# Generic Construction of Forward-Secure Identity-Based Encryption

Yang Lu
College of Computer and Information Engineering, Hohai University, Nanjing, China
Email: luyangnsd@163.com

Jiguo Li
College of Computer and Information Engineering, Hohai University, Nanjing, China
Email: lijiguo1688@163.com

***Abstract*—The primitive of forward-secure identity-based encryption was proposed in order to minimize the damage of private key exposure in identity-based encryption setting. This new primitive guarantees that even if the current private keys are compromised, it is not possible to compromise past secret keys and past communications. In this paper, we propose a generic construction of forward-secure identity-based encryption. To do so, we first introduce a relaxed variant of hierarchical identity-based encryption called identity-based binary tree encryption and construct a concrete identity-based binary tree encryption scheme that is provably chosen-ciphertext secure in the standard model. We then show how to generically construct forward-secure identity-based encryption from identity-based binary tree encryption. Based on the proposed identity-based binary tree encryption scheme, we obtain a forward-secure identity-based encryption scheme with chosen-ciphertext security in the standard model.**

***Index Terms*—forward security, identity-based encryption, identity-based binary tree encryption, chosen-ciphertext security, standard model**

## I. INTRODUCTION

Identity-based cryptography (IBC) was introduced by Shamir [1] to eliminate the need for public key certificates as used in traditional public key cryptography. In IBC, the public key of a user is his unique identity information such as e-mail address and telephone number, and his private key is generated by a trusted third party called Private Key Generator (PKG). Because the identity is a natural link to a user, the ability to use identities as public keys eliminates the need for public key certificates and certificate authorities. Although the notion of identity-based encryption (IBE) was introduced by Shamir in 1984 [1], it was until 2001 that a practical and provably secure IBE scheme was proposed by Boneh and Franklin [2]. Since then, IBE has undergone quite rapid development and a lot of schemes have been proposed, *e.g.* [3-9].

The standard security of IBE depends on the assumption that the private keys are kept perfectly secure. However, as more and more cryptographic operations are performed on insecure and unprotected devices in open environments, private key exposure seems to be inevitable. Actually, it is much easier for an adversary to obtain the private key from a naive user than to break the computational assumption on which the cryptosystem is based. Undoubtedly, private key exposure has become the most devastating attack on a public key cryptosystem, since it means all security guarantees are lost. To mitigate the damage caused by the private key exposure in IBE, one way is to build forward-secure IBE (FS-IBE) [10]. In a FS-IBE system, the lifetime of the system is divided into $N$ time periods labeled 0,…,$N$-1, and private keys are evolved at regular time periods with the time of the system. The basic idea is that a user *id*'s device begins by storing an initial private key $SK_{id|0}$ and this private key will evolve with time so that $SK_{id|0}$ will be used during time period 0, $SK_{id|1}$ will be used during time period 1, and so on. At the beginning of each time period $i$ ($i$ >0), the user *id*'s device applies some key-evolving algorithm to the previous private key $SK_{id|i-1}$ to derive the private key $SK_{id|i}$ which is used in the time period $i$, and then deletes the previous private key $SK_{id|i-1}$. On the other hand, the public key *id* used to encrypt messages remains fixed throughout the lifetime of the system. A forward-secure IBE scheme guarantees that an adversary who compromises the user *id*'s private key $SK_{id|i}$ for a time period $i$ will be unable to compromise his private keys for all time periods prior to $i$. Therefore, the paradigm of forward security provides an efficient approach to deal with the private key exposure problem in IBE setting.

The notion of forward security was first proposed in the context of key-exchange protocols by Günther [11] and later by Diffie *et al.* [12]. Subsequently, Anderson [13] suggested forward security for the non-interactive setting. The notion of non-interactive forward security was first formalized in the context of signature by Bellare and Miner [14], in which the first practical forward-secure signature scheme was proposed. Inspired by these initial works in [13, 14], a number of forward-secure signature schemes are proposed, *e.g.* [15-20]. Compared with signature schemes, there are fewer forward-secure encryption schemes. In [21], Bellare and Yee provided a comprehensive treatment of non-interactive forward security in the symmetric-key encryption setting. The


doi:10.4304/jcp.7.12.3068-3074

existence of non-interactive, forward-secure public key encryption (FS-PKE) schemes has been open since the question was first posed by Anderson in 1997 [13]. In 2003, Canetti *et al.* [22] proposed the first FS-PKE scheme based on the hierarchical identity-based encryption (HIBE) scheme proposed by Gentry and Silverberg [5]. In [23], Lu and Li proposed another FS-PKE scheme with shorter ciphertext and less decryption time than the one in [22]. The first forward-secure encryption scheme in IBE setting was proposed by Yao *et al.* [10]. In [10], Yao *et al.* proposed a forward-secure HIBE (FS-HIBE) scheme by combining the FS-PKE scheme proposed by Canetti *et al.* [22] with the HIBE scheme proposed by Gentry and Silverberg [5]. However, as the FS-PKE scheme in [22], the biggest drawback of Yao *et al.*'s scheme is that the complexity of any performance parameters is poly-logarithmic in terms of the total number of time periods. Clearly, Yao *et al.*'s scheme is quite inefficient for large values of $N$. In addition, the security of Yao *et al.*'s scheme is only proved in the random oracle model [24]. In [25], Yu *et al.* proposed a new FS-IBE scheme without random oracles based on Boneh *et al.*'s HIBE scheme [8]. Compared with the first level of Yao *et al.*'s FS-HIBE scheme in [10], Yu *et al.*'s FS-IBE scheme has shorter ciphertext and lower decryption cost. However, Yu *et al.*'s scheme only achieves chosen-plaintext security.

In this paper, we propose a generic construction of FS-IBE. We first introduce a notion called identity-based binary tree encryption (IB-BTE) which is a relaxed variant of HIBE and construct a concrete IB-BTE scheme which is chosen-ciphertext secure in the standard model. We then show that FS-IBE can be generically constructed from IB-BTE. We prove that the resulting FS-IBE scheme achieves chosen-ciphertext security if the underlying IB-BTE scheme satisfies chosen-ciphertext security. If taking the proposed IB-BTE scheme as input, our generic construction generates a chosen-ciphertext secure FS-IBE scheme without random oracles.

## II. PRELIMINARIES

### A. Bilinear Map and Computational Assumption

Let $p$ be a large prime number, $G_1$ and $G_2$ denote two multiplicative cyclic groups of the same order $p$. A mapping $e$: $G_1 \times G_1 \rightarrow G_2$ is called a bilinear map if it satisfies the following properties: (1) Bilinearity: $e(u^a, v^b) = e(u, v)^{ab}$ for all $u, v \in G_1$ and $a, b \in Z_p^*$; (2) Non-degeneracy: $e(g, g) \neq 1$ for a random generator $g \in G_1$; (3) Computablity: $e(u, v)$ can be efficiently computed for all $u, v \in G_1$.

The security of the schemes proposed in this paper is based on a complexity assumption called the truncated decision $q$-augmented bilinear Diffie-Hellman exponent ($q$-ABDHE) assumption proposed by Gentry in [9].

The truncated decision $q$-ABDHE problem in $(G_1, G_2)$ is as follows: Given a tuple $(g, g^{\alpha}, \ldots, g^{\alpha^q}, g', g'^{\alpha^{q+2}}) \in G_1^{q+3}$ and an element $T \in G_2$ as input, where $g$ and $g'$ be generators of $G$ and $\alpha$ be a random element in $Z_p^*$, decide whether $T = e(g, g')^{\alpha^{q+1}}$ or $T$ is a random element of $G_2$. Let $B$ be a probabilistic algorithm that takes as input a random instance of the truncated decision $q$-ABDHE problem and outputs a bit $b \in \{0,1\}$. The advantage of the algorithm $B$ in solving the above truncated decision $q$-ABDHE problem is defined to be

$$|\Pr\{B(g, g^{\alpha}, \ldots, g^{\alpha^q}, g', g'^{\alpha^{q+2}}, e(g, g')^{\alpha^{q+1}}) = 1\} - \Pr\{B(g, g^{\alpha}, \ldots, g^{\alpha^q}, g', g'^{\alpha^{q+2}}, T) = 1\}|$$

**Definition 1.** We say that the truncated decision ($t$, $\varepsilon$, $q$)-ABDHE assumption holds in ($G_1$, $G_2$) if no $t$-time algorithm has advantage at least $\varepsilon$ over random guessing in solving the truncated decision $q$-ABDHE problem in ($G_1$, $G_2$).

### B. Forward-Secure Identity-Based Encryption

Formally, a FS-IBE scheme is specified by the following five algorithms:

(1) Setup is the system initialization algorithm that takes a security parameter $1^k$ and the total number of time periods $N$ as input, and outputs a master key *msk* and a list of public parameters *params*. Usually, this algorithm is performed by a PKG. After the algorithm is performed, the PKG keeps the master key *msk* secret and publishes the public parameters *params*.

(2) KeyExtract is the private key extraction algorithm that takes *params*, *msk* and a user's identity *id* as input, and outputs the user *id*'s initial private key $SK_{id|0}$. This algorithm is also performed by a PKG.

(3) KeyUpdate is the private key update algorithm that takes *params*, the index $i \in [0, N\text{-}1)$ of the current time period and the current private key $SK_{id|i}$ as input, and outputs the private key $SK_{id|i+1}$ for the following time period $i$+1.

(4) Encrypt is the encryption algorithm that takes *params*, the receiver's identity *id*, the index $i \in [0, N)$ of the current time period and a message $M$ as input, and outputs a ciphertext $C$ for the time period $i$.

(5) Decrypt is the decryption algorithm that takes *params*, the index $i \in [0, N)$ of the current time period, the current private key $SK_{id|i}$ and a ciphertext $C$ as input, and outputs a message $M$ or a special symbol $\perp$ if the ciphertext $C$ is invalid.

The chosen-ciphertext security for FS-IBE schemes is defined via following game:

**Setup:** The challenger runs Setup($1^k$, $N$) to generate *params* and *msk*. It gives the adversary *params* and keeps *msk* to itself.

**Phase 1:** In this phase, the adversary can adaptively make a series of key extraction queries and decryption queries, and the challenger responds as follows:

(1) When receiving a key extraction query on ($id$, $i$) where $i \in [0, N)$, the challenger first runs KeyExtract to generate an initial private key $SK_{id|0}$ for the identity *id*, then runs KeyUpdate recursively to derive a private key $SK_{id|i}$ for the time period $i$. Finally, the challenger outputs the key $SK_{id|i}$ to the adversary.

(2) When receiving a decryption query on ($id$, $i$, $C$), the challenger first generates a private key $SK_{id|i}$ as above,

then runs Decrypt to decrypt the ciphertext $C$. Finally, it outputs the resulting plaintext to the adversary.

**Challenge:** Once the adversary decides that Phase 1 is over, it outputs an identity $id^*$, a time period $i^*$, and two equal length plaintexts $M_0$ and $M_1$ on which it wishes to be challenged. The constraint is that no key extraction query has been issued on $(id^*, j)$, where $0 \le j \le i^*$. The challenger first chooses a random bit $b \in \{0, 1\}$ and computes $C^* = \mathsf{Encrypt}(params, id^*, i^*, M_b)$. Then, it outputs $C^*$ as the challenge ciphertext to the adversary.

**Phase 2:** The adversary issues more key extraction queries and decryption queries. The constraint is that the adversary can not make a key extraction query on $(id^*, j)$ where $0 \le j \le i^*$ and a decryption query on $(id^*, i^*, C^*)$. The challenger responds as in Phase 1.

**Guess:** Finally, the adversary outputs a guess $b' \in \{0, 1\}$ and wins the game if $b = b'$. The advantage of the adversary in the above game is defined to be $|\Pr[b = b'] - 1/2|$.

We call an adversary in the above game a FS-ID-CCA adversary.

**Definition 2.** A FS-IBE scheme is said to be $(t, q_k, q_d, \varepsilon)$-FS-ID-CCA secure if for any $t$-time FS-ID-CCA adversary that makes at most $q_k$ key extraction queries and $q_d$ decryption queries has advantage at most $\varepsilon$ in the above game.

Similarly, FS-ID-CPA security can be defined for FS-IBE schemes, if the adversary is disallowed to make any decryption queries in the above game.

## III. Identity-Based Binary Tree Encryption

In this section, we first formalize the definition and security of IB-BTE. Then we construct a concrete IB-BTE scheme and prove its security in the standard model.

### A. Definitions

The notion of IB-BTE can be viewed as a relaxed variant of HIBE and also can be viewed as a combination of IBE and binary tree encryption (BTE) [26]. As in HIBE, in IB-BTE each node has a corresponding secret key. To send a message for some node, one uses the name of the target node as the public key to encrypt the message. After receiving the ciphertext, the target node can decrypt the ciphertext using its secret key. Moreover, as in HIBE, the secret key of any node in IB-BTE can be used to derive the secret keys for the children of that node. The main difference between HIBE and IB-BTE is that the latter is a binary tree. Each IB-BTE tree is associated with an identity $id$. We denote the label of a node in IB-BTE by $id|\omega$, where $\omega$ is a binary code. The root of the IB-BTE tree is labeled with $id|\varepsilon$, where $\varepsilon$ is an empty string, and furthermore if an internal node is labeled with a binary code $id|\omega$ then its left child is labeled with $id|\omega 0$ and its right child is labeled with $id|\omega 1$. The root node should request for a root secret key $sk_{id|\varepsilon}$ from a PKG and each of other nodes can derive its secret key from the secret key of its father node. That is, any internal node can use its secret key to derive the secret keys of its two children.

Formally, an IB-BTE scheme is specified by the following five algorithms:

(1) Setup is the system initialization algorithm that takes a security parameter $1^k$ and a value $1^l$ representing the depth of the tree as input, and outputs a master key $msk$ and a list of public parameters $params$. Usually, this algorithm is performed by a PKG.

(2) KeyExtract is the private key extraction algorithm that takes $params$, $msk$ and a user's identity $id$ as input, and outputs the root secret key $sk_{id|\varepsilon}$ for the IB-BTE tree corresponding to the identity $id$. Usually, this algorithm is performed by a PKG.

(3) KeyDerive is the key derivation algorithm that takes $params$, the label of a node $id|\omega \in \{0,1\}^{<l}$, and the secret key $sk_{id|\omega}$ associated with the node $id|\omega$ as input, and outputs the secret keys $sk_{id|\omega 0}$ and $sk_{id|\omega 1}$ for the two children of the node $id|\omega$.

(4) Encrypt is the encryption algorithm that takes $params$, the receiver's identity $id$, the label of a node $id|\omega \in \{0,1\}^{\le l}$, and a message $M$ as input, and outputs a ciphertext $C$.

(5) Decrypt is the decryption algorithm that takes $params$, the label of a node $id|\omega \in \{0,1\}^{\le l}$, the secret key $sk_{id|\omega}$ associated with the node $id|\omega$ and a ciphertext $C$ as input, and outputs a message $M$ or a special symbol $\perp$ if the ciphertext $C$ is invalid.

For correctness, it is required that, for any message $M$, if $C = \mathsf{Encrypt}(params, id, id|\omega, M)$, then $M = \mathsf{Decrypt}(params, id|\omega, sk_{id|\omega}, C)$.

The chosen-ciphertext security for IB-BTE schemes is defined via following game in which the adversary is allowed to adaptively choose an identity and a node to be challenged:

**Setup:** The challenger first runs $\mathsf{Setup}(1^k, 1^l)$ to generate $params$ and $msk$. It then gives the adversary $params$ and keeps $msk$ to itself.

**Phase 1:** In this phase, the adversary can adaptively make a series of key extraction queries and decryption queries, and the challenger responds as follows:

(1) When receiving a key extraction query on $id|\omega$, the challenger first runs $\mathsf{KeyExtract}(params, msk, id)$ to generate a root secret key $sk_{id|\varepsilon}$ for the IB-BTE tree corresponding to the identity $id$, then runs the algorithm KeyDerive recursively to derive a secret key $sk_{id|\omega}$ for the node $id|\omega$. Finally, it outputs the secret key $sk_{id|\omega}$ to the adversary.

(2) When receiving a decryption query on $(id|\omega, C)$, the challenger first generates a secret key $sk_{id|\omega}$ for the node $id|\omega$ as above, then runs $\mathsf{Decrypt}(params, id|\omega, sk_{id|\omega}, C)$ to decrypt the ciphertext $C$. Finally, it outputs the resulting plaintext to the adversary.

**Challenge:** Once the adversary decides that Phase 1 is over, it outputs an identity $id^*$, a node $id^*|\omega^*$ and two equal length plaintexts $M_0$ and $M_1$ on which it wishes to be challenged. The restriction is that the adversary did not previously issue a secret key query for the node $id^*|\omega^*$ or a prefix of $id^*|\omega^*$. The challenger picks a random bit $b \in \{0, 1\}$ and computes $C^* = \mathsf{Encrypt}(params, id^*, id^*|\omega^*, M_b)$. . Then, it outputs $C^*$ as the challenge ciphertext to the adversary.

**Phase 2:** The adversary issues more key extraction queries and decryption queries as in Phase 1. The restriction is that the adversary can not make a key extraction query on $id^*|\omega^*$ or a prefix of $id^*|\omega^*$ and a decryption query on $(id^*|\omega^*, C^*)$. The challenger responds as in Phase 1.

**Guess:** Finally, the adversary outputs a guess $b' \in \{0, 1\}$ and wins the game if $b = b'$. The advantage of the adversary is defined by $|\Pr[b = b'] - 1/2|$.

We call an adversary in the above game an IND-ID&NODE-CCA adversary.

**Definition 3.** An IB-BTE scheme is $(t, \varepsilon, q_k, q_d)$-IND-ID&NODE-CCA secure if for any $t$-time IND-ID&NODE-CCA adversary that makes at most $q_k$ key extraction queries and $q_d$ decryption queries has advantage at most $\varepsilon$ in the above game.

### *B. Description of the Proposed IB-BTE Scheme*

Next, we present a concrete IB-BTE scheme and prove its security under the truncated decision $q$-ABDHE assumption in the standard model. Our construction is based on Gentry's IBE scheme [9] and Boneh *et al.*'s HIBE scheme [8].

For simplicity, we assume that the identities in the proposed IB-BTE scheme are elements in $Z_p^*$. Of course, we can extend our scheme to identities over $\{0, 1\}^*$ by first hashing the identities into elements in $Z_p^*$ using a collision resistant hash function $H$: $\{0, 1\}^* \to Z_p^*$. Let $G_1$ and $G_2$ be the groups of prime order $p$, and let $e$: $G_1 \times G_1 \to G_2$ be a bilinear map. The proposed IB-BTE scheme is described as follows:

(1) Setup($1^k$, $1^l$): The PKG selects a random generator $g \in G_1$ and a random element $\alpha \in Z_p^*$. It sets $g_1 = g^\alpha$. It furthermore randomly choose $g_2, h_1, \ldots, h_l \in G_1$ and a hash function $H$ from a family of universal one-way hash functions. The system public parameters *params* and the master key *msk* are given by *params* = $(g, g_1, g_2, h_1, \ldots, h_l, H)$ and $msk = \alpha$.

(2) KeyExtract(*params*, *msk*, *id*): The PKG randomly selects $r_{id} \in Z_p^*$ and sets $sk_{id|\varepsilon} = (r_{id}, (g_2 g^{-r_{id}})^{id/\alpha})$ as the root secret key for the IB-BTE tree corresponding to the identity *id*.

Notice that the secret key of the non-root node labeled with $id \mid \omega_1 \ldots \omega_k \in \{0,1\}^{k \le l}$ consists of one element of $Z_p^*$ and $2+l-k$ elements of $G_1$, and is formed as $sk_{id|\omega_1 \ldots \omega_k} = (r_{id}, (g_2 g^{-r_{id}})^{id/\alpha} \cdot (\prod_{i=1}^{k} h_i^{\omega_i})^r, g_1^r, h_{k+1}^r, \ldots, h_l^r)$, where $r \in_R Z_p^*$.

(3) KeyDerive(*params*, *id*|$\omega$, $sk_{id|\omega}$): If $id|\omega = id|\varepsilon$, this algorithm randomly selects $s \in Z_p^*$ and computes

$$sk_{id|0} = (r_{id}, (g_2 g^{-r_{id}})^{id/\alpha}, g_1^r, h_2^r, \ldots, h_l^r),$$

$$sk_{id|1} = (r_{id}, (g_2 g^{-r_{id}})^{id/\alpha} \cdot h_1^r, g_1^r, h_2^r, \ldots, h_l^r).$$

Else, let $id|\omega = id|\omega_1 \ldots \omega_k$, it first parses the secret key $sk_{id|\omega}$ as $(a_0, a_1, a_2, b_{k+1}, \ldots, b_l)$, where $a_0 = r_{id}$, $a_1 = (g_2 g^{-r_{id}})^{id/\alpha} \cdot (\prod_{i=1}^{k} h_i^{\omega_i})^r$, $a_2 = g_1^r$, $b_{k+1} = h_{k+1}^r, \ldots, b_l = h_l^r$. It then randomly chooses $s \in Z_p^*$ and computes

$$sk_{id|\omega\omega_{k+1}} = (a_0, a_1 \cdot b_{k+1}^{\omega_{k+1}} \cdot (\prod_{i=1}^{k+1} h_i^{\omega_i})^s, a_2 \cdot g_1^s, b_{k+2} \cdot h_{k+2}^s, \ldots, b_l \cdot h_l^s),$$

where $\omega_{k+1} \in \{0,1\}$.

It is easy to deduce that

$$a_1 \cdot b_{k+1}^{\omega_{k+1}} \cdot (\prod_{i=1}^{k+1} h_i^{\omega_i})^s$$

$$= (g_2 g^{-r_{id}})^{id/\alpha} \cdot (\prod_{i=1}^{k} h_i^{\omega_i})^r \cdot (h_{k+1}^r)^{\omega_{k+1}} \cdot (\prod_{i=1}^{k+1} h_i^{\omega_i})^s$$

$$= (g_2 g^{-r_{id}})^{\frac{id}{\alpha}} \cdot (\prod_{i=1}^{k+1} h_i^{\omega_i})^{r+s},$$

$$a_2 \cdot g_1^s = g_1^r \cdot g_1^s = g_1^{r+s},$$

$b_i \cdot h_i^s = h_i^r \cdot h_i^s = h_i^{r+s}$ for each $i = k+2, \ldots, l$.

If let $r' = r + s$, then we get that $sk_{id|\omega\omega_{k+1}} = (r_{id}, (g_2 g^{-r_{id}})^{id/\alpha} \cdot (\prod_{i=1}^{k+1} h_i^{\omega_i})^{r'}, g_1^{r'}, h_{k+2}^{r'}, \ldots, h_l^{r'})$. Therefore, it is a valid secret key for the node $id|\omega_1 \ldots \omega_k \omega_{k+1}$.

(4) Encrypt(*params*, *id*, *id*|$\omega$, *M*): If $id|\omega = id|\varepsilon$, the sender selects a random $t \in Z_p^*$ and computes the ciphertext

$$C = (c_1, c_2, c_3, c_4) = (M \cdot e(g, g_2)^{-id \cdot t}, e(g, g)^{id \cdot t}, g_1^t, (h_1 h_2^\beta)^t),$$

where $\beta = H(c_1, c_2, c_3)$.

Else, let $id \mid \omega = id \mid \omega_1 \ldots \omega_k \in \{0,1\}^k$, the sender selects a random $t \in Z_p^*$ and computes

$$C = (c_1', c_2', c_3', c_4', c_5') = (M \cdot e(g, g_2)^{-id \cdot t}, e(g, g)^{id \cdot t}, g_1^t, (\prod_{i=1}^{k} h_i^{\omega_i})^t, (h_1 h_2^\beta)^t),$$

where $\beta = H(c_1', c_2', c_3', c_4')$.

(5) Decrypt(*params*, *id*|$\omega$, $sk_{id|\omega}$, *C*): If $id|\omega = id|\varepsilon$, the receiver first parses $C$ as $(c_1, c_2, c_3, c_4)$ and $sk_{id|\varepsilon}$ as $(a_0, a_1)$, computes $\beta = H(c_1, c_2, c_3)$, and then verifies whether $e(g_1, c_4) = e(c_3, h_1 h_2^\beta)$. If so, it outputs the plaintext

$$M = c_1 \cdot e(c_3, a_1) \cdot c_2^{a_0}$$

Else, let $id|\omega = id|\omega_1 \ldots \omega_k$, the receiver first parses $C$ as $(c_1', c_2', c_3', c_4', c_5')$ and $sk_{id|\omega}$ as $(a_0, a_1, a_2, b_{k+1}, \ldots, b_l)$, and then verifies whether $e(g_1, c_5') = e(c_3', h_1 h_2^\beta)$, where $\beta = H(c_1', c_2', c_3', c_4')$. If so, it outputs the plaintext

$$M = \frac{c_1' \cdot e(c_3', a_1) \cdot c_2'^{a_0}}{e(a_2, c_4')}$$

We now prove that the above IB-BTE scheme is IND-ID&NODE-CCA secure under the truncated decision $q$-ABDHE assumption.

**Theorem 1.** *Let $q = q_k + 1$. Assume that the truncated decision $(t, \varepsilon, q)$-ABDHE assumption holds in $(G_1, G_2)$. Then, the above IB-BTE scheme is $(t', \varepsilon, q_k, q_d)$-IND-ID&NODE-CCA secure for $t' = t - O(t_{exp} \cdot q^2 \cdot l)$, where $t_{exp}$ is the time required to compute the exponent in $G_1$.*

***Proof.*** Assume that $A$ is a $(t', q_k, q_d, \varepsilon')$ - IND-ID&NODE-CCA adversary. We show how to construct an algorithm $B$ to solve the truncated decision $q$-ABDHE

problem with advantage at least $\varepsilon$ and in time at most $t$. At the beginning of the game, the algorithm $B$ takes as input a random truncated decision $q$-ABDHE challenge $(g, g^{\alpha},\ldots, g^{\alpha^q}, g', g'^{\alpha^{q+2}}, T)$, where $T$ is either $e(g,g')^{\alpha^{q+1}}$ or a random element of $G_2$. The algorithm $B$'s goal is to output 1 when $T = e(g,g')^{\alpha^{q+1}}$ and 0 otherwise. To do so, it interacts with the adversary $A$ as follows:

In the setup phase, the algorithm $B$ chooses a random polynomial function $f(x) \in Z_p[x]$ of degree $q$ with $f(0) \neq 0$, and sets $g_1 = g^{\alpha}$ and $g_2 = g^{f(\alpha)}$. Clearly, $g_2$ can be computed from $(g, g^{\alpha},\ldots, g^{\alpha^q})$ which is known to the algorithm $B$. It then randomly chooses $r_1,\ldots,r_l \in Z_p^*$ and sets $h_i = g_1^{r_i}$ for $i = 1,\ldots,l$. It further chooses a one-way hash function $H: \{0,1\}^* \rightarrow Z_p^*$. Finally, it provides the adversary $A$ with *params* = $(g, g_1, g_2, h_1,\ldots, h_l, H)$ as the public parameters. The master key is $\alpha$ which is unknown to the algorithm $B$.

In the question-answer phase, the algorithm $B$ responds the adversary $A$'s queries as follows:

(1) When receiving a key extraction query on $id|\omega$, if $id = \alpha$, then the algorithm $B$ can use $\alpha$ to solve the truncated decision $q$-ABDHE problem immediately. Otherwise, let $F_{id}(x)$ denote the ($q$-1)-degree polynomial function $\frac{(f(x)-f(0))\cdot id}{x}$. If $id|\omega = id|\varepsilon$, the algorithm $B$ sets the secret key of the node $id|\varepsilon$ to be $sk_{id|\varepsilon} = (f(0), g^{F_{id}(\alpha)})$ which is a valid secret key of the root node $id|\varepsilon$ as $g^{F_{id}(\alpha)} = (g_2 g^{-f(0)})^{id/\alpha}$. Else, let $id|\omega = id \,|\, \omega_1 \ldots \omega_k \in \{0,1\}^k$, the algorithm $B$ chooses a random $r \in Z_p^*$ and sets the secret key of the node $id|\omega$ to be $sk_{id|\omega} = (a_0, a_1, a_2, b_{k+1},\ldots, b_l)$, where $a_0 = f(0)$, $a_1 = g^{F_{id}(\alpha)} \cdot (\prod_{i=1}^{k} h_i^{\omega_i})^r$, $a_2 = g_1^r$, $b_{k+1} = h_{k+1}^r, \ldots, b_l = h_l^r$. It is easy to verify that $sk_{id|\omega}$ is a valid secret key of the node $id|\omega$ as $a_1 = (g_2 g^{-f(0)})^{id/\alpha} \cdot (\prod_{i=1}^{k} h_i^{\omega_i})^r$.

(2) When receiving a decryption query on $(id|\omega, C)$, the algorithm $B$ first generates a secret key $sk_{id|\omega}$ for the node $id|\omega$ as above, and then runs Decrypt(*params*, $id|\omega$, $sk_{id|\omega}$, $C$) to decrypt $C$.

In the challenge phase, the adversary $A$ outputs $(id^*, id^*|\omega^*, M_0, M_1)$ on which it wishes to be challenged. If $id^* = \alpha$, then the algorithm $B$ can use $\alpha$ to solve the truncated decision $q$-ABDHE problem immediately. Otherwise, let $\omega^* = \omega_1^* \ldots \omega_h^* \in \{0,1\}^h$, it chooses a random bit $b \in \{0,1\}$ and computes a secret key $sk_{id^*|\omega^*} = (a_0^*, a_1^*, a_2^*, b_{h+1}^*,\ldots, b_l^*)$ for the node $id^*|\omega^*$ as in Phase 1. Then, the algorithm $B$ computes $C^* = (c_1^*, c_2^*, c_3^*, c_4^*, c_5^*)$ as

$$c_1^* = M_b \cdot e(c_3^*, a_1^*)^{-id^*} \cdot e(a_2^*, c_4^*)^{id^*} \cdot (c_2^*)^{-id^* \cdot a_0^*}, \; c_2^* = T^{id^*},$$

$$c_3^* = (g')^{\alpha^{q+2}}, \; c_4^* = \prod_{i=1}^{h} (g')^{\alpha^{q+2} \cdot r_i \cdot \omega_i^*}, \; c_5^* = (g')^{\alpha^{q+2}(r_1 + r_2 \beta^*)},$$

where $\beta^* = H(c_1^*, c_2^*, c_3^*, c_4^*)$. It outputs $C^*$ as the challenge ciphertext to the adversary $A$. Let $s = \log_g g' \cdot \alpha^{q+1}$. If $T = e(g,g')^{\alpha^{q+1}}$, then $c_1^* = M_b \cdot e(g,g_2)^{-id^* \cdot s}$, $c_2^* = e(g,g)^{id^* \cdot s}$, $c_3^* = (g_1)^s$, $c_4^* = \prod_{i=1}^{h} (h_i^{\omega_i^*})^s$, $c_5^* = (h_1 h_2^{\beta^*})^s$. Therefore, $C^*$ is a valid ciphertext of the message $M_b$. Notice that the challenge ciphertext is $C^* = (c_1^*, c_2^*, c_3^*, c_4^*)$ if $id^*|\omega^* = id^*|\varepsilon$.

Finally, the adversary $A$ outputs a guess $b' \in \{0,1\}$ for the bit $b$. The algorithm $B$ outputs a guess as follows: if $b = b'$, then it outputs 1 meaning $T = e(g,g')^{\alpha^{q+1}}$; otherwise, it outputs 0 meaning $T$ is a random element of $G_2$.

***Probability Analysis*:** If $T = e(g,g')^{\alpha^{q+1}}$, then the challenge ciphertext $C^*$ is a valid encryption of $M_b$ under the identity $id^*$ and the node $id^*|\omega^*$ chosen by the adversary $A$. Therefore, the view of the adversary $A$ is identical to its view in a real attack game, and thus its guess satisfies $|\Pr[b = b'] - 1/2| \geq \varepsilon$. On the other hand, when $T$ is a random element of $G_2$, the second component of the challenge ciphertext is a random element of $G_2$ and provides no information to the adversary $A$, then the adversary $A$'s guess satisfies $\Pr[b = b'] = 1/2$. Therefore, we have that the algorithm $B$'s advantage in solving the given truncated decision $q$-ABDHE problem satisfies $|(1/2 \pm \varepsilon) - 1/2| = \varepsilon$.

***Time Complexity Analysis*:** In the simulation, the algorithm $B$'s overhead is dominated by computing the secret key in response to the adversary $A$'s key extraction queries. Each such computation requires computing $O(q \cdot l)$ exponentiations in $G_1$. Since A makes at most $q - 1$ such queries, we get that $t = t' + O(t_{exp} \cdot q^2 \cdot l)$.

## IV. Constructing FS-IBE from IB-BTE

In this section, we show how to generically construct a FS-IBE scheme from an IB-BTE scheme. In our generic construction of FS-IBE, we take the IB-BTE tree as the key-evolving tree to update the users' private keys in the FS-IBE scheme. To do so, we associate the time periods with all nodes of the IB-BTE tree. Assume that the total number of time periods $N \leq 2^{l+1}$-1, we require that the IB-BTE tree has depth $l$. For an identity $id$, let $id|\omega^{(i)}$ denote the node associated with the time period $i$ ($0 \leq i \leq N$-1), we associate the time periods with all nodes of the IB-BTE tree according to a pre-order traversal as follows:

(1) Set $id|\omega^{(0)}$ to be the root node of the tree (*i.e.*, $id|\omega^{(0)} = id|\varepsilon$).

(2) If $id|\omega^{(i)}$ is an internal node then $id|\omega^{(i+1)} = id|\omega^{(i)}0$;

(3) Else if $id|\omega^{(i)}$ is a leaf node and $i < N$-1 then $id|\omega^{(i+1)} = id|\omega'1$, where $\omega'$ is the longest binary string such that $\omega'0$ is $\omega^{(i)}$ or a prefix of $\omega^{(i)}$.

The private key $SK_{id|i}$ for a given time period $i$ in the resulting FS-IBE scheme consists of two parts: the secret

key for the node $id|\omega^{(i)}$ (which is acted as the decryption key in the time period $i$) and the secret keys for all right siblings of the nodes on the path from the root to the node $id|\omega^{(i)}$ (which are used to update the private key from $SK_{id|i}$ to $SK_{id|i+1}$ at the beginning of the time period $i$+1).

We now present the details of our generic construction of FS-IBE. Let IB-BTE = (Setup, KeyExtract, KeyDerive, Encrypt, Decrypt) be an IB-BTE scheme. Then, a FS-IBE scheme FS-IBE = (Setup, KeyExtract, KeyUpdate, Encrypt, Decrypt) can be constructed as follows:

(1) FS-IBE.Setup($1^k$, $N$): To setup the system, run IB-BTE.Setup($1^k$,$1^l$) to generate *params* and *msk*, where $l$ is the smallest integer satisfying $N \le 2^{l+1}$-1, and output the public parameters $params' = params \cup \{N\}$ and the master key $msk' = msk$.

(2) FS-IBE.KeyExtract($params'$, *msk*, *id*): To generate an initial private key $SK_{id|0}$ for the identity *id*, run IB-BTE.KeyExtract(*params*, *msk*, *id*) to generate a root secret key $sk_{id|\varepsilon}$ for the IB-BTE tree corresponding to the identity *id*, and output $SK_{id|0} = sk_{id|\varepsilon}$.

(3) FS-IBE.KeyUpdate($params'$, $i$, $SK_{id|i}$): To generate the private key $SK_{id|i+1}$ for the time period $i$+1, perform as follows: If the node $id|\omega^{(i)}$ according to the time period $i$ is an internal node, then run IB-BTE.KeyDerive(*params*, $id|\omega^{(i)}$, $sk_{id|\omega^{(i)}}$) to generate the secret keys $sk_{id|\omega^{(i)}0}$ and $sk_{id|\omega^{(i)}1}$ for the two child nodes of the node $id|\omega^{(i)}$, and output $SK_{id|i+1} = \{ sk_{id|\omega^{(i)}0}, sk_{id|\omega^{(i)}1} \} \cup \{ SK_{id|i} - \{ sk_{id|\omega^{(i)}} \} \}$; Else if $id|\omega^{(i)}$ is a leaf node, then simply output $SK_{id|i+1} = SK_{id|i} - \{sk_{id|\omega^{(i)}}\}$.

(4) FS-IBE.Encrypt($params'$, *id*, $i$, $M$): To encrypt a message $M$ under the public key *id* in the time period $i$, run IB-BTE.Encrypt(*params*, *id*, $id|\omega^{(i)}$, $M$) to encrypt $M$ and output the resulting ciphertext.

(5) FS-IBE.Decrypt($params'$, $i$, $SK_{id|i}$, $C$): To decrypt a ciphertext $C$ under the public key *id* in the time period $i$, run IB-BTE.Decrypt(*params*, $id|\omega^{(i)}$, $sk_{id|\omega^{(i)}}$, $C$) to decrypt the ciphertext $C$ and output the result. Note that the secret key $sk_{id|\omega^{(i)}}$ of the node $id|\omega^{(i)}$ is the first component of the private key $SK_{id|i}$ for the time period $i$.

For the security of the scheme FS-IBE, we have the following theorem.

**Theorem 2.** *The scheme FS-IBE is ($t$, $q_k$, $q_d$, $\varepsilon$)-FS-ID-CCA secure if the scheme IB-BTE is ($t$, $q_k$, $q_d$, $\varepsilon$)-IND-ID&NODE-CCA secure.*

***Proof.*** Let $A$ be a FS-ID-CCA adversary with advantage $\varepsilon$ against the scheme FS-IBE with $N$ time periods. We show how to make use of the adversary $A$ to construct an IND-ID&NODE-CCA adversary $B$ against the scheme IB-BTE with depth $l$, where $l$ is the smallest integer satisfying $N \le 2^{l+1}$-1.

In the setup phase, the adversary $B$ is given the system public parameters *params* by its challenger in the IND-ID&NODE-CCA game, and then forwards *params* with $N$ as the public parameters to the adversary $A$ in the FS-ID-CCA game. Then, the adversary $B$ associates the time periods with all nodes of the binary tree in the scheme IB-BTE according to a pre-order traversal as described above.

In the question-answer phase, the adversary $B$ responds the adversary $A$'s queries as follows:

(1) When receiving a key extraction query on (*id*, $i$), let $\omega^{(i)}$ denote the node associated with the time period $i$, if $i$=0, the adversary $B$ forwards $id|\varepsilon$ to its challenger to ask for a root secret key $sk_{id|\varepsilon}$ for the binary tree corresponding to the identity *id*, and outputs $sk_{id|\varepsilon}$ to the adversary $A$ as the initial private key $SK_{id|0}$ for the identity *id*; otherwise, the adversary $B$ first obtains the initial private key $sk_{id|\varepsilon}$ from its challenger, and derives $SK_{id|i}$ from $sk_{id|\varepsilon}$ by performing the algorithm IB-BTE.KeyDerive recursively or making the key extraction query recursively to obtain all the secret keys for the node $id|\omega^{(i)}$ and all right siblings of the nodes on the path from the root to $id|\omega^{(i)}$, then outputs $SK_{id|i}$ to the adversary $A$.

(2) When receiving a decryption query on (*id*, $i$, $C$), let $\omega^{(i)}$ denote the node associated with the time period $i$, the adversary $B$ forwards ($id|\omega^{(i)}$, $C$) to its decryption oracle and outputs the result to the adversary $A$.

In the challenge phase, the adversary $A$ outputs ($id^*$, $i^*$, $M_0$, $M_1$) on which it wishes to be challenged. Let $\omega^{(i^*)}$ denote the node associated with the time period $i^*$, the adversary $B$ outputs ($id^*$, $id^*|\omega^{(i^*)}$, $M_0$, $M_1$) to its challenger. Then, it outputs the challenge ciphertext from its challenger to the adversary $A$ as the challenge ciphertext in the FS-ID-CCA game.

Finally, after the adversary $A$ outputs its guess $b'$ for $b$, the adversary $B$ outputs the same $b'$ as its own guess.

It is readily seen that the adversary $B$ perfectly simulates all the oracles for the adversary $A$. If the adversary $A$ succeeds in guessing the bit $b$, then it also succeeds in outputting the correct bit. Therefore, the advantage that the adversary $B$ breaks the IND-ID&NODE-CCA security of the scheme IB-BTE is $\varepsilon$. This completes the proof of Theorem 2.

Now, we can apply our generic construction to any IND-ID&NODE-CCA secure IB-BTE scheme to generate a FS-ID-CCA secure FS-IBE scheme. If taking our proposed IB-BTE scheme as input, our generic construction generates a quite efficient FS-IBE scheme without random oracles. From Theorem 1 and Theorem 2, we can directly conclude that this FS-IBE scheme satisfies the FS-ID-CCA security. It is easy to see that, in the derived FS-IBE scheme, the time required for key extraction and decryption and the length of ciphertext are independent on the total number of time periods $N$, and any other performance parameter has at most $\log^2 N$ complexity. Therefore, it is quite efficient. Most importantly, it achieves chosen-ciphertext security in the standard model.

## V. CONCLUSIONS

In this paper, we proposed a generic construction of FS-IBE. The proposed generic construction of FS-IBE

can generically convert a chosen-ciphertext secure IB-BTE scheme into a chosen-ciphertext secure FS-IBE scheme. We built a quite efficient IB-BTE scheme and proved its security in the standard model. Based on the proposed IB-BTE scheme, we obtained a practical and chosen-ciphertext secure FS-IBE scheme without random oracles.

## ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China [No. 61272542] and the Fundamental Research Funds for the Central Universities of China [No. 2010B06414].

## REFERENCES

[1] A. Shamir, "Identity-based cryptosystems and signature schemes," In Advances in Cryptology- CRYPTO'84, USA, LNCS 196, pp. 47-53, 1984.
[2] D. Boneh. and M. Franklin, "Identity-based encryption from the Weil pairing," In Advances in Crptology - CRYPTO'01, USA, LNCS 2139, pp.213-229, 2001.
[3] C. Cocks, "An identity based encryption scheme based on quadratic residues," In Advances in Cryptography and Coding 2001, LNCS 2260, pp.360-363, 2001.
[4] J. Horwitz and B. Lynn, "Toward hierarchical identity-based encryption," In Advances in Cryptology - Eurocrypt 2002, LNCS 2332, pp. 466-481, 2002.
[5] C. Gentry and A. Silverberg, "Hierarchical ID-based cryptography," In Advances in Cryptology - Asiacrypt 2002, LNCS 2501, pp. 548-566, 2002.
[6] D. Boneh and X. Boyen, "Efficient selective-ID identity based encryption without random oracles," In Advances in Cryptology - Eurocrypt 2004, LNCS 3027, pp. 223-238, 2004.
[7] B. Waters, "Efficient identity-based encryption without random oracles," In Advances in Cryptology - Eurocrypt 2005, LNCS 3494, pp. 114-127, 2005.
[8] D. Boneh, X. Boyen and E.J. Goh, "Hierarchical identity based encryption with constant size ciphertext," In Advances in Cryptology - Eurocrypt 2005, LNCS 3494, pp. 440-456, 2005.
[9] C. Gentry, "Practical identity-based encryption without random oracles," In Advances in Cryptology - Eurocrypt 2006, LNCS 4404, pp. 445-464, 2006.
[10] D. Yao, N. Fazio, Y. Dodis, and A. Lysyanskaya, "ID-based encryption for complex hierarchies with applications to forward security and broadcast encryption," In 11th ACM Conference on Computer and Communications Security, pp. 354-363, 2004.
[11] C.G. Günther, "An identity-based key-exchange protocol," In Advances in Cryptology – Eurocrypt 1989, LNCS 434, pp.29-37, 1990.
[12] W. Diffie, P. C. Van-Oorschot, and M. J. Weiner, "Authentication and authenticated key exchanges," Des., Codes, Cryptography, 2(2), pp. 107-125, 1992.
[13] R. Anderson, "Two remarks on public key cryptology," Invited Lecture of ACM CCS'97. Available at http://www.cl.cam.ac.uk/ftp/users/rja14/forwardsecure.pdf.
[14] M. Bellare and S. Miner, "A forward-secure digital signature scheme," In Advances in Cryptology - CRYPTO'99, USA, LNCS 1666, pp. 431-448, 1999.
[15] H. Krawczyk, "Simple forward-secure signatures from any signature scheme," In 7th ACM Conference on Computer and Communications Security, pp. 108-115, 2000.
[16] M. Abdalla and L. Reyzin, "A new forward-secure digital signature scheme," In Advances in Cryptology - Asiacrypt 2000, LNCS1976, pp. 116-129, 2000.
[17] M. Abdalla, S.K. Miner, and C. Namprempre, "Forward-secure threshold signature schemes," In Topics in Cryptography - CT-RSA 2001, LNCS 2020, pp. 441-456, 2001.
[18] G. Itkis and L. Reyzin, "Forward-secure signatures with optimal signing and verifying," In Advances in Cryptology - Crypto 2001, LNCS, 2139, pp. 499-514, 2001.
[19] A. Kozlov and L. Reyzin, "Forward-secure signatures with fast key update," In Security in Communication Networks, LNCS 2576, pp. 247-262, 2002.
[20] T. Malkin, D. Micciancio, S.K. Miner, "Efficient generic forward-secure signatures with an unbounded number of time periods," In Advances in Cryptology - Eurocrypt 2002, LNCS 2332, pp. 400-417, 2002.
[21] M. Bellare and B. Yee, "Forward security in private-key cryptography," In RSA Cryptographers' Track - CT-RSA 2003, LNCS 2612, pp. 1-18, 2003.
[22] R. Canetti, S. Halevi, J. Katz, "A forward-secure public-key encryption scheme," In Advances in Cryptology - Eurocrypt 2003, LNCS 2656, pp. 255-271, 2003.
[23] Y. Lu and J. Li, "A Practical Forward-Secure Public-Key Encryption Scheme," Journal of Networks, 6(9), pp. 1254-1261, 2011.
[24] M. Bellare and P. Rogaway, "Random oracles are practical: a paradigm for designing efficient protocols," In 1st ACM Conference on Communications and Computer Security, pp. 62-73, 1993.
[25] Jia Yu, Fanyu Kong, Xiangguo Cheng, Rong Hao, Jianxi Fan, "Forward-secure identity-based public-key encryption without random oracles," Fundamenta Informaticae, 111(2), pp. 241-256, 2011.
[26] J. Katz, "Binary tree encryption: constructions and applications," In 6th International Conference on Information Security and Cryptology, LNCS 2971, pp. 1-11, 2004.

**Yang Lu** was born in Yangzhou City, Jiangsu Province, P.R. China, in 1977. He received the B.S. degree in mathematics and the M.S. degree in computer science from Nanjing Normal University in 2000 and 2003 respectively, and his Ph.D. degree in information security science and technology from PLA University of Science and Technology in 2009. He has been working in HoHai University from 2003. Currently, he is an Assistant Professor in College of Computer and Information Engineering. He has published more than 30 papers in International conferences/journals and Chinese journals. His major research interests include network security and cryptography.

**Jiguo Li** received the B.S. degree from Heilongjiang University in 1996, M.S. and Ph.D. degree from Harbin Institute of Technology in 2000 and 2003. He has been working in HoHai University from 2003. Currently, he is a Professor in College of Computer and Information Engineering. His major research interests include information security and cryptography.