Identity-Based Chameleon Hash Scheme without Key Exposure

Xiaofeng Chen[1], Fangguo Zhang[2], Willy Susilo[3], Haibo Tian[2],
Jin Li[4], and Kwangjo Kim[5]

[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
{isszhfg,tianhb}@mail.sysu.edu.cn

[3] School of Computer Science and Software Engineering,
University of Wollongong, New South Wales 2522, Australia
wsusilo@uow.edu.au

[4] School of Computer Science and Educational Software,
Guangzhou University, Guangzhou 510006, P.R. China
jinli71@gmail.com

[5] Department of Computer Science, KAIST, Daejeon 305-714, Korea
kkj@kaist.ac.kr

Abstract. The notion of chameleon hash function without key exposure plays an important role in designing chameleon signatures. However, all of the existing key-exposure free chameleon hash schemes are presented in the setting of certificate-based systems. In 2004, Ateniese and de Medeiros questioned whether there is an efficient construction for identity-based chameleon hashing without key exposure.

In this paper, we propose the first identity-based chameleon hash scheme without key exposure based on the three-trapdoor mechanism, which provides an affirmative answer to the open problem.

Keywords: Chameleon hashing, Identity-based system, Key exposure.

1 Introduction

Chameleon signatures, introduced by Krawczyk and Rabin [28], 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 the properties of non-repudiation and non-transferability for the signed message as undeniable signatures [3,10,11,12,14,16,21,22,23,25,26,27,33] do, but the former allows for simpler and more efficient realization than the latter. In particular, chameleon signatures are non-interactive and less complicated. More precisely, the signer can generate

R. Steinfeld and P. Hawkes (Eds.): ACISP 2010, LNCS 6168, pp. 200–215, 2010.

© Springer-Verlag Berlin Heidelberg 2010

the chameleon signature without interacting with the designated recipient, and the recipient will be able to verify the signature without the collaboration of the signer. On the other hand, if presented with a forged signature, the signer can deny its validity by only revealing certain values. That is, the forged-signature denial protocol is also non-interactive. 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 (*i.e.*, collision computation) results in the signer recovering the recipient's trapdoor information, *i.e.*, the private key. This is named as the key exposure problem of chameleon hashing, firstly addressed by Ateniese and de Medeiros [1] in 2004. If the signer knows the recipient's trapdoor information, he then can use it to deny *other* signatures given to the recipient. In the worst case, the signer could collaborate with other individuals to invalidate any signatures which were designated to be verified by the same public key. This will create a strong disincentive for the recipient to compute the hash collisions and thus weakens the property of non-transferability.

The original two constructions of chameleon hashing [28] both suffer from the key exposure problem. Ateniese and de Medeiros [1] first introduced the idea of identity-based chameleon hashing to solve this problem. Due to the distinguishing property of identity-based system [37], 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 to 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. However, this kind of transaction-specific chameleon hash scheme still suffers from the key exposure problem unless an identity is never reused in the different chameleon signatures, which requires that the public/secret key pair of the recipient must be changed for each transaction. We argue that this idea only provides a partial solution for the key exposure problem of chameleon hashing.[1]

Chen et al. [17] 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 Medeiros [2] then presented three key-exposure free chameleon hash functions, two based on the RSA assumption, as well as a new construction based on bilinear pairings. Gao et al. [19] proposed a factoring-based chameleon hash scheme without key exposure. Recently, Gao et al. [20] 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. [18] propose the first discrete logarithm based

[1] A trivial solution for the key exposure problem is that the signer changes his key pair frequently in the chameleon signature scheme. However, it is only meaningful in theoretical sense because the key distribution problem arises simultaneously.

key-exposure free chameleon hash scheme without using the GDH groups. However, all of the above constructions are presented in the setting of certificate-based systems where the public key infrastructure (PKI) is required.Zhang et al. [38] presented two identity-based chameleon hash schemes from bilinear pairings, but neither of them is key-exposure free. As pointed out by Ateniese and de Medeiros, the single-trapdoor commitment schemes are not sufficient for the construction of key-exposure free chameleon hashing and the double-trapdoor mechanism [24] can be used to construct either an identity-based chameleon hash scheme or a key-exposure free one, but not both. Therefore, an interesting open problem is whether there is an efficient construction for identity-based chameleon hashing without key exposure [2].

Our Contribution. In this paper, we propose the first identity-based chameleon hash scheme without key exposure, which provides an affirmative answer to the open problem introduced by Ateniese and de Medeiros in 2004. Moreover, the proposed chameleon hash scheme is proved to achieve all the desired security notions in the random oracle model.

Organization. The rest of the paper is organized as follows: Some preliminaries are given in Section 2. The definitions associated with identity-based chameleon hashing are introduced in Section 3. The proposed identity-based key-exposure free chameleon hash scheme and its security analysis are given in Section 4. Finally, conclusions will be made in Section 5.

2 Preliminaries

In this section, we first introduce the basic definition and properties of bilinear pairings and some well-known number-theoretic problems in the gap Diffie-Hellman groups. We then present some proof systems for knowledge of discrete logarithms.

2.1 Bilinear Pairings and Number-Theoretic Problems

Let $\mathbb{G}_1$ be a cyclic additive group generated by P, whose order is a prime q, and $\mathbb{G}_2$ be a cyclic multiplicative group of the same order q. Let a and b be elements of $\mathbb{Z}_q^*$. A bilinear pairing is a map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ with the following properties:

1. Bilinear: $e(aR, bQ) = e(R, Q)^{ab}$ for all $R, Q \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_q^*$.
2. Non-degenerate: There exists R and $Q \in \mathbb{G}_1$ such that $e(R, Q) \neq 1$.
3. Computable: There is an efficient algorithm to compute $e(R, Q)$ for all $R, Q \in \mathbb{G}_1$.

In the following we introduce some problems in $\mathbb{G}_1$.

- Discrete Logarithm Problem (DLP): Given two elements P and Q, to find an integer $n \in \mathbb{Z}_q^*$, such that $Q = nP$ whenever such an integer exists.
- Computation Diffie-Hellman Problem (CDHP): Given P, aP, bP for $a, b \in \mathbb{Z}_q^*$, to compute abP.

- Decision Diffie-Hellman Problem (DDHP): Given P, aP, bP, cP for $a, b, c \in \mathbb{Z}_q^*$, to decide whether $c \equiv ab \bmod q$.

It is proved that the CDHP and DDHP are not equivalent in the group $\mathbb{G}_1$ and thus called a gap Diffie-Hellman (GDH) group. More precisely, we call $\mathbb{G}$ a GDH group if the DDHP can be solved in polynomial time but there is no polynomial time algorithm to solve the CDHP with non-negligible probability. The examples of such a group can be found in supersingular elliptic curves or hyperelliptic curves over finite fields. For more details, see [4,5,6,9,29,30,32,35]. Moreover, we call $< P, aP, bP, cP >$ a valid Diffie-Hellman tuple if $c \equiv ab \bmod q$.

Since the DDHP in the group $\mathbb{G}_1$ is easy, it cannot be used to design cryptosystems in $\mathbb{G}_1$. Boneh and Franklin [6] introduced a new problem in $(\mathbb{G}_1, \mathbb{G}_2, e)$ named Bilinear Diffie-Hellman Problem:

- Bilinear Diffie-Hellman Problem (BDHP): Given P, aP, bP, cP for $a, b, c \in \mathbb{Z}_q^*$, to compute $e(P, P)^{abc} \in \mathbb{G}_2$.

Trivially, the BDHP in $(\mathbb{G}_1, \mathbb{G}_2, e)$ is no harder than the CDHP in $\mathbb{G}_1$ or $\mathbb{G}_2$. However, the converse is still an open problem. On the other hand, currently it seems that there is no polynomial time algorithm to solve the BDHP in $(\mathbb{G}_1, \mathbb{G}_2, e)$ with non-negligible probability. The security of our proposed identity-based chameleon hash scheme without key exposure is also based on the hardness of the BDHP in $(\mathbb{G}_1, \mathbb{G}_2, e)$.

2.2 Proofs of Knowledge

A prover with possession a secret number $x \in \mathbb{Z}_q$ wants to show a verifier that $x = \log_g y$ without exposing x. This is named the proof of knowledge of a discrete logarithm.

This proof of knowledge is basically a Schnorr signature [36] on message (g, y): The prover chooses a random number $r \in_R \mathbb{Z}_q$, and then computes $c = H(g, y, g^r)$, and $s = r - cx \mod q$, where $H : \{0,1\}^* \rightarrow \{0,1\}^k$ is a collision-resistant hash function. The verifier accepts the proof if and only if $c = H(g, y, g^s y^c)$.

Definition 1. *A pair* $(c, s) \in \{0,1\}^k \times \mathbb{Z}_q$ *satisfying the equation*

$$c = H(g, y, g^s y^c)$$

is a proof of knowledge of a discrete logarithm of the element y *to the base* g.

Similarly, we can define the proof of knowledge for the equality of two discrete logarithms: A prover with possession a secret number $x \in \mathbb{Z}_q$ wants to show that $x = \log_g u = \log_h v$ without exposing x.

Chaum and Pedersen [15] firstly proposed the proof as follows: The prover chooses a random number $r \in_R \mathbb{Z}_q$, and then computes $c = H(g, h, u, v, g^r, h^r)$, and $s = r - cx \mod q$, where $H : \{0,1\}^* \rightarrow \{0,1\}^k$ is a collision-resistant hash function. The verifier accepts the proof if and only if $c = H(g, h, u, v, g^s u^c, h^s v^c)$. Trivially, the verifier can efficiently decide whether $< g, u, h, v >$ is a valid Diffie-Hellman tuple with the pair (c, s).

Definition 2. *A pair $(c, s) \in \{0,1\}^k \times \mathbb{Z}_q$ satisfying the equation*

$$c = H(g, h, u, v, g^s u^c, h^s v^c)$$

is a proof of knowledge for the equality of two discrete logarithms of elements u, v with respect to the base g, h.

The identity-based proof of knowledge for the equality of two discrete logarithms, first introduced by Baek and Zheng [8] from bilinear pairings. Define $g = e(P, P)$, $u = e(P, S_{ID})$, $h = e(Q, P)$ and $v = e(Q, S_{ID})$, where P and Q are independent elements of $\mathbb{G}_1$. The following non-interactive protocol presents a proof of knowledge that $\log_g u = \log_h v$: The prover chooses a random number $r \in_R \mathbb{Z}_q$, and then computes $c = H(g, h, u, v, g^r, h^r)$, and $S = rP - cS_{ID}$, where $H : \{0,1\}^* \to \{0,1\}^k$ is a collision-resistant hash function. The verifier accepts the proof if and only if $c = H(g, h, u, v, e(P, S)u^c, e(Q, S)v^c)$.

Definition 3. *A pair $(c, S) \in \{0,1\}^k \times \mathbb{G}_1$ satisfying the equation*

$$c = H(g, h, u, v, e(P, S)u^c, e(Q, S)v^c)$$

is an identity-based proof of knowledge for the equality of two discrete logarithms of elements u, v with respect to the base g, h.

3 Definitions

In this section, we introduce the formal definitions and security requirements of identity-based chameleon hashing [1,2].

3.1 Identity-Based 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 identity-based chameleon hash scheme, the hash key HK is just the identity information ID of the user. A trusted third party called Private Key Generator (PKG) computes the trapdoor key TK associated with HK for the user.

Definition 4. *An identity-based chameleon hash scheme consists of four efficiently computable algorithms:*

- **Setup:** *PKG runs this probabilistic polynomial-time algorithm to generate a pair of keys (SK, PK) defining the scheme. PKG publishes the system parameters SP including PK, and keeps the master key SK secret. The input to this algorithm is a security parameter k.*

- **Extract:** *A deterministic polynomial-time algorithm that, on input the master key SK and an identity string ID, outputs the trapdoor key TK associated to the hash key ID.*
- **Hash:** *A probabilistic polynomial-time algorithm that, on input the master public key PK, an identity string ID, a customized identity L,*[2] *a message m, and a random string r,*[3] *outputs the hash value $h = \mathsf{Hash}(PK, ID, L, m, r)$. Note that h does not depend on TK and we denote $h = \mathsf{Hash}(ID, L, m, r)$ for simplicity throughout this paper.*
- **Forge:** *A deterministic polynomial-time algorithm $\mathcal{F}$ that, on input the trapdoor key TK associated to the identity string ID, a customized identity L, a hash value h of a message m, a random string r, and another message $m' \neq m$, outputs a string r' that satisfies*

  $$h = \mathsf{Hash}(ID, L, m, r) = \mathsf{Hash}(ID, L, m', r').$$

  More precisely,

  $$r' = \mathcal{F}(TK, ID, L, h, m, r, m').$$

  Moreover, if r is uniformly distributed in a finite space $\mathcal{R}$, then the distribution of r' is computationally indistinguishable from uniform in $\mathcal{R}$.

3.2 Security Requirements

The most dangerous attack on the identity-based chameleon hashing is the recovery of either the master key SK or the trapdoor key TK. In this case, the chameleon hash scheme would be totally broken. A weaker attack is that an active adversary computes a collision of the chameleon hashing without the knowledge of the trapdoor TK. In this security model, the adversary is allowed to compromise various users and obtain their secrets, and makes queries to the algorithm **Extract** on the adaptively chosen identity strings except the target one. Therefore, the first essential requirement for identity-based chameleon hashing is the collision resistance against active attackers.

Definition 5. (Collision resistance against active attackers): *Let ID be a target identity string and m be a target message. Let k be the security parameter. The chameleon hash scheme is collision resistance against active attackers if, for all non-constant polynomials $f_1()$ and $f_2()$, there exists no efficient algorithm $\mathcal{A}$ that, on input a customized identity L, outputs a message $m' \neq m$, and two random strings r and r' such that $\mathsf{Hash}(ID, L, m', r') = \mathsf{Hash}(ID, L, m, r)$, with non-negligible probability. Suppose that $\mathcal{A}$ runs in time less than $f_1(k)$, and makes at most $f_2(k)$ queries to the* **Extract** *oracle on the adaptively chosen identity strings other than ID.*

[2] A customized identity is actually a label for each transaction. For example, we can let $L = ID_S||ID_R||ID_T$, where ID_S, ID_R, and ID_T denote the identity of the signer, recipient, and transaction, respectively [1].

[3] Note that r can be either a randomly chosen element in a finite space $\mathcal{R}$, or a bijective function of a random variant which is uniformly distributed in a domain $\mathcal{D}$.

The second requirement for identity-based chameleon hashing is the semantic security, *i.e.*, the chameleon hash value does not reveal anything about the possible message that was hashed.

Definition 6. (Semantic security): *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.*

The identity-based chameleon hashing must also be key-exposure free. It was pointed out that all key-exposure free chameleon hash schemes must have (at least) double trapdoors: a master trapdoor, and an ephemeral trapdoor associated with a customized identity [2]. Loosely speaking, key exposure freeness means that even if the adversary $\mathcal{A}$ has obtained polynomially many ephemeral trapdoors associated with the corresponding customized identities, there is no efficient algorithm for $\mathcal{A}$ to compute a new ephemeral trapdoor. Formally, we have the following definition.

Definition 7. (Key exposure freeness): *If a recipient with identity ID has never computed a collision under a customized identity L, then there is no efficient algorithm for an adversary $\mathcal{A}$ to find a collision for a given chameleon hash value $\mathsf{Hash}(ID, L, m, r)$. This must remain true even if the adversary $\mathcal{A}$ 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.*

4 Identity-Based Key-Exposure Free Chameleon Hashing

All of the existing identity-based chameleon hash schemes [1,38] are based on the double-trapdoor mechanism and suffer from the key exposure problem. In more detail, there are two trapdoors in these chameleon hash schemes: One is the master key x of PKG, and the other is the secret key S_{ID} of the user with identity information ID (In identity-based systems, S_{ID} is actually a signature of PKG on message ID with the secret key x). Given a collision of the chameleon hash function, the trapdoor key S_{ID} will be revealed. Ateniese and de Medeiros [2] thus concluded that the double-trapdoor mechanism cannot be used to construct an efficient chameleon hash scheme that is simultaneously identity-based and key-exposure free, but the multiple-trapdoor (more than two, and consecutive trapdoors) mechanism *perhaps* could provide such a construction.

In this section, we first propose an identity-based key-exposure free chameleon hash scheme based on bilinear pairings. There are three consecutive trapdoors in our chameleon hash scheme: The first one is the master key x of PKG, the second one is the secret key $S_{ID} = xH(ID)$ of the user with identity information ID, and the third one is the ephemeral trapdoor $e(H(L), S_{ID})$ for each transaction with the customized identity L. Given a collision of the chameleon hash function, only the ephemeral trapdoor $e(H(L), S_{ID})$ is revealed, but the permanent trapdoors x and S_{ID} still remain secret. Actually, even given polynomially many ephemeral trapdoors $e(H(L_i), S_{ID})$ associated with the label L_i, it is infeasible to compute a new ephemeral trapdoor $e(H(L), S_{ID})$ associated with the label $L \neq L_i$. Trivially, it is more difficult to compute the trapdoor x or S_{ID}. Therefore, the identity information ID and the corresponding secret key S_{ID} can be used repeatedly for different transactions.

4.1 The Proposed Identity-Based Chameleon Hash Scheme

- **Setup:** Let k be a security parameter. Let $\mathbb{G}_1$ be a GDH group generated by P, whose order is a prime q, and $\mathbb{G}_2$ be a cyclic multiplicative group of the same order q. A bilinear pairing is a map $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. Let $H : \{0,1\}^* \to \mathbb{G}_1$ be a full-domain collision-resistant hash function [7,13,34]. PKG picks a random integer $x \in_R \mathbb{Z}_q^*$ and computes $P_{pub} = xP$. The system parameters are $SP = \{\mathbb{G}_1, \mathbb{G}_2, q, e, P, P_{pub}, H, k\}$.
- **Extract:** Given an identity string ID, computes the trapdoor key $S_{ID} = xH(ID) = xQ_{ID}$.
- **Hash:** On input the hash key ID, a customized identity L, a message m, chooses a random integer $a \in_R \mathbb{Z}_q^*$, and computes $r = (aP, e(aP_{pub}, Q_{ID}))$. Our proposed chameleon hash function is defined as

  $$\mathcal{H} = \mathsf{Hash}(ID, L, m, r) = aP + mH(L).$$

  Note that $\mathcal{H}$ does not depend on the trapdoor key S_{ID}. Besides, if a is a uniformly random integer in $\mathbb{Z}_q^*$, then the string $r = (aP, e(aP_{pub}, Q_{ID}))$ can be viewed as a random input of the chameleon hash function $\mathcal{H}$. We argue that a is not an input of $\mathcal{H}$.
- **Forge:** For any valid hash value $\mathcal{H}$, the algorithm $\mathcal{F}$ can be used to compute a string r' with the trapdoor key S_{ID} as follows:

  $$r' = \mathcal{F}(S_{ID}, ID, L, \mathcal{H}, m, aP, e(aP_{pub}, Q_{ID}), m') = (a'P, e(a'P_{pub}, Q_{ID})),$$

  where

  $$a'P = aP + (m - m')H(L),$$
  $$e(a'P_{pub}, Q_{ID}) = e(aP_{pub}, Q_{ID})e(H(L), S_{ID})^{m-m'}.$$

Note that

$$\mathsf{Hash}(ID, L, m', a'P, e(a'P_{pub}, Q_{ID})) = \mathsf{Hash}(ID, L, m, aP, e(aP_{pub}, Q_{ID}))$$

and

$$\begin{aligned} e(a'P_{pub}, Q_{ID}) &= e(a'P, S_{ID}) \\ &= e(aP + (m - m')H(L), S_{ID}) \\ &= e(aP, S_{ID})e(H(L), S_{ID})^{m-m'} \\ &= e(aP_{pub}, Q_{ID})e(H(L), S_{ID})^{m-m'} \end{aligned}$$

Therefore, the forgery is successful. Moreover, if $(aP, e(aP_{pub}, Q_{ID}))$ is uniformly distributed, then the distribution of $(a'P, e(a'P_{pub}, Q_{ID}))$ is computationally indistinguishable from uniform.

Remark 1. Given a string $r = (aP, e(aP_{pub}, Q_{ID}))$, a necessary condition is the equality of two discrete logarithms of elements aP and $e(aP_{pub}, Q_{ID})$ with respect to the base P and $e(P_{pub}, Q_{ID})$, *i.e.*, $\log_P aP{=}\log_{e(P_{pub},Q_{ID})} e(aP_{pub}, Q_{ID})$. Obviously, the holder R of the trapdoor key S_{ID} can be convinced of the fact if the equation $e(aP, S_{ID}) = e(aP_{pub}, Q_{ID})$ holds: If $e(aP, S_{ID}) = e(aP_{pub}, Q_{ID})$ holds, then we have $\log_P aP{=}\log_{e(P,S_{ID})} e(aP, S_{ID}) = \log_{e(P,S_{ID})} e(aP_{pub}, Q_{ID}) = \log_{e(P_{pub},Q_{ID})} e(aP_{pub}, Q_{ID})$.

In the chameleon signatures, it is also essential for any third party without knowing S_{ID} (e.g., a Judge) to verify the validity of r. Due to the identity-based proof of knowledge for the equality of two discrete logarithms in section 2.2, R can prove that $< e(P, P), e(P_{pub}, Q_{ID}), e(aP, P), e(aP_{pub}, Q_{ID}) >$ is a valid Diffie-Hellman tuple. If $< e(P, P), e(P_{pub}, Q_{ID}), e(aP, P), e(aP_{pub}, Q_{ID}) >$ is a valid Diffie-Hellman tuple, then $< e(P, P), e(aP, P), e(P_{pub}, Q_{ID}), e(aP_{pub}, Q_{ID}) >$ is also a valid Diffie-Hellman tuple. So, we have $\log_P aP = \log_{e(P,P)} e(aP, P) = \log_{e(P_{pub},Q_{ID})} e(aP_{pub}, Q_{ID})$. Moreover, it also holds for any other string $r' = (a'P, e(a'P_{pub}, Q_{ID}))$. That is to say, for any given string r', R can prove that $< e(P, P), e(P_{pub}, Q_{ID}), e(a'P, P), e(a'P_{pub}, Q_{ID}) >$ is a valid Diffie-Hellman tuple in a computationally indistinguishable way. For more details, please refer to Appendix A.

4.2 Security Analysis

Theorem 1. *In the random oracle model, the proposed identity-based chameleon hash scheme is collision resistance against active attackers under the assumption that the BDHP in* $(\mathbb{G}_1, \mathbb{G}_2, e)$ *is intractable.*

Proof. Given a random instance $< P, xP, yP, zP >$ of BDHP, the aim of algorithm $\mathcal{B}$ is to compute $e(P, P)^{xyz}$. $\mathcal{B}$ runs the **Setup** algorithm of the proposed identity-based chameleon hash scheme and sets $P_{pub} = xP$. The resulting system parameters $\{\mathbb{G}_1, \mathbb{G}_2, q, e, P, H, k, P_{pub}\}$ are given to the adversary $\mathcal{A}$. The security analysis will view H as a random oracle.

Let ID be the target identity string and m be the target message. Suppose that $\mathcal{A}$ makes at most $f_1(k)$ queries to the **Extract** oracle, where $f_1(k)$ is a non-constant polynomial. $\mathcal{B}$ randomly chooses $b_i \in \mathbb{Z}_q^*$ for $i \in \{1, 2, \cdots, f_1(k)\}$, and responds to the H query and **Extract** query of $\mathcal{A}$ as follows:

$$H(L) = yP$$

$$H(ID_i) = \begin{cases} b_iP, & \text{if } ID_i \neq ID \\ zP, & \text{Otherwise} \end{cases}$$

$$S_{ID_i} = \begin{cases} b_iP_{pub}, & \text{if } ID_i \neq ID \\ \text{“Fail”}, & \text{Otherwise} \end{cases}$$

if $\mathcal{A}$ can output a message $m' \neq m$, and two strings $r = (aP, e(aP_{pub}, Q_{ID}))$ and $r' = (a'P, e(a'P_{pub}, Q_{ID}))$ such that $\mathsf{Hash}(ID, L, m', r') = \mathsf{Hash}(ID, L, m, r)$ in time T with a non-negligible probability ϵ, then $\mathcal{B}$ can compute

$$e(H(L), S_{ID}) = (e(a'P_{pub}, Q_{ID})/e(aP_{pub}, Q_{ID}))^{(m-m')^{-1}}$$

in time T as the solution of the BDHP. The success of probability of $\mathcal{B}$ is also ϵ.

Theorem 2. *The proposed identity-based chameleon hash scheme is semantically secure.*

Proof. Given an identity ID and a customized identity L, there is a one-to-one correspondence between the hash value $\mathcal{H} = \mathsf{Hash}(ID, L, m, r)$ and the string $r = (aP, e(aP_{pub}, Q_{ID}))$ for each message m. Therefore, the conditional probability $\mu(m|\mathcal{H}) = \mu(m|r)$. Note that m and r are independent variables, the equation $\mu(m|\mathcal{H}) = \mu(m)$ holds. Then, we can prove that the conditional entropy $H[m|\mathcal{H}]$ equals the entropy $H[m]$ as follows:

$$\begin{aligned} H[m|\mathcal{H}] = & -\sum_{m}\sum_{\mathcal{H}} \mu(m, \mathcal{H}) \log(\mu(m|\mathcal{H})) = -\sum_{m}\sum_{\mathcal{H}} \mu(m, \mathcal{H}) \log(\mu(m)) \\ = & -\sum_{m} \mu(m) \log(\mu(m)) = H[m]. \end{aligned}$$

Theorem 3. *In the random oracle model, the proposed identity-based chameleon hash scheme is key-exposure free under the assumption that the BDHP in $(\mathbb{G}_1, \mathbb{G}_2, e)$ is intractable.*

Proof. Loosely speaking, the ephemeral trapdoor $e(H(L), S_{ID})$ can be viewed as the partial signature on message L in the Libert and Quisquater's identity-based undeniable signature scheme [31]. Also, in the random oracle model, their undeniable signature scheme is proved secure against existential forgery on adaptively chosen message and ID attacks under the assumption that the BDHP in $(\mathbb{G}_1, \mathbb{G}_2, e)$ is intractable. That is, even if the adversary has obtained polynomially many signatures $e(H(L_j), S_{ID})$ on message L_j, he cannot forge a signature $e(H(L), S_{ID})$ on message $L \neq L_j$. So, our chameleon hash scheme satisfies the property of key exposure freeness.

Now we give the formal proof of our chameleon hash scheme in details. Given a random instance $< P, xP, yP, zP >$ of BDHP, the aim of algorithm $\mathcal{B}$ is to compute $e(P, P)^{xyz}$ using the adversary $\mathcal{A}$. $\mathcal{B}$ firstly provides $\mathcal{A}$ the system parameters $\{\mathbb{G}_1, \mathbb{G}_2, q, e, P, H, k, P_{pub}\}$ such that $P_{pub} = xP$. The security analysis will view H as a random oracle.

Note that in our chameleon hash scheme, the ephemeral trapdoor $e(H(L), S_{ID})$ can be used to compute a collision (m', r') of the given chameleon hash value $\mathcal{H}$ in any desired way. On the other hand, any collision (m', r') will result in the recovery of the ephemeral trapdoor $e(H(L), S_{ID})$. For the ease of explanation, in the following we let the output of the algorithm $\mathcal{F}$ be the ephemeral trapdoor $e(H(L), S_{ID})$ instead of a collision (m', r'), *i.e.*, $\mathcal{F}(\cdot) = e(H(L), S_{ID})$.

Let ID_t and L_t be the target identity and customized identity, respectively. We stress that L_t is a label only related to the target identity ID_t. That is, $(ID_i,\ L_t)$ cannot be the input of the query to oracle $\mathcal{F}$ for any other identity $ID_i \neq ID_t$. Suppose that $\mathcal{A}$ makes at most $f(k)$ queries to the **Extract** oracle, where $f(k)$ is a non-constant polynomial. For each $i \in \{1, 2, \cdots, f(k)\}$, assume that $\mathcal{A}$ makes at most $g_i(k)$ queries to the $\mathcal{F}$ oracle on four-triples $(L_{i_j}, m_{i_j}, a_{i_j}P, e(a_{i_j}P_{pub}, Q_{ID_i}))$ of his choice, where $g_i(k)$ are non-constant polynomials and $j \in \{1, 2, \cdots, g_i(k)\}$. That is, $\mathcal{A}$ could obtain $g_i(k)$ ephemeral trapdoors $e(H(L_{i_j}), S_{ID_i})$ for each $i \in \{1, 2, \cdots, f(k)\}$. At the end of the game, the output of $\mathcal{A}$ is a collision of the hash value $\mathcal{H} = \mathsf{Hash}(ID_t, L_t, m, aP, e(aP_{pub}, Q_{ID_t}))$ where $L_t \neq L_{t_j}$ and $j \in \{1, 2, \cdots, g_t(k)\}$, *i.e.*, a new ephemeral trapdoor $e(H(L_t), S_{ID_t})$ for $H(L_t) \neq H(L_{t_j})$.

$\mathcal{B}$ randomly chooses $b_i \in \mathbb{Z}_q^*$ and $c_{i_j} \in \mathbb{Z}_q^*$ for $i \in \{1, 2, \cdots, f(k)\}$, $j \in \{1, 2, \cdots, g_i(k)\}$, and then responds to the H query, **Extract** query, and $\mathcal{F}$ query of $\mathcal{A}$ as follows:

$$H(L_{i_j}) = \begin{cases} c_{i_j}P, & \text{if } L_{i_j} \neq L_t \\ yP, & \text{Otherwise} \end{cases}$$

$$H(ID_i) = \begin{cases} b_iP, & \text{if } ID_i \neq ID_t \\ zP, & \text{Otherwise} \end{cases}$$

$$S_{ID_i} = \begin{cases} b_iP_{pub}, & \text{if } ID_i \neq ID_t \\ \text{``Fail''}, & \text{Otherwise} \end{cases}$$

$$\mathcal{F}(\cdot) = \begin{cases} e(c_{i_j}P, b_iP_{pub}), & \text{if } ID_i \neq ID_t \\ e(c_{t_j}P_{pub}, zP), & \text{if } ID_i = ID_t \text{ and } L_{i_j} \neq L_t \\ \text{``Fail''}, & \text{if } ID_i = ID_t \text{ and } L_{i_j} = L_t \end{cases}$$

We say $\mathcal{A}$ wins the game if $\mathcal{A}$ outputs a new valid trapdoor $e(H(L_t), S_{ID_t})$ in time T with a non-negligible probability ϵ. Note that $e(H(L_t), S_{ID_t}) = e(P, P)^{xyz}$, so $\mathcal{B}$ can solve the BDHP in time T with the same probability ϵ.

5 Conclusions

Chameleon signatures simultaneously provide the properties of 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. However, the original constructions suffer from the so-called key exposure problem of chameleon hashing. Recently, some constructions of key-exposure free chameleon hash schemes [2,17] are presented using the idea of "Customized Identities" while in the setting of certificate-based systems. Besides, all of the existing identity-based chameleon hash schemes suffer from the key exposure problem. To the best of our knowledge, there seems no research work on the identity-based chameleon hash scheme without key exposure.

In this paper, we propose the first identity-based chameleon hash scheme without key exposure, which gives an affirmative answer for the open problem introduced by Ateniese and de Medeiros in 2004.

Acknowledgement

We are grateful to the anonymous referees of ACISP 2010 for their invaluable suggestions. This work is supported by the National Natural Science Foundation of China (No. 60970144 and 60773202), Doctoral Fund of Ministry of Education of China for New Teachers (No. 20090171120006), Guangdong Natural Science Foundation (No. 8451027501001508), Program of the Science and Technology of Guangzhou, China (No. 2008J1-C231-2) and ARC Future Fellowship (FT0991397).

References

1. Ateniese, G., de Medeiros, B.: Identity-based chameleon hash and applications. In: Juels, A. (ed.) FC 2004. LNCS, vol. 3110, pp. 164–180. Springer, Heidelberg (2004)
2. Ateniese, G., de Medeiros, B.: On the key exposure problem in chameleon hashes. In: Blundo, C., Cimato, S. (eds.) SCN 2004. LNCS, vol. 3352, pp. 165–179. Springer, Heidelberg (2005)
3. Boyar, D., Chaum, D., Damgå, I., Pedersen, T.: Convertible undeniable signatures. In: Menezes, A., Vanstone, S.A. (eds.) CRYPTO 1990. LNCS, vol. 537, pp. 183–195. Springer, Heidelberg (1991)
4. Barreto, P., Kim, H., Lynn, B., Scott, M.: Efficient algorithms for Pairing-based cryptosystems. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 354–368. Springer, Heidelberg (2002)
5. Boneh, D., Lynn, B., Shacham, H.: Short signatures from the Weil pairings. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 514–532. Springer, Heidelberg (2001)
6. Boneh, D., Franklin, M.: Identity-based encryption from the Weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
7. Bellare, M., Rogaway, P.: The exact security of digital signatures-How to sign with RSA and Rabin. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 399–416. Springer, Heidelberg (1996)
8. Baek, J., Zheng, Y.: Identity-based threshold decryption. In: Bao, F., Deng, R., Zhou, J. (eds.) PKC 2004. LNCS, vol. 2947, pp. 248–261. Springer, Heidelberg (2004)
9. Cha, J., Cheon, J.: An identity-based signature from gap Diffie-Hellman groups. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 18–30. Springer, Heidelberg (2002)
10. Chaum, D.: Zero-knowledge undeniable signatures. In: Damgård, I.B. (ed.) EUROCRYPT 1990. LNCS, vol. 473, pp. 458–464. Springer, Heidelberg (1991)
11. Chaum, D.: Designated confirmer signatures. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 86–91. Springer, Heidelberg (1995)
12. Chaum, D., van Antwerpen, H.: Undeniable signatures. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 212–216. Springer, Heidelberg (1990)
13. Coron, J.: On the exact security of full domain hash. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 229–235. Springer, Heidelberg (2000)
14. Chaum, D., van Heijst, E., Pfitzmann, B.: Cryptographically strong undeniable signatures, unconditionally secure for the signer. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 470–484. Springer, Heidelberg (1992)

15. Chaum, D., Pedersen, T.: Wallet databases with observers. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 89–105. Springer, Heidelberg (1993)
16. Camenisch, J., Michels, M.: Confirmer signature schemes secure against adaptive adversaries. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 243–258. Springer, Heidelberg (2000)
17. Chen, X., Zhang, F., Kim, K.: Chameleon hashing without key exposure. In: Zhang, K., Zheng, Y. (eds.) ISC 2004. LNCS, vol. 3225, pp. 87–98. Springer, Heidelberg (2004)
18. Chen, X., Zhang, F., Tian, H., Wei, B., Kim, K.: Key-exposure free chameleon hashing and signatures based on discrete logarithm systems, Cryptology ePrint Archive: Report 2009/035 (2009)
19. Gao, W., Wang, X., Xie, D.: Chameleon hashes without key exposure based on factoring. Journal of Computer Science and Technology 22(1), 109–113 (2007)
20. Gao, W., Li, F., Wang, X.: Chameleon hash without key exposure based on Schnorr signature. Computer Standards and Interfaces 31, 282–285 (2009)
21. Galbraith, S., Mao, W., Paterson, K.G.: RSA-based undeniable signatures for general moduli. In: Preneel, B. (ed.) CT-RSA 2002. LNCS, vol. 2271, pp. 200–217. Springer, Heidelberg (2002)
22. Galbraith, S., Mao, W.: Invisibility and anonymity of undeniable and confirmer signatures. In: Joye, M. (ed.) CT-RSA 2003. LNCS, vol. 2612, pp. 80–97. Springer, Heidelberg (2003)
23. Gennaro, S., Krawczyk, H., Rabin, T.: RSA-based undeniable signatures. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 132–149. Springer, Heidelberg (1997)
24. Gennaro, R.: Multi-trapdoor commitments and their applications to proofs of knowledge secure under concurrent man-in-the-middle attacks. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 220–236. Springer, Heidelberg (2004)
25. Jakobsson, M., Sako, K., Impagliazzo, R.: Designated verifier proofs and their applications. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 143–154. Springer, Heidelberg (1996)
26. Kurosawa, K., Heng, S.: 3-move undeniable signature scheme. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 181–197. Springer, Heidelberg (2005)
27. Kurosawa, K., Heng, S.: Relations among security notions for undeniable signature schemes. In: De Prisco, R., Yung, M. (eds.) SCN 2006. LNCS, vol. 4116, pp. 34–48. Springer, Heidelberg (2006)
28. Krawczyk, H., Rabin, T.: Chameleon signatures. In: Proc. of NDSS, pp. 143–154 (2000); A preliminary version can be found at Cryptology ePrint Archive: Report 1998/010
29. Hess, F.: Efficient identity based signature schemes based on pairings. In: Nyberg, K., Heys, H.M. (eds.) SAC 2002. LNCS, vol. 2595, pp. 310–324. Springer, Heidelberg (2003)
30. Joux, A.: The Weil and Tate pairings as building blocks for public key cryptosystems. In: Fieker, C., Kohel, D.R. (eds.) ANTS 2002. LNCS, vol. 2369, pp. 20–32. Springer, Heidelberg (2002)
31. Libert, B., Quisquater, J.: ID-based undeniable signatures. In: Okamoto, T. (ed.) CT-RSA 2004. LNCS, vol. 2964, pp. 112–125. Springer, Heidelberg (2004)
32. Miller, V.: The Weil pairing, and its efficient calculation. Journal of Cryptology 17(4), 235–261 (2004)
33. Monnerat, J., Vaudenay, S.: Generic homomorphic undeniable signatures. In: Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, vol. 3329, pp. 354–371. Springer, Heidelberg (2004)

34. Ogata, W., Kurosawa, K., Heng, S.: The security of the FDH variant of Chaum's undeniable signature scheme. In: Vaudenay, S. (ed.) PKC 2005. LNCS, vol. 3386, pp. 328–345. Springer, Heidelberg (2005)
35. Okamoto, T., Pointcheval, D.: The gap-problems: a new class of problems for the security of cryptographic schemes. In: Kim, K.-c. (ed.) PKC 2001. LNCS, vol. 1992, pp. 104–118. Springer, Heidelberg (2001)
36. Schnorr, C.P.: Efficient signature generation for smart cards. Journal of Cryptology 4(3), 239–252 (1991)
37. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
38. Zhang, F., Safavi-Naini, R., Susilo, W.: ID-based chameleon hashes from bilinear pairings, Cryptology ePrint Archive: Report 2003/208 (2003)

Appendix A: The Resulting Chameleon Signature Scheme

Since chameleon signatures are based on well established hash-and-sign paradigm, we can construct an identity-based chameleon signature scheme by incorporating the proposed identity-based chameleon hash scheme Hash and any secure identity-based signature scheme SIGN.

There are two users, a signer S and a recipient R, in the proposed identity-based chameleon signature scheme. When dispute occurs, a judge J is involved in the scheme. Our signature scheme consists of four efficient algorithms **Setup**, **Extract**, **Sign**, **Verify**, and a specific protocol **Deny**. The algorithms of **Setup** and **Extract** are the same as in section 4.1. Let (S_{ID_S}, ID_S) be the signing/verification key pair of S, and (S_{ID_R}, ID_R) be the trapdoor/hash key pair of R.

Given a message m and a customized identity L, S randomly chooses an integer $a \in_R \mathbb{Z}_q^*$, and computes $r = (aP, e(aP_{pub}, Q_{ID_R}))$. The signature σ for message m is $\sigma = (m, r, L, \mathsf{SIGN}_{S_{ID_S}}(\mathcal{H}))$, where $\mathcal{H} = \mathsf{Hash}(ID_R, L, m, r)$.

Given a signature σ, R first uses his trapdoor key S_{ID_R} to verify whether the equation $e(aP, S_{ID_R}) = e(aP_{pub}, Q_{ID_R})$ holds. If the verification fails, he rejects the signature; else, he computes the chameleon hash value $\mathcal{H}$=$\mathsf{Hash}(ID_R, L, m, r)$ and verifies the validity of $\mathsf{SIGN}_{S_{ID_S}}(\mathcal{H})$ with the verification key ID_S.

When dispute occurs, R provides J a signature σ=$(m', r', L, \mathsf{SIGN}_{S_{ID_S}}(\mathcal{H}))$ and a non-interactive identity-based proof of knowledge Π' for the equality of two discrete logarithms that $\log_{e(P,P)} e(P_{pub}, Q_{ID_R}) = \log_{e(a'P,P)} e(a'P_{pub}, Q_{ID_R})$. If either $\mathsf{SIGN}_{S_{ID_S}}(\mathcal{H})$ or Π' is invalid, J rejects it. Otherwise, J summons S to accept/deny the claim. If S wants to accept the signature, he just confirms to J this fact. Otherwise, he provides a collision of the chameleon hash function as follows:

– If S wants to achieve the property of "message recovery", *i.e.*, he wants to prove which message was the one originally signed. In this case, S provides J the tuple (m, r, Π) as a collision, where Π is a non-interactive proof of knowledge for the equality of two discrete logarithms that $a =$

$\log_{e(P,P)} e(aP, P) = \log_{e(P_{pub},Q_{ID_R})} e(aP_{pub}, Q_{ID_R})$. If and only if $m \neq m'$, $\mathcal{H} = \mathsf{Hash}(ID_R, L, m, r)$, and Π is valid, then J can be convinced that R forged the signature on message m' and S only generated a valid signature on message m.

- If S wants to achieve the property of "message hiding", *i.e.*, he wants to protect the confidentiality of the original message even against the judge. In this case, S provides J the tuple (m'', r'') such that $\mathcal{H} = \mathsf{Hash}(ID_R, L, m'', r'')$ as a collision. Note that given two pairs (m, r) and (m', r') such that $\mathcal{H} = \mathsf{Hash}(ID_R, L, m', r') = \mathsf{Hash}(ID_R, L, m, r)$, S can compute the ephemeral trapdoor $e(H(L), S_{ID_R})$=$(e(a'P_{pub}, Q_{ID_R})/e(aP_{pub}, Q_{ID_R}))^{(m-m')^{-1}}$. Then, for a randomly chosen message m'', the string $r'' = (a''P, e(a''P_{pub}, Q_{ID_R}))$ can be computed as follows: $a''P = aP + (m - m'')H(L)$, $e(a''P_{pub}, Q_{ID_R}) = e(aP_{pub}, Q_{ID_R})e(H(L), S_{ID_R})^{m-m''}$. If R accepts the collision (m'', r''), J can be convinced that R forged the signature on message m' and the original message m is never revealed. Otherwise, R provides a non-interactive knowledge proof that r'' is not valid: Let $r'' = (U, V)$, R provide a value $W \neq V$ and a non-interactive knowledge proof that $\log_{e(P,P)} e(P_{pub}, Q_{ID_R}) = \log_{e(U,P)} W$, then J can be convinced that S generated a valid signature on message m'.[4]

Remark 2. Note that if (g, g^a, g^b, g^{ab}) is a valid Diffie-Hellman tuple, then (g, g^b, g^a, g^{ab}) is also a valid Diffie-Hellman tuple, vice versa. That is, there are two different ways (based on the knowledge a or b, respectively) to prove that (g, g^a, g^b, g^{ab}) is a valid Diffie-Hellman tuple when using the proof of knowledge for the equality of two discrete logarithms: $\log_g g^a = \log_{g^b} g^{ab}$ or $\log_g g^b = \log_{g^a} g^{ab}$. This is the main trick of the **Deny** protocol in our signature scheme. We explain it in more details.

For any random string $r' = (a'P, e(a'P_{pub}, Q_{ID_R}))$, R cannot provide a proof that $\log_P a'P = \log_{e(P_{pub},Q_{ID_R})} e(a'P_{pub}, Q_{ID_R})$. However, R (with the knowledge of S_{ID_R}) could provide a proof that

$$\log_{e(P,P)} e(P_{pub}, Q_{ID_R}) = \log_{e(a'P,P)} e(a'P_{pub}, Q_{ID_R}).$$

That is, $\log_{e(P,P)} e(a'P, P) = \log_{e(P_{pub},Q_{ID_R})} e(a'P_{pub}, Q_{ID_R})$. So, we can easily deduce that $\log_P a'P = \log_{e(P,P)} e(a'P, P) = \log_{e(P_{pub},Q_{ID_R})} e(a'P_{pub}, Q_{ID_R})$. In particular, it is also holds even when $r' = r$. That is, the original input r is totally indistinguishable with any collision r'. Moreover, we stress that it is **NOT** required for R to know the value a' or a in the knowledge proof that $\log_{e(P,P)} e(P_{pub}, Q_{ID_R}) = \log_{e(a'P,P)} e(a'P_{pub}, Q_{ID_R})$.

[4] We must consider the case that R provides the original collision (m', r') (that is, m' is the original message to be signed) while S provides an invalid collision (m'', r'') to cheat J. Note that if $\log_{e(P,P)} e(P_{pub}, Q_{ID_R}) = \log_{e(U,P)} W$, then we have $W = e(U, S_{ID_R}) = e(a''P, S_{ID_R})$. Trivially, $V \neq e(a''P_{pub}, Q_{ID_R})$. This means that the tuple (m'', r'') provide by S is not a valid collision.

On the other hand, note that only S knows the knowledge a and no one knows the knowledge $a' \neq a$. Therefore, only S can provide a proof of knowledge that $a = \log_{e(P,P)} e(aP, P) = \log_{e(P_{pub},Q_{ID_R})} e(aP_{pub}, Q_{ID_R})$, and no one can provide a proof of knowledge that $a'{=}\log_{e(P,P)} e(a'P, P){=}\log_{e(P_{pub},Q_{ID_R})} e(a'P_{pub}, Q_{ID_R})$ when $a' \neq a$. This ensures that S can efficiently prove which message was the original one if he desires.

Remark 3. We can also give a new solution to achieve the property of "message hiding" in the resulting identity-based chameleon signature scheme. S chooses a random integer $\theta \in_R \mathbb{Z}_q^*$ and computes $m'' = \theta m$ and $a'' = \theta a$. Let $\mathcal{H}'' = a''P + m''H(L)$, S then provides J the tuple (m'', r'', Σ, Π) as a collision, where $r'' = (a''P, e(a''P_{pub}, Q_{ID_R}))$, Σ is a non-interactive proof of knowledge of a discrete logarithm that $\theta = \log_{\mathcal{H}} \mathcal{H}''$, and Π is a non-interactive proof of knowledge for the equality of two discrete logarithms that $a'' = \log_P a''P = \log_{e(P_{pub},Q_{ID_R})} e(a''P_{pub}, Q_{ID_R})$. If and only if $m'\mathcal{H}'' \neq m''\mathcal{H}$, and Σ and Π are both valid, then J can be convinced that R forged the signature on message m' and the original message m is still confidential. The reason is as follows: if $\mathcal{H}'' = \theta\mathcal{H}$, then the pair $(m, a) = (\theta^{-1}m'', \theta^{-1}a'')$ is the original tuple of S due to the hardness of discrete logarithm assumption. Otherwise, we could compute the discrete logarithm $\log_P H(L)$ while $H(L)$ can be viewed a random element in $\mathbb{G}_1$. Obviously, $(m, r) = (m, (aP, e(aP_{pub}, Q_{ID_R})))$ is the original input of chameleon hashing. Besides, $m'\mathcal{H}'' \neq m''\mathcal{H}$ implies $m \neq m'$. This means that S is capable of providing a new collision different from (m', r'). Due to the randomness of θ, the original message m is kept secret in the sense of semantic security. The more detailed proof will be presented in the full version of this paper.

Remark 4. Compared with the confirm protocol of the identity-based undeniable signature scheme [31], the **Verify** algorithm in our proposed identity-based chameleon signature scheme is non-interactive, *i.e.*, the recipient can verify the signature without the collaboration of the signer. The **Deny** protocol is also non-interactive in our signature scheme. Moreover, our signature scheme is based on the well established hash-and-sign paradigm and thus can provide more flexible constructions. Another distinguishing advantage of our scheme is that the property of "message hiding" or "message recovery" can be achieved freely by the signer.

Compared with the existing identity-based chameleon signature schemes [1,38], our proposed scheme is as efficient as them in the **Sign** and **Verify** algorithms. While in the **Deny** protocol, it requires a (very) little more computation and communication cost for the *non-interactive* proofs of knowledge. However, none of the schemes [1,38] is key-exposure free. Currently, it seems that our proposed scheme is the unique choice for the efficient and secure identity-based chameleon signature scheme in the real applications.