Chameleon Hashing without Key Exposure

Xiaofeng Chen[1], Fangguo Zhang[2] and Kwangjo Kim[1]

[1] International Research center for Information Security (IRIS)
Information and Communications University(ICU),
58-4 Hwaam-dong Yusong-ku, Taejon, 305-732 KOREA
{crazymount,kkj}@icu.ac.kr

[2] School of Information Technology and Computer Science
University of Wollongong, NSW 2522 Australia
fangguo@uow.edu.au

Abstract. Chameleon signatures are based on well established hash-and-sign paradigm, where a *chameleon hash function* is used to compute the cryptographic message digest. Chameleon signatures simultaneously provide the properties of non-repudiation and non-transferability for the signed message, *i.e.*, the designated recipient is capable of verifying the validity of the signature, but cannot disclose the contents of the signed information to convince any third party without the signer's consent.
One disadvantage of the initial chameleon signature scheme is that signature forgery results in the signer recovering the recipient's trapdoor information, *i.e.*, private key. Therefore, the signer can use this information to deny *other* signatures given to the recipient. This creates a strong disincentive for the recipient to forge signatures, partially undermining the concept of non-transferability. In this paper, we firstly propose a chameleon hashing scheme in the gap Diffie-Hellman group to solve the problem of key exposure. We can prove that the recipient's trapdoor information will never be compromised under the assumption of Computation Diffie-Hellman Problem (CDHP) is intractable. Moreover, we use the proposed chameleon hashing scheme to design a chameleon signature scheme.

Key words: Chameleon hashing, Gap Diffie-Hellman group, Key exposure, Digital signatures.

1 Introduction

The ordinary digital signature provides the functions of integration, authentication, and non-repudiation for the signed message. Anyone can verify the signature with the signer's public key. However, it may be undesirable in many business situations that a signature can be verified universally. For example, disclosing a signed contract to a competitor can benefit one party but jeopardize the interests of the other. This is the conflict between authenticity (non-repudiation) and privacy (controlled verifiability) in the digital signatures. Chaum and Antwerpen [10] firstly introduced the notion of undeniable signatures to solve this conflict. The distinct property of undeniable signatures is that verification of a signature requires the collaboration of the signer. So the signer can control to whom the signed document is being disclosed. After the initial work of Chaum and Antwerpen, plenty of undeniable signature schemes were proposed [6, 9, 18, 16, 17, 23].

Chameleon signatures, introduced by Krawczyk and Rabin [22], 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 do, but the former allows for simpler and more efficient realization than the latter.[1] More precisely, chameleon signatures are non-interactive and do not involve the design and complexity of zero-knowledge proofs on which traditional undeniable signatures are based. Though there exist non-interactive versions of undeniable signatures [21], chameleon signatures are considerably less complex, at the sacrifice of not conferring the signer the ability to engage in non-transferable secondary proofs of signature (non-)validity [1].

One limitation of the initial chameleon signature scheme is that signature forgery results in the signer recovering the recipient's trapdoor information, *i.e.*, private key. Such feather has some advantages in certain applications. For example, the user can deny the forged message without revealing the original message. However, the signer can use this information to deny *other* signatures given to the recipient. In the worst case, the signer can sign any document or decrypt any message on behalf of the recipient. In fact, exposure of secret keys is perhaps the most important devastating attack on a cryptosystem [12]. This potential damage will create a strong disincentive for the recipient to forge signatures and thus weakens the property of non-transferability.

Ateniese and de Mederious [1] argue that non-transferability is more convincing if the scheme is such that forgery of a hash does not compromise the secret key of the recipient.[2] They firstly introduced the idea of identity-based chameleon hashing to solve this problem. Due to the distinguishing characteristic of identity-based system, the signer can sign a message to an intended recipient, without having to first retrieve the recipient's certificate.[3] Furthermore, the signer uses a different public key (corresponding a different private key) for each

[1] There are some difference on non-transferability between undeniable signatures and chameleon signatures. In undeniable signatures, the verification of the signature needs the cooperation of the signer, which ensures the non-transferability. In chameleon signatures, the recipient is fully capable of providing any indistinguishable chameleon hashing inputs to satisfy the signature, thus the third party can not trust the recipient's claim.

[2] This problem can be solved in any chameleon signature scheme if the signer changes his key pair frequently, however, it is only meaningful in theory sense because the key distribution problem arises.

[3] It is easily to see that any key-insulted systems with a physically secure device can substitute the identity-based systems.

transaction with a recipient, so that signature forgery only results in the signer recovering the trapdoor information associated to a single transaction. Therefore, if the recipient produces a hash collision, the signer can recover the corresponding trapdoor information to deny the forged signature by providing a different collision. However, she will not be capable of denying signatures on any message in other transactions. We argue that their scheme does not solve the problem of key exposure essentially. The basic idea is still that the recipient's public keys are changed often.[4] To the best of our knowledge, there seems no efficient chameleon hashing scheme which enjoys the message hiding property without exposing the private key. In this paper, we propose a novel chameleon hashing scheme without key exposure under certificate-based systems, which enjoys all the properties of traditional chameleon hashing schemes, but the trapdoor information cannot be compromised even if the recipient forges a hash collision. Thus, the non-transferability is strengthened.

1.1 Related Work

There are plenty of research on the conflict between authenticity (non-repudiation) and privacy (controlled verifiability) in the digital signatures. Undeniable signatures enable the signer to decide *when* her signature can be verified. An extended notion is "designated confirmer signatures" [8], where a designated confirmer, instead of the signer, can be involved in the verification of the signature when the signer is inconvenient to cooperate. In some applications, it is important for the signer to decide not only *when* but also *by whom* her signature can be verified due to the blackmailing [14, 20] and mafia [13] attacks. For example, the voting center presents a proof to convince a certain voter that his vote was counted while without letting him to convince others (*e.g.*, a coercer) of his vote, which is important to design a receipt-free electronic voting scheme preventing vote buying and coercion. This is the motivation of the concept of "designated verifier signatures" [21]. The designated verifier will trust the signer indeed signed a message with a proof of the signer. However, he cannot present the proof to convince any third party because he is fully capable of generating the same proof by himself. Very recently, Steinfeld *et al.* [26] introduced the conception of "universal designated verifier signatures", which can be viewed as an extended notion of designated verifier signatures. Universal designated verifier signatures allow any holder of the signature (not necessarily the signer) to designate the signature to any desired designated verifier. The verifier can be convinced that the signer indeed generated the signature, but cannot transfer the proof to convince any third party. In some applications, it is also important for the recipient to decide *when* and *whom* the signer's signature should be verified. This facilitates the concept of "limited verifier signatures" [2, 11].

[4] In identity-based system, the identity information acts as the public key of the user. Identity-based chameleon hash can be computed under a *customized* identity $J = \mathcal{C}(ID_{Recipient}||ID_{Signer}||ID_{Transaction})$. The signer uses a different public key J for each transaction with different $ID_{Transaction}$.

The rest of the paper is organized as follows: Some preliminary works are given in Section 2. Our novel chameleon hashing scheme is given in Section 3. The proposed chameleon signature scheme is given in Section 4. Finally, conclusions will be made in Section 5.

2 Preliminary Works

In this Section, we will briefly describe the basic definition and properties of gap Diffie-Hellman group. We also introduce the formal definition and properties of chameleon hashing scheme.

2.1 Gap Diffie-Hellman Group

Let G be a cyclic multiplicative group generated by g with the prime order q. Assume that the inversion and multiplication in G can be computed efficiently. We introduce the following problems in G.

1. Discrete Logarithm Problem (DLP): Given two elements g and h, to find an integer $a \in Z_q^*$, such that $h = g^a$ whenever such an integer exists.
2. Computation Diffie-Hellman Problem (CDHP): Given (g, g^a, g^b) for $a, b \in Z_q^*$, to compute g^{ab}.
3. Decision Diffie-Hellman Problem (DDHP): Given (g, g^a, g^b, g^c) for $a, b, c \in Z_q^*$, to decide whether $c \equiv ab \bmod q$.

We call G a gap Diffie-Hellman group if DDHP can be solved in polynomial time but there is no polynomial time algorithm to solve CDHP with non-negligible probability. Such group can be found in supersingular elliptic curve or hyperelliptic curve over finite field. For more details, see [4, 5, 7, 15, 19].

We call $< g, g^a, g^b, g^c >$ a valid Diffie-Hellman tuple if $c \equiv ab \bmod q$.

2.2 Chameleon Hashing

A chameleon hashing function is a trapdoor collision resistant hash function, which is associated with a key pair (sk, pk). Anyone who knows the public key pk can efficiently compute the hash value for each input. However, there exists no efficient algorithm for anyone except the holder of the secret key sk, called a trapdoor, to find collisions for every given input. Formally, a chameleon hashing scheme consists of the following efficient algorithms:

- **System Parameters Generation $\mathcal{PG}$:** An efficient probabilistic algorithm that, on input a security parameter k, outputs the system parameters SP.
- **Key Generation $\mathcal{KG}$:** An efficient algorithm that, on input the system parameters SP, outputs a secret/public key pair (sk, pk) for each user.
- **Hashing Computation $\mathcal{H}$:** An efficient probabilistic algorithm that, on input the public key pair pk of a certain user, a message m, and a random integer $r \in Z_q^*$, outputs the hash value $h = \mathrm{Hash}(m, r)$.

– **Collision Computation $\mathcal{F}$:** An efficient algorithm that, on input the secret key sk of the user, a message m, a random integer r, and another message m', outputs an integer $r' \in Z_q^*$ that satisfies

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

A secure chameleon hashing scheme satisfies the following properties:

– **Collision resistance:** Without the knowledge of trapdoor information sk, there exists no efficient algorithm that, on input a message m, a random integer r, and another message m', outputs a random integer r' that satisfy $\mathrm{Hash}(m', r') = \mathrm{Hash}(m, r)$, with non-negligible probability.
– **Semantic security:** For all pairs of message m and m', the probability distribution of the random value $\mathrm{Hash}(m', r)$ and $\mathrm{Hash}(m, r)$ are computationally indistinguishable.

3 The Proposed Chameleon Hashing Scheme

We describe our scheme in two stages. First we present a basic chameleon hashing scheme without key exposure which enjoys the properties of message hiding and semantic security. However, it is not secure against collision forgery if a signer tries to re-use the hashing scheme for a same recipient. The only reason for describing the basic scheme is to make the presentation easier to follow. We then propose our full chameleon hashing scheme without key exposure in Section 3.3.

3.1 The Basic Chameleon Hashing Scheme

– **System Parameters Generation $\mathcal{PG}$:** Let G be a Gap Diffie-Hellman group generated by g, whose order is a prime q. The system parameters are $SP = \{G, q, g\}$.
– **Key Generation $\mathcal{KG}$:** Each user randomly chooses an integer $x \in Z_q^*$ as his private key, and publishes his public key $y = g^x$. The validity of y can be ensured by a certificate issued by a trusted third party.
– **Hashing Computation $\mathcal{H}$:** On input the public key y of a certain user. Randomly chooses an integer $a \in Z_q^*$, and computes (g^a, y^a). Our novel hash function is defined as

$$h = \mathrm{Hash}(m, g^a, y^a) = g^m y^a$$

– **Collision Computation $\mathcal{F}$:** For any valid hash value h, the algorithm $\mathcal{F}$ can be used to compute a hash collision with the trapdoor information x

$$\mathcal{F}(x, h, m, g^a, y^a, m') = (g^{a'}, y^{a'}),$$

where $g^{a'} = g^a g^{x^{-1}(m-m')}$ and $y^{a'} = y^a g^{m-m'}$.

Note that

$$\begin{aligned}\mathrm{Hash}(m', g^{a'}, y^{a'}) &= g^{m'} y^{a'} \\ &= g^{m'} y^{a} g^{m-m'} \\ &= g^{m} y^{a} \\ &= \mathrm{Hash}(m, g^{a}, y^{a})\end{aligned}$$

and $< g, y, g^{a'}, y^{a'} >$ is a valid Diffie-Hellman tuple. Therefore, the forgery is successful.

3.2 Security Analysis

We firstly introduce two variants of CDHP in G:

1. Square Computation Diffie-Hellman Problem (Squ-CDHP): Given (g, g^a) for $a \in Z_q^*$, to compute g^{a^2}.
2. Inverse Computation Diffie-Hellman Problem (Inv-CDHP): Given (g, g^a) for $a \in Z_q^*$, to compute $g^{a^{-1}}$.

Lemma 1. *Squ-CDHP, Inv-CDHP and $CDHP$ are polynomial-time equivalent to each other in G [3, 24, 27].*

Proof. Iv-CDHP $\Longrightarrow$ Squ-CDHP: Given $(g, y = g^a)$ for $a \in Z_q^*$, then $g = y^{a^{-1}}$. Suppose we can solve Iv-CDHP in G, then we can compute $y^a = g^{a^2}$ from $(y, y^{a^{-1}})$. Thus, we can solve Squ-CDHP in G.

Squ-CDHP $\Longrightarrow$ CDHP: Given (g, g^a, g^b) for $a, b \in Z_q^*$, suppose we can solve Squ-CDHP in G, then we can compute g^{a^2}, g^{b^2}, and $g^{(a+b)^2}$ respectively. Note that $g^{2ab} = g^{(a+b)^2}/g^{a^2+b^2}$, we can compute g^{ab} by using the algorithm of [25]. Thus, we can solve CDHP in G

CDHP $\Longrightarrow$ Iv-CDHP: Given $(g, y = g^a)$ for $a \in Z_q^*$, then $g = y^{a^{-1}}$. Suppose we can solve CDHP in G, then we can compute $y^{a^{-2}} = g^{a^{-1}}$ from (y, g, g). Thus, we can solve Iv-CDHP in G. □

Theorem 1. *The basic chameleon hashing scheme is resistant to forgery provided that the CDHP in G is intractable.*

Proof. (sketch) Given two collisions (m, g^a, y^a) and $(m', g^{a'}, y^{a'})$ which satisfy $\mathrm{Hash}(m', g^{a'}, y^{a'}) = \mathrm{Hash}(m, g^a, y^a)$, *i.e.*, $g^{m'} y^{a'} = g^m y^a$, we can easily deduce $g^{x^{-1}} = (g^a/g^{a'})^{(m'-m)^{-1}}$. From Lemma 1, we know it is equivalent to solve the CDHP in G. □

Theorem 2. *The basic chameleon hashing scheme is semantically secure.*

Proof. The proof is similar to [1]. Given a hash value h, and any message m, there exists exactly one pair (g^a, y^a) such that $h = \mathrm{Hash}(m, g^a, y^a)$. □

3.3 The Full Chameleon Hashing Scheme

In the basic scheme, collision forgery will result in the signer recovering the information $g^{x^{-1}}$. Though the secret key x cannot be recovered from this information, it enables the signer to compute the hash collisions if he re-uses the hash function for other transactions with the same recipient. Therefore, the signer should use different public key for each transaction with the recipient, *i.e.*, the recipient should change his private key often. Identity-based chameleon hashing scheme also has this disadvantage: Though the signer does not need to retrieve the certificate of the intended recipient, the recipient should apply for his private key whenever he wants to compute a collision.

We propose our full chameleon hashing scheme without key exposure by using the idea of "Customized Identities" [1]. Let $H : \{0,1\}^* \rightarrow G^*$ is a secure cryptographic hash function, define $I = H(ID_S||ID_R||ID_T)$, where ID_S, ID_R, and ID_T denote the identity of signer, recipient, and transaction, respectively.

3.3.1 The scheme

- **System Parameters Generation** $\mathcal{PG}$**:** Let G be a Gap Diffie-Hellman group generated by g, whose order is a prime q. Define a secure cryptographic hash function $H : \{0,1\}^* \rightarrow G^*$. The system parameters are $SP = \{G, q, g, H\}$.
- **Key Generation** $\mathcal{KG}$ **:** Each user randomly chooses an integer $x \in Z_q^*$ as his private key, and publishes his public key $y = g^x$. The validity of y can be ensured by a certificate issued by a trusted third party.
- **Hashing Computation** $\mathcal{H}$**:** On input the public key y of a certain user. Randomly chooses an integer $a \in Z_q^*$, and computes (g^a, y^a). Our novel hash function is defined as
$$h = \mathrm{Hash}(m, I, g^a, y^a) = (g * I)^m y^a,$$
where I is a customized identity.
- **Collision Computation** $\mathcal{F}$**:** For any valid hash value h, the algorithm $\mathcal{F}$ can be used to compute a hash collision
$$\mathcal{F}(x, h, m, g^a, y^a, m', I) = (g^{a'}, y^{a'}),$$
where $g^{a'} = g^a(g * I)^{x^{-1}(m-m')}$ and $y^{a'} = y^a(g * I)^{m-m'}$.

Note that

$$\begin{aligned}
\mathrm{Hash}(m', I, g^{a'}, y^{a'}) &= (g * I)^{m'} y^{a'} \\
&= (g * I)^{m'} y^a (g * I)^{m-m'} \\
&= (g * I)^m y^a \\
&= \mathrm{Hash}(m, I, g^a, y^a)
\end{aligned}$$

and $< g, y, g^{a'}, y^{a'} >$ is a valid Diffie-Hellman tuple. Therefore, the forgery is successful.

3.3.2 Security Analysis

Theorem 3. *The full chameleon hashing scheme is resistant to forgery under the assumption of CDHP in G is intractable.*

Proof. Given (g', g'^x), let $g = g'/I$, where I is the customized identity. Define the chameleon hash function $h = \mathrm{Hash}(m, I, g^a, y^a) = (g{*}I)^m y^a$. Given collisions (m, g^a, y^a) and $(m', g^{a'}, y^{a'})$ that satisfy $\mathrm{Hash}(m', I, g^{a'}, y^{a'}) = \mathrm{Hash}(m, I, g^a, y^a)$, *i.e.*, $(g{*}I)^{m'} y^{a'} = (g{*}I)^m y^a$, we can deduce $g'^{x^{-1}} = (g{*}I)^{x^{-1}} = (g^a/g^{a'})^{(m'-m)^{-1}}$. That is, we can solve CDHP in G.

Note that the signer uses a different customized identity for each transaction. We remark that collision forgery in a transaction I will result in the signer recovering the information $(g * I)^{x^{-1}}$, however, the signer cannot use it to compute the information of $(g * I')^{x^{-1}}$ for a different transaction I'. Otherwise, he can compute $(I'I^{-1})^{x^{-1}}$, which is equivalent to solve CDHP in G. Therefore, the recipient does not need to change his key pair even if the hash function is reused by the same signer. □

Theorem 4. *The full chameleon hashing scheme is semantically secure.*

Proof. Given a hash value h, a customized identity I, and any message m, there exists exactly one pair (g^a, y^a) such that $h = \mathrm{Hash}(m, I, g^a, y^a)$. □

4 The Proposed Chameleon Signature Scheme

4.1 Precise Definition

A chameleon signature is generated by digitally signing a chameleon hash value of the message. The signer cannot repudiate his signature, but he can deny an invalid signature if he can provide a collision of the chameleon hash function. A good chameleon signature should satisfy the properties of *unforgeability*, *non-transferability*, *non-repudiation*, *deniability*, *message hiding* [1, 22]. Besides, we add the property of *key exposure freeness*, *i.e.*, collision forgery does not result in the signer recovering the recipient's trapdoor information. Formally, a chameleon hashing scheme consists of the following efficient algorithms and a specific denial protocol:

- **System Parameters Generation $\mathcal{PG}$:** An efficient probabilistic algorithm that, on input a security parameter k, outputs the system parameters SP.
- **Key Generation $\mathcal{KG}$:** An efficient algorithm that, on input the system parameters SP, outputs a secret/public key pair (sk, pk) for each user.
- **Signature Generation $\mathcal{SG}$:** An efficient probabilistic algorithm that, on input the public key pair pk_R of the recipient, the secret key sk_S of the signer, a message m, a customized identity I, and a random integer $a \in Z_q^*$, outputs a signature σ on the chameleon hash value $h = \mathrm{Hash}(m, I, g^a, y^a)$.

- **Signature Verification $\mathcal{SV}$:** An efficient deterministic algorithm that, on input the public key pk_R of the recipient, the public key pk_S of the signer, a message m, a customized identity I, g^a, y^a, and a chameleon signature σ, outputs a verification decision $b \in \{0, 1\}$.
- **Denial Protocol $\mathcal{DP}$:** A non-interactive protocol between the signer and the judge. Given a signature σ on the message m, the signer computes a different collision $(m', g^{a'}, y^{a'})$. If and only if $m \neq m'$ and $< g, y, g^{a'}, y^{a'} >$ is a valid Diffie-Hellman tuple, the judge claims that the signature is a forgery.

4.2 Our Scheme

There are two users, a signer S and a recipient R, in our scheme. When dispute occurs, a judge J is involved in the scheme.

- **System Parameters Generation $\mathcal{PG}$:** Let G be a gap Diffie-Hellman group generated by g, whose order is a prime q. Define a secure cryptographic hash function $H : \{0,1\}^* \rightarrow G^*$. The system parameters are $SP = \{G, q, g, H\}$.
- **Key Generation $\mathcal{KG}$:** Each user U randomly chooses an integer $x_U \in Z_q^*$ as his private key, and publishes his public key $y_U = g^{x_U}$. The validity of y_U can be ensured by a certificate issued by a trusted certification authority.
- **Signature Generation $\mathcal{SG}$:** Suppose the signed message is m. The signer S randomly chooses an integer $a \in Z_q^*$, and computes the chameleon hash function value $h = (g*I)^m y_R^a$, here y_R denotes the public key of the recipient R. Assume SIGN is any secure signature scheme based on the assumption that CDHP in G is intractable. The signature σ for the message m consists of
$$(m, I, g^a, y_R^a, \mathrm{SIGN}_{x_S}(h)).$$
Where x_S denotes the private key of the singer S.
- **Signature Verification $\mathcal{SV}$:** Given a signature σ, the recipient first verifies whether $< g, y_R, g^a, y_R^a >$ is a valid Diffie-Hellman tuple. If tuple is invalid, he rejects the signature; else, he then computes the chameleon hash value $h = (g * I)^m y_R^a$ and verifies the validity of $\mathrm{SIGN}_{x_S}(h)$ with the public key y_S of the signer.
- **Denial Protocol $\mathcal{DP}$:** When dispute occurs, *i.e.*, the recipient provides a signature of the signer σ=$(m^*, I, g^{a^*}, y_R^{a^*}, \mathrm{SIGN}_{x_S}(h))$ to the judge J . The judge then asks the signer to provide a collision $(m', g^{a'}, y_R^{a'})$ for the chameleon hash. If the signer can provide such a collision which satisfy that $< g, y_R, g^{a'}, y_R^{a'} >$ is a valid Diffie-Hellman tuple and $m \neq m'$, the judge can be convinced that the recipient forged the signature. If the signer cannot provide such a collision, the judge can be convinced that the signer indeed generated the signature.
  The signer can simply provide (m, g^a, y^a) as the hash collision. However, it will reveal the information of the original message m, which is undesirable in some applications. In section 4.3, we will show in detail how the signer can provide a different collision to ensure the property of message hiding .

4.3 Security Analysis

Theorem 5. *The proposed chameleon signature scheme satisfies the properties of unforgeability, non-transferability, non-repudiation, deniability, message hiding, key exposure freeness.*

Proof. We prove the proposed chameleon signature scheme satisfies the above properties one by one.

- *Unforgeability*: It is trivial because we assume SIGN is a secure signature scheme based on the assumption that CDHP is intractable. Though the recipient can generate random collisions for the chameleon hash function, it is meaningless since the judge can detect this forgery after the signer provides a different collision.
- *Non-transferability*: Note that the semantic security of a chameleon hashing scheme implies the non-transferability of the corresponding chameleon signature scheme [1]. Therefore, in our scheme, the recipient cannot transfer a signature of the signer to convince any third party.
- *Non-repudiation*: Given a valid signature $\sigma = (m, g^a, y_R^a, \mathrm{SIGN}_{x_S}(h))$, the signer cannot generate a valid hash collision $(m', g^{a'}, y_R^{a'})$ which satisfies $h = \mathrm{Hash}(m', g^{a'}, y_R^{a'})$ and $m \neq m'$ because it is equivalent to computing the CDHP in G.
- *Deniability*: It is ensured by the denial protocol.
- *Message hiding*: Given a forgery $\sigma' = (m', g^{a'}, y_R^{a'}, \mathrm{SIGN}_{x_S}(h))$ of the recipient, the signer can provide (m, g^a, y_R^a) as the collision in the denial protocol. However, this will reveal the information of the original message m. Note that $T = (g * I)^{x_R^{-1}} = (g^a/g^{a'})^{(m-m')^{-1}}$, the signer can provide any other collision $(m^*, g^{a^*} = g^a T^{m-m^*}, y_R^{a^*} = y_R^a (g * I)^{m-m^*})$ to ensure the confidentiality of the original message m even against the judge.
- *key exposure freeness*: Given a collision (m, g^a, y^a) and $(m', g^{a'}, y^{a'})$, the information of $(g * I)^{x_R^{-1}}$ can be recovered. However, it is impossible for anyone to compute x_R from $(g * I)^{x_R^{-1}}$. Therefore, collision forgery cannot result in the signer recovering the recipient's trapdoor information x_R, which strengthens the property of non-transferability.

□

4.4 Convertibility

In our chameleon signature scheme, it is impossible for the signer to prove which message was the original one, which is similar to the previous schemes [1, 22]. In some applications, it is more desirable that the signer can confirm the original message if required. Our scheme can be converted into a universally verifiable instance as [1]. The signer encrypts the message using a semantically secure probabilistic encryption algorithm ENC and includes the ciphertext in the signature.[5] That is, the signature σ for the message m becomes:

$$\sigma = (m, I, g^a, y^a, \mathrm{SIGN}_{x_S}(h, \mathrm{ENC}(m))).$$

[5] As [1] stated, the signer can just include the hash of the ciphertext in the signature.

Our scheme can also achieve selective convertibility by having the signer expose the random bits used for the specific probabilistic encryption algorithm, and complete convertibility by exposing the decryption key. For more details, refer to [22].

4.5 Comparison with Two Previous Schemes

The proposed chameleon hashing scheme is almost as efficient as the two previous schemes. Our scheme needs one more modular exponentiation computation in G. Besides, our scheme needs a (very) little more communication cost than the previous schemes. However, we argue that it is worthy to add a (very) little computation and communication expense to overcome the limitation of key exposure. In the Table 1, we present the comparison between our scheme and two previous schemes.

Properties	*Assumption*	*System*	*Key exposure*	*Message hiding*
Krawczyk et al.'s scheme	*DLP*	*CA-based*	*Yes*	*Yes*
Ateniese et al.'s scheme	*RSA*	*ID-based*	*Yes*	*Yes*
Our scheme	*CDHP*	*CA-based*	*No*	*Yes*

Table 1. Comparison with two previous schemes

5 Conclusions

Chameleon signatures are based on well established hash-and-sign paradigm, where a chameleon hash function is used to compute the cryptographic message digest. Chameleon signatures simultaneously provide non-repudiation and non-transferability for the signed message, thus can be used to solve the conflict between authenticity and privacy in the digital signatures. One limitation of the initial chameleon signature scheme is that signature forgery results in the signer recovering the recipient's trapdoor information, *i.e.*, private key. Therefore, the signer can use this information to deny *other* signatures given to the recipient. This creates a strong disincentive for the recipient to forge signatures, partially undermining the concept of non-transferability.

In this paper, we firstly propose a chameleon hashing scheme in the gap Diffie-Hellman group to solve the problem of key exposure. We can prove that the recipient's trapdoor information will never be compromised under the assumption of CDHP in the group is intractable. Moreover, we use the proposed chameleon hashing scheme to design a chameleon signature scheme, which enjoys all advantages of the previous schemes.

References

1. G. Ateniese and B. de Medeiros, *Identity-based chameleon hash and applications*, FC 2004, pp. 63-68, 2004.
2. S. Araki, S. Uehara, and K. Imamura, *The limited verifier signature and its application*, IEICE Trans. Fundamentals, vol.E82-A, No.1, pp. 63-68, 1999.
3. F. Bao, R. Deng and H. Zhu, *Variations of Diffie-Hellman Problem*, ICICS 2003, LNCS 2836, pp.301-312, Springer-Verlag, 2003.
4. D. Boneh and M. Franklin, *Identity-based encryption from the Weil pairings*, Advances in Cryptology-Crypto 2001, LNCS 2139, pp.213-229, Springer-Verlag, 2001.
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. D. Boyar, D. Chaum, and D. Damgård, *Convertible undeniable signatures*, Advances in Cryptology-crypto 1990, LNCS 537, pp.183-195, Springer-Verlag, 1991.
7. J.C. Cha and J.H. Cheon, *An identity-based signature from gap Diffie-Hellman groups*, Public Key Cryptography-PKC 2003, LNCS 2567, pp.18-30, Springer-Verlag, 2003.
8. D. Chaum, *Designated confirmer signatures*, Advances in Cryptology-Eurocrypt 1994, LNCS 950, pp.86-91, Springer-Verlag, 1994.
9. D. Chaum, *Zero-knowledge undeniable signatures*, Advances in Cryptology-Eurocrypt 1990, LNCS 473, pp.458-464, Springer-Verlag, 1991.
10. D. Chaum and H. van Antwerpen, *Undeniable signatures*, Advances in Cryptology-Crypto 1989, LNCS 435, pp.212-216, Springer-Verlag, 1989.
11. X. Chen, F. Zhang and K. Kim, *Limited verifier signature from bilinear pairings*, manuscript, 2004.
12. Y. Dodis, J. Katz, S. Xu and M. Yung, *Key-insulted public-key cryptosystems*, Advances in Cryptology-Eurpcrypt 2002, LNCS 2332, pp.65-82, Springer-Verlag, 2002.
13. Y. Desmedt, C. Goutier, and S. Bengio, *Special uses and abuses of the Fiat-Shamir passport protocol*, Advances in Cryptology-Crypto 1987, LNCS 293, pp.21-39, Springer-Verlag, 1988.
14. Y. Desmedt and M. Yung, *Weaknesses of undenaiable signature schemes*, Advances in Cryptology-Eurpcrypt 1991, LNCS 547, pp.205-220, Springer-Verlag, 1992.
15. S. D. Galbraith, K. Harrison, and D. Soldera, *Implementing the Tate pairings*, ANTS 2002, LNCS 2369, pp.324-337, Springer-Verlag, 2002.
16. S. Galbraith, W. Mao, and K. G. Paterson, *RSA-based undeniable signatures for general moduli*, Advances in CT-RSA 2002, LNCS 2271, pp.200-217, Springer-Verlag, 2002.
17. S. Galbraith and W. Mao, *Invisibility and anonymity of undeniable and confirmer signatures*, Advances in CT-RSA 2003, LNCS 2612, pp.80-97, Springer-Verlag, 2003.
18. S. Gennaro, H. Krawczyk, and T. Rabin, *RSA-based undeniable signatures*, Advances in Cryptology-Crypto 1997, LNCS 1294, pp.132-149, Springer-Verlag, 1997.
19. F. Hess, *Efficient identity based signature schemes based on pairingss*, Proc. 9th Workshop on Selected Areas in Cryptography- SAC 2002, LNCS 2595, Springer-Verlag, pp.310-324, 2002.
20. M. Jakobsson, *Blackmailing using undeniable signatures*, Advances in Cryptology-Eurocrypt 1994, LNCS 950, pp.425-427, Springer-Verlag, 1994.

21. M. Jakobsson, K. Sako, and R. Impagliazzo, *Designated verifier proofs and their applications*, Advances in Cryptology-Eurocrypt 1996, LNCS 1070, pp.143-154, Springer-Verlag, 1996.
22. H. Krawczyk and T. Rabin, *Chameleon hashing and signatures*, Proc. of NDSS 2000, pp.143-154, 2000.
23. B. Libert and J. Quisquater, *ID-based undeniable signatures*, Advances in CT-RSA 2004, LNCS 2694, pp.112-125, Springer-Verlag, 2004.
24. U. Maurer, *Towards the equivalenca of breaking hte Diffie-Hellman protocol and computing discrete logatithms*, Advances in Cryptology-Crypto 1994, LNCS 839, pp.271-281, Springer-Verlag, 1994.
25. R. Peralta, *A simple and fast probabilistic algorithm for computing square roots modulo a prime number*, IEEE Trans. on Information Theory, vol. 32, No. 6, pp.846-847, 1986.
26. R. Steinfeld, L. Bull, H. Wang, and J. Pieprzyk, *Universal designated-verifier signatures*, Advances in Cryptology-Asiacrypt 2003, LNCS 2894, pp.523-542, Springer-Verlag, 2003.
27. A. Sadeghi and M. Steiner, *Assumptions related to discrete logarithms: why subtleties make a real difference*, Advances in Cryptology-Eurocrypt 2001, LNCS 2045, pp.243-260, Springer-Verlag, 2001.