A Novel Non-interactive Deniable Authentication Protocol with Designated Verifier on elliptic curve cryptosystem

Yalin Chen[1], *Jue-Sam Chou[2], Chi-Fong Lin[3]

[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
jschou@mail.nhu.edu.tw
[3] Department of Information Management, Nanhua University
chejtmmc@gmail.com

Abstract

Recently, many non-interactive deniable authentication (NIDA) protocols have been proposed. They are mainly composed of two types, signature-based and shared-secrecy based. After reviewing these schemes, we found that the signature-based approach can not deny the source of the message and thus can not achieve full deniability; and that, the shared-secrecy based approach suffers KCI attack although it can achieve full deniability. In addition, both types of schemes lack efficiency consideration for they mainly base on DLP, factoring, or bilinear pairing. Due to this observation, in this paper, we use the Fiat-Shamir heuristic method to propose a new ECC-based NIDA protocol which not only can achieve full deniability but also is more efficient than all of the proposed schemes due to the inheritent property of elliptic curve cryptosystem. Further, we prove the properties of full deniability and KCI resistance conflict for a NIDA protocol. Besides, we deduce that a NIDA protocol is deniable if and only if it is perfect zero-knowledge.

Keyword: deniability, deniable authentication protocol, Fiat-Shamir heuristic, perfect zero-knowledge, simulatability, key compromise impersonation attack, elliptic curve cryptosystem.

1. Introduction

The basic security requirements such as integrity, confidentiality, non-repudiation, and authentication have been paid much attention over Internet communications. Recently, the property of “deniability” is getting attractive more and more since it can protect personal privacy which we often need in the real life or business activities. For example, a bidder of an action may not expect the content of his bid revealed to a third party. Even, he may wish nobody knows his participation. Under this requirement, the property must let the bidder be able to deny his participation if an unexpected event occurs. In a digital world, this privacy requirement can be implemented by a *deniable authentication protocol* in which the receiver can verify the authenticity of both the message and the sender, but afterwards the sender can deny to a third party that he had sent the message. Up to now, there are many schemes proposed in this area [1-10, 22-26].

In 1998, Dwork *et al.* [1] first proposed a deniable authentication protocol based on concurrent zero-knowledge proof. Their study permits a sender *S* to authenticate a message for a receiver *R*, but a third party can not verify the authentication. In other words, it does not permit *R* to convince a third party that *S* has authenticated *m* to him–as if there were no "paper trail" of the conversation left between them. In the same year, Aumann and Rabin [2] proposed another deniable authentication protocol based on factoring. They mentioned that if *R* can simulate all the communications between him and *S*, then *S* can deny the communications that he had ever taken. In 2006, Raimodo and Krawczyk [3] define an authentication and key exchange protocol to be deniable if *R*'s view (all the information that *R* obtains by participating in the protocol) can be simulated by an efficient machine (called the *simulator*) which doesn't know *S*'s secret key. Here, if *S*'s secret key needs to be known, then the deniability property fails since only *S* should know his secret key; and an efficient machine (simulator) means it can construct the transcripts without relying on deducing *S*'s secret key (from his public key). In addition, they also proposed the notion of "partial deniability" for SIGMA protocol [11] which uses non-repudiable signature for authentication; and proposed the notion of "full deniability" for SKEME protocol [12] which uses encryption-based method for the same purpose. In their deniability definition, *S* can only deny the content of message *m*, but for a fully deniable one, *S* can deny both the content of message *m* and where it comes from. From literatures [1-3], we can see that "simulatability" of the receiver's view implies "deniability" of the sender. In 2008, Li *et al.* [23] apply the deniable authentication property in an electronic voting protocol for mobile ad hoc networks. However, in their scheme, if the system is not equitable, it can impersonate any other voter to vote. As a result, the vote can not convince anyone. Conversely, if the system is equitable, there is no necessity for the system to vote for any voter. In other words, their application is not suitable.

Except for the schemes mentioned above, many non-interactive deniable authentication (NIDA) protocols [6-10, 22, 24-26] have been proposed. We classify these NIDA schemes into two types: (a) signature-based, and (b) shared-secrecy based. If an item (which is regarded as a signature by the sender) in the sent message can not be reproduced by the receiver. We term such a protocol as a signature-based protocol. Otherwise, we term it a shared-secrecy based protocol. By our classification, [6, 7, 8, 22] are signature-based and [9, 10, 24, 25, 26] are shared-secrecy based (Although this might not be consistent with the titles mentioned in the original papers.). In these schemes, a message with its *proof* produced by a sender is sent to a receiver in only one pass. Then, the receiver can use 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 had sent this message. Except for this property, a non-interactive protocol generally has the advantage of communication efficiency for its only one-pass communication and hence can be applied to off-line applications such as sending e-mails or signing documents. However, we found that these schemes either have security vulnerabilities or can not achieve the goal of full deniability. Below, we roughly describe the main frameworks of these schemes.

For signature-based schemes [6, 7, 8, 22], the sender *S* signs on a random nonce *r* as *sig_r*, encrypts *r* as *enc_r* by using *R*'s public key, and computes a MAC-based proof, *proof* = *HMAC* (*r*, *m*), which thus implicitly binds *m* with *S*'s *r*-related signature *sig_r*. *S* then sends the four-tuple message flow (*sig_r*, *enc_r*, *proof*, *m*) to *R*. After receiving this message flow, *R* first decrypts *enc_r* to obtain *r*, and then uses *r* to verify both *sig_r* and *proof*. If both verifications are valid, *R* accepts the message. The authors claimed that in their schemes, *S* can deny the content of *m* because *R* can choose arbitrary *m'* and computes *proof'* = *HMAC* (*r*, *m'*). Then, this newly formed four tuple (*sig_r*, *enc_r*, *proof'*, *m'*) is still a valid transcript. However, it can be easily seen that as long as *R* reveals *r*, *S* can not deny his *sig_r* on *r* because *sig_r* can not be efficiently simulated (forged) due to the unforgeability of a signature. More precisely, *S* can not deny his action of sending the four-tuple message flow. Hence, we consider that these signature-based NIDA schemes can not achieve the fully deniability (which means the sender can deny both the content of the message and its sender). Except for the unsatisfaction of full deniability, schemes [7, 8] also have another security hole. They suffer from the session key compromise impersonation attack as pointed by [10], which we denote as SKCI. From now on, we will use “deniability” to stand for “full deniability”.

For shared-secrecy based schemes [9, 10, 24, 25, 26], *S* and *R* use the pre-shared secrecy to achieve mutual authentication, secret communication and deniability. However, we found using this type of approach will result in KCI (key compromise impersonation) attack. KCI is a security notion which means that the loss of a user *u*'s secret value would enable an adversary *E* to impersonate any other party to communicate with *u* [13]. According to this definition, we know that there are two possible ways for *E* to launch such a KCI attack in an interactive two-party (say *A* and *B*) protocol: *E* compromises *A's* (or *B's*) private key and then impersonates *B* (or *A*) to communicate with A (or B). But for a one-pass NIDA protocol, only one KCI launching is possible that E compromises R's private key, and then impersonates S to authenticate a message m' to R. Therefore, we consider schemes [9, 10, 25] suffer from KCI attack since once E compromises R's secrecy (which is also the secrecy of S's), he can easily impersonate S to communicate with R. The similar argument for a deniable shared-secrecy based IKE (Internet key exchange) protocol can be seen in [5]. For scheme [24], we will demonstrate its drawback in Section 2.3. As for scheme [26], although it claims their scheme is fully deniable, we found it can not attain their goal. At most, it can be termed as a partially deniable authentication 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 a perfect zero-knowledge scheme (which we will describe in this paper).

From the above-mentioned, we know there still lacks a secure and complete NIDA protocol. (We will give a more detail drawback description about these schemes in Section 2.2 and 2.3, respectively.) Therefore, in this paper, we base on Fiat-Shamir heuristic [15] to propose a NIDA protocol, attempting to resolve the weaknesses found in both of the signature-based and shared-secrecy based schemes. After that, we prove a NIDA protocol is deniable if and only if it has the property of perfect zero-knowledge [17]. Our protocol can produce a *receiver-simulatable non-interactive proof*. It allows the designated R to simulate the real transcripts formed by S and him. Such a designation is similar to Jakobsson *et al.*'s "designated verifier proof" [16] that the designated verifier can always use his trapdoor to simulate any transcript initiated by S. The details will be discussed in Sec. 4. Unfortunately, we found our scheme still suffers from KCI attack. Hence, we go a step further to prove that the property of deniability conflicts with the property of KCI resistance, for a NIDA scheme. We will discuss and prove it in Sec. 5.

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 detailed discussion of previous work. In Sec. 3, we propose our protocol. The analyses of its deniability and other security features are described in Sec. 4 and Sec 5, respectively. Finally, a conclusion is given in Sec. 6.

2. Background and Related Work

In the following, we first give the definitions of used notations in this paper. After that, the Fiat-Shamir heuristic is introduced in Sec. 2.1 and then we discuss the three signature-based NIDA protocols [6, 7, 8] and the three shared-secrecy based approaches [9, 10, 25] together with their corresponding drawbacks in Sec. 2.2 and

Sec. 2.3, respectively.

Definitions of used notations:

p, q: two large primes satisfying $q|(p\text{-}1)$,
G: a group of order q,
g: the generator of G,
G_1, G_2: groups of order q on an elliptic curve,
P: a primitive element of G_1,
x_i: user i's private key,
Y_i: user i's public key which equal to g^{x_i} in DLP scheme or x_iP in pairing scheme,
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.

2.1 Fiat-Shamir heuristic

In 1986, Fiat and Shamir [15] suggested a heuristic means for designing a secure digital signature scheme which enables a user to prove his identity and authenticate his message by the following two steps:

(1) Choose a secure 3-pass identification scheme, e.g., Schnnor's identification scheme[17] in which the output transcript in each round is denoted as (*commitment*, *challenge*, *response*), where *commitment* is the first flow from the prover to the verifier, *challenge* is the second flow from the verifier to the prover, and *response* is the last flow from the prover to the verifier.

(2) Choose a secure hash function $\mathcal{H}$ to produce the *challenge*. When a signer wants to sign on a message *m*, he executes the above identification scheme by himself to produce an acceptable transcript (*commitment*, *challenge*, *response*) as his signature on *m*, where *challenge* equals $\mathcal{H}$(*commitment*, *m*). That is, he first generates the commitment then hashes the commitment with message m to produce the challenge and finally computes the response according to the identification scheme.

For clarity, in the following, we demonstrate the above two steps by first adopting Schnorr Identification Scheme in step (1) and supposing that a signer has his private key $SK = x \in_R Z_q$ and public key $PK = g^{-x}$. Then for step (2), when signing on message m, the signer computes commitment $t = g^k$, challenge $ch = \mathcal{H}(t, m)$, and response $s = k$

+ $SK * ch$, and forms (t, s) as his signature. The signature (t, s) can then be publicly verifiable by checking whether $t = g^{s}(PK)^{\mathcal{H}(t\|m)}$ holds or not.

The Fiat-Shamir heuristic is also treated as an efficient way in building non-interactive zero-knowledge proofs [18, 19, 20]. Its security is based on the secure hash function $\mathcal{H}$ and can be proved in the random oracle model [20].

2.2 Review of signature-based NIDA schemes

In 2004, Shao [6] proposed a signature-based scheme using generalized ElGamal signature. In the scheme, when Alice wants to send a message with its authenticator to Bob, she randomly chooses $t \in Z_q$ and computes $k = Y_A{}^{t}$ (mod p), $r = H(k)$, $s = t - x_A \cdot r$ (mod q), and $auth = H(k\|m)$. Then, she 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] pointed out Shao's scheme has a vulnerability that once the session key k was compromised, the attacker can take arbitrary message m' to form another valid $auth' = H_2(k\|m')$. Then, he can impersonate Alice to send Bob $(r, s, auth', m')$. Bob would then be fooled because he will extract the same k from the old (r, s) and thus verify $auth'$ as valid. We denote such an attack as SKCI attack. Except for the SKCI attack pointed by Lee *et al.*, this study also found Shao's scheme lacks the deniability property since nobody other than Alice can efficiently make the signature s on r. Hence, as long as Bob reveals both k and $r = H(k)$ to a third party, Alice can not deny her signature s . In addition, although R could arbitrarily produce $k' = (g^{s'} Y_A{}^{r'})^{x_B}$ by randomly choosing s' and r', the equation $r' = H(k')$ can be hardly satisfied because the probability that the hash value of k' would be equal to a pre-defined value r' is negligible. This demonstrates the undeniability of Shao's scheme.

In 2005, Lu and Cao [7] proposed a signature-based NIDA scheme based on Weil paring. In their scheme, when Alice wants to send a message m with its authenticator to Bob, she first randomly chooses $t \in Z_q$ and computes $r = H_1(e(P, P)^{t})$, $s = \frac{t}{r + x_A} Y_B$, and $auth = H_2(\hat{e}(P, P)^{t}, m)$. Then, she sends $(r, s, auth, m)$ to Bob. After receiving $(r, s, auth, m)$, Bob extracts the session key $k = \hat{e}(P, P)^{t}$ by using the session parameters (r, s), Bob's private key, and Alice's public key, i.e. $\hat{e}(s, x_B{}^{-1}(rP+Y_A)) = \hat{e}(P, P)^{t}$. Meanwhile, in 2005, Lu and Co [8] also proposed a signature-based 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 a random nonce r and (c, a_1, a_2) is r encrypted by Bob's public key. After receiving Alice's message, Bob first decrypts (c, a_1, a_2) to obtain r, and then verifies Alice's signature, (s, b_1, b_2) on r. If it is valid, Bob believes that the message is sent from Alice.

However, in 2006, Lee *et al.* [10] pointed out that both [7] and [8] have the vulnerability of SKCI attack. In addition, this study also found both schemes lack the deniability property. Because in [7], as long as R reveals $k = \hat{e}(P, P)^t$ and $r = H_1(\hat{e}(P, P)^t)$ to a third party, S can not deny his signature s on r; and in [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. That is, when Bob reveals (s, b_1, b_2, r) to a third party, Alice can not deny that she had ever sent (s, b_1, b_2, c, a_1, a_2, $auth$, m) to Bob. Moreover, in 2007, Lu *et al.* [22] proposed an improvement on [8] to include both identities of the two communicating parties. However, this study found their improvement still has the same deficiency as that existed in [8].

2.3 Review of shared-secrecy based NIDA schemes

In this section, we first introduce the conflict of deniability and KCI resistance in shared-secrecy based deniable protocols from studies [5, 14]. Then, we review three shared-secrecy based NIDA protocols, [9, 10, 24], and show they suffer from KCI attack.

In 2004, Boyd and Mao [5] pointed out that the two properties of allowing deniability and preventing KCI attack in the shared-secrecy based IKE protocol [4] conflict. For in [4], once the shared secret key of a party, say A (or B), was compromised, the attacker will know the secrecy shared between them. He then could impersonate B (or A) to talk with A (or B). This is exactly what the KCI attack means. Chou *et al.* [14] had demonstrated such a KCI attack. In 2005, Cao *et al.* [9] proposed a Weil pairing ID-based NIDA protocol. In their protocol, there exists a TA (Trust Agent) whose private key is $s \in Z_q$ and public key is $P_{pub} = sP$. TA computes Alice's public / private key pair as $Q_A = H_3(ID_A)$ / $S_A = sQ_A$ and computes Bob's public / private key pair as $Q_B = H_3(ID_B)$ / $S_B = sQ_B$. When Alice wants to send a message m and its authenticator to Bob, she computes $Y = \hat{e}(tP_{pub}+S_A, tP+Q_B)$, $k = H(Y, ID_A)$, and $auth = H(k||m)$, where t is a timestamp, then she sends (ID_A, t, $auth$, m) to Bob. After receiving (ID_A, t, $auth$, m), 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 the description in [5], we can see this scheme suffers from the KCI attack. Because if an adversary E compromised Bob's private key S_B, he can impersonate Alice to send a message to Bob 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, and sending (ID_A, t', $auth'$, m') to Bob. As a result, E can successfully fool Bob to accept his message.

For patching the vulnerability of SKCI in signature-based schemes [6, 7, 8], in 2006, Lee *et al.* [10] proposed a scheme using ElGamal signature with the sender's signature s sent in a hidden way. In addition, due to Alice and Bob had pre-shared a default long-term secrecy $(Y_A)^{x_B} = (Y_B)^{x_A} = g^{x_A x_B}$(mod p), we therefore classify their scheme as shared-secrecy based. In their scheme, when Alice wants to send a message m and its authenticator to Bob, she randomly chooses t and computes $r = g^t$ (mod p), $s = H(m)\, x_A + tr$ (mod q), $k = (Y_B)^S$ (mod p), and $auth = H(k||m)$, and then sends (m, r, $auth$) to Bob. Although, Alice does not send her signature s, Bob can extract the session key by computing $k' = (Y_A^{H(m)} r^r)^{x_B}$ (=k) and then verify whether $auth = H(k'||m)$ holds. Since Alice and Bob had pre-shared a default long-term secrecy, according to [5], it suffers from the KCI attack. Because if an adversary compromised Bob's long-term private key x_B, he can successfully impersonate Alice to communicate with Bob by randomly choosing r', computing $k' = (Y_A^{H(m')}(r')^{r'})^{x_B}$(mod p) and $auth' = H(k'||m')$, and sending (m', r', $auth'$) to Bob. Bob would accept this forged message (m', r', $auth'$) unconsciously. In 2009, Wang *et al.*[24] proposed a NIDA scheme based on designated verifier proofs. They claimed that their scheme is deniable and unforgeable against a polynomial-time adversary. However, this study found if an adversary E eavesdrops and obtains message M and authenticator Authen = (w, g^r, c, s) in the simulation phase, E can randomly pick (α, β, s) and compute $c' = g^{\alpha}$, $A = g^s(y_{1p})^{-\beta}$, $B = h^s(y_{2p})^{-\beta}$, and c = H(M, c′, A, B). He then computes $w = \beta$-c, and $g^r = g^{\alpha - w} / y_{1v}$. Thus, E can successfully simulate the transcript authen=(w, g^r, c, s) without the real value of r. It is obvious that this forged transcript can pass the verification by the designated verifier.

3. The proposed scheme

Our design bases on the Fiat-Shamir heuristic because it can produce a non-interactive proof (signature) on sender's message. The unforgeability of the proof can prevent the receiver's simulator from simulating. Therefore, for designing a non-interactive deniable authentication protocol, we modify the Fiat-Shamir heuristic to make the simulator possess the simulating ability. The main difference of the modified scheme from the original one is that we replace the one-way hash function with ECC-based ElGmal encryption (ELG_Enc) to produce the random-looking *challenge*. Thus, when simulating, the simulator can invert any pre-chosen random *challenge* by ElGmal decryption as long as it knows the decryption key. Besides, for attaining a better efficiency, we adopt elliptic curve cryptosystem into our scheme. Next followings are the details of our scheme which is also illustrated in Fig. 1.

Let operator · denotes a point multiplication, for example, $a{\cdot}B = aB$, where $a \in Z_q$, and $B \in G_1$, and operator $*$ denotes a modular multiplication. 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 (whose public/private key pair is $Y_A/-x_A$) wants to authenticate message M to Bob (whose public/private key pair is $Y_B/-x_B$), they together do the following steps. (Here, plaintext M is a point $(m_1, m_2) \in G_1$, m_1 denotes the x-coordinate, and m_2 denotes the y-coordinate of M on the elliptic curve.)

Alice's part

(1) Randomly chooses $k \in_R Z_q$ and computes the *commitment* $T = kP$.

(2) Generates a random-looking *challenge*, *CH*, by applying ECC-based ElGmal encryption to N, where $N = M+R+T = (n_1, n_2)$ and R is a random element of G_1. For encrypting (n_1, n_2), she does as follows:

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

(b) computes $V = rP$,

$W = rY_B = (w_1, w_2)$,

$c_1 = w_1 * n_1 \pmod q$,

$c_2 = w_2 * n_2 \pmod q$,

$C = (c_1, c_2)$, and

$CH = \text{ELG\_Enc(M+R+T)} = (V, C)$.

(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.

Bob's part

After receiving (M, T, CH, rsp, h) from Alice, Bob does the following.

(1) Verifies whether or not

$T = rsp \cdot P + H_2(CH) \cdot Y_A$ holds. ...... (E1)

If E1 does not hold, Bob rejects the received message.

(2) Decrypts CH (= $(V, C) = (V, (c_1, c_2))$) by using his private key, $-x_B$, obtaining N'. That is, he computes

$(w_1', w_2') = -x_B \cdot V$, ... (E2-1)

$n_1' = c_1 * (w_1')^{-1} \bmod q$,

$n_2' = c_2 * (w_2')^{-1} \bmod q$, and

$N' = (n_1', n_2')$. ...... (E2-2)

(3) Computes $R' = N' - M - T$ and verifies the following equation

$h = H_1(R')$. ...... (E3)

If it holds, Bob accepts; otherwise, aborts.

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

. randomly chooses
$k \in Z_q$ and $R \in G_1$.
.computes $T = k \cdot P$,
$CH = ELG\_Enc(M + R + T)$,
$rsp = k + x_A * H_2(CH)\ (\bmod q)$, 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'.
. extracts $R' = N' - M - T$.
. 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 the property of deniability in NIDA protocol. We claim that a NIDA protocol is deniable if and only if it has the property of perfect zero-knowledge**.** After that, we inspect the deniability for our protocol by using this claim.

4.1 Deniability for a NIDA protocol

Recalling the literatures [1-3], we see that “simulatability” of the receiver’s view in a protocol implies “deniability” of the sender. However, we think this definition on deniability is not enough. Consider the simulation in signature-based NIDA schemes [6], [7], and [8]. The *simulator* only can reuse the signatures that have ever appeared in the real transcripts to compose the simulated transcripts. Because of the inheritent characteristic of signature, we think these schemes don’t possess perfect simulatability and hence are not deniable. That is, the sender can not deny his participation in the protocol. Accordingly, we use the concept of perfect zero-knowledge [17] to inspect whether or not a NIDA protocol has perfect simulatability. Perfect zero-knowledge indicates both the transcript sets produced by the simulator and the sender are equal and their corresponding probability distributions are the same. For clarity, we use perfect zero-knowledge to rephrase the deniability property for a NIDA protocol as follows.

Let λ be a NIDA protocol in which a sender S can send a message m with its *proof* to a receiver R and all messages transferred between S and R in a round comprise 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 $\mathcal{V}_R$ alone 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 fully deniable if and only if it has the property of perfect zero-knowledge. That is, $\mathcal{V}_R = \mathcal{V}_{SIM}$ and for any $\mathcal{T}_R \in \mathcal{V}_R$, there exists a $\mathcal{T}_{SIM} \in \mathcal{V}_{SIM}$ such that $\mathcal{T}_R = \mathcal{T}_{SIM}$ and $\Pr[\mathcal{T}_R] = \Pr[\mathcal{T}_{SIM}]$. We prove the claim as follows.

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

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

4.2 The deniability of our protocol

In this section, we will use Claim 1 to inspect the deniability of our protocol. Before that, we first prove 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, in the following, 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 produce a transcript, (T, CH, rsp, h) for message M. For this transcript, we can construct an efficient simulator *SIM* to forge this transcript. On input the public parameters (q, G_1, P, H_1, H_2), 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 $rsp' \in_R Z_q$, and V', $C' \in_R G_1$.

Step 2. Sets $CH' = (V', C')$ and computes $T' = rsp' \cdot P + H_2(CH') \cdot Y_A$.

Step 3. Lets $C' = (c_1', c_2')$. Computes $W' = -x_B \cdot V' = (w_1', w_2')$ and $N' = (c_1' * (w_1')^{-1}, c_2' * (w_2')^{-1}) = (n_1', n_2')$.

Step 4. Computes $R' = N' - M - T'$ and $h' = H_1(R')$.

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

It is obvious that this simulated forged transcript (T', CH', rsp', h') for M is valid and *SIM* can run 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 probability can be determined by the randomly chosen elements $rsp' \in_R Z_q$, and V', $C' \in_R G_1$ since we can see that

(i) $|rsp'| = q$,

(ii) CH' is formed by V' and C' ($|CH'|$ hence is q^2),

(iii) T' is computed from rsp' and CH' (When rsp' and CH' are determined, T' is determined as well.),

(iv) h' is computed from $R' = N' - M - T'$ and N' is determined by C' and V' as described in Step 3 of (I) (When rsp', C', and V', are determined, CH', T', N', and hence R' are determined as well. Thus, h' is also determined.).

Consequently, the cardinality of space $\mathcal{V}_{SIM}$ for M is

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

Hence, the 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, the 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.

Because from (II), we know $|\mathcal{V}_R| = |\mathcal{V}_{SIM}|$, and the probability distributions of $\mathcal{V}_R$ and $\mathcal{V}_{SIM}$ are the same. Hence, to show the property of perfect zero-knowledge for our protocol, we only need to prove that for any given $\mathcal{T}_R \in \mathcal{V}_R$, we can find a $\mathcal{T}_{SIM} \in \mathcal{V}_{SIM}$ such that $\mathcal{T}_R = \mathcal{T}_{SIM}$. When given M's $\mathcal{T}_R = (T, CH\ (=(V, C)), rsp, h) \in \mathcal{V}_R$ and trying to find its $\mathcal{T}_{SIM} = (T', CH'\ (=(V', C')), rsp', h') \in \mathcal{V}_{SIM}$ such that $\mathcal{T}_R = \mathcal{T}_{SIM}$, we can do as follows.

(i) Since V', C' and rsp' can be arbitrarily chosen by SIM and $|\mathcal{V}_R|$=$|\mathcal{V}_{SIM}|$, there must exist a transcript in $\mathcal{V}_{SIM}$ satisfying V'=V, C'=C and rsp'=rsp.

(ii) under determined V' and C', both the values of T' $(=rsp' \cdot P + H_2(CH') \cdot Y_A)$ and N' can be uniquely determined as well.

(iii) under determined V', C', N' and T', the value of R' $(=N' - M - T')$ in the transcript can be uniquely determined and equal to R.

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

5. Security analyses and comparisons

In this section, we will examine our protocol with some needed properties for a NIDA protocol. By using Theorem 1 through Theorem 4, we will show that our scheme possesses the properties of correctness, unforgeability, authenticability, and SKCI resistance. Theorem 5 indicates that the deniability property of a NIDA protocol conflicts with KCI resistance. Finally, two comparison tables, Table 1 and Table 2, are made. Table 1 compares the three properties: SCKI resistance, KCI resistance, and deniability, among our scheme and protocols [6-10, 25]. Table 2 makes comparisons in both aspects of computation and communication cost between our scheme and protocols [10, 25].

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, equation E3, $h = H_1(R')$, will hold as well. Because the following three deductions hold.

(1) $(w_1', w_2') = -x_B \cdot V = -x_B \cdot (rP) = r \cdot (-x_B P) = rY_B = (w_1, w_2)$,

(2) $ELG\_Dec(CH) = N'$
$= (c_1 * (w_1')^{-1}, c_2 * (w_2')^{-1})$

$= (c_1*(w_1)^{-1}, c_2*(w_2)^{-1})$

$= (n_1, n_2)$

$= N$, and

(3) $R' = N' - M - T = N - M - T = R$.

Theorem 2. (Unforgeablity) An adversary E could produce a valid transcript which can be verified by Bob only with a negligible probability.

Proof: Although the non-interactive proof for a message generated by Fiat-Shamir heuristic can hardly be forged by anyone who hasn't the knowledge of sender's private key, our modified Fiat-Shamir heuristic leaves a trapdoor for the receiver to produce (forge) a valid one. In our scheme, without the knowledge of sender's private key the only way for adversary E to generate a forged transcript for message M's proof is to simulate the receiver. However, without the knowledge of receiver's private key $-x_B$, E can not decrypt the random *challenge*, CH', to make a valid pair (R', T') such that $ELG\_Enc(M+R'+T') = CH'$. Therefore, we conclude that the probability E could produce a valid transcript equals to break the ElGamal cryptosystem. This probability is negligible.

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

Proof: Since 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, he uses his secret key to decrypt *CH,* obtaining N' as E2 illustrates. Using N', he can correctly compute $R'=N'-M-T$. Hence, he can verify equation E3 successfully. It means the authenticity of both the identity of Alice and the transmitted message M can be satisfied. This completes the proof.

Theorem 4. The proposed scheme can resist SKCI attack.

Proof: Since our scheme doesn't require that the two communicating parties compute a session key to produce a MAC-based authenticator as the proof of a message (as does in the previous related studies). Therefore, our scheme is free from SKCI attack.

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

Proof: For there are only one message flow in a NIDA protocol, the only one 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 a NIA protocol can resist KCI attack, then it does not have the deniability property. Assume *E* knows Bob's private key but can not impersonate Alice to communicate with Bob, it implies that some component of a real transcript produced by Alice can not be forged by *E*. That means, even with Bob's private key, the unforgeable component of the real transcript can not be efficiently produced by a simulator. Therefore, the protocol does not have the deniability property. We prove the theorem.

After presenting the above five theorems and their proofs, in the following, we make a comparison among related schemes and ours by using three dimensions: SKCI resistance, KCI resistance, and deniability in Table 1. And then, make an efficiency comparison between schemes [10, 25] and ours in Table 2.

Table 1: some properties comparisons among NIDA protocols and ours

Scheme	Approach	SKCI resistance	KCI resistance	deniability
[6]	ElGamal signature-based	No	Yes	No
[7]	Weil paring signature-based	No	Yes	No
[8]	QR signature-based	No	Yes	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
[25]	RSA-based(based on trapdoor commitment)	Yes	No	Yes
Ours	ECC-based	Yes	No	Yes

Table 2: a performance comparison between scheme[10, 25] 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
[25]	510 MM (2 EXP + 2H)	510 MM (2 EXP + 2H)	1020 MM	1344 bits
Ours	58 MM (2 ECC-mul + 2H)	87 MM (3 ECC-Mul + 2 H)	145 MM	800 bits

MM: 1024-bit modular multiplication, **EXP**: $g^k \bmod p$, where $|q|$ is 160 bits and $|p|$ 1024 bits,
ECC-mul: ECC point multiplication, **H**: hash, 1**EXP**≒255MM, 1**ECC-mul**≒29MM

From Table 1, we see that our scheme and [10, 25] are competitive in required NIDA properties. However, schemes [10, 25] are designed from ElGamal signature scheme and RSA respectively, while ours is from modified Fiat-Shamir heuristic. This provides an alternative approach in designing a NIDA protocol. Besides, from Table 2, we can see that our scheme is more efficient than schemes [10, 25] in both computation and communication cost. This is because for efficiency consideration, we implement our scheme by using elliptic curve cryptography. The comparison outcome results from [21]. It 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. Hence, for the same security level, our scheme requires only 145 MM in computation cost and 800 bits for communication size; while scheme [10] needs 1275 MM in computation cost and 1184 bits for communication size, and scheme [25] needs 1020 MM in computation and 1344 bits for communication.

6. Conclusion

Many non-interactive deniable authentication protocols have been proposed. Among them, the signature-based NIDA schemes [6, 7, 8] obviously can not achieve deniability due to the unsimulatability (unforgeability) of the signature. The shared-secrecy based NIDA, schemes [9, 10, 25] although can achieve deniability; however, suffering from either SKCI attack or KCI attack. For avoiding the drawbacks in these schemes, we propose a novel ECC-based NIDA protocol by modifying the Fiat-Shamir heuristic to achieve full deniability and attain better efficiency. In addition, we are the first attempt in using perfect zero-knowledge to prove that a NIDA protocol is deniable if and only if it is perfect zero-knowledge. According to this claim, we show our protocol is deniable. Moreover, we also prove that our scheme has the properties of unforgeability, authenticability, and SKCI resistance required in a NIDA protocol. Nevertheless, it still suffers from KCI attack. Accordingly, we further prove that for a NIDA protocol, the properties of both deniability and KCI resistance conflict.

Reference

[1] C. Dwork, M. Naor, A. Sahai, “Concurrent zero-knowledge,” *Proceedings of 30th ACM STOC’98*, 1998, pp. 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 authentication," *Applied Cryptology and Network Security (ACNS'04)*, *LNCS 3089*, pp.248-262.

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

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

[8] R. Lu, Z. Cao, "Non-interactive deniable authentication protocol based on factoring," *Computer Standards & Interfaces 27 (4)*, 2005, pp.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*, pp 400–425.

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

[13] 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*, pp. 30-45, 1997.

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

[15] 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*, pp.186-194.

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

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

pp.387-409.

[18] R. Gennaro, "Using non-interactive proofs to achieve independence efficiently and securely," *Massachusetts Institute of Technology Technical Report: TM-515*, http://portal.acm.org/citation.cfm?id=889637, 1994.

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

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

[21] 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)*, pp.3113-3123, September 2008.

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

[23] 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)*, pp.2534-2540, June 2008.

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

[25] Taek-Young Youn, Changhoon Lee and Young-Ho Park, "An efficient non-interactive deniable authentication scheme based on trapdoor commitment schemes," *Computer Communications*, In Press, Corrected Proof, March 2010.

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