ECC-Based Non-Interactive Deniable Authentication with Designated Verifier

Yalin Chen[1], *Jue-Sam Chou[2]
[1]Institute of Information systems and applications, National Tsing Hua University
d949702@oz.nthu.edu.tw
[2] Department of Information Management, Nanhua University
*: corresponding author Tel: 886-5-2721001
jschou@mail.nhu.edu.tw

Abstract

Recently, researchers have proposed many non-interactive deniable authentication (NIDA) protocols. Most of them claim that their protocols possess full deniability. However, after reviewing, we found that they either cannot achieve full deniability, or suffer KCI or SKCI attack; moreover, lack efficiency, because they are mainly based on DLP, factoring problem, or bilinear pairings. Due to this observation, and that ECC provides the security equivalence to RSA and DSA by using much smaller key size, we used Fiat-Shamir heuristic to propose a novel ECC-based NIDA protocol for achieving full deniability as well as getting more efficient than the previous schemes. After security analyses and efficiency comparisons, we confirmed the success of the usage. Therefore, the proposed scheme was more suitable to be implemented in low power mobile devices than the others.

Keyword: deniable authentication protocol, Fiat-Shamir heuristic, perfect zero-knowledge, key compromise impersonation attack, voting systems

1. Introduction

People have paid much attention in several security features such as, integrity, confidentiality, non-repudiation, and authentication when communicating over the Internet. Recently, the **deniability** has gotten more attractive since it can protect personal privacy, which is often required in business activities. For example, in an auction, a bidder may not expect his bid's content revealed. Even, he may wish nobody knew his participation and thus can deny his attendance. This is called the content's source **deniability**. Further, if a protocol can let people deny both the content and its source, we call it a **fully deniable authentication (FDA) protocol**. Up to now, there are many FDA schemes [1-21, 28, 32-34] proposed. They include two types: (1) Interactive deniable authentication (IDA) protocol, and (2) Non-interactive deniable authentication (NIDA) protocol. Among them, [1-5, 11, 12, 14, 28, 33] are type (1) protocols and [6-10, 13, 15-21, 32, 34] are type (2) protocols. NIDA protocols have the advantage of communication efficiency over the IDA ones for using only one pass. However, after examining previous NIDA protocols, we found that they either have security vulnerabilities or cannot achieve **full deniability**. We described this in Section 2. Subsequently, in Section 3, we proposed a fully deniable NIDA protocol based on the Fiat-Shamir heuristic [22]. Fiat-Shamir heuristic can produce a non-interactive proof (signature) on the sender's message. However, the proof unforgeability prevents the receiver from simulating. This makes Fiat-Shamir heuristic cannot let the sender denies what he sent, because from [1-3], we see that receiver's "un-simulatability" implies sender's "un-deniability". To overcome the problem, we modify it by replacing the one-way hash function with ECC-based ElGamal encryption (ELG_Enc) to produce the random-looking *challenge*. When simulating, the simulator can recover any pre-chosen random *challenge* by ElGamal decryption. Besides, for attaining better efficiency, our design used of the elliptic curve cryptosystem. After security analyses and performance comparisons, we found that our scheme not only possessed the security properties: SKCI resistance, and deniability, but also was more efficient than the other schemes. Thus, the combination of Fiat-Shamir heuristic with ECC-based Elgamal encryption provides a better choice in designing an NIDA protocol.

The rest of this article is organized as follows. In Sec. 2, we introduce the Fiat-Shamir heuristic on which our scheme bases and then give a literature review on some previous works. In Sec. 3, we propose our protocol. The analyses of its deniability are discussed in Sec 4. And its security analyses and performance comparisons with other related works are described in Sec 5. Finally, a conclusion is given in Sec. 6.

2. Background and related work

In this section, we describe the used notations in Section 2.1, introduce Fiat-Shamir heuristic in Sec. 2.2, and finally review the related works in Sec. 2.3.

2.1 Definitions of used notations:

p, q: two large primes satisfying $q|(p\text{-}1)$,

G_1,: an additive group of order q on an elliptic curve,

G_2,: a multiplicative group of order q,

P : a primitive element of G_1,

g: the generator of G_2,

x_i: user i's private key,

Y_i: user i's public key, which equals to g^{x_i} in DLP scheme or x_iP in ECC,

H: a one-way hash function mapping from $\{0, 1\}^*$ to Z_q,

H_1: a one-way hash function mapping from G_1 to Z_q,

H_2: a one-way hash function mapping from $G_1 \times G_1$ to Z_q,
H_3: a one-way hash function mapping from $\{0, 1\}^*$ to G_1,
e: a pairing function mapping from $G_1 \times G_1$ to G_2,
auth: a message authenticator,
E: an adversary

2.2 Fiat-Shamir heuristic

In 1986, Fiat and Shamir [22] suggest a heuristic means for designing a secure digital signature scheme which enables a user to prove his identity and authenticate his message (but doesn't deal with **deniability**) by using the following two steps:

(1) Choose a secure 3-pass identification scheme, e.g., Schnnor's identification scheme [23] in which the output transcript in each round is denoted as (*commitment*, *challenge*, *response*), where *commitment* and *response* are the first and third flows from the prover to the verifier, and *challenge* the second from verifier to prover.

(2) According to the identification scheme chosen, the signer first generates the commitment, then hashes it with message *m* to produce the challenge, and finally computes the response. That is, when signing on message *m*, the signer produces an acceptable transcript (*commitment*, *challenge*, *response*) as the signature, where *the challenge* equals $\mathcal{H}$(*commitment*, *m*) and $\mathcal{H}$ is a cryptographic hash function.

The Fiat-Shamir heuristic is an efficient way in building non-interactive zero-knowledge proofs. Such constructions are provably secure based on cryptographic hash functions in the random oracle model [24-26].

Suppose that a signer with private/public (*SK /PK)* key pair $x/\ g^{-x}$, where $x \in_R Z_q$, adopts Schnorr Identification Scheme and performs the two steps. He signs on message *m* by computing commitment $t = g^k$, challenge $ch = \mathcal{H}(t, m)$, and response $s = k + SK * ch$, where *k* is a randomly chosen number, to form the signature (t, s), which can be publicly verifiable by checking whether $t = g^s(PK)^{\mathcal{H}(t\|m)}$ holds or not.

2.3 Literature review of deniable authentication protocols

As mentioned earlier, there are two types of deniable authentication protocols: (1) IDA protocol, and (2) NIDA protocol. We review some of them below.

(1). IDA protocols

In 1998, Dwork et al. [1] propose a deniable authentication protocol based on concurrent zero-knowledge proof. Their study permits a sender (*S*) to authenticate a message *m* for a receiver (*R*), but a third party cannot verify the authentication. In other words, it does not permit *R* to convince a third party that *S* has authenticated *m* to him. In the same year, Aumann and Rabin [2] propose another deniable authentication protocol based on factoring. They stipulated that if *R* can simulate all the communications between himself and *S*, then *S* can deny the communications. In 2006, Raimodo et al. [3] define an authentication and key exchange protocol to be deniable if *R*'s view can be simulated by an efficient machine (called the simulator) which doesn't know S's secret key. Here, a simulator can construct the transcripts without relying on deducing *S*'s secret key (from the corresponding public key), and *R*'s view means all the information *R* obtained by participating in the protocol. In addition, they also propose the notions of "**partial deniability**" and "**full deniability**". The former is used in SIGMA protocol [11] which adopts non-repudiable signature for authentication; and the latter used in SKEME protocol [12] which adopts an encryption-based method for the same purpose. In the partial deniability definition, *S* can deny only the content of a message. In a **fully deniable** one, he can further deny the message's source (except for the content).

In 2004, Boyd and Mao [5] point out that the two properties, **deniability** and KCI attack prevention, conflict in Boyd et al.'s shared-secrecy based IKE protocol [4]. Because once the secret key of a party, say *A* (or *B*), has been compromised, the attacker can impersonate *B* (or *A*) to talk with *A* (or *B*). This is exactly what the KCI attack means [27]. Chou *et al.* [28] exemplify such a KCI attack. In 2008, Li et al. [14] apply the deniable property in an electronic voting protocol. However, they assume that the system can simulate the voting for any voter. This is unreasonable. Since, if the system is not equitable, it can impersonate any voter. Consequently, the vote result cannot convince anyone. Conversely, if the system is equitable, it's unnecessary for the system's voting for any voter. In other words, the application in [14] is impractical. In 2013, J. Kar [33] proposed an ID-based IDA protocol. However, we found E can pretend S to send R $\sim Q_s{}'$ $(= E_{\Pi pub}(a_s{}'P||T'))$. R will authenticate E unconsciously. In other words, their protocol cannot work correctly.

(2) NIDA protocols

Because non-interactive protocols have the advantage of communication efficiency, many researchers propose deniable authentication protocols of this kind [6-10, 13, 15-21, 32, 34]. In this type, a message with its proof is sent to a receiver in only one pass. The receiver then uses this proof to verify the authenticity of both the message and its sender. But afterward, the sender can deny to a third party that he sent this message. NIDA protocols are generally applied to off-line applications such as, sending emails or signing documents. However, we found that they either suffer KCI attack, SKCI attack, or losing **full deniability** (Hereafter, we use the term deniability to stand for full deniability). In the following, we introduce the two attacks: KCI and SKCI. Then, roughly describe the main frames of some relevant NIDA schemes.

(a) KCI attack

KCI is a security notion which means that the loss of a user's secret would enable *E* to impersonate any party to communicate to the user [27]. According to this definition, we know that there are two possible ways for *E* to launch such attack on an IDA protocol performed; for example, *E* compromises *S's* (or *R's*) private key and then impersonates *R* (or *S*) to communicate with *S* (or *R*). For a one-pass NIDA protocol, only one KCI attack launching is possible; *E* compromises *R*'s private key, and impersonates *S* to authenticate another message *m'* to *R*.

(b). SKCI attack

In 2004, Shao [6] propose a NIDA scheme using generalized ElGamal signature. In the scheme, Alice randomly chooses $t \in Z_q$, computes $k = Y_B{}^t$ (mod p), $r = H(k)$, $s = t - x_A \cdot r$ (mod q), and $auth = H(k||m)$, and then sends (r, s, $auth$, m) to Bob. After receiving (r, s, $auth$, m), Bob computes $k' = (g^s Y_A{}^r)^{x_B}$ and verifies whether both $r = H(k')$ and $auth = H(k'||m)$ hold. However, in 2006, Lee *et al.* [10] point out Shao's scheme has a vulnerability that once the session key *k* has been compromised, the attacker can use arbitrary message *m'* to form a valid $auth' = H_2(k||m')$ and thus impersonate Alice to send Bob (r, s, $auth'$, m'). Bob would then be fooled, because Bob will extract *k* from (r, s) by computing $k' = (g^s Y_A{}^r)^{x_B}$ and hence verify *auth'* as valid. We denote such an attack as SKCI attack. Below, we define SKCI attack in *Definition 1*.

Definition 1. **SKCI (session key compromise impersonation) attack** *means if the receiver discloses part of the shared secret (between the sender and the receiver) to a third party, the third party can then use the leaked information to impersonate the sender by generating a proof on an arbitrary message to be successfully verified by the receiver.*

After describing the meanings of both SKCI and KCI attacks, in the following, we roughly describe the main frames of relevant NIDA schemes and their vulnerabilities.

(c) Related NIDA protocols

- **Shao's scheme [6]**

Except for its SKCI attack suffering (found by Lee et al.’s [10]), this study also found the scheme lacks the **deniability** property since nobody other than Alice can efficiently compute the signature *s* when given *r*. Although, Bob could compute $k' = (g^{s'}Y_A^{r'})^{x_B}$ by randomly choosing *s'* and *r'*, the equation $r' = H(k')$ can be hardly satisfied for a secure hash function [23]. Since the probability is negligible for the hash value of *k'* to be equal to a pre-defined value *r'*. This demonstrates the undeniability of Shao’s scheme.

· Cao *et al.*’s scheme [9]

In 2005, Cao *et al.* [9] propose a Weil pairing ID-based NIDA protocol. In the protocol, there exists a TA (Trust Agent) with a private/ public key pair $s \in Z_q$ / $P_{pub} = sP$. It computes Alice’s public / private key pair $Q_A = H_3(ID_A)$ / $S_A = sQ_A$ and Bob’s public / private key pair $Q_B = H_3(ID_B)$ / $S_B = sQ_B$. When Alice wants to send a message *m* with its authenticator to Bob, she computes $Y = \hat{e}(tP_{pub}+S_A, tP+Q_B)$, $k = H(Y, ID_A)$, and $auth = H(k||m)$, and then sends Bob (ID_A, *t*, *auth*, *m*), where *t* is a timestamp. After receiving the message flow, Bob can extract $Y = \hat{e}(tP+Q_A, tP_{pub}+S_B)$, because he and Alice had pre-shared a secrecy $e(P+Q_A, P+Q_B)^s$. From [5], we see this scheme suffers the KCI attack. Since if *E* compromised Bob’s private key S_B, he can impersonate Alice to send Bob (ID_A, *t'*, *auth'*, *m'*) for another message *m'*, by computing $Y' = \hat{e}(t'P+Q_A, t'P_{pub}+S_B)$, $k' = H(Y', ID_A)$, and $auth' = H(k'||m')$, where *t'* is a timestamp. As a result, *E* successfully fools Bob to accept his message *m'*. Therefore, scheme [9] suffers SKCI attack.

· Two Lu and Cao’s schemes [7, 8]

In 2005, Lu and Cao [7] propose an NIDA scheme based on Weil pairing. In the scheme, Alice randomly chooses $t \in Z_q$, computes $r = H_1(e(P, P)^t)$, $s = \frac{t}{r + x_s} Yr$, and $auth = H_2(\hat{e}(P, P)^t, m)$, and sends (*r*, *s*, *auth*, *m*) to Bob. After receiving (*r*, *s*, *auth*, *m*), Bob extracts the session key *k* by using the session parameters (*r*, *s*), Bob’s private key, and Alice’s public key, i.e. $k = \hat{e}(s, x_r^{-1}(rP+Y_s)) = \hat{e}(P, P)^t$.

Meanwhile, Lu and Co [8] also proposed an NIDA scheme based on factoring in which when Alice wants to send a message with its authenticator to Bob, she transmits (s, b_1, b_2, c, a_1, a_2, *auth*, *m*), where (s, b_1, b_2) is Alice’s signature on random nonce *r* and (c, a_1, a_2) is the result of *r* encrypted by using Bob’s public key. After receiving Alice’s message, Bob decrypts (c, a_1, a_2) to obtain *r* and verifies Alice’s signature, (s, b_1, b_2), on *r*. If it is valid, Bob believes the message is sent from Alice.

However, in 2006, Lee *et al.* [10] point out that both Lu and Cao's schemes suffer **SKCI** attack. In addition, this study also found the two schemes lack the **deniability** property. Because in study [7], if the receiver transforms s into s', the transformed (r, s', MAC) cannot pass the verification since r is not equal to $e(s', x_r^{-1}(rP+Y_s))$. And in work [8], for any given r, nobody other than Alice can efficiently compute r's signature (s, b_1, b_2) due to the difficulty of factoring [29]. That is, when Bob reveals (s, b_1, b_2, r) to a third party, Alice cannot deny that she sent (s, b_1, b_2, c, a_1, a_2, $auth$, m) to Bob.

· **Lee *et al.*'s scheme [10]**

For patching the vulnerability of **SKCI** in schemes [6, 7, 8], Lee *et al.* propose a deniable authentication scheme [10] by using ElGamal signature. In the scheme, Alice randomly chooses t, computes $r = g^t \pmod p$, $s = H(m)\, x_A + tr \pmod q$, $k = (Y_B)^s \pmod p$, and $auth = H(k||m)$, and sends (m, r, $auth$) to Bob. Bob can extract the session key by computing $k' = (Y_A^{H(m)} r^r)^{x_B}$ ($=k \bmod p$) and then verify whether $auth = H(k'||m)$ holds. However, we found if E compromised Bob's long-term private key x_B, he can successfully impersonate Alice to communicate with Bob by randomly choosing r', computing $k' = (Y_A^{H(m')}(r')^{r'})^{x_B} \pmod p$ and $auth' = H(k'||m')$, and sending Bob (m', r', $auth'$). Bob would accept this forged message (m', r', $auth'$) unconsciously. It, therefore, suffers the KCI attack.

· **Lu et al.'s scheme [13]**

In 2007, Lu *et al.* [13] propose an improvement on [8] by including the identities of both communicating parties. However, the improvement still lacks the **deniability** property, because Alice has to generate the signature (s, b_1, b_2, r) on r. It's well known that nobody other than Alice can efficiently compute the signature.

· **Shi and Li's scheme [19]**

In 2005, Shi and Li propose an identity-based deniable authentication protocol [19]. In the scheme, S first sends (U, δ, MAC, M) to R, R then verifies whether both Verify(δ, Q_s, K_R)=True and $MAC=H(K_R||M)$ hold. They claim their protocol is deniable, but we found it suffers SKCI attack. For that if K_R (= Ks) is revealed to E, and E intercepts the message flow (U, δ, MAC, M), he can use another message m' to compute $MAC'=H(K_R|| m')$ and send receiver the message flow (U, δ, MAC', m'). The receiver will be fooled. As for the **deniability**, it does not possess this property, because it cannot deny the source of the message due to the signature δ.

· **Harn and Ren's work [17]**

In 2008, Harn and Ren propose a fully deniable authentication service for E-mail applications [17]. Although, they claim that their scheme is fully deniable, on the contrary, we found it at most can be termed as a partially deniable scheme when the underlying scheme is ElGamal signature. Because the space cardinalities of both σ and C are different from the ones in the existential forgery. That means, it isn't perfect zero-knowledge and thus not a fully deniable protocol (which we will prove in claim 1 of Section 4.1). Moreover, using the existential forgeability to provide **deniability** is impractical, because signature algorithms suggest signing on the message digest rather than on the message directly.

· **Meng's work [18]**

In 2009, Meng [18] apply an NIDA protocol to voting systems and claim that their scheme is deniable. However, it suffers SKCI and KCI attacks, because if K is revealed to E, and E intercepts the message flow ($S_{pu}{}^{t}$, MAC, M), he can use another message m' to compute $MAC'=hash(K\|\, m')$ and send ($S_{pu}{}^{t}$, MAC', m') to the receiver. The receiver will be fooled. As for KCI attack, if E has the receiver's private key $\mathrm{R}_{\mathrm{PR},}$ he can compute $K'= [(S_{pu}{}^{t})^{hash(m')}]^{R_{PR}}$ and masquerade as the sender to send the receiver ($S_{pu}{}^{t}$, MAC', m'), the receiver will then be fooled.

· **Wang et al.'s scheme [15]**

In 2009, Wang *et al.* [15] propose an NIDA scheme based on designated verifier proofs. They claim that their scheme is deniable and unforgeable. However, this study found if E obtained message M and its authenticator Authen = (w, g^r, c, s) in the simulation phase, he can randomly pick (α, β, s), compute $c' = g^{\alpha}$, $A = g^{s}(y_{1p})^{-\beta}$, $B = h^{s}(y_{2p})^{-\beta}$ and $c = H(M, c', A, B)$, and finally compute $w =\beta\text{-}c$ and $g^{r} = g^{\alpha - w}/ y_{1v}$. Thus, E can successfully simulate (forge) the transcript Authen=(w, g^r, c, s), without the real value of r, to pass the designated verifier's verification.

· **Five recent NIDA protocols [16, 20-21, 32, 34]**

In 2011, Youn et al. [16], Zhang et al. [20], Shao et al. [21], and Hwang et al. [32] each proposes an NIDA protocol based on trapdoor commitment, generalized Elgamal signature, ECDLP and Schnorr signature scheme, respectively. Among them, though [16, 20] have the deniability, they suffer KCI attack. In addition, we found [20] further suffers SKCI attack and [21] is incorrect in defining both multiplicative and additive operations on a point group of an elliptic curve; scheme [32] has the deniability but suffers KCI and SKCI attacks. About the KCI attack in [32], if E compromised $x_{B,}$ he can impersonate Alice to send Bob $V'=H_1(g^{k'}\|m'\|R'\|y_B)$, $S'= g^{k'}.Y_A^{V'}$, $K'=H_2(S')^{x_B}$, $C'=E_{K'}(m'\|R')$. As for SKCI attack, if E compromised K and intercepted (C, V, S), he can decrypt C to obtain $m\|R$ and send Bob (C', V', S'), where $C'=E_K(m'\|R')$, $V'=H_1(g^{k'}\|m'\|R'\|y_B)$, *and* $S'= g^{k'}.Y_A^{V'}$, to pass Bob's authentication.

Most recently in 2013, J. Kar et al. [34] proposes an NIDA protocol using the generalized ECDSA signature scheme . However, their scheme has the KCI and SKCI attacks, because once *E* obtained Bob's secret d_b, he can impersonate Alice to send Bob $(U', MAC'(=\{ (H(M')Q_a+r'U')\} \cdot d_b), M')$, and thus be authenticated successfully. Meanwhile, if E got the common secret α_1 and knows $U(=kP)$ and r from the transferred message, where r = x_1 mod n and $x_1=(U)_x$, he can compute rUQ_b. From the equation, E knows $H(M)d_aQ_b=\alpha_1-rUQ_b$. He then can compute $H(M)^{-1}$ to multiply the equation's both sides and obtain d_aQ_b. By using this value and choosing a random k', E computes $U'= k'P$ and subsequently knows r'. Therefore, he can compute $\alpha_1'= H(M)' d_aQ_b+ r' k' Q_b$ and send (U', MAC', M') to Bob. Unconsciously, Bob will accept.

From the above-mentioned, we know that there still lacks a secure and efficient NIDA protocol. Therefore, in Section 3, we based on Fiat-Shamir heuristic [22] to propose an NIDA protocol, attempting to satisfy demand security features and get more efficiency than the other NIDA protocols. In Section 4, we prove that an NIDA protocol is deniable if and only if it is perfect zero-knowledge [23]. Our protocol can produce a *receiver-simulatable non-interactive proof.* It allows the designated *R* to simulate the real transcripts performed between *S* and himself. Such a design is similar to Jakobsson *et al.*'s "designated verifier proof" [30] that the designated verifier can always use his trapdoor to simulate any transcript initiated by *S*. Unfortunately, we found our scheme still suffers KCI attack. Hence, we doubt if an NIDA protocol inevitably suffers this attack. Regrettably, the answer is true. We prove this in Theorem 5 of Section 5.

3. The proposed scheme

Next followings are the details of our scheme, which is also illustrated in Fig. 1.

There exists a CA (Certificate Authority) to certify a user's public key $Y_u = -x_uP$, where $-x_u \in Z_q$ is the user's private key, and P is the base point of G_1. When Alice wants to authenticate message *m* to Bob, they cooperatively perform the following steps. Here, Alice's and Bob's public/private key pairs are $Y_A/-x_A$ and $Y_B/-x_B$, respectively, and the plaintext m is mapped by a hash function H_3 to a point M (=(m_1, m_2)) $\in G_1$, where m_1 and m_2 denote the x-coordinate and y-coordinate of M, respectively.

<u>Alice's part</u>

(1) Randomly chooses $k \in_R Z_q$, and computes $M = H_3(m)$, $T_B = kY_B$, and the *commitment* $T = kP$.

(2) Generates a random-looking *challenge*, CH, by applying ECC-based ElGamal encryption to N, where $N = M+R+T_B = (n_1, n_2)$ and R is a random point in G_1. For encrypting N, she performs the followings:

(a) randomly chooses $r \in_R Z_q$,

(b) computes $V = rP$, $W = rY_B$,

$C = N+ W$, and

$CH = \mathrm{ELG\_Enc}(N) = (V, C)$.

(3) Computes *response*, $rsp = k + (-x_A) * H_2(CH) \pmod q$.

(4) Computes hash value, $h = H_1(R)$.

(5) Sends (m, T, CH, rsp, h) to Bob.

<u>Bob's part</u>

After receiving (m, T, CH, rsp, h), Bob does the following.

(1) Verifies whether

$T = rsp \cdot P - H_2(CH) \cdot Y_A$. …… (E1)

If E1 does not hold, Bob rejects.

(2) Decrypts CH (= (V, C)) by using his private key, $-x_B$, obtaining N'. That is, computes

$W' = -x_B \cdot V$, …… (E2-1)

$N' = C - W'$. …… (E2-2)

(3) Computes $M = H_3(m)$, $T_B = -x_B T$, $R' = N' - M - T_B$, and verifies the following equation

$h = H_1(R')$. …… (E3)

If it holds, Bob accepts; otherwise, aborts.

Alice $\left\langle \begin{array}{l} \text{private key}: -x_A \\ \text{public key}: Y_A = -\, x_A P \end{array} \right.$ Bob $\left\langle \begin{array}{l} \text{private key}: -x_B \\ \text{public key}: Y_B = -\, x_B P \end{array} \right.$

. randomly chooses
$r, k \in Z_q$ and $R \in G_1$.
.computes $T = k \cdot P, T_B = kY_B$,
$M = H_3(m), N = M + R + T_B$,
$V = rP, W = rY_B$,
$CH = ELG\_Enc(M + R + T_B)$
$= (V, C) = (V, N + W)$,
$rsp = k + x_A * H_2(CH)$ (mod q), and
$h = H_1(R)$.

$\xrightarrow{m,\ T,\ CH,\ rsp,\ h}$

. verifies $T = ?\, rsp \cdot P - H_2(CH) \cdot Y_A$,
If the equation doesn't hold, aborts.
. decrypts CH to obtain N',
. computes $W' = -x_B \cdot V, N' = C - W'$,
$M = H_3(m), T_B = -x_B T$,
. extracts $R' = N' - M - T_B$,
. verifies $h = ?\, H_1(R')$,
If it equals, accepts.

Fig1. The proposed NIDA protocol

4. Deniability analysis

In this section, we introduce the concept of perfect zero-knowledge for relating it to the deniability property of NIDA protocols. We claim that an NIDA protocol is deniable if and only if it is perfect zero-knowledge. After that, we inspect the deniability of our protocol by using this claim.

4.1 Deniability for an NIDA protocol

From [1-3], we saw that "un-simulatability" of the receiver's view implies the sender's "un-deniability". However, we thought this definition on deniability is not enough. Examining the simulator in the NIDA schemes [6, 7, 8], we found that it reuses the signatures appeared in the real transcripts to compose the simulated ones. However, a signature has the undeniable characteristic. We, therefore, considered these schemes undeniable. That is, the sender cannot deny his participation in the protocol. Accordingly, we used perfect zero knowledge to inspect whether an NIDA protocol has the simulatability. Perfect zero knowledge [23] indicates that two transcript sets (produced by the simulator and the sender, respectively) are equal and their corresponding probability distributions are the same. For more clarity, we use perfect zero knowledge to rephrase the deniability of a NIDA protocol as follows.

Let λ be an NIDA protocol in which sender S can send a message m with its *proof* to receiver R and all messages transferred in a round make up a transcript. We denote the set of all possible valid transcripts for R (actually running λ with S) as $\mathcal{V}_R$. Assume that an efficient machine called *simulator*, *SIM*, can create $\mathcal{V}_R$ by input R's private key as if it were from the real protocol run (with S). If we denote the set of all possible transcripts produced by *SIM* as $\mathcal{V}_{SIM}$, then we claim λ is deniable if and only if it is perfect zero-knowledge. That is, $\mathcal{V}_R = \mathcal{V}_{SIM}$, and for any $\mathcal{T}_R \in \mathcal{V}_R$ there exists a $\mathcal{T}_{SIM} \in \mathcal{V}_{SIM}$, satisfying $\mathcal{T}_R = \mathcal{T}_{SIM}$ and $\Pr[\mathcal{T}_R] = \Pr[\mathcal{T}_{SIM}]$. We prove the related claims as follows.

Claim 1. λ is deniable iff it is perfect zero-knowledge.

Proof: For “<=”, it is obvious, because the indistinguishability between $\mathcal{V}_R$ and $\mathcal{V}_{SIM}$ makes S deny any transcript of λ. Next we prove “=>” by contraposition. That is, if an NIDA protocol does not have the perfect zero-knowledge property, then it is undeniable. Without loss of generality, suppose that there exists a valid transcript $\mathcal{T}$ and its probability distribution in $\mathcal{V}_R$ is different from the one ($\mathcal{T}$) in $\mathcal{V}_{SIM}$ with a non-negligible probability. Then, S cannot deny $\mathcal{T}$, because with non-negligible advantage, one can determine from which set, $\mathcal{V}_R$ or $\mathcal{V}_{SIM}$, $\mathcal{T}$ comes. We prove the claim.

Claim 2. λ is simulatable iff it is deniable.

Proof: For “<=”, we prove this by contraposition. If the receiver cannot simulate one of the transcripts, then this transcript must come from the real protocol run under the sender’s cooperation. That is, the sender cannot deny his participation in generating the transcript. Next, we prove “=>”. This directly comes from the deniability definition in [3]. We have thus proven the claim.

4.2 The deniability of our protocol

In this section, we used Claim 1 to inspect the deniability of our protocol. Before that, we proved our protocol to be perfect zero-knowledge by using three moves: (I) construct an efficient *SIM* to generate a valid transcript, (II) analyze the cardinality and probability distributions for spaces $\mathcal{V}_{SIM}$ and $\mathcal{V}_R$, respectively, and (III) show that sets $\mathcal{V}_R$ and $\mathcal{V}_{SIM}$ are identical. For simplicity, we omit the notations “mod *q*“ which are supposed to appear in the expressions.

(I) Construct an efficient *SIM*.

Assume that Alice and Bob execute the protocol honestly and generate a transcript, (T, CH, rsp, h) for message m. In the following, we construct an efficient simulator *SIM* to forge this transcript. On input the public parameters $(q, G_1, P, H_1, H_2, H_3)$, message m, Alice's public key Y_A, Bob's public key Y_B, and Bob's private key $-x_B$, *SIM* does the following steps.

Step 1. Randomly chooses $rsp', r' \in_R Z_q$, and $C' \in_R G_1$.

Step 2. Computes $V'= r'P$, $W' = -x_B V'$, $N'= C'- W'$

Step 3. Sets $CH' = (V', C')$, and computes $T' = rsp' \cdot P - H_2(CH') \cdot Y_A$.

Step 4. Computes $M = H_3(m)$, $T_B' = -x_B T'$, $R' = N' - M - T_B'$ and $h' = H_1(R')$.

Step 5. Outputs (T', CH', rsp', h') for message m.

It is obvious that this forged (simulated) transcript (T', CH', rsp', h') is valid and *SIM* runs efficiently.

(II) Analyze the cardinality and probability distribution for spaces $\mathcal{V}_{SIM}$ and $\mathcal{V}_R$, respectively.

(II.A). Analyze space $\mathcal{V}_{SIM}$

Considering the given simulated transcript, $\mathcal{T}_{SIM} = (T', CH', rsp', h') \in \mathcal{V}_{SIM}$ for m, the occurance probability can be determined by the randomly chosen elements $rsp' \in_R Z_q$, and $V', C' \in_R G_1$. Since from *SIM's* computations, we can see that

(i) CH' is formed by V' and C' in which $V'= r'P$, and C' is a random point in G_1. Hence, the cardinality $|CH'|$ is q^2.

(ii) T' is computed from rsp' and CH' ($|T'|$ hence is q^3),

(iii) Because $T_B{}' = -x_B T'$, when T' is obtained, with Bob's private key, $T_B{}'$ can be determined as well.

(iv) *Since* $h' = H_1(R')$ $(R' = N' - M - T_B')$, $N' = C'-W'$, and $W' = -x_B V'$, when rsp', C', and V' are determined, CH', T', T_B', N' and R' can all be determined as well. Thus, h' $(=H_1(R'))$ is also determined.

In short, the space cardinality of $\mathcal{V}_{SIM}$ for M, $|\mathcal{V}_{SIM}|$, is q^3.

Hence, under uniform distribution of the random parameters, the occurrence probability of any simulated transcript $\mathcal{T}_{SIM} \in \mathcal{V}_{SIM}$ is

$$\Pr[\mathcal{T}_{SIM}] = (1/q^3).$$

(II.B) Analyze space $\mathcal{V}_R$

Consider a real transcript $\mathcal{T}_R = (T, CH, rsp, h) \in \mathcal{V}_R$ for message m. The cardinality is determined by the random numbers $k, r \in_R Z_q$, and random point $R \in_R G_1$. Thus, the cardinality of space $\mathcal{V}_R$ is

$|\mathcal{V}_R| = q^3$.

Hence, under the uniform distribution of the random parameters, the occurrence probability of any real transcript $\mathcal{T}_R \in \mathcal{V}_R$ for m is

$\Pr[\mathcal{T}_R] = (1/\, q^3)$.

(III) Show that $\mathcal{V}_R$ and $\mathcal{V}_{SIM}$ are identical.

From (II), we see that $|\mathcal{V}_R| = |\mathcal{V}_{SIM}|$, and their probability distributions of $\mathcal{V}_R$ and $\mathcal{V}_{SIM}$ are the same. Hence, to show the perfect zero-knowledge property of our protocol, we need to prove that for any $\mathcal{T}_R \in \mathcal{V}_R$, we can find a $\mathcal{T}_{SIM} \in \mathcal{V}_{SIM}$, such that $\mathcal{T}_R = \mathcal{T}_{SIM}$. That is, given m's $\mathcal{T}_R = (T, CH\ (=(V, C)), rsp, h) \in \mathcal{V}_R$, find its correspondent $\mathcal{T}_{SIM} = (T', CH'\ (=(V', C')), rsp', h') \in \mathcal{V}_{SIM}$, satisfying $\mathcal{T}_R = \mathcal{T}_{SIM}$. For this purpose, we can do as follows.

(i) Since r', rsp' and C' can be arbitrarily chosen by SIM (as done in [**I**]'s Step 1) and $|\mathcal{V}_R|=|\mathcal{V}_{SIM}|$, when given $\mathcal{T}_R$ there must exist a transcript $\mathcal{T}_{SIM}$ in $\mathcal{V}_{SIM}$ satisfying $r'=r$, $rsp'=rsp$, and $C'=C$.

(ii) Once V' $(= r'P)$ and C' have been determined, the values of N' $(=(C'-W')=(C'-(-x_B)V')$, T' $(=rsp'\cdot P - H_2(CH')\cdot Y_A)$, and T_B' $(=-x_B\, T')$ can all be determined as well.

(iii) Under determined N' and T_B', the value R' $(=N' - M - T_B')$, which is equal to R in the transcript, can be uniquely determined.

From the above stated, we found a transcript $(T', CH'\ (=(V', C')), rsp', h')$ which equals to $\mathcal{T}_R$ and belongs to $\mathcal{V}_{SIM}$ with the same probability distribution. Therefore, we proved that our protocol possesses perfect zero-knowledge property. According to Claim 1, we concluded that our protocol is deniable.

5. Security analyses and comparisons

In this section, we first show the security analyses and then make comparisons with other works in the aspects of security and performance in section 5.1 and 5.2, respectively.

5.1 Security analyses

We examined our protocol by using some properties which an NIDA protocol demands. By using Theorem 1 through Theorem 4, we show that our scheme possesses correctness, unforgeability, authenticability, and SKCI resistance, respectively. Theorem 5 indicated that the deniability property of an NIDA protocol conflicts with KCI resistance. Finally, Table 1 compares three properties: SCKI resistance, KCI resistance, and deniability, among our scheme and protocols [6-10, 16-20, 32, 34].

Theorem 1. (Correctness) The proposed scheme is correct.

Proof: When Alice follows the protocol, equation E1 (verified by Bob) will hold since

$$rsp \cdot P - H_2(CH) \cdot Y_A = (k + (-x_A) * H_2(CH)) \cdot P - H_2(CH) \cdot (-x_A \cdot P) = kP = T.$$

Similarly, from the following three deductions

(1) $W' = -x_B \cdot V = -x_B \cdot (rP) = r \cdot (-x_B P) = rY_B$,
(2) $ELG\_Dec(CH) = N' = C' - W'$,
(3) $R' = N' - M - T_B = N - M - T_B = R$,

we can see that equation E3, $h = H_1(R')$, holds as well

Theorem 2. (Unforgeablity) With a negligible probability, E could produce a transcript to be successfully verified by Bob.

Proof: Although in Fiat-Shamir heuristic, the non-interactive proof generated can hardly be forged without sender's private key, our modified version leaves a trapdoor for the receiver to generate (forge) a valid one. In our modification, without sender's private key, the only possible way for E to forge m's transcript is to simulate the receiver. However, without receiver's private key $-x_B$, E cannot decrypt the random *challenge* (ciphertext), $CH'(=(V', C')=(r'P, N'+(-x_B)V') = (V', (M + R' + T_B') + (-x_B)V')$, for producing a valid pair (R', $T_B'(= -x_B T)$) in plaintext $N'(=M+R'+T_B')$ to satisfy $ELG\_Enc(N') = CH'$. Therefore, we concluded that the probability E could produce a valid transcript is less than breaking the ElGamal cryptosystem. Since if ElGamal cryptosystem is broken and E thus obtains N', he cannot extract R' to obtain value h without the knowledge of $-x_B$.

Theorem 3. (Authenticity) As long as Alice follows our protocol honestly, Bob can authenticate both Alice and her sent message.

Proof: When Alice follows the protocol honestly, the parameters T, $CH=(V, C)$, rsp, and h in the message flow would be generated correctly. Obviously, on receiving the message flow, Bob can use Alice's public key Y_A to verify equation E1 successfully. Then, decrypts CH by his secret key, obtaining N', as equation E2 illustrates. After this, he can compute $R'=N'-M-T_B$ and hence verify equation E3 successfully. It means that the authentications of both Alice's identity and her transmitted message m can be satisfied. This completes the proof.

Theorem 4. The proposed scheme can resist SKCI attack.

Proof: Because our scheme doesn't require both communicating parties to compute a session key for generating the MAC-based authenticator as a proof (which occurs in the previous works often). Thus, our work is free from SKCI attack.

Theorem 5. If a non-interactive authentication (NIA) protocol is deniable, then it inevitably suffers from KCI attack.

Proof: For there is only one message flow in an NIDA protocol, the possible KCI attack is pretending Alice to communicate with Bob. i.e. *E* compromises Bob's private key and impersonates Alice to communicate with Bob. We prove this theorem by contraposition. That is, if an NIA protocol can resist KCI attack, then it does not have the deniability property. Assume *E* knows Bob's private key but cannot impersonate Alice to communicate with Bob, which implies that some component of a real transcript produced by Alice cannot be forged by *E*. That means, even with Bob's private key, the unforgeable component of the actual transcript cannot be efficiently produced by a simulator. Therefore, from Claim 2, the protocol does not have the deniability property. We prove the theorem.

Table 1: security property comparisons among NIDA protocols

Scheme	Approach	SKCI resistance	KCI resistance	deniability
[6]	ElGamal signature-based	*	*	No
[7]	Weil paring signature-based	*	*	No
[8]	QR signature-based	*	*	No
[9]	Weil paring ID-based (but using implicit shared secrecy)	No	No	Yes
[10]	ElGamal signature-based (but using implicit shared secrecy)	Yes	No	Yes
[16]	RSA-based(based on trapdoor commitment)	Yes	No	Yes
[17]	Fully Deniable Authentication Service for E-mail	*	*	No
[18]	A Secure Internet Voting Protocol	No	No	Yes
[19]	Identity-based deniable authentication protocol	*	*	No
[20]	based on generalized ElGamal signature scheme	No	No	Yes

[32]	based on Schnorr signature scheme	No	No	Yes
[34]	using generalized ECDSA signature schemes	No	No	Yes
Ours	ECC-based	Yes	No	Yes

***: means don’t care, because it lacks deniability**

5.2 Performance Comparisons

From Table 1, we see that our protocol and [10, 16] are competitive in the aspects of required security properties. Therefore, Table 2 makes efficiency comparisons only among them, in the aspects of both computational and communication cost. Schemes [10, 16] are designed from ElGamal signature and RSA, respectively, while ours from modified Fiat-Shamir heuristic. Moreover, for efficiency consideration, we implement our protocol by using elliptic curve cryptography and thus make it more efficient in both computational and communication cost than the others. We can see the outcome in Table 2. It results from [31] which states that one exponentiation multiplication (EXP) is about 255 times the cost of a 1024-bit modular multiplication (MM) and one ECC-point multiplication (ECC-mul) is about 29 MM. In addition, the proposed protocol requires two and one ECC-mul for Elgamal encryption and decryption, respectively. Hence, for the same security level, our scheme requires only 253 MM in computational cost and 800 bits in communication size; whereas scheme [10] needs 1275 MM in computation and 1184 bits for communication and [16] requires 1020 MM in computation and 1344 bits for communication.

Table 2: a performance comparison among schemes [10, 16] and ours

Scheme	Sender’s computation	Receiver’s computation	Total computation	Size in communication
[10]	510 MM (2 EXP + 2 H)	765 MM (3 EXP + 2 H)	1275 MM	1184 bits
[16]	510 MM (2 EXP + 2H)	510 MM (2 EXP + 2H)	1020 MM	1344 bits
Ours	116 MM (4ECC-mul + 3H)	116 MM (4 ECC-Mul + 3H)	232MM	800 bits

MM: 1024-bit modular multiplication, **EXP**: g^k mod p, where $|q|$ is 160 bits, $|hash|$ is 160 bits, and $|p|$ 1024 bits, **ECC-mul**: ECC point multiplication, **H**: hash, 1**EXP**≒255MM, 1**ECC-mul**≒29MM, Sym-Encr denotes symmetric encryption, and Sym-Dec symmetric decryption

6. Conclusion

Many non-interactive deniable authentication protocols have been proposed.

Among them, schemes [6, 7, 8, 17, 19] cannot achieve deniability. Schemes [9, 10, 16, 18, 20, 32, 34] although can achieve deniability; however, suffer either SKCI attack or KCI attack. For avoiding the drawbacks, we proposed a novel ECC-based NIDA protocol by modifying Fiat-Shamir heuristic. After comparing, it shows that our scheme can achieve full deniability and attain better efficiency than [10, 16] which are competitive to ours in the three demand security features. In addition, we proved the equivalence of perfect zero knowledge and deniability for an NIDA protocol in Claim 1. According to this argument, we showed our protocol is deniable. Moreover, we also proved that our scheme has unforgeability, authenticability, and SKCI resistance by Theorem 2 through 4. We doubted if an NIDA protocol inevitably suffers KCI attack. Unfortunately, Theorem 5 confirms this suspecting.

From the works shown in Table 2, we concluded that the proposed scheme outperforms schemes [10, 16] in both computational and communication cost. Therefore, our research was more suitable to be applied in real applications, e.g., a secure Internet voting system.

Reference

[1] C. Dwork, M. Naor, A. Sahai, “Concurrent zero-knowledge, ” *Proceedings of 30th ACM STOC’98*, 1998, 409–418.

[2] Y. Aumann, M. Rabin, “Efficient deniable authentication of long messages,”*Int. Conf. on Theoretical Computer Science in Honor of Professor Manuel Blum’s 60th birthday*, http://www.cs.cityu.edu.hk/dept/video.html. April 20–24, 1998.

[3] Mario Di Raimondo, Rosario Gennaro and Hugo Krawczyk, “Deniable Authentication and Key Exchange, ”*ACM CCS'06*, October, 2006, Alexandria, Virginia, USA.

[4] C. Boyd, W. Mao, K. Paterson, “Deniable authenticated key establishment for Internet protocols, ”*11th International Workshop on Security Protocols*, Cambridge (UK), April 2003.

[5] C. Boyd & W. Mao, “Key agreement using statically keyed authenticators, ”*Applied Cryptology and Network Security (ACNS’04), LNCS 3089*, 248–262.

[6] Z. Shao, “Efficient deniable authentication protocol based on generalized ElGamal signature scheme, ”*Computer Standards & Interfaces 26 (5)*, 2004, 449–454.

[7] R. Lu, Z. Cao, “A new deniable authentication protocol from bilinear pairings, ”*Applied Mathematics and Computation 168 (2)*, 2005, 954–961.

[8] R. Lu, Z. Cao, “Non-interactive deniable authentication protocol based on

factoring, ”*Computer Standards & Interfaces 27 (4)*, 2005, 401–405.

[9] Tianjie Cao , Dongdai Lina and Rui Xue, “An efficient ID-based deniable authentication protocol from pairings, ”*Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA’05)*, IEEE, 2005.

[10] Wei-Bin Lee, Chia-Chun Wu and Woei-Jiunn Tsaur, “A novel deniable authentication protocol using generalized ElGamal signature scheme, ” *Information Science*, 2006.

[11] H. Krawczyk, “SIGMA: The SIGn and MAc approach to authenticated Diffie-Hellman and its use in the IKE protocols, ”In D. Boneh, editor, *Advances in Cryptology – Crypto 2003, LNCS 2729*, 400–425.

[12] H. Krawczyk, “SKEME: a versatile secure key exchange mechanism for Internet, ”*IEEE SNDSS ’96*, IEEE Press 1996, 114–127.

[13] Rongxing Lu and Zhenfu Cao, “Erratum to “Non-interactive deniable authentication protocol based on factoring”, ”Computer Standards & Interfaces 27 (2005) 401–405,” *Computer Standards & Interfaces 29*, February, 2007, 275.

[14] Chun-Ta Li, Min-Shiang Hwang and Chi-Yu Liu, “An electronic voting protocol with deniable authentication for mobile ad hoc networks, ”*Computer Communication 31(10)*, June 2008, 2534–2540.

[15] Bin Wang and ZhaoXia Song, “A non-interactive deniable authentication scheme based on designated verifier proofs, ”*Information Sciences 179(6)*, March 2009, 858–865.

[16] Taek-Young Youn, Changhoon Lee and Young-Ho Park, “An efficient non-interactive deniable authentication scheme based on trapdoor commitment schemes, ”*Computer Communications 34(3)*, March 2011, 353–357.

[17] Lein Harn and Jian Ren, “Design of Fully Deniable Authentication Service for E-mail Applications, ”*IEEE Communications Letters 12(3)*, March 2008, 219–221.

[18] Bo Meng, “A Secure Internet Voting Protocol Based on Non-interactive Deniable Authentication Protocol and Proof Protocol that Two Ciphertexts are Encryption of the Same Plaintext, ”*Journal of Networks 4(5)*, July 2009, 370–377.

[19] Y. Shi and J. Li, “Identity-based deniable authentication protocol, ”*IET ELECTRONICS LETTERS 41(5)*, 3rd March 2005

[20] Yuan Zhang, Qiuliang Xu, and Zhe Liu, “A new non-interactive deniable authentication protocol based on generalized ElGamal signature scheme, “*Proc. in 13th IEEE Joint International Computer Science and Information Technology Conference (2011).*

[21] Fei Shao and Bo Meng, “A Non-interactive Deniable Authentication Protocol

based on Elliptic Curve Discrete Logarithm Problem, “*Energy Procedia 11 (2011)*, 1018–1025.

[22] A. Fiat and A. Shamir, “How to prove yourself: practical solutions of identification and signature problems, ”*Advance in Cryptology – Proceeding of CRYPTO’86, LNCS 263*, 186–194.

[23] D. R. Stinson, *Cryptography Theory and Practice*, 1995, CRC press.

[24] R. Gennaro, “Using non-interactive proofs to achieve independence efficiently and securely, ”*Massachusetts Institute of Technology Technical Report: TM-515*, 1994.

[25] M. Blum, A. DeSantis, S. Micali, and G. Persiano, “Noninteractive zero-knowledge, ”*SIAM Journal of Computing, 20(6)*, December 1991, 1084–1118.

[26] J. Groth and S. Lu, “A non-interactive shuffle with pairing based verifiability, ” *ASIACRYPT 2007, LNCS #4833*, 2008.

[27] S. Blake-Wilson, D. Johnson, and A. Menezes, “Key agreement protocols and their security analysis, ”*In Sixth IMA International Conference on Cryptography and Coding, LNCS #1355*, 1997, 30–45.

[28] J. S. Chou, Y. Chen, and J. C. Huang, “An ID-Based Deniable Authentication Protocol on pairings, ”http://eprint.iacr.org/2006/335, 2006.

[29] Wenbo Mao, *Modern Cryptography: Theory and Practice*, Prentice Hall PTR, 2003.

[30] M. Jakobsson, K. Sako, and R. Impagliazzo, "Designated verifier proofs and their application, "*Advance in Cryptology – Proceeding of EUROCRYPT’96, LNCS 1070*, 143–154.

[31] Jue-Sam Chou, Yalin Chen and Tsung-Heng Chen, “An efficient session key generation for NTDR networks based on bilinear paring, ”*Computer Communication 31(14)*, September 2008, 3113–3123.

[32] Shin-Jia Hwang and Yun-Hao Sung, “Confidential deniable authentication using promised signcryption, “The journal of System and Software, 84 (2011), 1652-1659

[33] Jayaprakash Kar, “ID-based Deniable Authentication Protocol based on Diffie-Hellman Problem on Elliptic Curve, ” *International Journal of Network Security, Vol.15, No.1*, PP.295-302, Jan. 2013

[34] J. Kar, D. M. Alghazzawi, and S. H. Hasan, “A novel non-interactive deniable authentication protocol using generalized ECDSA signature schemes with message recovery, ” *INFORMATION –An International Interdisciplinary Journal, Japan, Vol: 16, No: 1(B)*, January 2013, Page: 807-813, ISSN: 1343-4500.