Intrusion-Resilient Key-Evolving Schnorr Signature

JoongMan Kim Kwangjo Kim

International Research center for Information Security (IRIS),
Information and Communications University. (ICU),
58-4 Hwa-am Dong, Yuseong Gu, Daejeon, Korea 305-732

{seopo, kkj}@icu.ac.kr

Abstract Exposure of secret keys is one of the most critical issues in a cryptographic system. It seems to be unavoidable. Recently, the notion of *key-evolving paradigm* was proposed as a means of mitigating the harmful effects that key exposure can cause. In this model, the whole lifetime is divided into distinct periods (*e.g.*, days) such that at time period j, the signer holds the secret key SK_j and updates it periodically, while the public key PK is fixed during its lifetime. We present an intrusion-resilient key-evolving Schnorr signature based on this notion. It has the following property: If secret keys of all periods are not compromised, it is not possible to forge signatures relating to non-exposed secret keys. Our scheme is constructed from unforgeably secure Schnorr signature scheme, which is based on the Discrete Logarithm Problem.

1 Introduction

Key exposures appear to be inevitable. Especially, methods to prevent key exposure entirely (*e.g.*, by using tamper-resistant devices) seem cost-prohibitive and impractical for most common applications. Thus minimizing their negative impacts is extremely important. A long line of researches for dealing with this issue has been proposed.

In *forward-secure schemes* [1,2], the secret key is stored by a single signer and this key is updated by the signer at the beginning of every time period. this security preserves the security of past signatures even after the secret signing key has been exposed.

Threshold schemes [6,12] distribute secrets among n devices so that exposure of secrets from, say, t of these devices will not allow an adversary to "break" the scheme.

In *key-insulated schemes* [7], the adversary cannot generate signatures for the future (as well as past) time periods even after learning the current signing key. This security assumes secure storage on a server with which the user periodically communicates.

Recently presented *intrusion-resilient schemes* [8] assume two modules: a (possibly mobile) "*signer*" and a (generally stationary) "*base*" like key-insulated model. Intrusion-resilience preserves the security of past and future time periods even if both signer and base are compromised, as long as the compromises are not simultaneous. In the case of simultaneous compromise, the security of past (not the future) time periods is preserved.

Tzeng *et al.* [13] proposed a *key-evolving paradigm*, like the one used in forward-secure digital signature schemes. They deal with the key exposure problem of public-key encryption schemes, but signature schemes also will have same paradigm. Let the whole lifetime be divided into periods, starting with 0. The public key PK of the signer is fixed for the whole lifetime. The signer's secret key at time period i is $SK_i, i \geq 0$. When time runs from period i to period $i+1$, the signer updates his secret key from SK_i to SK_{i+1} and then deletes SK_i immediately, possibly with help from a trusted agent TA. If attacker breaks into the signer's system during time period i and gets the signer's secret key SK_i at that time period, attacker cannot get the secret keys in the other time periods directly since they have been deleted with the key evolving paradigm, even if the signer is not aware of losing his secret key, he can be sure that only those signatures generated in the time period are forged. In the next time period, the security of newly created signatures is guaranteed. So in this paradigm, we will present the notion of intrusion-resilience different from one which Itkis *et al.* [8]

propose. Our intrusion-resilience will be achieved using the properties of a linear system of equations and the threshold cryptography [6].

The organization of this paper is as follows : We give the functional definition in Section 2. The description of our intrusion-resilient key-evolving Schnorr signature scheme we call IRKE is given in Section 3. We then analyze correctness, efficiency and security of our scheme in Section 4. Finally, we close by conclusion in Section 5.

2 Intrusion-Resilient Security Model

Our definitions are based on the definition of the key-evolving protocols [9]. In the definition, we may assume that there is a trusted agent TA, who holds some secret share for updating secret keys of the signer.

2.1 Functional Definition

Definition 1. A key-evolving *signature scheme is a quadruple of probabilistic polynomial-time algorithms (Gen, Upd, Sign, Ver):*

1. *Gen, the key generation algorithm.*

   In: *security parameter(s)(in unary), the total number T of time periods*

   Out: the public key PK

2. *Upd, the key update algorithm.*

   In: *current secret key SK_i*

   Out: *new secret key SK_{i+1}*

3. *Sign, the signing algorithm.*

   In: *current secret key SK_i , message M*

   Out: *signature (sig, i) on M for current time period i*

4. *Ver, the verifying algorithm.*

   In: *message M, signature (sig, i) and public key PK*

   Out: *"valid" or "invalid"*

We may assume a single TA for simplicity. In practice, we distribute trust to multiple trusted agents such that each TA_j holds a share s_j of the system secret s. The signer with secret key SK_{i-1} and the TA's together can compute SK_i in a secure way, through Shamir's (k, n) threshold scheme [12]. We discuss this in detail in Section 3.

Tzeng *et al.* [13] introduced the concept of resilience for public-key encryption scheme. It will be similar for the signature scheme as follows.

Definition 2.(Resilience) *Assume a security model for signature scheme. A key evolving signature scheme is* z-resilient *if the attacker cannot break the signature scheme under the assumed security model even if he gets z secret keys $SK_{i_1}, SK_{i_2}, ..., SK_{i_z}$.*

Even if the attacker gets z secret keys $SK_{i_1}, SK_{i_2}, ..., SK_{i_z}$ of z time periods, he cannot get an another secret key SK_i, for $i \neq i_l, 1 \leq l \leq z$. Actually, our scheme becomes to be $(T-1)$-resilient scheme, where T represents the total number of time periods. So we present an additional definition about the concept of resilience.

Definition 3. *A key-evolving protocol is intrusion-resilient if it is $(T-1)$-resilient.*

The *random oracle model* assumes that hash functions used in a scheme are "truly random" hash functions [3]. Although the security under the random oracle model is not rigid, it does provide satisfactory security argument to related schemes in most cases [4].

2.2 Cryptographic Assumptions

Our scheme is based on Schnorr signature scheme whose security is based on the Discrete Logarithm Problem. It is described as follow. Given an element g in a group G of order t, and another element y of G, the problem is to find x, where $0 \leq x \leq t-1$, such that y is the result of composing g with itself x times. So the security of our scheme relies on the assumption that discrete logarithms are difficult to compute.

3 Our Intrusion-Resilient Scheme: Construction

Our scheme (denoted IRKE) using the Shamir's (k, n) threshold scheme [12], where several polynomials are employed, is presented as below. We assume that hash functions $h_1 : \{0,1\}^* \rightarrow \{0,1\}^t$ and $h_2 : \{0,1\}^* \rightarrow Z_q$ are defined. h_1 is used in defining a vector function H which is shown in IRKE.*Gen*. And h_2 is used in IRKE.*Sign* and IRKE.*Ver* as in the Schnorr signature scheme.

KEY GENERATION. We first generate N-bit *safe* prime $p : p = 2q + 1$ such that q is odd prime (such q, satisfying $2q + 1$ is prime, is known as *Sophie Germain prime*[5]). Then the signer actually selects the secret information $S = (s_1, s_2, ..., s_T)$

at random, and computes the public key $PK = (g^{s_1}, g^{s_2}, ..., g^{s_T})$ using S. The signer randomly selects T polynomials used in sharing each element s_l of secret information S, $1 \leq l \leq T$. Note that after sharing, the signer discards the secret information S and such T polynomials. The detailed method how the signer makes and distributes shares will be described in update algorithm. Figure 1 describes key generation algorithm.

algorithm IRKE.*Gen*$(1^N, T)$

`Parameter :`
Generate N-bit prime $p \leftarrow 2q+1$ with that q is a prime of at least 160-bit long.
Let G_q denote the subgroup of the quadratic residues modulo p and g the generator of G_q.
`Setting :`
$S \leftarrow (s_1, s_2, ..., s_T)$ in G_q (secret information)
$PK \leftarrow (g^{s_1}, g^{s_2}, ..., g^{s_T})$ in Z_p^*
Select T $(T-1)^{th}$ degree polynomials

$$f_1(x) \equiv s_1 + \sum_{i=1}^{T-1} \alpha_{1,i} x^i \bmod q$$
$$f_2(x) \equiv s_2 + \sum_{i=1}^{T-1} \alpha_{2,i} x^i \bmod q$$
$$\vdots$$
$$f_T(x) \equiv s_T + \sum_{i=1}^{T-1} \alpha_{T,i} x^i \bmod q$$

,where each f_l is used to share s_l, $1 \leq l \leq T$
`Distributing shares :`
Let the signer and TA hold multiple share $(f_1(x_l), f_2(x_l), ..., f_T(x_l))$, for some random $x_l \in Z_q, 1 \leq l \leq n$
`Define :`
$H(x) := h_1(x)$'s bit representation
i.e. $H(x) := (e_1, e_2, ..., e_T) \rightarrow T$ vector
,where each e_i, $1 \leq i \leq T$, is bit (0 or 1) and $h_1(x)$ is a hash function with T-bit output

`Return` (PK)

Fig.1. Key generation.

UPDATE. Key generation is immediately followed by key update. The signer first divides each element s_l of secret information S into n shares $f_l(x_1)$, $f_l(x_2), ..., f_l(x_n)$, $1 \leq l \leq T$, where x_i's, $1 \leq i \leq n$, are distinct and large enough so that the maximum time period never reaches them, and then makes n multiple shares as follows:

$$(f_1(x_1), f_2(x_1), ..., f_T(x_1))$$
$$(f_1(x_2), f_2(x_2), ..., f_T(x_2))$$
$$\vdots$$
$$(f_1(x_n), f_2(x_n), ..., f_T(x_n))$$

Assume that there are j TA's, $j < k$, i.e., TA_1, TA_2,..., TA_j, and each pair of the signer and TA's share a private channel by which secret information can be passed between them. So the signer gives j multiple shares $(f_1(x_{m_1}), f_2(x_{m_1}), ..., f_T(x_{m_1}))$, $1 \leq m_1 \leq j$, to all j TA's individually and stores the remaining multiple shares $(f_1(x_{m_2}), f_2(x_{m_2}), ..., f_T(x_{m_2}))$, $j+1 \leq m_2 \leq n$, by himself.

algorithm IRKE.*Upd*

Assume that j TA's, $j < k$ (threshold value), i.e., $TA_1, TA_2, ..., TA_j$.

1. $TA_{r_1}, 1 \leq r_1 \leq j$, computes
$SKU_{r_1} \leftarrow \{(f_1(x_{r_1}), f_2(x_{r_1}), ..., f_T(x_{r_1}))$
$\bullet H(i+1)\} \bmod q$
,where $\bullet$ means inner product here and throughout this paper, and then sends each SKU_{r_1} to the signer.

2. The signer selects $(k-j)$ multiple shares randomly and computes, $d_1 \leq r_2 \leq d_{k-j}$,
$SKU_{r_2} \leftarrow \{(f_1(x_{r_2}), f_2(x_{r_2}), ..., f_T(x_{r_2}))$
$\bullet H(i+1)\} \bmod q$

3. Finally, the signer computes
$SK_{i+1} \leftarrow \sum_{r_1=1}^{j} SKU_{r_1} \cdot (\prod_{t_1 \leq I \neq r_1 \leq t_k} \frac{x_I}{x_I - x_{r_1}})$
$+ \sum_{r_2=d_1}^{d_{k-j}} SKU_{r_2} \cdot (\prod_{t_1 \leq I \neq r_2 \leq t_k} \frac{x_I}{x_I - x_{r_2}})$
$\bmod q$

$$((t_1, t_2, ..., t_k) = (1, ..., j, d_1, ..., d_{k-j}))$$

`Return` (SK_{i+1})

Fig.2. Key update.

At this time, the following two inequalities must be satisfied :

$$n - j < k \quad \text{and} \quad n < 2k - 2$$

In order that only the signer as well as only TA's cannot update secret key, *i.e.*,the signer must collude some TA's. At time period i, the signer holds SK_i. the signer and TA's would like to compute SK_{i+1}, which shall be known to the signer only. Figure 2 describes key update algorithm.

In Figure 2, we can make the computation verifiable by letting each TA_l publish $g^{f_1(x_l)}, g^{f_2(x_l)}, ..., g^{f_T(x_l)}$. the signer then verifies whether he receives the right share from $TA_l, 1 \leq l \leq j$, by checking

$$g^{SKU_l} \equiv g^{(f_1(x_l), f_2(x_l), ..., f_T(x_l)) \bullet H(i)} \bmod p$$

,where $g^{(f_1(x_l), f_2(x_l), ..., f_T(x_l)) \bullet H(i)}$ means the random multiplication of TA_l's published values based on the hash value of time period i.

algorithm IRKE.*Sign*(M, SK_i)

$k \overset{R}{\leftarrow} Z_q^*$
$r \leftarrow g^k \bmod p$
$e \leftarrow h_2(M, r)$
$z \leftarrow (SK_i) \cdot e + k \bmod q$

`Return` (z, e, i)

Fig.3. Signing algorithm.

algorithm IRKE.*Ver*$(M, PK, (z, e, i))$

Let $PK = (g^{s_1}, g^{s_2}, ..., g^{s_T})$
$v \leftarrow g^z \cdot (g^{(s_1, s_2, ..., s_T) \bullet H(i)})^{-e} \bmod p$
$e' \leftarrow h_2(M, v)$
If $e = e'$, then return 1 else 0

Fig.4. Verifying algorithm.

Signing and Verifying. Our signature and verification algorithms, which are described in Figures 3 and 4, are exactly the same as in the Schnorr signature scheme [11].

In Figure 4, we have a pre-computation process in IRKE.*Ver*. $g^{(s_1, s_2, ..., s_T) \bullet H(i)}$ means the random multiplication of PK's elements based on the hash value of time period i.

4 Security Consideration

We now discuss the correctness, complexity and security of our proposed scheme. Afterwards, T means the total time period.

4.1 Correctness

Theorem 1. *Let* IRKE.*Upd take output* SK_{i+1} *for* $1 \le i < T$. *Then,* $SK_{i+1} \equiv (s_1, s_2, ..., s_T) \bullet H(i+1) \bmod q$.

Proof. By the Lagrange interpolation method, the following equations are satisfied.

$$\begin{aligned}
SK_{i+1} &\equiv \textstyle\sum_{r_1=1}^{j} SKU_{r_1} \cdot (\prod_{t_1 \le I \ne r_1 \le t_k} \frac{x_I}{x_I - x_{r_1}}) \\
&\quad + \textstyle\sum_{r_2=d_1}^{d_{k-j}} SKU_{r_2} \cdot (\prod_{t_1 \le I \ne r_2 \le t_k} \frac{x_I}{x_I - x_{r_2}}) \\
&\quad \bmod q \\
&\quad ((t_1, t_2, ..., t_k) = (1, .., j, d_1, ..., d_{k-j})) \\
&\equiv [\textstyle\sum_{r_1=1}^{j} \{(f_1(x_{r_1}), f_2(x_{r_1}), ..., f_T(x_{r_1})) \\
&\quad \bullet H(i+1)\} \cdot (\prod_{t_1 \le I \ne r_1 \le t_k} \frac{x_I}{x_I - x_{r_1}})] \\
&\quad + [\textstyle\sum_{r_2=d_1}^{d_{k-j}} \{(f_1(x_{r_2}), f_2(x_{r_2}), ..., f_T(x_{r_2})) \\
&\quad \bullet H(i+1)\} \cdot (\prod_{t_1 \le I \ne r_2 \le t_k} \frac{x_I}{x_I - x_{r_2}})] \\
&\quad \bmod q \\
&\equiv [\textstyle\sum_{r=t_1}^{t_k} \{f_1(x_r), f_2(x_r), ..., f_T(x_r)\} \\
&\quad \bullet H(i+1)] \cdot (\prod_{t_1 \le I \ne r \le t_k} \frac{x_I}{x_I - x_r}) \bmod q \\
&\equiv (\textstyle\sum_{r=t_1}^{t_k} f_1(x_r) \cdot (\prod_{t_1 \le I \ne r \le t_k} \frac{x_I}{x_I - x_r}), \\
&\quad \textstyle\sum_{r=t_1}^{t_k} f_2(x_r) \cdot (\prod_{t_1 \le I \ne r \le t_k} \frac{x_I}{x_I - x_r}), ..., \\
&\quad \textstyle\sum_{r=t_1}^{t_k} f_T(x_r) \cdot (\prod_{t_1 \le I \ne r \le t_k} \frac{x_I}{x_I - x_r})) \\
&\quad \bullet H(i+1) \bmod q \\
&\equiv (s_1, s_2, ..., s_T) \bullet H(i+1) \bmod q
\end{aligned}$$

as desired.

We now verify that the verification is performed correctly.

Theorem 2. *Let* $PK = (g^{s_1}, g^{s_2}, ..., g^{s_T})$ *be a public key generated by the above key generation algorithm. Let* $< M, (z, e, i) >$ *be an output of* IRKE.*Sign*(M, SK_i). *Then* IRKE.*Ver*$(M, PK, (z, e, i))$ $= 1$.

Proof. We will show that $v \equiv r (= g^k) \bmod q$. The process is the same as one of the Schnorr signature scheme.

$$\begin{aligned}
v &\equiv g^z \cdot (g^{(s_1, s_2, ..., s_T) \bullet H(i)})^{-e} \bmod p \\
&\equiv g^{(SK_i) \cdot e + k} \cdot (g^{SK_i})^{-e} \bmod p \\
&\equiv g^{(SK_i) \cdot e + k} \cdot g^{-(SK_i) \cdot e} \bmod p \\
&\equiv g^k \bmod p
\end{aligned}$$

Hence $h_2(M, v) = h_2(M, r)$, i.e., $e = e'$, always holds. This means that the verification succeeds.

4.2 Complexity and Efficiency

Our scheme has public key and secret information of size $O(T)$.

Key Generation. In order to complete the key generation algorithm, we first have to generate secret information, and calculate the public key using it. It requires T modular exponentiations.

UPDATE. Key update algorithm consists of summation and multiplication. Since we use Shamir's (k, n) threshold scheme, key update algorithm requires $(k \cdot \frac{T}{2})$ modular summations and k modular multiplications. $\frac{T}{2}$ means that ideal hash operation generates $\frac{T}{2}$'s $'1'$ bits on the average.

SIGNING AND VERIFYING. Our scheme has same signature and verification algorithm as Schnorr one. So it has same efficiency and complexity except for another pre-computation process in our verification algorithm, which takes only $\frac{T}{2}$ modular multiplications. Also $\frac{T}{2}$ has same meaning as above.

4.3 Security Proof

Pointcheval and Stern[10] proved that Schnorr signature scheme is UF-CMA(Unforgeable Against Chosen Message Attack). This means the following Proposition.

Proposition 1. *In the random oracle model, our proposed scheme is unforgeable.*

To complete this proof, we need the following lemma 1.

Lemma 1. *If Schnorr signature scheme is unforgeable, then our proposed scheme is unforgeable.*

Proof. We can rewrite this lemma as *"if proposed scheme is forgeable, then we can build a forger F which can break Schnorr signature scheme"*. Now we construct F. Let's assume signer can create an another valid signature. All information which is given signer is just $(M, < z, e, i >)$. Finally we can know that signer's cheating is $z = (SK_i) \cdot e + k \mod q$. This is the Schnorr signature scheme for M. So F can break Schnorr signature scheme by using signer.

Lu and Shieh [9] proved the following lemma 2, which describes that our scheme is based on the Discrete Logarithm Problem. We present it without proof.

Lemma 2. *The ability of the attacker, who is able to compute the corresponding SK of a given PK $(PK = g^{SK})$, is equivalent to the solving of the Discrete Logarithm Problem in Z_P^*.*

For Theorem 3, we need the following definition.

Definition 4. *A linear system of equations, which is* a set of n linear equations in t unknowns, has the following properties.

1. If $t < n$, then the system is (in general) overdetermined and there is no solution.
2. If $t = n$ and the set of n linear equations in t unknowns are linearly independent, then the system has a unique solution in the t unknowns.
3. Finally, if $t > n$, then the system is underdetermined. In this case, the first $(t - n)$ unknown variables can be solved form in terms of the last n unknown variables to find the solution.

Now we prove Theorem 3.

Theorem 3. *In the random oracle model, our proposed scheme is an intrusion-resilient key-evolving Schnorr signature scheme.*

Proof. Attacker can consider that each element of secret information S is unknown variable and each secret key is linear equation by definition 4. Since the number of all elements in secret information S is T, which represents the total time period, attacker must know T secret keys to compromise a secret key with non-exposed (past or future) time period by definition 4, *i.e.*, he must know secret keys of all periods. So if secret keys of all periods are not compromised, it is not possible to forge signatures relating to non-exposed secret keys. This means that our scheme is $(T-1)$-resilient. Thus it is intrusion-resilient.

4.4 Attacks

We assume that each pair of the signer and TA's share a private channel by which secret information can be passed between them. For now we suppose that such channel does not exist. Since information flows only from all TA's to the signer, we can consider the adversary's possible active/passive attacks related to the SKU value in IRKE.*Upd*. The only way in which the adversary attacks the scheme actively is just to send a bad SKU value. But this can always prohibit him from issuing a valid signature. Also, we can easily know that passive attacker who merely obtains SKU values can not do anything worse that merely sabotage the system. While we do not consider these situations for simplicity, it is easy to show that our proposed scheme is secure against these attacks.

5 Conclusion

We have presented an intrusion-resilient key-evolving Schnorr signature scheme. We believe that intrusion-resilience has the best strength against key expo-

sure problem. To get intrusion-resilience, we have used the properties of a linear system of equations and for updating secret key, we have also used the Shamir's (k, n) threshold scheme together with TA. In our scheme, the signing procedure is as efficient as the signing protocol in the underlying scheme. The main drawback of our scheme is that key update needs help from TA. So it would be interesting to find a key-evolving signature scheme, in which key update can be done by the signer alone.

Application. Since our scheme is based on the discrete logarithm, we can apply for other discrete logarithm based schemes such as the ElGamal scheme (encryption and signature), the Digital Signature Algorithm (DSA), and so on. And the Schnorr signature scheme is particularly suited for smart cards. So our scheme can be also applicable to this application.

References

[1] R. Anderson, "Two Remarks on Public-Key Cryptology", Invited lecture, Fourth Annual Conference on Computer and Communications Security, ACM, 1997, Availabe at **http://www.cl.cam.ac.uk/users/rja14/**.

[2] M. Bellare and S. Miner, "A Forward-Secure Digital Signature Scheme", In Michael Wiener, editor, Advances in Cryptology - Crypto '99, Springer-Verlag, 15-19 August 1999. Revised version is avaiable from **http://www.cs.ucsd.edu/∼mihir/**.

[3] M. Bellare and S. K. Miner, "Random oracles are practical: a paradigm for designing efficient protocols", Proceedings of the First ACM Conference on Computer and Communications Security, pp. 62-73, 1993.

[4] R. Canetti, O. Goldreich, and S. Halevi, "The random oracle methodology revisited", Proceedings of the 30th ACM Annual Symposium on Theory of Computing, pp. 209-218, 1998.

[5] R. Cramer and V. Shoup, "Signature schemes based on the strong RSA assumption", ACM Transactions on Information and System Security, 3(3):161-185, 2000.

[6] Y. Desmedt and Y. Frankel, "Threshold cryptosystems", In G.Brassard, editor, Advances in Cryptology - Crypto '89, Springer-Verlag, LNCS 435, pp. 307-315, 1990.

[7] Y. Dodis, J. Katz, S. Xu, and M. Yung, "Key-Insulated Public-Key Cryptosystems", In Lars Knudsen, editor, Advances in Cryptology - Eurocrypt 2002, Springer-Verlag, LNCS, 28 April-2 May 2002.

[8] G. Itkis and L. Reyzin, "SiBIR: Signer-Base Intrusion-Resilient Signatures", In Moti Yung, editor, Advances in Cryptology - Crypto 2002, Springer-Verlag, LNCS, 18-22 August 2002, Available from **http://eprint.iacr.org/2002/054/**.

[9] C. -F. Lu and S. W. Shieh, "Secure Key-Evolving Protocols for Discrete Logarithm Schemes", In Proceedings of The Cryptographer's Track at the RSA Conference 2002., Springer-Verlag 2002, LNCS 2271, pp. 300-310.

[10] D. Pointcheval and J. Stern, "Security Proofs for Signature Schemes",In Ueli Maurer, editor, Advances in Cryptology - Eurocrypt '96, Springer-Verlag, LNCS 1070, pp. 387-398, 12-16 May 1996.

[11] C. P. Schnorr, "Efficient Identification and Signatures for Smart Card", Advances in Cryptology - Eurocrypt '89, Springer-Verlag, pp. 235-251.

[12] A. Shamir, "How to share a secret", Communications of the ACM, 22(11), pp. 612-613, 1979.

[13] W. -G. Tzeng and Z. -J. Tzeng, "Robust Key-Evolving Public-Key Encryption Schemes", Record 2001/009, Cryptology ePrint Archive, 2001.