Deniable Ring Authentication Revisited

Willy Susilo and Yi Mu

Centre for Information Security Research
School of Information Technology and Computer Science
University of Wollongong
Wollongong 2522, AUSTRALIA
{wsusilo,ymu}@uow.edu.au

Abstract. Ring signatures allow a signer in an ad-hoc group to authenticate a message on behalf of the group without revealing which member actually produced the signature [8]. Recently, this notion has been extended by Naor by introducing *Deniable Ring Authentication*: it is possible to convince a verifier that a member of an ad-hoc subset of participants is authenticating a message without revealing which member has issued the signature, and the verifier V cannot convince any third party that message m was indeed authenticated. Unfortunately, the scheme proposed in [7] requires an *interactive* protocol, which requires an assumption that an anonymous routing channel (eg. MIX-net) exists. Having this restriction, the primitive cannot be used in practice without the existence of the anonymous routing channel. In this paper, we introduce a non-interactive version of deniable ring authentication. This work proposes a deniable ring authentication without any interactive protocol required (cf. [7]). We present a generic construction that can convert *any* existing ring signature schemes to deniable ring authentication schemes. Our generic construction combines *any* ring signature scheme with an ID-based chameleon hash function. We also present three ID-based chameleon hash functions and show that our schemes outperform the construction proposed in [2].

1 Introduction

A ring signature scheme [8] can be used to convince a verifier that a document is legally signed by one of the n possible independent signers without revealing the identity of the signer. This signature scheme can be seen as a simple group signature scheme that has no group manager who can revoke the identity of the signer in the case of forgery. To produce a ring signature, the signer constructs an ad-hoc collection of signers that includes himself, and computes the signature entirely by himself using only secret key and the others' public keys. This primitive is formalized by Rivest, Shamir and Tauman in [8], and the construction presented in [8] is based on RSA.

In [1], Abe, Ohkubo and Suzuki presented a scheme to use public-keys of several different signature schemes (that are based on discrete logarithm problem

M. Jakobsson, M. Yung, J. Zhou (Eds.): ACNS 2004, LNCS 3089, pp. 149–163, 2004.

and/or factorization) to generate a ring signature scheme (that they call 1-out-of-n signature scheme). Unlike the previous construction, their contribution allows a mixture of DL-type keys and RSA-type keys in the ring signature construction.

Recently, Naor extended this work to introduce a new primitive called *Deniable Ring Authentication* [7]. Deniable Ring Authentication allows a signer, who forms an ad-hoc collection of participants, to convince a single verifier, V, that a member of an ad-hoc group is authenticating a message m, without revealing which one. Moreover, the verifier V cannot convince any third party that message m was indeed authenticated. This is done by showing that the verifier V could have produced such signature by himself, without any interaction with the signers.

The primitive introduced in [7] is particularly useful in the case where the signer would like to *designate* his authenticated message to a particular verifier. The construction provided in [7] is based on the assumption that users have public-keys of some good encryption schemes. However, the drawbacks of the presented scheme are as follows. Firstly, the scheme requires an interactive zero knowledge protocol. It is assumed that an anonymous channel routing (eg. MIX-net) exists and can be used. Secondly, the message size is longer compared to a normal ring signature. This is due to the interactivity required in the protocol.

In this paper, we provide a generic construction for Deniable Ring Authentication that does not require any interaction. We provide a generic construction for Deniable Ring Authentication that is non-interactive. By removing the interactivity of the protocol, the primitive can be used more widely in practice (cf. [7]).

1.1 Related Work

In [8], the definition of *ring signatures* was formalized and an efficient scheme based on RSA was proposed. A ring signature scheme is based on trapdoor one-way permutations and an ideal block cipher that is regarded as a perfectly random permutation. A ring signature scheme allows a signer who knows at least one secret information (or trapdoor information) to produce a sequence of n random permutations and form them into a ring. This signature can be used to convince any third party that one of the participants in the group (who knows the trapdoor information) has authenticated the message on behalf of the group. The authentication provides *signer ambiguity*, in the sense that no one can identify who has actually signed the message.

In [1], a method to construct a ring signature from different types of public keys, such as these for integer factoring based schemes and discrete log based schemes, was proposed. The proposed scheme is more efficient than [8]. The formal security definition of a ring signature is also defined in [1].

Dwork, Naor and Sahai proposed *deniable authentication* in [5]. Deniable authentication provides a system that addresses the deniability aspects, i.e. the protocol does not leave any paper trail for the authentication of the message. This work allows a single signer to achieve this property.

In [7], the notion of ring signatures was combined with deniable authenticaton [5]. The result is called *Deniable Ring Authentication* that allows a signer to authenticate a message m on behalf of an ad hoc collection of users and to convince a verifier that this authentication is done correctly. Moreover, the verifier cannot convince any third party that the message m was indeed authenticated. There is no 'paper trail' of the conversation, other than what could be produced by the verifier alone, as in zero-knowledge [7]. However, the verification is done interactively, and hence, the requirement of having an anonymous routing, such as MIX-nets, is essential. Moreover, as a result of the requirement of this new notion, the message size is longer compared to a normal ring signature.

In [11], we constructed a non-interactive version of deniable ring authentication scheme. The scheme uses a combination of a ring signature scheme and a chameleon hash function. However, we assume that the verifier has setup a chameleon hash function before a message can be sent to him/her, and this is certainly not practical.

Our Contributions

Essentially, we provide a generic construction for non-interactive deniable authentication schemes. Our schemes follow all the requirements defined in [7], but there is no interactivity involved. The recipient of the deniable ring authentication can verify the correctness of an authenticated message without any interaction with the ad-hoc signers. This will certainly improve the usage of deniable ring authentication in practice. The size of the our signature scheme is the same as the original ring signature scheme together with a random number. This is significantly shorter compared to the previous construction in [7]. Our scheme is an ID-based scheme, which means that the only requirement for the verifier (or signature recipient) is to have his ID (such as email address, a person's address, etc) published. We assume that there is a trusted authority TA, that is only required when the verifier wants to generate his secret key based on his ID. We note that this assumption always exists in ID-based cryptography, as pointed out in its seminal paper in [10]. As pointed out in [7], the verifier V does not necessary have to setup his public-private key before a signer (on behalf of an ad-hoc group) decides to send him a message. Based on our generic construction, we can convert *any* ring signature schemes to deniable ring authentication schemes. We note that as in any other ID based system, our scheme is very applicable in a closed network [10] where a TA trusted by all participants exists.

The rest of this paper is organized as follows. In the next section, we will review some cryptographic tools that are required in this paper. In section 3, we present three constructions of ID-based Chameleon Hashing that are based on the difficulty of factorization problem. We evaluate the efficiency of our schemes and show that they are more efficient than the scheme proposed in [2]. In section 4, we present our generic construction for deniable ring authentication schemes that do not require any interaction with the signers to verify the authenticity of the message. We also present an example of such construction in the same section. Section 5 concludes the paper.

2 Cryptographic Tools

2.1 Chameleon Hashing and ID-Based Chameleon Hashing

Chameleon hashing (or *trapdoor commitment*) is basically non-interactive commitment schemes as proposed by Brassard, Chaum and Crepeau [3]. The idea of chameleon hash functions was introduced and formalized in [6] in the construction of their chameleon signature schemes. The name "chameleon" refers to the ability of the owner of the trapdoor information to change the input to the function to any value of his choice without changing the resulting output.

A chameleon hash function is associated with a pair of public and private keys and has the following properties [6]: (1) Anyone who knows the public key can compute the associated hash function. (2) For people who do not have the knowledge of the trapdoor (i.e. the secret key), the hash function is collision resistant: it is infeasible to find two inputs which are mapped to the same output. (3) The trapdoor information's holder can easily find collisions for every given input.

Several constructions of chameleon hashing have been proposed in [6], which are based on discrete log and [4], which is based on the hardness of deciding whether an element is a "small" e-th residue modulo N^2.

The idea of chameleon hashing has been extended in [2] to construct an Identity-based chameleon hash. An ID-based chameleon hash scheme is defined by a family of efficiently computable algorithms $(\mathsf{Setup}, \mathsf{Extract}, \mathsf{Hash}, \mathsf{Forge})$ as follows.

- Setup: A probabilistic algorithm that is run by a trusted authority TA to generate a pair of keys $\mathcal{SK}$ and $\mathcal{PK}$ defining the scheme. TA publishes $\mathcal{PK}$ and keeps $\mathcal{SK}$ secret.
- $\mathsf{Extract}$: A deterministic algorithm that accepts $\mathcal{SK}$ and an identity string ID and outputs the trapdoor information $\mathcal{T}$ associated with the identity ID.
- Hash: A probabilistic algorithm that accepts $\mathcal{PK}$, an identity string ID and a message m to produce a hash value h.
- Forge: An algorithm that, on input $\mathcal{PK}$, an identity string ID, the trapdoor information $\mathcal{T}$ associated with ID, a message m', and a hash value $h = \mathsf{Hash}(\mathcal{PK}, \mathsf{ID}, m')$, outputs a sequence of random bits that correspond to a valid computation of $\mathsf{Hash}(\mathcal{PK}, \mathsf{ID}, m')$ yielding a collision on the same target value h.

Related to this definition is the notion of *collision forgery* defined [2] as follows.

Definition 1. *A collision forgery strategy is a probabilistic algorithm that given identity string* ID*, a message* m *and random bits* r*, outputs another message* m' *and random bits* r'*, where* $m \neq m'$ *and* $r \neq r'$*, such that* $\mathsf{Hash}(\mathsf{ID}, m, r) = \mathsf{Hash}(\mathsf{ID}, m', r')$ *with non-negligible probability.*

A hashing scheme is said to be *secure against existential collision forgery by passive attacks* if no collision-forgery strategy against it exists.

The semantic security for chameleon hashing scheme is defined as follows [2].

Definition 2. *The chameleon hashing scheme is said to be semantically secure if for all identity strings* ID *and all pairs of messages* (m, m')*, the probability distributions of the random variables* $\mathsf{Hash}(\mathsf{ID}, m, r)$ *and* $\mathsf{Hash}(\mathsf{ID}, m', r')$ *are computationally indistinguishable.*

In [2], an ID-based chameleon hash function based on factorization is proposed. It is also shown an application of ID-based chameleon hash function for a sealed-bid auction system.

2.2 Ring Signature Schemes

For convenience of presentation, we review ring signature schemes in this section. We use the notation proposed in [1] to define ring signature schemes. We note that the ring signature schemes are referred to 1-out-of-n in [1].

Definition 3. *[1] A ring signature scheme consists of three polynomial time algorithms*

- $(s_k, p_k) \leftarrow \mathcal{G}(1^\kappa)$*: A probabilistic algorithm that takes security parameter* κ *and outputs private key* s_k *and public key* p_k*.*
- $\sigma \leftarrow \mathcal{S}(m, s_k, L)$*: A probabilistic algorithm that takes a message* m*, a list* L *that contains public keys including the one that corresponds to* s_k *and outputs a signature* σ*.*
- $\{\texttt{True} \text{ or } \perp\} \leftarrow \mathcal{V}(m, \sigma, L)$*: A deterministic algorithm that takes a message* m *and a signature* σ*, and outputs either* `True` *or* $\perp$ *meaning* accept *or* reject*, respectively. It is required to have* $\texttt{True} \leftarrow \mathcal{V}(m, \mathcal{S}(m, s_k, L), L)$ *with an overwhelming probability.*

A ring signature scheme that allows a mixture of factorization and discrete log based public keys has been constructed in [1].

2.3 Deniable Ring Authentication

The notion of *deniable ring authentication* is formalized in [7]. The setup and requirements of a deniable ring authentication scheme is summarized as follows.
Setup. We assume that the participants have published their public keys. The public keys are generated via a standard public key generation algorithm. We define the *ring* as follows.

A ring $\mathbb{S}$ contains any subset of participants. An authenticator $\mathsf{S}_i \in \mathbb{S}$ can sign on behalf of $\mathbb{S}$. The verifier of a message, V, is an arbitrary party. We require that $\mathsf{V} \not\subset \mathbb{S}$. We assume that both verifier and the authenticator have access to the public keys of all members $\mathsf{S}_i \subset \mathbb{S}$. The verifier V can verify an authenticated message. In Naor's construction in [7], the verification must be done interactively with the help of the ad-hoc group $\mathbb{S}$. However, as we will show in this paper, we can remove this requirement by allowing the verifier V to test the authenticity of the signature by himself.

In the following definition, we denote $< s_{k_i}, p_{k_i} >$ as a pair of secret and public key according to a specific algorithm, that is owned by S_i. A deniable authentication scheme consists of the following algorithms:

- `DeniableSign`(m, s_k, L, V): is a probabilistic polynomial time algorithm that takes a message $m \in \{0,1\}^*$ and a list L that contains a set of public keys, including the one that corresponds to the secret key, s_k, and outputs a signature σ, that can only be verified by V.
- `DeniableVerify`(m, σ, L): is a deterministic non-interactive polynomial-time algorithm that takes a message m, a signature σ and a list of public keys L, and outputs either `True` or $\perp$ meaning `accept` or `reject`, respectively. We require that

$$\mathtt{Pr}\begin{pmatrix} \{m, \sigma, L\} : \sigma \leftarrow \mathtt{DeniableSign}(m, s_k, L, \mathsf{V}); \\ \mathtt{True} \leftarrow \mathtt{DeniableVerify}(m, \sigma, L) \end{pmatrix} = 1.$$

L includes public keys based on different security parameters, and the security of `DeniableSign` (m, s_k, L, V) is set to the smallest one among them. L can include several types of public-keys at the same time, such as for RSA and Schnorr in a particular construction.

We note that the verifier V cannot convince any other third party about the authenticity of the message because he can always forge the signature by creating the required proof in the verification by himself [7].

As presented in [7], the verification requires V to interact with the ad-hoc group of participants to test the authenticity of the message. This restriction requires an existence of an anonymous routing channel [7]. The purpose of this work is to remove this requirement and to allow V to verify the authenticity of the signature without any communication with $\mathbb{S}$.

Intuitively, our idea is to *combine* any ring signature scheme with an ID-based chameleon hash function to obtain a deniable ring authentication scheme. In the following section, we will present three novel constructions of ID-based chameleon hash functions, that are based on the hardness of factorization problem, and we will proceed with our generic construction for deniable authentication schemes in section 4.

3 Three Constructions of ID-Based Chameleon Hash Schemes Based on Factorization

In this section, we will present three ID-based chameleon hash functions. We will also show that our schemes are more efficient than the one proposed in [2]. The settings for the three ID-based chameleon hash functions are as follows.

Model

We assume there is a trusted authority TA which exists to assist the receiver to "extract" his secret key whenever needed. As noted in [10], the existence of TA can be completely removed after this process. Let ID denote an identity string associated to some party. We note that this ID can be an email address, a person's address, etc. that can uniquely determine the party [10]. Let $\mathcal{H}_{\mathsf{ID}}$ be a secure public one way hash function (for instance, the hash function as defined and used in the ID-based signature scheme in [10]) or a public secure hash-and-encode scheme (eg. EMSA-PSS encoding defined in [9]).

3.1 Scheme 1: An ID-Based Chameleon Hash Based on Factorization

Setup: Following the above setting, the TA generates two safe prime numbers p and q (where $p = 2p' + 1, q = 2q' + 1$, and p', q' are also prime) and computes $n = pq$. Then, he selects a random element $\alpha \in \mathbb{Z}_n^*$, where $ord_n(\alpha) = p'q'$. The public key $\mathcal{PK}$ is (n, α). TA's secret key $\mathcal{SK}$ is (p, q).

$\mathsf{Extract}$: To extract his secret key, a party obtains his identity ID and applies the public hash function $\mathcal{H}_{\mathsf{ID}}$ to obtain $\mathsf{Q}_{\mathsf{ID}} = \mathcal{H}_{\mathsf{ID}}(\mathsf{ID})$. The secret key is extracted as $\mathcal{T} = \alpha^{\mathsf{Q}_{\mathsf{ID}}{}^{-1}} \pmod n$. Note that this value can only be computed by TA who knows the factorization of n, because $\mathsf{Q}_{\mathsf{ID}}{}^{-1}$ is computed modulo $\phi(n)$.

Hash: The $\mathsf{Hash}(\cdot)$ algorithm is defined as

$$\mathcal{H}(\mathsf{ID}, m, r) = \alpha^{h(m)} r^{\mathsf{Q}_{\mathsf{ID}}} \pmod n$$

where $h(\cdot)$ is a secure hash function and $\mathsf{Q}_{\mathsf{ID}} = \mathcal{H}_{\mathsf{ID}}(\mathsf{ID})$.

Forge: The Forge algorithm is defined as follows.

$$\mathsf{Forge}(\mathsf{ID}, \mathsf{Q}_{\mathsf{ID}}, m, r, h, m') = r' = \mathcal{T}^{h(m)-h(m')} r \pmod n.$$

Completeness. The completeness of the Forge algorithm is justified as follows.

$$\begin{aligned}
\mathsf{Hash}(\mathsf{ID}, m', r') &= \alpha^{h(m')} (r')^{\mathsf{Q}_{\mathsf{ID}}} \pmod n \\
&= \alpha^{h(m')} \left\{ \mathcal{T}^{h(m)-h(m')} r \right\}^{\mathsf{Q}_{\mathsf{ID}}} \pmod n \\
&= \alpha^{h(m')} \left\{ \alpha^{\mathsf{Q}_{\mathsf{ID}}{}^{-1}(h(m)-h(m'))} r \right\}^{\mathsf{Q}_{\mathsf{ID}}} \pmod n \\
&= \alpha^{h(m')} \left\{ \alpha^{(h(m)-h(m')} r^{\mathsf{Q}_{\mathsf{ID}}} \right\} \pmod n \\
&= \alpha^{h(m)} r^{\mathsf{Q}_{\mathsf{ID}}} \pmod n \\
&= \mathsf{Hash}(\mathsf{ID}, m, r).
\end{aligned}$$

We note that the owner of the secret key can always produce a collision in the hash function with an overwhelming probability. ◇

Security Analysis
As noted in [2], we need to show the following security requirement.

Theorem 1. *Our first ID-based chameleon hash function is resistant to forgery, assuming that RSA signature scheme is resistant.*

Proof. We will prove our argument with a contradiction. Firstly, we assume there is an algorithm $\mathcal{F}$ that can produce a collision for our first ID-based chameleon hash function without the knowledge of the trapdoor information $\mathcal{T}$, and we will build an algorithm $\mathcal{A}$ that uses $\mathcal{F}$ to generate an RSA signature without the trapdoor information. The algorithm $\mathcal{F}$ can produce a collision such that

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

for a given $c = \mathsf{Hash}(\mathsf{ID}, m, r)$, a pair of messages (m, m') and a random number r. We build the algorithm $\mathcal{A}$ as follows.

- Run algorithm $\mathcal{F}$ given (c, m, m', r) to produce $r' \neq r$.
- From this collision, $\alpha^{h(m)}(r)^{\mathsf{Q_{ID}}} = \alpha^{h(m')}(r')^{\mathsf{Q_{ID}}} \pmod n$ holds. That means, $(r/r')^{\mathsf{Q_{ID}}} = \alpha^{h(m')-h(m)} \pmod n$.
- From the above knowledge, we can compute $(r/r') = \left(\alpha^{(h(m')-h(m))\mathsf{Q_{ID}}^{-1}}\right) \pmod n$, which was assumed to be infeasible without the knowledge of the factorization of n.

We note that by running our algorithm $\mathcal{A}$, we have successfully "extract" an RSA signature on $\alpha^{h(m')-h(m)}$ (with a "public key" $\mathsf{Q_{ID}}$ associated with n) without the knowledge of the factorization of n. This result contradicts with the assumption that it is infeasible to compute an RSA signature on a message without the knowledge of the factorization of n (the difficulty of finding the e-th root modulo n). ◇

3.2 Scheme 2: An ID-Based Chameleon Hash Based on RSA

In this section, we design an ID-based chameleon hash function based on RSA. Essentially, this construction simplifies the construction proposed in [2]. We note that our construction is inspired by Shamir's ID based signature scheme proposed in [10]. The Setup and Extract algorithms follow the same setting as the construction in [10].

Setup: The TA generates two safe prime numbers p and q (where $p = 2p' + 1, q = 2q' + 1$, and p', q' are also prime). Then, he generates an RSA-key pair (e, d), where $d = e^{-1} \pmod{4p'q'}$, together with computing $n = pq$. The published values, $\mathcal{PK}$, are (e, n), and d is kept secret by TA (as TA's $\mathcal{SK}$). We note that in several occasions, we also would like to keep p and q as part of the secret information (eg. to make the computation faster with Chinese Remainder Theorem).

Extract: To extract his secret key, a party obtains his identity ID and applies the public hash function $\mathcal{H}_{\mathsf{ID}}$ to obtain $\mathsf{Q_{ID}} = \mathcal{H}_{\mathsf{ID}}(\mathsf{ID})$. The secret key is extracted as $\mathcal{T} = \mathsf{Q_{ID}}^d \pmod n$. Note that this process can only be performed by TA

who knows the secret key d, under the published public key (e, n). The values p and q are discarded afterwards.

Hash: The $\mathsf{Hash}(\cdot)$ algorithm is defined as follows.

$$\mathsf{Hash}(\mathsf{ID}, m, r) = {\mathsf{Q}_{\mathsf{ID}}}^{h(m)} r^e \pmod n$$

where $h(\cdot)$ is a secure hash function, and $\mathsf{Q}_{\mathsf{ID}} = \mathcal{H}_{\mathsf{ID}}(\mathsf{ID})$.

Forge: The Forge algorithm is defined as follows.

$$\mathsf{Forge}(\mathsf{ID}, \mathsf{Q}_{\mathsf{ID}}, m, r, h, m') = r' = \mathcal{T}^{h(m)-h(m')} r \pmod n.$$

Completeness. The completeness of the Forge algorithm for Scheme 2 is justified as follows.

$$\begin{aligned}
\mathsf{Hash}(\mathsf{ID}, m', r') &= {\mathsf{Q}_{\mathsf{ID}}}^{h(m')} (r')^e \pmod n \\
&= {\mathsf{Q}_{\mathsf{ID}}}^{h(m')} \left\{ \mathcal{T}^{h(m)-h(m')} r \right\}^e \pmod n \\
&= {\mathsf{Q}_{\mathsf{ID}}}^{h(m')} \left\{ {\mathsf{Q}_{\mathsf{ID}}}^{(h(m)-h(m'))d} r \right\}^e \pmod n \\
&= {\mathsf{Q}_{\mathsf{ID}}}^{h(m')} \left\{ {\mathsf{Q}_{\mathsf{ID}}}^{(h(m)-h(m'))} r^e \right\} \pmod n \\
&= {\mathsf{Q}_{\mathsf{ID}}}^{h(m)} r^e \pmod n \\
&= \mathsf{Hash}(\mathsf{ID}, m, r).
\end{aligned}$$

◇

Security Analysis

Theorem 2. *Our ID-based chameleon hash function based on RSA is resistant to forgery, assuming that RSA signature scheme is resistant.*

Proof. We assume there is an algorithm $\mathcal{F}$ that can produce a collision for our ID-based chameleon hash function, without the knowledge of the trapdoor information $\mathcal{T}$. We will construct an algorithm $\mathcal{A}$ that will use the algorithm $\mathcal{F}$ to generate an RSA signature as follows.

We assume that there exists an algorithm $\mathcal{F}$ can produce a collision

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

for a given $c = \mathsf{Hash}(\mathsf{ID}, m, r)$, a pair of messages (m, m') and a random number r. We construct our algorithm $\mathcal{A}$ as follows.

- Run algorithm $\mathcal{F}$ given (c, m, m', r), to produce $r' \neq r$, so that the collision occurs.
- From this collision, we will obtain $\mathsf{Hash}(\mathsf{ID}, m, r) = \mathsf{Hash}(\mathsf{ID}, m', r')$, or ${\mathsf{Q}_{\mathsf{ID}}}^{h(m)} r^e = {\mathsf{Q}_{\mathsf{ID}}}^{h(m')} (r')^e \pmod n$.

- From the above equation, we obtain

$$(r/r')^e = \mathsf{Q_{ID}}^{h(m')-h(m)} \pmod{n}.$$

- The above equation will be equivalent to

$$(r/r') = \left\{\mathsf{Q_{ID}}^{h(m')-h(m)}\right\}^d \pmod{n}.$$

- Note that $\left\{\mathsf{Q_{ID}}^{h(m')-h(m)}\right\}^d$ is an RSA signature on $\mathsf{Q_{ID}}^{h(m')-h(m)}$, which is assumed to be infeasible to compute without the knowledge of the trapdoor d.
- Hence, we have successfully "extract" an RSA signature on $\mathsf{Q_{ID}}^{h(m')-h(m)}$ without the knowledge of d.

We note that the success probability of the algorithm $\mathcal{A}$ is the same as the algorithm $\mathcal{F}$. Assuming that RSA is secure, then our ID-based scheme is also secure. ◇

3.3 Scheme 3: An ID-Based Chameleon Hash Based on Factorization

In this section, we design an ID-based chameleon hash function based on factorization. Unlike the previous two constructions, the TA does not require to keep any information other than the factorization of n as his secret keys, $\mathcal{SK}$.

Setup: The TA generates two safe prime numbers p and q, and compute $n = pq$. The public key $\mathcal{PK}$ is n, and the secret key $\mathcal{SK}$ is (p, q).

Extract: To extract his secret key, a party obtains his identity ID and applies the public hash function $\mathcal{H}_{\mathsf{ID}}$ to obtain $\mathsf{Q_{ID}} = \mathcal{H}_{\mathsf{ID}}(\mathsf{ID})$. The secret key is extracted as $\mathcal{T} = \mathsf{Q_{ID}}^{\mathsf{Q_{ID}}^{-1}} \pmod{n}$. Note that the computation $\mathsf{Q_{ID}}^{-1}$ is performed under modulo $\phi(n)$ which is infeasible to be performed without the knowledge of the factorization of n.

Hash: The $\mathsf{Hash}(\cdot)$ algorithm is defined as follows.

$$\mathcal{H}(\mathsf{ID}, m, r) = \mathsf{Q_{ID}}^{h(m)} r^{\mathsf{Q_{ID}}} \pmod{n}$$

where $h(\cdot)$ is a secure hash function, and $\mathsf{Q_{ID}} = \mathcal{H}_{\mathsf{ID}}(\mathsf{ID})$.

Forge: The Forge algorithm is defined as follows.

$$\mathsf{Forge}(\mathsf{ID}, \mathsf{Q_{ID}}, m, r, h, m') = r' = \mathcal{T}^{h(m)-h(m')} r \pmod{n}.$$

Completeness. The completeness of the Forge algorithm for Scheme 3 is justified as follows.

$$\mathsf{Hash}(\mathsf{ID}, m', r') = \mathsf{Q_{ID}}^{h(m')} (r')^{\mathsf{Q_{ID}}} \pmod{n}$$

$$\begin{aligned}
&= {\mathsf{Q}_{\mathsf{ID}}}^{h(m')} \left\{\mathcal{T}^{h(m)-h(m')} r\right\}^{\mathsf{Q}_{\mathsf{ID}}} \pmod n \\
&= {\mathsf{Q}_{\mathsf{ID}}}^{h(m')} \left\{{\mathsf{Q}_{\mathsf{ID}}}^{{\mathsf{Q}_{\mathsf{ID}}}^{-1}(h(m)-h(m'))} r\right\}^{\mathsf{Q}_{\mathsf{ID}}} \pmod n \\
&= {\mathsf{Q}_{\mathsf{ID}}}^{h(m')} \left\{{\mathsf{Q}_{\mathsf{ID}}}^{h(m)-h(m')} r^{\mathsf{Q}_{\mathsf{ID}}}\right\} \pmod n \\
&= {\mathsf{Q}_{\mathsf{ID}}}^{h(m)} r^{\mathsf{Q}_{\mathsf{ID}}} \pmod n \\
&= \mathsf{Hash}(\mathsf{ID}, m, r).
\end{aligned}$$

Theorem 3. *Our third scheme is resistant to forgery, assuming that RSA signature scheme is resistant.*

Proof. The proof is very similar to Theorem 1 and Theorem 2. Therefore, we omitted the proof. ◇

3.4 Efficiency Comparison

In this section we compare efficiency of our proposed schemes with the scheme proposed in [2]. Efficiency of ID-based chameleon hash functions can be measured in terms of the parameters lengths: the length of TA's public key, the length of TA's secret key and the length of recipient's secret key (after $\mathsf{Extract}$). To compare two ID-based chameleon hash functions, we fix the level of security provided by the two schemes and find the size of the three length parameters. Table 1 gives the results of comparison of four ID-based chameleon hash functions. We fix the size of the prime numbers p and q, and without losing generality, assume that their size are equal. Let $\tau = |p|_2 \approx |q|_2$. Therefore, we have $|n|_2 \approx 2\tau$. We assume that the length of the elements to construct the secret/public key parameters are represented by κ. The first scheme refers to the scheme proposed in [2]. We refer this scheme as AM scheme (that stands for "Ateniese-Medeiros" scheme). The next three columns refer to the three schemes presented earlier.

In the scheme proposed in [2], TA's public key $\mathcal{PK}$ is (n, v), where $n = pq$, and v is a random integer. The secret key $\mathcal{SK}$ is (p, q, w), where $vw+z(p-1)(q-1) = 1$. The Hash function is defined as $\mathsf{Hash}(\mathsf{ID}, m, r) = {\mathsf{Q}_{\mathsf{ID}}}^{h(m)} r^v \pmod n$. The recipient's secret key is extracted from ${\mathsf{Q}_{\mathsf{ID}}}^w \pmod n$.
As shown in Table 1, our schemes outperform the scheme proposed in [2]. In particular, scheme 2 requires the shortest $\mathcal{SK}$ length for the TA and scheme 3 requires TA's $\mathcal{PK}$ = TA's $\mathcal{SK}$ = Recipient's $\mathcal{SK} = 2\tau$.

4 Generic Construction for Deniable Ring Authentication Schemes

In this section, we describe our generic construction for deniable ring authentication schemes. Our construction is based on the ID-based chameleon hash functions $\mathcal{H}_{\mathsf{ID}}(\cdot)$ described in the previous section. Let V_{ID} be the recipient of the deniable ring authentication, who has his identity ID published. The construction is defined as follows.

Table 1. Comparison of Efficiency Parameters

	AM scheme [2]	*Scheme 1*	*Scheme 2*	*Scheme 3*
TA's $\mathcal{PK}$ length	$2\tau + \kappa$	$2\tau + \kappa$	$2\tau + \kappa$	2τ
TA's $\mathcal{SK}$ length	$2\tau + \kappa$	2τ	κ	2τ
Recipient's $\mathcal{SK}$ length	2τ	2τ	2τ	2τ
$\mathsf{Hash}(\mathsf{ID}, \mathrm{m}, \mathrm{r})$ $(\mathrm{mod}\ n)$	${\mathsf{Q_{ID}}}^{h(m)} r^v$	$\alpha^{h(m)} r^{\mathsf{Q_{ID}}}$	${\mathsf{Q_{ID}}}^{h(m)} r^e$	${\mathsf{Q_{ID}}}^{h(m)} r^{\mathsf{Q_{ID}}}$
$\mathsf{Extract}(\cdot) \quad (\mathrm{mod}\ n)$	${\mathsf{Q_{ID}}}^w$	$\alpha^{\mathsf{Q_{ID}}^{-1}}$	${\mathsf{Q_{ID}}}^d$	${\mathsf{Q_{ID}}}^{\mathsf{Q_{ID}}^{-1}}$
Underlying hard problem	Factorization	RSA	RSA	Factorization

1. Define:

$$\texttt{DeniableSign}(m, s_k, L, \mathsf{V_{ID}}) \stackrel{\Delta}{=} \begin{cases} \tilde{h} \leftarrow \mathcal{H}_{\mathsf{V_{ID}}}(\mathsf{ID}, m, r), \text{for a random } r; \\ \sigma_1 \leftarrow \mathcal{S}(\tilde{h}, s_k, L); \\ \sigma \leftarrow (\sigma_1 || r). \end{cases}$$

   The signed message is $\sigma \stackrel{\Delta}{=} (m, \sigma)$.
2. Define:

$$\texttt{DeniableVerify}(m, \sigma, L) \stackrel{\Delta}{=} \begin{cases} (\sigma_1 || r) \leftarrow \sigma; \\ \tilde{h} \leftarrow \mathcal{H}_{\mathsf{V_{ID}}}(\mathsf{ID}, m, r); \\ \texttt{Result} \leftarrow \mathcal{V}(\tilde{h}, \sigma_1, L). \end{cases}$$

   The result of the verification is defined as

$$\texttt{Result} \leftarrow \texttt{DeniableVerify}(m, \sigma, L)$$

   which is either `True` or $\perp$, meaning `accept` or `reject`, respectively.

Theorem 4. *The resulting signature is non-transferable.*

Proof. We note that the resulting deniable ring authentication does not allow the verifier V to convince any third party about this fact. This is due to the use of ID-based chameleon hash function $\mathcal{H}(\cdot)$. The verifier V can always contact the TA to extract his secret key and execute the Forge algorithm to create a valid pair of (m', r'), for $m' \neq m$, that will pass under the ring signature verification algorithm. ◇

Theorem 5. *A signer* S *can always create an ad-hoc group* $\mathbb{S}$ *and generate a deniable ring authentication without contacting the verifier* V.

Proof. Due to the use of ID-based chameleon hash function, the verifier does not need to have her public key setup before receiving a message that is signed with a deniable ring authentication scheme. The signer is only required to contact TA if he wants to 'forge' a signature. ◇

We note that an interesting property of the above deniable ring authentication scheme is to allow a signer to form an ad-hoc group and sign on behalf of the group without contacting the verifier. The verifier is only obliged to contact TA if he wants to 'forge' a signature. However, since there is no way to know whether the verifier has contacted TA or not, then the resulting signature cannot be used to convince any other third party (*non-transferability* property).

4.1 Comparison with Other Schemes

In this section, we provide a complete comparison between our scheme and the other deniable authentication schemes, namely RST scheme proposed in [8] (achieved by adding the verifier to the ring) and Naor's scheme proposed in [7]. The result of this comparison is illustrated in Table 2. In the comparison below, we assume that the length of any ring signature scheme is denoted by $|l|_2$. The length of the random number r required in our scheme is denoted by $|r|_2$.

Table 2. Comparison of Deniable Authentication Schemes

	RST Scheme [8]	**Naor's Scheme [7]**	**Our Scheme**
Additional Assumption	The verifier V is required to have his public key setup	An anonymous routing channel exists (for interactive protocol)	n/a
Implication of the Assumption	The verifier V can be added to the ring $\mathbb{S}$	An Interactive Protocol	n/a
Requirements	$\mathsf{V} \subset \mathbb{S}$	$\mathsf{V} \not\subset \mathbb{S}$	$\mathsf{V} \not\subset \mathbb{S}$
Protocol	Non interactive	Interactive	Non interactive
Signature Length	$|l|_2$	at least $2|l|_2$	$|l|_2 + |r|_2$
Size of the ring $\mathbb{S}$	2 (can be extended to n)	n	n

From the comparison table above, we can conclude that our scheme is *the only* scheme that satisfies all the requirements of deniable authentication schemes [7] but without any interactive protocol required. In the scheme proposed in [8], although a non interactive protocol is used, it is assumed that $\mathsf{V} \subset \mathbb{S}$, which violates the original assumption proposed in [7]. Our scheme also produces a shorter signature compared to [7].

4.2 An Example

We present a sample conversion of the ring signature scheme proposed in [1] to construct a deniable ring authentication scheme as described in previous section. We will use a ring signature scheme based on RSA proposed in [1], together with our ID-based chameleon hash function based on RSA presented in section 3.2. The ID-based chameleon hash function is defined as

$$\mathsf{Hash}(\mathsf{ID}, m, r) = {\mathsf{Q}_{\mathsf{ID}}}^{h(m)} r^e \pmod{n}.$$

For $i = 0, \cdots, N-1$, let (e_i, n_i) be RSA public keys and $H_i : \{0,1\}^* \to \mathbb{Z}_{n_i}$ be hash functions. Let L be a list of these public-keys. TA has published his public key (e, n) as described in section 3.2. We assume that the verifier V has his ID, ID, published. For simplicity, we also assume a signer S_k would like to send a deniable ring authenticated message to V. Let the size of the ring be N.

A signer S_k who owns the private key d_k generates a signature for a message m as follows.

- Obtain the identity of the recipient, ID, and compute $\mathsf{Q}_{\mathsf{ID}} = \mathcal{H}_{\mathsf{ID}}(\mathsf{ID})$.
- Select a random number $r \in \mathbb{Z}_n$ and compute
$$\tilde{h} = {\mathsf{Q}_{\mathsf{ID}}}^{h(m)} r^e \pmod{n}.$$
- Select N random numbers $r_1 \in \mathbb{Z}_{n_1}, \cdots, r_n \in \mathbb{Z}_{n_N}$.
- From r_k, $k \in \{1, \cdots, N\}$, compute $c_{k+1} = H_{k+1}(L, \tilde{h}, r_k)$.
- For $i = k+1, \cdots, N-1, 0, 1, \cdots, k-1$, select $s_i \in \mathbb{Z}_{n_i}$ and compute $c_{i+1} = H_{i+1}(L, \tilde{h}, c_i + s_i^{e_i} \pmod{n_i}))$.
- Compute $s_k = (r_k - c_k)^{d_k} \pmod{n_k}$.

The resulting signature is $(r, c_0, s_1, s_1, \cdots, s_{N-1})$.
To verify a signature, the verifier V performs the following.

- Generate $\mathsf{Q}_{\mathsf{ID}} = \mathcal{H}_{\mathsf{ID}}(\mathsf{ID})$ for his ID.
- Compute $\tilde{h} = {\mathsf{Q}_{\mathsf{ID}}}^{h(m)} r^e \pmod{n}$.
- For $i = 0, \cdots, N-1$, compute
  - $r_i = c_i + s_i^{e_i} \pmod{n_i}$;
  - $c_{i+1} = H_{i+1}(L, \tilde{h}, r_i)$ if $i \neq N-1$.
- Accept if $c_0 \overset{?}{=} H_0(L, \tilde{h}, r_{n-1})$ holds. Otherwise, reject.

Theorem 6. *The above signature scheme is a non-interactive deniable ring authentication scheme.*

Proof (sketch). The proof can be derived from the use of ID-based chameleon hash function described in section 3.2. The verifier V can contact TA to retrieve his secret key $\mathcal{T}$. Obtaining his secret key, he can select any message $m' \neq m$ and execute the Forge algorithm to retrieve the associated $r' \neq r$ that will pass the verification test. The underlying ring signature used remains the same, and hence, we have obtained a deniable ring authentication scheme. ◇

5 Conclusions

In this paper, we presented a novel construction of deniable ring authentication scheme that does not require any interaction to verify the authenticity of the message. Our scheme combines *any* ring signature schemes with an ID-based chameleon hash function that allows the resulting signature to be *non-transferable*. In our construction, the verifier V (or the signature recipient) does not necessarily need to retrieve the associated secret key that is related to his published identification, ID, unless he wants to 'forge' a signature. Based on this idea, the resulting signature becomes non-transferable, since any third party cannot determine whether the verifier has retrieved his secret key and produce a collision on the hash function or not. We presented a generic construction of deniable ring authentication schemes. Unlike the construction proposed in [7], our scheme produces a shorter signature size (cf. [7]). We presented three ID-based chameleon hash functions that outperform the construction proposed in [2].

References

1. M. Abe, M. Ohkubo, and K. Suzuki. 1-out-of-n Signatures from a Variety of Keys. *Advances in Cryptology - Asiacrypt 2002, Lecture Notes in Computer Science 2501*, pages 415 – 432, 2002.
2. G. Ateniese and B. de Medeiros. Identity-based Chameleon Hash and Applications. *Financial Cryptography 2004*, 2004 (to appear).
3. G. Brassard, D. Chaum, and C. Crépeau. Minimum Disclosure Proofs of Knowledge. *JCSS, 37(2)*, pages 156–189, 1988.
4. D. Catalano, R. Gennaro, N. Howgrave-Graham, and P. Q. Nguyen. Paillier's Cryptosystem Revisited. *ACM CCS 2001*, 2001.
5. C. Dwork, M. Naor, and A. Sahai. Concurrent Zero-Knowledge. *Proc. 30th ACM Symposium on the Theory of Computing*, pages 409–418, 1998.
6. H. Krawczyk and T. Rabin. Chameleon hashing and signatures. *Network and Distributed System Security Symposium, The Internet Society*, pages 143–154, 2000.
7. M. Naor. Deniable Ring Authentication. *Advances in Cryptology - Crypto 2002, Lecture Notes in Computer Science 2442*, pages 481–498, 2002.
8. R. L. Rivest, A. Shamir, and Y. Tauman. How to Leak a Secret. *Advances in Cryptology - Asiacrypt 2001, Lecture Notes in Computer Science 2248*, pages 552–565, 2001.
9. RSA Labs. RSA Cryptography Standard: EMSAPPS - PKCS # 1 v2.1. June 2002.
10. A. Shamir. Identity-based cryptosystems and signature schemes. *Advances in Cryptology - Crypto '84, Lecture Notes in Computer Science 196*, pages 47–53, 1985.
11. W. Susilo and Y. Mu. Non-Interactive Deniable Ring Authentication. *The 6th International Conference on Information Security and Cryptology (ICISC 2003)*, pages 397–412, 2003.