Forward Secure Identity-Based Signature Scheme with RSA

Hankyung Ko, Gweonho Jeong, Jongho Kim, Jihye Kim, Hyunok Oh

Forward Secure Identity-based Signature Scheme with RSA

Hankyung Ko[1], Gweonho Jeong[1], Jongho Kim[2], Jihye Kim[2] and Hyunok Oh[1]

[1] Hanyang University, Seoul, Korea,
{hankyungko,jkho1229,hoh}@hanyang.ac.kr
[2] Kookmin University, Seoul, Korea,
{bi04208, jihyek}@kookmin.ac.kr

Abstract. A forward secure identity based signature scheme (FSIBS) provides forward secrecy of secret keys. In order to mitigate the damage when keys are leaked, it is desirable to evolve all the secret keys, i.e., *both* the user keys and the master key. In this paper, we propose a new RSA-based FSIBS scheme which requires constant size keys and generates constant size signatures. The experimental results show that it takes 3ms to generate a signature in the proposed scheme while it takes 75ms in the existing pairing based approach. The proposed scheme is provably secure under the factoring assumption in the random oracle model.

Keywords: Forward security · Digital signature · ID based · Private key generator · RSA.

1 Introduction

The identity based signatures, so called IBS, are digital signatures where an identifier, such as e-mail address etc., is used as a public key. In the IBS system, a private key generator(PKG) publishes a public parameter and issues a secret signing key to an identified user using its master secret key. Because the IBS system utilizes publicly known information such as an identifier as its verification key, it allows any party to verify signatures without the explicit authenticated procedure of the verification key. On the other hand, the forward secure signature is a way to mitigate the damage caused by key exposure. A forward secure signature scheme divides the lifetime into several time periods, and uses a different key at each time period. With this idea, even if a signature key is exposed at a specific point of time, all signatures which are generated before the exposure time can be kept valid. In order to have advantages of both the IBS system and the forward secure signature system, several researches [6, 12, 17–20] have been conducted to add forward security to the signing key issued by a private key generator(PKG). However, forward secrecy in such forward secure ID based signature schemes is limited and incomplete because only the user private keys evolve each time period; Once the master secret key is compromised, all signatures made by any user under this system cannot be considered valid even if the signing keys evolve properly. Mainly, these attacks can be done by the powers, buying off the system managers. In order to minimize the risk from the leakage of the master secret key, the PKG master

key as well as the user's signing key also needs to satisfy forward security. This paper simply notates FSIBS *with* forward-secure PKG as FSIBS hereafter.

The notion of FSIBS was considered first in [15] and the scheme was based on an elliptic curve pairing function. The proposed scheme in [15] satisfies the forward security of both the master secret key and the users' signing keys. That is, even if a master secret key of time period t is exposed, all signatures generated before t are not invalidated. Although the scheme in [15] efficiently generates a signature, its application is limited due to the non-constant size signing keys. Considering the forward secrecy is widely applied including the IoT environment [9], providing more options for FSIBS is desirable. Given the only pairing-based scheme, it may be difficult to cover the resource-constrained IoT devices due to its complex pairing operation.

In this paper, we propose a new forward secure identity based signature scheme (FSIBS) in the RSA setting. Our scheme is constructed by extending the forward secure RSA based signature scheme in [1] into an identity-based scheme. The proposed scheme also allows the master key update for the forward secrecy of the master key. To update the keys, the scheme does not require any interaction between the user and the PKG, once an initial signing key is delivered. The proposed scheme is secure under the factoring assumption in the random oracle model.

Table 1: Comparison of our scheme and [15]

		Our scheme	[15]
Size	Master Secret Key	$O(1)$	$O(\log^2 T)$
	User Signing Key	$O(1)$	$O(\log^2 T)$
	Verification Key	$O(1)$	$\log T + 2l + 5$
	Signature	$O(1)$	$O(1)$
Computation	PKGKeyGen	$O(lTk^2)$	$O((a+e)\cdot \log^2 T + p)$
	KeyIssue	$O(lTk^2)$ (opt. $O(k^3)$)	$O(a \cdot l \log T + (a+e)\cdot \log^2 T)$
	MSKUpdate	$O(lk^2)$	$O((a+e)\cdot \log^2 T)$
	UKUpdate	$O(lk^2)$	$O((a+e)\cdot \log^2 T)$
	Sign	$O(lTk^2)$ (opt. $O(k^3)$)	$O((\log T + 2l)\cdot a + 3\cdot e)$
	Verify	$O(lTk^2)$	$O(5p + (\log T + 2l)\cdot a)$

Table 1 summarizes the comparison between the proposed scheme and [15]. In Table 1, T is the maximum number of periods, l is the bit length of the hash output, k is the bit length in RSA, a is multiplication time in bilinear group, p is pairing time, and e is exponentiation time in bilinear group.As shown in Table 1, in the proposed scheme, the sizes of all keys such as a master secret key, a user signing key and a verification key, and a signature are constant while they are not in [15]. The costs of KeyIssue, Update, and Sign algorithms in the optimized proposal are independent of the maximum number of period T. According to our experiment, overall, our (optimized) scheme is faster than [15], except in the Verify algorithm. In particular, for $T = 2^{15}$, generating a signature requests only 3ms with 2048-bit k of RSA, while the scheme in [15] requires 75ms with the 224-bit ECC key.

This paper is organized as follows. We begin by discussing the related works in Section 2. In Section 3, we define the notion of forward-secure ID based signature scheme with forward-secure key generation, the background assumption that our scheme is based on, and the formal security model. We construct our signature scheme in Section 4. After that, we describe the security proof in Section 5. In Section 6, we extend our proposed scheme to improve the signing performance. Moreover, experimental results are shown in Section 7. Finally, we conclude in Section 8.

2 Related Work

The concept of forward security was first proposed by Anderson [2], and Bellare and Miner [3] made the formal definition of forward secure digital signatures(FSS). [3] proposed two forward secure signature constructions. One of them is a generic construction which can be derived by any signature scheme, of which complexity is at least $O(\log T)$-factor times of the original scheme, where T is the number of total time periods. The other is a variation on the Fiat-Shamir signature scheme [7] for a forward secure version that has constant-size signatures and takes $O(T)$-time for signing and verification. Next, Abdalla and Reyzin proposed a FSS scheme with a short public key [1]. However, its key generation, signing and verification are slow. Itkis and Reyzin suggested a FSS scheme with efficient signing and verification based on Guillous-Quisquater signatures [8], however, it costs more update time. On the other hands, Kozlov and Reyzin suggested a FSS scheme with fast update, but its signing and verification takes longer. So far, time complexity of some algorithm components for any suggested FSS constructions is depends on the total number of period T.

Krawczyk proposed a generic FSS construction with a constant-size private key, but with $O(T)$-size storage (possibly non-private) [11]. An efficient generic construction with unlimited time periods [13] is suggested by Malkin, Micciancio, and Miner using Merkletrees [14].

Boyen et al. [4] proposed the concept of forward secure signatures with untrusted updates for additional protection of the private key. Its private key is encrypted by the second factor(i.e., user password), and the key update proceeds with encryption. They construct an efficient scheme with constant-size signatures and provide the security reduction based on the Bilinear Diffie-Hellman Inversion assumption (BDHI) without random oracles.

Liu et al. proposed the forward-secure ID-based signature scheme without providing specific security definition and proof [12]. Yu et al [18] formalized the security definition and proposed a scheme with a formal security proof. Meanwhile, Ebri et al. [6] proposed an efficient generic construction of forward-secure ID-based signature. When it is instantiated with Schnorr signatures [16], their construction provides an efficient scheme in the random oracle model. Yu et al. [19] made an extension of forward-secure ID-based signature with untrusted updates. Zhang et al. [20] proposed the two forms of lattice-based constructions under lattice assumption, in the random oracle model and in the standard model. Presently, Wei et al. [17] proposed an efficient revocation forward-secure ID-based signature to provide forward security and backward security upon key exposure. However, user key update requires interaction with PKG for each period to

support revocation. The update algorithm run by the PKG generates an updated private key for each unrevoked user at each period. However, the master secret key of PKG is not updated.

All of the above [6, 12, 17–20] consider only the security of the user's private keys. Oh et al. [15] proposed first and only FSIBS system with forward secure PKG in a bilinear group setting. The scheme is constructed as a three-dimensional hierarchical ID based signature scheme; the first dimension is related with periods, the second dimension with identifiers, and the third dimension with messages. The hierarchical ID based signatures are taken in a double manner; first for issuing users' signing keys, and second for signing. To implement [15], no extra interaction between PKG and users is required, if the signing key is issued properly.

3 Preliminaries

We define the model for a FSIBS scheme with forward secure private key generator as in [15] in this section. And we define the security model of our scheme and describe its underlying assumption.

3.1 FSIBS scheme

A FSIBS scheme with forward secure private key generator consists of six algorithms Σ_{FSIBS}=(PKGKeyGen, MSKUpdate, KeyIssue, UKUpdate, Sign, Verify) as following:

- PKGKeyGen(k, T): This algorithm takes security parameter k and the maximum number of time periods T, and outputs the public verification key VK and the PKG master key MSK_t. The PGK master key MSK_t includes the current time period t which is initialized to 1.
- MSKUpdate(MSK_t): This algorithm takes the PKG master key at current time period t as an input, and outputs a new PKG master key MSK_{t+1}. Once a new PKG master key is computed, the previous PKG master key is removed. If $t + 1 \geq T$, then the old key is removed and no new keys are created.
- KeyIssue(MSK_t, ID): This algorithm takes the PKG master key at current time period t and an identifier ID. It outputs a secret signing key $SK_{t,ID}$ for the specific identifier ID. The signing key also includes the time period t.
- UKUpdate$(SK_{t,ID})$: This algorithm takes in the signing key of an identifier ID at current time period t and outputs a new signing key $SK_{t+1,ID}$. Once a new signing key is created, the previous signing key is removed. If $t + 1 \geq T$, then the old key is removed and no new keys are created.
- Sign$(SK_{t,ID}, M)$: This algorithm takes in the signing key of an identifier ID at current time period t, and a message M. It outputs a signature σ for time period t.

– Verify(ID, σ, M, VK, t): This algorithm takes an identifier ID, a signature σ, a message M, a verification key VK, and a time period t. It outputs either valid or invalid.

Definition 1. *A FSIBS scheme is perfectly correct if*

$$\Pr\left[\begin{array}{l}\mathsf{Verify}(ID, \mathsf{Sign}(SK_{t,ID}, M), M, VK, t) = 1| \\ SK_{i,ID} \leftarrow \mathsf{KeyIssue}(MSK_i, ID) \vee \mathsf{UKUpdate}(SK_{i-1,ID}) \\ \wedge\, (MSK_1, VK) \leftarrow \mathsf{PKGKeyGen}(k, T) \\ \wedge\, MSK_j \leftarrow \mathsf{MSKUpdate}^j(MSK_1)\end{array}\right] = 1$$

3.2 Security model

We use a game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ to define the security. The game captures the notion of PKG master key forward security, users' signing key forward security and the traditional unforgeability with security parameter k, maximum time period T, and negligible function ϵ. The game proceeds as following :

[Setup phase] The time period t is set to 1. Challenger $\mathcal{C}$ generates verification key VK and master key MSK_1 through PKGKeyGen, and give VK to adversary $\mathcal{A}$.

[Interactive query phase] In this phase the adversary $\mathcal{A}$ is allowed to adaptively query the following oracles.

KeyIssue : The adversary $\mathcal{A}$ can choose a specific ID and ask the challenger $\mathcal{C}$ for the signing key for ID of current time period t. The challenger $\mathcal{C}$ returns the signing key $SK_{t,ID}$ for ID of current time period t to adversary $\mathcal{A}$.

Update : The adversary $\mathcal{A}$ can request the challenger $\mathcal{C}$ to execute the MSKUpdate algorithm and UKUpdate algorithm.

Sign : The adversary $\mathcal{A}$ can choose a random message M and a user identifier ID and ask the challenger to sign a message M for ID on the current time period t. Then, it receives a signature σ from the challenger.

Hash: The adversary $\mathcal{A}$ can make queries to the random oracle and get the corresponding random values.

[PKG corrupt phase] The adversary $\mathcal{A}$ requests the PKG master key of the current time period t' from the challenger $\mathcal{C}$. The challenger $\mathcal{C}$ returns the master key $MSK_{t'}$. Note that the signing key of any identifier at the current time period and after can be generated using $MSK_{t'}$.

[Final forgery phase] The adversary $\mathcal{A}$ produces a fake signature which consists of a time period, user's ID, message and signature tuple (t^*, ID^*, M^*, σ^*). The adversary is successful if $t^* < t'$, the signature is valid for time t^*, and the adversary had not queried for a signature on M^* and ID^* at the time period t^*.

We define the advantage of an algorithm $\mathcal{A}$ attacking scheme Σ by the probability that $\mathcal{A}$ is successful in the game, and denote the advantage by $Adv^{\mathsf{FSIBS}}_{\mathcal{A},\Sigma}$. We say that a PKG forward secure scheme Σ is $(\tau, q_{key}, q_{sign}, q_{hash}, \epsilon)$-secure if for all adversaries

$\mathcal{A}$ running in time τ, making q_{key} key issue query, q_{sign} sign query and q_{hash} hash query, if:

$$Adv^{\mathsf{FSIBS}}_{\mathcal{A},\Sigma}(k,T) \leq \epsilon(k).$$

3.3 Factoring assumption

Let $\mathcal{A}$ be an adversary for the problem of factoring Blum integers; an integer $n = pq$, where p, q are distinct odd primes and $p \equiv q \equiv 3 \pmod 4$, is called a Blum integer. We define the following experiment.

Experiment Factoring$_{\mathcal{A}}(k)$

Randomly choose two k/2 bit primes p and q, s.t. :
$p \equiv q \equiv 3 \pmod 4$
$N \leftarrow pq$
$(p', q') \leftarrow \mathcal{A}(N)$
If $p'q' = N$ and $p' \neq 1$ and $q' \neq 1$ then return 1 else return 0

Let $Adv^{FAC}_{\mathcal{A}}(k)$ denote the probability that experiment Factoring$_{\mathcal{A}}(k)$ returns 1. We say that the factoring assumption holds if for all PPT algorithm $\mathcal{A}$ with negligible function ϵ,

$$Adv^{FAC}_{\mathcal{A}}(k) \leq \epsilon(k).$$

3.4 Multiple forking

Multiple forking (MF) is an extension of general forking to accommodate nested oracle replay attacks [5]. The modularity of the MF Lemma allows one to abstract out the probabilistic analysis of the rewinding process from the actual simulation in the security argument. The MF algorithm $\mathcal{M}_{\mathcal{Y},n}(x)$ associated to $\mathcal{Y}$ and n is defined in [5].

Lemma 1. *Let $\mathcal{G}$ be a randomized algorithm that takes no input and returns a string. Let $\mathcal{Y}$ be a randomized algorithm that on input a string x and elements $s_1, ..., s_q \in \mathbb{S}$ returns a triple (I, J, σ) consisting of two integers $0 \leq J < I \leq q$ and a string σ. Let*

$$mfrk := Pr[(b = 1) \mid x \xleftarrow{\$} \mathcal{G}; (b, \{\sigma_0, ..., \sigma_n\}) \xleftarrow{\$} \mathcal{M}_{\mathcal{Y},n}(x)] \; and$$
$$acc := Pr[(I \geq 1) \wedge (J \geq 1)] \mid x \xleftarrow{\$} \mathcal{G}; \{s_1, ..., s_q\} \xleftarrow{U} \mathbb{S}; (I, J, \sigma) \xleftarrow{\$} \mathcal{Y}(x, s_1, ..., s_q)]$$

then

$$mfrk \geq acc \cdot \left(\frac{acc^n}{q^{2n}} - \frac{(n+1)(n+3)}{8|\mathbb{S}|} \right),$$

where q is the sum of the upper bound on the queries to the random oracles involved and n is, loosely speaking, the number of forking.

4 The Proposed Scheme

In this section, we construct a forward-secure ID based digital signature scheme with forward-secure key generation, based on the factoring assumption. Our construction is based on a forward secure signature scheme in [1] and extends it into the identity based schemes. The proposed scheme can be considered as a recursive signature scheme; The first signature is generated with the master key to issue the user's signing key and the second signature is performed with the signing key to sign messages. This doubly recursive design should be carefully handled so that the update of the master key is correctly and independently synchronized with the update of the user's signing key. Finally, verifying the final signature must validate the authenticity of identifier, message, and the period. Given a maximum time period T, a security parameter k, and a maximum hash length l, The detailed construction of our FSIBS scheme is described in Algorithm 1.

Algorithm 1 Factoring based FSIBS

PKGKeyGen(k, T, l):
Generated random primes p,q s.t:
$p \equiv q \equiv 3 \pmod 4$
$2^{k-1} \leq (p-1)(q-1)$
$pq < 2^k$
$N \leftarrow pq$
$S \xleftarrow{\$} \mathbb{Z}_N^*$
$msk_1 \leftarrow S^{2^{3l}} \bmod N$
$U \leftarrow 1/S^{2^{3l(T+1)}} \bmod N$
$MSK_1 \leftarrow (N, T, 1, msk_1)$
$VK \leftarrow (N, U, T)$
return (MSK_1, VK)

KeyIssue(MSK_i, ID):
parse MSK_i as (N, T, i, msk_i)
$R \xleftarrow{\$} \mathbb{Z}_N^*$
$Y \leftarrow R^{2^{3l(T+1-i)}} \bmod N$
$sk_{i,ID} \leftarrow R \cdot (msk_i)^{H_1(Y \| ID)} \bmod N$
$SK_{i,ID} \leftarrow (N, T, Y, sk_{i,ID})$
return $SK_{i,ID}$

MSKUpdate(MSK_j):
parse MSK_j as (N, T, j, msk_j)
if $j = T$ then
$MSK_{j+1} \leftarrow \perp$
else
$msk_{j+1} \leftarrow (msk_j)^{2^{3l}} \bmod N$
$MSK_{j+1} \leftarrow (N, T, j+1, msk_{j+1})$
return MSK_{j+1}

UKUpdate$(SK_{j,ID})$:
parse $SK_{j,ID}$ as $(N, T, Y, sk_{j,ID})$
if $j = T$ then
$SK_{j+1,ID} \leftarrow \perp$
else
$sk_{j+1,ID} \leftarrow (sk_{j,ID})^{2^{3l}} \bmod N$
$SK_{j+1,ID} \leftarrow (N, T, Y, sk_{j+1,ID})$
return $SK_{j+1,ID}$

Sign$(SK_{j,ID}, M, j)$:
parse $SK_{j,ID}$ as $(N, T, Y, sk_{j,ID})$
$R' \xleftarrow{\$} \mathbb{Z}_N^*$
$Y' \leftarrow (R')^{2^{3l(T+1-j)}} \bmod N$
$\sigma_j \leftarrow R' \cdot (sk_{j,ID})^{H_2(Y \| Y' \| j \| M)} \bmod N$
return (σ_j, Y', Y)

Verify$(ID, M, (\sigma_i, Y', Y), VK, i)$:
parse VK as (N, U, T)
$h_1 \leftarrow H_1(Y \parallel ID)$
$h_2 \leftarrow H_2(Y \parallel Y' \parallel i \parallel M)$
if $\sigma_i^{2^{3l(T+1-i)}} \cdot U^{h_1 h_2} = Y' \cdot Y^{h_2} \pmod N$
return 1
else
return 0

For the following correctness description, we assume that the user signing key is generated as $SK_{j,ID} = (N, T, Y, sk_{j,ID} = R \cdot (msk_j)^{h_1})$ for some time period j where $j < T$. Then, notice that the user signing key at time period i for $i > j$ evolves into $SK_{i,ID} = (N, T, Y, sk_{i,ID} = (R \cdot (msk_j)^{H_1(Y \| ID)})^{2^{3l(i-j)}} \bmod N)$ by the UKUpdate algorithm.

Correctness : We show that our scheme is correct by computing the equation below:

$$\begin{aligned}
\sigma_i^{2^{3l(T+1-i)}} &= (R' \cdot (sk_{i,ID})^{h_2})^{2^{3l(T+1-i)}} \\
&= R'^{2^{3l(T+1-i)}} (R^{2^{3l(i-j)}} (msk_j^{2^{3l(i-j)}})^{h_1})^{h_2 2^{3l(T+1-i)}} \\
&= R'^{2^{3l(T+1-i)}} (R^{2^{3l(T+1-j)}} (msk_j^{2^{3l(T+1-j)}})^{h_1})^{h_2} \\
&= R'^{2^{3l(T+1-i)}} (R^{2^{3l(T+1-j)}} (S^{2^{3lj}})^{h_1 2^{3l(T+1-j)}})^{h_2} \\
&= R'^{2^{3l(T+1-i)}} (R^{2^{3l(T+1-j)}})^{h_2} (S^{2^{3l(T+1)}})^{h_1 h_2} \\
&= Y'(Y)^{h_2} (1/U)^{h_1 h_2}
\end{aligned}$$

where $h_1 = H_1(Y \parallel ID)$, $h_2 = H_2(Y \parallel Y' \parallel i \parallel M)$.

5 Security Proof

Let k and l be two security parameters. Let p and q be primes and $N = pq$ be a k-bit integer (Since $p \equiv q \equiv 3 \pmod 4$, N is a Blum integer). Let Q denote the set of non-zero quadratic residues modulo N. Note that for $x \in Q$, exactly one of its four square roots is also in Q.

Lemma 2. *Given $\alpha \neq 0, \lambda > 0, v \in Q$ and $X \in \mathbb{Z}_N^*$ such that $v^\alpha \equiv X^{2^\lambda} \pmod N$ and $\alpha < 2^\lambda$, one can easily compute y such that $v \equiv y^2 \pmod N$.*

Proof. Let $\alpha = 2^\gamma \beta$ where β is odd. Note that $\lambda > \gamma$. Let $\beta = 2\delta + 1$. Then $(v^{2\delta+1})^{2^\gamma} \equiv v^\alpha \equiv X^{2^\lambda} \pmod N$, so $v^{2\delta+1} \equiv X^{2^{\lambda-\gamma}} \pmod N$. Note that it is allowed to take roots of degree 2^γ since both sides are in Q. Let $y = X^{2^{\lambda-\gamma-1}} / v^\delta \bmod N$. Then $y^2 \equiv X^{2^{\lambda-\gamma}} / v^{2\delta} \equiv v \pmod N$. Note that since $\alpha < 2^\lambda, \lambda - \gamma - 1 \geq 0$.

Theorem 1. *Let $\mathcal{A}$ be an adversary attacking Σ_{FSIBS} with running in time τ, making q_{key} number of key issue queries, q_{sign} number of sign queries and q_{hash} number of hash queries. If the advantage $Adv^{\mathsf{FSIBS}}_{\mathcal{A},\Sigma}$ is ϵ, then there exists an algorithm $\mathcal{B}$ that succeeds in solving the factoring problem in expected time at most τ' with probability at least ϵ', where $\tau' = 4\tau + O(k^2 lT + k^3)$ and $\epsilon' = \frac{\epsilon}{2T} \cdot \left(\frac{\epsilon^3}{T^3 \cdot q^6} - \frac{28}{2^{k+3}} \right)$ with $q = q_{hash} + q_{sign} + q_{key}$.*

Proof. Given $\mathcal{A}$, we show how to construct an algorithm $\mathcal{B}$ of solving the factoring problem with success probability $\frac{\epsilon}{2T} \cdot \left(\frac{\epsilon^3}{T^3 \cdot q^6} - \frac{28}{2^{k+3}} \right)$ approximately in time $4\tau + O(k^2 lT + k^3)$. To factor its input N, $\mathcal{B}$ will select a random $x \in \mathbb{Z}_N^*$, compute $v \equiv x^2 \bmod N$, and attempt to use the adversary $\mathcal{A}$ to find a square root y of v. Because v has four square roots and x is random, with probability $1/2$ we have that $x \not\equiv \pm y \pmod N$

and, then $\mathcal{B}$ will be able to find a factor of N by computing the gcd of $x - y$ and N. To do so, we utilize the forking lemma technique in the following. Before start of the game, $\mathcal{B}$ guesses the time period b', $1 < b' \leq T$ that $\mathcal{A}$ will make a forgery.

[Setup phase] $\mathcal{B}$ selects a random value $x \in \mathbb{Z}_N^*$, compute $v \equiv x^2 \bmod N$ and sets $msk_{b'} = v$. Then, $\mathcal{B}$ sets $U \equiv 1/msk_{b'}^{2^{3l(T+1-b')}} \bmod N$ and gives $VK = (N, U, T)$ to $\mathcal{A}$. $\mathcal{B}$ maintains two hash tables to respond to the hash queries to H_1 and H_2.

[Interactive query phase] In this phase, $\mathcal{A}$ is allowed to query four types of oracles **KeyIssue, Update, Sign,** and **Hash** as follows.

KeyIssue query : The adversary $\mathcal{A}$ chooses a specific ID and a time period i, and sends them to $\mathcal{B}$. If $i < b'$ then $\mathcal{B}$ selects a random $Z \in Q$ and chooses random string $h_1 \in \{0,1\}^l$. Y is computed as $U^{h_1} Z^{2^{3l(T+1-i)}}$. If $H_1(Y, ID)$ is defined in the H_1 table then fails and aborts ; otherwise, sets $H_1(Y, ID)$ as h_1 and adds the entry (Y, ID, h_1) into the H_1 table. Set $SK_{i,ID}$ = Z. After that, $\mathcal{B}$ returns the signing key $SK_{i,ID}$ = $(N, T, Y, sk_{i,ID})$ for ID of current time period i to adversary $\mathcal{A}$. Consider $i \geq b'$. Since $\mathcal{B}$ has $msk_{b'} = v$, it can obtain msk_i by executing MSKUpdate algorithm. $\mathcal{B}$ returns the singing key $SK_{i,ID}$ simply by applying the KeyIssue algorithm. $\mathcal{B}$ maintains a KeyIssue table and updates the entry $(SK_{i,ID}, i, ID)$ into the table.

Update query : The adversary $\mathcal{A}$ requests the challenger $\mathcal{C}$ to execute the MSKUpdate algorithm and UKUpdate algorithm. If $i \leq b'$ then nothing is performed. Otherwise MSKUpdate and UKUpdate is called.

Sign query : $\mathcal{A}$ chooses a random message M and a user identifier ID, and asks $\mathcal{B}$ to sign a message M for ID on the current time period i. If $i \geq b'$, $\mathcal{B}$ can generate any signing key as above. Otherwise, $\mathcal{B}$ first checks whether the ID and i entry exist in the KeyIssue table. If so, $\mathcal{B}$ generates a signature using the private key. If the ID and i entry does not exist in the table, $\mathcal{B}$ generates the key in a manner similar to the **KeyIssue query** phase. $\mathcal{B}$ selects random $\sigma_i \in \mathbb{Z}_N^*$ and $h_2 \in \{0,1\}^l$. $\mathcal{B}$ selects a random Z and h_1 again, computes $Y = Z^{2^{3l(T+1-i)}} \cdot U^{h_1}$, and adds them into H_1 and uses them. Then, Y' is computed as $\sigma_j^{2^{3l(T+1-t)}} \cdot U^{h_1 \cdot h_2} / Y^{h_2}$. $\mathcal{B}$, finally, returns (σ', Y, Y') to $\mathcal{A}$.

Hash query : The adversary $\mathcal{A}$ can make queries to the random oracle and get the corresponding random values.

[PKG corrupt phase] Upon the request of the PKG master secret key, $\mathcal{B}$ marks the current time period as b. If $b < b'$, $\mathcal{B}$ aborts. Otherwise, since $\mathcal{B}$ has $msk_{b'}$, $\mathcal{B}$ can compute $msk_b \leftarrow v^{2^{3l(b-b')}}$ and return it to $\mathcal{A}$.

[Final forgery phase] In order to compute the factors of blum integer N, we utilize double forking here. As usual, $\mathcal{B}$ runs $\mathcal{A}$ twice with the same random tape, then $H_1(Y \parallel ID^*)$ would be h_1 at the first case, and $\overline{h_1}$ for the second case. In each case, $\mathcal{B}$ runs $\mathcal{A}$ twice, so that each of them outputs different h_2's this time.

In this phase, $\mathcal{A}$ gives a forgery signature (σ_j, Y'^*, Y^*) of identifier ID^* and message M^* to $\mathcal{B}$, where $j < b'$. As we described before, $H_1(Y^* \parallel ID^*)$ returns different h_1 and $\overline{h_1}$ for each case. First of all, when h_1 was returned, $H_2(Y^* \parallel Y'^* \parallel j \parallel M^*)$ is h_2, and $\mathcal{B}$ is given the forgery signature (σ_j, Y'^*, Y^*) from $\mathcal{A}$. So, the equation (1) is

valid.

$$(\sigma_j)^{2^{3l(T+i-j)}} \cdot U^{h_1 \cdot h_2} = Y'^* \cdot Y^{*h_2} \tag{1}$$

$\mathcal{B}$, then, rewinds with the same random tape, $H_2(Y^* \parallel Y'^* \parallel j \parallel M^*)$ is h_2^* and get the different forgery signature (σ_j^*, Y'^*, Y^*) from $\mathcal{A}$. We obtain the equation (2) at this time.

$$(\sigma_j^*)^{2^{3l(T+i-j)}} \cdot U^{h_1 \cdot h_2^*} = Y'^* \cdot Y^{*h_2^*} \tag{2}$$

With dividing (1) by (2), equation (3) comes out.

$$(\sigma_j/\sigma_j^*)^{2^{3l(T+1-j)}} = (Y^*/U^{h_1})^{h_2 - h_2^*} \tag{3}$$

$\mathcal{B}$ resets the nesting fork, $H_1(Y^* \parallel ID^*)$ is $\overline{h_1}$ in this time. In this round, we can get two equations (4) and (5), and equation (6) is computed by (4)/(5).

$$\overline{(\sigma_{j'})}^{2^{3l(T+1-j')}} \cdot U^{\overline{h_1} \cdot \overline{h_2}} = \overline{Y'^*} \cdot Y^{*\overline{h_2}} \tag{4}$$

$$\overline{(\sigma_{j'}^*)}^{2^{3l(T+1-j')}} \cdot U^{\overline{h_1} \cdot \overline{h_2^*}} = \overline{Y'^*} \cdot Y^{*\overline{h_2^*}} \tag{5}$$

$$(\overline{\sigma_{j'}}/\overline{\sigma_{j'}^*})^{2^{3l(T+1-j')}} = (Y^*/U^{\overline{h_1}})^{\overline{h_2} - \overline{h_2^*}} \tag{6}$$

By calculating $(3)^{(\overline{h_2} - \overline{h_2^*})}/(6)^{(h_2 - h_2^*)}$ and replacing U with $1/v^{2^{3l(T+1-b')}}$, the following equation (7) can be derived. We assume without loss of generality that $j \geq j'$.

$$\frac{(\sigma_j/\sigma_j^*)^{(\overline{h_2} - \overline{h_2^*}) \cdot 2^{3l(b'-j)}}}{(\overline{\sigma_j}/\overline{\sigma_j^*})^{(h_2 - h_2^*) \cdot 2^{3l(b'-j')}}} = v^{(h_1 - \overline{h_1})(h_2 - h_2^*)(\overline{h_2} - \overline{h_2^*})} \tag{7}$$

By Lemma 2 in, if we set $\alpha = (h_1 - \overline{h_1})(h_2 - h_2^*)(\overline{h_2} - \overline{h_2^*}), X = \frac{Z}{\overline{Z}^{2^{3l(j-j')}}}, \lambda = 3l(b'-j)$, we can easily get the square root of v, where $Z = (\sigma_j/\sigma_j^*)^{(\overline{h_2} - \overline{h_2^*})}$ and $\overline{Z} = (\overline{\sigma_j}/\overline{\sigma_j^*})^{(h_2 - h_2^*)}$.

Because v has four square roots and x is random, with probability 1/2 we have that $x \not\equiv \pm y \pmod{N}$ and, hence $\mathcal{B}$ will be able to find a factor of N by computing the gcd of $x - y$ and N.

To succeed the forgery, i should be $j < i \leq t'$. So, the probability loss is at most $\frac{1}{T}$, and this is used as acc in Lemma 1. According to Lemma 1, the final probability that $\mathcal{B}$ succeeds becomes at least $\frac{\epsilon}{2T} \cdot \left(\frac{\epsilon^3}{T^3 \cdot q^6} - \frac{28}{2^{k+3}} \right)$ where $n = 3$, and $q = q_{hash} + q_{sign} + q_{key}$.

6 Optimization of Performance Improvement

The key issuing and singing in our scheme require computing Y and Y' such that $Y \leftarrow R^{2^{3l(T+1-j)}} \bmod N$ and $Y' \leftarrow (R')^{2^{3l(T+1-i)}} \bmod N$ for randomly chosen numbers R, and R', of which time complexity is proportional to the maximum period T. Fast-AR scheme [10] accelerates the signing algorithm without sacrificing the other factors.

In the scheme, instead of squaring R to compute Y, it chooses an exponent r randomly and then computes $R \leftarrow g^r$ and $Y \leftarrow X^r$ where X is precomputed and $X = g^{2^{l(T+1)}} \pmod N$.

We can apply the Fast-AR scheme to the proposed FSIBS. In the KeyIssue algorithm, we choose a random exponent r to compute R and Y such that $R \leftarrow g^r$ and $Y \leftarrow X^r$. In the Sign algorithm, similarly we choose an exponent r' randomly and compute R' and Y' such that $R' \leftarrow g^{r'}$ and $Y' \leftarrow X^{r'}$. Note that X is precomputed in advance. The verification algorithm is revised similar to the Fast-AR [10]. Then the time complexities of the KeyIssue and the Sign algorithms are independent of the maximum period T. In the experiment we compare the optimized FSIBS with the original FSIBS.

7 Experiment

In this section, we implement the proposed FSIBS and the optimized FSIBS(opt-FSIBS) in section 6 to validate the practicality. The FSIBS and opt-FSIBS are implemented using openssl library in C. For comparison, we also implement the pairing based scheme [15] using PBC library in C. The experiment is performed on Intel i7-8700K 3.7GHz machine with 24GB RAM using Ubuntu 18.04.

Table 2: Comparison of FSIBS, opt-FSIBS, and [15]

		FSIBS	**opt-FSIBS**	**[15]**
PKGKeyGen	[s]	0.070354	0.065916	2.026031
KeyIssue	[s]	24.142792	0.003250	1.991108
Sign	[s]	24.112566	0.003366	0.075265
Verify	[s]	24.169596	24.017984	0.061528
MSKUpdate	[s]	0.000275	0.000360	1.869376
UKUpdate	[s]	0.000269	0.000275	1.871108
MSK	[Byte]	264	264	2644
VK	[Byte]	260	260	5440
SK	[Byte]	520	520	3060
Signature	[Byte]	388	388	104

Table 2 shows the execution time and the key sizes when the security parameter k is 2048 for FSIBS and opt-FSIBS and $k = 224$ for [15]. The maximum time period T is 2^{15} and the hash length l is 160. The execution times of FSIBS and opt-FSIBS are almost equivalent except for the KeyIssue and Sign time. KeyIssue and Sign times decrease to 3 ms from 24s in opt-FSIBS which is much faster than the pairing based approach in [15]. Note that the verification time is quite large in the proposed scheme. However, the proposed scheme is practically applicable when the verification is not frequently occurred while the sign is performed periodically such as a streaming application [9]. The proposed schemes require less memory space to store keys than the pairing based approach.

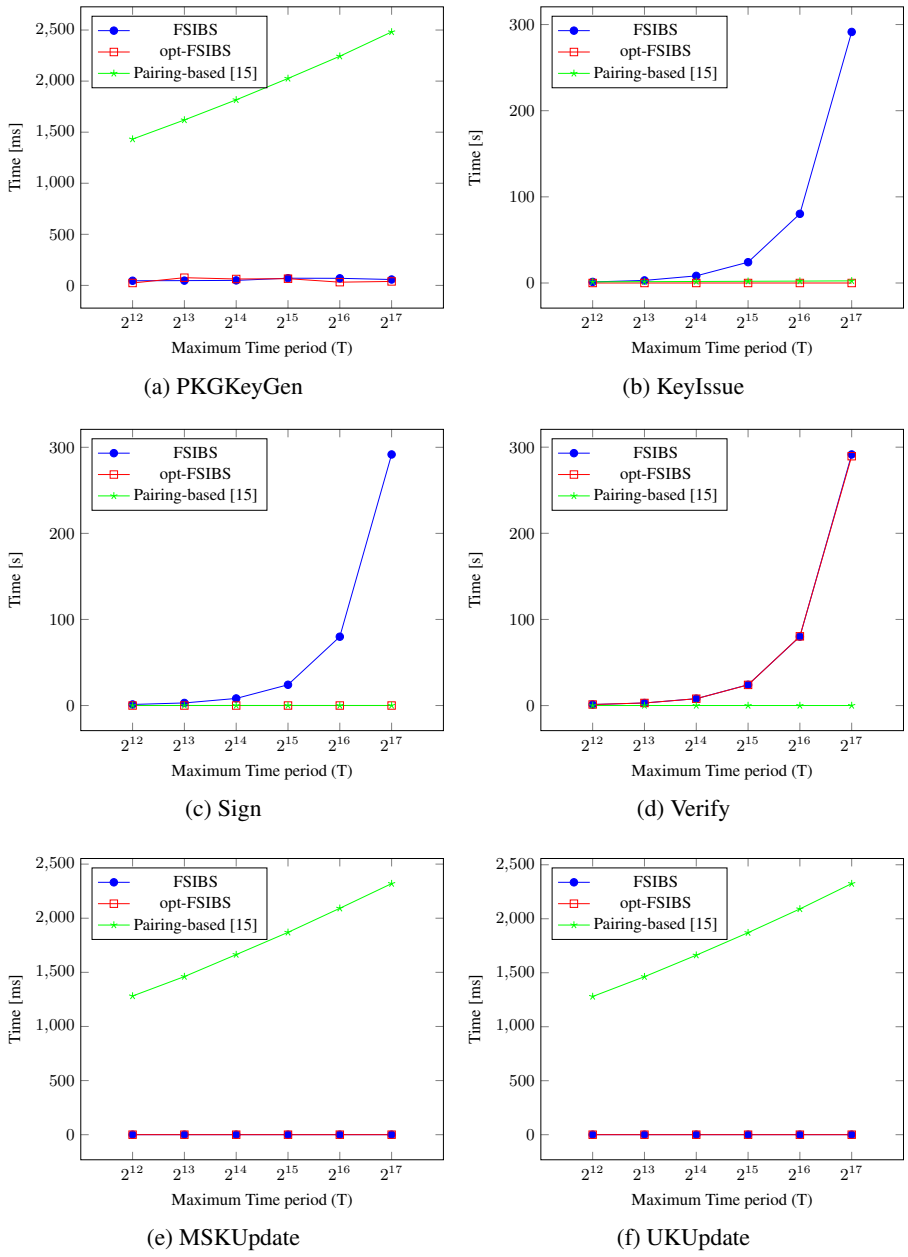

Fig. 1: The execution time by varying the maximum time periods

Figure 1 illustrates the execution time by varying the maximum time periods. Figure 1(a), 1(e), and 1(f) show the execution times of PKGKeyGen, MSKUpdate, and

UKUpdate algorithms in each scheme. They show that the execution times of our schemes are shorter than those of [15] always. Figure 1(b) and 1(c) illustrate the execution time of the KeyIssue and Sign algorithm in each scheme. According to these graphs, the KeyIssue time and Sign time of opt-FSIBS and [15] are both short enough. Lastly, Figure 1(d) shows that the Verify algorithm in both FSIBS and opt-FSIBS takes much longer than [15]. Note that although Verify is much slower in our scheme than [15], it is relatively insignificant in a streaming application.

8 Conclusion

This paper proposes a forward-secure ID-based digital signature scheme with forward secure private key generator based on RSA assumption. We define its notion and provide practical constructions and its security proof under the factoring assumption in the random oracle model. The signing algorithm in the proposed scheme is fast enough to be applied in a streaming real-time application with constant size keys and constant size signatures while the verification time is quite long. The proposed scheme is implemented in a real system and the experimental results validate the practicality of the proposed scheme.

Acknowledgement

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (No.2016-6-00599, A Study on Functional Signature and Its Applications and No. 2017-0-00661, Prevention of video image privacy infringement and authentication technique), by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (No. 2017R1A2B4009903 and No. 2016R1D1A1B03934545), and by Basic Research Laboratory Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT & Future Planning(MSIP)(No. 2017R1A4A1015498). J. Kim and H. Oh are co-corresponding authors.

References

1. M. Abdalla and L. Reyzin. A new Forward-Secure digital signature scheme. In *ASIACRYPT*, pages 116–129, 2000.
2. R. Anderson. Two remarks on public-key cryptology - invited lecture. In *In the fourth ACM Conference on Computer and Communications Security (CCS)*, 1997.
3. M. Bellare and S. K. Miner. A forward-secure digital signature scheme. In M. J. Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 431–448. Springer, 1999.
4. X. Boyen, H. Shacham, E. Shen, and B. Waters. Forward-secure signatures with untrusted update. In A. Juels, R. N. Wright, and S. D. di Vimercati, editors, *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006, Alexandria, VA, USA, Ioctober 30 - November 3, 2006*, pages 191–200. ACM, 2006.

5. S. Chatterjee and C. Kamath. A closer look at multiple forking: Leveraging (in)dependence for a tighter bound. *Algorithmica*, 74(4):1321–1362, 2016.
6. N. A. Ebri, J. Baek, A. Shoufan, and Q. H. Vu. Forward-secure identity-based signature: New generic constructions and their applications. *JoWUA*, 4(1):32–54, 2013.
7. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In A. M. Odlyzko, editor, *Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, 1986.
8. G. Itkis and L. Reyzin. Forward-secure signatures with optimal signing and verifying. In J. Kilian, editor, *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*, volume 2139 of *Lecture Notes in Computer Science*, pages 332–354. Springer, 2001.
9. J. Kim, S. Lee, J. Yoon, H. Ko, S. Kim, and H. Oh. Pass: Privacy aware secure signature scheme for surveillance systems. In *Advanced Video and Signal-based Surveillance (AVSS), 2017 IEEE Symposium on*. IEEE, 2017.
10. J. Kim and H. Oh. Forward-secure digital signature schemes with optimal computation and storage of signers. In S. De Capitani di Vimercati and F. Martinelli, editors, *ICT Systems Security and Privacy Protection*, pages 523–537, Cham, 2017. Springer International Publishing.
11. H. Krawczyk. Simple forward-secure signatures from any signature scheme. In *CCS 2000, Proceedings of the 7th ACM Conference on Computer and Communications Security, Athens, Greece, November 1-4, 2000.*, pages 108–115, 2000.
12. Y. Liu, X. Yin, and L. Qiu. Id-based forward-secure signature scheme from the bilinear pairings. In F. Yu, Q. Luo, Y. Chen, and Z. Chen, editors, *Proceedings of The International Symposium on Electronic Commerce and Security, ISECS 2008, August 3-5, 2008, Guangzhou, China*, pages 179–183. IEEE Computer Society, 2008.
13. T. Malkin, D. Micciancio, and S. K. Miner. Efficient generic forward-secure signatures with an unbounded number of time periods. In L. R. Knudsen, editor, *Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings*, volume 2332 of *Lecture Notes in Computer Science*, pages 400–417. Springer, 2002.
14. R. C. Merkle. A digital signature based on a conventional encryption function. In C. Pomerance, editor, *Advances in Cryptology - CRYPTO '87, A Conference on the Theory and Applications of Cryptographic Techniques, Santa Barbara, California, USA, August 16-20, 1987, Proceedings*, volume 293 of *Lecture Notes in Computer Science*, pages 369–378. Springer, 1987.
15. H. Oh, J. Kim, and J. S. Shin. Forward-secure ID based digital signature scheme with forward-secure private key generator. *Inf. Sci.*, 454-455:96–109, 2018.
16. C. Schnorr. Efficient identification and signatures for smart cards (abstract). In *Advances in Cryptology - EUROCRYPT '89, Workshop on the Theory and Application of of Cryptographic Techniques, Houthalen, Belgium, April 10-13, 1989, Proceedings*, pages 688–689, 1989.
17. J. Wei, W. Liu, and X. Hu. Forward-secure identity-based signature with efficient revocation. *Int. J. Comput. Math.*, 94(7):1390–1411, 2017.
18. J. Yu, R. Hao, F. Kong, X. Cheng, J. Fan, and Y. Chen. Forward-secure identity-based signature: Security notions and construction. *Inf. Sci.*, 181(3):648–660, 2011.
19. J. Yu, H. Xia, H. Zhao, R. Hao, Z. Fu, and X. Cheng. Forward-secure identity-based signature scheme in untrusted update environments. *Wireless Personal Communications*, 86(3):1467–1491, 2016.
20. X. Zhang, C. Xu, C. Jin, and R. Xie. Efficient forward secure identity-based shorter signature from lattice. *Computers & Electrical Engineering*, 40(6):1963–1971, 2014.