Information Sciences 265 (2014) 198–210

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

# Identity-based chameleon hashing and signatures without key exposure [☆]

Xiaofeng Chen [a,*], Fangguo Zhang [b], Willy Susilo [c], Haibo Tian [b], Jin Li [d], Kwangjo Kim [e]

[a] State Key Laboratory of Integrated Service Networks (ISN), Xidian University, Xi'an 710071, PR China
[b] School of Information Science and Technology, Sun Yat-sen University, Guangzhou 510275, PR China
[c] School of Computer Science and Software Engineering, University of Wollongong, New South Wales 2522, Australia
[d] School of Computer Science and Educational Software, Guangzhou University, Guangzhou 510006, PR China
[e] Department of Computer Science, KAIST, Daejeon 305-714, Republic of Korea

ARTICLE INFO

*Article history:*
Received 3 August 2010
Received in revised form 22 September 2013
Accepted 14 December 2013
Available online 31 December 2013

*Keywords:*
Chameleon hashing
Identity-based system
Key exposure

ABSTRACT

The notion of chameleon hash function without key exposure plays an important role in designing secure 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. Moreover, we use the proposed chameleon hash scheme to design an identity-based chameleon signature scheme, which achieves all the desired security properties.

## 1. Introduction

Chameleon signatures, introduced by Krawczyk and Rabin [32], 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 [12] 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 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 schemes [32] 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. To illustrate this, we take the

[☆] An extended abstract of this paper has been presented at the 15th Australasian Conference on Information Security and Privacy (ACISP), LNCS 6168, Springer, pp. 200–215, 2010 [19].

[*] Corresponding author. Tel.: +86 2988204749.
*E-mail address:* xfchen@xidian.edu.cn (X. Chen).

0020-0255/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2013.12.020

chameleon signature scheme employed for the Chaum–Pedersen trapdoor commitment as the chameleon hash function for example. More precisely, a potential recipient chooses and publishes a regular discrete logarithm-based public key $y = g^x$, where $g$ is the generator of a cyclic group $\mathbb{G}$ and $x$ is the secret key. Later, a signer with the message $m$ to be signed can compute the chameleon hash value $h = g^m y^r$, where $r$ is an auxiliary integer chosen uniformly at random by the signer. Trivially, if the value of $m$ is larger than the order of the group $\mathbb{G}$, we could first hash the message using a cryptographic hash function such as SHA-1. Then the signer can compute the signature $\sigma = \mathrm{SIGN}(h)$, here SIGN is any provable secure signature scheme. Given the triple $(m, r, \sigma)$, the recipient can verify the validity of the signature. However, any third party could not be convinced of the fact since the recipient is capable of providing any new collision $(m', r')$ such that $h = g^{m'} y^{r'}$. Moreover, if the recipient provides the original pair $(m, r)$, the signer cannot repudiate his signature because he cannot compute a new collision under the assumption of discrete logarithm in $\mathbb{G}$ is intractable. Therefore, the chameleon signature scheme satisfies the properties of non-repudiation and non-transferability. On the other hand, with the two pairs $(m, r)$ and $(m', r')$, the signer can recover the secret key $x$ of the recipient from the equation $h = g^m y^r = g^{m'} y^{r'}$, giving $x = (m' - m)(r - r')^{-1}$. This is a highly undesirable outcome from the recipient's viewpoint. 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. Therefore, a third party is more likely to believe claims made by the recipient about presenting an original (non-forged) signature and thus the property of non-transferability of chameleon signature scheme is weakened.

Ateniese and de Medeiros [1] firstly introduced the idea of identity-based chameleon hashing to solve this problem. Due to the distinguishing property of identity-based systems [41], 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. [21] proposed a factoring-based chameleon hash scheme without key exposure. However, Chen et al. [20] presented some security flaws of the scheme and proposed an improved chameleon hash scheme without key exposure based on factoring. Recently, Gao et al. [22] also claimed to present a key-exposure free chameleon hash scheme based on the Schnorr signature. Nevertheless, it requires an *interactive* protocol between the signer and the recipient and thus violates the basic definition of chameleon hashing and signatures. Besides, Chen et al. [18] propose the first discrete logarithm based key-exposure free chameleon hash scheme without using the GDH groups. However, we argue that all of the above constructions are presented in the setting of certificate-based systems where the public key infrastructure (PKI) is required.

Identity-based systems [41] can be an alternative for certificate-based public key systems in some occasions, especially when efficient key management and moderate security are required. Ateniese and de Medeiros [1] proposed the first identity-based chameleon hashing and used it to design a sealed-bid auction scheme. Zhang et al. [42] presented two identity-based chameleon hash schemes from bilinear pairings. However, none 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 [26] 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 construction for identity-based chameleon hashing 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. We then use the proposed chameleon hash scheme to design an identity-based chameleon signature scheme without key exposure.

### *1.1. Related work*

Digital signature is arguably one of the most significant applications of public key cryptography. The ordinary digital signatures can be verified by any intended recipient with the signer's public key, *i.e.*, universal verifiability. However, it may be undesirable in many business situations that a signature can be verified universally. In the past two decades, there are plenty of researches on the conflict between authenticity and privacy in the digital signatures. The notion of undeniable signatures,

[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.

introduced by Chaum and van Antwerpen [12], is such a kind of digital signature which enables the signer to decide *when* his/her signature can be verified. An extended notion is designated confirmer signatures [11], 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 also important for the signer to decide not only *when* but also *by whom* her signature can be verified. This is the motivation of the concept of designated verifier signatures [28]. 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 (non-transferability). Obviously, the two-party ring signatures [38] can provide an alternative solution for designated verifier signatures. However, we argue that the designated verifier signatures (and two-party ring signatures) do not satisfy the property of non-repudiation, which is different from undeniable signatures and chameleon signatures. Steinfeld et al. [40] introduced an extended notion named universal 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. Similarly, the verifier can be convinced that the signer indeed generated the signature, but cannot transfer the proof to convince any third party.

After initial work of Chaum and van Antwerpen [12], plenty of constructions [3,10,11,14,16,23–25,28,30,31,35] for undeniable signatures based on various assumptions have been proposed. Libert and Quisquater [33] proposed the first provable secure undeniable signatures in the identity-based setting. Trivially, identity-based chameleon signatures could provide more alternative solutions for identity-based undeniable signatures. Unfortunately, both of the constructions for identity-based chameleon signatures [1,42] suffer from the problem of key exposure. Ateniese and de Medeiros [2] thus questioned whether there is an efficient construction for identity-based chameleon hashing without key exposure in 2004.

### 1.2. 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. The resulting identity-based chameleon signature scheme is given in Section 5. Finally, conclusions will be made in Section 6.

## 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 $e: \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ 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 [4–6,9,27,29,34,37]. Moreover, we call $\langle P, aP, bP, cP \rangle$ 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 the 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 [39] on message $(g, y)$: The prover chooses a random number $r \in_R \mathbb{Z}_q$, and then computes $c = H(g, y, g^r)$,[2] and $s = r - cx \bmod q$, where $H: \{0,1\}^* \to \{0,1\}^k$ is a collision-resistant hash function. The verifier accepts the proof if and only if $c = H(g, y, g^s y^c)$.

**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 $s = r - cx \bmod q$, where $H: \{0,1\}^* \to \{0,1\}^k$ is a collision-resistant hash function. The verifier accepts the proof if and only if $c = H(g, h, u, v, g^s u^c, h^s v^c)$. Trivially, the verifier can efficiently decide whether $\langle g, u, h, v\rangle$ 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 $\mathbb{G}_1$, and $S_{ID}$ is the private key of the prover with identity information $ID$. 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 secret/public keys $(SK, PK)$ defining the scheme. PKG publishes the system parameters $SP$ including the public key $PK$, and keeps its secret key $SK$ as the master key. The input to this algorithm is a security parameter $k$.

[2] Note that $H(g, y, g^r)$ means $H(g \| y \| g^r)$, where "$\|$" is the concatenation operation.

- **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 $L$,[3] a message $m$, and a random string $r$,[4] 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 $ID$.

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 [1,42] 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

[3] 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].

[4] 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 $\mathcal{D}$.

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 $e: \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. Let $H: \{0,1\}^* \rightarrow \mathbb{G}_1$ be a full-domain collision-resistant hash function [7,13,36]. 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 $\mathcal{H}$. Furthermore, it is essential to ensure the validity of randomness $r$. That is, the equation $\log_P aP = \log_{e(P_{pub}, Q_{ID})} e(aP_{pub}, Q_{ID})$ must hold. For more details, please refer to Remark 1.

- **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

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

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 (e.g., a Judge) without knowing $S_{ID}$ to verify the validity of $r$. Due to the identity-based knowledge proof for the equality of two discrete logarithms in Section 2.2, $R$ can prove that $\langle e(P,P), e(P_{pub}, Q_{ID}), e(aP, P), e(aP_{pub}, Q_{ID}) \rangle$ is a valid Diffie–Hellman tuple. If $\langle e(P,P), e(P_{pub}, Q_{ID}), e(aP, P), e(aP_{pub}, Q_{ID}) \rangle$ is a valid Diffie–Hellman tuple, then $\langle e(P,P), e(aP, P), e(P_{pub}, Q_{ID}), e(aP_{pub}, Q_{ID}) \rangle$ 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 $\langle e(P,P), e(P_{pub}, Q_{ID}), e(a'P, P), e(a'P_{pub}, Q_{ID}) \rangle$ is a valid Diffie–Hellman tuple in a computationally indistinguishable way. For more details, please refer to Section 5.

### 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 $\langle P, xP, yP, zP\rangle$ 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 $i \in \{1, 2, \ldots, f_1(k)\}$, 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 $\epsilon$, 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 in $(\mathbb{G}_1, \mathbb{G}_2, e)$. The success of probability of $\mathcal{B}$ is also $\epsilon$. □

**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 probability $\mu(\mathcal{H}) = \mu(r)$ and $\mu(\mathcal{H}|m) = \mu(r|m)$. Then, it easily follows that the conditional probability $\mu(m|\mathcal{H}) = \mu(m|r)$. Actually, we have

$$\mu(m|\mathcal{H}) = \frac{\mu(m,\mathcal{H})}{\mu(\mathcal{H})} = \frac{\mu(m)\mu(\mathcal{H}|m)}{\mu(\mathcal{H})} = \frac{\mu(m)\mu(r|m)}{\mu(r)} = \frac{\mu(m,r)}{\mu(r)} = \mu(m|r)$$

Besides, 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:

$$H[m|\mathcal{H}] = -\sum_m\sum_{\mathcal{H}}\mu(m,\mathcal{H})\log(\mu(m|\mathcal{H})) = -\sum_m\sum_{\mathcal{H}}\mu(m,\mathcal{H})\log(\mu(m)) = -\sum_m\mu(m)\log(\mu(m)) = H[m] \qquad \Box$$

**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 [33]. 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 $\langle P, xP, yP, zP\rangle$ 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 $i \in \{1, 2, \ldots, f(k)\}$, assume that $\mathcal{A}$ makes at most $g_i(k)$ queries to the $\mathcal{F}$ oracle on four-tuple $(L_{i_j}, m_{i_j}, a_{i_j}P, e(a_{i_j}P_{pub}, Q_{ID_i}))$ of his choice, where $g_i(k)$ are non-constant polynomials and $j \in \{1, 2, \ldots, g_i(k)\}$. That is, $\mathcal{A}$ could obtain $g_i(k)$ ephemeral trapdoors $e(H(L_{i_j}), S_{ID_i})$ for each $i \in \{1, 2, \ldots, f(k)\}$. At the end of the game, $\mathcal{A}$ outputs 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 $j \in \{1, 2, \ldots, g_t(k)\}$, *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 $i \in \{1, 2, \ldots, f(k)\}$, $j \in \{1, 2, \ldots, g_i(k)\}$, 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 $\epsilon$. Note that $e(H(L_t), S_{ID_t}) = e(P, P)^{xyz}$, so $\mathcal{B}$ can solve the BDHP in $(\mathbb{G}_1, \mathbb{G}_2, e)$ in time $T$ with the same probability $\epsilon$. □

## 5. Identity-based chameleon signature scheme

Chameleon signatures are based on well established hash-and-sign paradigm, and thus we can construct an identity-based chameleon signature scheme without key exposure by incorporating the proposed identity-based chameleon hash scheme Hash and any provable secure identity-based signature scheme SIGN against existential forgery on adaptively chosen message and ID attacks such as [9,27]. There are two users, a signer $S$ and a recipient $R$, in the identity-based chameleon signature scheme. When dispute occurs, a judge $J$ is involved in the scheme. In the following, we present the formal definition of identity-based chameleon signatures.

### 5.1. Precise definition

**Definition 8.** An identity-based chameleon signature scheme without key exposure consists of the following efficient algorithms and a specific denial protocol:

- **Setup:** PKG runs this probabilistic polynomial-time algorithm to generate a pair of secret/public keys $(SK, PK)$ defining the scheme. PKG publishes the system parameters $SP$ including the public key $PK$, and keeps the secret key $SK$ as the master key. 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 $ID$.
- **Sign:** An efficient probabilistic algorithm that, on input the public key $ID_R$ of the recipient $R$, the secret key $S_{ID_S}$ of the signer $S$, a message $m$, a customized identity $L$, and a random integer $a \in \mathbb{Z}_q^*$, outputs a signature $\sigma = \mathsf{SIGN}_{S_{ID_S}}(\mathcal{H})$ on the chameleon hash value $\mathcal{H} = \mathsf{Hash}(ID_R, L, m, r)$, where $r$ is a bijective function of the random variant $a$.
- **Verify:** An efficient deterministic algorithm that, on input the public key $ID_R$ of the recipient $R$, the public key $ID_S$ of the signer, a message $m$, a customized identity $L$, a value $r$, and a chameleon signature $\sigma$, outputs a verification decision $b \in \{0, 1\}$.
- **Deny:** A non-interactive protocol between the signer and the judge. Given a signature $\sigma$ on the message $m'$, the signer computes a different collision $(m^*, r^*)$ and some auxiliary information $\Pi^*$. If and only if $m^* \neq m'$ and $\Pi^*$ is valid, the judge claims that the signature on the message $m'$ is a forgery.

Inherently, a secure (identity-based) chameleon signature scheme should satisfy the following properties [1,17,32]:

- **Unforgeability:** No party can produce a valid chameleon signature that is not previously generated by the signer. Also, the recipient can only produce a forgery of a chameleon signature previously generated by the signer.
- **Non-transferability:** The recipient can not convince a third party that the signer indeed generated a signature on a certain message, thus the signature is not universal verifiable.
- **Non-repudiation:** The signer cannot deny legitimate signature claims.

- **Deniability:** The signer can deny a forgery of the signature.
- **Message hiding:** The signer does not have to reveal the original message to deny the validity of a forgery.
- **Message recovery** (or **Convertibility**)**:** A variant of the chameleon signature can be transformed into a regular signature by the signer.

### 5.2. The proposed signature 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. 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 $S_{ID} = xH(ID) = xQ_{ID}$. 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 $R$.
- **Sign:** 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 on the chameleon hash value $\mathcal{H} = \mathsf{Hash}(ID_R, L, m, r)$ is $\sigma = \mathsf{SIGN}_{S_{ID_S}}(\mathcal{H})$.
- **Verify:** Given a four-tuple $(m, r, L, \sigma)$, $R$ 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 $ID_S$.
- **Deny:** When a dispute occurs, $R$ provides $J$ a four-tuple $(m', r', L, \mathsf{SIGN}_{S_{ID_S}}(\mathcal{H}))$ such that $\mathcal{H} = \mathsf{Hash}(ID_R, L, m', r')$ and a non-interactive identity-based proof of knowledge $\Pi'$ 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 $\Pi'$ 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 triple $(m, r, \Pi)$ as a collision, where $\Pi$ 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 $\Pi$ 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 $J$. 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

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

Given a random 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 $m'$.[5]

**Remark 2.** The **Verify** algorithm in our proposed identity-based chameleon signature scheme is non-interactive, *i.e.*, $R$ can verify the signature without the collaboration of $S$. However, the signature verification must require the collaboration of the signer in the identity-based undeniable signature scheme [33]. That is, it is interactive even if the confirm protocol [33] only uses non-interactive designated-verifier knowledge proof. Moreover, our proposed signature scheme is based on the well established hash-and-sign paradigm and thus can provide more flexible constructions.

**Remark 3.** 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 $\log_P a'P = \log_{e(P_{pub}, Q_{ID_R})} e(a'P_{pub}, Q_{ID_R})$ since he never knows the value of $a'$. 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, this also holds even when $r' = r$. That is, the original input $r$ is totally indistinguishable

[5] We must consider the case that $R$ provides the original collision $(m, r)$ (that is, $(m', r') = (m, r)$) 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.

with any collision $r'$. Moreover, we stress that it is **NOT** required for $R$ to know the value $a'$ (or $a$ in the case of $r' = r$) in the knowledge proof that $\log_{e(P,P)} e(P_{pub}, Q_{ID_R}) = \log_{e(a'P,P)} e(a'P_{pub}, Q_{ID_R})$.

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 4.** Given a valid four-tuple $(m, r, L, \sigma = \mathsf{SIGN}_{S_{ID_S}}(\mathcal{H}))$, $R$ can compute a new collision $(m', r', L')$ of the chameleon hashing for any new chosen $m'$ and $L' \neq L$. That is, $\mathcal{H} = \mathsf{Hash}(ID_R, L, m, r) = \mathsf{Hash}(ID_R, L', m', r')$. However, $R$ cannot use the four-tuple $(m', r', L', \sigma = \mathsf{SIGN}_{S_{ID_S}}(\mathcal{H}))$ to convince any third party. We argue that $R$ (with the trapdoor information $S_{ID_R}$) has the ability to compute any collision of the chameleon hashing. Trivially, $S$ can also deny the signature by providing a four-tuple $(m, r, L, \Pi)$ as a collision, where $\Pi$ is a non-interactive proof of knowledge for the equality of two discrete logarithms that $a = \log_{e(P,P)} e(aP, P) = \log_{e(P_{pub},Q_{ID_R})} e(aP_{pub}, Q_{ID_R})$.

### 5.3. Security analysis

**Theorem 4.** *The proposed chameleon signature scheme satisfies the property of unforgeability.*

**Proof.** Due to the well established hash-and-sign paradigm, no third party can produce a valid chameleon signature of $S$. Otherwise, the adversary can either break the underlying signature scheme SIGN, or find a valid collision of the chameleon hash function Hash. However, SIGN is a provable secure identity-based signature scheme against existential forgery on adaptive chosen message and ID attacks, and Hash is collision resistance against active attackers under the assumption that the BDHP in $(\mathbb{G}_1, \mathbb{G}_2, e)$ is intractable.

On the other hand, it is trivial that $R$ can only produce a forgery of a chameleon signature previously generated by $S$. However, it is meaningless since $J$ can detect this forgery after $S$ provides a different collision of the chameleon hashing. □

**Theorem 5.** *The proposed chameleon signature scheme satisfies the property of non-transferability.*

**Proof.** The semantic security of the proposed chameleon hash scheme implies the non-transferability of the resulting chameleon signature scheme [1]. □

**Theorem 6.** *The proposed chameleon signature scheme satisfies the property of non-repudiation.*

**Proof.** If $R$ provides $J$ a valid four-tuple $(m, r, L, \mathsf{SIGN}_{S_{ID_S}}(\mathcal{H}))$ previously generated by $S$, then $S$ cannot provide a valid collision of the chameleon hash function since it is equivalent to solve the BDHP in $(\mathbb{G}_1, \mathbb{G}_2, e)$ as proved in Theorem 1. Therefore, $S$ must confirm to $J$ that he indeed generated the signature. □

**Theorem 7.** *The proposed chameleon signature scheme satisfies the property of deniability.*

**Proof.** It is ensured by the denial protocol. If the dispute occurs, $S$ can provide $J$ a new collision of the chameleon hash function to deny the signature forgery of $R$. □

**Theorem 8.** *The proposed chameleon signature scheme satisfies the property of message hiding.*

**Proof.** As pointed out in Section 5.2, given two valid collisions $(m, r)$ 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 $e(H(L), S_{ID_R})$ and then provide $J$ a new collision $(m^*, r^*)$ for any randomly chosen message $m^*$. The original message $m$ is never revealed. However, it requires the cooperation of $R$ to prove the fact when $r^*$ is not valid.

In the following, we give a new solution to achieve the property of message hiding while $R$ is never involved in the **Deny** protocol. The trick is that we can use the so-called **blinding** technique for the original collision of chameleon hashing. $S$ chooses a random blinding factor $\theta \in_R \mathbb{Z}_q^*$ and computes $m^* = \theta m$, $a^* = \theta a$, and $\mathcal{H}^* = a^* P + m^* H(L)$, $S$ then provides $J$ the four-tuple $(m^*, r^*, \Sigma, \Pi)$ as a new collision, where $r^* = (a^* P, e(a^* P_{pub}, Q_{ID_R}))$, $\Sigma$ is a non-interactive proof of knowledge of a discrete logarithm that $\theta = \log_{\mathcal{H}} \mathcal{H}^*$, and $\Pi$ is a non-interactive proof of knowledge for the equality of two discrete logarithms that $a^* = \log_{e(P,P)} e(a^* P, P) = \log_{e(P_{pub},Q_{ID_R})} e(a^* P_{pub}, Q_{ID_R})$. If and only if $m' \mathcal{H}^* \neq m^* \mathcal{H}$, and $\Sigma$ and $\Pi$ 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:

**Table 1**
Comparison with identity-based undeniable signature scheme.

| | Scheme [33] | Our scheme |
|---|---|---|
| Computation (**Sign**) | $1P + 1H$ | $1P + 5M + 2H$ |
| Computation (**Verify**) | **S** : $4P + 1M + 1E + 3H$ | **S** : / |
| | **R** : $4P + 1M + 3E + 2H$ | **R** : $3P + 2M + 2H$ |
| Proof Computation (**Deny**) | **S** : $5P + 1M + 4E + 3H$ | **S** : $2P + 2M + 4E + 1H$ |
| Proof Verification (**Deny**) | **R** : $4P + 4E + 2H$ | **J** : $3P + 1M + 2E + 2H$ |
| Assumption | BDHP; Random Oracle | BDHP; Random Oracle |
| Convertibility | Explicit | Explicit |
| Construction | Specific | Flexible |

if $\mathcal{H}^* = \theta\mathcal{H}$, then the pair $(\theta^{-1}m^*, \theta^{-1}a^*)$ is equal to the original tuple $(m, a)$ of $S$ due to the hardness of discrete logarithm assumption. Otherwise, we have two distinct representations of $\mathcal{H}$ with respect to the base $(P, H(L))$. Then we could compute the discrete logarithm $\log_P H(L)$ while $H(L)$ can be viewed a random element in $\mathbb{G}_1$. Besides, $m'\mathcal{H}^* \neq m^*\mathcal{H}$ implies $m \neq m'$. This means that $S$ is capable of providing a new collision $(m, r)$ different from $(m', r')$. Due to the randomness of $\theta$, the original message $m$ is kept secret in the sense of semantic security. □

**Theorem 9.** *The proposed chameleon signature scheme satisfies the property of message recovery.*

**Proof.** The enhanced schemes [1,17,32] can be converted into universally verifiable instances. The trick is that the signer encrypts the message using a semantically secure probabilistic encryption scheme ENC and then includes the ciphertext in the signature. However, as noted in [1], this solution does not provide the recipient with a mechanism for adjudicated convertibility, because the recipient has no guarantee that the signer has encrypted the correct information during the signing step.

In our proposed chameleon signature scheme, note that only $S$ can provide a knowledge proof 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 knowledge proof 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 $a' \neq a$. Therefore, given a valid four-tuple $(m, r, L, \mathsf{SIGN}_{S_{ID_S}}(\mathcal{H}))$ and a proof of knowledge that $a = \log_{e(P,P)} e(aP, P) = \log_{e(P_{pub}, Q_{ID_R})} e(aP_{pub}, Q_{ID_R})$, any verifier can be convinced that the original message to be signed is $m$. That is, our proposed solution provides more efficient and explicit convertibility. □

### 5.4. Comparison

Compared with the existing identity-based chameleon signature schemes [1,42], 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 [1,42] 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.

Since both undeniable signatures and chameleon signatures can simultaneously satisfy the properties of non-repudiation and non-transferability, we compare the proposed identity-based chameleon signature scheme with Libert–Quisquater's identity-based undeniable signature scheme [33]. The **Setup** and **Extract** algorithms are the same in the both schemes. The **Verify** algorithm in our proposed signature scheme is non-interactive, while the confirm protocol of [33] requires the collaboration of the signer to verify the signature. The **Deny** protocol is non-interactive in both signature schemes. However, our proposed scheme is superior to [33] in the computation cost.

Table 1 presents the comparison between Libert–Quisquater's identity-based undeniable signature scheme and our identity-based chameleon signature scheme. We denote by $P$ a computation of the pairing, by $M$ a scalar multiplication in $\mathbb{G}_1$, by $H$ a hash operation, and by $E$ an exponentiation in $\mathbb{G}_2$. We omit other operations such as point addition in both schemes.

## 6. 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. Moreover, we use the proposed chameleon hash scheme to design an identity-based chameleon signature scheme, which achieves all the desired security properties.

## Acknowledgements

We are grateful to the anonymous referees for their invaluable suggestions. This work is supported by the National Natural Science Foundation of China (Nos. 61272455 and 61100224), China 111 Project (No. B08038), the Fundamental Research Funds for the Central Universities (Nos. K50511010001 and JY10000901034), and ARC Future Fellowship (FT0991397).

## References

[1] G. Ateniese, B. de Medeiros, Identity-based chameleon hash and applications, in: FC 2004, LNCS, vol. 3110, Springer-Verlag, 2004, pp. 164–180.
[2] G. Ateniese, B. de Medeiros, On the key exposure problem in chameleon hashes, in: SCN 2004, LNCS, vol. 3352, Springer-Verlag, 2005, pp. 165–179.
[3] D. Boyar, D. Chaum, I. Damgård, T. Pedersen, Convertible undeniable signatures, in: Advances in Cryptology-Crypto 1990, LNCS, vol. 537, Springer-Verlag, 1991, pp. 183–195.
[4] P. Barreto, H. Kim, B. Lynn, M. Scott, Efficient algorithms for pairing-based cryptosystems, in: Advances in Cryptology-Crypto 2002, LNCS, vol. 2442, Springer-Verlag, 2002, pp. 354–368.
[5] D. Boneh, B. Lynn, H. Shacham, Short signatures from the Weil pairings, in: Advances in Cryptology-Asiacrypt 2001, LNCS, vol. 2248, Springer-Verlag, 2001, pp. 514–532.
[6] D. Boneh, M. Franklin, Identity-based encryption from the Weil pairing, in: Advances in Cryptology-Crypto 2001, LNCS, vol. 2139, Springer-Verlag, 2001, pp. 213–229.
[7] M. Bellare, P. Rogaway, The exact security of digital signatures-How to sign with RSA and Rabin, in: Advances in Cryptology-Eurocrypt 1996, LNCS, vol. 1070, Springer-Verlag, 1996, pp. 399–416.
[8] J. Baek, Y. Zheng, Identity-based threshold decryption, in: PKC 2004, LNCS, vol. 2947, Springer-Verlag, 2004, pp. 248–261.
[9] J. Cha, J. Cheon, An identity-based signature from gap Diffie–Hellman groups, in: PKC 2003, LNCS, vol. 2567, Springer-Verlag, 2003, pp. 18–30.
[10] D. Chaum, Zero-knowledge undeniable signatures, in: Advances in Cryptology-Eurocrypt 1990, LNCS, vol. 473, Springer-Verlag, 1991, pp. 458–464.
[11] D. Chaum, Designated confirmer signatures, in: Advances in Cryptology-Eurocrypt 1994, LNCS, vol. 950, Springer-Verlag, 1994, pp. 86–91.
[12] D. Chaum, H. van Antwerpen, Undeniable signatures, in: Advances in Cryptology-Crypto 1989, LNCS, vol. 435, Springer-Verlag, 1989, pp. 212–216.
[13] J. Coron, On the exact security of full domain hash, in: Advances in Cryptology-Crypto 2000, LNCS, vol. 1880, Springer-Verlag, 2000, pp. 229–235.
[14] D. Chaum, E. van Heijst, B. Pfitzmann, Cryptographically strong undeniable signatures, unconditionally secure for the signer, in: Advances in Cryptology-Crypto 1991, LNCS, vol. 576, Springer-Verlag, 1991, pp. 470–484.
[15] D. Chaum, T. Pedersen, Wallet databases with observers, in: Advances in Cryptology-Crypto 1992, LNCS, vol. 740, Springer-Verlag, 1993, pp. 89–105.
[16] J. Camenisch, M. Michels, Confirmer signature schemes secure against adaptive adversaries, in: Advances in Cryptology-Eurocrypt 2000, LNCS, vol. 1870, Springer-Verlag, 2000, pp. 243–258.
[17] X. Chen, F. Zhang, K. Kim, Chameleon hashing without key exposure, in: ISC 2004, LNCS, vol. 3225, Springer-Verlag, 2004, pp. 87–98.
[18] X. Chen, F. Zhang, H. Tian, B. Wei, K. Kim, Discrete logarithm based chameleon hashing and signatures without key exposure, Comput. Electr. Eng. 37 (4) (2011) 614–623.
[19] X. Chen, F. Zhang, W. Susilo, H. Tian, J. Li, K. Kim, Identity-based chameleon hash scheme without key exposure, in: ACISP 2010, LNCS, vol. 6168, Springer-Verlag, 2010, pp. 200–215.
[20] X. Chen, F. Zhang, H. Tian, Y. Ding, Comments and improvements on key-exposure free chameleon hashing based on factoring, in: Inscrypt 2010, LNCS, vol. 6584, Springer-Verlag, 2011, pp. 415–426.
[21] W. Gao, X. Wang, D. Xie, Chameleon hashes without key exposure based on factoring, J. Comput. Sci. Technol. 22 (1) (2007) 109–113.
[22] W. Gao, F. Li, X. Wang, Chameleon hash without key exposure based on Schnorr signature, Comput. Stand. Interf. 31 (2009) 282–285.
[23] S. Galbraith, W. Mao, K.G. Paterson, RSA-based undeniable signatures for general moduli, in: CT-RSA 2002, LNCS, vol. 2271, Springer-Verlag, 2002, pp. 200–217.
[24] S. Galbraith, W. Mao, Invisibility and anonymity of undeniable and confirmer signatures, in: CT-RSA 2003, LNCS, vol. 2612, Springer-Verlag, 2003, pp. 80–97.
[25] S. Gennaro, H. Krawczyk, T. Rabin, RSA-based undeniable signatures, in: Advances in Cryptology-Crypto 1997, LNCS, vol. 1294, Springer-Verlag, 1997, pp. 132–149.
[26] R. Gennaro, Multi-trapdoor commitments and their applications to proofs of knowledge secure under concurrent man-in-the-middle attacks, in: Advances in Cryptology-Crypto 2004, LNCS, vol. 3152, Springer-Verlag, 2004, pp. 220–236.
[27] F. Hess, Efficient identity based signature schemes based on pairings, in: SAC 2002, LNCS, vol. 2595, Springer-Verlag, 2002, pp. 310–324.
[28] M. Jakobsson, K. Sako, R. Impagliazzo, Designated verifier proofs and their applications, in: Advances in Cryptology-Eurocrypt 1996, LNCS, vol. 1070, Springer-Verlag, 1996, pp. 143–154.
[29] A. Joux, The Weil and Tate pairings as building blocks for public key cryptosystems, in: ANTS 2002, LNCS, vol. 2369, Springer-Verlag, 2002, pp. 20–32.
[30] K. Kurosawa, S. Heng, 3-Move undeniable signature scheme, in: Advances in Cryptology-Eurocrypt 2005, LNCS, vol. 3494, Springer-Verlag, 2005, pp. 181–197.
[31] K. Kurosawa, S. Heng, Relations among security notions for undeniable signature schemes, in: SCN 2006, LNCS, vol. 4116, Springer-Verlag, 2006, pp. 34–48.
[32] H. Krawczyk, T. Rabin, Chameleon signatures, in: Proc. of NDSS 2000, A Preliminary Version can be Found at Cryptology ePrint Archive: Report 1998/010, 2000, pp.143–154.
[33] B. Libert, J. Quisquater, ID-based undeniable signatures, in: CT-RSA 2004, LNCS, vol. 2694, Springer-Verlag, 2004, pp. 112–125.
[34] V. Miller, The Weil pairing, and its efficient calculation, J. Cryptol. 17 (4) (2004) 235–261.
[35] J. Monnerat, S. Vaudenay, Generic homomorphic undeniable signatures, in: Advances in Cryptology-Asiacrypt 2004, LNCS, vol. 3329, Springer-Verlag, 2004, pp. 354–371.
[36] W. Ogata, K. Kurosawa, S. Heng, The security of the FDH variant of Chaum's undeniable signature scheme, in: PKC 2005, LNCS, vol. 3386, Springer-Verlag, 2005, pp. 328–345.
[37] T. Okamoto, D. Pointcheval, The gap-problems: a new class of problems for the security of cryptographic schemes, in: PKC 2001, LNCS, vol. 1992, Springer-Verlag, 2001, pp. 104–118.
[38] R. Rivest, A. Shamir, Y. Tauman, How to leak a secret, in: Advances in Cryptology-Asiacrypt 2001, LNCS, vol. 2248, Springer-Verlag, 2001, pp. 552–565.
[39] C.P. Schnorr, Efficient signature generation for smart cards, J. Cryptol. 4 (3) (1991) 239–252.

[40] R. Steinfeld, L. Bull, H. Wang, J. Pieprzyk, Universal designated-verifier signatures, in: Advances in Cryptology-Asiacrypt 2003, LNCS, vol. 2894, Springer-Verlag, 2003, pp. 523–542.

[41] A. Shamir, Identity-based cryptosystems and signature schemes, in: Advances in Cryptology-Crypto 1984, LNCS, vol. 196, Springer-Verlag, 1984, pp. 47–53.

[42] F. Zhang, R. Safavi-Naini, W. Susilo, ID-Based Chameleon Hashes from Bilinear Pairings, Cryptology ePrint Archive: Report 2003/208.