On The Security of Two Key-Updating Signature Schemes

Xingyang Guo

National University of Defense Technology, College of Electronic Science and Engineering,
410073 Changsha, China
{saga_gxy}@sina.com

Abstract. In ICICS 2004, Gonzalez-Deleito, Markowitch and Dall'Olio proposed an efficient strong key-insulated signature scheme. They claimed that it is $(N-1, N)$-key-insulated, i.e., the compromise of the secret keys for arbitrarily many time periods does not expose the secret keys for any of the remaining time periods. But in this paper, we demonstrate an attack and show that an adversary armed with the signing keys for any two time periods can compute the signing keys for the remaining time periods except for some very special cases. In a second attack, the adversary can forge signatures for many remaining time periods without computing the corresponding signing keys. Therefore it is only equivalent to a $(1, N)$-key-insulated signature scheme. A variant forward-secure signature scheme was also presented in ICICS 2004 and claimed more robust than traditional forward-secure signature schemes. But we find that the scheme has two similar weaknesses. We try to repair the two schemes in this paper.

1 Introduction

Many cryptographic techniques today, whether only available in the literature or actually used in practice, are believed to be quite secure. Several, in fact, can be proven secure (with appropriate definitions)under very reasonable assumptions. In a vast majority of solutions, however, security guarantees last only as long as secrets remain unrevealed. If a secret is revealed (either accidentally or via an attack), security is often compromised not only for subsequent uses of the secret, but also for prior ones. For example, if a secret signing key becomes known to an adversary, one cannot trust any signature produced with that key and the signer is forced to revoke its public key. Unfortunately, this does not always suffice as even valid signatures having been produced before the revealment become invalid, unless a time-stamping authority has attested that they were produced before the corresponding public key was revoked.

Getting rid of the revocation and time-stamping mechanisms in order to simplify key management is an active research topic. In recent years, some key-updating approaches are presented to limit the damages arising when secret keys are exposed.

An approach to this problem is the forward-secure cryptosystem. In the forward-secure model [2,3], the lifetime of secret keys is divided into discrete time periods. At the beginning of each period, users compute a new secret key by applying a public one-way function to the secret key used during the previous time period, while public keys remain unchanged. An adversary compromising the secret signing key at a given time period will be unable to produce signatures for previous periods, but will still be able to sign messages during the current and future time periods. Unlike classical schemes, the validity of previously produced signatures is therefore assured, but public keys have to be revoked. Several recent investigations in forward-secure signature scheme are given in [1,7,9].

The notion of key-insulated cryptosystems, which was introduced by Dodis et al. [4], generalises the concept of forward-secure cryptography. In this model, lifetime of secret keys is also divided into discrete periods and, as in previous models, signatures are supposed to be generated by relatively insecure devices. However, the secret associated with a public key is here shared between the user and a physically secure device. At the beginning of each time period the user obtains from the device a partial secret key for the current time period. By combining this partial secret key with the secret key for the previous period, the user derives the secret key for the current time period. Exposure of the secret key at a given period will not enable an adversary to derive secret keys for the remaining time periods. More precisely, in a (t, N)-key-insulated scheme the compromise of the secret key for up to t time periods does not expose the secret key for any of the remaining $N - t$ time periods. Therefore, public keys do not need to be revoked unless t periods have been exposed. Strong key-insulated schemes [5] guarantee that the physically secure device (or an attacker compromising the partial secrets held by this device) is unable to derive the secret key for any time period. This is an extremely important property if the physically secure device serves several different users.

Itkis and Reyzin [8] introduced the notion of intrusion-resilient signatures, which strengthens the one of key-insulation by allowing an arbitrary number of non-simultaneous compromises of both the user and the device, while preserving security of prior and future time periods.

In ICICS'04, Gonzalez-Deleito, Markowitch and Dall'Olio [6] proposed a new strong $(N-1, N)$-key-insulated signature scheme (GMD scheme, from now on). The scheme is more efficient than previous proposals and has the property that becomes forward-secure when all the existing secrets at a given time period are compromised. They also presented a variant forward-secure signature scheme.

In this paper, we demonstrate two attacks on the two schemes, respectively. These attacks may have many variations. The two attacks on the key-insulated signature scheme show that it is only equivalent to a $(1, N)$-key-insulated signature scheme. The attacks on the forward-secure signature scheme are similar to the attacks on the key-insulated signature scheme. It is showed that the scheme cannot be used as a forward-secure signature scheme. We try to repair the two schemes.

The remaining of the paper is organized as follows: section 2 briefly describes some definitions of key-updating signature schemes. Section 3 reviews and analyzes the GMD key-insulated signature scheme, section 4 reviews and analyzes the GMD forward-secure signature scheme, section 5 describes our attempts at repairing the two schemes, section 6 concludes.

2 Definitions of Key-Updating Signature Schemes

The following definitions of key-insulated signature schemes are based on the definitions given by Gonzalez-Deleito et al.[6].

A *key-insulated signature scheme* is a 5-tuple of polynomial time algorithms (KGen, UpdD, UpdU, Sig, Ver) such that:

- KGen, the key generation algorithm, is a probabilistic algorithm taking as input one or several security parameters *sp* and (possibly) the total number of periods N, and returning a public key PK, a master secret key MSK and a user's initial secret key USK_0.

- UpdD, the physically secure device key-update algorithm, is a (possibly) probabilistic algorithm which takes as input the index i of the next time period, the master secret key MSK and (possibly) the total number of periods N, and returns a partial secret key PSK_i for the i-th time period.

- UpdU, the user key-update algorithm, is a deterministic algorithm which takes as input the index i of the next time period, the user's secret key USK_{i-1} for the current time period and the partial secret key PSK_i. It returns the user's secret key USK_i and the secret signing key SK_i for the next time period.

- Sig, the signing algorithm, is a probabilistic algorithm which takes as input the index i of the current time period, a message M and the signing key SK_i for the time period i; it returns a pair $< i, s >$ composed of the time period i and a signature s.

- Ver, the verification algorithm, is a deterministic algorithm which takes as input a message M, a candidate signature $< i, s >$ on M, the public key PK and (possibly) the total number of periods N; it returns **true** if $< i, s >$ is a valid signature on M for period i, and **false** otherwise.

The life cycle of keys in a key-insulated scheme can be described as follows. A user begins by running the KGen algorithm, obtaining a public key PK, as well as the corresponding master secret key MSK and user's initial secret key USK_0. The public key PK is certified through a certification authority (CA) and made publicly available, while MSK is stored on the physically secure device and USK_0 is stored by the user himself. For each time period i, $1 \leq i \leq N$, the user is now able to obtain a partial secret key PSK_i by asking the device to run the UpdD algorithm. By executing UpdU, the user transforms, with the help of USK_{i-1}, the partial secret key received from the device into a signing key SK_i for time period i which may be used to sign messages during this time period. Furthermore, the user updates USK_{i-1} to USK_i and erases USK_{i-1} and SK_{i-1}.

We suppose that an adversary may

- ask for signatures on adaptively chosen messages for adaptively chosen time periods;

- either expose the insecure signing device for up to t adaptively chosen time periods or expose once the physically secure device;

- compromise the insecure signing device during an update.

If the adversary cannot succeed to forge a valid signature $< i, s >$ on a message M for which he never requested a signature for time period i and he never exposed the insecure device at this time period, the key-insulated signature scheme is *secure*.

The forward-secure signature scheme can be regarded as the simplified version of the key-insulated signature scheme. In traditional forward-secure signature schemes, there are no physically secure devices and UpdD phases and the only secure time periods is that prior to the compromised time periods.

3 The GMD Key-Insulated Signature Scheme and Its Security

3.1 Review of the GMD Key-Insulated Signature Scheme

KeyGen(k, l) k and l are two security parameters. Let $n = pq$ be a k-bit modulus, where $p = 2p' + 1$ and $q = 2q' + 1$ are safe primes numbers such that p' and q' are also safe primes. Let v be an $(l + 1)$-bit prime number. And let h be a one-way hash function $h : \{0, 1\} \to \{0, 1\}^l$ (in the following we will note by $h(a, b)$ the result of applying to h the concatenation of a value a with a value b). The user randomly chooses $s, t, u \in Z_n^*$, such that $s^2 \neq s^{2^{8+1}} \bmod n$, $t^2 \neq t^{2^{8+1}} \bmod n$ and $u^2 \neq u^{2^{8+1}} \bmod n$. The public key PK is composed of $PK_1 = s^{-v} \bmod n$, $PK_2 = t^{-v} \bmod n$ and $PK_3 = u^{-v} \bmod n$. The master secret key MSK is composed of $MSK_1 = s^2 \bmod n$ and $MSK_2 = t^2 \bmod n$, and the user's initial secret key is $USK_0 = u^2 \bmod n$.

UpdD(i,N,MSK) The physically secure device computes the partial secret key for the i-th time period as follows:

$$PSK_i = (MSK_1)^{2^i} \cdot (MSK_2)^{2^{N-i}} \bmod n = s^{2^{i+1}} \cdot t^{2^{N+1-i}} \bmod n.$$

UpdU(i, USK_{i-1}, PSK_i) The user computes the user's secret key for the time period i

$$USK_i = (USK_{i-1})^2 \bmod n = u^{2^{i+1}} \bmod n$$

and the corresponding signing key

$$SK_i = PSK_i \cdot USK_i \bmod n = s^{2^{i+1}} \cdot t^{2^{N+1-i}} \cdot u^{2^{i+1}} \bmod n.$$

$Sig_{SKi}(i, M)$ In order to sign a message M during the time period i, the user randomly chooses a value $x \in Z_n^*$, computes $y = x^v \bmod n$, $d = h(i, M, y)$ and $D = x \cdot (SK_i)^d \bmod n$. The signature on M for the time period i is (i, d, D).

$Ver_{PK}(M, (i, d, D), N)$ For verifying whether (i, d, D) is a valid signature on M for the time period i, an entity computes

$$h(i, M, D^v \cdot ((PK_1)^{2^{i+1}} \cdot (PK_2)^{2^{N+1-i}} \cdot (PK_3)^{2^{i+1}})^d \bmod n)$$

and accepts the signature only if the result is equal to d.

In paper[6], the authors claimed that it is a $(N-1, N)$-key-insulated signature scheme. It is also be claimed that the scheme can be used for signature delegation. In this context, a user grants to another user the right to sign messages on his behalf during a limited amount of time. It is suggested that this kind of delegation can be simply achieved by giving to this second user a signing key for the corresponding time period. We demonstrate two attacks on the scheme assuming that an adversary has derived signing keys SK_i and SK_j $(i < j)$ for time periods i and j.

3.2 The First Attack on The Scheme

With the signing keys SK_i and SK_j, the adversary can carry out the attack to compute the signing keys for all the remaining time periods as follows:

step 1 Computes

$$\begin{aligned} K_{su} &= SK_j^{2^{j-i}} \cdot SK_i^{-1} = (su)^{2^{i+1} \cdot (2^{2j-2i}-1)} \bmod n \\ K_t &= SK_i^{2^{j-i}} \cdot SK_j^{-1} = t^{2^{N-j+1} \cdot (2^{2j-2i}-1)} \bmod n \end{aligned}$$

step 2 Computes

$$\begin{aligned} (su)^v &= (PK_1 \cdot PK_3)^{-1} \bmod n \\ t^v &= (PK_2)^{-1} \bmod n \end{aligned}$$

step 3 Computes a_{su} and a_t such that

$$\begin{aligned} v \cdot a_{su} &= 1 \bmod \quad 2^{i+1} \cdot (2^{2j-2i} - 1) \\ v \cdot a_t &= 1 \bmod \quad 2^{N-j+1} \cdot (2^{2j-2i} - 1) \end{aligned}$$

step 4 Computes b_{su} and b_t

$$b_{su} = \frac{v \cdot a_{su} - 1}{2^{i+1} \cdot (2^{2j-2i} - 1)}$$
$$b_t = \frac{v \cdot a_t - 1}{2^{N-j+1} \cdot (2^{2j-2i} - 1)}$$

step 5 Computes

$$su = ((su)^v)^{a_{su}} (K_{su}^{b_{su}})^{-1} \bmod n$$
$$t = (t^v)^{a_t} (K_t^{b_t})^{-1} \bmod n$$

step 6 For the time period r, the adversary computes the corresponding signing key

$$SK_r = (su)^{2^{r+1}} \cdot t^{2^{N+1-r}} \bmod n$$

It should be noticed that the computation of a_{su} and a_t in step 3 will not always succeed. The computation will succeed when v is coprime with $2^{i+1} \cdot (2^{2j-2i} - 1)$ and $2^{N-j+1} \cdot (2^{2j-2i} - 1)$. Since v is a prime number, the only case when v is not coprime with the two numbers is that v is a factor of $2^{2j-2i} - 1$. When $v > 2^{2j-2i} - 1$, v is not a factor of $2^{2j-2i} - 1$. Since v is a $(l+1)$-bit big prime number, for example $l = 128$ (MD5 for h), the case that v exactly divides $2^{2j-2i} - 1$ is singular.

3.3 The Second Attack on The Scheme

In this attack, we assume that the adversary also has the signing keys SK_i and SK_j. The adversary can forge signatures for some other time periods on an arbitrary message m with non-negligible probability, without computing the corresponding signing keys. This attack shows another weakness that the power step in the key-updating phase of the scheme is too small. To forge a signature for time period r $(r \neq i, j)$, the adversary carries out as follows:

step 1 Computes

$$K_{su} = SK_j^{2^{j-i}} \cdot SK_i^{-1} \bmod n$$
$$K_t = SK_i^{2^{j-i}} \cdot SK_j^{-1} \bmod n$$

step 2 Randomly chooses a value $x \in Z_n^*$, sets $w = 0$ and $y = 1$, computes $x_v = x^v \bmod n$.

step 3 Computes $w = w + 1$ and $y = y \cdot x_v \bmod n$. If $y = 1$, turns back to step 2, else computes $d = h(i, m, y)$.

step 4 Checks whether d can be exactly divided by

$$\begin{cases} (2^{2j-2i} - 1) \cdot 2^{i-r} & case \quad r < i; \\ (2^{2j-2i} - 1) & case \quad i < r < j; \\ (2^{2j-2i} - 1) \cdot 2^{r-j} & case \quad r > j; \end{cases}$$

If not the adversary turns back to step 3, else continues.

step 4 Computes D as

$$\begin{cases} x^w \cdot K_{su}^{(d \;\; div \;\; (2^{2j-2i}-1)\cdot 2^{i-r})} \cdot K_t^{(d \;\; div \;\; (2^{2j-2i}-1))\cdot 2^{j-r}} & mod \;\; n \quad case \quad r < i; \\ x^w \cdot K_{su}^{(d \;\; div \;\; (2^{2j-2i}-1))\cdot 2^{r-i}} \cdot K_t^{(d \;\; div \;\; (2^{2j-2i}-1))\cdot 2^{j-r}} & mod \;\; n \quad case \quad i < r < j; \\ x^w \cdot K_{su}^{(d \;\; div \;\; (2^{2j-2i}-1))\cdot 2^{r-i}} \cdot K_t^{(d \;\; div \;\; (2^{2j-2i}-1)\cdot 2^{r-j})} & mod \;\; n \quad case \quad r > j; \end{cases}$$

The signature on m for the time period r is (r, d, D).

Correctness For simplicity we only demonstrate the validity of the signature in case $r < i$.

$$\begin{aligned} & h(r, m, D^v \cdot ((PK_1)^{2^{r+1}} \cdot (PK_2)^{2^{N+1-r}} \cdot (PK_3)^{2^{r+1}})^d \bmod n) \\ &= h(r, m, D^v \cdot (PK_1 \cdot PK_3)^{d\cdot 2^{r+1}} \cdot (PK_2)^{d\cdot 2^{N+1-r}} \bmod n) \\ &= h(r, m, x^{wv} \cdot K_{su}^{(d \;\; div \;\; (2^{2j-2i}-1)\cdot 2^{i-r})\cdot v} \cdot K_t^{(d \;\; div \;\; (2^{2j-2i}-1))\cdot 2^{j-r}\cdot v} \\ & \qquad \cdot (su)^{-v\cdot d\cdot 2^{r+1}} \cdot t^{-v\cdot d\cdot 2^{N+1-r}} \bmod n) \\ &= h(r, m, x^{wv} \cdot (su)^{2^{i+1}\cdot(2^{2j-2i}-1)(d \;\; div \;\; (2^{2j-2i}-1)\cdot 2^{i-r})\cdot v} \\ & \quad \cdot t^{2^{N-j+1}\cdot(2^{2j-2i}-1)(d \;\; div \;\; (2^{2j-2i}-1))\cdot 2^{j-r}\cdot v} \cdot (su)^{-v\cdot d\cdot 2^{r+1}} \cdot t^{-v\cdot d\cdot 2^{N+1-r}} \bmod n) \\ &= h(r, m, x^{wv} \cdot (su)^{2^{r+1}\cdot d\cdot v} \cdot t^{2^{N-r+1}\cdot d\cdot v} \cdot (su)^{-v\cdot d\cdot 2^{r+1}} \cdot t^{-v\cdot d\cdot 2^{N+1-r}} \bmod n) \\ &= h(r, m, x^{wv} \bmod n) \\ &= d. \end{aligned}$$

Efficiency of the attack We take the case $r < i$ for example. The efficiency of the attack mostly depends on finding d that can be divided by $(2^{2j-2i} - 1) \cdot 2^{i-r}$ by trail and error. Since h is a hash function, its output distribution will be uniform in $[0, 2^l]$. Hence the success probability of finding a proper d is $\frac{1}{(2^{2j-2i}-1)\cdot 2^{i-r}}$ with 1 try and $1 - (1 - \frac{1}{(2^{2j-2i}-1)\cdot 2^{i-r}})^n$ with n tries. The attack is most efficient in the case $j = i + 1$ and $r = i - 1$ while the success probability of finding a proper d will be more than 99% with 26 tries. The attack will be more inefficient when i is more less than j, r more less than i in case $r < i$ and j more less than r when $r > j$. But if i is not very less than j, we think that there always are many time periods that can be attacked with non-negligible probability in polynomial time.

In the GMD key-insulated signature scheme, if an adversary obtains more singing keys or compromises a user at more time periods, he will carry out some variant attacks. The above two attacks show that the scheme is only equivalent to a $(1, N)$-key-insulated signature scheme and vulnerable in the signature delegation scenario.

4 The GMD Forward-Secure Signature Scheme and Its Security

4.1 Review of The GMD Forward-Secure Signature Scheme

KeyGen(k, l) n, v and h are selected as same as that in the key-insulated scheme.The user randomly chooses $t, u \in Z_n^*$, such that $u^2 \neq u^{2^{8+1}}$ mod n and $t^2 \neq t^{2^{8+1}}$

mod n. The public key PK is composed of $PK_1 = t^{-v}$ mod n and $PK_2 = u^{-v}$ mod n. The master secret key is $MSK = t^2$ mod n and the user's initial secret key is $USK_0 = u^2$ mod n.

UpdD(i,N,MSK) The physically secure device computes the partial secret key

$$PSK_i = (MSK)^{2^{N-i}} \bmod n = t^{2^{N+1-i}} \bmod n.$$

UpdU(i,USK_{i-1}, PSK_i) The user computes the user's secret key for the time period i

$$USK_i = (USK_{i-1})^2 \bmod n = u^{2^{i+1}} \bmod n$$

and the corresponding signing key

$$SK_i = PSK_i \cdot USK_i \bmod n = t^{2^{N+1-i}} \cdot u^{2^{i+1}} \bmod n.$$

$Sig_{SKi}(i, M)$ In order to sign a message M during the time period i, the user randomly chooses a value $x \in Z_n^*$, computes $y = x^v$ mod n, $d = h(i, M, y)$ and $D = x \cdot (SK_i)^d$ mod n. The signature on M for the time period i is (i, d, D).

$Ver_{PK}(M, (i, d, D), N)$ For verifying whether (i, d, D) is a valid signature on M for the time period i, an entity computes

$$h(i, M, D^v \cdot ((PK_1)^{2^{N+1-i}} \cdot (PK_2)^{2^{i+1}})^d \bmod n)$$

and accepts the signature only if the result is equal to d.

4.2 The First Attack on The Scheme

We demonstrate an attack quite similar to the first attack on the key-insulated scheme, on the assumption that an adversary compromises a user at one time period i and gets SK_i and USK_i, that are stored by the user himself in the relatively insecure device. The adversary can compute the signing key for any time period r while $r < i$.

step 1 Computes $PSK_i = SK_i \cdot USK_i^{-1} = t^{2^{N+1-i}}$ mod n.

step 2 Computes

$$t^v = PK_1^{-1} \bmod n$$
$$u^v = PK_2^{-1} \bmod n$$

step 3 Computes a_t and a_u such that

$$v \cdot a_t = 1 \bmod \quad 2^{N+1-i}$$
$$v \cdot a_u = 1 \bmod \quad 2^{i+1}$$

Since v is coprime with 2^{N+1-i} and 2^{i+1}, the adversary will succeed to compute a_t and a_u.

step 4 Computes b_t and b_u

$$b_t = \frac{v \cdot a_t - 1}{2^{N+1-i}}$$
$$b_u = \frac{v \cdot a_u - 1}{2^{i+1}}$$

step 5 Computes

$$t = (t^v)^{a_t}(PSK_i^{b_t})^{-1} \bmod n$$
$$u = (u^v)^{a_u}(USK_i^{b_u})^{-1} \bmod n$$

step 6 For the time period r, the adversary computes the corresponding signing key

$$SK_r = t^{2^{N+1-r}} \cdot u^{2^{r+1}} \bmod n$$

4.3 The Second Attack on The Scheme

This attack is quite similar to the second attack on the key-insulated signature scheme. The adversary with SK_i and USK_i can forge a signature for some time period r $(r < i)$ on an arbitrary message m, with a probability that is not negligible. The adversary carries out as follows:

step 1 Computes $PSK_i = SK_i \cdot USK_i^{-1} \bmod n$.

step 2 Randomly chooses a value $x \in Z_n^*$, sets $w = 0$ and $y = 1$, computes $x_v = x^v \bmod n$.

step 3 Computes $w = w + 1$ and $y = y \cdot x_v \bmod n$. If $y = 1$, turns back to step 2, else computes $d = h(i, m, y)$.

step 4 Checks whether d can be exactly divided by 2^{i-r}. If not the adversary turns back to step 3, else continues.

step 5 Computes $D = x^w \cdot (PSK_i^{2^{i-r}})^d \cdot (USK_i)^{(d \;\; div \;\; 2^{i-r})} \bmod n$. The signature on m for the time period r is (r, d, D).

Efficiency of the Attack The efficiency of the attack mostly depends on finding d that can be divided by 2^{i-r}. The success probability of finding a proper d is $\frac{1}{2^{i-r}}$ with 1 try and $1 - (1 - \frac{1}{2^{i-r}})^n$ with n tries. The attack on time period $i - 1$ is most efficient while the success probability of finding a proper d will be more than 99% with 7 tries. The attack will be more inefficient when r is more less than i, but obviously many time periods less than i can be attacked with non-negligible probability in polynomial time.

With the above two attacks, an adversary with secret keys for one time period can compute signing keys or forge signatures for some previous time periods. Therefore the scheme is **not** a forward-secure signature scheme.

5 Attempt to Repair The two Schemes

We only try to repair the key-insulated signature scheme. The forward-secure signature scheme can be repaired in a similar way.

For the first attack, the natural countermeasure is to select the scheme parameter v such that the adversary can not succeed in step 3. A natural idea to resist the second attack, since the efficiency of our attack depends on finding d that can be divided by some certain values, is to remove these values in the scheme. A direct way is to make these values large enough so that an adversary is unable to find a proper d. In our improved schemes, we will update secret keys with the power step 2^l or 2^{-l} rather than 2 or 2^{-1}.

KeyGen(k, l) In this phase, all parameters are generated as same as that in the original scheme except that $v = 2v'$, while v' is a $(l+1)$-bit prime number, $MSK_1 = s^{2^l} \bmod n$, $MSK_2 = t^{2^l} \bmod n$ and $USK_0 = u^{2^l} \bmod n$. Notice that t, s and u should be chosen such that every possible secret key that will be updated takes a large number of values before cycling.

UpdD(i,N,MSK) The physically secure device computes the partial secret key for the i-th time period as follows:

$$PSK_i = (MSK_1)^{2^{l \cdot i}} \cdot (MSK_2)^{2^{l\cdot(N-i)}} \bmod n = s^{2^{l\cdot(i+1)}} \cdot t^{2^{l\cdot(N+1-i)}} \bmod n.$$

UpdU(i, USK_{i-1}, PSK_i) The user computes the user's secret key for the time period i

$$USK_i = (USK_{i-1})^{2^l} \bmod n = u^{2^{l\cdot(i+1)}} \bmod n$$

and the corresponding signing key

$$SK_i = PSK_i \cdot USK_i \bmod n = s^{2^{l\cdot(i+1)}} \cdot t^{2^{l\cdot(N+1-i)}} \cdot u^{2^{l\cdot(i+1)}} \bmod n.$$

$Sig_{SKi}(i, M)$ In order to sign a message M during the time period i, the user randomly chooses a value $x \in Z_n^*$, computes $y = x^v \bmod n$, $d = h(i, M, y)$. The user computes $D = x \cdot (SK_i)^d \bmod n$. The signature on M for the time period i is (i, d, D).

$Ver_{PK}(M, (i, d, D), N)$ For verifying whether (i, d, D) is a valid signature on M for the time period i, an entity computes

$$h(i, M, D^v \cdot ((PK_1)^{2^{l\cdot(i+1)}} \cdot (PK_2)^{2^{l\cdot(N+1-i)}} \cdot (PK_3)^{2^{l\cdot(i+1)}})^d \bmod n)$$

and accepts the signature only if the result is equal to d.

The parameter v in the improved scheme is not coprime with $2^{i+1} \cdot (2^{2j-2i} - 1)$ and $2^{N-j+1} \cdot (2^{2j-2i} - 1)$. An adversary cannot derive any useful numbers in our first attack and its variations that are coprime with v, even armed with the seret keys for $N-1$ time periods. Since d is the output of the hash function h and a l-bit integer, it cannot be exactly divided by the values, such as $(2^{l\cdot(2j-2i)} - 1) \cdot 2^{l\cdot(i-r)}$ and $2^{l\cdot(r-i)}$, that may be conscribed by the adversary in our second attack and its variations.

6 Conlusions

In this paper, we presented security analysis of Gonzalez-Deleito et al.'s key-insulated signature scheme and forward-secure scheme proposed in [6]. By successfully identifying four attacks, we demonstrated that their schemes are insecure. We tried to repair the two schemes. In fact, how to design a secure and efficient key-insulated signature scheme is still a hot topic.

References

1. M. Abdalla and L. Reyzin. A new forward-secure digital signature scheme. In Proceedings of Advances in Cryptology-ASIACRYPT 2000, volume 1976 of Lecture Notes in Computer Science, pages 116-129. Springer-Verlag, Dec. 2000.
2. R. Anderson. Invited lecture, 4th Conference on Computer and Communications Security. ACM, 1997.
3. M. Bellare and S. K. Miner. A forward-secure digital signature scheme. In Proceedings of Advances in Cryptology-CRYPTO 99, volume 1666 of Lecture Notes in Computer Science, pages 431-448. Springer-Verlag, Aug. 1999.
4. Y. Dodis, J. Katz, S. Xu, and M. Yung. Key-insulated public key cryptosystems. In Proceedings of Advances in Cryptology-EUROCRYPT 2002, volume 2332 of Lecture Notes in Computer Science, pages 65-82. Springer-Verlag, Apr. 2002.
5. Y. Dodis, J. Katz, S. Xu, and M. Yung. Strong key-insulated signature schemes. In Proceedings of the 6th International Workshop on Practice and Theory in Public Key Cryptography (PKC 2003), volume 2567 of Lecture Notes in Computer Science, pages 130-144. Springer-Verlag, Jan. 2003.
6. N. Gonzalez-Deleito, O. Markowitch and E. Dall'Olio. A New Key-Insulated Signature Scheme. In Proceedings of the 6th International Conference on Information and Communications Security (ICICS 2004), volume 3269 of Lecture Notes in Computer Science, pages 465-479. Springer-Verlag, October 2004.
7. G. Itkis and L. Reyzin. Forward-secure signatures with optimal signing and verifying. In Proceedings of Advances in Cryptology-CRYPTO 2001, volume 2139 of Lecture Notes in Computer Science, pages 332-354. Springer-Verlag, Aug. 2001.
8. G. Itkis and L. Reyzin. SiBIR: Signer-base intrusion-resilient signatures. In Proceedings of Advances in Cryptology-CRYPTO 2002, volume 2442 of Lecture Notes in Computer Science, pages 499-514. Springer-Verlag, Aug. 2002.
9. A. Kozlov and L. Reyzin. Forward-secure signatures with fast key update. In Proceedings of the 3rd International Conference on Security in Communication Networks (SCN 2002), volume 2576 of Lecture Notes in Computer Science, pages 241-256. Springer-Verlag, Sept. 2002.