Advance Access publication on May 11, 2009
doi:10.1093/comjnl/bxp044

Efficient On-line/Off-line Signature Schemes Based on Multiple-Collision Trapdoor Hash Families

LEIN HARN[1], WEN-JUNG HSIN[2] AND CHANGLU LIN[3,4,*]

[1]*Department of Computer Science and Electrical Engineering, University of Missouri-Kansas City, Kansas City MO 64110, USA*
[2]*Department of Information and Computer Science, Park University, Parkville, MO 64152, USA*
[3]*State Key Laboratory of Information Security, Graduate University of Chinese Academy of Sciences, Beijing 10049, P.R. China*
[4]*Key Laboratory of Network Security and Cryptology, Fujian Normal University, Fujian, 35007, P.R. China*
[*]*Corresponding author: lincl@is.ac.cn*

The first on-line/off-line signature scheme introduced by Even *et al.* in 1990 has two problems: (a) impractical signature length and (b) a one-time use of signature generated during the off-line phase. In 2001, Shamir and Tauman significantly shortened the length of the signature by using trapdoor hash families introduced by Krawczyk and Rabin in 2000. However, each trapdoor hash value and its signature in the off-line phase of Shamir and Tauman's signature scheme can be used for signing only one message in the on-line phase. In this paper, we propose *multiple-collision trapdoor hash families* based on discrete logarithm and factoring assumptions, and provide formal proofs of their security. We also introduce an efficient on-line/off-line signature scheme based on our proposed trapdoor hash families. Our on-line/off-line signature scheme can re-use a trapdoor hash value for signing multiple messages. If a signer includes this trapdoor hash value in the public-key digital certificate, there is no need to have any regular digital signature scheme to sign the trapdoor hash value in the off-line phase.

Keywords: public-key digital certificate; on-line/off-line signature; discrete logarithm; integer factoring trapdoor hash family

Received 21 November 2008; revised 8 April 2009
Handling editor: Yannis Manolopoulos

1. INTRODUCTION

In 1990, Even *et al.* [1] introduced the first on-line/off-line signature scheme in which the signing of a message is accomplished in two phases, namely, *off-line* and *on-line*. In their scheme, a signer performs moderate off-line computation and performs fast on-line computation when a message is ready. The on-line/off-line signature is very useful for applications such as smart cards and mobile devices in which on-line resources are limited. Nonetheless, there are a few drawbacks in the scheme of Even *et al.* First, the length of the signature in their scheme is not practical as it is increased by a quadratic factor with the length of the message [1, 2]. Second, a signature generated in the off-line phase can be used for no more than one message in the on-line phase (the so-called *one-time signature* will be defined later in Section 4.2). In 2001, Shamir and Tauman [2] proposed an improved on-line/off-line signature scheme based on (a) trapdoor hash families introduced by Krawczyk and Rabin [3] and (b) a regular digital signature algorithm, to significantly shorten the length of the signature. Krawczyk and Rabin [3] used trapdoor hash families to construct a *chameleon signature*, which is a signature verifiable only by an intended party and not by any other party. However, the trapdoor hash value and its signature in the off-line phase in Shamir and Tauman's signature scheme can only be used for one message in the on-line phase, as multiple uses of the same trapdoor hash value in different messages will lead to the disclosure of the secret key. We call this type of trapdoor hash families one-time collision trapdoor hash families. Chen *et al.* [4, 5] proposed a special double-trapdoor hash family to overcome the above problem. Their scheme is based on the elliptic curve for implementation. They have claimed that their

scheme is optimal with regard to the length of the signature. The use of elliptic curves in cryptography was suggested independently by Koblitz [6] and Miller [7] in 1985. It was commonly believed that a group with fewer elements can be used to obtain the same level of security as RSA-based systems. Based on this assumption, the key length of elliptic curve-based cryptographic algorithms can be much smaller than the key length of most RSA or discrete logarithm (DL)-based cryptographic algorithms. However, no mathematical proof of this assumption for elliptic curve-based cryptosystem has been published so far. Catalano *et al.* [8] unified Even *et al.*'s paradigm based on one-time signatures and Shamir–Tauman paradigm based on trapdoor hash functions, in the sense that they both use an ordinary signature scheme and a (weak) one-time signature scheme as components. Recent works in on-line/off-line signatures have been done in improving the efficiency [9], eliminating the random oracle model [10], constructing ID-based on-line/off-line signature schemes [11], constructing on-line/off-line threshold signature schemes [12, 13], avoiding key exposure [4] and avoiding trapdoor hash primitives [14].

In this paper, we propose *multiple-collision trapdoor hash families* based on DL and factoring assumptions. In multiple-collision trapdoor hash families, revealing multiple collisions of the same hash value will not leak the secret key of trapdoor functions. Our multiple-collision trapdoor hash families are similar to the ElGamal signature scheme [15] in which a pair of *long-term* and *one-time* private keys is used for generating an ElGamal digital signature. Based on our proposed trapdoor hash families, we introduce an efficient on-line/off-line signature scheme which can significantly improve the efficiency of the Shamir and Tauman's on-line/off-line signature. Specifically, in our signature scheme, a trapdoor hash value can be re-used for signing multiple messages. If a signer includes this trapdoor hash value in the public-key digital certificate, there is no need to sign the hash value in the off-line phase.

This paper is organized as follows. Section 2 reviews the trapdoor hash families. Section 3 describes our proposed trapdoor hash families. Section 4 reviews Shamir and Tauman's on-line/off-line signature scheme. Section 5 describes our on-line/off-line signature schemes. Section 6 provides a conclusion.

2. REVIEW OF TRAPDOOR HASH FAMILIES

A trapdoor hash family was introduced in [3] and formally defined in [2]. For reviewing purpose, we describe the definition in the following. For a comprehensive study on trapdoor related schemes, readers are referred to [16].

DEFINITION 2.1 (trapdoor hash family [2]). *A trapdoor hash family consists of a pair* $(\mathcal{I}, \mathcal{H})$*, where* $\mathcal{I}$ *is a probabilistic polynomial-time key generation algorithm and* $\mathcal{H}$ *is a family of randomized hash functions.* $\mathcal{I}$ *generates a pair* (HK, TK)*, where* HK *is a hash key and* TK *is its associated trapdoor key. A trapdoor hash function in* $\mathcal{H}$ *is a hash function with a trapdoor secret. It is denoted as* $h_{\mathrm{HK}}(m, s)$*, where* HK *is a public hash key,* TK *is a private trapdoor key,* m *is a message and* s *is an auxiliary random number.*

A trapdoor hash family $(\mathcal{I}, \mathcal{H})$ *satisfies the following three properties:*

(i) *Efficiency: Given* HK *and any tuple* (m, s)*, the hash value* $h_{\mathrm{HK}}(m, s)$ *can be computed efficiently* (*i.e. in polynomial time*).

(ii) *Trapdoor Collision: Based on a trapdoor one-way property, the entity with the knowledge of the trapdoor secret can generate collisions in polynomial time.*

(iii) *Collision Resistance: This property is from the perspective of an outsider* (*e.g. an entity without the knowledge of the trapdoor key*). *Without knowing the trapdoor secret, it is computationally infeasible for an outsider to generate any two arbitrary points* (m_1, s_1) *and* (m_2, s_2) *such that these two points collide, i.e.* $h_{\mathrm{HK}}(m_1, s_1) = h_{\mathrm{HK}}(m_2, s_2)$ *where* $m_1 \neq m_2$.

In the following subsections, we first review Krawczyk and Rabin's DL-based trapdoor hash family. We then review Shamir and Tauman's factoring-based trapdoor hash family.

2.1. Krawczyk and Rabin's DL-based trapdoor hash family

Krawczyk and Rabin [3] described the trapdoor hash function in $\mathcal{H}$ based on DL assumption as follows. Randomly select a large prime q. Randomly choose a safe prime p (i.e. a prime p such that $q = (p-1)/2$ is prime) and an element g of order q. Choose a random element x and compute $y = g^x \pmod p$. The public hash key is $\mathrm{HK} = (p, g, y)$ and the private trapdoor key is $\mathrm{TK} = x$. The trapdoor hash function $h_{\mathrm{HK}}(m, s)$ is defined as follows:

$$h_{\mathrm{HK}}(m, s) \stackrel{\mathrm{def}}{=} g^m y^s \pmod p.$$

To show that the $h_{\mathrm{HK}}(m, s)$ is a trapdoor hash function under the DL assumption, one needs to show that it fulfills three main properties of a trapdoor hash function, i.e. efficiency, trapdoor collision and collision resistance. A lemma and its formal proof asserting that $h_{\mathrm{HK}}(m, s)$ is a DL-based trapdoor hash family can be found in [2].

2.2. Shamir and Tauman's factoring-based trapdoor hash family

Shamir and Tauman [2] described the trapdoor hash function in $\mathcal{H}$ based on factoring assumption as follows. Randomly choose two safe primes p and q (i.e. primes such that $p' = (p-1)/2$ and $q' = (q-1)/2$ are primes) and compute $n = pq$. Randomly choose an element g of order $\lambda(n)$, where $\lambda(n) = \mathrm{lcm}(p-1, q-1) = 2p'q'$. The public hash key is $\mathrm{HK} = (n, g)$

and the private trapdoor key is TK $= (p, q)$. The trapdoor hash function $h_{\mathrm{HK}}(m, s)$ is defined as follows:

$$h_{\mathrm{HK}}(m, s) \stackrel{\mathrm{def}}{=} g^{m\|s} \pmod{n},$$

where '$\|$' denotes concatenation. A lemma and its formal proof asserting that $h_{\mathrm{HK}}(m, s)$ is a factoring-based trapdoor hash family can be found in [2].

3. OUR MULTIPLE-COLLISION TRAPDOOR HASH FAMILIES

In Krawczyk and Rabin's DL-based trapdoor hash family, reviewed in Section 2.1, there is only one secret element in TK, i.e. x. Therefore, there is only one associated public value y in HK. Shamir and Tauman [2] utilized this trapdoor hash family to construct a one-time on-line/off-line signature scheme (see Section 4.2 for the reason why their signature is one-time).

To efficiently use a trapdoor hash value for signing multiple messages, we propose to include an additional one-time secret in the trapdoor hash function. This is similar to ElGamal's signature scheme [15] in which a one-time secret key was introduced for signing each message. That is, we introduce a new parameter r for our proposed trapdoor hash function, and denote $h_{\mathrm{HK}}(m, r, s)$ as a trapdoor hash function in $\mathcal{H}$. In Section 5.1, we will use our modified trapdoor hash families in an on-line/off-line signature scheme so that a signer can use the same trapdoor hash value to sign multiple messages.

In the following subsections, we first introduce a new trapdoor hash family based on the DL assumption, and then introduce a new trapdoor hash family based on the Factoring assumption. In both cases, we also present formal security proofs of our proposed trapdoor hash families.

3.1. DL-based multiple-collision trapdoor hash family

The original ElGamal signature scheme [15] was proposed in 1985; but its security was never proved equivalent to the DL problem. In 1996, Pointcheval and Stern [17] used the Forking Lemma to prove the security of a slight variant of the original ElGamal signature scheme, called *modified ElGamal signature scheme*. Later in this section, we shall show that the modified ElGamal signature scheme is a special function of our proposed trapdoor hash function. In this section, we first review the modified ElGamal signature scheme. We then introduce a new trapdoor hash family based on the DL and present a formal security proof of our proposed scheme.

Modified ElGamal signature scheme consists of three steps as follows:

- Let p be a random large prime and g be a generator of Z_p^*; then the public key is $y = g^x \pmod{p}$ and the secret key is x.
- Picks $k \in Z_{p-1}$ randomly and a cryptographic hash function f, the signature of message m is (r, s), where $r = g^k \pmod{p}$ and solves the linear equation $f(m, r) = xr + ks \pmod{p-1}$.
- The verification of the signature checks the equation $g^{f(m,r)} = y^r r^s \pmod{p}$.

The following paragraph describes the DL-based multiple-collision trapdoor hash function. Choose at random a safe prime p (i.e. a prime p such that $q = (p-1)/2$ is prime) and an element g of order q. Choose random elements $x, k, s \in Z_q$, and compute $y = g^x \pmod{p}$ and $r = g^k \pmod{p}$. The public hash key is HK $= (p, g, y)$ and the private trapdoor key is TK $= (x, k)$. The randomized hash function in $\mathcal{H}$ is defined as follows:

$$h_{\mathrm{HK}}(m, r, s) \stackrel{\mathrm{def}}{=} g^{-f(m,r)} y^r r^s \pmod{p},$$

where $f : \{0, 1\}^* \times Z_p \to Z_q$ is a cryptographic hash function. The auxiliary parameter r does not depend on message m and can be computed off-line. The computation and characteristics of r are the same as those in all DL-based cryptographic algorithms.

Remark 1. We recall that the verification equation of the modified ElGamal signature scheme is $g^f = y^r r^s \pmod{p}$, where $r = g^k$ and $f = f(m, r)$. This equation can be modified into a form as $h_{\mathrm{HK}}(m, r, s) = 1 = g^{-f} y^r r^s \pmod{p}$. Thus, the modified ElGamal signature is just a special trapdoor hash function in our proposed trapdoor hash family.

Remark 2. There are three reasons why we use function f to hash message m and commitment r: (1) Efficiency—for any message with an arbitrary length, the length of the output of function f is fixed, (2) Security—similar to using function f in any digital signature scheme to prevent existential forgery, our proposed trapdoor hash function can easily and effectively prevent collisions when using the hash function f, (3) Security proof—we assume that hash function f behaves like a random oracle, and hence we follow the established cryptographic techniques, i.e. *the Oracle Replay Attack* and *the Forking Lemma* as proposed in [17], to prove the collision resistance of our proposed trapdoor hash family.

Remark 3. Later in this paper we shall show that the auxiliary number r must not be re-used in signing more than one message. Since $r = g^k$, where k is a random value from Z_q, it is probabilistically negligible to select the same k for different messages.

THEOREM 3.1. *The pair $(\mathcal{I}, \mathcal{H})$ is a trapdoor hash family under the DL assumption.*

Proof. We show that the new trapdoor hash family based on the DL satisfies the three properties: efficiency, trapdoor collision and collision resistance.

Efficiency: Given any (m, r, s) and the public hash key HK $= (p, g, y)$, the function $h_{\mathrm{HK}}(m, r, s) = g^{-f(m,r)} y^r r^s \pmod{p}$ is computable in polynomial time.

Trapdoor Collision: The following shows that calculating the trapdoor collision by the user with the knowledge of the trapdoor secret can be done in polynomial time. Given (m, r, s) and additional m' and r', where $r' = g^{k'} \pmod p$, the user wants to find s' such that

$$g^{-f} y^r r^s = g^{-f'} y^{r'} r'^{s'} \pmod p.$$

Since the user knows the trapdoor secret (x, k) and can also compute $r' = g^{k'} \pmod p$, where k' is randomly selected from Z_q, the value of s' can be calculated in polynomial time as follows:

$$s' = (k')^{-1}((f' - f) + x(r - r') + ks) \pmod q.$$

Collision Resistance: For a formal security proof, the hash function $f = f(m, r)$ in our proposed trapdoor hash function will be treated as a random oracle. Assume to the contrary that without knowing the trapdoor secret (x, k), given input HK $= (p, g, y)$, the adversary can generate a collision (m_1, f_1, r_1, s_1) such that $H = g^{-f_1} y^{r_1} r_1^{s_1} \pmod p$, where $f_1 = f(m_1, r_1)$, by making some oracle queries in probabilistic polynomial time. Then, based on the well-known cryptographic techniques, *the Oracle Replay Attack* and *the Forking Lemma* as proposed in [17], the adversary uses the oracle replay attack by a polynomial replay of the attack with the same random tape and a different oracle. Readers can refer to the original works in [17] for the detailed description. The adversary obtains two collisions of a special form as (m_1, f_1, r_1, s_1) and (m_2, f_2, r_2, s_2), where $(m_2, r_2) = (m_1, r_1)$ and $f_1 \neq f_2, s_1 \neq s_2$.

Since (f_1, r_1, s_1) and (f_2, r_1, s_2) are two pairs of collisions for H where $f_1 \neq f_2$, we obtain the following two equations:

$$H = g^{-f_1} y^{r_1} r_1^{s_1} \pmod p,$$
$$H = g^{-f_2} y^{r_1} r_1^{s_2} \pmod p.$$

Thus, we have $r_1^{s_2 - s_1} = g^{f_2 - f_1} \pmod p$. Since $s_2 - s_1 \neq 0$, it is easy to compute the DL of r_1 as

$$k_1 = (f_2 - f_1)(s_2 - s_1)^{-1} \pmod q.$$

Repeating the same procedure from the beginning, the adversary can generate a second collision (m_2, f_2, r_2, s_2) such that $H = g^{-f_2} y^{r_2} r_2^{s_2} \pmod p$, where $m_1 \neq m_2$ and $r_1 \neq r_2$. The adversary can also compute the DL of r_2. We assume that the DL of r_1 and r_2 are k_1 and k_2, respectively. Then, the adversary can establish the following equation as

$$-f_1 + xr_1 + k_1 s_1 = -f_2 + xr_2 + k_2 s_2 \pmod q.$$

The adversary can solve the DL of y as

$$x = (r_1 - r_2)^{-1}((f_1 - f_2) + (k_2 s_2 - k_1 s_1)) \pmod q.$$

This result contradicts the DL assumption. □

3.2. Factoring-based multiple-collision trapdoor hash family

This section describes another new trapdoor hash function in $\mathcal{H}$ based on factoring assumption. Choose at random two safe primes p and q (i.e. primes such that $p' = (p - 1)/2$ and $q' = (q - 1)/2$ are primes) and compute $n = pq$. Choose at random an element g of order $\lambda(n)$, where $\lambda(n) = \text{lcm}(p - 1, q - 1) = 2p'q'$. Choose a random element k and compute $r = g^k \pmod n$. The public hash key HK $= (n, g)$ and the private trapdoor key TK $= (p, q, k)$. The randomized hash function in $\mathcal{H}$ is defined as follows:

$$h_{\text{HK}}(m, r, s) \stackrel{\text{def}}{=} rg^{f(m,r)s} \pmod n,$$

where $f : \{0, 1\}^* \to Z_{\lambda(n)}$ is a cryptographic hash function.

THEOREM 3.2. *The pair $(\mathcal{I}, \mathcal{H})$ is a trapdoor hash family under the factoring assumption.*

Proof. We show that the proposed trapdoor hash family based on factoring assumption satisfies the three properties: efficiency, trapdoor collision and collision resistance.

Efficiency: Given any tuple (m, r, s) and the public hash key (n, g), the hash value $h_{\text{HK}}(m, r, s) = rg^{f(m,r)s} \pmod n$ can be computed in polynomial time.

Trapdoor Collision: The following shows that calculating the trapdoor collision (m', r', s') by the user with the knowledge of trapdoor secret TK $= (p, q, k)$ can be done in polynomial time. Given (m, r, s) and additional m' and r', where $r' = g^{k'} \pmod n$, the user wants to find s' such that

$$rg^{fs} = r'g^{f's'} \pmod n.$$

Since the user knows the trapdoor secret (p, q, k) and can compute $r' = g^{k'} \pmod n$, where k' is randomly selected, the value of s' can be calculated in polynomial time as follows:

$$s' = (f')^{-1}(fs + (k - k')) \pmod{\lambda(n)}.$$

Collision Resistance: For a formal security proof, the hash function $f = f(m, r)$ in our proposed trapdoor hash function will be treated as a random oracle. Assume to the contrary that without knowing the trapdoor secret (p, q, k), given input HK $= (n, g)$, the adversary can generate a collision (m_1, f_1, r_1, s_1) such that $H = r_1 g^{f_1 s_1} \pmod n$ where $f_1 = f(m_1, r_1)$, by making some oracle queries in probabilistic polynomial time. Then, based on the well-known cryptographic techniques, *the Oracle Replay Attack* and *the Forking Lemma* as proposed in [17], the adversary uses the oracle replay attack by a polynomial replay of the attack with the same random tape and a different oracle. The adversary obtains two collisions of a special form as (m_1, f_1, r_1, s_1) and (m_2, f_2, r_2, s_2), where $(m_1, r_1) = (m_2, r_2)$, $f_1 \neq f_2$ and $s_1 \neq s_2$.

Since (f_1, r_1, s_1) and (f_2, r_1, s_2) are collisions for H, we get the following two equations:

$$H = r_1 g^{f_1 s_1} \pmod{n},$$
$$H = r_1 g^{f_2 s_2} \pmod{n}.$$

Thus, we have $g^{k_1+f_1 s_1} = g^{k_1+f_2 s_2} \pmod{n}$ and set $x = f_1 s_1 - f_2 s_2$. Since $f_1 \neq f_2$ and $s_1 \neq s_2$, the probability that $x = 0$ is negligible. Therefore, $\lambda(n)$ divides x. Thus, $\phi(n)$ divides $2x$. In essence, there exists a probabilistic polynomial time algorithm such that given input HK, it outputs a multiple of $\phi(n)$. In [18], Miller shows that the factorization of n can be computed from any multiple of $\phi(n)$. Therefore, this contradicts the factoring assumption. □

4. REVIEW OF SHAMIR AND TAUMAN'S SCHEME

In [2], Shamir and Tauman introduced a *Hash–Sign–Switch* paradigm in which any regular digital signature scheme (such as DSA [19] or RSA [20]) combined with a trapdoor hash family in $(\mathcal{I}, \mathcal{H})$ can be converted into a on-line/off-line signature scheme. Basically, in the off-line phase, a signer generates a hash value to commit to an arbitrarily selected message. In the on-line phase, given a message, the signer finds a collision of the trapdoor hash to the previously calculated hash value. The collision point and the signature generated in the off-line phase can be used as the signature for message generated in the on-line phase. In Shamir and Tauman's efficiency analysis [2], they showed that their scheme is efficient in that the computational load in the on-line phase is about 0.1 modular multiplication, and the signature size increases only by a factor of two, instead of a quadratic factor as in the approach of Even *et al.* [1].

In Section 4.1, we describe Shamir and Tauman's on-line/off-line signature scheme in details. Section 4.2 describes an issue associated with their approach.

4.1. Approach

Shamir and Tauman's approach can combine any trapdoor hash family $(\mathcal{I}, \mathcal{H})$ and any regular digital signature scheme (GEN, SIGN, VERF) to generate an on-line/off-line signature scheme (GEN′, SIGN′, VERF′). The following gives an example of a signature scheme where the trapdoor hash family is based on the DL assumption.

Let $h_{\mathrm{HK}}(m, s) = g^m y^s \pmod{p}$, where $\mathrm{HK} = (y, g, p)$, $\mathrm{TK} = x$, $y = g^x \pmod{p}$, p is a safe prime, g is a generator of order q and q is a factor of $p - 1$. Denote a verification key by VK and a signing key by SK for any regular digital signature scheme.

- The Key Generation Algorithm GEN′: Generate (SK, VK) using a key generation algorithm GEN and a pair (TK, HK) using algorithm $\mathcal{I}$. The signing key is (SK, TK, HK) and the verification key is (VK, HK).
- The Signing Algorithm SIGN′: Given a signing key (SK, TK, HK), the signing algorithm operates as follows.
  - Off-line phase: The signer randomly picks a pair (m, s), computes hash value $h_{\mathrm{HK}}(m, s) = g^m y^s \pmod{p}$ and uses the private key SK to sign $h_{\mathrm{HK}}(m, s)$ to obtain $\mathrm{SIGN}_{\mathrm{SK}}(h_{\mathrm{HK}}(m, s))$. The signer stores m, s, and $\mathrm{SIGN}_{\mathrm{SK}}(h_{\mathrm{HK}}(m, s))$.
  - On-line phase: Given a message m', the signer finds a collision of the trapdoor hash such that $h_{\mathrm{HK}}(m', s') = h_{\mathrm{HK}}(m, s)$ by solving s' such that it satisfies $m + xs = m' + xs' \pmod{q}$. The signature of message m' is $\langle \mathrm{SIGN}_{\mathrm{SK}}(h_{\mathrm{HK}}(m, s)), s' \rangle$.
- The Verification Algorithm VERF′: First compute $h_{\mathrm{HK}}(m', s')$, and then verify $\mathrm{SIGN}_{\mathrm{SK}}(h_{\mathrm{HK}}(m, s))$ using VK and $h_{\mathrm{HK}}(m', s')$.

4.2. Issue associated with Shamir and Tauman's Scheme

Although the Shamir and Tauman's Hash-Sign-Switch signature scheme is secure, the main issue associated with the scheme resides in the inefficient use of $h_{\mathrm{HK}}(m, s)$ and the signature of $h_{\mathrm{HK}}(m, s)$. To clearly explain this issue, we first define the term *one-time signature scheme* and then show that Shamir and Tauman's on-line/off-line signature scheme is a one-time signature scheme.

DEFINITION 4.1 (one-time signature scheme). *One-time signature scheme is an on-line/off-line signature scheme in which a signature generated off-line can be presented for no more than one message in the on-line phase. The use of the same signature for multiple messages leads to signature forgery.*

The following shows that Shamir and Tauman's on-line/off-line signature scheme is a one-time signature scheme. During the off-line phase, since the signer is required to commit to the trapdoor hash value $h_{\mathrm{HK}}(m, s)$, he needs to sign $h_{\mathrm{HK}}(m, s)$ using any regular digital signature scheme. For ease of reference, denote the signature of $h_{\mathrm{HK}}(m, s)$ as $\mathrm{SIGN}_{\mathrm{SK}}(h_{\mathrm{HK}}(m, s))$. Each signature $\mathrm{SIGN}_{\mathrm{SK}}(h_{\mathrm{HK}}(m, s))$ can be used for exactly one message as multiple uses of the same signature for different messages will lead to the disclosure of the trapdoor key TK. Specifically, suppose that $\mathrm{SIGN}_{\mathrm{SK}}(h_{\mathrm{HK}}(m, s))$ is used for two different messages m_1 with corresponding s_1 and m_2 with corresponding s_2. That is, $h_{\mathrm{HK}}(m, s) = h_{\mathrm{HK}}(m_1, s_1)$, and $h_{\mathrm{HK}}(m, s) = h_{\mathrm{HK}}(m_2, s_2)$, which lead to $h_{\mathrm{HK}}(m_1, s_1) = h_{\mathrm{HK}}(m_2, s_2)$. Therefore, $g^{m_1+xs_1} \pmod{p} = g^{m_2+xs_2} \pmod{p}$. As $m_1 + xs_1 = m_2 + xs_2 \pmod{q}$, one can solve x as $x = (m_1 - m_2)(s_2 - s_1)^{-1} \pmod{q}$. Thus, each hash value $h_{\mathrm{HK}}(m, s)$ and the corresponding signature $\mathrm{SIGN}_{\mathrm{SK}}(h_{\mathrm{HK}}(m, s))$ can be used only once. Hence Shamir and Tauman's on-line/off-line signature is a one-time signature.

One-time signatures are not efficient in terms of the cost incurred for generating the hash value and the signature of the hash value.

5. ON-LINE/OFF-LINE SIGNATURE SCHEMES

5.1. On-line/off-line signature scheme based on Shamir and Tauman's scheme

To efficiently use a trapdoor hash value, we propose an efficient on-line/off-line signature scheme (GEN′, SIGN′, VERF′) based on the modified trapdoor hash families $(\mathcal{I}, \mathcal{H})$ proposed in Section 3 and any regular signature scheme (GEN, SIGN, VERF).

- The Key Generation Algorithm GEN′: Generate a pair (SK, VK) using a key generation algorithm GEN and a pair (TK, HK) using algorithm $\mathcal{I}$. The signing key is (SK, TK, HK) and the verification key is (VK, HK).
- The Signing Algorithm SIGN′:
  - Off-line phase: A signer randomly selects a message m and values r and s. He then computes $h_{\mathrm{HK}}(m, r, s)$ and uses SK to sign $h_{\mathrm{HK}}(m, r, s)$ to obtain $\mathrm{SIGN}_{\mathrm{SK}}(h_{\mathrm{HK}}(m, r, s))$. The signer stores (m, r, s) and $\mathrm{SIGN}_{\mathrm{SK}}(h_{\mathrm{HK}}(m, r, s))$. In addition, the signer generates and stores (k_i, r_i) by randomly selecting k_i and calculating r_i according to the selected $\mathcal{H}$. We should point out that each pair (k_i, r_i) can only be used for signing one message. This requirement is the same as in a modified ElGamal signature.
  - On-line phase: Given a message m_i, the signer uses (k_i, r_i) pre-computed in the off-line phase, and solves s_i such that $h_{\mathrm{HK}}(m, r, s) = h_{\mathrm{HK}}(m_i, r_i, s_i)$. The signature of message m_i is $\langle \mathrm{SIGN}_{\mathrm{SK}}(h_{\mathrm{HK}}(m, r, s)), r_i, s_i \rangle$.
- The Verification Algorithm VERF′: First compute $h_{\mathrm{HK}}(m_i, r_i, s_i)$, and then verify $\mathrm{SIGN}_{\mathrm{SK}}(h_{\mathrm{HK}}(m, r, s))$ using VK and $h_{\mathrm{HK}}(m_i, r_i, s_i)$.

We now analyze the security of the proposed on-line/off-line signature scheme. Our scheme is obtained using the general conversion technique as Shamir and Tauman's scheme [2]. Thus, we can easily get the following result (Theorem 5.1) for our scheme. Since the proof of this theorem is similar to the Shamir and Tanman's scheme, we omit the detailed description of the proof.

THEOREM 5.1. *Let* $(\mathcal{I}, \mathcal{H})$ *be a multiple-collision trapdoor hash family. Our proposed on-line/off-line signature scheme* (GEN′, SIGN′, VERF′) *based on* $(\mathcal{I}, \mathcal{H})$ *is secure against adaptive chosen message attacks if the regular digital signature scheme* (GEN, SIGN, VERF) *is secure against generic chosen message attacks.*

5.2. Modified on-line/off-line signature scheme

Our trapdoor hash value can be used for signing multiple messages. Thus, it can be included in a regular digital certificate to provide its authenticity. In essence, our proposed on-line/off-line signature scheme does not need any regular digital signature scheme to sign a hash value in the off-line phase. As we have pointed out in remark 1 that the modified ElGamal signature is a special function of our proposed trapdoor hash family, the modified ElGamal signature does not need any regular digital signature scheme to sign its hash value in the off-line phase. Based on this observation, we give a modified version for our scheme.

We now present the modified on-line/off-line signature scheme (GEN, SIGN, VERF) based on the multiple-collision trapdoor hash families $(\mathcal{I}, \mathcal{H})$ described in Section 3 as follows.

- The Key Generation Algorithm GEN: Generate a pair of HK and TK by applying $\mathcal{I}$. A signer randomly selects a message m and values r and s, and computes $h_{\mathrm{HK}}(m, r, s)$. As both HK and $h_{\mathrm{HK}}(m, r, s)$ need to be authenticated for verifying different signatures, these values can be included in the signer's regular public-key digital certificate for long-term use. In contrast with Shamir and Tauman's signature scheme, our $h_{\mathrm{HK}}(m, r, s)$ is computed only one time.
- The Signing Algorithm SIGN:
  - Off-line phase: The signer generates and stores (k_i, r_i) by randomly selecting k_i and calculating r_i according to the selected $\mathcal{H}$. Each pair (k_i, r_i) can only be used for signing one message. This requirement is the same as in the modified ElGamal signature.
  - On-line phase: Given a message m_i, the signer uses (k_i, r_i) pre-computed in the off-line phase, and solves s_i such that $h_{\mathrm{HK}}(m, r, s) = h_{\mathrm{HK}}(m_i, r_i, s_i)$. The signature of message m_i is (r_i, s_i). Note that HK and $h_{\mathrm{HK}}(m, r, s)$ are included in the signer's regular public-key digital certificate.
- The Verification Algorithm VERF: From the signer's public-key digital certificate, the verifier can obtain HK and $h_{\mathrm{HK}}(m, r, s)$. The verifier computes $h_{\mathrm{HK}}(m_i, r_i, s_i)$, and checks if $h_{\mathrm{HK}}(m_i, r_i, s_i) \stackrel{?}{=} h_{\mathrm{HK}}(m, r, s)$.

6. CONCLUSION

In this paper, multiple-collision trapdoor hash families under both DL and factoring assumptions are introduced and are secure under the random oracle model. We propose an efficient on-line/off-line signature scheme based on these multiple-collision trapdoor hash families. In our proposed scheme, as a trapdoor hash value can be used for signing multiple messages, the hash value can be included in the signer's public-key digital

certificate for long-term use, thereby eliminating the need for a regular digital signature for each message, signing in the off-line phase of Shamir and Tauman's on-line/off-line signature scheme.

REFERENCES

[1] Even, S., Goldreich, O. and Micali, S. (1990) On-line/Off-line Digital Signatures. *Proc. Crypto'89*, Santa Barbara, CA, USA, August 20–24, Lecture Notes in Computer Science, Vol. 435, pp. 263–277. Springer, Berlin.

[2] Shamir, A. and Tauman, Y. (2001) Improved On-line/Off-line Signature Schemes. *Proc. Crypto'01*, Santa Barbara, CA, USA, August 19–23, Lecture Notes in Computer Science, Vol. 2139, pp. 355–367. Springer, Berlin.

[3] Krawczyk, H. and Rabin, T. (2000) Chameleon Signature. *Proc. NDSS'00*, San Diego, CA, USA, February 3–4, pp. 143–154. The Internet Society.

[4] Chen, X., Zhang, F., Susilo, W. and Mu, Y. (2007) Efficient Generic On-Line/Off-Line Signatures Without Key Exposure. *Proc. ACNS'07*, Zhuhai, China, June 5–8, Lecture Notes in Computer Science, Vol. 4521, pp. 18–30. Springer, Berlin.

[5] Chen, X., Zhang, F., Tian, H., Wei, B., Susilo, W., Mu, Y., Lee, H. and Kim, K. (2008) Efficient generic on-line/off-line (threshold) signatures without key exposure. *Inform. Sci.*, **178**, 4192–4203.

[6] Koblitz, N. (1987) Elliptic curve cryptosystems. *Math. Comput.*, **48**, 203–209.

[7] Miller, V.S. (1986) Use of Elliptic Curves in Cryptography. *Proc. Crypto'85*, Santa Barbara, CA, USA, August 18–22, Lecture Notes in Computer Science, Vol. 218, pp. 417–426. Springer, Berlin.

[8] Catalano, D., Di Raimondo, M., Fiore, D. and Gennaro, R. (2008) Off-line/On-line Signatures: Theoretical Aspects and Experimental Results. *Proc. PKC'08*, Barcelona, Spain, March 9–12, Lecture Notes in Computer Science, Vol. 4939, pp. 101–120. Springer, Berlin.

[9] Schmidt-Samoa, K. and Takagi, T. (2005) Paillier's Cryptosystem Modulo p^2q and its Applications to Trapdoor Commitment Schemes. *Proc. Mycrypt'05*, Kuala Lumpur, Malaysia, September 28–30, Lecture Notes in Computer Science, Vol. 4939, pp. 296–313. Springer, Berlin.

[10] Kurosawa, K and Schmidt-Samoa, K. (2006) New Online/Offline Signature Schemes Without Random Oracles. *Proc. PKC'06*, New York, NY, USA, April 24–26, Lecture Notes in Computer Science, Vol. 3958, pp. 330–346. Springer, Berlin.

[11] Xu, S., Mu, Y. and Susilo, W. (2006) Online/Offline Signatures and multisignatures for AODV and DSR Routing Security. *Proc. ACISP*, Melbourne, Australia, July 3–5, Lecture Notes in Computer Science, Vol. 4058, pp. 99–110. Springer, Berlin.

[12] Bresson, E., Catalano, D. and Gennaro, R. (2007) Improved On-line/Off-line Threshold Signatures. *Proc. PKC'07*, Beijing, China, April 16–20, Lecture Notes in Computer Science, Vol. 4450, pp. 217–232. Springer, Berlin.

[13] Crutchfield, C., Molnar, D., Turner, D. and Wagner, D. (2006) Generic On-line/Off-line Threshold Signatures. *Proc. PKC'06*, New York, NY, USA, April 24–26, Lecture Notes in Computer Science, Vol. 3958, pp. 58–74, Springer, Berlin.

[14] Yu, P. and Tate, S.R. (2007) An Online/Offline Signature Scheme Based on the Strong RSA Assumption. *Proc. AINA Workshops(1)*, Niagara Falls, Canada, May 21–23, Vol. 1, pp. 601–606. IEEE Computer Society, Washington.

[15] ElGamal, T. (1985) A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Inform. Theory*, **31**, 469–472.

[16] Fischlin, M. (2001) Trapdoor commitment schemes and their applications. PhD Thesis, Johann Wolfgang Goethe-University, Frankfurt am Main.

[17] Pointcheval, D. and Stern, J. (1996) Security Proofs for Signature Schemes. *Proc. Eurocrypt'96*, Saragossa, Spain, May 12–16, Lecture Notes in Computer Science, Vol. 1070, pp. 387–398. Springer, Berlin.

[18] Miller, G. (1976) Reimann's hypothesis and tests for primality. *J. Comput. System Sci.*, **13**, 300–317.

[19] FIPS PUB 186-2 (2000) *Digital Signature Standard* (*DSS*). Federal Information Processing Standards Publication, National Institute of Standards and Technology, USA.

[20] Rivest, R., Shamir, A. and Adleman, L. (1978) A method for obtaining digital signatures and public key cryptosystems. *Commun. ACM*, **21**, 120–126.