A Strong Identity Based Key-Insulated Cryptosystem*

Jin Li[1], Fangguo Zhang[2,3], and Yanming Wang[1,4]

[1] School of Mathematics and Computational Science,
Sun Yat-sen University, Guangzhou, 510275, P.R. China
sysjinli@yahoo.com.cn
[2] Department of Electronics and Communication Engineering,
Sun Yat-sen University, Guangzhou, 510275, P.R. China
[3] Guangdong Key Laboratory of Information Security Technology,
Sun Yat-sen University, Guangzhou, 510275, P.R. China
isszhfg@mail.sysu.edu.cn
[4] Lingnan College, Sun Yat-sen University,
Guangzhou, 510275, P.R.China
stswym@mail.sysu.edu.cn

Abstract. Key-insulated cryptosystem was proposed in order to minimize the damage of secret key exposure. In this paper, we propose a strong identity based (ID-based) key-insulated cryptosystem security model, including ID-based key-insulated encryption (IB-KIE) security model and ID-based key-insulated signature (IB-KIS) security model. Based on the security models, provably secure strong IB-KIE and IB-KIS schemes are constructed in order to decrease the damage of user's secret key exposure. These schemes are secure in the remaining time periods against an adversary who compromises the insecure device and obtains secret keys for the periods of its choice. Furthermore, the schemes remain secure (for all time periods) against an adversary who compromises only the physically-secure device. All the key-insulated encryption and signature schemes in this paper are provably secure in the random oracle model and support random-access key-updates.

Keywords: Key-insulated cryptosystem, ID-based, Bilinear pairings.

1 Introduction

The notion of key-insulated public key cryptosystem was first introduced by Dodis et al. [5] to minimize the damage of key exposures. In a certificate-based key-insulated public key cryptosystem, a user begins by registering a single public key pk which remains for the lifetime of the scheme. The secret key associated

* This work is supported by the National Natural Science Foundation of China (No. 60403007 and No. 10571181) and Natural Science Foundation of Guangdong Province, China (No. 04205407) and the Project-sponsored by SRF for ROCS, SEM.

X. Zhou et al. (Eds.): EUC Workshops 2006, LNCS 4097, pp. 352–361, 2006.

with a public key is here shared between the user and a physically-secure device. The master key is stored on a physically-secure device and a temporary secret key used to perform cryptographic operations is stored in an insecure device and updated regularly with the help of a physically-secure device that stores a master key.

In order to simplify key management procedures of certificate-based public key infrastructures (PKIs), Shamir [11] introduced the idea of ID-based cryptosystem in 1984. In such cryptosystem, the public key of a user is associated with his identity information ID and his private key s_{ID} is generated by a trusted third party called Private Key Generator (PKG). However, key exposure problem also exists in ID-based cryptosystem: User may still store his or her private key s_{ID} in an insecure place to do cryptographic protocols such as ID-based signature or decryption.

To mitigate the damage caused by the private key exposure in ID-based cryptosystem, one way is to construct ID-based key-insulated cryptosystem [10] that allows each user in this system update his or her private key periodically while keep the public key the same. In ID-based key-insulated cryptosystem, the user's master key s^*_{ID} is stored on a physically-secure device and a temporary secret key used to perform cryptographic operations is stored in an insecure device and updated regularly with the help of a physically-secure device. The lifetime of the protocol is divided into distinct periods $1, \cdots, N$. At the beginning of each period, the user interacts with the secure device to derive a temporary secret key which will be used to decrypt messages sent during that period. We denote by s^i_{ID} the temporary key for ID at period i, which is stored on an insecure device and will be used to perform cryptographic operations such as signing and deciphering for the time period i. On the other hand, the public key ID used to encrypt messages does not change at each period. We call a scheme ID-based (t, N)-key-insulated if an adversary who compromises the insecure device of an identity ID up to $t < N$ periods cannot break the remaining $N - t$ periods. Additionally, a scheme is called a strong ID-based (t, N)-key-insulated scheme if an adversary who compromises only the physically-secure device cannot break the scheme at any time periods. As also stated in [5], besides the direct application to minimizing the risk of key exposures across multiple time periods, ID-based key-insulated security may also be used to protect against key exposures across multiple locations, or users. Furthermore, it may also be used for purposes of delegation. Although the IB-KIS security model has been proposed recently [15] by Zhou, the strong secure IB-KIS security model and scheme still have not been formalized and constructed.

Contribution. First, we give a more efficient strong IB-KIE scheme. The scheme is ID-based $(N - 1, N)$-key-insulated encryption scheme. Then, the first strong IB-KIS security model and scheme are also presented. The schemes constructed in this paper have random-access key updates property. That is to say, it is possible to update the secret key of ID from s^i_{ID} to s^j_{ID} in one step.

2 Definitions and Security Model

2.1 Definition

Definition 1. [**IB-KIE**] *An IB-KIE consists of 7-tuple of poly-time algorithms (Setup, Extract, Gen, Upd*, Upd, Enc, Dec) defined as follows:*

- *Setup: is a probabilistic algorithm run by a private key generator (PKG) that takes as input a security parameter* 1^k*. It returns a public key* pk*, a master key* sk*.*
- *Extract: the ID-Extraction algorithm, that takes as input ID, master key* sk*, returns the secret key* s_{ID} *for ID.*
- *Gen: the user key generation algorithm, is a probabilistic algorithm that takes as input the private key* s_{ID} *and the total number of time periods* N*, outputs user's master private key* s^*_{ID} *and user's initial secret key* s^0_{ID}*.*
- *Upd*: the device key-update algorithm, is a probabilistic algorithm that takes as input indices* i, j *for time periods (*$1 \leq i, j \leq N$*) and the master private key* s^*_{ID}*. It returns a partial secret key* $s'^{i,j}_{ID}$ *.*
- *Upd: the user key-update algorithm, is a deterministic algorithm that takes as input indices* i, j*, a secret key* s^i_{ID}*, and a partial secret key* $s'^{i,j}_{ID}$*. It returns the secret key* s^j_{ID} *for time period* j*.*
- *Enc: the encryption algorithm, is a probabilistic algorithm which takes as input a public-key* pk*, a time period* i*, and a message* M*. It returns a ciphertext* (i, C) *for ID at time period* i*.*
- *Dec: the decryption algorithm, is a deterministic algorithm which takes as input a secret key* s^i_{ID} *and a ciphertext* (ID, i, C)*. It returns a message* M *or the special symbol* $\perp$*.*

We define the following oracles:

- $\mathcal{EO}$: The Extraction Oracle, on input ID, a master key sk, output the corresponding secret key s_{ID} by running algorithm Extract.
- $\mathcal{KEO}$: The Key Exposure Oracle, on input signer ID and i, the oracle first runs Extract(ID) to get s_{ID}, and gets (s^*_{ID}, s^0_{ID}) by running algorithm Gen. Then run $\mathsf{Upd}^*(0, i, s^*_{ID})$ to get $s'^{0,i}_{ID}$ followed by running $\mathsf{Upd}(0,\ i, s^0_{ID}, s'^{0,i}_{ID})$ to get s^i_{ID}, returns and stores the value s^i_{ID}.
- $\mathcal{DO}$: The Decryption Oracle, on input (ID, i, C), run Extract algorithm to get s^i_{ID}, return $\mathsf{Dec}_{s^i_{ID}}(i, C)$.

We say that an IB-KIE $\mathcal{E}$ is semantically secure against an adaptive chosen ciphertext attack (IND-ID-CCA) if no polynomially bounded adversary $\mathcal{F}$ has a non-negligible advantage against the challenger $\mathcal{C}$ in the following IND-ID-CCA game.

First, $\mathcal{C}$ runs Setup of the scheme. The resulting system parameters are given to $\mathcal{F}$. $\mathcal{F}$ issues the following queries as he wants:
Phase 1: $\mathcal{F}$ queries $\mathcal{EO}(ID), \mathcal{KEO}(ID, i)$ and $\mathcal{DO}(ID, i, C)$ in arbitrary interleave.

Challenge: Once the adversary decides that Phase 1 is over it outputs two equal length plaintexts M_0^*, M_1^*, period i and an identity ID* on which it wishes to be challenged. The challenger picks a random bit $b \in \{0,1\}$ and sets $C^* = Enc(ID^*, i, M_b^*)$. It sends (ID^*, i, C^*) as the challenge to the adversary.
Phase 2: $\mathcal{F}$ queries more $\mathcal{EO}(ID)$, $\mathcal{KEO}(ID, i)$ and $\mathcal{DO}(ID, i, C)$ in arbitrary interleave.
Guess: Finally, the adversary outputs a guess $b' \in \{0,1\}$. The adversary wins the game if $b' = b$ and ID*, (ID*, i), (ID^*, i, C^*) has never been queried to $\mathcal{EO}, \mathcal{KEO}$ and $\mathcal{DO}$, respectively. We refer to such an adversary $\mathcal{F}$ as an IND-ID-CCA adversary. We define adversary $\mathcal{F}$'s advantage in attacking the scheme $\mathcal{E}$ as the following function of the security parameter k: $Adv_{\mathcal{E},\mathcal{F}}(k) = |Pr[b = b'] - \frac{1}{2}|$.

The proof of security for our IB-KIE makes use of a weaker notion of security known as semantic security (also known as semantic security against a chosen plaintext attack or IND-ID-CPA) [2]. Semantic security is similar to chosen ciphertext security except that the adversary is more limited: it cannot issue decryption queries while attacking the challenge public key.

Definition 2. *We say that the IB-KIE $\mathcal{E}$ is semantically secure against an adaptive chosen plaintext attack if for any polynomial time* **IND-ID-CPA** *adversary $\mathcal{F}$ the function $Adv_{\mathcal{E},\mathcal{F}}(k)$ is negligible.*

Definition 3. *An IB-KIE has secure key updates if the view of any adversary $\mathcal{F}$ making a key-update exposure at (i, j) can be perfectly simulated by an adversary $\mathcal{F}'$ making key exposure requests at periods i and j.*

Definition 4. *An IB-KIE is called (t, n)-key-insulated if the scheme remains secure for the remaining $N - t$ time periods against any adversary $\mathcal{F}$ who compromises only the insecure device for t time periods.*

Definition 5. **[IB-KIS]** *An IB-KIS consists of 7-tuple of poly-time algorithms (Setup, Extract, Gen, Upd*, Upd, Sign, Vrfy.*

- Definition of algorithms Setup, Extract, Gen, Upd* and Upd are the same with corresponding algorithms in IB-KIE.
- Sign The signing algorithm, on input ID, i and message m, output signature σ.
- Vrfy The verification algorithm, on input σ, ID, i and message m, output 1 if it is true; otherwise, output 0.

We also define the signing oracle as follows:

- $\mathcal{SO}$: The signing Oracle, on input message M, ID, i, and partial secret key s_{ID}^i, output σ as the signature.

We define the following game: First, $\mathcal{C}$ runs Setup of the scheme. The adversary $\mathcal{F}$ can query $\mathcal{EO}, \mathcal{KEO}$ and $\mathcal{SO}$ adaptively. We say $\mathcal{F}$ wins the game if it outputs (ID, i, M, σ), such that ID, (ID, i) and (ID, i, m) are not equal to the inputs

of any query to $\mathcal{EO}$, $\mathcal{KEO}$ and $\mathcal{SO}$, respectively. σ is a valid signature of M for identity ID at period i.

3 A Strong ID-Based Key-Insulated Encryption Scheme

Our scheme uses bilinear pairings on elliptic curves. We now give a brief revision on the property of pairings and some candidate hard problems from pairings that will be used later.

Let $\mathbb{G}_1, \mathbb{G}_2$ be cyclic groups of prime order q, writing the group action multiplicatively. Let g be a generator of $\mathbb{G}_1$.

Definition 6. *A map $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ is called a bilinear pairing if, for all $x, y \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_q$, we have $e(x^a, y^b) = e(x, y)^{ab}$, and $e(g, g) \neq 1$.*

Definition 7. *Bilinear Diffie-Hellman (BDH) Problem: Given a randomly chosen $g \in \mathbb{G}_1$, as well as g^a, g^b, and g^c (for unknown randomly chosen $a, b, c \in_R \mathbb{Z}_q^*$), compute $e(g, g)^{abc}$.*

We say that the (t, ϵ)-BDH assumption holds in $\mathbb{G}_1$ if no t-time algorithm has non-negligible advantage ϵ in solving the BDH problem in $\mathbb{G}_1$.

3.1 The Scheme

1. **Setup:** To generate parameters for the system of time periods N, select a random generator $g \in \mathbb{G}_1$, a random $x \in \mathbb{Z}_q^*$, and set $g_1 = g^x$. Next, pick random elements $g_2, h, h_1, \cdots, h_N \in \mathbb{G}_1$, choose cryptographic hash functions $H_1 : \{0,1\}^* \to \mathbb{Z}_q$, $H_2 : \mathbb{G}_2 \to \mathbb{Z}_q$. The public parameters are $params = (g, g_1, g_2, h, h_1, \cdots, h_N, H_1, H_2)$, master key is g_2^x.
2. **Extract:** On input a private key g_2^x, to generate the private key s_{ID} for an identity ID, pick a random $r \in \mathbb{Z}_q^*$ and output the private key $s_{ID} = (g_2^x \cdot (g_1^{H_1(ID)} \cdot h)^r,\ g^r)$.
3. **Gen:** On input s_{ID}, parse it as $s_{ID}=(s_{ID}^{(1)}, s_{ID}^{(2)})$, choose a random element $\eta \in \mathbb{G}_1$, set $s_{ID}^* = (s_{ID}^{(1)}/\eta, s_{ID}^{(2)})$ and $s_{ID}^0 = (\eta, \phi, \phi, \phi)$.
4. **Upd*:** On input indices i, j and s_{ID}^*, parse s_{ID}^* as $(s_{ID}^{*(1)}, s_{ID}^{*(2)})$, choose $t \in_R \mathbb{Z}_q^*$ and return a partial secret key $s_{ID}'^{i,j} = (s_{ID}^{*(1)} \cdot h_j^t,\ s_{ID}^{*(2)},\ g^t)$.
5. **Upd:** On input indices i, j, a secret key s_{ID}^i and a partial secret key $s_{ID}'^{i,j} = (u', v', w')$, parse s_{ID}^i as $(s_{ID}^{i(1)},\ s_{ID}^{i(2)},\ s_{ID}^{i(3)}, s_{ID}^{i(4)})$. Output s_{ID}^j=$(s_{ID}^{j(1)},\ s_{ID}^{j(2)},\ s_{ID}^{j(3)},\ s_{ID}^{j(4)})$, where $s_{ID}^{j(1)} = s_{ID}^{i(1)}$ (in fact $s_{ID}^{i(1)} = \eta$ for all i), $s_{ID}^{j(2)} = s_{ID}^{i(1)} \cdot u'$, $s_{ID}^{j(3)} = v'$, $s_{ID}^{j(4)} = w'$ and erase $(s_{ID}^i,\ s_{ID}'^{i,j})$.
6. **Enc:** On input an index i of a time period, a message M, and an identity ID, pick $r' \in_R \mathbb{Z}_q^*$ and compute $\mathcal{C}$=$(A,\ B,\ C,\ D)$, where A=$H_2(e(g_1, g_2)^s) \oplus M$, $B = g^s$, $C = (g_1^{H_1(ID)} \cdot h)^s$, $D = h_i^s$.
7. **Dec:** On input $s_{ID}^i = (s_{ID}^{i(1)}, s_{ID}^{i(2)}, s_{ID}^{i(3)}, s_{ID}^{i(4)})$ and ciphertext $\mathcal{C} = (A,\ B,\ C,\ D)$ for an identity ID at period i, compute M=$A \oplus H_2(\frac{e(s_{ID}^{i(2)}, B)}{e(s_{ID}^{i(3)}, C) \cdot e(s_{ID}^{i(4)}, D)})$.

3.2 Correctness

The decryption of the IB-KIE is justified by the following equations:

$$\frac{e(s_{ID}^{i(2)}, B)}{e(s_{ID}^{i(3)}, C) \cdot e(s_{ID}^{i(4)}, D)} = \frac{e(g_1, g_2)^s \cdot e((g_1^{H_1(ID)} h)^r, g^s) \cdot e(h_i^t, g^s)}{e(g^r, (g_1^{H_1(ID)} h)^s) \cdot e(g^t, h_i^s)} = e((g_1, g_2)^s).$$

3.3 Security Analysis

Theorem 1. *The IB-KIE has secure key updates and supports random key updates.*

Proof. Let $\mathcal{F}$ be an adversary who makes a key-update exposure at (i, j). This adversary can be perfectly simulated by an adversary $\mathcal{F}'$ who makes key exposure requests at periods i and j. Since $\mathcal{F}'$ can get s_{ID}^i and s_{ID}^j, he can compute $s_{ID}^{\prime i,j} = (u', v', w')$, where $u' = s_{ID}^{j(2)} / s_{ID}^{j(1)}$, $v' = s_{ID}^{j(3)}$, $w' = s_{ID}^{j(4)}$. The proof that the scheme supports random key updates is trivial.

Theorem 2. *In the random oracle model, suppose the (t', ϵ')-BDH assumption holds in $\mathbb{G}_1$ and the adversary makes at most q_{H_1}, q_{H_2}, q_E and q_K times queries to hash functions H_1, H_2, private key extraction and key-exposure, respectively, then this ID-based key-insulated encryption scheme is $(t, q_{H_1}, q_{H_2}, q_E, q_K, \epsilon)$-semantically secure (IND-ID-CPA), where $t' < t + (2q_E + 4q_K)t_{exp}$ and t_{exp} is the maximum time for an exponentiation in $\mathbb{G}_1$, $\epsilon' \approx \frac{1}{q_{H_1} \cdot q_{H_2} \cdot q_K} \cdot \epsilon$.*

Proof is given in Appendix A.

Theorem 3. *The IB-KIE is a strong ID-based $(N-1, N)$-key-insulated encryption scheme.*

Proof. Assume an adversary $\mathcal{F}$ succeeds to attack the IB-KIE with access to the secure device, we will construct an algorithm $\mathcal{C}$ described below solves BDH problem in $\mathbb{G}_1$ for a randomly given instance $\{g, X = g^x, Y = g^y, Z = g^z\}$ and asked to compute $e(g, g)^{xyz}$. The details are as follows.

First, $\mathcal{C}$ puts $g_1 = X$ as the PKG's public key and sends it to $\mathcal{F}$. Then $\mathcal{C}$ randomly selects an element $s_{ID}^* \in \mathbb{G}_1^2$ and gives s_{ID}^* to $\mathcal{F}$. $\mathcal{C}$ will answer hash function, extract, key exposure queries as the proof in theorem 2. If $\mathcal{F}$ could break the scheme, from the simulation we can infer that $\mathcal{C}$ can solve the BDH problem as the proof in theorem 2. Meanwhile, in the proof of theorem 2, the adversary can query key exposure oracle up to $N-1$ (i.e. $q_K = N-1$) different time periods for an identity ID, so it is obvious that the key-insulated encryption scheme is $(N-1, N)$-key-insulated.

By using the technique due to Fujisaki-Okamoto [7], the scheme can be converted into a chosen ciphertext secure ID-based key-insulated system in the random oracle model.

4 A Strong ID-Based Key-Insulated Signature Scheme

An IB-KIS consists of 7-tuple of poly-time algorithms (Setup, Extract, Gen, Upd*, Upd, Sign, Vrfy. In this section, a strong IB-KIS is proposed based on the scheme in section 3.

- **Setup:** The public parameters are $params = (g, g_1, g_2, h, h_1, \cdots, h_N)$, master key is g_2^x, which is the same with IB-KIE. Define three hash functions as $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_q$ and $H_2 : \{0,1\}^* \rightarrow \mathbb{G}_1$.
- Algorithms Extract, Gen, Upd* and Upd are the same with the corresponding in IB-KIE.
- **Sign:** On input an index i of a time period, a message M, an identity ID, and s_{ID}^i=$(s_{ID}^{i(1)}, s_{ID}^{i(2)}, s_{ID}^{i(3)}, s_{ID}^{i(4)})$, pick $r' \in_R \mathbb{Z}_q^*$, output $\sigma = (A, B, C, D)$, where $A = s_{ID}^{i(2)}[H_3(M, g^{r'})]^{r'}$, $B = s_{ID}^{i(3)}$, $C = s_{ID}^{i(4)}$, $D = g^{r'}$).
- **Vrfy:** On input $\sigma = (A, B, C, D)$ on message M for an identity ID at period i, the verifier checks $e(g, A) \stackrel{?}{=} e(g_1, g_2) \cdot e(B, g_1^{H_1(ID)} h) \cdot e(C, h_j) \cdot e(D, H_3(M, D))$. Output 1 if it is true. Otherwise, output 0.

4.1 Security Analysis

Theorem 4. *If the CDH assumption holds in $\mathbb{G}_1$, then the IB-KIS is secure in the random oracle model.*

Proof. The proof is given in appendix B.

Theorem 5. *The IB-KIS is a strong $(N-1, N)$-IB-KIS, has secure key updates and supports random key updates.*

From the proof of theorem 1 and 3, the result can be easily deduced.

5 Conclusion

Key-insulated cryptosystem was proposed in order to minimize the damage of secret key exposure and has many other important applications. In order to decrease the damage of secret key exposure in identity based cryptosystem, a strong ID-based $(N-1, N)$-key-insulated encryption and a strong ID-based $(N-1, N)$-key-insulated signature schemes are proposed. The schemes in this paper are provably secure in the random oracle model and support random-access key-updates.

References

1. M. Bellare and S.K. Miner. *A Forward-Secure Digital Signature Scheme.* Crypto'99, pp. 431-448, Springer-Verlag, 1999.
2. D. Boneh and X. Boyen. *Efficient selective-ID identity based encryption without random oracles.* EuroCrypt'04, LNCS 3027, pp. 223-238, Springer-Verlag, 2004.
3. Y. Desmedt and Y. Frankel. *Threshold cryptosystems.* Crypto'89, LNCS 435, pp. 307-315, Springer-Verlag, 1989.

4. Y. Dodis, M. Franklin, J. Katz, A. Miyaji, and M. Yung, *Intrusion-resilient publickey encryption*. CT-RSA'03, LNCS 2612, pp. 19-32, Springer-Verlag, 2003.
5. Y. Dodis, J. Katz, S. Xu and M. Yung. *Key-Insulated Public-Key Cryptosystems*. EuroCrypt'02, pp. 65-82, Springer-Verlag, 2002.
6. Y. Dodis, J. Katz, S. Xu and M. Yung. *Strong Key-Insulated Signature Schemes*. PKC'03, LNCS 2567, pp. 130-144, Springer-Verlag, 2003.
7. E. Fujisaki and T. Okamoto. *Secure integration of asymmetric and symmetric encryption schemes*. Crypto'99, LNCS 1666, pp. 537-554, Springer-Verlag, 1999.
8. C. Gentry and A. Silverberg. *Hierarchical ID-Based Cryptography*. AsiaCrypt'02, LNCS 2501, pp. 548-566, Springer-Verlag, 2002.
9. M. Girault. *Relaxing Tamper-Resistance Requirements for Smart Cards Using (Auto)-Proxy Signatures*. CARDIS'98, LNCS 1820, pp. 157-166, Springer-Verlag, 1998.
10. Y.Hanaoka, G.Hanaoka, J.Shikata, H.Imai. *Identity-Based Hierarchical Strongly Key-Insulated Encryption and Its Application*. AsiaCrypt'05, LNCS 3788, pp. 495-514, Springer-Verlag, 2005.
11. A. Shamir. *How to share a secret*. Comm. 22(11):612-613, ACM, 1979.
12. A. Shamir. *Identity-based cryptosystems and signature schemes*. Crypto'84, LNCS 196, pp.47-53, Springer-Verlag, 1984.
13. D.Yao, N.Fazio, Y.Dodis, A.Lysyanskaya. *ID Based Encryption for Complex Hierarchies with Applications to Forward Security and Broadcast Encryption*. CCS'04, pp. 354-363, ACM, 2004.
14. D.H. Yum and P.J. Lee. *Efficient Key Updating Signature Schemes Based on IBS*. Cryptography and Coding'03, LNCS 2898, pp. 167-182, Springer-Verlag, 2003.
15. Y. Zhou, Z. Cao, Z. Chai. *Identity Based Key Insulated Signature*. ISPEC'06, LNCS 3903, pp. 226-234, Springer-Verlag, 2006.

Appendix A: Proof of Theorem 2

Proof

Suppose an adversary $\mathcal{F}$ has an advantage ϵ in attacking the scheme, we build an algorithm $\mathcal{C}$ that uses $\mathcal{F}$ to solve the BDH problem. Algorithm $\mathcal{C}$ is given a random $(g, X = g^x, Y = g^y, Z = g^z)$ and asked to compute $e(g,g)^{xyz}$.

Algorithm $\mathcal{C}$ publishes $params$=$(g, g_1 = X,\ g_2 = Y,\ h = g_1^{-\omega} g^{\omega'},\ h_1 = g_1^{\omega_1},\ \cdots,\ h_{k-1} = g_1^{\omega_{k-1}},\ h_k = g^{\omega_k},\ h_{k+1} = g_1^{\omega_{k+1}},\ \cdots,\ h_N = g_1^{\omega_N})$ as the public parameters, $k \in [1, N]$ is chosen randomly by $\mathcal{C}$. Algorithm $\mathcal{C}$ interacts with $\mathcal{F}$ as follows:

– Hash function query: There are two types of hash function query H_1 and H_2. $\mathcal{C}$ maintains a list of tuples called the H_1^{list} and chooses a random $k' \in [1, q_{H_1}]$. Initially the list is empty. If the query ID_i already appears on the H_1^{list} in a tuple (ID_i, a_i) then respond with $H_1(ID_i) = a_i$. Otherwise, $\mathcal{C}$ chooses $a_i \in_R \mathbb{Z}_q$ and answers $H_1(ID_i)$=a_i for $1 \leq i \leq q_{H_1}$ if $i \neq k'$. And answers $H_1(ID_i)$=ω if $i = k'$. At any time algorithm $\mathcal{F}$ may issue queries to the random oracle H_2. To respond to these queries, $\mathcal{C}$ maintains a list of tuples called the H_2^{list}. Each entry in the list is a tuple of the form (T_i, b_i). Initially the list is empty. To respond to query T_i algorithm $\mathcal{C}$ does the following: If the query T_i already appears on the H_2^{list} in a tuple (T_i, b_i)

then respond with $H_2(T_i) = b_i$. Otherwise, $\mathcal{C}$ just picks a random string b_i and $H_2(T_i) = b_i$ and adds the tuple to the list.

- $\mathcal{EO}$: If $\mathcal{F}$ issues extraction queries ID_i, $\mathcal{C}$ first find $H_1(ID_i) = a_i$ in the H_1^{list}, returns $s_{ID_i} = (Y^{\frac{\omega'}{\omega - a_i}}, Y^{\frac{1}{\omega - a_i}})$ to $\mathcal{F}$ as the response if $i \neq k'$. It is easy to verify this is a valid private key: Let $r = \frac{y}{\omega - a_i}$ (In fact, $\mathcal{C}$ doesn't know the value of r), then $s_{ID_i}=(g_2^x(g_1^{H_1(ID_i)} \cdot h)^r,\ g^r)=(Y^{\frac{\omega'}{\omega - a_i}},\ Y^{\frac{1}{\omega - a_i}})$. Otherwise, the process stops and $\mathcal{C}$ fails.
- $\mathcal{KEO}$: If $\mathcal{F}$ issues key exposure queries (ID_i, j), $\mathcal{C}$ first computes $s_{ID_i} = (Y^{\frac{\omega'}{\omega - a_i}}, Y^{\frac{1}{\omega - a_i}})$ and then returns $(Y^{\frac{\omega'}{\omega - a_i}} h_j^t, Y^{\frac{1}{\omega - a_i}}, g^t)$ to $\mathcal{F}$ as the response if $i \neq k'$. If $i = k'$ and $j \neq k$, $\mathcal{C}$ chooses $r \in \mathbb{Z}_q$ and returns $(g^{\omega' r}, g^r, Y^{-\frac{1}{\omega_j}})$ to $\mathcal{F}$ as the response. It is valid key for period j: Let $t = -\frac{y}{\omega_j}$, then $s_{ID_i}=(g_2^x(g_1^{H_1(ID_{k'})} \cdot h)^r \cdot h_j^t,\ g^r,\ g^t)=(g^{\omega' r}, g^r, Y^{-\frac{1}{\omega_j}})$ Otherwise, $\mathcal{C}$ fails and exits.

$\mathcal{F}$ outputs two messages M_0, M_1 and ID at time period j. If $ID = ID_{k'}$ and $j = k$, $\mathcal{C}$ picks a random bit $b \in \{0, 1\}$ and responds with the ciphertext as $C = (R \oplus M_b, Z, Z^{\omega'}, Z^{\omega_k})$ for a random $R \in Z_q$. The ciphertext is simulated correctly: Let $H_2(e(g, g)^{xyz}) = R$, then the ciphertext is $(H_2(e(g, g)^{xyz}) \oplus M_b, g^z, (g_1^{H_1(ID)} h)^z, h_j^z)=(R \oplus M_b, Z, Z^{\omega'}, Z^{\omega_k})$.

$\mathcal{F}$ issues more private key queries ID and key exposure queries (ID, j), restriction is that $ID \neq ID_{k'}$ and $j \neq k$. $\mathcal{A}$ responds as before.

This completes the description of algorithm $\mathcal{C}$ and $\mathcal{F}$ outputs guess b' with advantage ϵ'. If $\mathcal{C}$ does not abort, then, $\mathcal{C}$ chooses one of the q_{H_2} values T that is sent for H_2-query and outputs as the result to the BDH problem. For $\mathcal{F}$ has an advantage ϵ in attacking the scheme, from the simulation we can infer that $\mathcal{C}$ can solve the BDH problem with advantage $\epsilon' \approx \frac{1}{q_{H_1}} \cdot \frac{1}{q_{H_2}} \cdot \frac{1}{q_K} \epsilon$, which is the success probability of the events that $ID = ID_{k'}$, $j = k$ and the T is exact value randomly selected from H_2-query.

Appendix B: Proof of Theorem 4

Proof

If an adversary $\mathcal{A}$ succeeds to attack our scheme, then we can construct an algorithm $\mathcal{C}$ described below solves CDH problem for a randomly given instance $\{g, X = g^x, Y = g^y\}$ and asked to compute g^{xy}. The details are as follows.

The public parameters are the same with the simulation in theorem 2 as $params$=$(g, g_1 = X,\ g_2 = Y,\ h = g_1^{-\omega} g^{\omega'},\ h_1 = g_1^{\omega_1},\ \cdots,\ h_{k-1} = g_1^{\omega_{k-1}},\ h_k = g^{\omega_k},\ h_{k+1} = g_1^{\omega_{k+1}},\ \cdots,\ h_N = g_1^{\omega_N})$, $k \in [1, N]$ is chosen randomly by $\mathcal{C}$. Algorithm $\mathcal{C}$ interacts with $\mathcal{F}$ as follows:

- Hash function query: There are two types of hash function query H_1 and H_2. $\mathcal{C}$ maintains a list of tuples called the H_1^{list} and chooses a random $k' \in [1, q_{H_1}]$. Initially the list is empty. If the query ID_i already appears on the H_1^{list} in a tuple (ID_i, a_i) then respond with $H_1(ID_i) = a_i$. Otherwise, $\mathcal{C}$

chooses $a_i \in_R \mathbb{Z}_q$ and answers $H_1(ID_i)=a_i$ for $1 \leq i \leq q_{H_1}$ if $i \neq k'$. And answers $H_1(ID_i)=\omega$ if $i = k'$. To respond to H_2 queries, $\mathcal{C}$ maintains a list of tuples called the H_2^{list}. Each entry in the list is a tuple of the form (M_i, u_i, b_i). Initially the list is empty. To respond to query M_i, u_i, algorithm $\mathcal{C}$ does the following: If the query M_i, u_i already appears on the H_2^{list} in a tuple (M_i, u_i, b_i) then respond with $H_2(M_i, u_i) = g^{b_i}$. Otherwise, $\mathcal{C}$ just picks a random string $b_i \in \mathbb{Z}_q$, back patches $H_2(M_i, u_i) = g^{b_i}$, adds the tuple to the list.

- The simulation of $\mathcal{EO}$, $\mathcal{KEO}$ is the same with the proof in theorem 2.
- Signature query: On input (ID, i, M), $\mathcal{C}$ chooses $r, r', t \in \mathbb{Z}_q$, patches $H_2(M, Y^{-\frac{1}{c}})=X^c$ and returns $((g_1^{H_1(ID)}h)^r h_j^t, g^r, g^t, Y^{-\frac{1}{c}})$ to $\mathcal{F}$ as the response. It is valid signature from the view of adversary.

This completes the description of algorithm $\mathcal{C}$. After the simulation, the adversary outputs a forged ID-based key-insulated signature as (A, B, C, D) for identity $ID_{k'}$, at time period $i = k$ on a message M. Then $\mathcal{C}$ can solve CDH problem as follows: From H_2 list, $\mathcal{C}$ can recover the triple (M, D, b) such that $H_2(M\|D) = g^b$ with probability $1 - \frac{1}{q}$ (it is the probability that $\mathcal{F}$ does not query H_2 random oracle and outputs the correct value) . Then $g^{xy} = \frac{A}{B^{\omega'} \cdot C^{\omega} \cdot D^b}$, that is to say, $\mathcal{C}$ solves the CDH problem. The probability that $\mathcal{C}$ doesn't abort in $\mathcal{EO}$,$\mathcal{KEO}$ simulation is not less than $1 - \frac{q_E}{q_{H_1}}$ and $1 - \frac{q_K}{q_{H_1}}$. So, if IB-KIS is broken with non-negligible probability ϵ', then CDH problem can be solved with probability $(1 - \frac{q_E}{q_{H_1}}) \cdot (1 - \frac{q_K}{q_{H_1}}) \cdot (1 - \frac{1}{q})\epsilon'$. Then we can say that under the CDH assumption, the IB-KIS is secure.