Efficient Generic On-Line/Off-Line Signatures Without Key Exposure

Xiaofeng Chen[1,3], Fangguo Zhang[2,3], Willy Susilo[4], and Yi Mu[4]

[1] Department of Computer Science,
Sun Yat-sen University, Guangzhou 510275, P.R. China
isschxf@mail.sysu.edu.cn
[2] Department of Electronics and Communication Engineering,
Sun Yat-sen University, Guangzhou 510275, P.R. China
isszhfg@mail.sysu.edu.cn
[3] Guangdong Key Laboratory of Information Security Technology
Guangzhou 510275, P.R. China
[4] Centre for Computer and Information Security Research,
School of Computer Science and Software Engineering,
University of Wollongong, Australia
{wsusilo,ymu}@uow.edu.au

Abstract. The "hash-sign-switch" paradigm was firstly proposed by Shamir and Tauman with the aim to design an efficient on-line/off-line signature scheme. However, all existing on-line/off-line signature schemes based on Shamir-Tauman's paradigm suffer from the key exposure problem of chameleon hashing. That is, if the signer applies the same hash value more than once to obtain two signatures on two different messages, the recipient can obtain a hash collision and use it to recover the signer's trapdoor information. Therefore, the signer should pre-compute and store plenty of different chameleon hash values and the corresponding signatures on the hash values in the off-line phase, and send the collision and the signature for a certain hash value in the on-line phase. Hence, the computation and storage cost for the off-line phase and the communication cost for the on-line phase in Shamir-Tauman's signature scheme are still a little more overload.

In this paper, we first introduce a special double-trapdoor hash family based on the discrete logarithm assumption to solve this problem. We then apply the "hash-sign-switch" paradigm to propose a much more efficient generic on-line/off-line signature scheme. Additionally, we use a one-time trapdoor/hash key pair for each message signing, which prevents the recipient from recovering the trapdoor information of the signer and computing other collisions.

Keywords: On-line/off-line signatures, Chameleon hashing, Key exposure.

1 Introduction

The notion of on-line/off-line signatures was introduced by Even, Goldreich and Micali [10,11]. It performs the signature generating procedure in two phases.

J. Katz and M. Yung (Eds.): ACNS 2007, LNCS 4521, pp. 18–30, 2007.

The first phase is performed *off-line* (without knowing the signed message) and the second phase is performed *on-line* (after knowing the signed message). On-line/off-line signatures are particularly useful in smart card applications: The off-line phase is implemented either during the card manufacturing process or as a background computation whenever the card is connected to power, and the on-line phase uses the stored result of the off-line phase to sign actual messages. The on-line phase is typically very fast, and hence can be extended efficiently even on a weak processor.

Even, Goldreich and Micali proposed a general method for converting any signature scheme into an on-line/off-line signature scheme. However, the method is not practical because it increases the size of the signature by a quadratic factor. In Crypto 2001, Shamir and Tauman [22] used the so called "chameleon hash functions" to develop a new paradigm, named "hash-sign-switch", for designing much more efficient on-line/off-line signature schemes.

Chameleon hash functions, first introduced by Krawczyk and Rabin [16], are trapdoor one-way hash functions which prevent everyone except the holder of the trapdoor information from computing the collisions for a randomly given input. Chameleon hash functions were originally used to design chameleon signatures, which simultaneously provide non-repudiation and non-transferability for the signed message as undeniable signatures [7] do. In the chameleon signature schemes, the recipient is the holder of trapdoor information, while in case of on-line/off-line signatures, the signer is the holder of the trapdoor information. Therefore, in the off-line phase the signer generates a signature σ by using a provably secure signature scheme to sign the chameleon hash value $h(m', r')$ of a random message m' and a random auxiliary number r'. In the on-line phase, the signer computes a collision r of the chameleon hash function for the given message m such that $h(m, r) = h(m', r')$. The signature for the message m is the pair (σ, r).

In the Shamir-Tauman's on-line/off-line signature schemes, one limitation is that the signature for the different messages must use different chameleon hash values. Otherwise, if the signer uses the same hash value twice to obtain two signatures on two different messages, the recipient can obtain a hash collision and use it to recover the signer's trapdoor information, *i.e.*, the private key. To avoid this problem, the signer must compute and store plenty of different chameleon hash values and the corresponding signatures on the hash values in the off-line phase. Given a signed message in the on-line phase, the signer first chooses a one-time hash value, and then computes a hash collision for the hash value. He then sends the hash collision and the corresponding signature to the recipient. Hence, the computation and storage cost for the off-line phase and the communication cost for the on-line phase in Shamir-Tauman's signature scheme are still a little more overload.

In this paper, for the first time in the literature, we address this problem by introducing a double-trapdoor hash family based on the discrete logarithm assumption and then apply the "hash-sign-switch" paradigm to propose a much more efficient generic on-line/off-line signature scheme. In our signature scheme, the hash value and the corresponding signature are always identical and can be viewed as the public key of the signer. Hence, it is not required to compute and store them in the off-line phase. Additionally, we introduce the idea of *long-term* trapdoor and *one-time* trapdoor in our chameleon hash families, which is similar to the idea of *master* trapdoor and *specific* trapdoor in the multi-trapdoor commitment schemes [13]. The *one-time* trapdoor is used only once for each message signing in the on-line phase, which prevents the recipient from recovering the trapdoor information of the signer and computing other collisions.

In order to achieve the communication and computation advantages of our on-line/off-line signature scheme, we adopt elliptic curve cryptosystems [15,19] to present our double-trapdoor hash family. Certainly, we can design such a double-trapdoor hash family over other generic groups, *e.g.*, the subgroup of $\mathbb{Z}_p^*$. However, we argue that such a double-trapdoor hash family over generic groups is unsuitable for designing *efficient* generic on-line/off-line signature schemes. The reason is as follows: Since the "hash-sign-switch" paradigm is a generic method, it is required that any provably secure signature scheme $\mathcal{S}$ can be used to design the on-line/off-line signature scheme. However, only when the signature length of original signature scheme $\mathcal{S}$ is less than that of a group element, our proposed on-line/off-line signature scheme is superior to Shamir-Tauman's scheme in communication cost.[1] Currently, for any provably secure signature scheme, the signature length is more than 160 bits. Therefore, the elliptic curve cryptosystems seem to be the optimal choice. If we adopt other generic group such as the subgroup of $\mathbb{Z}_p^*$, many signature schemes including some short signature schemes [3,5] can not be used to design our on-line/off-line signature scheme. For more details, please refer to Section 5.2.

1.1 Related Works

As noted in [22], some signature schemes such as Fiat-Shamir, Schnorr, and ElGamal signature schemes [12,21,9] can be naturally partitioned into on-line and off-line phases. The reason is that the first step in these signature schemes does not depend on the given message, and can thus be carried out off-line. However, these are particular schemes with special structure and specific security assumptions rather than a general and provably secure conversion technique for arbitrary signature schemes. Shamir and Tauman introduced the "hash-sign-switch" method for simultaneously improving both the security and the real-time efficiency of any signature scheme by converting it into an efficient on-line/off-line signature scheme. Generally, a new chameleon hash family results in a new on-line/off-line signature scheme. Recently, some variants of on-line/off-line signature schemes [6,17] have been proposed based on Shamir-Tauman's general construction.

However, it seems that all existing on-line/off-line signature schemes based on Shamir-Tauman's paradigm suffer from the key exposure problem of chameleon

[1] In any case, our proposed scheme is no inferior to Shamir-Tauman's scheme in computation and storage cost.

hashing. This problem is firstly addressed by Ateniese and de Medeiros [1] in the original chameleon signature schemes. Chen *et al.* [8] proposed the first full construction of a chameleon hash function without key exposure. Later, Ateniese and de Medeiros presented several constructions of exposure-free chameleon hash functions based on different cryptographic assumptions [2]. However, to the best of our knowledge, there seems to be no existing work that solves the key exposure problem in the generic on-line/off-line signature schemes.

1.2 Organization

The rest of the paper is organized as follows: Some preliminaries are provided in Section 2. The new double-trapdoor chameleon hash family based on the discrete logarithm assumption is presented in Section 3. Our efficient generic on-line/off-line signature scheme is given in Section 4. The security and efficiency analysis of our scheme are given in Section 5. Finally, conclusions will be made in Section 6.

2 Preliminaries

In this section, we introduce the basic notion of chameleon hash family and Shamir-Tauman's "hash-sign-switch" paradigm [22].

2.1 Chameleon Hash Family

Definition 1. *(chameleon hash family) A chameleon hash family consists of a pair* $(\mathcal{I}, \mathcal{H})$*:*

- $\mathcal{I}$ *is a probabilistic polynomial-time key generation algorithm that on input* 1^k*, outputs a pair* (HK, TK) *such that the sizes of* HK, TK *are polynomially related to* k*.*
- $\mathcal{H}$ *is a family of randomized hash functions. Every hash function in* $\mathcal{H}$ *is associated with a hash key* HK*, and is applied to a message from a space* $\mathcal{M}$ *and a random element from a finite space* $\mathcal{R}$*. The output of the hash function* H_{HK} *does not depend on* TK*.*

A chameleon hash family $(\mathcal{I}, \mathcal{H})$ has the following properties:

1. *Efficiency:* Given a hash key HK and a pair $(m, r) \in \mathcal{M} \times \mathcal{R}$, $H_{HK}(m, r)$ is computable in polynomial time.
2. *Collision resistance:* There is no probabilistic polynomial time algorithm $\mathcal{A}$ that on input HK outputs, with a probability which is not negligible, two pairs $(m_1, r_1), (m_2, r_2) \in \mathcal{M} \times \mathcal{R}$ that satisfy $m_1 \neq m_2$ and $H_{HK}(m_1, r_1) = H_{HK}(m_2, r_2)$ (the probability is over HK, where $(HK, TK) \leftarrow \mathcal{I}(1^k)$, and over the random coin tosses of algorithm $\mathcal{A}$).
3. *Trapdoor collisions:* There exists a probabilistic polynomial time algorithm that given a pair $(HK, TK) \leftarrow \mathcal{I}(1^k)$, a pair $(m_1, r_1) \in \mathcal{M} \times \mathcal{R}$, and an additional message $m_2 \in \mathcal{M}$, outputs a value $r_2 \in \mathcal{R}$ such that:
   - $H_{HK}(m_1, r_1) = H_{HK}(m_2, r_2)$.
   - If r_1 is uniformly distributed in $\mathcal{R}$ then the distribution of r_2 is computationally indistinguishable from uniform in $\mathcal{R}$.

2.2 Shamir-Tauman's "Hash-Sign-Switch" Paradigm

Shamir and Tauman introduced the following"hash-sign-switch" paradigm to get a generic on-line/off-line signature scheme.

- **System Parameters Generation:** Let $(\mathcal{I}, \mathcal{H})$ be any trapdoor hash family and $(\mathcal{G}, \mathcal{S}, \mathcal{V})$ be any provably secure signature scheme. The system parameters are $SP = \{(\mathcal{I}, \mathcal{H}), (\mathcal{G}, \mathcal{S}, \mathcal{V})\}$.
- **Key Generation Algorithm:**
  - On input 1^k, run the key generation algorithm of the original signature scheme $\mathcal{G}$ to obtain a signing/verification key pair (SK, VK).
  - On input 1^k, run the key generation algorithm of the trapdoor hash family $(\mathcal{I}, \mathcal{H})$ to obtain a hash/trapdoor key pair (HK, TK).

  The signing key is (SK, TK) and the verification key is (VK, HK).
- **The Signing Algorithm:**
  1. Off-line phase
     - Choose at random $(m_i, r_i) \in_R \mathcal{M} \times \mathcal{R}$, and compute the chameleon hash value $h_i = H_{HK}(m_i, r_i)$.
     - Run the signing algorithm $\mathcal{S}$ with the signing key SK to sign the message h_i. Let the output be $\sigma_i = \mathcal{S}_{SK}(h_i)$.
     - Store the pair (m_i, r_i), and the signature σ_i.
  2. On-line phase
     - For a given message m, retrieve from the memory a random pair (m_i, r_i) and the signature σ_i.
     - Compute $r \in \mathcal{R}$ such that $H_{HK}(m, r) = H_{HK}(m_i, r_i)$.
     - Send (r, σ_i) as the signature of the message m.
- **The Verification Algorithm:**
  - Compute $h_i = H_{HK}(m, r)$.
  - Verify that σ_i is indeed a signature of the hash value h_i with respect to the verification key VK.

In the following, we present Shamir-Tauman's "hash-sign-switch" paradigm with elliptic curve analogue of the chameleon hash family based on the discrete logarithm assumption [16,22], so that we can fairly compare it with our proposed signature scheme.

- **System Parameters Generation:** Let t be a prime power, and $E(\mathbb{F}_t)$ an elliptic curve over finite field $\mathbb{F}_t$. Let $\#E(\mathbb{F}_t)$ be the number of points of $E(\mathbb{F}_t)$, and P be a point of $E(\mathbb{F}_t)$ with prime order q where $q|\#E(\mathbb{F}_t)$. Denote $\mathbb{G}$ the subgroup generated by P. Let $(\mathcal{I}, \mathcal{H})$ be the trapdoor hash family based on the discrete logarithm assumption and $(\mathcal{G}, \mathcal{S}, \mathcal{V})$ be any provably secure signature scheme. The system parameters are $SP = \{E, t, q, P, \mathbb{G}, (\mathcal{G}, \mathcal{S}, \mathcal{V})\}$.
- **Key Generation Algorithm:**
  - On input 1^k, run the key generation algorithm of the original signature scheme $\mathcal{G}$ to obtain the signing/verification key pair (SK, VK).

- On input 1^k, run the key generation algorithm of the trapdoor hash family $(\mathcal{I}, \mathcal{H})$ to obtain the hash/trapdoor key pair $(Y = xP, x)$.

The signing key is (SK, x) and the verification key is (VK, Y).[2]

– **The Signing Algorithm:**

1. Off-line phase
   - Choose at random $(m_i, r_i) \in_R \mathcal{M} \times \mathcal{R}$, and computes the chameleon hash value $h_i = H_Y(m_i, r_i) = m_i P + r_i Y$.
   - Run the signing algorithm $\mathcal{S}$ with the signing key SK to sign the message h_i. Let the output be $\sigma_i = \mathcal{S}_{SK}(h_i)$.
   - Store the pair (m_i, r_i), and the signature σ_i.
2. On-line phase
   - For a given message m, retrieve from the memory x^{-1} and a random pair (m_i, r_i).
   - Compute $r = x^{-1}(m_i - m) + r_i \mod q$.
   - Send (r, σ_i) as the signature of the message m.

– **The Verification Algorithm:**

- Compute $h_i = H_Y(m, r) = mP + rY$.
- Verify that σ_i is indeed a signature of the hash value h_i with respect to the verification key VK.

3 A Double-Trapdoor Chameleon Hash Family

Chameleon hashing is very closely related to chameleon commitment schemes [4]. Gennaro [13] first introduced the notion of multi-trapdoor commitments. Ateniese and de Medeiros [2] observed that any stateless trapdoor commitment with two trapdoors may be adequate for designing a chameleon hash scheme without key exposure, which can be used to design a chameleon signature scheme. However, it seems that the current chameleon hash schemes without key exposure are not suitable for designing efficient on-line/off-line signature schemes. The reasons are twofold: Firstly, collision computation in these chameleon hash schemes usually requires the costly modular exponentiation operation. Secondly, though collision forgery will not reveal the signer's trapdoor information, it allows the verifier to compute other collisions for the same hash value.[3]

In this section, we first propose a new double-trapdoor chameleon hash family based on the discrete logarithm assumption as follows, which is a main ingredient for designing our efficient on-line/off-line signature scheme.

[2] The value of x^{-1} should be pre-computed and stored in order to decrease the computation cost in the on-line phase of the signature scheme.

[3] Note that this feature has some advantages in the chameleon signatures. For example, the signer can provide a different collision to hide the original signed message. While in the case of on-line/off-line signatures, it means that the verifier can universally forge a signature of the signer.

- **System Parameters Generation:** Let t be a prime power, and $E(\mathbb{F}_t)$ an elliptic curve over finite field $\mathbb{F}_t$. Let $\#E(\mathbb{F}_t)$ be the number of points of $E(\mathbb{F}_t)$, and P be a point of $E(\mathbb{F}_t)$ with prime order q where $q|\#E(\mathbb{F}_t)$. Denote by $\mathbb{G}$ the subgroup generated by P. Define a cryptographic secure hash function $f : \mathbb{Z}_q \times \mathbb{G} \to \mathbb{Z}_q$. Choose two random elements $k, x \in_R \mathbb{Z}_q^*$, and compute $K = kP, Y = xP$. The public hash key is $HK = (K, Y)$, and the private trapdoor key is $TK = (k, x)$.
- **The Hash Family:** Given the hash key HK, the proposed chameleon hash function $H_{HK} : \mathbb{Z}_q \times \mathbb{Z}_q \to \mathbb{G}$ is defined as follows:

$$H_{HK}(m, r) = f(m, K) \cdot K + rY.$$

Theorem 1. *The construction above is a chameleon hash family under the assumption of the discrete logarithm problem in G is intractable.*

Proof. We prove that the scheme satisfies the properties defined in Section 2.1.

1. *Efficiency:* Given the hash key HK and a pair $(m, r) \in \mathbb{Z}_q \times \mathbb{Z}_q$, $H_{HK}(m, r) = f(m, K) \cdot K + rY$ is computable in polynomial time.
2. *Collision resistance:* Assume to the contrary, that there exists a polynomial time algorithm $\mathcal{A}$ that on input HK outputs, with a probability which is not negligible, two pairs $(m_1, r_1), (m_2, r_2) \in \mathbb{Z}_q \times \mathbb{Z}_q$ that satisfy $m_1 \neq m_2$ and $H_{HK}(m_1, r_1) = H_{HK}(m_2, r_2)$. Then, we can use $\mathcal{A}$ to solve the discrete logarithm problem in $\mathbb{G}$ as follows: For a randomly given instance (P, aP), choose a random integer $b \in_R \mathbb{Z}_q$ and define $K = aP$, and $Y = bP$. Therefore, if
$$f(m_1, aP) \cdot aP + r_1Y = f(m_2, aP) \cdot aP + r_2Y,$$
we can compute $a = (f(m_1, aP) - f(m_2, aP))^{-1}(r_2 - r_1)b \mod q$.
3. *Trapdoor collisions:* Assume that we are given the hash and trapdoor key pair (HK, TK), a pair $(m_1, r_1) \in \mathbb{Z}_q \times \mathbb{Z}_q$, and an additional message $m_2 \in \mathbb{Z}_q$, we want to find $r_2 \in \mathbb{Z}_q$ such that
$$f(m_1, kP) \cdot kP + r_1Y = f(m_2, kP) \cdot kP + r_2Y.$$
The value of r_2 can be computed in polynomial time as follows:
$$r_2 = r_1 + kx^{-1}(f(m_1, kP) - f(m_2, kP)) \mod q.$$
Also, if r_1 is uniformly distributed in $\mathcal{R}$ then the distribution of r_2 is computationally indistinguishable from uniform in $\mathcal{R}$. □

4 Our Efficient On-Line/Off-Line Signature Scheme

In this section, we apply the "hash-sign switch" paradigm to propose a much more efficient on-line/off-line signature scheme. We can adopt any provably secure digital signature scheme to design our on-line/off-line signature scheme, so it is a general construction. The main idea is that the hash value and the corresponding signature in the signature scheme are always identical and can be viewed as the public key of the signer. Hence, it is not required to compute and store them in the off-line phase. However, if we directly use the proposed double-trapdoor chameleon hash function to design the on-line/off-line signature scheme, the key exposure problem still arises.

We introduce the idea of *long-term* trapdoor and *one-time* trapdoor in our chameleon hash family. The *one-time* trapdoor is used **only once** for each message signing in the on-line phase, which prevents the recipient from recovering the trapdoor information of the signer and computing other collisions. The *long-term* trapdoor can be used repeatedly during its life span.

The proposed on-line/off-line signature scheme consists of the following efficient algorithms:

– **System Parameters Generation:** Let t be a prime power, and $E(\mathbb{F}_t)$ an elliptic curve over finite field $\mathbb{F}_t$. Let $\#E(\mathbb{F}_t)$ be the number of points of $E(\mathbb{F}_t)$, and P be a point of $E(\mathbb{F}_t)$ with prime order q where $q|\#E(\mathbb{F}_t)$. Denote $\mathbb{G}$ the subgroup generated by P. Define a cryptographic secure hash function $f : \mathbb{Z}_q \times \mathbb{G} \rightarrow \mathbb{Z}_q$. Given a hash key $HK = (K, Y)$, the chameleon hash function $H_{HK} : \mathbb{Z}_q \times \mathbb{Z}_q \rightarrow \mathbb{G}$ is defined as follows:

$$H_{HK}(m, r) = f(m, K) \cdot K + rY.$$

Let $(\mathcal{G}, \mathcal{S}, \mathcal{V})$ be any provably secure signature scheme. The system parameters are $SP = \{E, t, q, P, \mathbb{G}, f, H_{HK}, (\mathcal{G}, \mathcal{S}, \mathcal{V})\}$.

– **Key Generation Algorithm:**
- On input 1^k, run the key generation algorithm of the original signature scheme $\mathcal{G}$ to obtain the signing/verification key pair (SK, VK).
- On input 1^k, run the key generation algorithm of the trapdoor hash family to obtain the *long-term* hash/trapdoor key pair, denote by $HK = Y = xP, TK = x$.
- Choose at random $k^* \in_R \mathbb{Z}_q$, and compute the chameleon hash value $h = k^*Y$. Run the signing algorithm $\mathcal{S}$ with the signing key SK to sign the message h. Let the output be $\sigma = \mathcal{S}_{SK}(h)$.

The signing key is (SK, x, k^*) and the verification key is (VK, Y, σ).

– **The Signing Algorithm:**

1. Off-line phase
   - Choose at random $k_i \in_R \mathbb{Z}_q$, and computes $k_i x^{-1} \mod q$ and $k_i P$.
   - Store the *one-time* trapdoor/hash key pair $(k_i x^{-1}, k_i P)$.
2. On-line phase
   - For a given signed message m_i, retrieve from the memory a random pair $(k_i x^{-1}, k_i P)$.
   - Compute $r_i = k^* - f(m_i, k_i P) k_i x^{-1} \mod q$.
   - Send $(r_i, k_i P)$ as the signature of the message m_i.

- **The Verification Algorithm:**
  - Compute $h = f(m_i, k_iP)k_iP + r_iY$ by using the *one-time* hash key k_iP and the *long-term* hash key Y.
  - Verify that σ is indeed a signature of the hash value h with respect to the verification key VK.

Note that

$$\begin{aligned} h = & \; f(m_i, k_iP)k_iP + r_iY \\ = & \; f(m_i, k_iP)k_iP + (k^* - f(m_i, k_iP)k_ix^{-1})Y \\ = & \; k^*Y \end{aligned}$$

The proposed scheme satisfies the property of completeness.

Remark 1. We argue that the value of x^{-1} should be pre-computed and stored in both our scheme and Shamir-Tauman's scheme.

Note that $r_i = k^* - f(m_i, k_iP)k_ix^{-1} \mod q$, it also requires only 1 modular multiplication of $\mathbb{Z}_q$ in the on-line phase of our scheme since k_ix^{-1} is stored in the off-line phase.

Remark 2. Note that in our proposed on-line/off-line signature scheme, the hash key $K_i = k_iP$ is used only **once** for signing a message m_i, while the other hash key $Y = xP$ can be used repeatedly. This is why we named them the *one-time* hash key and the *long-term* hash key, respectively.

5 Analysis of the Proposed Schemes

5.1 Security

The most general known security notion of a signature scheme is security against existential forgery on adaptively chosen message attacks, which was firstly defined by Goldwasser, Micali and Rivest [14] as follows:

Definition 2. *A signature scheme $\Omega = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ is existentially unforgeable under adaptive chosen message attacks if for any probabilistic polynomial time adversary $\mathcal{A}$ there exist no non-negligible probability ϵ such that*

$$\boldsymbol{Adv}(\mathcal{A}) = \Pr \left[\begin{array}{l} \langle pk, sk \rangle \leftarrow \mathsf{Gen}(1^l); \\ for \; i = 1, 2, \ldots, k; \\ m_i \leftarrow \mathcal{A}(pk, m_1, \sigma_1, \ldots, m_{i-1}, \sigma_{i-1}), \sigma_i \leftarrow \mathsf{Sign}(sk, m_i); \\ \langle m, \sigma \rangle \leftarrow \mathcal{A}(pk, m_1, \sigma_1, \ldots, m_k, \sigma_k); \\ m \notin \{m_1, \ldots, m_k\} \wedge \mathsf{Ver}(pk, m, \sigma) = accept \end{array} \right] \geq \epsilon.$$

Now we give the formal security proof of our on-line/off-line signature scheme. More precisely, we have the following theorem:

Theorem 2. *In the random oracle model, the resulting on-line/off-line signature scheme is existentially unforgeable against adaptive chosen message attacks, provided that the discrete logarithm problem in G is intractable.*

Proof. In our proposed on-line/off-line signature scheme, the corresponding signature σ on the chameleon hash value h is viewed as the public key of the signer. Therefore, a hash collision r and a one-time hash key kP are the real signature on the message m.

Suppose that $\mathcal{A}$ is a probabilistic algorithm that given a verification key (VK, HK, σ), forges a signature with respect to the proposed on-line/off-line signature scheme by an adaptively chosen message attack in time T with success probability ϵ. We denote respectively by q_H and q_S the number of queries that $\mathcal{A}$ can at most ask to the hash oracle and the signing oracle. Let $(m_i, K_i = k_iP)$ denote the input of i-th query to the hash oracle, and e_i denote the corresponding answer to it. Let m_j denote the j-th query to the signing oracle, and (r'_j, K'_j) denote the corresponding signatures produced by the signing oracle. Let (m, r, kP) denote the output of $\mathcal{A}$. Since the success probability of $\mathcal{A}$ is ϵ, it follows that

$$Pr[V_{VK}(h, \sigma) = 1 \wedge h = H_{HK,kP}(m, r) = H_{HK,k_iP}(m_i, r_i)] \geq \epsilon.$$

Then we can construct a probabilistic algorithm $\mathcal{M}$ to compute a for a randomly given instance (P, aP) where P is a generator of $\mathbb{G}$ as follows:

- Let (SK, VK) be the signing/verification key pair of the original signature scheme. Choose a random integer $b \in_R \mathbb{Z}_q$, and let $HK = Y = bP$. Define the chameleon hash value $h = b \cdot aP$. Run the signing algorithm S with the signing key SK to sign the message h. Let the output be $\sigma = S_{SK}(h)$. Publish the pair (VK, Y, σ).
- Maintain a list, called f-list, which is initially set to empty. If the i-th query (m_i, K_i) to the hash oracle f is not in the list, choose a random element $e_i \in_R \mathbb{Z}_q$ and respond it as the answer of i-th query. Then add (m_i, K_i, e_i) to the f-list.
- Let m_j denote the input of j-th query to the signing oracle, choose at random $(e'_j, r'_j) \in_R \mathbb{Z}_q \times \mathbb{Z}_q$ (Note that e'_j is not in the f-list) and define

  $$K'_j = e'^{-1}_j(h - r'_jY),$$

  respond e'_j as the hash oracle answer to the query (m_j, K'_j), and (K'_j, r'_j) as the signing oracle answer to the query m_j. Then add (m_j, K'_j, e'_j) to the f-list.

Suppose the output of $\mathcal{A}$ is (m, K, r). If $m \neq m_j$ for $j = 1, ..., q_S$ and $h = f(m, K)K + rY$, we say that $\mathcal{A}$ forges a signature (K, r) on the message m with respect to the proposed on-line/off-line signature scheme.

By replays of $\mathcal{A}$ with the same random tape but different choices of oracle f, as done in the Forking Lemma [20], we can obtain two valid signatures (m, K, r) and (m, K, r') with respect to different hash oracles f and f'.

Note that $h = f(m, K)K + rY$ and $h = f'(m, K)K + r'Y$, we can compute $a = (f'(m, K) - f(m, K))^{-1}(f'(m, K)r - f(m, K)r')$ as the discrete logarithm of aP with respect to the base P.

The success probability of $\mathcal{M}$ is also ϵ, and the running time of $\mathcal{M}$ is equal to the running time of the Forking Lemma which is bounded by $23Tq_R/\epsilon$ [20]. □

5.2 Efficiency

We compare the efficiency of our scheme with that of Shamir-Tauman's scheme given in Section 2.2. We denote by $C(\theta)$ the computation cost of operation θ, and by $|\lambda|$ the bits of λ. Also, we denote by M a scalar multiplication in $\mathbb{G}$, by SM a simultaneous scalar multiplication of the form $\lambda P + \mu Q$ in $\mathbb{G}$, and by m the modular multiplication in $\mathbb{Z}_q$. We omit other operations such as point addition and hash in both schemes.

Table 1 and Table 2 present the comparison of the computation cost, the storage cost, and the communication cost for each message signing between Shamir-Tauman's scheme and our scheme.

Table 1. Comparison of the computation cost

	Shamir-Tauman's scheme	Our scheme
Off-line phase	$1C(h) + 1C(\sigma)$ $= 1SM + 1C(\sigma)$	$1C(kP) + 1C(kx^{-1})$ $= 1M + 1m$
On-line phase	$1m$	$1m$

Table 2. Comparison of the storage and communication cost

	Shamir-Tauman's scheme	Our scheme
Storage off-line phase	$2\|q\| + 1\|\sigma\|$	$1\|q\| + 1\|t\| + 1$
Communication on-line phase	$1\|q\| + 1\|\sigma\|$	$1\|q\| + 1\|t\| + 1$

Since a 160-bit ECC key offers more or less the same level of security as a 1024-bit RSA key [18], we let $|q|$=160 in the following. Currently, for any secure signature scheme, the signature length $|\sigma| \geq |t| + 1$ since $|t|$ is about 160 (In the optimal case, we can choose an elliptic curve $E(\mathbb{F}_t)$ such that $\#E(\mathbb{F}_t)$ is just a 160-bit prime q. From Hasse theorem, we know that $|t| = |\#E(\mathbb{F}_t)| = 160$). Therefore, the proposed scheme is much superior to Shamir-Tauman's scheme in the computation cost of off-line phase, storage cost and communication cost, while the computation cost in the on-line phase is same. So, we argue that our signature scheme is more suitable for smart-card applications where both the computation and storage resources are limited.

Remark 3. However, if we adopt other generic group such as the subgroup of $\mathbb{Z}_p^*$ to present our double-trapdoor chameleon hash family and on-line/off-line signature scheme, the communication cost for our on-line/off-line signature scheme is $1|q| + 1|p|$. For most current signature schemes, the signature length $|\sigma| < |p|$ if we let $|p| = 1024$. So, our proposed on-line/off-line signature scheme is inferior to Shamir-Tauman's scheme in communication cost since $1|q|+1|p| > 1|q|+1|\sigma|$. This is the reason why we choose the elliptic curve cryptosystems.

6 Conclusions

On-line/off-line signatures are particularly useful in smart card applications. In this paper, we first introduce a special double-trapdoor chameleon hash family based on the discrete logarithm assumption and then apply the "hash-sign-switch" paradigm to propose a much more efficient generic on-line/off-line signature scheme. Compared with Shamir-Tauman's signature scheme, the advantages of our signature scheme are the lower computation and storage cost for the off-line phase, and the lower communication cost for the on-line phase.

Acknowledgement

We are grateful to the anonymous referees for their invaluable suggestions. This work is supported by National Natural Science Foundation of China (No. 60503006, 60403007 and 60633030), Natural Science Foundation of Guangdong, China (No. 04205407), 973 Program (2006CB303104) and ARC Discovery Grant DP0663306.

References

1. G. Ateniese and B. de Medeiros, *Identity-based chameleon hash and applications*, Finacial Cryptography and Data Security-FC 2004, LNCS 3110, pp.164-180, Springer-Verlag, 2004.
2. G. Ateniese and B. de Medeiros, *On the key-exposure problem in chameleon hashes*, Proceeding of the 4th Conference on Security in Communication Networks-SCN 2004, LNCS 3352, pp.165-179, Springer-Verlag, 2005.
3. D. Boneh and X. Boyen, *Short signatures without random oracles*, Advances in Cryptology-Eurocrypt 2004, LNCS 3027, pp.56-73, pringer-Verlag, 2004.
4. G. Brassard, D. Chaum, and C. Crepeau, *Minimum disclosure proofs of knowledge*, Journal of Computer and System Sciences, 37(2), pp.156-189, 1988.
5. D. Boneh, B. Lynn, and H. Shacham, *Short signatures from the Weil pairings*, Advances in Cryptology-Asiacrypt 2001, LNCS 2248, pp.514-532, Springer-Verlag, 2001.
6. C. Crutchfield, D. Molnar, D. Turner, and D. Wagner, *Generic on-line/off-line threshold signatures*, Proceeding of the 9th International Conference on Theory and Practice in Public-Key Cryptography-PKC 2006, LNCS 3958, pp.58-74, Springer-Verlag, 2006.
7. D. Chaum and H. van Antwerpen, *Undeniable signatures*, Advances in Cryptology-Crypto 1989, LNCS 435, pp.212-216, Springer-Verlag, 1989.
8. X. Chen, F. Zhang, and K. Kim, *Chameleon hashing without key exposure*, Proceeding of the 7th International Information Security Conference-ISC 2004, LNCS 3225, pp.87-98, Springer-Verlag, 2004.
9. T. ElGamal, *A public-key cryptosystem and a signature scheme based on discrete logarithms*, IEEE Transactions on Information Theory, 31(4), pp.469-472, 1985.
10. S. Even, O. Goldreich, and S. Micali, *On-line/Off-line digital signatures*, Advances in Cryptology-Crypto 1989, LNCS 2442, pp.263-277, Springer-Verlag, 1989.

11. S. Even, O. Goldreich, and S. Micali, *On-line/Off-line digital signatures*, Journal of Cryptology, 9(1), pp.35-67, Springer-Verlag, 1996.
12. A. Fiat and A. Shamir, *How to prove yourself: Practical solutions to identification and signature problems*, Advances in Cryptology-Crypto 1986, LNCS 263, pp. 186-194, Springer-Verlag, 1986.
13. R. Gennaro, *Multi-trapdoor commitments and their applications to proofs of knowledge secure under concurrent man-in-the-middle attacks*, Advances in Cryptology-Crypto 2004, LNCS 3152, pp.220-236, Springer-Verlag, 2004.
14. S. Goldwasser, S. Micali, and R. Rivest, *A digital signature scheme secure against adaptive chosen-message attacks*, SIAM Journal on Computing, 17(2), pp.281-308, 1988.
15. N. Koblitz, *Elliptic curve cryptosystems*, Mathematics of Computation, 48(177), pp.203-209, 1987.
16. H. Krawczyk and T. Rabin, *Chameleon hashing and signatures*, Proceeding of Network and Distributed System Security 2000, pp.143-154, 2000.
17. K. Kurosawa and K. Schmidt-Samoa, *New on-line/off-line signature schemes without random oracles*, Proceeding of the 9th International Conference on Theory and Practice in Public-Key Cryptography-PKC 2006, LNCS 3958, pp.330-346, Springer-Verlag, 2006.
18. A.K. Lenstra and E.R. Verheul, *Selecting cryptographic key sizes*, Journal of Cryptology, 14(4), pp.255-293, Springer-Verlag, 2001.
19. V. Miller, *Uses of elliptic curves in cryptography*, Advances in Cryptology-Crypto 1985, LNCS 218, pp.417-426, Springer-Verlag, 1986.
20. D. Pointcheval and J. Stern, *Security arguments for digital signatures and blind signatures*, Journal of Cryptography, 13(3), pp.361-396, Springer-Verlag, 2000.
21. C. P. Schnorr, *Efficient signature generation for smart cards*, Journal of Cryptology, 4(3), pp.239-252, Springer-Verlag, 1991.
22. A. Shamir and Y. Tauman, *Improved online/offline signature schemes*, Advances in Cryptology-Crypto 2001, LNCS 2139, pp.355-367, Springer-Verlag, 2001.