Comments and Improvements on Chameleon Hashing Without Key Exposure Based on Factoring

Xiaofeng Chen[1], Haibo Tian[2], Fangguo Zhang[2]

[1] Key Laboratory of Computer Networks and Information Security, Ministry of Education, Xidian University, Xi'an 710071, P.R.China
xfchen@xidian.edu.cn
[2] School of Information Science and Technology, Sun Yat-sen University, Guangzhou 510275, P.R.China
{tianhb,isszhfg}@mail.sysu.edu.cn

Abstract. In this paper, we present some security flaws of the key-exposure free chameleon hash scheme based on factoring [9]. Besides, we propose an improved chameleon hash scheme without key exposure based on factoring which enjoys all the desired security notions of chameleon hashing.

Key words: Chameleon hashing, Factoring problem, Key exposure.

1 Introduction

Chameleon signatures, introduced by Krawczyk and Rabin [12], are based on well established hash-and-sign paradigm, where a chameleon hash function is used to compute the cryptographic message digest. A chameleon hash function is a trapdoor one-way hash function, which prevents everyone except the holder of the trapdoor information from computing the collisions for a randomly given input. Chameleon signatures simultaneously provide non-repudiation and non-transferability for the signed message as undeniable signatures [4] do, but the former allows for simpler and more efficient realization than the latter. In particular, chameleon signatures are non-interactive and less complicated. Besides, since the chameleon signatures are based on well established hash-and-sign paradigm, it provides more generic and flexible constructions.

One limitation of the original chameleon signature scheme is that signature forgery results in the signer recovering the recipient's trapdoor information, *i.e.*, the private key. The signer then can use this information to deny other signatures given to the recipient. Ateniese and de Mederious [1] firstly addressed the key exposure problem of chameleon hashing and introduced the idea of identity-based chameleon hashing to solve this problem. Due to the distinguishing property of identity-based system, the signer can sign a message to an intended recipient, without having to first retrieve the recipient's certificate. Moreover, the

signer uses a different public key (corresponding a different private key) for each transaction with a recipient, so that signature forgery only results in the signer recovering the trapdoor information associated to a single transaction. Therefore, the signer will not be capable of denying signatures on any message in other transactions. We argue that this idea only provides a partial solution for the problem of key exposure since the recipient's public key is changed for each transaction.

Chen et al. [6] proposed the first full construction of a key-exposure free chameleon hash function in the gap Diffie-Hellman (GDH) groups with bilinear pairings. Ateniese and de Mederious [2] then presented three key-exposure free chameleon hash schemes, two based on the RSA assumption, as well as a new construction based on pairings. Recently, Gao et al. [10] claimed to present a key-exposure free chameleon hash scheme based on the Schnorr signature. However, it requires an interactive protocol between the signer and the recipient and thus violates the basic definition of chameleon hashing and signatures. Chen et al. [7] proposed the first discrete-logarithm-based key-exposure free chameleon hashing without using the GDH groups. Besides, Gao et al. [9] proposed a factoring-based chameleon hash scheme without key exposure, which we call Gao-Wang-Xie's chameleon hash scheme. Independently, Kurosawa et al. [11] proposed a double-trapdoor commitment scheme based on factoring. Since any commitment scheme with a non-interactive commitment phase induces a chameleon hash function and vice versa, these two schemes are actually equivalent to each other. Also, we argue that they are both closely related to the presentation problem of factoring [8].

Our Contribution. In this paper, we give a comment on Gao-Wang-Xie's chameleon hash scheme and point out some security flaws of the scheme. We also propose an improved chameleon hash scheme based on factoring which achieves all the desired security notions of chameleon hashing.

Organization. The rest of the paper is organized as follows: Some preliminaries are given in Section 2. Gao-Wang-Xie's chameleon hash scheme is introduced in Section 3. The comment on Gao-Wang-Xie's chameleon hash scheme is given in Section 4. The improved key-exposure free chameleon hashing based on factoring is proposed in Section 5. Finally, conclusions will be made in Section 6.

2 Preliminaries

In this section, we first introduce the formal definitions and security requirements of chameleon hashing [1, 2], and then introduce a variant Rabin signature scheme [9].

2.1 Chameleon Hashing

A chameleon hash function is a trapdoor collision-resistant hash function, which is associated with a trapdoor/hash key pair (TK, HK). Anyone who knows the

public key HK can efficiently compute the hash value for each input. However, there exists no efficient algorithm for anyone except the holder of the secret key TK, to find collisions for every given input. In the following, we present a formal definition of a chameleon hash scheme.

Definition 1. *A chameleon hash scheme consists of four efficient algorithms* (**GenKey**, **Hash**, **UForge**, **IForge**)*:*

- **GenKey***: A probabilistic polynomial-time algorithm that, on input a security parameter k, outputs a trapdoor/hash key pair (TK, HK).*
- **Hash***: A probabilistic polynomial-time algorithm that, on input the hash key HK, a label L, a message m, and a random string r, outputs the hashed value $h = \mathrm{Hash}(HK, L, m, r)$. Note that h does not depend on TK.*
- **UForge** *(universal forge): A deterministic polynomial-time algorithm $\mathcal{F}$ that, on input the trapdoor key TK, a message m, a random string r, and another message $m' \neq m$, outputs a string r' that satisfies*

$$\mathrm{Hash}(HK, L, m', r') = \mathrm{Hash}(HK, L, m, r).$$

  Moreover, if r is uniformly distributed in a finite space $\mathcal{R}$, then the distribution of r' is computationally indistinguishable from uniform in $\mathcal{R}$.
- **IForge** *(instance forge): A deterministic polynomial-time algorithm that, on input a tuple (HK, L, m, r, m', r') such that $h = \mathrm{Hash}(HK, L, m', r') = \mathrm{Hash}(HK, L, m, r)$, outputs a new collision (m'', r'') that also satisfies $h = \mathrm{Hash}(HK, L, m'', r'')$.*

A secure chameleon hashing scheme satisfies the following properties:

- **Collision resistance:** Without the knowledge of trapdoor key TK, there exists no efficient algorithm that, on input a message m, a random string r, and another message m', outputs a string r' that satisfy $\mathrm{Hash}(HK, L, m', r') = \mathrm{Hash}(HK, L, m, r)$, with non-negligible probability.
- **Semantic security:** For all pairs of messages m and m', the probability distributions of the random values $\mathrm{Hash}(HK, L, m', r)$ and $\mathrm{Hash}(HK, L, m, r)$ are computationally indistinguishable. In formal terms, let $H[X]$ denote the entropy of a random variable X, and $H[X|Y]$ the entropy of the variable X given the value of a random function Y of X. Semantic security is the statement that the conditional entropy $H[m|h]$ of the message given its chameleon hash value h equals the total entropy $H[m]$ of the message space.
- **Message hiding:** Given a collision (m', r') and (m, r) of the chameleon hash scheme, *i.e.*, $h = \mathrm{Hash}(HK, L, m', r') = \mathrm{Hash}(HK, L, m, r)$. Then the sender can successfully contest this invalid claim by releasing a third pair (m'', r'') such that $h = \mathrm{Hash}(HK, L, m'', r'')$, without having to reveal the original signed message m.
- **Key exposure freeness:** If a recipient has never computed a collision under a label L, then there is no efficient algorithm for an adversary to find a collision for a given chameleon hash value $\mathrm{Hash}(HK, L, m, r)$. This must remain true even if the adversary has oracle access to $\mathcal{F}$ and is allowed polynomially many queries on triples (L_j, m_j, r_j) of his choice, except that L_j is not allowed to equal the challenge L.

2.2 A Variant of Rabin Signature Scheme

Let $N = pq$ is a Blum integer, where p, q are two random primes such that $p = q = 3 \mod 4$. Denote by QR_N the set of all quadratic residue modulo N, we know that either $m \in QR_N$ or $-m \in QR_N$ if the Jacobi symbol $(\frac{m}{N}) = +1$. Note that the Jacobi symbol can be calculated without knowledge of the factorization of N. Also, for a Blum integer, squaring is a permutation on the group of quadratic residues QR_N. Trivially, it can be extended to 2^l-th power for any positive integer l.

Define a cryptographic hash function $H : \{0,1\}^* \to Z_N^*[+1]$, where $Z_N^*[+1] = \{a | a \in Z_N^*, (\frac{m}{N}) = +1\}$ is the set of elements of Z_N^* with Jacobi symbol is $+1$. Constructions of the hash function H can be found in [5, 9]. A variant Rabin signature scheme based on factoring assumption is given as follows:

- **Sign:** Given a message m, compute the signature $\sigma = |H(m)|^{\frac{1}{2}} \mod N$, where $|H(m)| = H(m)$ if $H(m) \in QR_N$; $|H(m)| = -H(m)$ otherwise.
- **Verify:** Given a pair (m, σ), if either $\sigma^2 = H(m) \mod N$ or $\sigma^2 = -H(m) \mod N$ holds, then σ is a valid signature for message m.

3 Gao-Wang-Xie's Chameleon Hashing

In this section, we introduce Gao-Wang-Xie's chameleon hash scheme without key exposure based on factoring [9], which consists of the following efficient algorithms.

- **GenKey:** Given a security parameter k, let $N = pq$ where p, q are two distinct odd primes with the same length such that $p = q = 3 \mod 4$. Define a cryptographic hash function $H : \{0,1\}^* \to Z_N^*[+1]$. The public key is N and the secret key is (p, q). Additionally, we restrict the considered message space of the chameleon hash is $\{0,1\}^{f(k)}$ where $f(k)$ is super-logarithmic in k, *i.e.*, $0 \leq m \leq 2^{f(k)} - 1$. Trivially, the case of the message space of $\{0,1\}^*$ can be easily extended by using a resistant hash function from $\{0,1\}^*$ to $\{0,1\}^{f(k)}$.
- **Hash:** Given the public key N, a label L, and a message $m \in \{0,1\}^{f(k)}$, firstly choose a random string $r \in \mathbb{Z}_N$ and compute the hash value

$$h = \text{Hash}(N, L, m, r) = bJ^m r^{2^{f(k)}} \mod N,$$

  where $J = H(L)$, $b \in \{+1, -1\}$.
- **Uforge:** Given the secret key p, q, the original input (m, r), another message $m' \neq m$, first compute the ephemeral trapdoor $B = |H(L)|^{\frac{1}{2^{f(k)}}} \mod N$ for the label L, here $|H(L)| = H(L)$ if $H(L) \in QR_N$; $|H(L)| = -H(L)$ otherwise. Then compute the corresponding random string $r' = rB^{m-m'}$

mod N. Note that

$$\begin{aligned}\text{Hash}(N, L, m', r') &= \pm H(L)^{m'} r'^{2^{f(k)}} \\ &= \pm H(L)^{m'} (rB^{m-m'})^{2^{f(k)}} \\ &= \pm H(L)^{m'} |H(L)|^{m-m'} r^{2^{f(k)}} \\ &= \pm H(L)^{m} r^{2^{f(k)}} \\ &= \pm \text{Hash}(N, L, m, r)\end{aligned}$$

Since the only difference between $\text{Hash}(N, L, m, r)$ and $\text{Hash}(N, L, m', r')$ is $\pm$, (m, r) and (m', r') are viewed as a valid collision of the chameleon hash function.

– **IForge:** Given a valid collision (m, r) and (m', r'), we have $\text{Hash}(N, L, m, r) = \pm\text{Hash}(N, L, m', r') \mod N$, *i.e.*, $|H(L)|^{m-m'} = (r'/r)^{2^{f(k)}} \mod N$. Similar to the technique in [8], we can compute a square root θ of $J' = |H(L)|$ as follows:
Let $2^s = \gcd(m - m', 2^{f(k)})$, where $0 \leq s < f(k)$. Compute $u, v \in Z$ such that $u(m - m') + v2^{f(k)} = 2^s$ and then compute

$$\begin{aligned}J'^{2^s} &= J'^{u(m-m')+v2^{f(k)}} \\ &= (J'^{m-m'})^u (J'^v)^{2^{f(k)}} \\ &= ((r'/r)^u J'^v)^{2^{f(k)}} \mod N\end{aligned}$$

Let $\theta = ((r'/r)^u J'^v)^{2^{f(k)-s-1}}$, we have $J'^{2^s} = (\theta^2)^{2^s}$. Since $J', \theta \in QR_N$ and square is a permutation of the group QR_N, we have $J' = \theta^2 \mod N$.
Now if $m' \geq 2^{f(k)-1}$, let $m'' = m' - 2^{f(k)-1}$ and $r'' = r'\theta \mod N$; if $m' < 2^{f(k)-1}$, let $m'' = m' + 2^{f(k)-1}$ and $r'' = r'/\theta \mod N$. We can verify that $\text{Hash}(N, L, m'', r'') = \pm\text{Hash}(N, L, m', r') \mod N$.

Theorem 1. *The above chameleon hash scheme enjoys the properties of collision resistance, message hiding, semantic security, and key-exposure freeness.*

Proof. – Collision Resistance: Exposing a collision allows anybody to compute a variant Rabin signature $|H(L)|^{\frac{1}{2}}$ on message L. Since the variant Rabin signature is existentially unforgeable under the factoring assumption, the proposed chameleon hash function is collision resistance.
– Message Hiding: Given a collision (m, r) and (m', r'), we can use the algorithm **IForge** to compute another pair (m'', r'').
– Semantic Security: For each message m, the hash value $h = \text{Hash}(N, L, mr,)$ is uniquely determined by the value $r^{2^{f(k)}}$ with ignoring $\pm$, and vice versa. So, the conditional probability taken over the message space $\mu(m|h)=\mu(m|r^{2^{f(k)}})$. Also, $\mu(m|r^{2^{f(k)}}) = \mu(m)$ since m and r are independent variables. So, $\mu(m|h) = \mu(m)$, *i.e.*, the chameleon hash value h discloses no information about m.

– Key-exposure Freeness: If an attacker $\mathcal{A}_1$ against the above chameleon hash scheme can be successful with respect to the property of key-exposure freeness, then we can use it to construct an attacker $\mathcal{A}_2$ of type *uf-ecma* against the above variant Rabin signature as follows: First $\mathcal{A}_2$ is given the public parameters $(N, H, f(k))$ of the variant Rabin signature, and $\mathcal{A}_2$ passes them to $\mathcal{A}_1$. Then when $\mathcal{A}_1$ makes a query (L_i, m_i, r_i) to the oracle **UForge**, $\mathcal{A}_2$ can get the ephemeral trapdoor $|H(L)|^{\frac{1}{2^{f(k)}}} \mod N$ from its own oracle access and further compute a collision (m_i', r_i') as in **UForge** and return it. At last, $\mathcal{A}_1$ returns a collision (m, r) and (m', r') and a never queried label L such that $\text{Hash}(N, L, m', r') = \text{Hash}(N, L, m, r)$, $\mathcal{A}_2$ can compute $|H(L)|^{\frac{1}{2}} \mod N$ as in **IForge**, which is the variant Rabin signature for message L.

4 Comments on Gao-Wang-Xie's Chameleon Hashing

In this section, we present some security flaws of Gao-Wang-Xie's chameleon hash scheme.

Firstly, we point out that the definition of Gao-Wang-Xie's chameleon hashing is not rigorous. For a given input, the hash value h is a random variable dependent on the random bit b. This is considered to be a main trick to design key-exposure free chameleon hashing based on factoring. For more details, please refer to the remark 2 of [9]. Also, (m, r) and (m', r') is a valid collision if $\text{Hash}(N, L, m', r') = \pm\text{Hash}(N, L, m, r)$ holds. This strongly violates the original definition of chameleon hashing and the collisions. The reason for this paradoxical definition is that anyone without the information of p, q can not know whether $H(L)$ is a quadratic residue. We present a solution to this problem as follows:

Define the chameleon hash function $h = \text{Hash}(N, L, m, r) = H(L)^m r^{2^{f(k)}} \mod N$. We consider the following situations:

– If $H(L) \in QR_N$, the receiver with the trapdoor $H(L)^{1/2^{f(k)}}$ to compute a pair (m', r') such that $h = H(L)^{m'} r'^{2^{f(k)}}$.
– If $H(L) \notin QR_N$, then $-H(L) \in QR_N$.
  • If m is an even, then $h = H(L)^m r^{2^{f(k)}} = (-H(L))^m r^{2^{f(k)}} \mod N$, the receiver can use the trapdoor $(-H(L))^{1/2^{f(k)}}$ to compute a collision (m', r') where m' is also an even.
  • If m is an odd, then $h = H(L)^m r^{2^{f(k)}} = -(-H(L))^m r^{2^{f(k)}} \mod N$, the receiver can use the trapdoor $(-H(L))^{1/2^{f(k)}}$ to compute a collision (m', r') where m' is also an odd.

Therefore, we can always define $h = \text{Hash}(N, L, m, r) = H(L)^m r^{2^{f(k)}} \mod N$. This makes the chameleon hash scheme very simple and easily to be understand. In the section 5, we present another solution which still uses the random bit b to fix this problem.

The second security flaw is the proof for key-exposure freeness. When $\mathcal{A}_1$ makes queries (L_i, m_i, r_i) to **Uforge**, can $\mathcal{A}_2$ always know the information $|H(L_i)|^{1/2^{f(k)}}$? Note that $\mathcal{A}_2$ can not know the master trapdoor (p, q).

Let us consider **IForge** more carefully: Given a collision (m, r) and (m', r') for L, we have $|H(L)|^{m-m'} = (r'/r)^{2^{f(k)}}$. Define $\gcd(m - m', 2^{f(k)}) = 2^s$, here $0 \leq s < k$ (this imply that $m \leq 2^{f(k)} - 1$). Compute (u, v) such that $u(m - m') + v2^{f(k)} = 2^s$, so we have $((r'/r)^u |H(L)|^v)^{2^k} = |H(L)|^{2^s}$.

Trivially, $\theta = ((r'/r)^u |H(L)|^v)^{2^{f(k)-s-1}} = |H(L)|^{\frac{1}{2}} \mod N$ (this is the result of [9]). On the other hand, if we define $\theta' = (r'/r)^u |H(L)|^v$, then we have $\theta' = |H(L)|^{\frac{1}{2^{f(k)-s}}} \mod N$. Of course, if we know θ', we can compute θ easily. However, for any integer $s > 0$, it is difficult to compute $|H(L)|^{\frac{1}{2^{f(k)}}}$.

In the proof for key-exposure freeness of Gao-Wang-Xie's chameleon hash scheme, the attacker $\mathcal{A}_2$ can always obtain the ephemeral trapdoor key $|H(L)|^{\frac{1}{2^{f(k)}}} \mod N$ from its own oracle access. This requires that the variant Rabin signature is still existentially unforgeable against the so-called uf-ecma attacker under the factoring assumption. The uf-ecma attacker is more powerful than the traditional adaptively chosen message attacker because uf-ecma attacker can always access to an oracle to obtain $|H(L)|^{\frac{1}{2^{f(k)}}} \mod N$. This seems to be a much stronger assumption to prove the security of the variant Rabin signature, *i.e.*, it is much more difficult to prove the unforgeability of the variant Rabin signature. The authors [9] do not provide the complete proof. Actually, observe that $|H(L)|^{\frac{1}{2^l}}$ is $f(k)$ consecutive trapdoors, where $1 \leq l \leq f(k)$. A higher trapdoor $|H(L)|^{\frac{1}{2^l}}$ can be used to compute a lower trapdoor $|H(L)|^{\frac{1}{2^{l-1}}}$. In the random oracle model, we argue that it is enough to compute a collision of the chameleon hash scheme with the trapdoor $|H(L)|^{\frac{1}{2}}$. Therefore, it only requires that the variant Rabin signature is existentially unforgeable against the traditional adaptively chosen message attacker, which can be easily proven based on the technique [3]. We will present the details in the section 5.

Finally, the collision (m'', r'') is a fixed pair in **IForge** of Gao-Wang-Xie's chameleon hash scheme. Actually, we can provide plenty of other collisions since the real ephemeral trapdoor is not $H(L)^{\frac{1}{2}}$, but $|H(L)|^{\frac{1}{2^{f(k)-s}}}$ as discussed above. Therefore, for any message m'' such that $2^s | m' - m''$, we can compute the corresponding r'' as a collision. Only when $s = f(k) - 1$, the pair (m'', r'') is unique determined. For more details, please refer to section 5.

5 Improved Chameleon Hashing Based on Factoring

In this section, we present an improved chameleon hashing without key exposure based on factoring. Our chameleon hash scheme is defined as

$$h = \text{Hash}(N, L, m, r, b) = bJ^m r^{2^{f(k)}} \mod N,$$

where $J = H(L)$, $b \in \{+1, -1\}$.

Though we also use a random bit b, it is viewed as a part of the input of the chameleon hash scheme. This modification makes the chameleon hash value h is a constant for a given input. Also, (m, r, b) and (m', r', b') is a valid collision if

$\mathrm{Hash}(N, L, m', r', b') = \mathrm{Hash}(N, L, m, r, b)$ holds. This consists with the original definition of the collisions since we avoid the notation " $\pm$ ".

The improved chameleon hash scheme based on factoring consists of the following efficient algorithms:

- **GenKey:** The system parameters are the same as that of Gao-Wang-Xie's chameleon hash scheme.
- **Hash:** Given the public key N, a label L, and a message $m \in \{0,1\}^{f(k)}$, firstly choose a random string $r \in \mathbb{Z}_N$ and a random bit $b \in \{+1, -1\}$, compute the hash value

$$h = \mathrm{Hash}(N, L, m, r, b) = bJ^m r^{2^{f(k)}} \mod N,$$

  where $J = H(L)$.
- **Uforge:** Given the secret key (p, q), the original input (m, r, b), another message $m' \neq m$, first compute the trapdoor $B = |H(L)|^{\frac{1}{2^{f(k)}}} \mod N$ for the label L, here $|H(L)| = H(L)$ if $H(L) \in QR_N$; $|H(L)| = -H(L)$ otherwise. Then the corresponding collision (r', b') can be given as follows:

$$r' = rB^{m-m'} \mod N,$$

$$b' = \begin{cases} b, & \text{if } H(L) \in QR_N \\ b(-1)^{m-m'}, & \text{Otherwise} \end{cases}$$

  Note that

$$\begin{aligned} \mathrm{Hash}(N, L, m', r', b') = \; & b' J^{m'} r'^{2^{f(k)}} \\ & = b' H(L)^{m'} (rB^{m-m'})^{2^{f(k)}} \\ & = b' H(L)^{m'} |H(L)|^{m-m'} r^{2^{f(k)}} \\ & = bH(L)^m r^{2^{f(k)}} \\ & = \mathrm{Hash}(N, L, m, r, b) \end{aligned}$$

  Therefore, the forgery is successful. Moreover, if (r, b) is uniformly distributed, then the distribution of (r', b') is computationally indistinguishable from uniform.
- **IForge:** Given a collision (m, r, b) and (m', r', b'), we have $\mathrm{Hash}(N, L, m, r, b) = \mathrm{Hash}(N, L, m', r', b') \mod N$, *i.e.*, $|H(L)|^{m-m'} = (r'/r)^{2^{f(k)}} \mod N$. Let $2^s = \gcd(m - m', 2^{f(k)})$, where $0 \leq s < f(k)$. Compute $u, v \in Z$ such that $u(m - m') + v2^{f(k)} = 2^s$. Similarly, we can compute $\theta = (r'/r)^u |H(L)|^v = |H(L)|^{\frac{1}{2^{f(k)-s}}} \mod N$. Trivially, we can compute $|H(L)|^{\frac{1}{2}} \mod N$. Moreover, if $\theta^{2^{f(k)-s}} = H(L)$, then $H(L) \in QR_N$; else, $-H(L) \in QR_N$. That is, it is efficient to check whether $H(L)$ is a quadratic residue modulo N.

  For any message m'' such that $0 \leq m'' \leq 2^{f(k)} - 1$ and $2^s | m' - m''$, the corresponding collision (r'', b'') can be given as follows:

$$r'' = r'\theta^{2^{-s}(m'-m'')} \mod N,$$
$$b'' = \begin{cases} b', & \text{if } H(L) \in QR_N \\ b'(-1)^{m'-m''}, & \text{Otherwise} \end{cases}$$

Actually, note that

$$\begin{aligned} \text{Hash}(N, L, m'', r'', b'') = & \ b''J^{m''}r''^{2^{f(k)}} \\ & = b''H(L)^{m''}(r'\theta^{2^{-s}(m'-m'')})^{2^{f(k)}} \\ & = b''H(L)^{m''}\theta^{2^{(f(k)-s)(m'-m'')}}r'^{2^{f(k)}} \\ & = b''H(L)^{m''}|H(L)|^{m'-m''}r'^{2^{f(k)}} \\ & = b'H(L)^{m'}r'^{2^{f(k)}} \\ & = \text{Hash}(N, L, m', r', b') \end{aligned}$$

Thus, the instance forgery is successful.

Theorem 2. *The proposed chameleon hash scheme enjoys the properties of collision resistance, message hiding, semantic security, and key-exposure freeness.*

Proof. We prove that the proposed chameleon hash scheme satisfies all the desired security properties.

- Collision Resistance: Given two pairs (m, r) and (m', r') with the label L such that $\text{Hash}(N, L, m', r', b') = \text{Hash}(N, L, m, r, b)$, then as in **IForge** the trapdoor $|H(L)|^{\frac{1}{2^{f(k)-s}}} \mod N$ is revealed, which allows anybody to compute a variant Rabin signature $|H(L)|^{\frac{1}{2}}$ on message L. Since the variant Rabin signature is existentially unforgeable under the factoring assumption, the proposed chameleon hash function is collision resistance.
- Message Hiding: Given a collision (m, r) and (m', r'), we can use **IForge** to compute another pair (m'', r'').
- Semantic Security: For each message m, the hash value $h = \text{Hash}(N, L, m, r, b)$ is uniquely determined by the value $(r^{2^{f(k)}}, b)$, and vice versa. So, the conditional probability taken over the message space $\mu(m|h)$=$\mu(m|(r^{2^{f(k)}}, b))$. Also, $\mu(m|(r^{2^{f(k)}}, b)) = \mu(m)$ since m and (r, b) are independent variables. So, $\mu(m|h) = \mu(m)$, *i.e.*, the chameleon hash value h discloses no information about m.
- Key-exposure Freeness: If an attacker $\mathcal{A}_1$ against the above chameleon hash scheme can be successful with respect to the property of key-exposure freeness, then we can use it to construct an adaptive chosen message attacker $\mathcal{A}_2$ against the above variant Rabin signature as follows:
  Suppose $\mathcal{A}_2$ is given the public parameters $(N, H, f(k))$ of the variant Rabin signature, and $\mathcal{A}_2$ is allowed to makes queries to the H oracle and **Sign** oracle of the variant Rabin signature scheme. $\mathcal{A}_2$ then passes $(N, \text{Hash}(), H, f(k))$ to $\mathcal{A}_1$, where Hash() is the proposed chameleon hash scheme. Similar to [9],

the security analysis will view H as a random oracle. When $\mathcal{A}_1$ makes a query (L_i, m_i, r_i, b_i) to the oracle **UForge**, $\mathcal{A}_2$ firstly makes a query L_i to the H oracle and **Sign** oracle to get a pair $(H(L_i), \sigma_i = |H(L_i)|^{\frac{1}{2}} \mod N)$, and then uses the trapdoor σ_i to compute a collision (m_i', r_i', b_i') as follows:

Let $s = f(k) - 1$ in **IForge**, we have $m_i - m_i' = \pm 2^{f(k)-1}$. Therefore, if $m_i \geq 2^{f(k)-1}$, then the collision is $(m_i - 2^{f(k)-1}, r_i\sigma_i, b_i)$; if $m_i < 2^{f(k)-1}$, then the collision is $(m_i + 2^{f(k)-1}, r_i/\sigma_i, b_i)$. $\mathcal{A}_2$ sends $H(L_i)$ and the collision (m_i', r_i', b_i') to $\mathcal{A}_1$. At the end of the game, the output of $\mathcal{A}_1$ is a collision (m, r, b) and (m', r', b') for a never queried label $L \neq L_i$ such that $\text{Hash}(N, L, m', r', b') = \text{Hash}(N, L, m, r, b)$. Then $\mathcal{A}_2$ can compute $|H(L)|^{\frac{1}{2}} \mod N$ as in **IForge**, which is the variant Rabin signature for message L. Since the variant Rabin signature is existentially unforgeable against the adaptively chosen message attacker in the random oracle model, the proposed chameleon hash scheme is key-exposure free. So, it is unnecessary to prove that the variant Rabin signature is still existentially unforgeable against the so-called uf-ecma attacker in the random oracle model (even the claim is true).

6 Conclusions

Chameleon signatures simultaneously provide the properties of non-repudiation and non-transferability for the signed message. However, the initial constructions of chameleon signatures suffer from the problem of key exposure. This creates a strong disincentive for the recipient to forge signatures, partially undermining the concept of non-transferability. Recently, some constructions of chameleon hashing and signatures without key exposure are presented based on different mathematical assumptions.

In this paper, we present some security flaws and disadvantages of the key-exposure free chameleon hash scheme based on factoring [9]. Moreover, we propose an improved chameleon hash scheme without key exposure based on factoring which enjoys all the desired security notions of chameleon hashing.

References

1. G. Ateniese and B. de Medeiros, *Identity-based chameleon hash and applications*, 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*, SCN 2004, LNCS 3352, pp.165-179, Springer-Verlag, 2005.
3. M. Bellare and P. Rogaway, *The exact security of digital signatures-How to sign with RSA and Rabin*, Advances in Cryptology -Eurocrypt 1996, LNCS 1070, pp.399-416, Springer-Verlag, 1996.
4. D. Chaum and H. van Antwerpen, *Undeniable signatures*, Advances in Cryptology-Crypto 1989, LNCS 435, pp.212-216, Springer-Verlag, 1989.
5. C. Cocks, *An identity based encryption scheme based on quadratic residues*, Cryptography and Coding 2001, LNCS 2260, pp.360-363, Springer-Verlag, 2001.

6. X. Chen, F. Zhang, and K. Kim, *Chameleon hashing without key exposure*, ISC 2004, LNCS 3225, pp.87-98, Springer-Verlag, 2004.
7. X. Chen, F. Zhang, W. Susilo, Y. Mu, H. Lee, and K. Kim, *Key-exposure free chameleon hashing and signatures based on discrete logarithm sytems*, available at: http://eprint.iacr.org/2009/035.
8. M. Fischlin, and R. Fischlin, *The representation problem based on factoring*, CT-RSA 2002, LNCS 2271, pp.96-113, Springer-Verlag, 2002.
9. W. Gao, X. Wang, and D. Xie, *Chameleon hashes without key exposure based on factoring*, Journal of Computer Science and Technology, 22(1), pp.109-113, 2007.
10. W. Gao, F. Li, and X. Wang, *Chameleon hash without key exposure based on Schnorr signature*, Computer Standards and Interfaces, vol. 31, pp.282-285, 2009.
11. K. Kurosawa and K. Schmidt-Samoa, *New online/offline signature schemes without random oracles*, PKC 2006, LNCS 3958, pp.330-346, Springer-Verlag, 2006.
12. H. Krawczyk and T. Rabin, *Chameleon hashing and signatures*, Proc. of NDSS 2000, pp.143-154, 2000.