ID-Based Chameleon Hashes from Bilinear Pairings

Fangguo Zhang[1], Reihaneh Safavi-Naini[2] and Willy Susilo[2]

[1] Department of Electronics and Communication Engineering,
Institute of Information Security Technology,
Sun Yat-Sen University, Guangzhou 510275, P.R.China
isdzhfg@zsu.edu.cn
[2] School of Information Technology and Computer Science
University of Wollongong, NSW 2522 Australia
{rei,wsusilo}@uow.edu.au

Abstract. Chameleon hash function is a trapdoor one-way hash function. The ID-based chameleon hash function was first introduced by Ateniese and Medeiros [1]. As discussed by [1], the general advantages of ID-based cryptography over conventional cryptography with respect to key distribution are even more pronounced in a chameleon hashing scheme, because the owner of a public key does not necessarily need to retrieve the associated secret key. In this paper, we propose two new ID-based Chameleon hashing schemes from bilinear pairings. Also we analyze their security and efficiency. Based on these ID-based chameleon hashes, ID-based chameleon signature schemes can be designed.

Key words: Chameleon hash, Chameleon signature, ID-based cryptosystem, Bilinear pairings.

1 Introduction

Chameleon signature schemes were introduced in [9]. Similar to regular digital signature, chameleon signatures can provide undeniable commitment of a signer to the content of a signed document, at the same time, they have the added property that they do not allow the recipient of the signature to pass the signed document to a third party without the consent of the signer. Chameleon signatures are closely related to undeniable signatures [5], but are non-interactive and are based on the hash-and-sign paradigm. The main difference between regular signatures and chameleon signatures is in the type of hash function. Chameleon signatures use a chameleon hash function which is a trapdoor one-way hash function and without the knowledge of the trapdoor behaves like a cryptographic hash function, that is it provides collision resistance and finding the pre-images of hash values is hard. However, with the knowledge of the trapdoor, it is easy to find collisions.

The concept of ID-based cryptosystem was fist introduced by Shamir in [14]. The main idea of such cryptosystem is that the identity information of each user works as his public key, in other words, the user's public key can be calculated directly from his identity rather than being extracted from a certificate issued by a certificate authority (CA). ID-based public key setting can be a good alternative for certificate-based public key setting, especially when efficient key management and moderate security are required. In [1], Ateniese and Medeiros introduced the concept of ID-based chameleon hash function. ID-based cryptography in general has the advantage of easier key distribution when compare to conventional public key cryptography. In the case of chameleon hashing these advantages are multiplied by the fact that the owner of a public key does not necessarily need to retrieve the associated secret key. Therefore, ID-based chameleon hashing can support single-use public keys very efficiently. Ateniese and Medeiros's ID-based chameleon hash function is based on RSA.

In the last couple of years, bilinear pairings have been found various applications in cryptography, they can be used to realize some cryptographic primitives that were previously unknown or impractical [2, 3, 8, 12, 16]. More precisely, they are basic tools for construction of ID-based cryptographic schemes, many ID-based cryptographic schemes have been proposed using them.

In this paper, we propose two new constructions of ID-based chameleon hash from bilinear pairings. Based on these ID-based chameleon hashes, ID-based chameleon signature schemes are proposed.

2 Bilinear Pairing and Some 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. A bilinear pairing is a map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ with the following properties:

P1 *Bilinearity:* $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in \mathbb{G}_1, a, b \in \mathbb{Z}_q$;
P2 *Non-degeneracy:* There exists $P, Q \in \mathbb{G}_1$ such that $e(P, Q) \neq 1$;
P3 *Computability:* There is an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in \mathbb{G}_1$.

When the DDHP (Decision Diffie-Hellman Problem) is easy but the CDHP (Computational Diffie-Hellman Problem) is hard on the group $\mathbb{G}$, we call $\mathbb{G}$ a *Gap Diffie-Hellman* (*GDH*) *group* (We say a problem is *hard* if there does not exist a polynomial time algorithm to solve it). Such groups can be found on supersingular elliptic curves or hyperelliptic curves over finite field, and the bilinear paring can be derived from the Weil or Tate pairing.

Throughout this paper, we define the system parameters are as follows: Let P be a generator of $\mathbb{G}_1$ with order q, the bilinear pairing is given by $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. These system parameter can be obtained using a **GDH Parameter Generator** $\mathcal{IG}$ [2]. Define a cryptographic hash function $H_0 : \{0, 1\}^* \rightarrow \mathbb{G}_1^*$. Denote $params = \{\mathbb{G}_1, \mathbb{G}_2, e, q, P, H_0\}$.

3 Definitions

We assume that all system users are identifiable by a bit-string easily derivable from public knowledge about the individual. We call such string an identity string.

Definition 1 (ID-Based Chameleon Hash [1]). *An ID-based chameleon hashing scheme is defined by a family of efficiently computable algorithms:*

- **Setup:** *A trusted party, called Private Key Generator (PKG), runs this probabilistic algorithm to generate a pair of keys SK and PK defining the scheme. It publishes PK and keeps SK secret.*
- **Extract:** *A deterministic algorithm that, on inputs SK and an identity string ID, outputs the trapdoor information S_{ID} associated to the identity.*
- **Hash:** *A probabilistic algorithm that, on inputs PK, an identity string ID, a message m and a randomness R , outputs a hash value h, i.e., $h = Hash(PK, ID, m, R)$.*
- **Forge:** *An algorithm that, on inputs PK, an identity string ID, the trapdoor information S_{ID} associated with ID, a message m', and a hash value h of a message m and the randomness R, outputs a sequence of random bits R' that correspond to a valid computation of $Hash(PK, ID, m', R')$ yielding the target value h.*

The security of an ID-based chameleon hashing scheme consists of two requirements:

***Resistance to collision forgery by active attacks*:** We follow the definition in [1] and describe it as follows: Let ID be a target identity string. Let m be a target message. The chameleon hashing scheme is secure against (existential) collision forgery by active attacks if, for all four-tuples of nonconstant polynomials f_1, f_2, f_3, f_4, and for large enough k, there is no probabilistic algorithm $\mathcal{A}$ which runs in time less than $f_1(k)$, makes at most $f_2(k)$ queries to an **Extract** oracle (on identity strings other than ID), and succeeds with probability larger than $\frac{1}{f_3(k)}$ in computing randomness R and R', and binary string m', where $m' \neq m$, such that $Hash(PK, ID, m, R) = Hash(PK, ID, m', R')$, where $Hash$ is an instance of the scheme with security parameter $f_4(k)$.

***Semantic security*:** From the hash value it is infeasible to determine which message is likely to have resulted in such value by an application of the hash algorithm.

4 New Constructions of ID-Based Chameleon Hash from Pairings

In this section, we propose two ID-based chameleon hash functions from bilinear pairings.

4.1 Scheme 1

- **Setup:** PKG chooses a random number $s \in \mathbb{Z}_q^*$ and sets $P_{pub} = sP$. Define another cryptographic hash function: $H_1 : \{0,1\}^* \to \mathbb{Z}_q^*$. PKG publishes $\{\mathbb{G}_1, \mathbb{G}_2, e,\ q, \lambda, P,\ P_{pub}, H_0, H_1\}$ and keeps s as the *master-key*, which is known only by itself.
- **Extract:** A user submits his identity information ID to PKG. PKG computes the user's public key as $Q_{ID} = H_0(ID)$, and returns $S_{ID} = sQ_{ID}$ to the user as his private key.
- **Hash:** Given a message m, choose a random element R from $\mathbb{G}_1$, define the hash as

$$Hash(P_{pub}, ID, m, R) = e(R,\ P)e(H_1(m)H_0(ID),\ P_{pub}).$$

- **Forge:**

$$Forge(P_{pub}, ID, S_{ID}, m, R, m') = R' = (H_1(m) - H_1(m'))S_{ID} + R.$$

The forgery is right because of the following equation.

$$\begin{aligned}
&Hash(P_{pub}, ID, m', R') \\
&= e(R',\ P)e(H_1(m')H_0(ID),\ P_{pub}) \\
&= e((H_1(m) - H_1(m'))S_{ID} + R,\ P)e(H_1(m')H_0(ID),\ P_{pub}) \\
&= e((H_1(m) - H_1(m'))S_{ID},\ P)e(R,\ P)e(H_1(m')H_0(ID),\ P_{pub}) \\
&= e((H_1(m) - H_1(m'))H_0(ID),\ P_{pub})e(R,\ P)e(H_1(m')H_0(ID),\ P_{pub}) \\
&= e(R,\ P)e(H_1(m)H_0(ID),\ P_{pub}) \\
&= Hash(P_{pub}, ID, m, R)
\end{aligned}$$

4.2 Scheme 2

- **Setup:** PKG chooses a random number $s \in \mathbb{Z}_q^*$ and sets $P_{pub} = sP$. Define another cryptographic hash function: $H_1 : \{0,1\}^* \to \mathbb{Z}_q^*$. PKG publishes $\{\mathbb{G}_1, \mathbb{G}_2, e,\ q, \lambda, P,\ P_{pub}, H_0, H_1\}$ and keeps s as the *master-key*, which is known only by itself.
- **Extract:** A user submits his identity information ID to PKG. PKG returns

$$S_{ID} = \frac{1}{s + H_1(ID)}P$$

to the user as his private key.

- **Hash:** Given a message m, choose a random element R from $\mathbb{G}_1$, define the hash as

$$Hash(P_{pub}, ID, m, R) = e(P,\ P)^{H_1(m)}e(H_1(ID)P + P_{pub},\ R)^{H_1(m)}.$$

- **Forge:**

$$\begin{aligned}
&Forge(P_{pub}, ID, S_{ID}, m, R, m') \\
&= R' = H_1(m')^{-1}((H_1(m) - H_1(m'))S_{ID} + H_1(m)R).
\end{aligned}$$

The forgery is right because of the following equation.

$$\begin{aligned}
&Hash(P_{pub}, ID, m', R')\\
&= e(P,\ P)^{H_1(m')} e(H_1(ID)P + P_{pub},\ R')^{H_1(m')}\\
&= e(P,\ H_1(m')P) e(H_1(ID)P + P_{pub},\ H_1(m') \cdot H_1(m')^{-1}((H_1(m) - H_1(m'))S_{ID}\\
&\quad +H_1(m)R))\\
&= e(P,\ H_1(m')P) e(H_1(ID)P + P_{pub},\ ((H_1(m) - H_1(m'))S_{ID}) e(H_1(ID) + P_{pub},\\
&\quad H_1(m)R))\\
&= e(P,\ H_1(m')P) e(P,\ ((H_1(m) - H_1(m'))P) e(H_1(ID)P + P_{pub},\ H_1(m)R))\\
&= e(P,\ P)^{H_1(m)} e(H_1(ID)P + P_{pub},\ R)^{H_1(m)}\\
&= Hash(P_{pub}, ID, m, R)
\end{aligned}$$

5 Analysis of the ID-Based Chameleon Hashes

5.1 Security

In the ID-based public key setting of **scheme 1**, we use Boneh, Lynn, and Shacham's [3] short signature scheme (BLS scheme) as the private key extract process. BLS scheme is secure against existential forgery under a chosen-message attack (in the random oracle model) assuming the CDH problem is hard. In the **scheme 2**, the private key extract process is same as in Sakai-Kasahara's ID-based encryption scheme [13]. It can be a signature scheme, and in [16], the authors proved that this signature scheme (ZSS scheme) was secure against existential forgery on adaptive chosen-message attacks (in the random oracle model) assuming the k-weak Computational Diffie-Hellman Problem (k-wCDHP) [10] is hard in $\mathbb{G}_1$.

For the security against collision forgery by active attacks of proposed chameleon hash fuctions, we have the following claims.

Claim 1. The chameleon hashing **scheme 1** is resistant to collision forgery under active attacks, provided that the BLS signature scheme is similarly resistant.
Proof. Given a collision, $Hash(P_{pub}, ID, m, R) = Hash(P_{pub}, ID, m', R')$, it is possible to extract the secret key S_{ID} associated to the public key $Q_{ID} = H_0(ID)$.

From $Hash(P_{pub}, ID, m, R) = Hash(P_{pub}, ID, m', R')$, we have

$$e(R,\ P) e(H_1(m)H_0(ID),\ P_{pub}) = e(R',\ P) e(H_1(m')H_0(ID),\ P_{pub}).$$

So

$$e(R - R',\ P) = e((H_1(m') - H_1(m))H_0(ID),\ P_{pub}) = e((H_1(m') - H_1(m))S_{ID},\ P).$$

Hence

$$S_{ID} = ((H_1(m') - H_1(m))^{-1}(R - R').$$

□

Claim 2. The chameleon hashing **scheme 2** is resistant to collision forgery under active attacks, provided that the ZSS signature scheme is similarly resistant.
Proof. Given a collision, $Hash(P_{pub}, ID, m, R) = Hash(P_{pub}, ID, m', R')$, it is possible to extract the secret key S_{ID} associated to the public key $H_1(ID)$.

From $Hash(P_{pub}, ID, m, R) = Hash(P_{pub}, ID, m', R')$, we have

$$e(P,\ P)^{H_1(m)} e(H_1(ID)P + P_{pub},\ R)^{H_1(m)}$$
$$= e(P,\ P)^{H_1(m')} e(H_1(ID)P + P_{pub},\ R')^{H_1(m')}.$$

So

$$e(P,\ P)^{H_1(m)-H_1(m')} = e(H_1(ID)P + P_{pub},\ H_1(m')R' - H_1(m)R)$$

$$e(P,\ P) = e(H_1(ID)P + P_{pub},\ (H_1(m) - H_1(m'))^{-1}(H_1(m')R' - H_1(m)R).$$

Hence

$$S_{ID} = (H_1(m) - H_1(m'))^{-1}(H_1(m')R' - H_1(m)R).$$

□

For the semantic security, we have the following claim:

Claim 3. The two chameleon hashing schemes are semantically secure.
Proof. The chameleon hashing scheme is said to be semantically secure if, for all identity strings ID, and all pairs of messages m and m', the probability distributions of the random variables $Hash(P_{pub}, ID, m, R)$ and $Hash(P_{pub}, ID, m', R)$ are computationally indistinguishable, i.e., given m, m', $z = Hash(P_{pub}, ID, m, R)$ and $z' = Hash(P_{pub}, ID, m', R)$, an adversary cannot distinguish in polynomial time between (z, z') of any pair of messages m and m'. The proposed hashing schemes satisfies the semantic security. This is because given a hash value z, any identity string ID and any message m, there is exactly one random element $R \in \mathbb{G}_1$, such that $Hash(P_{pub}, ID, m, R)$ equals z (Because of the non-degeneracy of bilinear pairing). Due to the randomness of R, the semantic security follows. □

5.2 Efficiency

For **scheme 1**, to give the chameleon hash of a message m, the sender needs to compute one point scalar multiplication of $\mathbb{G}_1$ and two pairing operations. The computation of pairing requires high cost compared with the computation cost for power operation over the finite fields or on the elliptic curve when the parameters are provided. Using the pre-computation, there will be no pairing computation in the chameleon hash. We pre-compute $a = e(P, P)$ and $b = e(H_0(ID),\ P_{pub})$, then to compute the chameleon hash of a message m, the sender only need to compute one point scalar multiplication of $\mathbb{G}_1$ and two exponentiations in $\mathbb{G}_2$, i.e., $R = rP$, $Hash(P_{pub}, ID, m, R) = a^r b^{H_1(m)}$.

Let $r \in_R Z_q^*$, then rS_{ID} is a random element in $\mathbb{G}_1$, so the sender can pre-computes $b = e(H_0(ID),\ P_{pub})$. To compute the chameleon hash of a message m, he only needs to compute one point scalar multiplication of $\mathbb{G}_1$ and one exponentiations in $\mathbb{G}_2$, i.e., $R = rS_{ID}$, $Hash(P_{pub}, ID, m, R) = b^{r+H_1(m)}$. This is because of

$$\begin{aligned} & b^{r+H_1(m)} \\ &= e(H_0(ID),\ P_{pub})^r e(H_0(ID),\ P_{pub})^{H_1(m)} \\ &= e(rS_{ID},\ P)e(H_0(ID),\ P_{pub})^{H_1(m)} \\ &= e(R,\ P)e(H_0(ID),\ P_{pub})^{H_1(m)} \\ &= Hash(P_{pub}, ID, m, R) \end{aligned}$$

Similar discussion can be used to **scheme 2**, we pre-compute $a = e(P, P)$, then to compute the chameleon hash of a message m, the sender only need to compute one point scalar multiplication of $\mathbb{G}_1$ and one exponentiations in $\mathbb{G}_2$, i.e., $R = rS_{ID}$, $Hash(P_{pub}, ID, m, R) = a^{(r+1)H_1(m)}$.

6 New ID-Based Chameleon Signature Scheme

ID-based undeniable signatures can provide non-repudiation and non-transferability, but usually, they are interactive protocols. In [6], Han, Yeung and Wang proposed an ID-based undeniable signature scheme using pairings. However, in [15], we have shown that their scheme is not secure.

ID-based chameleon signature can achieve the same goals of ID-based undeniable signature and is non-interactive. An ID-based chameleon signature scheme is an ID-based signature computed over the ID chameleon hash of m under the identity of the intended recipient. The recipient can verify that the signature of a certain message m is valid but cannot prove to others that the signer actually signed m and not another message. Indeed, the recipient can find collisions of the chameleon hash function, thus finding a message different from m which would pass the signature verification procedure.

Combining the existed ID-based signature schemes [4, 7, 11, 12] and our ID-based chameleon hashes, we can construct some ID-based chameleon signature schemes. Now, we only give a new ID-based chameleon signature scheme based on Cha-Cheon's [4] ID-based signature and ID-based chameleon hash **scheme 1**.

1. **Setup:** Define another cryptographic hash function: $H_2 : \mathbb{G}_2 \times \mathbb{G}_1 \to \mathbb{Z}_q^*$. The system parameters $params = \{\mathbb{G}_1, \mathbb{G}_2, e, q, P, H_0, H_1, H_2\}$, PKG chooses a random number $s \in \mathbb{Z}_q^*$ and sets $P_{pub} = sP$.
2. **Extract:** Let Alice be the signer with identity public key $Q_A = H_0(ID_A)$ and private key S_A, and Bob be the recipient with identity public key $Q_B = H_0(ID_B)$ and private key S_B.

3. **Signing:** For a message m, Alice chooses a random number $r \in_R \mathbb{Z}_q^*$, and a random element $R \in_R \mathbb{G}_1$, computes $U = rQ_A$ and
$$z = Hash(P_{pub}, ID_B, m, R) = e(R,\ P)e(H_1(m)Q_B,\ P_{pub}).$$
Then, computes $h = H_2(z||U)$ and $V = (h + r)S_A$. The message-signature pair is $\{m, U, V, R\}$.
4. **Verification:** Verify that
$$e(V, P) = e(U + H_2(Hash(P_{pub}, ID_B, m, R)||U)Q_A, P_{pub}).$$

Because Bob can find a message different from m which would pass the signature verification procedure using the forge algorithm of our chameleon hash function, Bob cannot prove to others that the signer Alice actually signed m and not another message. So, this ID-based chameleon signature scheme satisfies the non-transferability.

This ID-based chameleon signature scheme can provide the non-repudiation (so, it can be regarded as an non-interactive ID-based undeniable signature). In case of a dispute on the validity of a chameleon signature $\{m', U', V', R'\}$, Bob can sends $\{m', U', V', R'\}$ to a judge. The judge first checks that whether this $\{m', U', V', R'\}$ satisfies the verification equation, if it's true, then sends them to Alice. If Alice wants to accept this signature, he simply confirms to the judge this fact. If Alice wants to claim that this signature is invalid, he will need to provide a message-signature pair $\{m, U, V, R\}$, here $U = U', V = V'$, i.e., Alice need to provide a collision in the chameleon hash function. Notice that if $\{m', U', V', R'\}$ is invalid, Alice can always provide such collision (m, R), since $\{m, U, V, R\}$ was originally generated by Alice with some (m, R) different then (m', R'). If $\{m', U', V', R'\}$ is valid, Alice cannot find collisions of the chameleon hash function and the signature cannot be repudiated.

The unforgeability (for any third party) of this ID-based chameleon signature scheme (even under the adaptive chosen-message attacks) is based on the security of Cha-Cheon's [4] ID-based signature scheme and our ID-based chameleon hash function.

7 Conclusion

ID-based chameleon hash can be used to construct ID-based chameleon signature scheme which can provide non-repudiation and non-transferability, at the same time, it is non-interactive. We proposed two new ID-based Chameleon hashes from bilinear pairings in this paper. Also we analyzed their security and efficiency. Combining the existed ID-based signature schemes and our ID-based chameleon hashes, we can construct some ID-based chameleon signature schemes.

ID-based chameleon signature can be used to construct sealed-bid auction scheme [1] or electronic voting scheme. For the further works, we try to find some new applications of the proposed ID-based chameleon hash scheme and ID-based chameleon signature scheme.

References

1. G. Ateniese and B. de Medeiros, *Identity-based chameleon hash and applications*, In Proceedings of Financial Cryptography 2004 (FC'04), LNCS, Springer-Verlag. This paper is available at Cryptology ePrint Archive, http://eprint.iacr.org/2003/167/.
2. D. Boneh and M. Franklin, *Identity-based encryption from the Weil pairing*, Advances in Cryptology-Crypto 2001, LNCS 2139, pp.213-229, Springer-Verlag, 2001.
3. D. Boneh, B. Lynn, and H. Shacham, *Short signatures from the Weil pairing*, Advances in Cryptology-Asiacrypt 2001, LNCS 2248, pp.514-532, Springer-Verlag, 2001.
4. J.C. Cha and J.H. Cheon, *An identity-based signature from gap Diffie-Hellman groups*, Public Key Cryptography - PKC 2003, LNCS 2139, pp.18-30, Springer-Verlag, 2003.
5. D. Chaum and H. V. Antwerpen, *Undeniable signatures*, Advances in Cryptology-Crypto '89, LNCS 435, pp.212-217, Springer-Verlag, 1989.
6. S. Han, K.Y. Yeung and J. Wang, *Identity-based confirmer signatures from pairings over elliptic curves*, Proceedings of ACM conference on Electronic commerce citation 2003, San Diego, CA, USA, June 09 - 12, 2003. pp.262-263.
7. F. Hess, *Efficient identity based signature schemes based on pairings*, SAC 2002, LNCS 2595, pp.310-324, Springer-Verlag, 2002.
8. A. Joux, *A one round protocol for tripartite Diffie-Hellman*, ANTS IV, LNCS 1838, pp.385-394, Springer-Verlag, 2000.
9. H. Krawczyk and T. Rabin, *Chameleon signatures.* In Proceedings of NDSS 2000, pp. 143-154, 2000.
10. S. Mitsunari, R. Sakai and M. Kasahara, *A new traitor tracing*, IEICE Trans. Vol. E85-A, No.2, pp.481-484, 2002.
11. K.G. Paterson, *ID-based signatures from pairings on elliptic curves*, Electron. Lett., Vol.38, No.18, pp.1025-1026, 2002.
12. R. Sakai, K. Ohgishi and M. Kasahara, *Cryptosystems based on pairing*, SCIS 2000-C20, Jan. 2000. Okinawa, Japan.
13. R. Sakai and M. Kasahara, *Cryptosystems based on pairing over elliptic curve*, SCIS 2003, 8C-1, Jan. 2003. Japan. This paper is available at Cryptology ePrint Archive, http://eprint.iacr.org/2003/054/.
14. A. Shamir, *Identity-based cryptosystems and signature schemes*, Advances in Cryptology-Crypto 84, LNCS 196, pp.47-53, Springer-Verlag, 1984.
15. F. Zhang, R. Safavi-Naini and W. Susilo, *Attack on Han et al.'s ID-based confirmer (undeniable) signature at ACM-EC'03*, Cryptology ePrint Archive, Report 2003/129, http://eprint.iacr.org/2003/129/.
16. F. Zhang, R. Safavi-Naini and W. Susilo, *An efficient signature scheme from bilinear pairings and its applications*, Public Key Cryptography - PKC 2004, Singapore. LNCS 2947, pp.277-290, Springer-Verlag, 2004.