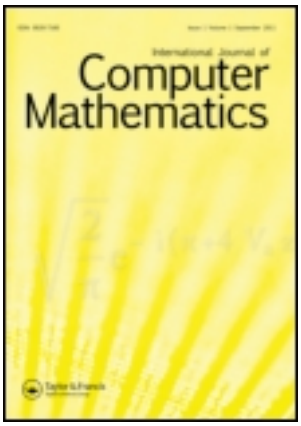

International Journal of Computer Mathematics
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/gcom20

New forward-secure public-key encryption without random oracles

Yang Lu[a] & Jiguo Li[a]
[a] College of Computer and Information Engineering, Hohai University, No. 8, Focheng Xi Road, Jiangning, Nanjing 211100, Jiangsu Province China

Accepted author version posted online: 31 May 2013.Published online: 01 Jul 2013.

To cite this article: Yang Lu & Jiguo Li (2013) New forward-secure public-key encryption without random oracles, International Journal of Computer Mathematics, 90:12, 2603-2613, DOI: 10.1080/00207160.2013.807915

To link to this article: http://dx.doi.org/10.1080/00207160.2013.807915

International Journal of Computer Mathematics, 2013
Vol. 90, No. 12, 2603–2613, http://dx.doi.org/10.1080/00207160.2013.807915

New forward-secure public-key encryption without random oracles

Yang Lu* and Jiguo Li

College of Computer and Information Engineering, Hohai University, No. 8, Focheng Xi Road, Jiangning, Nanjing 211100, Jiangsu Province, China

(*Received 12 September 2012; revised version received 21 January 2013; second revision received 5 May 2013; third revision received 19 May 2013; accepted 20 May 2013*)

Forward-secure public-key cryptography is an important technique for protecting private keys. It provides the benefits of frequent updating private keys without changing public keys. The most attractive property of forward security is that even if an attacker obtains the private key for the current time period, she still cannot compromise the private keys for the past time. In this paper, we newly present a forward-secure public-key encryption scheme without random oracles and prove it to be chosen-ciphertext secure in the standard model. In the proposed scheme, the ciphertext size and the decryption time have no correlation with the number of time periods and other performance indices have at most poly logarithmic complexities in terms of the number of time periods. As far as we know, it is the first forward-secure public-key encryption scheme that achieves direct chosen-ciphertext security in the standard model.

Keywords: public-key cryptography; private key exposure; forward security; chosen-ciphertext security; standard model

2010 AMS Subject Classifications: 94A60; 68P25

1998 ACM Computing Classification System Codes: E.3; K.6.5

1. Introduction

In public-key cryptography, each user has a pair of keys, namely public key and private key. The public key is usually published and publicly accessible while the private key is kept secret by its owner. The security of public-key cryptography hinges on the condition that the users' private keys are kept secret, but this is very difficult to achieve in reality. As our world is growing increasingly dependent on the digital devices, the cryptographic computations are performed more frequently on some insecure digital devices. The exposure of private keys seems unavoidable as the attackers have a wide range of methods for obtaining a private key from an insecure device.

The goal of forward-secure public-key cryptography is to provide the benefits of frequent updating private keys without changing public keys. More concretely, the forward-security technique enables a user to frequently evolve his private key while maintaining a same public key. In a forward-secure public-key cryptosystem the whole lifetime of the system is divided into T time periods. A user begins by generating an initial private key and a public key. At the end of each

*Corresponding author. Email: luyangnsd@163.com

time period, a new private key that will be used in the next time period is evolved from the old key, and then the old key is deleted. Meanwhile, this user's public key keeps unchanged during the whole lifetime of the cryptosystem. The forward-security property implies that even if an attacker obtains a user's private key for the current time period, she still cannot compromise this user's private keys for the past time. Therefore, the forward-security technique can effectively limit the damages caused by the exposure of private keys.

1.1 *Related work*

The concept of forward security can be traced back to the notion of perfect forward secrecy for the interactive key exchange protocols [8,11]. In 1997, Anderson [3] first introduced forward security into non-interactive public-key cryptography and proposed a generic construction of forward-secure signature. Subsequently, Bellare and Miner [4] presented the formal definition of forward-secure signature alone with a practical forward-secure signature scheme. Since then, the topic of forward-secure signature has attracted great interest and many forward-secure signature schemes have been proposed [1,2,12–14,18]. Moreover, forward security in the context of symmetric-key encryption was also considered by Bellare and Yee [5].

The first non-interactive and forward-secure public-key encryption scheme was presented by Canetti *et al.* [7] in 2003. This scheme is built on the hierarchical identity-based encryption by Gentry and Silverberg [10] and achieves chosen-plaintext security in the standard model. In [16], Lu and Li proposed the second forward-secure public-key encryption scheme. This scheme is based on the hierarchical identity-based encryption by Boneh *et al.* [6] and achieves chosen-plaintext security in the standard model too. Compared with Canetti *et al.*'s scheme [7], this scheme has obvious advantage in both the ciphertext size and the decryption time. In [20], Yao *et al.* introduced forward security into the identity-based setting and presented a hierarchical identity-based encryption scheme with forward security. In [21], Yu *et al.* proposed the first forward-secure identity-based encryption scheme that is proven secure in the standard model. In [15], Lu proposed another forward-secure identity-based encryption scheme without random oracles from the identity-based encryption scheme by Gentry [9]. Compared with Yu *et al.*'s scheme [21], Lu's scheme enjoys shorter initial private key while achieving direct chosen-ciphertext security. More recently, Lu and Li [17] proposed a generic construction of forward-secure identity-based encryption from identity-based binary tree encryption.

1.2 *Contributions*

Security against adaptive chosen-ciphertext attacks (CCA) (i.e. CCA security) is the de facto level of security required for the public-key encryption schemes used in practice. However, the previous two forward-secure public-key encryption schemes [7,16] merely satisfy the security against chosen-plaintext attacks (CPA) (i.e. CPA security). In this paper, we newly propose a forward-secure public-key encryption scheme from the identity-based encryption scheme by Gentry [9]. We prove that the proposed scheme is secure against adaptive CCA under the truncated decision q-augmented bilinear Diffie–Hellman exponent (q-ABDHE) assumption in the standard model.

Although our new scheme and Lu's forward-secure identity-based encryption scheme [15] are both based on the identity-based encryption scheme by Gentry [9], they are different in several aspects: (1) cryptographic workflow. As these two schemes belong to two different cryptographic primitives, they have entirely different cryptographic workflows. For example, in Lu's scheme each user's public key is his identity and the corresponding initial private key is generated by a trusted private key generator while in our new scheme, each user's public key and initial private key are generated by himself. (2) Encryption/decryption process. In Lu's scheme, messages are

encrypted in two different ways in the different time periods, and thus ciphertexts have to be decrypted in two different ways in the different time periods too. In our new scheme, due to some improvements made on the constructing technique, such fault is avoided effectively. (3) Initial private key. In Lu's scheme, each user has a short initial private key that consists of two elements while in our new scheme, each user has a long initial private key which consists of $2(m + 2)$ elements. In forward-secure identity-based encryption, each user's initial private key is generated by a trusted private key generator and may be sent to its holder online. Therefore, the initial private key should be generated as short as possible in order to lessen the computation and communication load of the private key generator. In order to achieve this goal, Lu's scheme has to generate the initial private keys with no connection to the system time periods. Different from Lu's scheme, our new scheme generates all private keys (including the initial private keys) according to the system time periods. Although the resulting initial private keys are longer than the ones in Lu's scheme, they never aggravate the computation and communication load of the cryptosystem as each user can pre-generate his initial private key in an offline mode. In addition, such modification enables our new scheme to avoid the fault existing in the encryption/decryption process of Lu's scheme. (4) Key-escrow problem. Compared with Lu's scheme, the biggest merit of our new scheme is that it does not suffer from the key-escrow problem. In Lu's scheme, if the private key generator becomes dishonest, it can impersonate any user using its knowledge of the user's initial private key. This is due to the key-escrow problem inherent in identity-based cryptography.

Additionally, compared with the previous two forward-secure public-key encryption schemes [7,16], our new scheme enjoys the following nice features: (1) the new scheme achieves direct chosen-ciphertext security in the standard model. Of course, as introduced in [7,16], both the previous schemes can be modified to achieve chosen-ciphertext security in the standard model by applying the techniques of Sahai [19] based on the non-interactive zero knowledge proof system. However, this will significantly increase both the computation and communication costs. (2) The new scheme has the constant ciphertext size and decryption time. In our new scheme, the ciphertext size and the decryption time have no correlation with the number of time periods and other performance indices have at most poly logarithmic complexities in terms of the number of time periods. The comparison shows that the performance of our scheme is almost as efficient as the scheme by Lu and Li [16] and outperforms the one by Canetti *et al.* [7].

2. Preliminaries

2.1 *Forward-secure public-key encryption*

Usually, a forward-secure public-key encryption scheme is composed of four algorithms: (1) key generation algorithm **KeyGen**, which is performed by the user to generate a public key and an initial private key; (2) key update algorithm **KeyUpdate**, which is performed by the user to generate a new private key for the next time period from the current one; (3) encryption algorithm **Encrypt**, which is performed by a sender to encrypt the messages in the current time period; (4) decryption algorithm **Decrypt**, which is performed by a receiver to decrypt the ciphertext sent to him in the current time period.

Figure 1 gives the functional description of a forward-secure public-key encryption scheme.

DEFINITION 1 *A forward-secure public-key encryption scheme* $\Pi =$ (**KeyGen**, **KeyUpdate**, **Encrypt**, **Decrypt**) *is said to be correct if for any message M,* **Decrypt**$(PK, SK_\tau,$ **Encrypt** $(\tau, PK, M)) = M$, *where PK is the public key obtained from the key generation algorithm* **KeyGen**, SK_τ *is the private key for the time period* τ *obtained from the key update algorithm* **KeyUpdate**.

KeyGen$(k, T) \rightarrow (PK, SK_0)$
Input: a security parameter $k \in Z^+$, the total number of time periods $T \in Z^+$
Output: a public key PK and an initial private key SK_0

KeyUpdate$(\tau, PK, SK_\tau) \rightarrow SK_{\tau+1}$
Input: a time period $\tau \in [0, T\text{-}1]$, a public key PK and a private key SK_τ for the time period τ
Output: a private key $SK_{\tau+1}$ for the next time period $\tau + 1$

Encrypt$(\tau, PK, M) \rightarrow C$
Input: a time period $\tau \in [0, T\text{-}1]$, a public key PK and a message M
Output: a ciphertext $<\tau, C>$ of the message M for the time period τ

Decrypt$(PK, SK_\tau, <\tau, C>) \rightarrow M$
Input: a public key PK, the private key SK_τ for the time period τ and a ciphertext $<\tau, C>$
Output: a message M or an error symbol $\perp$ if $<\tau, C>$ is an invalid ciphertext

Figure 1. Functional description of forward-secure public-key encryption.

As introduced in [7], chosen-ciphertext security for forward-secure public-key encryption schemes (*fs*-CCA2 security) is defined via the following two-stage adversarial game between an adversary $\mathcal{A}$ and a game simulator (or challenger):

fs-CCA2 adversarial game
1. $(\mathrm{PK}, \mathrm{SK}_0) \leftarrow \mathbf{KeyGen}(k, T)$
2. $(i^*, M_0, M_1, \mathrm{st}) \leftarrow \mathcal{A}^{O^{\text{Break-in}}(\cdot), O^{\text{Decrypt}}(\cdot)}(\mathrm{PK})$
3. $b \leftarrow \{0, 1\}$
4. $\langle i^*, C^* \rangle \leftarrow \mathbf{Encrypt}(i^*, \mathrm{PK}, M_b)$
5. $b' \leftarrow \mathcal{A}^{O^{\text{Decrypt}}(\cdot)}(\mathrm{PK}, \langle i^*, C^* \rangle, M_0, M_1, \mathrm{st})$

In the above game, st is some state information, $O^{\text{Break-in}}(\cdot)$ and $O^{\text{Decrypt}}(\cdot)$ are two oracles to which the adversary $\mathcal{A}$ has access. The break-in oracle $O^{\text{Break-in}}(\cdot)$, which models the private key disclosure attack carried out by the adversary $\mathcal{A}$, takes a time period i (satisfying $i > i^*$) as input and outputs a private key SK_i for the time period i. The adversary $\mathcal{A}$ can query such oracle only one time. The decryption oracle $O^{\text{Decrypt}}(\cdot)$ takes a ciphertext $\langle j, C \rangle$ as input and outputs the decryption of $\langle j, C \rangle$. Note that the adversary $\mathcal{A}$ is not allowed to query the oracle $O^{\text{Decrypt}}(\cdot)$ on the challenge ciphertext $\langle i^*, C^* \rangle$. The advantage of the adversary $\mathcal{A}$ in the above game is defined to be

$$\mathrm{Adv}_{\mathcal{A}}^{fs\text{-CCA2}}(k) = |\Pr[b = b'] - \tfrac{1}{2}|.$$

DEFINITION 2 *A forward-secure public-key encryption scheme is secure against CCA (fs-CCA2 secure) if the advantage* $\mathrm{Adv}_{\mathcal{A}}^{fs\text{-CCA2}}(k)$ *is negligible for any probabilistic polynomial-time adversary* $\mathcal{A}$.

Similarly, chosen-plaintext security for forward-secure public-key encryption schemes (*fs*-CPA security) can be defined by disallowing the adversary to make any queries to the decryption oracle $O^{\text{Decrypt}}(\cdot)$ in the adversarial game. In [16], Lu and Li introduced a weaker security model for forward-secure public-key encryption schemes, in which the adversary selects a target time period i^* at the very beginning of the adversarial game. We denote chosen-ciphertext security and chosen-plaintext security defined in such a security model by *fs*-ST-CCA2 (i.e. forward security against selective time period and adaptive CCA) and *fs*-ST-CPA (i.e. forward security against selective time period and CPA) respectively.

2.2 *Bilinear groups and truncated decision q-ABDHE assumption*

Our forward-secure public-key encryption scheme is constructed using prime order bilinear groups.

Let $\mathcal{G}$ be a bilinear group generator that takes a security parameter $k \in Z^+$ as input and outputs the description of the bilinear groups $\mathbb{G} = (G, G_T, p, g, e)$, where G and G_T are two multiplicative cyclic groups of prime order p, g is the generator of G, and e: $G \times G \to G_T$ is an admissible bilinear map that satisfies the following properties:

(1) Bilinear: We say that the map e is bilinear if $\forall\, h_1, h_2 \in G, a, b \in Z_p^*$, $e(h_1^a, h_2^b) = e(h_1, h_2)^{ab}$.
(2) Non-degenerate: We say that the map e is non-degenerate if $e(g, g) \neq 1_{G_T}$, where 1_{G_T} is the identity element of G_T.
(3) Computable: We say that the map e is computable if $\forall\, h_1, h_2 \in G$, there exists an efficient algorithm to compute $e(h_1, h_2)$.

Let $\mathbb{G} = (G, G_T, p, g, e)$ be the description of bilinear groups. The truncated decision q-ABDHE assumption [9] states that given $(\mathbb{G}, g'^{x^{q+2}}, g^x, \ldots, g^{x^q}) \in G^{q+2}$ it is hard to distinguish $e(g, g')^{x^{q+1}} \in G_T$ from a random element $X \in G_T$. Let $\mathcal{B}$ be an algorithm that takes $(\mathbb{G}, g', g'^{x^{q+2}}, g^x, \ldots, g^{x^q}, X)$ as input and outputs 1 if $X = e(g, g')^{x^{q+1}}$ and 0 otherwise. We define the advantage of the algorithm $\mathcal{B}$ to be

$$\mathrm{Adv}_{\mathcal{B}}^{q\text{-ABDHE}}(k) = \left| \begin{array}{c} \Pr[\mathcal{B}(\mathbb{G}, g', g'^{x^{q+2}}, g^x, \ldots, g^{x^q}, e(g, g')^{x^{q+1}}) = 1] \\ -\Pr[\mathcal{B}(\mathbb{G}, g', g'^{x^{q+2}}, g^x, \ldots, g^{x^q}, X) = 1] \end{array} \right|.$$

DEFINITION 3 *We say that the truncated decision q-ABDHE assumption holds in (G, G_T) if the advantage $\mathrm{Adv}_{\mathcal{B}}^{q\text{-ABDHE}}(k)$ is negligible for any probabilistic polynomial-time algorithm $\mathcal{B}$.*

2.3 *Collision-resistant hash function*

In our construction of forward-secure public-key encryption, we require a collision-resistant hash function.

DEFINITION 4 *We say that a hash function H is collision-resistant if the advantage $\mathrm{Adv}_{\mathcal{B}}^{\mathrm{CR}}(k) = \Pr[H(x) = H(y) \wedge x \neq y | (x, y) \leftarrow \mathcal{B}(k, H)]$ is negligible for any probabilistic polynomial-time algorithm $\mathcal{B}$.*

3. Description of the proposed scheme

Like the previous forward-secure public-key encryption schemes [7,16], we use a bintree structure to update private keys. In order to generate a forward-secure public-key encryption scheme with total time periods $T = 2^{m+1} - 2$, we use a full bintree with depth m. In our scheme, all time periods are associated with the non-root nodes of the bintree in a pre-order style. Each non-root node of the bintree is labelled with a binary string λ. We label the left child of the root node with 0 and the right one with 1, respectively. When a node is labelled with λ, then its two children are respectively labelled with $\lambda 0$ and $\lambda 1$ (from left to right). Let λ^τ be the node associated with the time period τ, the private key SK_τ for the time period τ is a set of node keys which contains the node keys of λ^τ and the right brothers of all the nodes on the route from λ^τ to the root node in the bintree. For simplicity of description, we represent SK_τ as a stack of node keys and put the node key of λ^τ on top of the stack.

The proposed forward-secure public-key encryption scheme consists of the following four algorithms:

KeyGen(k, T): To generate a pair of public key PK and initial private key SK_0, this algorithm performs the following steps:

(1) Run $\mathcal{G}(k)$ to generate $\mathbb{G} = (G, G_T, p, g, e)$.
(2) Choose $\alpha, \beta \in Z_p^*$ and $h_1 \in G$ at random, compute $h_2 = g^\alpha$.
(3) Choose a random m-length vector $\bar{V} = (v_1, \ldots, v_m) \in G^m$.
(4) Choose a collision-resistant hash function $H : G^2 \times G_T^2 \to Z_p^*$.
(5) Choose $s \in Z_p^*$ randomly, compute

$$n\mathrm{key}_0 = (\beta, (g^{-\beta} h_1)^{1/\alpha} \cdot (v_1^0)^s, h_2^s, v_2^s, \ldots, v_m^s),$$
$$n\mathrm{key}_1 = (\beta, (g^{-\beta} h_1)^{1/\alpha} \cdot (v_1^1)^s, h_2^s, v_2^s, \ldots, v_m^s).$$

(6) Output $\mathrm{PK} = (\mathbb{G}, h_1, h_2, \bar{V}, H)$ and $\mathrm{SK}_0 = (n\mathrm{key}_0, n\mathrm{key}_1)$.

KeyUpdate$(\tau, \mathrm{PK}, \mathrm{SK}_\tau)$: Let $\lambda^\tau = \lambda_1 \lambda_2 \ldots \lambda_d$ be the node associated with the time period τ and $n\mathrm{key}_{\lambda^\tau}$ be the node key associated with λ^τ. To generate the private key $\mathrm{SK}_{\tau+1}$, this algorithm performs the following steps:

(1) If λ^τ is a leaf node, pop $n\mathrm{key}_{\lambda^\tau}$ off the stack. Now, the node key on the top of the stack is $n\mathrm{key}_{\lambda^{\tau+1}}$. Then, set the remaining node keys in the stack as the private key $\mathrm{SK}_{\tau+1}$ for the time period $\tau + 1$.
(2) Else if λ^τ is an internal node, pop $n\mathrm{key}_{\lambda^\tau}$ off the stack, parse λ^τ as $\lambda_1 \lambda_2 \ldots \lambda_d$ and $n\mathrm{key}_{\lambda^\tau}$ as $(\beta, (g^{-\beta} h_1)^{1/\alpha} \cdot (\prod_{j=1}^d v_j^{\lambda_j})^s, h_2^s, v_{d+1}^s, \ldots, v_m^s)$, respectively, choose $s' \in Z_p^*$ randomly, compute $n\mathrm{key}_{\lambda_1 \ldots \lambda_d 1}$ and $n\mathrm{key}_{\lambda_1 \ldots \lambda_d 0}$ as

$$n\mathrm{key}_{\lambda_1 \ldots \lambda_d \lambda_{d+1}} = \left(\beta, (g^{-\beta} h_1)^{1/\alpha} \cdot \left(\prod_{j=1}^{d} v_j^{\lambda_j}\right)^s \cdot v_{d+1}^{s\lambda_{d+1}} \cdot \left(\prod_{j=1}^{d+1} v_j^{\lambda_j}\right)^{s'}, h_2^{s+s'}, v_{d+2}^{s+s'}, \ldots, v_m^{s+s'}\right)$$
$$= \left(\beta, (g^{-\beta} h_1)^{1/\alpha} \cdot \left(\prod_{j=1}^{d+1} v_j^{\lambda_j}\right)^{s''}, h_2^{s''}, v_{d+2}^{s''}, \ldots, v_m^{s''}\right),$$

where $s'' = s + s'$. Push $n\mathrm{key}_{\lambda_1 \ldots \lambda_d 1}$ and then $n\mathrm{key}_{\lambda_1 \ldots \lambda_d 0}$ onto the stack. Now, the node keys in the stack compose the private key $\mathrm{SK}_{\tau+1}$.
(3) Erase SK_τ and output $\mathrm{SK}_{\tau+1}$.

Encrypt(τ, PK, M): Let $\lambda^\tau = \lambda_1 \lambda_2 \ldots \lambda_d$ be the node associated with the time period τ. To encrypt a message M, this algorithm performs the following steps:

(1) Parse λ^τ as $\lambda_1 \lambda_2 \ldots \lambda_d$.
(2) Choose $r \in Z_p^*$ at random, compute

$$C = (C_1, C_2, C_3, C_4, C_5) = \left(h_2^r, \left(\prod_{j=1}^{d} v_j^{\lambda_j}\right)^r, e(g, g)^r, M \cdot e(g, h_1)^{-r}, (v_1 v_2^\gamma)^r\right),$$

where $\gamma = H(C_1, C_2, C_3, C_4)$.
(3) Output the ciphertext $\langle \tau, C \rangle$.

Decrypt(PK, SK_τ, $< \tau, C >$): Let $\lambda^\tau = \lambda_1\lambda_2 \ldots \lambda_d$ be the node associated with the time period τ and $n\text{key}_{\lambda^\tau}$ be the node key associated with λ^τ. To decrypt a ciphertext $< \tau, C >$, this algorithm performs the following steps:

(1) Parse $n\text{key}_{\lambda^\tau}$ as $(\beta, (g^{-\beta}h_1)^{1/\alpha} \cdot (\prod_{j=1}^{d} v_j^{\lambda_j})^s, h_2^s, v_{d+1}^s, \ldots, v_m^s)$.
(2) Check whether $e(C_1, v_1 v_2^\gamma)/e(h_2, C_5) = 1$, where $\gamma = H(C_1, C_2, C_3, C_4)$.
(3) If it does, compute and output the message

$$M = e\left(C_1, (g^{-\beta}h_1)^{1/\alpha} \cdot \left(\prod_{j=1}^{d} v_j^{\lambda_j}\right)^s\right) \cdot e(C_2, h_2^s)^{-1} \cdot C_3^\beta \cdot C_4.$$

Otherwise, output an error symbol $\perp$.

4. Analysis of the proposed scheme

4.1 *Correctness*

THEOREM 1 *The above forward-secure public-key encryption scheme is correct.*

Proof This theorem obviously holds as we have

$$\begin{aligned}
\frac{e(C_1, v_1 v_2^\gamma)}{e(h_2, C_5)} = \frac{e(h_2^r, v_1 v_2^\gamma)}{e(h_2, (v_1 v_2^\gamma)^r)} = 1, \quad & e\left(C_1, (g^{-\beta}h_1)^{1/\alpha} \cdot \left(\prod_{j=1}^{d} v_j^{\lambda_j}\right)^s\right) \cdot e(C_2, h_2^s)^{-1} \cdot C_3^\beta \cdot C_4 \\
&= e\left(h_2^r, (g^{-\beta}h)^{1/\alpha} \cdot \left(\prod_{j=1}^{d} v_j^{\omega_j}\right)^s\right) \cdot e\left(\left(\prod_{j=1}^{d} v_j^{\omega_j}\right)^r, h_2^s\right)^{-1} \\
&\quad \cdot (e(g,g)^r)^\beta \cdot M \cdot e(g,h)^{-r} \\
&= e(h_2^r, (g^{-\beta}h)^{1/\alpha}) \cdot e(g,g)^{r\beta} \cdot M \cdot e(g,h)^{-r} = M.
\end{aligned}$$

■

4.2 *Security*

THEOREM 2 *Assume that H is a collision-resistant hash function and the truncated decision q-ABDHE assumption holds in (G, G_T), then the above forward-secure public-key encryption scheme is fs-CCA2 secure in the standard model. More concretely, suppose that $\mathcal{A}$ is an fs-CCA2 adversary that makes at most q_D queries to the decryption oracle $O^{Decrypt}$, then there exists an algorithm $\mathcal{B}_1$ against the collision resistance of the hash function H that has advantage $\text{Adv}_{\mathcal{B}_1}^{\text{CR}}(k)$ and an algorithm $\mathcal{B}_2$ against the truncated decision q-ABDHE problem in (G, G_T) that has advantage $\text{Adv}_{\mathcal{B}_2}^{q\text{-ABDHE}}(k)$, such that the advantage of $\mathcal{A}$ is bounded by*

$$Adv_{\mathcal{A}}^{fs\text{-}CCA2}(k) \leq Adv_{\mathcal{B}_1}^{CR}(k) + Adv_{\mathcal{B}_2}^{q\text{-}ABDHE}(k),$$

where $q = (q_D + 1) \cdot m + 2$.

Proof To prove this theorem, we define a sequence of six games. All games involve the adversary $\mathcal{A}$ who attempts to guess the random bit b for which it eventually outputs a guess b'. For all $I \in$[1,6], we let E_i be the event that $b' = b$ in Game i.

Game 1. This game is defined to be the original adversarial game played by the adversary $\mathcal{A}$. Therefore, we have that $|\Pr[E_1] - \frac{1}{2}| = \mathrm{Adv}_{\mathcal{A}}^{fs-\mathrm{CCA2}}(k)$.

Game 2. This game is identical to Game 1 except that some values of the public key are replaced. In this game, the game simulator first randomly picks $a \in Z_p^*$ to set $h_2 = g^{\alpha}$. It then randomly chooses a q-degree function $f(x) \in Z_p[x]$ with $f(0) \neq 0$, sets $\beta = f(0)$ and computes $h_1 = g^{f(\alpha)}$. Note that $g^{f(\alpha)}$ can be computed from $(g, g^a, \ldots, g^{\alpha^q})$. It further randomly chooses $r_1, \ldots, r_m \in Z_p^*$ and defines a vector $= (v_1, \ldots, v_m) = (h_2^{r_1}, \ldots, h_2^{r_m})$. Clearly, the replaced values are distributed identically to the corresponding values in Game 1. Hence, we have $\Pr[E_1] = \Pr[E_2]$.

Game 3. This game is identical to Game 2 except that the game is stopped if the following event (denoted by E) happens: the adversary $\mathcal{A}$ submits $< j, C = (C_1, C_2, C_3, C_4, C_5) >$ to the oracle $O^{Decrypt}$ such that $(C_1, C_2, C_3, C_4) \neq (C_1^*, C_2^*, C_3^*, C_4^*)$ and $H(C_1, C_2, C_3, C_4) = H(C_1^*, C_2^*, C_3^*, C_4^*)$, where $(C_1^*, C_2^*, C_3^*, C_4^*)$ is the first four parts of the challenge ciphertext $C^* = (C_1^*, C_2^*, C_3^*, C_4^*, C_5^*)$. Since Game 1 and Game 2 proceed identically unless the event E occurs, we have $|\Pr[E_2] - \Pr[E_3]| = \Pr[E]$. As the event E implies a collision for H, there must exist an algorithm $\mathcal{B}_1$ against the collision resistance of H such that $\Pr[E] \leq \mathrm{Adv}_{\mathcal{B}_1}^{\mathrm{CR}}(k)$. Therefore, we get $|\Pr[E_2] - \Pr[E_3]| \leq \mathrm{Adv}_{\mathcal{B}_1}^{\mathrm{CR}}(k)$.

Game 4. This game is identical to Game 3 except that we change the way that the queries are answered. We define a node key generation algorithm **NodeKeyGen** that takes as input a node label λ and a public key PK, and outputs a node key $n\mathrm{key}_\lambda$ for the node λ.

NodeKeyGen(λ, PK)

Choose $s \in Z_p^*$ at random

Parse λ as $\lambda_1 \lambda_2 \ldots \lambda_d$

Compute $n\mathrm{key}_\lambda = (\beta, g^{(f(\alpha)-f(0))/\alpha} \cdot (\prod_{j=1}^{d} v_j^{\lambda_j})^s, h_2^s, v_{d+1}^s, \ldots v_m^s)$

Return $n\mathrm{key}_\lambda$

Note that $g^{(f(\alpha)-f(0))/\alpha}$ can be computed from $(g, g^a, \ldots, g^{\alpha^{q-1}})$. Because $g^{(f(\alpha)-f(0))/\alpha} \cdot (\prod_{j=1}^{d} v_j^{\lambda_j})^s = (g^{-\beta} h_1)^{1/\alpha} \cdot (\prod_{j=1}^{d} v_j^{\lambda_j})^s$, $n\mathrm{key}_\lambda$ is a valid node key for λ.

In this game, the game simulator responds the adversary $\mathcal{A}$'s queries as follows:

$O^{\text{Break-in}}(\tau)$: Let λ^τ be the node associated with the time period τ. Recall that a user's private key SK_τ for the time period τ is composed of the secret keys of λ^τ and the right brothers of all the nodes on the route from λ^τ to the root node in the bintree. Obviously, the game simulator can correctly answer such query by recursively executing the algorithm **NodeKeyGen** to generate the node keys contained in SK_τ.

$O^{\text{Decrypt}}(\langle j, C \rangle)$: Let λ^j be the node associated with the time period j. The game simulator firstly generates the private key SK_j for the time period $j$as above, and then decrypts the ciphertext $\langle j, C \rangle$ according to the specification of the algorithm **Decrypt**.

Since the game simulator can correctly answer the adversary $\mathcal{A}$' queries as in Game 3, we get $\Pr[E_3] = \Pr[E_4]$.

Game 5. This game is identical to Game 4 except that we change the generation of the challenge ciphertext. Let $\lambda^* = \lambda_1^* \lambda_2^* \ldots \lambda_n^*$ be the node associated with the target time period i^*. The game simulator first runs the algorithm **NodeKeyGen** to generate the node key $n\mathrm{key}_{\lambda^*} = (\beta, (g^{-\beta} h_1)^{1/\alpha} \cdot (\prod_{j=1}^{n} v_j^{\lambda_j^*})^{s^*}, h_2^{s^*}, v_{n+1}^{s^*}, \ldots, v_m^{s^*})$ for λ^*, where $s^* \in Z_p^*$. It then computes the challenge ciphertext $\langle i^*, C^* = (C_1^*, C_2^*, C_3^*, C_4^*, C_5^*) \rangle$ as

$$C_1^* = g'^{\alpha^{q+2}}, \quad C_2^* = \prod_{j=1}^{n} (g'^{\alpha^{q+2}})^{r_j \cdot \lambda_j^*}, \quad C_3^* = e(g, g')^{\alpha^{q+1}},$$

$$C_4^* = M_b \cdot e\left(C_1^*, (g^{-\beta}h_1)^{1/\alpha} \cdot \left(\prod_{j=1}^{n} v_j^{\lambda_j^*}\right)^{s^*}\right)^{-1} \cdot e(h_2^{s^*}, C_2^*) \cdot (C_3^*)^{-\beta}, \quad C_5^* = (g'^{\alpha^{q+2}})^{(r_1+r_2\gamma^*)},$$

where $\gamma^* = H(C_1^*, C_2^*, C_3^*, C_4^*)$. Let $r^* = \alpha^{q+1} \cdot \log_g g'$, then we have

$$C_1^* = g'^{\alpha^{q+2}} = (g^{\alpha})^{\alpha^{q+1}\cdot\log_g g'} = h_2^{r^*},$$

$$C_2^* = \prod_{j=1}^{n}(g'^{\alpha^{q+2}})^{r_j\cdot\lambda_j^*} = \prod_{j=1}^{n}(g^{\alpha\cdot r_j\cdot\lambda_j^*})^{\alpha^{q+1}\cdot\log_g g'} = \prod_{j=1}^{n}(v_j^{\lambda_j^*})^{r^*},$$

$$C_3^* = e(g, g')^{\alpha^{q+1}} = e(g, g)^{\alpha^{q+1}\cdot\log_g g'} = e(g, g)^{r^*},$$

$$C_4^* = M_b \cdot e\left(C_1^*, (g^{-\beta}h_1)^{1/\alpha} \cdot \left(\prod_{j=1}^{n} v_j^{\lambda_j^*}\right)^{s^*}\right)^{-1} \cdot e(h_2^{s^*}, C_2^*) \cdot (C_3^*)^{-\beta}$$

$$= M_b \cdot e\left(h_2^{r^*}, (g^{-\beta}h_1)^{1/\alpha} \cdot \left(\prod_{j=1}^{n} v_j^{\lambda_j^*}\right)^{s^*}\right)^{-1} \cdot e\left(h_2^{s^*}, \prod_{j=1}^{n}(v_j^{\lambda_j^*})^{r^*}\right) \cdot (e(g, g)^{r^*})^{-\beta}$$

$$= M_b \cdot e(g^{r^*}, g^{-\beta}h_1)^{-1} \cdot e\left(h_2^{r^*}, \left(\prod_{j=1}^{n} v_j^{\lambda_j^*}\right)^{s^*}\right)^{-1} \cdot e\left(h_2^{s^*}, \prod_{j=1}^{n}(v_j^{\lambda_j^*})^{r^*}\right) \cdot e(g, g)^{-r^*\beta}$$

$$= M_b \cdot e(g, h_1)^{-r^*},$$

$$C_5^* = (g'^{\alpha^{q+2}})^{(r_1+r_2\gamma^*)} = (g^{\alpha})^{(r_1+r_2\gamma^*)\cdot\alpha^{q+1}\cdot\log_g g'} = (h_2)^{(r_1+r_2\gamma^*)\cdot\alpha^{q+1}\cdot\log_g g'} = (v_1 v_2^{\gamma^*})^{r^*}.$$

Clearly, the challenge ciphertext $\langle i^*, C^*\rangle$ is a valid encryption of the message M_b. Therefore, we have $\Pr[E_4] = \Pr[E_5]$.

Game 6. In this game, the game simulator forgets the value α and simply retains $(\mathbb{G}, g', g'^{\alpha^{q+2}}, g^a, \ldots, g^{\alpha^q})$. The challenge ciphertext $\langle i^*, C^*\rangle$ is computed as in Game 5 but using a random element X from G_T to set $C_3^* = X$. The whole simulation only depends on a truncated decision q-ABDHE tuple $(\mathbb{G}, g', g'^{\alpha^{q+2}}, g^a, \ldots, g^{\alpha^q}, X)$ and the game simulator does not use the value α at all. Clearly, Game 6 and Game 5 are equal unless there is an algorithm $\mathcal{B}_2$ that distinguishes $e(g, g')^{\alpha^{q+1}}$ from X. Therefore, we have $|\Pr[E_5] - \Pr[E_6]| \leq \mathrm{Adv}_{\mathcal{B}_2}^{q-\mathrm{ABDHE}}(k)$. In addition, we have $\Pr[E_6] = 1/2$ as C_3^* is completely independent from the bit b.

From the above game-hopping steps, we have $\Pr[E_1] = \Pr[E_2]$, $|\Pr[E_2] - \Pr[E_3]| \leq \mathrm{Adv}_{\mathcal{B}_1}^{\mathrm{CR}}(k)$, $\Pr[E_3] = \Pr[E_4] = \Pr[E_5]$, $|\Pr[E_5] - \Pr[E_6]| \leq \mathrm{Adv}_{\mathcal{B}_2}^{q\text{-ABDHE}}(k)$ and $\Pr[E_6] = 1/2$. Because $\mathrm{Adv}_{\mathcal{A}}^{fs\text{-CCA2}}(k) = |\Pr[E_1] - (1/2)| \leq |\Pr[E_1] - \Pr[E_2]| + |\Pr[E_2] - \Pr[E_3]| + |\Pr[E_3] - \Pr[E_4]| + |\Pr[E_4] - \Pr[E_5]| + |\Pr[E_5] - \Pr[E_6]| + |\Pr[E_6] - (1/2)|$, we get

$$\mathrm{Adv}_{\mathcal{A}}^{fs\text{-CCA2}}(k) \leq \mathrm{Adv}_{\mathcal{B}_1}^{\mathrm{CR}}(k) + \mathrm{Adv}_{\mathcal{B}_2}^{q\text{-ABDHE}}(k).$$

■

4.3 *Performance*

In Table 1, we make a comparison of our scheme with the previous forward-secure public-key encryption schemes [7,16]. We compare the complexity of the performance parameters of these

Table 1. Comparison of the forward-secure public-key encryption schemes.

Schemes		Scheme in [7]	Scheme in [16]	Our scheme
Standard model?		Yes	Yes	Yes
Security		*fs*-CPA	*fs*-ST-CPA	*fs*-CCA2
Computational complexity	Key update time	$O(\log T)$	$O(\log T)$	$O(\log T)$
	Encryption time	$O(\log T\ (\log\log T)^2)$	$O(\log T)$	$O(\log T)$
	Decryption time	$O(\log T)$	$O(1)$	$O(1)$
Communicational complexity	Public key size	$O(\log T)$	$O(\log T)$	$O(\log T)$
	Ciphertext size	$O(\log T)$	$O(1)$	$O(1)$

schemes in terms of the total number of time periods T. The performance parameters include key update time, encryption time, decryption time, public key size and ciphertext size.

We briefly analyse the complexity of the performance parameters of our new scheme. Both the key update algorithm and the encryption algorithm need to compute $O(\log T)$ exponentiations in G, therefore, they need $O(\log T)$ times. The decryption algorithm requires computing four pairings, two exponentiations in G_T and two exponentiations in G, therefore, it needs $O(1)$ times. The public key includes $m + 3$ group elements. Therefore the size of the public key is $O(\log T)$ bits. The ciphertext includes only five group elements and therefore has size $O(1)$ bits.

From Table 1, we can see that the performance of our new scheme is almost as efficient as the scheme by Lu and Li [16] and outperforms the one by Canetti *et al.* [7]. In addition and most importantly, our scheme achieves the stronger chosen-ciphertext security.

5. Conclusions

We have presented the first forward-secure public-key encryption scheme that achieves direct chosen-ciphertext security. We have proved in the standard model that the proposed scheme is *fs*-CCA2 secure under the truncated decision q-ABDHE assumption. In the proposed scheme, the ciphertext size and the decryption time have no correlation with the number of time periods and other performance indices have at most poly logarithmic complexities in terms of the number of time periods.

Like the previous forward-secure public-key encryption schemes, the total number of time periods in our scheme is bounded and known at the time of key generation. However, in some environments where the number of time periods is very large, a scheme with bounded time periods may be inefficient as its performance depends on the number of time periods. Therefore, the construction of forward-secure public-key encryption that can support unbounded number of time periods becomes an important and worthwhile effort.

Acknowledgements

We would like to thank the anonymous referees for their helpful comments. This work is supported by the National Natural Science Foundation of China (grant number 61272542).

References

[1] M. Abdalla and L. Reyzin, *A New Forward-Secure Digital Signature Scheme*, Advances in Cryptology – Asiacrypt 2000, Kyoto, Japan, 2000.

[2] M. Abdalla, S.K. Miner, and C. Namprempre, *Forward-Secure Threshold Signature Schemes*, The Cryptographer's Track at RSA Conference 2001, San Francisco, CA, 2001.

[3] R. Anderson, *Two remarks on public key cryptology*, Invited Lecture at the 4th ACM Conference on Computer and Communications Security, Zurich, Switzerland, 1997. Available at http://www.cl.cam.ac.uk/ftp/users/rja14/forwardsecure.pdf
[4] M. Bellare and S.K. Miner, *A Forward-Secure Digital Signature Scheme*, Advances in Cryptology – Crypto 1999, Santa Barbara, CA, 1999.
[5] M. Bellare and B. Yee, *Forward Security in Private-Key Cryptography*, The Cryptographer's Track at RSA Conference 2003, San Francisco, CA, 2003.
[6] D. Boneh, X. Boyen, and E.J. Goh, *Hierarchical Identity Based Encryption with Constant Size Ciphertext*, Advances in Cryptology – Eurocrypt 2005, Aarhus, Denmark, 2005.
[7] R. Canetti, S. Halevi, and J. Katz, *A Forward-Secure Public-Key Encryption Scheme*, Advances in Cryptology – Eurocrypt 2003, Warsaw, Poland, 2003.
[8] W. Diffie, P.C. Van-Oorschot, and M.J. Weiner, *Authentication and authenticated key exchanges*, Des. Codes Cryptogr. 2 (1992), pp. 107–125.
[9] C. Gentry, *Practical Identity-Based Encryption without Random Oracles*, Advances in Cryptology – Eurocrypt 2006, Saint Petersburg, Russia, 2006.
[10] C. Gentry and A. Silverberg, *Hierarchical ID-Based Cryptography*, Advances in Cryptology – Asiacrypt 2002, Queenstown, New Zealand, 2002.
[11] C.G. Günther, *An Identity-Based Key-Exchange Protocol*, Advances in Cryptology – Eurocrypt 1989, Houthalen, Belgium, 1989.
[12] G. Itkis and L. Reyzin, *Forward-Secure Signatures with Optimal Signing and Verifying*, Advances in Cryptology – Crypto 2001, Santa Barbara, CA, 2001.
[13] A. Kozlov and L. Reyzin, *Forward-Secure Signatures with Fast Key Update*, 3rd Conference on Security in Communication Networks, Amalfi, Italy, 2002.
[14] H. Krawczyk, *Simple forward-secure signatures from any signature scheme*, 7th ACM Conference on Computer and Communications Security, Athens, Greece, 2000.
[15] Y. Lu, *Efficient forward-secure identity-based encryption scheme in the standard model*, 2nd International Conference on Theoretical and Mathematical Foundations of Computer Science, Singapore, 2011.
[16] Y. Lu and J. Li, *A practical forward-secure public-key encryption scheme*, J. Netw. 6 (2011), pp. 1254–1261.
[17] Y. Lu and J. Li, *Generic construction of forward-secure identity-based encryption*, J. Comput. 7 (2012), pp. 3068–3074.
[18] T. Malkin, D. Micciancio, and S.K. Miner, *Efficient Generic Forward-Secure Signatures with an Unbounded Number of Time Periods*, Advances in Cryptology – Eurocrypt 2002, Amsterdam, The Netherlands, 2002.
[19] A. Sahai, *Non-malleable non-interactive zero-knowledge and adaptive chosen-ciphertext security*, 40th IEEE Symposium on Foundations of Computer Science, New York, NY, 1999.
[20] D. Yao, N. Fazio, Y. Dodis, and A. Lysyanskaya, *ID-based encryption for complex hierarchies with applications to forward security and broadcast encryption*, 11th ACM Conference on Computer and Communications Security, Washington, DC, 2004.
[21] J. Yu, F. Kong, X. Cheng, R. Hao, and J. Fan, *Forward-secure identity-based public-key encryption without random oracles*, Fundam. Inform. 111 (2011), pp. 241–256.