Cryptanalysis on Improved Chou et al.'s ID-Based Deniable Authentication Protocol

Meng-Hui Lim
Department of Ubiquitous IT,
Graduate School of Design and IT,
Dongseo University,
Busan, 617-716, Korea
menghui.lim@gmail.com

Sanggon Lee, Hoonjae Lee
Department of Information and Communication,
Dongseo University,
Busan 617-716, Korea
{nok60, hjlee}@dongseo.ac.kr

Abstract

A deniable authentication protocol enables the protocol participants to authenticate their respective peers, while able to deny their participation after the protocol execution. This protocol can be extremely useful in some practical applications such as online negotiation, online shopping and electronic voting. Recently, we have improved a deniable authentication scheme proposed by Chou et al. due to its vulnerability to the key compromise impersonation attack in our previous report. However, we have later discovered that our previous enhanced protocol is vulnerable to the insider KCI attack and key replicating attack. In this paper, we will again secure this protocol against these attacks and demonstrate its heuristic security analysis.

1. Introduction

Privacy of communication has always been a major concern in various personal and business communications. This has in fact motivated the research and development of *Deniable Authentication* in cryptography field for centuries. With proper deniable authentication, the legal parties are able to authenticate their peers via exchanging messages over an insecure communication channel, and at the same time, the message receiver would not be able to convince a third party (may or may not be the adversary) on the identity of the sender even if the receiver reveal his own long-lived private key to the third party. Hence, the common association of digital signature with message authentication in the public key scenario is often undesirable since only at most weak deniability [16] (the receiver can prove to have spoken with the sender but not the content of what the sender authenticated) can be guaranteed in this case.

Over the years, numerous deniable authentication protocols have been proposed. However, due to the rush in exploiting new ideas which results in careless design, many of them have been proven to be vulnerable to a variety of cryptanalytic attacks [4, 8, 9, 15, 21]. The notion of deniability in public key authentication is pioneered by Dwork et al. [11], which is based on the concurrent zero knowledge proof. However, this scheme requires a timing constraint and the proof of knowledge is time-consuming. In 2003, Boyd, Mao and Paterson [5] have proposed 2 deniable authenticated key establishment schemes for Internet protocols based on elliptic curve cryptography. These schemes are conjectured to be able to solute the complexity of computation and appear to be more efficient than others. However, Chou et al. [9] have proved their security flaw by demonstrating a valid key compromise impersonation (KCI) attack on their scheme in 2005. Another notable deniable authentication scheme which was proposed by Fan et al. [12] in 2002 is based on Diffie-Hellman key distribution protocol. Unfortunately, Yoon et al. [21] have pointed out the susceptibility of Fan et al.'s scheme to the intruder masquerading attack in 2005 and subsequently, they have proposed their enhanced deniable authentication protocol. In addition, in 2005, Cao et al. [7] have proposed an efficient ID-based deniable authentication protocol which enables a dynamic shared secret to be derived as a session key. Unfortunately, in 2006, both Yoon et al.'s enhanced scheme and Cao et al.'s scheme were proven to be impractical and susceptible to KCI attack respectively by Chou et al. [8]. Moreover, Chou et al. have proposed another new deniable authentication protocol and they have conjectured their proposed protocol to possess strong deniability as well as authenticity with great resistance against KCI attack. Recently, we have proven them wrong by launching a valid KCI attack on their scheme and subsequently proposed our improvement scheme in [15]. However, we have spotted a few security flaws in our previous improve-

ment [15] that might results in some further undesirable cryptanalytic attacks. Hence, we aim to address them thoroughly by proposing our latest improvements in this paper.

The structure of this paper is organized as follows. In the next section, we will illustrate some basic properties of bilinear pairings and underlying assumptions. In Section 3, we will review Chou et al.'s ID-based deniable authentication protocol and our previous improvement. In Section 4, we will demonstrate the security flaws lying in our previous improvement scheme. In Section 5, we will illustrate our latest improvement scheme and its associated security analysis. Last but not least, we will conclude this paper in Section 6.

2. Preliminaries

In this section, we introduce the basic properties of bilinear pairings, the Bilinear Diffie-Hellman Problem and the Discrete Logarithm Problem. Let $\mathbf{G}_1$ be an additive group of a large prime order, q and $\mathbf{G}_2$ be a multiplicative group of the same order, q. Let $P, Q \in \mathbf{G}_1$ and $\hat{e} : \mathbf{G}_1 \times \mathbf{G}_1 \longrightarrow \mathbf{G}_2$ be a bilinear pairing with the following properties:

- **Bilinearity**: $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab} = \hat{e}(abP, Q)$ for any $a, b \in Z_q^*$.
- **Non-degeneracy**: $\hat{e}(P, Q) \neq 1$.
- **Computability**: There exists an efficient algorithm to compute $\hat{e}(P, Q)$.

A bilinear map which satisfies all three properties above is considered as *admissible bilinear*. It is noted that the Weil and Tate pairings associated with the supersingular elliptic curves or abelian varieties, can be modified to create such bilinear maps. Now, we describe some cryptographic problems:

Bilinear Diffie-Hellman Problem (BDHP). Let $\mathbf{G}_1$, $\mathbf{G}_2$, P and $\hat{e}$ be as above with the order q being prime. Given $\langle P, aP, bP, cP \rangle$ with $a, b, c \in Z_q^*$, compute $\hat{e}(P, P)^{abc} \in \mathbf{G}_2$. An algorithm α is deemed to have an advantage ϵ in solving the BDHP in $\langle \mathbf{G}_1, \mathbf{G}_2, \hat{e} \rangle$ based on the random choices of a, b, c in Z_q^* and the internal random operation of α if

$$Pr[\alpha(\langle P, aP, bP, cP \rangle) = \hat{e}(P, P)^{abc}] \geq \epsilon.$$

Discrete Logarithm Problem (DLP). Given two groups of elements P and Q, such that $Q = nP$. Find the integer n whenever such an integer exists.

Throughout this paper, we assume that BDHP is intractable, which means that there is no polynomial time algorithm to solve BDHP and DLP with non-negligible probability.

3. Review of Chou et al.'s ID-based Deniable Authentication Protocol and its variant

In this section, we look at a specific ID-based deniable authentication protocol proposed by Chou, Chen and Huang [8] and subsequently our previous improvement [15].

3.1. Chou et al.'s Scheme

As usual, we specify the two communication parties in a protocol run as Alice and Bob. Now suppose that they wish to communicate with each other. To achieve this, they perform an instance of the protocol run. Initially, the Private Key Generator (PKG) picks a master private key $s \in Z_q^*$ and sets the master public key

$$P_{pub} = sP. \tag{1}$$

The PKG then publishes $\{\mathbf{G}_1, \mathbf{G}_2, \hat{e}, P, H_1, H_2, H_3\}$, where $H_1 : \{0,1\}^* \longrightarrow \mathbf{G}_1$, $H_2 : \mathbf{G_2} \longrightarrow \{0,1\}^m$ (m is a security parameter) and $H_3 : \{0,1\}^* \longrightarrow \{0,1\}^*$ are one-way collision-free cryptographic hash functions. For a given string $ID \in \{0,1\}^*$, the PKG computes the public key,

$$Q_{ID} = H_1(ID) \tag{2}$$

and the private key,

$$S_{ID} = sQ_{ID}. \tag{3}$$

Hence, Alice and Bob's public/private key pairs are denoted as Q_A/S_A and Q_B/S_B respectively. We describe Chou et al.'s protocol as follows:

Step 1. Alice chooses a random number, $r_A \in Z_q^*$, computes

$$u = r_A Q_A \tag{4}$$

and then sends (ID_A, u) to Bob.

Step 2. Upon receipt of (ID_A, u), Bob chooses a random number, $r_B \in Z_q^*$ and calculates

$$h_B = H_2(\hat{e}(u, S_B)), \tag{5}$$

$$f = h_B \oplus r_B, \tag{6}$$

and sends (ID_B, f) to Alice.

Step 3. After receiving (ID_B, f), Alice computes

$$h_A = H_2(\hat{e}(r_A S_A, Q_B)), \tag{7}$$

$$r_B = h_A \oplus f, \tag{8}$$

$$X_A = H_2(x_A), \text{where } x_A = \hat{e}(r_B Q_B, P_{pub}), \tag{9}$$

$$Y_A = H_2(y_A), \text{where } y_A = \hat{e}(r_B S_A, P), \quad (10)$$

and subsequently computes the session key,

$$K_A = \hat{e}(S_A, Q_B)^{X_A Y_A} \quad (11)$$

Suppose that m_A is the message which Alice's would like to send together with her ID. She computes

$$g_A = H_3(ID_B, m_A, x_A, y_A, K_A) \quad (12)$$

and sends (g_A, m_A) to Bob.

Step 4. After receiving (g_A, m_A), Bob calculates

$$X_B = H_2(x_B), \text{where } x_B = \hat{e}(r_B S_B, P), \quad (13)$$

$$Y_B = H_2(y_B), \text{where } y_B = \hat{e}(r_B Q_A, P_{pub}). \quad (14)$$

Then, he computes the session key

$$K_B = \hat{e}(Q_A, S_B)^{X_B Y_B}. \quad (15)$$

Finally, Bob computes

$$g_B = H_3(ID_B, m_A, x_B, y_B, K_B) \quad (16)$$

and compares whether

$$g_A \stackrel{?}{=} g_B. \quad (17)$$

If it does (does not), Bob accepts (rejects) the session key.

The authors claimed their scheme to be deniable, authenticated and resilient against the KCI attack as shown in their heuristic security analysis [8]. However, we had proved the opposite in [15].

3.2. The Key Compromise Impersonation Attack and Our Previous Improvement

The KCI attack can be informally defined as a kind of known-key attack, which can be carried out by an adversary after compromising a protocol entity's private key. We often refer such entity as corrupted. An entity can be corrupted easily in many real-world scenarios. For example, a malicious party may hack into or "hijack" the entity's computer or machine in order to learn the entity's private key. Even worse, the corrupted party may not even aware of this intrusion and this significantly benefits the adversary in utilizing the corrupted party's private key in his bad deeds, at least until the corrupted entity has detected this compromisation. However, instead of impersonating the corrupted party directly, the general adversarial goal in the KCI attack is to masquerade as any other legitimate party and engage with the corrupted party in a session by means of establishing a valid session key with him. With this, the adversary may appear to be an "authentic" bank officer to the corrupted party and subsequently capture some valuable information from him (e.g. credit card number or bank account password). If the adversary is able to do so, we can then intuitively speculate the protocol to be susceptible to the KCI attack.

As shown in our previous report [15], the security of Chou et al.'s scheme can be breached by imposing such KCI attack on their scheme. In the attack, a malicious adversary is capable of impersonating the sender (Alice) and completing a specific protocol run with the receiver (Bob) if the receiver's private key has been exposed by some means. Due to the bilinear property of pairing, it should be easily noted that by learning the value of S_B in prior, h_A in Step 3. can be calculated by using Eq. (5) and the subsequent parameters $(r_B, X_A, Y_A(\text{using Eq. } (14)), K_A(\text{using Eq. } (15)) \text{ and } g_A)$ can be derived accordingly.

In order to defeat this attack, we suggested an improvement scheme in [15] with the intention to prevent the adversary's ability in computing h_B if S_B happens to be compromised. Now, we describe the improved protocol as follows:

Step 1. Similar to the original protocol, Alice chooses a random number, $r_A \in Z_q^*$, computes u from Eq. (4) and sends (ID_A, u) to Bob.

Step 2. After receiving (ID_A, u), Bob chooses a random number, $r_B \in Z_q^* *$ and computes

$$v = r_B Q_B, \quad (18)$$

$$h_B = H_2(\hat{e}(u, r_B S_B)), \quad (19)$$

and f from Eq. (6), and sends (ID_B, f, v) to Alice.

Step 3. Upon receipt of (ID_B, f, v), Alice computes

$$h_A = H_2(\hat{e}(r_A S_A, v)), \quad (20)$$

and r_B from Eq. (8). Then, she calculates X_A, Y_A, and the session key K_A from Eqs. (9), (10) and (11) respectively. Eventually, she computes g_A from Eq. (12) and sends (g_A, m_A) to Bob.

Step 4. After receiving (g_A, m_A), Bob calculates X_B, Y_B and the session key K_B from Eqs. (13), (14) and (15) respectively. At last, he computes g_B from Eq. (16) and checks whether Eq. (17) holds. If it does (does not), Bob accepts (rejects) the sesson key.

With this improvement, our previous work seems to be secure as it is immune to the KCI attack now. However, we discover that this improvement is strongly deficient as it in turns results in another two flaws which we will describe in the next section.

4. Flaws in the Previous Improved Scheme

4.1. Insider KCI Attack

The adversary in the KCI attack that we discussed so far is preceived as an outsider. What if the KCI attacker is an insider? It should be noted that malicious insiders may exist as the number of protocol principals grows. In order to secure a protocol against the KCI attack, we stress that the existence of such malicious insiders should be regarded as important as outsider adversaries in the KCI analysis. For instance, the presence of a malicious bank user impersonating the bank officer to cheat another corrupted bank user, after obtaining the corrupted user's secret key.

How would our previous work be affected if the adversary is an insider? To answer this, we provide an analysis on our previous improvement scheme in detail by considering the KCI adversary to be an insider. As usual, we designate the two communicating parties as Alice and Bob, and the insider adversary as Eve (with his public/private key pair as Q_E/S_E). Assume that Bob's private key S_B has been compromised in prior. With the intention to impersonate Alice, Eve is now more powerful since she can exploit the advantage of possessing both his private key and Bob's private key in mounting the attack. To cheat Bob, Eve initially initiates an instance of the protocol run where we assume that Alice does not know anything about this. The attack algorithm can be described as follows:

Step 1. Impersonating Alice, Eve chooses a random number, $r_E \in Z_q^*$, computes

$$u' = r_E Q_E \tag{21}$$

and sends (ID_A, u') to Bob. Since Bob does not know r_E due to intractability of DLP, he would not be able to distinguish whether u is really originated from Alice.

Step 2. After receiving (ID_A, u'), Bob follows the protocol procedures in Section 3.2 as usual. He chooses a random number, $r_B \in Z_q^*$* and computes v from Eq. (18), h_B (by using u' instead of u) from Eq. (19) and f from Eq. (6), and sends (ID_B, f, v) to Alice.

Step 3. Before reaching Alice, Eve intercepts (ID_B, f, v), and computes

$$h'_A = H_2(\hat{e}(r_E S_E, v)), \tag{22}$$

and subsequently extracts r_B from Eq. (8). Note that h'_A and h_B are consistent:

$$\begin{aligned} h'_A &= H_2(\hat{e}(r_E S_E, v)) \\ &= H_2(\hat{e}(u', r_B S_B)) \\ &= h_B. \end{aligned} \tag{23}$$

Then, she calculates X_A, Y_A, and the session key K_A from Eqs. (9), (14) and (15) respectively. At last, she computes g_A from Eq. (12) and sends (g_A, m_A) to Bob.

Step 4. After receiving (g_A, m_A), Bob calculates X_B, Y_B and the session key K_B from Eqs. (13), (14) and (15) respectively. At last, he computes g_B from Eq. (16) and he would find that Eq. (17) eventually holds.

As a result, we have shown that the security of the previous improved scheme can be penetrated by the insider KCI attack. Since Bob would accept the session key at the end by falsely authenticating the sender's identity, our previous improment scheme is therefore insecure.

4.2. Key Replicating Attack

A further cryptanalytic attack can be carried out on our previous improvement scheme, namely the key replicating attack [4, 13], in which it deals closely with the oracle queries described in Bellare and Rogaway's formal model [1, 2]. This attack, if successfully carried out, would enable the adversary to succeed in forcing the establishment of a session, $\mathcal{S}$ (other than the Test session or its matching session) to possess the same session key as the Test session. Since the Test session and $\mathcal{S}$ are non-matching, the adversary may issue a Reveal query to the oracle associated with $\mathcal{S}$ and he can then distinguish whether the Test session key is real or random.

Now let us scrutinize our previous improvement scheme with such a key replicating attack. Similarly, we assume Alice and Bob are the communicating parties and Eve remains to be the active adversary, who has full control over the unauthenticated communication channel.

Step 1. Initially, Alice chooses a random number, $r_A \in Z_q^*$, computes u from Eq. (4) and sends (ID_A, u) to Bob.

Step 2. Before reaching Bob, Eve intercepts the message. She chooses a random number $x \in Z_q^*$ and fabricates

$$u' = x \cdot u = x \cdot (r_A Q_A). \tag{24}$$

After that, Eve sends (ID_A, u') to Bob on behalf of Alice. Upon receiving the altered message, without suspicion, Bob chooses a random number, $r_B \in Z_q^*$* and computes v from Eq. (18), h_B (by using u' instead of u) from Eq. (19) and f from Eq. (6), and sends (ID_B, f, v) to Alice.

Step 3. Before reaching Alice, Eve intercepts (ID_B, f, v), fabricates

$$v' = x \cdot v = x \cdot (r_B Q_B), \tag{25}$$

and sends (ID_B, f, v') to Alice. Alice computes h_A by using v' and extract r_B from f subsequently without being aware of the modified value of v. Then, she calculates X_A, Y_A, and the session key K_A from Eqs. (9), (10) and (11) respectively. Eventually, she computes g_A from Eq. (12) and sends (g_A, m_A) to Bob.

Step 4. After receiving (g_A, m_A), Bob calculates X_B, Y_B and the session key K_B from Eqs. (13), (14) and (15) respectively. At last, he computes g_B from Eq. (16) and he would find that Eq. (17) eventually holds.

Notice that in this scenario, although both Alice and Bob have non-matching conversation at the end of the protocol execution, they have accepted the same session key, that is

$$K_A = \hat{e}(S_A, Q_B)^{X_A Y_A} = \hat{e}(Q_A, S_B)^{X_B Y_B} = K_B.$$

Hence, after the protocol execution, Eve is allowed to expose a fresh session key by revealing either Alive or Bob in his attack and he would be able to guess correctly on the genuineness of the Test-session key for the other non-matching session (either Bob or Alice's).

5. Enhancement and Security Analysis

As discussed in the previous session, our previous improvement scheme contains flaws which would pose a serious and subtle threat to the protocol participants, despite possessing the attractive deniability property. In order to address the defects, we generally base our proposed solutions on two approaches:

1. Restrict the parameters $u(= r_A \cdot Q_A)$ and $v(= r_B \cdot Q_B)$ in this protocol to be computed by using the intended public keys (to defeat the insider KCI attack), and any alteration to such parameters should be detected through subsequent verfication by the respective communicating partner.

2. Use a key derivation function to derive the session key (K_A and K_B). Other than the shared secret as shown in Eqs. (11) as well as (15), the inclusion of the unique session identifiers into the key derivation function should be treated equally essential especially in preventing a variety of undesirable cryptographic attacks, such as key replicating attack and triangle attack [6].

With this, we propose our enhancement scheme as follows:

Step 1. Initially, Alice picks a random number, $r_A \in Z_q^*$, computes u from Eq. (4) and

$$w = r_A^{-1} P, \tag{26}$$

and then sends (ID_A, u, w) to Bob.

Step 2. Upon receiving Alice's message, Bob checks whether

$$\hat{e}(w, u) \stackrel{?}{=} \hat{e}(P, Q_A). \tag{27}$$

If it does not, Bob terminates the session. Otherwise, Bob chooses a random number $r_B \in Z_q^*$ and computes v from Eq. (18), h_B from Eq. (19) and f from Eq. (6), and sends (ID_B, f, v) to Alice.

Step 3. On receipt of Bob's message, Alice computes h_A from Eq. (20) and extract r_B from Eq. (8). Then, Alice computes $r_B Q_B$ and checks whether

$$r_B \cdot Q_B \stackrel{?}{=} v. \tag{28}$$

Alice terminates the session if the verification fails. Otherwise, she calculates X_A from Eq. (9), Y_A from Eq. (10), and the session key

$$K_A = kdf(\hat{e}(S_A, Q_B)^{X_A Y_A} \parallel u \parallel w \parallel f \parallel v). \tag{29}$$

Eventually, she computes g_A from Eq. (12) and sends (g_A, m_A) to Bob.

Step 4. After receiving (g_A, m_A), Bob calculates X_B from Eq. (13), Y_B from Eq. (14) and the session key

$$K_B = kdf(\hat{e}(Q_A, S_B)^{X_B Y_B} \parallel u \parallel w \parallel f \parallel v). \tag{30}$$

At last, he computes g_B from Eq. (16) and checks whether Eq. (17) holds. If it does (does not), Bob accepts (rejects) the sesson key.

Now, let us analyze this protocol to ensure that the existing flaws have been eliminated with our latest improvements.

Lemma 1. *Our improved protocol is absolutely immune to the KCI attack, despite the adversary is an outsider or an insider.*

Suppose that Eve is an outsider adversary who has learned Alice's private key and wishes to cheat Alice by impersonating Bob in a protocol instance. In the middle of the protocol execution, she would fail to compute h_B as she does not know r_A or S_B. On contrary, if Bob's private key is compromised and Eve would want to cheat Bob by impersonating Alice, she would fail in computing h_A since she does not know r_B or S_A. Hence, our protocol can evidently resist an outsider KCI attack. We now turn to assume that Eve is an insider adversary. Apparently, Eve could no longer mount a KCI attack described in Section 4.1 on our enhanced protocol. With our additional parameter verification processes in Eqs. (27) and (28), we mandate u (if Eve impersonates Alice) and v (if Eve impersonates Bob) to be computed by using the designated

public key. If any of these parameters is forged, the respective verification process would fail. As a result, the insider KCI attack can totally be prevented in both cases for the exposure of either Alice or Bob's private key in each scenario.

Lemma 2. *Our improved protocol is able to withstand the key replicating attack, where the protocol participants would agree on a different session key if their conversation is non-matching.*

As shown in the protocol above, the session key is derived by a key deriving function which takes in the shared secret and the transcripts as the session identifiers. If Eve carry out the key replicating attack in Section 4.2, Alice's session key $K_A = kdf(\hat{e}(S_A, Q_B)^{X_A Y_A} \parallel r_A Q_A \parallel r_A P \parallel h_B \oplus r_B \parallel x \cdot r_B Q_B)$ would be different from Bob's session key $K_B = kdf(\hat{e}(Q_A, S_B)^{X_B Y_B} \parallel x \cdot r_A Q_A \parallel r_A P \parallel h_B \oplus r_B \parallel r_B Q_B)$ at the end of the protocol execution. With this, Eve would not be able to force the establishment of non matching sessions to possess a same session key. As a result, she would end up guessing the genuineness of Test-session key on her luck.

Lemma 3. *Our improved protocol remains deniable.*

Our improved protocol enables both Alice and Bob to simulate the transcripts perfectly. As the messages are properly authenticated during the protocol execution, both parties are in fact aware of the identity of their respective partner. However, since Alice and Bob are holding the same session key at the end of the protocol execution, Bob would not be able to prove to a third party that (h, m) is originated from Alice since Alice can later deny her participation by claiming that such message can also be simulated by Bob.

6. Conclusions

In a nutshell, we have proven our previous improvement scheme to be flawed due to inadequate scrutiny of security in our previous report. Based on these deficiencies, we have further proposed several enhancements in order to defeat the vulnerabilities. Besides justifying the improvements, we have carried out a heuristic security analysis to ensure the deniability property is perserved.

References

[1] M. Bellare, P. Rogaway, Entity Authentication and Key Distribution, Advances in Cryptology-CRYPTO 1993, LNCS, vol. 773, 1993, pp. 110-125.

[2] M. Bellare and P. Rogaway, Provably Secure Session Key Distribution: The Three Party Case, 27th ACM Symposium on the Theory of Computing - ACM STOC, 1995, pp. 57-66.

[3] D. Boneh, M. Franklin, Identity-based encryption from the Weil pairing, Advances in Cryptology-Crypto 2001, LNCS, vol. 2139, 2001, pp. 213-229.

[4] C. Boyd, K.-K.R. Choo, Security of Two-Party Identity-Based Key Agreement, Proceedings of First International Conference on Crytology in Malaysia (MyCrypt 2005), LNCS, vol.3715, 2005, pp. 229-243.

[5] C. Boyd, W. Mao, K. G. Paterson, Deniable Authenticated Key Establishment for Internet Protocols, 11th International Workshop on Security Protocols, LNCS, vol. 3364, 2003, pp. 255 - 271.

[6] M. Burmester, On the Risk of Opening Distributed Keys, *Crypto'94*, LNCS, vol.839, 1994, pp. 308-317

[7] T.J. Cao, D.D. Lin, R. Xue, An Efficient ID-based Deniable Authentication Protocol from Pairings, 19th International Conference on Advanced Information Networking and Applications (AINA 2005), IEEE Computer Society, 2005, pp. 388-391.

[8] J.S. Chou, Y.L. Chen, J.C. Huang, A ID-Based Deniable Authentication Protocol on Pairings, Cryptology ePrint Archive: Report, (335)(2006).

[9] J.S. Chou, Y.L. Chen, M.D. Yang, Weaknesses of the Boyd-Mao Deniable Authenticated Key Establishment for Internet Protocols, Cryptology ePrint Archive: Report, (451)(2005).

[10] X. Deng, C.H. Lee, H. Zhu, Deniable Authentication Protocols, IEE Proc. Comput. Digit. Tech., vol. 148 (2), 2001, pp. 101-104.

[11] C. Dwork, M. Naor, A. Sahai, Concurrent Zero-Knowledge, Proc. 30th ACM STOC '98, Dallas TX, USA, 1998, pp. 409-418.

[12] L. Fan, C.X. Xu, J.H. Li, Deniable Authentication Protocol based on Diffie-Hellman algorithm, Electronics Letters 38. (4) (2002) pp. 705-706.

[13] H. Krawczyk, HMQV: A High-Performance Secure Diffie-Hellman Protocol, CRYPTO 2005, LNCS, vol. 3621, pp. 546-566.

[14] I.-E. Liao, C.-C. Lee, M.-S. Hwang, Identity-Based Deniable Authentication Protocol from Pairings, Proceedings of the 10th IASTED International Conference Internet and Multimedia Systems and Applications (IMSA2006), 2006, pp. 112-114.

[15] M.-H. Lim, S.G. Lee, Y.H. Park, H.J. Lee, An Enhanced ID-Based Deniable Authentication Protocol on Pairings, ICCSA (2) 2007, LNCS, vol. 4706, 2007, pp. 1008-1017.

[16] M.D. Raimondo, R. Gennaro, New Approaches for Deniable Authentication, Proceedings of the 12th ACM conference on Computer and Communications Security, 2005, pp. 112-121

[17] M.D. Raimondo, R. Gennaro, H. Krawczyk, Deniable Authentication and Key Exchange, Proceedings of the 13th ACM conference on Computer and Communications Security, 2006, pp. 400-409.

[18] M.A. Strangio, On the Resilience of Key Agreement Protocols to Key Compromise Impersonation, EuroPKI06, LNCS, vol. 4043, pp. 233-247.

[19] S.B. Wilson, A. Menezes, Authenticated Diffie-Hellman key agreement protocols, Proceedings of the 5th Annual Workshop on Selected Areas in Cryptography (SAC 98), LNCS, vol. 1999, pp. 339-361.

[20] A.C. Yao, F.F. Yao, Y.L. Zhao, B. Zhu, Deniable Internet Key Exchange, Cryptology ePrint Archive: Report, (191)(2007).

[21] E.J. Yoon, E. K. Ryu, K. Y. Yoo, Improvement of Fan et al.'s Deniable Authentication Protocol based on Diffie-Hellman Algorithm, Applied Mathematics and Computation, Vol. 167 (1), 2005, pp. 274-280.