# A non-interactive deniable authentication scheme based on designated verifier proofs

Bin Wang

Information Engineering College of Yangzhou University

No.36 Middle JiangYang Road, Yangzhou City, Jiangsu Province, P.R.China

**Tel:** 086-0514-82220820

**E-mail:**xiaobinw@yahoo.com

***Abstract:*** A deniable authentication protocol enables a receiver to identify the source of the given messages but unable to prove to a third party the identity of the sender. In recent years, several non-interactive deniable authentication schemes have been proposed in order to enhance efficiency. In this paper, we propose a security model for non-interactive deniable authentication schemes. Then a non-interactive deniable authentication scheme is presented based on designated verifier proofs. Furthermore, we prove the security of our scheme under the DDH assumption.

***Keywords:*** Deniable authentication, Authentication, Designated verifier proofs, DDH assumption, Provable security;

## 1. Introduction

In an open network environment, authentication protocols enable a receiver to make sure that a particular sender wants to authenticate a message to him. A deniable authentication protocol is an authentication protocol which enables the receiver to identify the source of the given messages but unable to prove to a third party the identity of the sender.

The notion of deniable authentication was developed by Aumann and Rabin [1]. Later, Deng et al.[4] proposed two improved schemes based on Aumann and Rabin's work. In 2002, Fan et al. [6] proposed a simple deniable authentication protocol based on the Diffie-Hellman key distribution protocol. Since then deniable authentication schemes have enjoyed a considerable amount of interest from the cryptographic research community. However, security of these schemes is argued by presenting attacks that fail, which provides very weak security guarantees. Several proposed schemes were broken or improved [17,18].

On the other hand, the notion of deniable authentication was formalized by Dwork et al. [5] using simulation paradigm. In other words, an authentication protocol is deniable if the view of any verifier can be simulated without knowing the secret key of the prover. Nevertheless, Dwork et al. introduced timing constraints on the network in order to prove security in the concurrent setting and the proof of knowledge is rather time-consuming. Raimondo et al. [13,14] combined simulation paradigm with the security model for authenticated key exchange protocols [2] to present a formal definition of deniability. However, as their model aims at deniable key exchange protocols (requiring to use special protocol compilers), it is not appropriate to directly adopt their method for designing deniable authentication protocols; otherwise the resulting security reduction would not be tight. Furthermore, all the above-mentioned interactive schemes require several communication rounds between the prover and the verifier. Consequently, several non-interactive deniable authentication schemes have been proposed in order to reduce the communication cost [10,11,15].

To the author's knowledge, only Susilo et al. [16] provided a formal definition of a non-interactive deniable ring authentication scheme. A generic method was proposed in [16] for converting a ring signature scheme to a deniable ring authentication scheme by means of the concept of chameleon hash function. Hence the security of their scheme relies on the underlying ring signature scheme and chameleon hash function, which are rather costly primitives to be used for achieving deniability (requiring a lot of pairing operations as well as exponentiation operations).

Our goal in this paper is to define the notion of non-interactive deniable authentication by combining simulation paradigm with the model for authentication schemes defined in [3, 7]. In 1996, Jacobsson et al. introduced the concept of designated-verifier proofs [9]. Such proofs enable a prover Alice to convince a designated verifier Bob that a statement is true. However, Bob cannot transfer the proofs to a third party. The reason is that Bob himself can simulate such proofs. Then we propose a non-interactive deniable authentication scheme on the basis of designated-verifier proofs. Subsequently, we prove that our scheme is secure under the DDH assumption and deniable in the concurrent setting. Finally we compare the efficiency with other related non-interactive deniable authentication schemes in terms of the

computational cost and underlying security assumptions.

## 2. Preliminaries

### 2.1 DDH Problem

**DDH problem:** Let $G$ be a multiplicative cyclic group of prime order $q$ generated by $g$. Given $< g^x, g^y, g^z >$, $x, y, z \in Z_q$, decide whether $z = (xy) \bmod q$.

A distinguishing algorithm $D$ is said to $(t, \varepsilon)$-solve the DDH problem in group $G$ if $D$ runs in time at most $t$ and

$$| \Pr[x, y, z \leftarrow Z_q : D(g^x, g^y, g^z) = 1] - \Pr[x, y \leftarrow Z_q : D(g^x, g^y, g^{xy}) = 1] | \geq \varepsilon .$$

We say that $G$ is a $(t, \varepsilon)$-DDH group if there is no polynomial time algorithm $D$ $(t, \varepsilon)$-solves the DDH problem in group $G$.

### 2.2 Notation

If $A$ is a randomized algorithm, then $y \leftarrow A(x_1, x_2, \cdots)$ means that $A$ has input $x_1, x_2, \cdots$, and the output of $A$ is assigned to $y$.

## 3. Deniable authentication schemes

### 3.1 Syntax of deniable authentication schemes

A non-interactive deniable authentication scheme is a tuple $(\mathrm{Kg,P,V,Sim})$. On input a security parameter $k \in \mathrm{N}$, the randomized polynomial time algorithm **Kg** generates a public key $pk$ and a matching secret key $sk$. The public/secret key pairs for the prover and the verifier are represented by $(pk_P, sk_P)$, $(pk_V, sk_V)$ respectively. **P** (resp. **V**) is a polynomial time algorithm called the prover (resp. verifier) algorithm.

In an initialization step, the prover picks a message $M$. Then the interaction between the prover and the verifier can be described as follows:

$\begin{bmatrix}(M, Authen) \leftarrow P(M, sk_P, pk_V, \rho)\\ d \leftarrow V(pk_P, sk_V, M, Authen)\end{bmatrix}$, where $\rho$ denotes random coins chosen by the prover algorithm. $Authen$ is the authenticator for the given message $M$ produced by the prover algorithm. The decision bit $d \in \{0,1\}$ is output by the verifier. $d = 1$ means that the verifier accepts. A conversation transcript is defined to be $C = M \,||\, Authen$.

**Correctness:** For all $k \in \mathrm{N}$, $(pk, sk) \leftarrow \mathrm{Kg}(1^k)$, we require perfect consistency, meaning that $\Pr\left[d = 1 : \begin{bmatrix}(M, Authen) \leftarrow P(M, sk_P, pk_V, \rho)\\ d \leftarrow V(pk_P, sk_V, M, Authen)\end{bmatrix}\right] = 1$.

On input the prover's public key $pk_P$, the receiver's secret key $sk_V$, and a message $M$, the simulation algorithm Sim should generate a simulated transcript $\overline{C}$ which is computationally indistinguishable from a real transcript $C$ for the given message $M$.

### 3.2 Security model for deniable authentication schemes

#### 3.2.1 Unforgeability of deniable authentication schemes

Let NDI = (Kg,P,V,Sim) be a non-interactive deniable authentication scheme. Consider the following game $\mathrm{Exp}^{\mathrm{imp}}_{\mathrm{NDI,A}}(k)$ between an adversary $A$ and a game challenger S:

**Stage1:** The challenger S runs the algorithm $\mathrm{Kg}(1^k)$ to obtain key pairs $(pk_P, sk_P)$, $(pk_V, sk_V)$. An empty set Res is also created, which is used to store the conversation transcripts. Then the adversary $A$ is provided with the public keys $pk_P, pk_V$.

**Stage2:** The adversary $A$ makes the following queries:

(1) Conv queries: $A$ adaptively picks a message $M$. On input the message $M$, the challenger S sets the input of the prover algorithm to $St_P = (M, sk_P, pk_V, \rho)$, where $\rho$ denotes fresh random coins chosen by S. Then the challenger S executes $\begin{bmatrix}(M, Authen) \leftarrow P(St_P)\\ d \leftarrow V(pk_P, sk_V, M, Authen)\end{bmatrix}$. Note that the input of the verifier algorithm is set to

$(pk_P, sk_V, M, Authen)$ by S. In the end, S provides $A$ with $C = M \| Authen$ (the conversation transcript) and sets $\text{Res} \leftarrow \text{Res} \cup \{C\}$.

**Output:** Eventually, $A$ outputs $St_A$, which represents knowledge gained by $A$ during stage 2. Then $A$ picks a message $M^*$. The adversary $A$ wins if :

(1) $\begin{bmatrix} (M^*, Authen^*) \leftarrow A(M^*, St_A) \\ d^* \leftarrow V(pk_P, sk_V, M^*, Authen^*) \end{bmatrix}$, $C^* = M^* \| Authen^*$ ; and

(2) $d^* = 1$, $C^* \notin \text{Res}$.

The advantage of $A$ in this game is $\text{Adv}^{\text{imp}}_{\text{NDI,A}}(k) = \Pr[\text{Exp}^{\text{imp}}_{\text{NDI,A}}(k) = 1]$. We say that NDI is secure against impersonation attack if $\text{Adv}^{\text{imp}}_{\text{NDI,A}}(k)$ is negligible for every polynomial time adversary $A$.

### 3.2.2 Deniability

Consider the following game $\text{Exp}^{\text{Den}}_{\text{NDI,D}}(k)$ between a distinguisher $D$ and a game challenger S :

**Stage1:** The challenger S runs the algorithm **Kg** to obtain key pairs $(pk_P, sk_P)$, $(pk_V, sk_V)$. Two empty sets Res and $\overline{\text{Res}}$ are created. Then the distinguisher $D$ is provided with the public keys $pk_P, pk_V$.

**Stage2:** The distinguisher $D$ makes the following queries:

(1) Conv queries: On input a message $M$ chosen by $D$, the challenger S sets the input of the prover algorithm to $St_P = (M, sk_P, pk_V, \rho)$, where $\rho$ denotes fresh random coins chosen by the challenger S . Then the challenger S executes $\begin{bmatrix} (M, Authen) \leftarrow P(St_P) \\ d \leftarrow V(pk_P, sk_V, M, Authen) \end{bmatrix}$. Note that input of the verifier algorithm is set to $(pk_P, sk_V, M, Authen)$ by S . In the end, S provides the distinguisher $D$ with $C = M \| Authen$ (the conversation transcript) and sets $\text{Res} \leftarrow \text{Res} \cup \{C\}$.

(2) $\overline{\text{Conv}}$ queries: On input a message $M$ chosen by $D$, the challenger S sets the input of the simulation algorithm $Sim$ to $St=(M, sk_V, pk_P, \rho)$, where $\rho$ denotes fresh random coins chosen by S. Then the challenger S runs the simulation algorithm by executing $M \| \overline{Authen} \leftarrow Sim(St)$. In the end, S provides $D$ with $\overline{C} = M \| \overline{Authen}$, and sets $\overline{\text{Res}} \leftarrow \overline{\text{Res}} \bigcup \{\overline{C}\}$.

**Challenge:** Once $D$ decides that Stage2 is over, $D$ picks a message $M^*$ such that $M^*$ has not been submitted as one of the Conv queries, $\overline{\text{Conv}}$ queries. Then the challenger picks a random bit $b \in \{0,1\}$. If $b=0$, $S$ performs in the same way as it answers Conv queries. If $b=1$, $S$ performs in the same way as it answers $\overline{\text{Conv}}$ queries.

**Guess:** Finally, $D$ outputs a bit $b'$. $D$ wins the game if $b'=b$.

The advantage of $D$ in this game is $\text{Adv}^{\text{Den}}_{\text{NDI,D}}(k) = |\Pr[b'=b] - \frac{1}{2}|$. We say that NDI is deniable if $\text{Adv}^{\text{Den}}_{\text{NDI,D}}(k)$ is negligible for every polynomial time $D$.

**4. Our scheme**

Our scheme is based on the techniques used in [8,9]. Let $G$ be a multiplicative cyclic group of prime order $q$ generated by $g$, $\log_2 q \approx k$, where $k$ is a security parameter. Then we choose a secure collision-resistant hash function $H$, $H: \{0,1\}^* \rightarrow Z_q$. The system parameters are **params=** $< G, q, g, H >$.

**Kg**(Key Generation)**:** Chooses a random $h \in G$ and a random $x \in Z_q$. Computes $y_1 = g^x$ and $y_2 = h^x$. The public key is $pk = (h, y_1, y_2)$ and the secret key is $x$. The public/secret key pairs for the prover and the verifier are represented by $(pk_P, x_P)$, $(pk_V, x_V)$ respectively, where $pk_P = (h, y_{1P}, y_{2P})$, $pk_V = (h, y_{1V}, y_{2V})$.

**P**(**Prover**)**:** The prover picks a message $M$ and performs as follows:

(1) Picks random $w \in Z_q, r \in Z_q$ and computes $c' = g^w (y_{1V})^r$.

(2) Picks a random $k \in Z_q^*$ and computes $A = g^k, B = h^k$.

(3) Computes $c = H(M, c', A, B)$. If $(c + w) = 0 \bmod q$, then goto step (2); otherwise computes $s = x_P (c + w) + k \bmod q$.

(4) Sends $M$ and $Authen = (w, g^r, c, s)$ to the verifier.

**V(Verifier):** Given a tuple $Authen = (w, g^r, c, s, M)$, the verifier performs as follows:

(1) Computes $c' = g^w (g^r)^{x_V}$.

(2) Computes $A = (g^s (y_{1P})^{-(c+w)}), B = (h^s (y_{2P})^{-(c+w)})$. Returns 1 if and only if $c = H(M, c', A, B)$.

**Sim**(Simulation algorithm)**:** On input a message $M$, the prover's public key $pk_P$ and the verifier's secret key $x_V$, the algorithm performs as follows:

(1) Picks random $\alpha \in Z_q, \beta \in Z_q^*, s \in Z_q$ and computes $c' = g^\alpha$.

(2) Computes $A = (g^s (y_{1P})^{-\beta}), B = (h^s (y_{2P})^{-\beta})$.

(3) Computes $c = H(M, c', A, B)$.

(4) Computes $w = \beta - c \bmod q$ and $r = (\alpha - w)/x_V$.

(5) Produces $M$ and $\overline{Authen} = (w, g^r, c, s)$.

**Lemma 1:** Given the public/secret key pairs $(pk_P, x_P)$, $(pk_V, x_V)$, the statistical distance of the following distributions is negligible.

$$\delta = \left\{ (w, g^r, c, s) \middle| \begin{array}{l} w \in_R Z_q, r \in_R Z_q \\ k \in_R Z_q^*, c \in_R Z_q, c + w \neq 0 \bmod q \\ s = x_P (c + w) + k \bmod q \end{array} \right\}$$

$$\delta^{/} = \left\{ (w, g^{r}, c, s) \left| \begin{array}{l} \alpha \in_R Z_q, \beta \in_R Z_q^* \\ s, c \in_R Z_q \\ w = \beta - c \bmod q \\ r = (\alpha - w)/x_V \bmod q \end{array} \right. \right\}$$

**Proof:** We use the notation $x \in_R S$ to mean "the element $x$ is chosen with uniform probability from the set $S$ ". At first, we choose a valid tuple $(\overline{w}, R, \overline{c}, \overline{s})$ , $\overline{w}, \overline{c}, \overline{s} \in Z_q, R \in G$ , such that for some message $M$

(1) $\overline{c^{/}} = g^{\overline{w}}(R)^{x_V}, \overline{A} = (g^{\overline{s}}(y_{1P})^{-(\overline{c}+\overline{w})}), \overline{B} = (h^{\overline{s}}(y_{2P})^{-(\overline{c}+\overline{w})})$.

(2) $\overline{c} = H(M, \overline{c^{/}}, \overline{A}, \overline{B})$.

We then compute the probability of appearance of this tuple following each distribution of probabilities.

$$\Pr_{\delta}[(w, g^{r}, c, s) = (\overline{w}, R, \overline{c}, \overline{s})] = \Pr_{\delta}[\left\{ \begin{array}{l} g^{r} = R, r \in_R Z_q \\ w = \overline{w}, c = \overline{c}, c + w \neq 0 \bmod q \\ w, c \in_R Z_q \\ s = \overline{s}, s \in_R Z_q \end{array} \right\}] = 1/(q^3(q-1))$$

$$\Pr_{\delta^{/}}[(w, g^{r}, c, s) = (\overline{w}, R, \overline{c}, \overline{s})] = \Pr_{\delta^{/}}[\left\{ \begin{array}{l} s = \overline{s}, s \in_R Z_q \\ c = \overline{c}, w = \overline{w} \\ w, c \in_R Z_q \\ g^{r} = R, r \in_R Z_q \end{array} \right\}] = 1/q^4$$

The statistical distance of the distribution $\delta^{/}$ (simulated transcripts) from the correct distribution $\delta$ (real transcripts) is at most $1/q$ . Hence, the statistical distance is negligible by the choice of security parameter $k$ .

**5. Security Analysis**

**Theorem 1:** Let $G$ be a multiplicative cyclic group of prime order $q$ generated by $g$ and assume $G$ is a $(t^{/}, \varepsilon^{/})$-DDH group such that the exponentiation in $G$ takes time $t_1$. Assume there is a polynomial-time adversary $A$ can break our non-interactive deniable

authentication scheme with success probability $\varepsilon$ in time at most $t$. Suppose $A$ makes at most $q_h$ random oracle queries and $q_c$ Conv queries. Then there is an algorithm $B$ that solves the DDH problem in $G_1$ with

$$\varepsilon' \geq (\varepsilon - (q_c + q_h + 2)/q)$$

$$t' \approx t + O(q_c \cdot t_1)$$

**Proof:** Algorithm $B$ is given as input a tuple $(g, g^x, g^y, g^z)$. The system parameters are **params**$=< G, q, g, H >$, where $H$ is a random oracle controlled by $B$.

$B$ sets the public key of the prover to $pk_P = (h, y_{1P}, y_{2P})$, where $h = g^y$, $y_{1P} = g^x, y_{2P} = g^z$. Next, $B$ picks a random $x' \in Z_q$ and sets the public key of the verifier to $pk_V = (h, y_{1V} = g^{x'}, y_{2V} = h^{x'})$. Then $B$ works by interacting with the adversary $A$.

**Stage1:** An empty set Res is created by $B$. Then the adversary $A$ is provided with $pk_P, pk_V$.

**Stage2:** $B$ should answer $A$'s queries as follows:

$H$ queries: $A$ picks $k \in Z_q^*, c' \in G$, and computes $A' = g^k, B' = h^k$. In response to a $H$-query $(M, c', A', B')$ issued by $A$, $B$ first checks if the output of $H$ on this input has been previously defined. If so, $B$ returns the previously assigned value. Otherwise, $B$ responds with a value chosen uniformly at random from $Z_q$ and stores $((M, c', A', B'), H(M, c', A', B'))$.

Conv queries: $A$ picks a message $M$ and issues a Conv query. In response to a Conv query, $B$ performs as follows:

(1) Picks random $\alpha \in Z_q, \beta \in Z_q^*$, $s \in Z_q$ and computes $c' = g^\alpha$.

(2) Computes $A' = (g^s (y_{1P})^{-\beta}), B' = (h^s (y_{2P})^{-\beta})$.

(3) Computes $c = H(M, c', A', B')$ by making a $H$-query with $(M, c', A', B')$.

(4) Computes $w = \beta - c \bmod q$ and $r = (\alpha - w)/x'$.

(5) Sets $Authen = (w, g^r, c, s)$. Then $B$ provides $A$ with $C = M \,||\, Authen$ and sets $\text{Res} \leftarrow \text{Res} \bigcup \{C\}$.

Eventually, $A$ picks a message $M^*$ and outputs $C^* = M^* \,||\, Authen^*$. If the verifier algorithm returns 1 on input $C^*$ and $C^* \notin \text{Res}$, then $B$ outputs 1; otherwise $B$ outputs 0.

**Lemma 2:** Assume $g, h, y_{1P}, y_{2P} \in G$, but there is no $x \in Z_q$ with $g^x = y_{1P}$, $h^x = y_{2P}$. Given $w \in Z_q, r \in Z_q$ and a message $M$, a cheating prover with the public key $pk_P = (h, y_{1P}, y_{2P})$ can pick at most one value of $c$ over $Z_q$ for which the honest verifier will accept $M \,||\, (w, g^r, c, s)$.

**Proof:** Given $w, r$, assume that the cheating prover can pick two $c_1, c_2, c_1 \neq c_2 \bmod q$, such that the honest verifier will accept $(w, g^r, c_i, s_i, M)$, $i \in \{1, 2\}$. Then we have

$c' = g^w (g^r)^{x_V}$, $A_i = (g^{s_i} (y_{1P})^{-(c_i + w)}), B_i = (h^{s_i} (y_{2P})^{-(c_i + w)})$,

$c_i = H(M, c', A_i, B_i), i \in \{1, 2\}$

The probability that at least one of $(M, c', A_i, B_i)$, $i \in \{1, 2\}$ not having been queried by $A$ is at most $2/q$ ( i.e., $A$ can guess $B$'s uniform random answer). As $2/q$ is negligible, we always assume that $(M, c', A_i, B_i)$, $i \in \{1, 2\}$ have been queried by $A$. Hence we know that $A_i = g^{k_i}, B_i = h^{k_i}$, $i \in \{1, 2\}$, where $k_1$ and $k_2$ are chosen by $A$.

Since $c_1 \neq c_2 \bmod q$, we have

$g^{((s_1 - s_2) - (k_1 - k_2))(c_1 - c_2)^{-1} \bmod q} = y_{1P}$, $h^{((s_1 - s_2) - (k_1 - k_2))(c_1 - c_2)^{-1} \bmod q} = y_{2P}$, which is contrary to the assumption of the lemma.

We first analyze the probability that $B$ outputs 1 when $(g, g^x, g^y, g^z)$ is a

Diffie-Hellman tuple. According to Lemma 1, the statistical distance of the distribution $\delta'$ (simulated transcripts) from the correct distribution is at most $1/q$. Taking as a whole, $q_c$ simulated transcripts are jointly distributed with statistical distance at most $q_c/q$ from the correct distribution. Then $B$ outputs 1 with probability at least $\varepsilon-(q_c/q)$ in this case.

Secondly, if $(g, g^x, g^y, g^z)$ is a random tuple, then it is not a Diffie-Hellman tuple with probability $1-1/q$. In this case, for any $w, r$, it follows from Lemma 2 that there is at most one possible value of $c$ for which there exists some $s$ satisfying

$$c' = g^w (g^r)^{x_V}, \quad A = (g^s (y_{1P})^{-(c+w)}), B = (h^s (y_{2P})^{-(c+w)})$$

$$c = H(M, c', A, B)$$

Thus $A$ outputs a forgery (and hence $B$ outputs 1 ) with probability $\varepsilon/q + ((q_h+1)/q)(1-1/q) \le 1/q + ((q_h+1)/q)$ in this case (the additive factor of 1 occurs in case $A$ did not make the relevant $H$-query for its forgery). Putting everything together, we have

$$|\Pr[x, y \leftarrow Z_q : D(g^x, g^y, g^{xy}) = 1] - \Pr[x, y, z \leftarrow Z_q : D(g^x, g^y, g^z) = 1]|$$

$$\ge (\varepsilon - (q_c + q_h + 2)/q)$$

**Theorem 2:** Let $G$ be a multiplicative cyclic group of prime order $q$ generated by $g$. Suppose a polynomial-time adversary $A$ makes at most $q_h$ random oracle queries, $q_c$ Conv queries and $q_{\overline{c}}$ $\overline{\text{Conv}}$ queries. Our non-interactive deniable authentication scheme is deniable against $A$.

**Proof:** The system parameters are **params**$=<G, q, g, H>$, where $H$ is a random oracle controlled by $B$.

$B$ first picks a random $h \in G$ and random $x_P, x_V \in Z_q$. The public key of the prover is $pk_P = (h, y_{1P}, y_{2P})$, where $y_{1P} = g^{x_P}, y_{2P} = h^{x_P}$. Similarly, the public key of the verifier

is $pk_V = (h, y_{1V}, y_{2V})$, where $y_{1V} = g^{x_V}, y_{2V} = h^{x_V}$. Obviously, $B$ generates key pairs with the same distribution as in the real interaction. Then $B$ works by interacting with the adversary $A$.

**Stage1:** Two empty sets Res and $\overline{\text{Res}}$ are created. Then the adversary $A$ is provided with $pk_P, pk_V$.

**Stage2:** Since $B$ knows the secret keys of the prover and the verifier, Conv queries and $\overline{\text{Conv}}$ queries issued by $A$ can be answered correctly.

**Challenge:** At the end of Stage2, $A$ picks a message $M^*$ such that $M^*$ has not been submitted as one of the Conv queries, $\overline{\text{Conv}}$ queries. Then $B$ picks a random bit $b \in \{0,1\}$. If $b = 0$, $S$ performs in the same way as it answers Conv queries. If $b = 1$, $S$ performs in the same way as it answers $\overline{\text{Conv}}$ queries.

It follows from Lemma 1 that the advantage of $A$ is negligible since the statistical distance of the above-mentioned distributions is negligible.

**Theorem 3:** Concurrent composition of our scheme preserves deniability.

**Proof:** Assume there is a verifier interacting with $t$ independent copies of the prover algorithm simultaneously, where $t$ is polynomial in the security parameter $k$ defined in section 4. Let $T$ be the concurrent composition of $t$ individual transcripts. Then we have

$$T = ((M_1 \,||\, Authen_1), \cdots, (M_i \,||\, Authen_i), \cdots, (M_t \,||\, Authen_t))$$

We can construct a simulator to produce a simulated concurrent transcript $\overline{T}$ as follows by means of the simulation algorithm defined in section 4:

$$\overline{T} = (M_1 \,||\, \overline{Authen_1}, \cdots, M_i \,||\, \overline{Authen_i}, \cdots, M_t \,||\, \overline{Authen_t})$$

According to Lemma 1, it is easy to see that the statistical distance of the distribution with respect to simulated concurrent transcripts from the correct distribution $\leq t/q \approx poly(k)/2^k$. For any negligible function $\mu(n)$ and any polynomial $p(n)$, $\mu(n)p(n)$ is negligible. Hence the resulting statistical distance is negligible.

## 6. Performance Analysis

In this section, we evaluate the performance of the proposed scheme and other related non-interactive deniable authentication schemes proposed in [10,15,16] in terms of the computational cost and underlying security assumptions. The result is stated in Table 1. Hash denotes a hash operation. Exp, Pair and Cham_Hash denote an exponentiation operation, a pairing operation and a Chameleon hash operation respectively, which are the most time-consuming operation. Note that Susilo et al's scheme [16] used the chameleon hash function $\text{Cham\_Hash}_V(m,r) = g^m y^r \bmod p$.

In the table, the computational cost of a multi-exponentiation(that is, computing $g^a h^b$) is assumed to be equivalent to 1.5 exponentiations[12]. Although the schemes proposed in [10,15] are more efficient, no formal security analysis is presented for their schemes. In addition, the security of Susilo et al's scheme relies on the underlying ring signature scheme and chameleon hash function, which are rather costly primitives (requiring a lot of pairing operations as well as exponentiation operations). In addition, our scheme is proven to be deniable in the concurrent setting.

**Table 1. Performance comparison with other related schemes**

| Scheme | Prover's computational cost | Verifier's computational cost | Assumptions | Deniable in the concurrent setting |
|---|---|---|---|---|
| Lee et al's Scheme [10] | 2Exp+2Hash | 2.5Exp+2Hash | Unproven | Unproven |
| Shao's Scheme [15] | 1Exp+2Hash | 2.5Exp+2Hash | Unproven | Unproven |
| Susilo et al's Scheme [16] | (2n－1)Exp +Cham_Hash | (n+1)Pair+ Cham_Hash | Ring signature and Chameleon hash function | Unproven |
| Our Scheme | 3.5Exp+Hash | 4.5Exp+Hash | DDH | Proven |

## 7.Conclusion

In this paper, we present a security model for non-interactive deniable authentication schemes. Then we construct a non-interactive deniable authentication scheme based on the concept of designated-verifier proofs. Finally, we show that our scheme satisfies the deniable property and is unforgeable against a polynomial time adversary. The security of our scheme is proved under the DDH assumption. Moreover, we show that our scheme is deniable in the concurrent setting.

## References


[1] Y. Aumann, M.O. Rabin, "Authentication enhanced security and error correcting codes", in Proceedings of CRYPTO 1998. Springer, 1998, LNCS 1462, 299–303.

[2] M. Bellare, R. Canetti and H. Krawczyk, "A Modular Approach to the Design and Analysis of Authentication and Key Exchange Protocols", proc. of 30th Symposium on Theory of Computing (STOC), ACM, pp. 419–428, 1998.

[3] M. Bellare , C. Namprempre , G. Neven, "Security Proofs for Identity-Based Identification and Signature Schemes", in EUROCRYPT 2004, LNCS 3027,2004, 268-286.

[4] X. Deng, C. Lee, H. Lee, H. Zhu, "Deniable Authentication Protocols", IEE Proc.Comput. Digit. Tech., 148(2)(2001), 101–104.

[5] C. Dwork, M. Naor, A. Sahai, "Concurrent zero-knowledge", in: Proceedings of 30th ACM STOC'98, 1998, 409–418.

[6] L. Fan, C.X. Xu, J.H. Li, "Deniable authentication protocol based on Diffie–Hellman algorithm", Electronics Letters, 38 (4) (2002), 705–706.

[7] U. Feige, A. Fiat, and A. Shamir, "Zero knowledge proofs of identity", Journal of Cryptology, 1(2)(1988), 77–94.

[8] E. J. Goh, S. Jarecki, J. Katz, N. Wan, "Efficient Signature Schemes with Tight Reductions to the Diffie-Hellman Problems", Journal of Cryptology, 20(4)(2007), 493–514.

[9] M. Jacobsson, K. Sako, R, Impagliazzo, "Designated verifier proofs and their applications", in EUROCRYPT 1996, LNCS 1070, 1996, 143-154.

[10] W.B. Lee, C.C. Wu, W.J. Tsaur, "A novel deniable authentication protocol using generalized ElGamal signature scheme", Information Sciences, 177 (2007), 1376–1381.

[11] R. Lu, Z.F. Cao, "A new deniable authentication protocol from bilinear pairings", Applied Mathematics and Computation, 168 (2005), 954–961.

[12] A.J.Menezes, P.C.van Oorschot and S.A. Vanstone, "Handbook of Applied Cryptography", CRC Press, 1997

[13] M.D. Raimondo and R. Gennaro, "New Approaches for Deniable Authentication", http://eprint.iacr.org/2005/046.pdf.

[14] M.D. Raimondo, R. Gennaro and H. Krawczyk, "Deniable Authentication and Key Exchange", proc. of the 13th ACM conference on Computer and communications security, ACM, 2006, pp. 400–409.

[15] Z.Shao, "Efficient deniable authentication protocol based on generalized ElGamal signature scheme", Computer Standards & Interfaces, 26 (2004), 449–454.

[16] W. Susilo and Y. Mu, "Non-interactive Deniable Ring Authentication", in ICISC 2003, LNCS 2971, pp. 386-401.

[17] E.J. Yoon, E. K. Ryu, K.Y. Yoo, "Improvement of Fan et al.'s deniable authentication protocol based on Diffie–Hellman algorithm", Applied Mathematics and Computation, 167 (2005), 274–280.

[18] R.W. Zhu, D.S. Wong, and C.H. Lee, "Cryptanalysis of a suite of deniable authentication protocols", IEEE Communications Letters, 10(6) (2006), 504-506.

**Corresponding author:** Bin Wang

**Address:** P.O.Box 153#

Information Engineering College
No.36 Middle JiangYang Road
Yangzhou University, Yangzhou City, Jiangsu Province, Peoples Republic of China

**Postal code:** 225009

**E-mail:** xiaobinw@yahoo.com