International Journal of Applied Mathematics

Volume 29 No. 2 2016, 227-242
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)
doi: http://dx.doi.org/10.12732/ijam.v29i2.7

ID-BASED CHAMELEON HASHING AND CHAMELEON SIGNATURE BASED ON GQ SCHEME

Tejeshwari Thakur[1] §, Birendra Kumar Sharma[2]

[1,2]School of Studies in Mathematics
Pt. Ravishankar Shukla University
Raipur (C.G.), 492010, INDIA

Abstract: In this manuscript, we propose key-exposure free chameleon hash and chameleon signature scheme in the framework of Guillou-Quisquater (GQ) scheme [9]. This scheme provides more flexible construction of efficient key-exposure free chameleon hash and signature scheme. Moreover, one benefit of consequential the chameleon signature scheme is that the property of message hiding or message realization can be achieved without obstacle by the signer, i.e., the signer can efficiently prove which message was the original one if he desires. The proposed scheme presented in this article improve and extend the corresponding schemes of others several authors. It is also implemented in Mathematica 7.0.

AMS Subject Classification: 94A60
Key Words: chameleon hashing, ID-based cryptography, chameleon signature, RSA, Mathematica 7.0

1. Introduction

The digital signature is a very important tool in cryptography. It provides signed message with the capabilities like integration, authentication and non repudiation. Anybody can use the signers public key to prove the authenticity of the signature, but sometimes, the signer may need to keep certain interest,

Received: January 13, 2016

§Correspondence author

and therefore, do not wish their signature to be checked by anyone other than the specified message recipient. Chaum and Van Antwerpen [4] first proposed an undeniable signature to solve the above problem. Undeniable signature requires the collaborate of signers during its verification. Therefore, signer can control whether or not the signed message is open to verification by a recipient, this is known as non-transferability property.

Krawczyk and Rabin [10] proposed a new type of signature scheme called chameleon signature. Chameleon signature is based on the well established hash-and-sign paradigm, where a chameleon hash function is used to calculate the message digest. A chameleon hash function is a trapdoor collision-resistant hash function. It has the same characteristics, such as pre-image and collision-resistance hash function. However, collisions and second pre-images can be simply computed the trapdoor is known. Chameleon signature has the same characteristics of an undeniable signature, that is, it is non-repudiable and non-transferable. It is, in fact, a variation of undeniable signature. In traditional public key cryptography, a digital certificate generated by a trusted third party is needed to ensure the binding between the public key and the users identity. Such system may face the certificate management problem. To solve this problem, Shamir [11] proposed the identity-based (ID-based) cryptosystem based on factoring problem, wherein, the users public key could be computed from his identity and the users secret key is generated by private key generater (PKG). However, the ID-based cryptosystem suffers from the key escrow problem, i.e. the PKG knows all the users secret keys. In 1988, Guillou and Quisquater proposed ID-based signature scheme [9], which is constructed from a zero-knowledge identification protocol. Ateniese and Medeiros [1] introduced the concept of ID-based chameleon hash function. ID-based cryptography in general, has the advantage of easier key distribution as compare to the 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 Medeiross ID-based chameleon hash function is based on RSA assumption. The ID-based chameleon hashing using bilinear pairing is designed by Zhang Naini and Susilo [12]. Again Chen et al. [5] proposed the first key-exposure free chameleon hash function based on bilinear pairing. Ateniese and De Mederious [2] presented three key-exposure free chameleon hashing schemes. Out of these three, only two are key exposure free. Interestingly Ateniese and De Mederious [1] introduced an interesting open problem. Is there an efficient construction for ID-based chameleon hash function without key exposure? Recently, Chen

et al. [6] proposed the first ID-based chameleon hash scheme without key exposure, which gave positive answer for the open problem. However, the scheme is constructed in the setting of Diffie-Hellman group with pairings and thus it was less efficient. Zhan et al.[13], first proposed an ID-based chameleon hash scheme without key exposure based on the RSA assumption. It also answered affirmatively to the open problem possed by Ateniese and De Mederious.

Our Contribution: In this paper we propose key-exposure free chameleon hash and chameleon signature scheme based on GQ scheme. One advantage of the chameleon signature scheme is that the property of message hiding or message recovery can be achieved by the signer. Other result we practically proof proposed algorithm in Mathematica 7.0.

Organization: We describe the preliminaries in Section 2. The algorithm of ID- based chameleon hashing and chameleon signature is introduced in Section 3. Proposed algorithm and security scheme is introduced in the Section 4. The Implementation and efficiency of proposed scheme is given in Section 5. Finally, we conclude our work in Section 6.

2. Preliminaries

In this section, we describe the basic definitions.

- **RSA Problem:** RSA public key (n, e) and a message m.

$$c = m^e (\ mod\ n), \text{ to compute } m.$$

- **RSA Signature:** The signature S for a message m or $H(m)$, with H hash or redundancy function and private key (n, d) to obtain S by exponential: $S = m^d\ mod\ n$ or $S = H(m)^d mod\ n$ to verify a signature S, the public key of the signer, the exponential and check that the message m or $H(m)$ is recovered: $m = S^e (\ mod\ n)$ or $H(m) = S^e (\ mod\ n)$.

3. Definitions

In this section, we first recall the definitions and properties of chameleon hashing and signatures [1, 10], as below:

3.1. ID-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, and collisions for every given input. We present a formal definition of a chameleon hash scheme as follows:

Definition 1. A chameleon hash scheme consists of four efficient algorithms:

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

- **Extract:** A deterministic polynomial-time algorithm that, on input the master key SK and an identity string ID, outputs the trapdoor key TK associated to the hash key ID.

- **Hash:** A probabilistic polynomial-time algorithm that, on input an identity string ID, a customized identity L, a message m, and a random string r, outputs the hashed value $\mathcal{H} = Hash(ID, L, m, r)$. Note that h does not depend on TK.

- **Forge:** A deterministic polynomial-time algorithm F that, on input the trapdoor key TK associated to the identity string ID, a customized identity L, a hash value $\mathcal{H}$ of a message m, a random string r, and another message $m \neq m'$, outputs a string r' that satisfies $\mathcal{H} = Hash(ID, L, m, r) = Hash(ID, L, m', r')$. Moreover, if r is uniformly distributed in a finite space $\mathcal{R}$, then the distribution of r' is computationally indistinguishable from uniform in $\mathcal{R}$.

Definition 2. A secure chameleon hashing scheme satisfy the following properties [1]:

- **Collision Resistance:** Without the knowledge of trapdoor key TK, there exists no efficient algorithm that, on input a message m, a random string r, and another message m', outputs a string r' that satisfy $Hash(ID, m', r') = Hash(ID, m, r)$, with non negligible probability.

- **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 $\mathcal{H}$ equals the total entropy $H[M]$ of the message space.

- **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 A to find a collision for a given chameleon hash value $Hash(ID, L, m, r)$. This must remain true even if the adversary A has oracle access to F and is allowed polynomial 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.

- **Message Hiding:** All identity strings ID, and all customized identity L, assume the recipient has computed a collision (m', r') s.t. $\mathcal{H} = Hash(ID, L, m', r') = Hash(ID, L, m, r)$, where m is the original message that was hashed. Then the signer,upon seeing the claimed values (m', r'), can successfully compute another collision (m'', r'') such that $\mathcal{H} = Hash(ID, L, m'', r'')$, without revealing the message m.

3.2. ID-Based Chameleon Signature

A chameleon signature is generated by digitally signing a chameleon hash value of the message. More precisely, we reproduce its definition as given below:

Definition 1. A chameleon signature scheme consists of the following efficient algorithm and a specific denial protocol:

- **Setup:** PKG run this probabilistic polynomial time algorithm to generated a pair of secret/public key (SK, PK) define the scheme. The system parameters SP including PK, and keeps the master key SK is secret. The input to this algorithm is a security parameter k.

- **Extract:** A deterministic polynomial-time algorithm that, on input the master key SK and identity string ID, output the trapdoor key TK associated to the hash key ID.

- **Signature Generation:** An efficient probabilistic algorithm that, on input the public key ID_R of the recipient R,the secret key SK_{ID_S} of the signer S, a message m, customized identity L and a random integer $r \in Z_q^*$, output a signature $\sigma = SIGN_{SK_{ID_S}}(\mathcal{H})$ on the chameleon hash value $\mathcal{H} = Hash(ID, L, m, r)$.

- **Signature Verification:** An efficient deterministic algorithm that on input the public key ID_R, of the recipient R,the public key ID_S of the signer and customized identity ID, a message m, a random string r, and a signature σ, outputs a verification decision $b \in \{0, 1\}$.

- **Denial Protocol:** Non-interactive protocol between the signer and the judge. Given a chameleon signature (σ, r) on the message m', the signer compute different collision (m', r') and some auxiliary information Σ. If and only if $m \neq m'$ and Σ is valid, the judge claims that the signature σ on the message m' is a forgery.

Definition 2. A secure chameleon signature scheme should satisfy the following properties [1, 5, 10]:

- **Unforgeability:** No party can produce a valid chameleon signature which 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 cannot 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:** In case of a dispute, the signer can compute a new collision to deny the forgery and thus the original message is never revealed.

- **Message Recovery (or Convertibility):** A variant of the chameleon signature can be transformed into a regular signature by the signer. That is, the signer is also able to prove which message is the original one in case of forgery.

3.3. Guillou Quisquater ID-Based Signature Scheme

First we describe in details the specification of the ID-based signature scheme proposed by Guillou and Quisquater [9]. The protocol and each part in the scheme are detailed below:

1. **Setup.** On input of a security parameter k, the master entity generates two random k-bit prime numbers p and q. Then he computes $n = pq$ and chooses at random a prime number e satisfying the $gcd(e, \phi(n)) = 1$, and computes $d = e^{-1} \ mod \ \phi(n)$. Furthermore, the master entity chooses hash functions $H_1 : \{0,1\}^* \rightarrow Z_n^*$ and other hash function is $h : \{0,1\}^* \rightarrow \{0,1\}^n$. The public outputs of this setup algorithm are $params = (k, n, e, H_1, h)$. The secret information stored by the master entity is master key is (p, q, d).

2. **Extract.** When a user with identity $ID \in \{0,1\}^*$ requests for his secret key, the master entity computes a RSA signature on the message $H_1(ID) \in Z_n^*$. That is, he computes $J = H_1(ID)^d \ mod \ n$. Then, this value J is sent to the user throughout a secure channel. The user can verify if the received secret key is consistent by checking if $J^e = H_1(ID) \ mod \ n$.

3. **Signature.** To sign a message $m \in \{0,1\}^*$, a user with identity ID acts as follows:

   - He chooses uniformly at random an element $a \in Z_n^*$.
   - He computes the values $r = a^e \ mod \ n$ and $\mathcal{H} = Hash(ID, m, r)$.
   - Finally, he computes the value $\sigma = a \cdot J^h \ mod \ n$. The resulting signature is $\theta = (m, r, \sigma)$.

4. **Verification.** Given a signature $\sigma = (ID, m, r, \sigma)$, the recipient acts as follows:

   - He computes $\mathcal{H} = Hash(ID, m, r)$.
   - He checks if $\sigma^e = r \cdot H_1(ID) \ mod \ n$.

If the check is correct, then the output of the verification algorithm is 1 (valid signature). Otherwise, the output is 0 (invalid signature).

4. The Proposed Scheme

In this section, we present the construction of an efficient ID-based chameleon hashing and chameleon signature which is key exposure free and based on RSA Algorithm. We first propose chameleon hashing scheme based on RSA Algorithm.

4.1. The Proposed Chameleon Hashing Scheme

1. **Setup Phase:** Let with the security parameter 1^k as input, the master entity generates two random k-bit prime numbers p and q. Then he computes $n = pq$, he chooses at random a prime number e satisfying $gcd(e, \phi(n)) = 1$, and computes $d = e^{-1}\ mod \phi(n)$. Furthermore, the master entity chooses two hash functions H_1, h. The public outputs of this setup algorithm are $params = (k, n, e, H_1, h)$. The secret information stored by the master entity is master-key is (p, q, d).

2. **Extract Phase:** When a user with identity $ID \in \{0,1\}^*$ requests for his secret key, the master entity computes $J = H_1(ID)^{d^{-1}}\ mod\ n$. This value J is sent to the user throughout a secure channel. The user can verify if the received secret key is consistent by checking if $J^{e^{-1}} = H_1(ID)\ mod\ n$.

3. **Hashing Phase:** On input the hash key ID_S and ID_R, a message m, chooses a random integer $a \in Z_N^*$ and compute the value $r = a^e\ mod\ n$. Our proposed chameleon hash function is defined as

$$\mathcal{H} = Hash(ID_S, ID_R, m, r) = r^{e^{-1}} H_1(ID)^{h(m)}\ mod\ n.$$

4. **Forge Phase:** For any hash value $\mathcal{H}$, the algorithm F can be used to compute a string with the trapdoor key SK_{ID} as follows: $r' = r \cdot J^{(h(m)-h(m'))}$ Note that if $Hash(ID, m, r) = Hash(ID, m', r')$ then forgery is successful.

4.1.1. Security Analysis

The above ID-based chameleon hash scheme satisfies security properties such as collision-resistance, key-exposure-freeness, semantic security and message hiding as given below.

Theorem 1. *The proposed chameleon hashing scheme is collision-resistance and key-exposure-freeness under the RSA problem is intractable.*

Proof. Given collisions (m, r) and (m', r'), it is satisfied

$$Hash(ID, m, r) = Hash(ID, m', r'),$$

we have that

$$r'^{e^{-1}} H_1(ID)^{h(m')} = r^{e^{-1}} H_1(ID)^{h(m)} \Rightarrow r'^{e^{-1}}$$

$$= r^{e^{-1}} H_1(ID)^{h(m)-h(m')} = r \cdot J^{h(m)-h(m')}.$$

Let $\triangle = h(m) - h(m')$. We can see that these values are relatively prime, i.e. $gcd(\triangle, e) = 1$. Using the Extending Euclidean Algorithm for the GCD, one computes a and b such that $a \triangle + be = 1$. J can now be extracted:

$$(\frac{r'}{r})^a H_1(ID)^b = J^{a\triangle+be} = J.$$

As H_1 is secure RSA signature on identity string, private key generater cannot compute collision (m', r') without knowledge of the trapdoor. Finally, notice that since revealing collision is equivalent to computing signature the scheme is safe from key exposure. □

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

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

$$\mu(m|\mathcal{H}) = \sum_m \sum_{\mathcal{H}} \mu(m|\mathcal{H}) log(m|\mathcal{H}) = \sum_m \sum_{\mathcal{H}} log(\mu(m))$$
$$= -\sum_m \mu(m) log(\mu(m)) = \mathcal{H}[M].$$

□

Theorem 3. *The proposed ID-based chameleon hash scheme satisfies the property of message hiding.*

Proof. Given the collisions (m, r) and (m', r'), we can compute the trapdoor J same as given in Theorem 1. Then for any message m'', a string r'' can be computed with the trapdoor key SK as follows: $r' = r \cdot J^{h(m)-(m')}$. □

4.2. The Proposed Chameleon Signature Scheme

Now, we give a new ID-based chameleon signature scheme based on (GQ) [9] scheme.

There are two users, a signer S and a recipient R, in the proposed ID-based chameleon signature scheme. When dispute occurs, a judge j is involved in the scheme. Our signature scheme consists of five algorithms Setup, Extract, Sign, Verify, and a specific protocol Deny. The algorithms of Setup and Extract are the same as Section 4.1, we describe the signing and verification phase only.

1. **Signature Phase:** Signer choose a message m by using his secret key SK, a user with identity ID as follows:
   - He chooses uniformly at random an element $a \in Z_N^*$.
   - He computes the values $r = a^e \ mod \ n$ and $\mathcal{H} = H(ID, m, r)$.
   - He computes the value $sig = SIGN_{SK_S}(\mathcal{H}, ID)$.
   - The signature on the message m consists of $SIG(m) = (m, r, sig)$.
2. **Verification Phase:** Given a signature $SIG(m) = (m, r, sig)$, the recipient work as follows:
   - Compute $\mathcal{H} = Hash(ID, m, r) = r^{e^{-1}} \cdot H_1(ID)^{h(m)-h(m')}$

Dispute: The signer has to provide a pair of values, different from (m, r), which would pass the signature verification procedure. If the signer does not provide such a pair then the signature on m is considered valid. If the signer provides a different pair $(m', r') \neq (m, r)$, which passes the signature verification procedure, then the judge can conclude that the recipient has cheated and the signature on m is marked as invalid. As with all ID-based schemes, only the trusted third party can extract the secret key the value J such that $H_1(ID) = J^{e^{-1}} \ mod \ n$. One fundamental feature of an ID-based chameleon signature scheme, computed under a hashed identity $H_1(ID)$, is that the recipient does not have to know the secret SK unless he wants to forge the signature. In case, the recipient may never ask for the secret but still successfully complete all transactions.

4.3. Security Analysis

Theorem 4. *The proposed chameleon signature scheme satisfies the properties of unforgeability, non-transferability, non repudiation, deniability, message hiding, and key exposure freeness.*

Proof. We show the proposed chameleon signature scheme satisfies the above properties.

1. **Unforgeability:** No third party can produce a valid chameleon signature which has not been previously generated by the signer, as this requires either to break the underlying signature scheme $SIGN$, or find a valid collision of the chameleon hash function $\mathcal{H}$. Also, it is trivial that the recipient can only produce a forgery of a chameleon signature previously generated by the signer. However, it is meaningless since the judge can detect this forgery soon after the signer when provides a different collision.

2. **Non-Transferability:** Note that the semantic security of a chameleon hashing scheme implies the non-transferability of the corresponding chameleon signature scheme [1]. Therefore, the recipient cannot transfer a signature of the signer to convince any third party.

3. **Non-Repudiation:** Given a $SIG(m) = (m, r, sig)$ generated by the signer S, the signer cannot generate a valid hash collision (m', r') which satisfies $\mathcal{H} = (ID, m', r')$ and $m \neq m'$ as this would be equivalent to finding a collision of ID-based chameleon hash function, which is infeasible by the hardness of the factoring problem.

4. **Deniability:** The signer can convince the Judge to reject a forgery signature.

5. **Message Hiding:** When a dispute takes place, it is often desirable to protect the confidentiality of the original message even against the Judge. As suggested in [10], whenever the recipient cheats, the Judge can solve any dispute without knowing the message originally signed by the signer. Indeed, it would be enough to reveal any collision of the chameleon hash function to convince the Judge that the recipient is cheated. This can be easily accomplished since the secret trapdoor information associated with the recipients public key is revealed whenever a collision of the chameleon hash function is known. For details, see in the proof 3.

6. **Message Recovery (or Convertibility):** In case of forgery, the recipients key compromise results in the signer being capable of claiming any message as the one originally signed. Moreover, it becomes impossible for the signer to prove which message was the original one. The convertible variant of the basic scheme provides the signer with a noninteractive algorithm to transform any instance of the chameleon signature into a universally verifiable instance. A different possibility is the original message be recoverable even without the signers cooperation. In these cases, it may be necessary to add to the signature some additional information

about the message itself. One possibility is to include in the signature an encryption of the pair (m, r) computed under the public key of the Judge. Thus is, the signature becomes:

$$\sigma = (m, r, SIGN_{SK_S}(\mathcal{H}, ID)),$$

where the public-key encryption is assumed to be semantically secure. In practice, one would sign a hash of the encryption, so as to compute the signature on parameters of fixed size.

□

5. Efficiency

5.1. Implementation of Proposed Algorithm

We describe the efficiency of the proposed algorithm implemented in Mathematica 7.0 as below:

(*Program for Chameleon Signature on RSA*)
(*Setup Generation*)
$FirstPrimeAbove[n] := Block[k, k = n;$
$While[!PrimeQ[k], k = k + 1];$
$Return[k]]$
$p = FirstPrimeAbove[123456789123456789];$
$q = FirstPrimeAbove[987654321987654321];$
$n = pq;$
$Print["n\ is:",\ n];$
$Z = (p^2 - 1)(q^2 - 1)$
$FactorInteger[\%]$
$d = 1713232931$
$GCD[d, Z]$
$e := PowerMod[d, -1, Z]$
(***Extract Generation***)
$ID := ImportString["ID","Bit"]$
(***Signature Generation***)
(***Hash Generation***)
$h := Hash[ID,"SHA512"]$
$a2 = PowerMod[h, -d, n]$

$a3 = Mod[h, n]$
$a4 = PowerMod[a2, -e, n]$
$a5 := RandomInteger[7, 16]$
$r = PowerMod[a5, e, n]$
$m := 12345$
$m' := 67890$
$a6 = Hash[m]$
$a16 = Hash[m']$
$a7 := PowerMod[r, -e, n]$
$a8 := PowerMod[h, Hash[m], n]$
$a18 := PowerMod[h, Hash[m'], n]$
$a9 := Mod[a7 * a8, n]$
$lhs = a9$
$Print["lhsis :", lhs];$
(*Forge Algorithm*)
$b1 := Hash[m] - Hash[m']$
$b2 := PowerMod[h, b1, n]$
$b3 := Mod[r * b2, n]$
$r' := b3$
$c1 := PowerMod[r', -e, n]$
(***verification***)
$c2 := Mod[a18 * c1, n]$
$c3 := a18 * a7 * b2$
$c4 := Mod[c3, n]$
$rhs = c4$
$Print["rhsis :", rhs];$
(***Time***)
$TimeUsed[]$
(***Output***)
$lhsis$: 29582767669031226056312411776734649
$rhsis$: 29582767669031226056312411776734649
$0.264 second CPU time.$

5.2. Efficiency compute in proposed scheme

The performance analysis of ID-based chameleon hash function and chameleon signature based on RSA is executed as per following table.

In the next table we compare the computational complexity of our scheme with the existing GQ signature scheme [3],[7],[8].

Phase	Exponent	Multiplication Modulo	Hash Function
Setup	$1T_e$	$1T_m$	$2T_h$
Extract	$2T_e$	$--$	$1T_h$
Hash generation	$2T_e$	$1T_m$	$1T_h$
Forge	$1T_e$	$1T_m$	$--$
Signature generation	$1T_e$	$--$	$1T_h$
Verification	$2T_e$	$1T_m$	$1T_h$

Table 1: Computational time cost in proposed scheme.

Phase	Mihir Bellare et al.[3]	Cheng-Kang Chu et al.[7]	Cheng Fan lu et al.[8]	Our Scheme
Key Generation	$T_e + T_o + T_h$	$3T_e + 2T_m + 3T_o$	$T_e + T_m + 2T_o + 2T_h$	$2T_e + T_o + 2T_h$
Signature Generation	$2T_e + 4T_o + 2T_h$	$8T_e + 2T_m + 6T_o + T_h$	$2T_e + T_m + T_h$	$T_e + T_o + T_h$
Verification	$2T_e + T_o + 2T_h$	$2T_e + T_o + T_h$	$2T_e + T_m + T_h$	$2T_e + T_m + T_h$

Table 2: Comparison of computational time with previous schemes.

The notations used in Tables 1 and 2 are as follows:
T_e: computation time for an exponentiation operation;
T_m: computation time for a multiplication operation;
T_o: computation time for a modular operation;
T_h: computation time for a hash operation.

The computation time of different phases of the schemes is given in Table 2. It is important to note that the computation time for a valid chameleon signature falls into three parts. The first part consists of the time taken for the key generation, second is signature generation and last is verification process, which are a one-time computation and remain fixed for the entire period.

6. Conclusion

In this paper, we have proposed a new ID-based chameleon hash scheme and chameleon signature based RSA without the key escrow problems. Moreover, the proposed scheme is that the property of message hiding or message recovery can be achieved by the signer. And proposed scheme is implemented in Mathematica version 7.

Acknowledgements

The first author [1] is thankful to UGC, New Delhi, India for providing Rajiv Gandhi National Fellowship (F1-17.1/2010-13/RGNF-2012-13-ST-CHH-35416) as financial assistance for this research work.

References

[1] G. Ateniese, B. de Medeiros, Identity-based chameleon hash and applications, *Financial Cryptography 2004*, LNCS 3110, Springer-Verlag (2004), 164-180.

[2] G. Ateniese, B.de Medeiros, On the key-exposure problem in chameleon hashes, *Security in Communication Networks 2004*, LNCS 3352, Springer-Verlag (2005), 165-179.

[3] M. Bellare, G. Neven, Identity-based multi-signatures from RSA, *Topics in CryptologyCT-RSA*, LNCS 4377, Springer-Heidelberg (2006), 145-162.

[4] D. Chaum and H. van Antwerpen, Undeniable signatures, *Advances in Cryptology-Crypto*, LNCS 435, Springer-Verlag (1989), 212-216.

[5] X. Chen, F. Zhang, and K. Kim, Chameleon hashing without key exposure, *Information Security*, LNCS 3225, Springer-Verlag (2004), 87-98.

[6] X. Chen, F. Zhang, H. Tian, and K. Kim, Identity-based chameleon hash scheme without key exposure, *Information Security and Privacy*, LNCS 6168, Springer-Verlag (2010), 200-215.

[7] C.K. Chu, L.S. Liu, W.G.Tzeng, A Threshold GQ Signature Scheme, *Applied Cryptography and Network Security*, LNCS 2846 (2003), 137-150.

[8] F.L. Cheng and S. Shiuhpyng, Efficient key-evolving protocol for the GQ signature, *J. of Information Science and Engineering*, **20** (2004), 763-769.

[9] L.C. Guillou, J.J. Quisquater, A paradoxical identity-based signature scheme resulting from zero-knowledge, *Proceedings of Crypto88*, LNCS 403, Springer-Verlag (1988), 216-231.

[10] H. Krawczyk and T. Rabin, Chameleon hashing and signatures, *Proc. of NDSS 2000*, A Preliminary version can be found at *Cryptology ePrint Archive: Report 1998/01* (2000), 143-154.

[11] A. Shamir, Identity-based cryptosystems and signature schemes, *Advances in Cryptology-Crypto*, LNCS 196, Springer-Verlag (1984), 47-53.

[12] F. Zhang, R. Safavi-Naini, W. Susilo, ID-Based chameleon hashes from bilinear pairings, *Cryptology ePrint Archive, Report 2003/208*, available at http://www.iacr.org/2003/208.

[13] Y. Zhan, X. Chen, H. Tian, Y. Wang, Identity-based key-exposure free chameleon hashing based on the RSA Assumption, *J. of Computational Information Systems*, **7**, No 2 (2011), 350-358.