Chameleon Signature from Bilinear Pairing

Xinjun Du, Ying Wang, Jianhua Ge and Yumin Wang
Key Laboratory of Computer
Networks and Information Security
Xidian University

Abstract:
Chameleon signatures are non-interactive signatures based on a hash-and-sign paradigm, and similar in efficiency to regular signatures. The distinguishing characteristic of chameleon signatures is that there are non-transferable, with only the designated recipient capable of asserting its validity. In this paper, we introduce a new ID-based chameleon hash function based on bilinear pairing and build the ID-based chameleon signature scheme. Compared with the conventional chameleon hashing functions, the owner of a public hash key in the ID-based chameleon hashing scheme does not necessarily need to retrieve the associated secret key. The scheme enjoys all the attributes in the normal chameleon signature and the added characteristics of ID-based cryptography based on bilinear pairing.

Keywords: Digital signatures, bilinear pairing, chameleon hashing

1. Introduction

The conventional digital signatures can be validated by any party, but this may be undesirable in many business and e-commerce situations. Previous work has dealt with the problem of bridging between the contradictory requirements of non-repudiation and controlled dissemination via the notion of *undeniable signatures*. The notion was introduced by Chaum and van Antwerpen [1] and followed by many research works, e.g. [2,3,4,5,6]. The basic paradigm behind this type of signatures is that verification of signature requires the collaboration of signer, so that the latter can control to whom the singed document is being disclosed. The crucial requirement is non-transferable, i.e. a signature issued to a designated recipient cannot be validated by another party. To prevent leaking of information these protocols are based on zero-knowledge proofs and this add to the complexity of the schemes relative to regular digital signatures.

Chameleon signature schemes were introduced in [7] which is a much simple implementation of the notion of undeniable signatures. The main technical novelty of chameleon signatures is in departing from the zero-knowledge paradigm. Unlike undeniable signatures, which also provide non-repudiation and non-transferability, chameleon signatures are non-interactive protocols. More precisely, the signer can generate the chameleon signature without interacting with the designated recipient, and the latter will be able to verify the signature without interacting with the former. Similarly, if presented with a forged signature, the signer can deny its validity by revealing certain values. These values will the original signature and the forged one simultaneously, and the revocation can be universally verified. In other words, the forged-signature denial protocol is also non-interactive. Chameleon signatures are based on the 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.

In this paper, we present a new ID-based chameleon signature scheme using a chameleon hash function from bilinear pairing. The scheme enjoys all the attributes of the chameleon signature and the advantages of ID-based cryptography from Bilinear pairing over the elliptic curve.

The rest of the paper is organized as follows: the next section briefly explains the bilinear pairing and the Decisional Hash Bilinear Diffie-Hellman (DHBD) assumption. Section 3 gives a detailed description of our ID-based chameleon signature scheme. In section 4, a heuristic security analysis is presented. Section 5 concludes this paper.

2. Bilinear maps and the Bilinear Diffie-Hellman Assumption

Let G_1 and G_2 be two cyclic groups of order q for some large prime q. G_1 is a cyclic additive group and G_2 is a cyclic multiplicative group. We assume that the discrete logarithm problems in both G_1 and G_2 are hard. Let $e: G_1 \times G_1 \rightarrow G_2$ be a pairing which satisfies the following conditions:

(1) Bilinear: $e(aP, bQ) = e(P, Q)^{ab}$,for all $P, Q \in G_1$ and all $a, b \in \mathbb{Z}_q^*$;

(2) Non-degenerate: there exists $P \in G_1$ and $Q \in G_1$, such that $e(P, Q) \neq 1$;

(3) Computability: there is an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in G_1$.

We note that the Weil and Tate pairings associated with supersingular elliptic curves or abelian varieties can be modified to create such bilinear maps. We refer to [8,9,10,11] for more details.

BDH Parameter Generator: We say that a randomized algorithm $\mathcal{IG}$ be a BDH parameter generator if (1) $\mathcal{IG}$ takes a security parameter $0 < k \in \mathbb{Z}$, (2) $\mathcal{IG}$ runs in polynomial time in k, and (3) $\mathcal{IG}$ outputs the description of two groups G_1, G_2 and the description of a bilinear map $e: G_1 \times G_1 \rightarrow G_2$ described above.

Decisional Hash Bilinear Diffie-Hellman (DHBDH) problem in $< G_1, G_2, e >$:

Instance: (P, aP, bP, cP, r) for some $a, b, c, r \in \mathbb{Z}_q^*$ and a one way hash function $H: G_2 \rightarrow \mathbb{Z}_q^*$.

Solution: Output *yes* if $r = H(e(P, P)^{abc}) \bmod q$ and output *no* otherwise.

The advantage of any probabilistic, polynomial time, 0/1-valued algorithm $\mathcal{A}$ in solving DHBDH problem in $<G_1, G_2, e>$ is defined to be:

$$Adv_{\mathcal{A}}^{DHBDH} = | \Pr ob[\mathcal{A}(P, aP, bP, cP, r) = 1] - \Pr ob[\mathcal{A}(P, aP, bP, cP, H(e(P,P)^{abc})) = 1] | ,$$

$$a, b, c, r \in_R \mathbb{Z}_q^* .$$

DHBDH assumption: There exists no polynomial time algorithm which can solve the DHBDH problem with non-negligible probability of success. In other words, for every probabilistic, polynomial time, 0/1-valued algorithm $\mathcal{A}$, $Adv_{\mathcal{A}}^{BHBDH} < \frac{1}{m^{\ell}}$ for every fixed $\ell > 0$ and sufficiently large m.

3. ID-based Chameleon Signature Scheme

The ID-based chameleon signatures apply a regular digital signature scheme (such as RSA or DSS) to a special type of hashing called ID-based Chameleon hash functions. The basic idea is to build the signature scheme in such a way that a signature provided by a signer S to a recipient R gives R the ability to forge further signatures of S at will. Clearly this prevents R from proving the validity of $S's$ signature to a third party as he could have produced such a signature by himself.

3.1 ID-based Chameleon Hashing

Here we present an ID-based chameleon hashing scheme from bilinear pairing and based on DHBDH assumption. We assume that all system users are identifiable by a bit-string easily derivable from public knowledge about the individual. Formally, an ID-based chameleon hashing scheme is defined by a family of efficiently computable algorithms: **Setup**, **Extract**, **Hash** and **Forge**.

Setup: A trusted party, Trusted Authorities (TA), works as follows:

Setup 1: Run some BDH parameter generator $\mathcal{IG}$ on input a security parameter k to generate two prime order groups G_1, G_2 and the description of a bilinear map $e: G_1 \times G_1 \to G_2$ described above. Choose an arbitrary generator $P \in G_1$.

Setup 2: Pick a random $s \in \mathbb{Z}_q^*$ and set $P_{pub} = sP$.

Setup 3: Choose a cryptographic hash function $H_1 : \{0,1\}^* \to G_1^*$. Choose a cryptographic hash function $H_2 : \{0,1\}^* \to \mathbb{Z}_q^*$. Choose a cryptographic hash function $H : G_2 \to \{0,1\}^n$ for some n. The system public parameters are $params = < G_1, G_2, e, n, P, P_{pub}, H_1, H_2, H >$. The master-key is $s \in \mathbb{Z}_q^*$.

Extract: A deterministic algorithm run by TA, outputs the trapdoor information B associated to some identity. For a given string $ID \in: \{0,1\}^*$, the algorithm dose: (1) computes $Q_{ID} = H_1(ID) \in G_1^*$, and (2) sets the trapdoor information $B = sQ_{ID}$.

Hash: A probabilistic algorithm that, on inputs the system public parameters $params$, an identity string ID, a random $r \in_R G_1$ and a message m, outputs a hash value h. The algorithm is always run by the signer S and ID is the identity string of the recipient R. The algorithm does:

(1) $Q_{ID} = H_1(ID) \in G_1^*$.

(2) $h = Hash(params, ID, m, r) = H(e(Q_{ID}, P_{pub})^{H_2(m)} e(r, P))$.

Forge: A algorithm that, on inputs the system public parameters $params$, an identity string ID, the trapdoor information B associated with ID, a message $m^{'}$, and a hash value h of a message m, outputs a random $r^{'} \in_R G_1$ that correspond to a valid computation of **Hash** yielding the target value h.

The **Forge** algorithm is:

$$Forge(params, ID, B, m, r, h, m^{'}) = r^{'} = H_2(m)B + r - H_2(m^{'})B.$$

Note indeed that

$$\begin{aligned} Hash(params, ID, m^{'}, r^{'}) &= H(e(Q_{ID}, P_{pub})^{H_2(m^{'})} e(r^{'}, P)) \\ &= H(e(sQ_{ID}, P)^{H_2(m^{'})} e(r^{'}, P)) \\ &= H(e(B, P)^{H_2(m^{'})} e(r^{'}, P)) \\ &= H(e(H_2(m^{'})B, P) e(r^{'}, P)) \\ &= H(e(H_2(m^{'})B + r^{'}, P)) \\ &= H(e(H_2(m^{'})B + H_2(m)B + r - H_2(m^{'})B, P)) \\ &= H(e(H_2(m)B + r, P)) \\ &= H(e(Q_{ID}, P_{pub})^{H_2(m)} e(r, P)) \\ &= Hash(params, ID, m, r) \end{aligned}$$

3.2 ID-based Chameleon Signature Schemes

Here we present in some detail the ID-based chameleon signature scheme. An ID-based chameleon signature is generated by digitally signing a chameleon hash value of message. The digital signature scheme used here is some regular digital signature scheme (such as RSA or DSS). We start by describing the setting for ID-based chameleon signatures. The setting defines the players and the agreement upon functions and keys.

Players: Signer S and recipient R. In addition we shall refer to a judge J who represents a party in charge of settling disputes between S and R, and with whom S is assumed to collaborate.

Functions: The players agree on:

- A digital signature scheme (e.g., RSA, DSS) which defines a set of public and private keys associated with the signer and the usual operations of signing, denoted by $SIGN$, and verification, denoted by $VERIFY$.
- A chameleon hashing function $Hash$ which defines a set of public and private keys associated with the owner of the hash function. This function has been described in Section3.1.

Key:

- The signer S has a public and private signature keys which correspond to the agreed on signature scheme, denoted by VK_S and SK_S, respectively.
- The recipient R has a public and private key as required by the agreement upon chameleon hashing scheme. Here the public key is R's identifier ID and the private key is the trapdoor information $B = sQ_{ID}$ (Section3.1)

ID-based Chameleon Signature Generation- $CHAM - SIG$ **:**

Input of Signer: Message m

Private signing key of S, SK_S

R's chameleon hashing public key, i.e. R's identifier ID

1. Generate the chameleon hash of m by choosing a random $r \in_R G_1$ and computing

$$hash = Hash(params, ID, m, r) = H(e(Q_{ID}, P_{pub})^{H_2(m)} e(r, P))$$

2. Set $sig = SIGN_{SK_S}(hash, ID)$.

3. The signature on the message m consists of $SIG(m) = (m, r, sig)$.

ID-based Chameleon Signature Verification- $CHAM - VER$ **:**

Input: $SIG(m) = (m, r, sig)$

Public verification key of S: VK_S

R's chameleon hashing private key, i.e. R's trapdoor information B

1. Compute $hash = Hash(params, ID, m, r)$

2. output= $\begin{cases} proper & VERIFY_{VK_S}((hash, ID), sig) = valid \\ improper & otherwise \end{cases}$

Dispute:

In case of a dispute on the validity of a signature, R can turn to an authorized judge J. J gets from R a triple $SIG(\widehat{m}) = (\widehat{m}, \hat{r}, \widehat{sig})$.

1. J applies the above $CHAM-VER$ function. If this verification fails then the alleged signature is rejected by J. Otherwise,
2. J summons the signer to deny/accept the claim. J sends to S the triple $SIG(\widehat{m})$.
3. If S wants to claim that the signature is invalid he will need to provide a collision in the chameleon hash function. Otherwise, S simple confirms to the judge this fact.

The following is the process that S generates collision in the hash function.

Generate Collision:

Input: a forgery $SIG(m^{'}) = (m^{'}, r^{'}, sig)$

1. S retrieves the original value m, r used to compute sig. It holds that

   $Hash(params, ID, m, r) = Hash(params, ID, m^{'}, r^{'})$, while $m \neq m^{'}$.

2. S computers $B = \frac{r^{'} - r}{H_2(m) - H_2(m^{'})}$.

3. S chooses any message $\overline{m}$ and computes $\overline{r} = \frac{H_2(m) - H_2(\overline{m})}{H_2(m) - H_2(m^{'})}(r^{'} - r) + r$.

4. Output $(\overline{m}, \overline{r})$.

With the triple $SIG(\overline{m}) = (\overline{m}, \overline{r}, sig)$, S can convince the judge to reject the false signature $SIG(m^{'}) = (m^{'}, r^{'}, sig)$.

4. Security Analysis

First we summarize the security properties that we require from a chameleon signature scheme.

- **Unforgeability.** No third party can produce an (R, S)-proper signature not previously generated by the signer.
- **Non-transferability.** Except for the signer himself, no one can prove to another

party that the signer produced a given signature.

- **Denial.** The signer can convince the judge to reject a forgery signature.
- **Non-repudiation.** The signer cannot convince the judge to reject a signature produced by him.
- **Exposure freeness.** A chameleon signature scheme is exposure free if the signer can deny a false signature without exposing any other message actually signed by him.

If the above properties are satisfied, the ID-based chameleon signature from bilinear pairing is a secure chameleon signature scheme.

Theorem 1. Assuming a secure digital signature scheme and the hardness of hardness of DHBDH problem, the ID-based chameleon signature from bilinear pairing is secure.

Proof.

Unforgeability. No third party can produce an (R,S) -proper $SIG(m)=(m,r,sig)$ not previously generated by the signer, as this requires either to break the underlying regular digital signature scheme, or to find collision in the ID-based chameleon hash function which, in turn, implies settling the DHBDH problem. The recipient also cannot produce a signature with a new component sig, as this requires to break the regular digital signature.

Non-transferability. Given a signature $SIG(m)=(m,r,sig)$ generated by S for R, the recipient cannot convince a third party of its validity. Form the **Forge** produce in the ID-based chameleon hashing scheme (Section3.1), we can see that for every possible message $m^{'}$, R can computer a value $r^{'}=H_2(m)B+r-H_2(m^{'})B$ such that $Hash(params,ID,m^{'},r^{'})=Hash(params,ID,m,r)$. Thus, $(m^{'},r^{'},sig)$ is an (R,S) -proper signature. Furthermore, since for every possible message $m^{'}$ there exists exactly one value $r^{'}$ that produces a proper triple $(m^{'},r^{'},sig)$ then nothing is learned about the value of m from seeing the signature string sig . Thus non-transferability is achieved unconditionally, i.e. in the information theoretic sense.

Non-repudiation. Given a $SIG(m)=(m,r,sig)$ generated by the signer S , S can not generate another (R,S) -proper triple $SIG(m^{'})=(m^{'},r^{'},sig)$ for $m\neq m^{'}$, as this would be equivalent to finding a collision of the ID-based chameleon hash function, which we assume to be infeasible by the hardness of the DHBDH problem.

Exposure freeness. From the **Dispute** process (Section3.2), we can see the signature utilize the a false signature and the original signature to produce false signature for any message with the same component *sig* without leaking anything about the original signature.

5. Conclusion

An ID-based chameleon signature from bilinear pairing is presented in this paper, which enjoys all the attributes in the normal chameleon signature. Additionally, it owns the characteristics of ID-based cryptography based on bilinear pairing. For example, a signer can sign a message to an intended recipient without having to first retrieve the recipient's certificate, because everyone knows the identifier of a recipient can produce the public key of the corresponding ID-based chameleon hash function. The signer can use a different public key for each transaction with a recipient without having to retrieve a new certificate. Only the trusted third party can extract the trapdoor information and the recipient doest not have to know the trapdoor information unless he wants to forge the signature.

Reference:

[1] David Chaum and Hans Van Antwerpen, Undeniable signatures, In Crypto'89, pages 212-217, 1989. LNCS No. 435.

[2] J. Boyar, D. Chaum, I. B. Damgard and T. P. Pedersen, Convertible undeniable signatures, In Advances in Cryptology-CRYPTO'90, volume 537 of LNCS, pages 189-205, Springer-Verlag, 1990.

[3] D. Chaum, Zero-knowledge undeniable signature, In Advances in Cryptology -EURPCRYPT'90, volume 473 of LNCS, pages 458-464, Springer-Verlag, 1990.

[4] D Chaum and H. Antwerpen, Undeniable signatures, In Advances in Cryptology-CRYPTO'89, volume 435 of LNCS, pages 212-216, Springer-Verlag, 1991.

[5] D. Chaum, E. van Heijst and B. Pfitzmann, Cryptographically strong undeniable signatures, unconditionally secure for the signer, In Proc. of Advance in Cryptolgoy-CRYPTO'91, volume 576 of LNCS, pages 470-ff. Springer-Verlag, 1991.

[6] E. van Heijst and T. Pedersen, How to make efficient fail-stop signatures, In Proc. of Advances in Cryptology-EUROCRYPT'92, volume 658 of LNCS, pages 366-377. Springer-Verlag, 1993.

[7] H. Krawczyk and T. Rabin, Chameleon signature, In proceeding of NDDS 2000, pages 143-154, 2000.

[8] 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.

[9] D. Boneh, B. Lynn and H. Shacham, Short signatures form the Weil pairing, In C. Boyd, editor, Advances in Cryptology-Asiacrypt 2001, LNCS 2248, pp.514-532, Springer-Verlag, 2001.

[10] C. Gentry and A. Silverberg, Hierarchical ID-based cryptography, To appear in Cryptology-Asiacrypt 2002.

[11] J. Horwitz and B. Lynn, Toward hierarchical identity-based encryption, Proc. Eurocrypt 2002, LNCS2332, pp.466-481, Springer-Verlag,2002.