trapdoor information to user through a secure channel. Even if ICA is honest, he may be curious. Once ICA eavesdrops the key distributed to a user by KGC, or even uses the

evidence to request a key from KGC, he can use the trapdoor information to recover user's private key. Of course, the evidence mentioned here is known to the user and not maliciously generated by ICA.

To address this passive attack by ICA, we have modified the evidence generation phase. The user's pseudo-identity in an evidence is jointly generated by ICA and user, providing secret value $y_{ID,1}$ and r respectively. ICA has no access to the secret value r used by user which hides his identity. In order to obtain the trapdoor information $y_{ID,1}$, the user must be verified by ICA after hiding the identity, otherwise the private key cannot be recovered. Additionally, the user needs to choose his own signature algorithm Π_{Sig_user} and sign the pseudo-identity $u_{ID,2}$, thus ensuring secure transmission of $u_{ID,2}$ and eliminating the need for secure channels.

The improved solution where the user and ICA cooperate to generate an evidence that hides the user's true identity and jointly ensure the user anonymity, successfully resists ICA's passive attack and effectively avoids the problem of users being out of control of ICA.

6 Comparison and Analysis

We compare several IBE schemes using AKI architecture. They do not change the basic architecture of BF-IBE, therefore we only consider the extra computational and communication overhead relative to the original IBE scheme.

Table 2. Extra Costs Comparison

Protocols	Signature	Verification	Interactive Communication
Chow et al. [14]	1	1	8
Emura et al. [16]	1	1	1
Ours	2	1+2	2

The AKI architecture is based on user anonymity, i.e. KGC cannot access the real identity of the user during the key generation phase. Since KGC owns the list of users' identity-password pairs in Sui's scheme [13], it can only provide anonymity against external attackers, which we called partial anonymity, while the rest of schemes can provide user anonymity against KGC. We do not compare the overhead of Sui's scheme. Chow's scheme [14] uses a commitment to hide the user's identity and therefore uses a complex secure two-party computation during the key generation phase. Secure two-party computation generally ensures the confidentiality of both inputs through obfuscation circuits, and even with hardware acceleration, its processing throughput and computational efficiency are low. Besides, the interactive protocol between KGC and users contains 8 complex steps, which is given in [20]. Similar to Emura's scheme [16], we use a blind signature algorithm to implement AKI protocol. In Emura's scheme, ICA

independently distributes an evidence for the user, containing a pseudo-identity and its signature, and KGC only needs to verify that the evidence is provided by ICA and then issue a key. While Our protocol requires two rounds of interaction between the user and ICA, i.e., twice blinding the identity and signing it, thus preventing ICA from intercepting and recovering the user's private key. This process involves one bilinear mapping verification and one signature verification to guarantee data integrity without a secure channel. Similarly, KGC requires one verification to check the validity of pseudo-identity. In Table 2, the numbers indicate the extra computation or communication times. Although our protocol has more computational and communication overhead than Emura's scheme, it is still under control and does not affect the efficiency of the system.

Through the above comparison, the protocol proposed in this paper can successfully solve the key escrow problem, and effectively resist the ICA's passive attack, which is more practical and more perfect.

7 Conclusion

In this paper, we propose an anonymous key issuing-based IBE protocol, and achieve recipient anonymity and ciphertext indistinguishability, which successfully solves the key escrow problem. AKI architecture separates the tasks of authentication and key issuing, which are performed by ICA and KGC respectively. The protocol is secure against attacks by malicious users and KGC, and solves the problem of the honest but curious ICA's passive attack with a more realistic design. Therefore, it has some theoretical guidance for the practical application of IBE.

References

1. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985). https://doi.org/10.1007/3-540-39568-7_5
2. Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. SIAM J. Comput. **32**(3), 586–615 (2003)
3. Chen, L., Harrison, K., Soldera, D., Smart, N.P.: Applications of multiple trust authorities in pairing based cryptosystems. In: Davida, G., Frankel, Y., Rees, O. (eds.) InfraSec 2002. LNCS, vol. 2437, pp. 260–275. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45831-X_18
4. Paterson, K.G.: Cryptography from pairings: a snapshot of current research. Inf. Secur. Tech. Rep. **7**(3), 41–54 (2002)
5. Hess, F.: Efficient identity based signature schemes based on pairings. In: Nyberg, K., Heys, H. (eds.) SAC 2002. LNCS, vol. 2595, pp. 310–324. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36492-7_20
6. Lee, B., Boyd, C., Dawson, E., Kim, K., Yang, J., Yoo, S.: Secure key issuing in id-based cryptography. In: Proceedings of the Second Workshop on Australasian Information Security, Data Mining and Web Intelligence, and Software Internationalisation, vol. 32, pp. 69–74. Citeseer (2004)

7. Sharma, D., Jinwala, D.: Id-based secure key generation protocol. In: 2011 2nd International Conference on Computer and Communication Technology (ICCCT-2011), pp. 416–421. IEEE (2011)
8. Li, X.Y.: Research on key escrow problem in identity-based cryptography. Xidian University. 2019
9. Kumar, M., Chand, S.: ESKI-IBE: efficient and secure key issuing identity-based encryption with cloud privacy centers. Multimedia Tools Appl. **78**(14), 19753–19786 (2019)
10. Gentry, C.: Certificate-based encryption and the certificate revocation problem. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 272–293. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-39200-9_17
11. Al-Riyami, S.S., Paterson, K.G.: Certificateless public key cryptography. In: Laih, C.-S. (ed.) ASIACRYPT 2003. LNCS, vol. 2894, pp. 452–473. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-40061-5_29
12. He, N.Q., L.Y., Zhang, H.: An identity-based online/offline encryption scheme without key escrow. J. Cyber Secur. **3**(2), 31–40 (2018)
13. Sui, A.F., et al.: Separable and anonymous identity-based key issuing. In: 11Th International Conference on Parallel and Distributed Systems (ICPADS 2005), vol. 2, pp. 275–279. IEEE (2005)
14. Chow, S.S.M.: Removing escrow from identity-based encryption. In: Jarecki, S., Tsudik, G. (eds.) PKC 2009. LNCS, vol. 5443, pp. 256–276. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00468-1_15
15. Wei, Q., Qi, F., Tang, Z.: Remove key escrow from the BF and gentry identity-based encryption with non-interactive key generation. Telecomm. Syst. **69**(2), 253–262 (2018). https://doi.org/10.1007/s11235-018-0461-1
16. Emura, K., Katsumata, S., Watanabe, Y.: Identity-based encryption with security against the KGC: a formal model and its instantiations. Theoret. Comput. Sci. **900**, 97–119 (2022)
17. Boldyreva, A.: Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 31–46. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36288-6_3
18. Cao, F., Cao, Z., Wang, L.: An improvement of an identity-based key issuing protocol. In: First International Multi-Symposiums on Computer and Computational Sciences (IMSCCS 2006), vol. 2, pp. 13–18. IEEE (2006)
19. Diffie, W., Van Oorschot, P.C., Wiener, M.J.: Authentication and authenticated key exchanges. Des. Codes Crypt. **2**(2), 107–125 (1992)
20. Chase, M.: Efficient Non-interactive Zero-Knowledge Proofs for Privacy Applications. Brown University, Providence (2008)