A Deniable Authentication Protocol with Anonymous Sender Protection

Hwang, Shin-Jia and Ma, Juei-Che*
Department of Computer Science and Information Engineering,
TamKang University, Tamsui, Taipei Hsien, 251, Taiwan, R.O.C.
*E-mail: sjhwang@mail.tku.edu.tw, 695411974&95.tku.edu.tw**

Abstract-In *deniable authentication protocols, the senders' right cannot be protected due to the deniable property. The deniable property causes the receiver's implication because the sender cannot prove the sender's identity to the third party. To overcome this problem, Hwang and Ma first proposed their protocol with sender protection to provide evidences for the sender. To protect senders' privacy, it is better that the senders should be anonymous. Therefore, a deniable authentication protocol with anonymous sender protection is proposed. The extra bonus of our sender protection is that, in our protocol, the sent deniable messages can be converted to undeniable signatures without additional computational cost. Then the converted signatures can be directed used in real applications.*

Keywords: Deniable authentication protocols, concurrent signatures, authentication.

1. Introduction

Deniable authentication protocols [2, 3] are used to prove the sender's identity of the origin of some messages to an intended receiver. However, the intended receiver cannot convince anyone that the sender has sent the message to the intended receiver. Therefore, deniable authentication protocols must satisfy authentication and deniable properties between two entities over communicating channels. The authentication property means that only the intended receiver can find out the sender's identity of the origin of a received message. The deniable property means that the receiver cannot prove the origin's identity of the message to a third party.

Due to these two properties, deniable authentication protocols are useful in the electronic transaction [7], electronic voting system and secure negotiation [2, 3] over the Internet. There is another application for deniable authentication protocols [11]. Suppose that a customer wants to order goods from a merchant, the customer needs the merchant's quotation. To prevent the misuse of the quotation, the merchant has to make sure that the quotation is used only for the intended customer but not anyone else.

Aumann and Rabin [2, 3] first proposed their deniable authentication protocol based on the factoring problem. Aumann and Rabin's protocol needs a public trusted directory between the sender and the receiver. Dwork et al. [8] developed a notable deniable authentication protocol based on concurrent zero knowledge proof. However, Dwork et al.'s protocol has a timing constraint because the proof of the knowledge is subject to time the delay during the authenticate process.

Afterward, Deng et al. [7] proposed two deniable authentication protocols based on factoring problem and the discrete logarithm problem, respectively. Unfortunately, Deng et al.'s protocols suffer from the person-in-the-middle attack [20]. Moreover, in Deng et al.'s protocols, a trust public directory is also needed. To maintain the trust public directory is a heavy cost. To remove public trusted directories, Fan et al. [9] proposed another deniable authentication protocol based on the Diffie-Hellman key distribution protocol. Later, Chang et al. [5] points out that the sender did not authenticate the receiver in Fan et al.'s protocol, so that the sender cannot tell who the receiver actually *is*. In [5], Chang et al. use a signature to avoid the flaw of no intended receiver. But Sun et al. [16] announced that Chang et al.'s protocol cannot resist an adaptive attack, and also proposed a new session key generation scheme by using temporary private/ public key pair in the protocol.

Even though many deniable authentication protocols [2-3, 7, 9] are proposed, all those proposed protocols are interactive protocols. To improve efficiency, Shao [15] proposed the non-interactive deniable authentication protocol based on generalized EIGamal signature scheme. Shao pointed out that interactive deniable authentication protocols cannot prevent the impersonate attack. Moreover, a non-interactive protocol is more efficient than an interactive protocol. Lu et al. [12] proposed a non-interactive deniable authentication protocol by using the improved Rabin signature scheme. But

in Lu et al.'s protocol, an impersonation attack is easy to disorder the execution of the protocol. Thus, the intended receiver cannot sure that if the received message is really sent from the real sender. To solve this flaw, Lu et al. then modify [13] the original protocol a little bit by putting the sender and the intended receiver's identities into the hash function on transmission. Later, Lee et al. [11] proposed their protocol whose communication cost is less than the one of Shao's protocols.

Some interesting applications and variances are proposed by deniable authentication protocol. Shi et al. [18] proposed an identity-based deniable authentication protocol by using bilinear pairings and reduces to the hardness of bilinear Diffie-Hellman problem. Brown [4] proposed another deniable authentication protocol for the multicasting. Lu et al. [14] also address a group oriented deniable authentication protocol. In this group oriented protocol, the sender side must be a group (which means more than one identity) and there will be a threshold for generating the message to an intended receiver.

In those proposed protocols, the sender can deny that he/she sent the message to receiver because both the sender and receiver have the same ability to generate the sent messages. However, the sender has no evidence to prove that he/she is the real sender to prevent receivers' implication. Let us consider the situation that a company wants to buy some product with the lowest price. The company will ask the product's providers the quotation. In this situation, the deniable authentication protocols are used to provide authenticated but deniable quotations for the company. However, the deniability is caused by forgeability of the receiver. The company can easily forge authenticated but deniable quotations while the providers cannot prevent this circumvention. To prevent receivers' circumvention, Hwang and Ma [10] proposed their deniable authentication protocol with sender protection. Here and after, Hwang and Ma's protocol is named as DAP-SP.

In the quotation situation, the providers do not want anyone know their identities to avoid unnecessary annoyance. But, in the deniable authentication protocols, the sender's identity is not protected anymore. That is everyone may know who the sender is. To protect the sender's privacy, it is better that the sender's identity is hidden from the other ones except the receiver. Therefore, a deniable authentication protocol with anonymous sender protection is proposed.

In the following section, the concurrent signature scheme iPCS1 [19] and DAP-SP are reviewed in Section 2. Our new non-interactive deniable authentication protocol with anonymous sender protection is proposed in Section 3. The corresponding security and performance analysis is given in the same section. The performance analysis and discussions about our proposed protocol is given in Section 4. Finally, Section 5 is our conclusions.

2. Review of iPCS1 and DAP-SP

2.1 Review of iPCS1

Wang et al. [19] proposed their concurrent signature schemes, iPCS1 and iPCS2, to improve the fairness and ambiguity of Chen et al.'s concurrent signature schemes [6]. The review of iPCS1 is given here.

The scheme iPCS1 is consisted of fours algorithms SETUP, ASIGN, AVERIFY, and VERIFY, and one protocols. In the following, these algorithms are first given.

— SETUP. The input of the SETUP algorithm is a security parameter 1. On this security parameter, the SETUP algorithm first generates the public system parameters and public functions. The public system parameters are two large prime numbers p and q with ql(p-1), and an element g of order q in Z_p. The public function is a collision-free hash function whose image is Z_q*. The SETUP algorithm also generates the public key and private key for each user. The private key of the user i is a random number $x_{ie} Z_{q^*}$ and the public key of the user i is computed by Y_i= mod p.

— ASIGN. The ASIGN algorithm is used to generate an ambiguous signature on some message ms. The input of ASIGN is (Y_S, Y_R, X_S, e_R, m_S), where Xs is the private key of sender S, Ys is the public key of sender S, Y_R is the public key of receiver R, e_R is an integer generated by the keystone generation algorithm h(k), k is a randomly chosen keystone. Then the output of ASIGN(Ys, Y_R, X_s, e_R, ms) is an ambiguous signature as= (fs, es, e_R). The concrete algorithm for ASIGN is given below.

1. Select a random integer rsc Z_{q^*}.
2. Evaluate fs= $h(ms1181.SY_{ReR}$ mod p).
3. Compute es = (rs — fs)X_s^{-1} mod q.
4. Output an ambiguous signature as = (^{f}s, es, ex)•

— AVERIFY. Given an ambiguous-signature-message pair (as,Ys,Y_R, ms), the algorithm outputs accept if the equation fs= h(mslleY$s^{eS}Y_R^{eR}$ mod p) holds; otherwise, the algorithm outputs reject.

— **VERIFY.** The input of the algorithm VERIFY is the tuple (k, (a_5,Y_5,Y_R,m_5)), where k is a keystone and (as, ^{Y}s, ^{Y}R, ^{m}s) is an ambiguous-signature-message pair. Then VERIFY algorithm returns accept if AVERIFY(as, ^{Y}s, $^{Y_R.}$ ms)$^{=}$ accept and the keystone k satisfies the keystone verification equation e_R= h(k). Otherwise, VERIFY outputs reject.

These algorithms are used to construct the iPCS1 protocol. The interesting reader can refer [19] to find the detail of iPCS1. The following theorem in [19] shows that both iPCS1 and iPCS2 satisfy ambiguity, fairness, and existentially unforgeablity properties under the hardness of the discrete logarithm problem and the bilinear Diffie-Hellman problem.

Theorem 1: According to the formal definitions given in [6, 17], the protocols, iPCS1 and iPCS2, are secure perfect concurrent signature protocols, under the hardness of discrete logarithm problem and the bilinear Diffie-Hellman problem. In other words, both iPCS1 and iPCS2 have ambiguity, fairness, and existentially unforgeablity under a chosen message attack in the multi-party setting [19].

2.2 DAP-SP

Hwang and Ma's deniable authentication protocol with sender protection (DAP-SP for short) is consisted of two phases: Setup phase and deniable authentication phase. Setup phase is described first and then deniable authentication phase is given.

Setup Phase

A trusted authority (TA for short) is response to generate and publish public system parameters and functions in this phase. Then the private-public key pair of each user is also generated and certificated by TA. TA first selects two large prime numbers p and q such that q is a large prime divisor of p-1. Then TA finds an element g of order q in GF(p) and defines a collision-free hash function h whose image is Z_q*. The private key of the user i is a random number Xie Z q* and the public key of the user i is computed by Y_i= gxi mod p.

Deniable Authentication Phase

Suppose that one user S wants to be authenticated by the intended receiver R to transmit the message ms. The DAP-SP between S and R is described below.

Protocol DAP-SP

Sten 1: Sender S chooses two random numbers k and rse Z_q^4.

Step 2: S computes e_R = h(k), $_{fs}$ = ^{11}(^{11}$_{s\ 1e}$$^{SY_R eR}$ mod p), and es = (r_5 —f_5)X_5^{-1} mod q.

Step 3: S sends (es, e_R, f_5) and ms to Receiver R.

Sten 4: R performs the validation equation fs $^{h(ms}$llYSeSgfSYReR mod p) (mod q) to authenticate S and (es, e_R, fs) after receiving (es, e_R, fs) and ms. If the validation equation holds, then the intended receiver R is assured that (es, ex, fs) and ms are indeed sent by User S.

3. Our Protocol with Anonymous Sender Protection

Our deniable authentication protocol with anonymous sender protection (DAP-ASP for short) is also consisted of two phases: Setup phase and deniable authentication phase. The setup phase is the same as the setup phase in DAP-SP. Only the deniable authentication phase is given.

3.1 Deniable Authentication Phase

Suppose that the sender S wants to be authenticated by the intended receiver R to transmit the message m_5. The DAP-ASP between S and R is stated below.

P$_{s}$ **t**$_{r}$**e**$_{o}$**p**$_{t}$ **o**$_{s}$$_{l}$**e**$_{r}$**DAP-sASP**$_{1:}$ chooses three random numbers k,

x, and rse Z_q^*.

Step 2: S computes A= h(Y_R^x mod p) and e= g^x mod p.

Step 3: S computes e_R = h(k), fs = $^{h(m}$sllgrsYxeR mod p), es = (rs—f_5)X_5^{-1} mod q, and e_R' = e_R— A mod q.

Step 4:S sends {g', (es, ex$^{?}$, ^{f}s), ms} to Receiver R.

Step 5: R recovers A= h((e)^{xR} mod p).Then R performs the validation

f_5 ^{h}au$^{(i}$th^{ils}e^{l}ng^{t}fi^{s}cYat^{s}e$^{es(YS^{R})e}$an^{o-}; to (es, mod p) fs) after a $^{\Downarrow}$fter (m o d

receiving (es, e_R', f_5) and m_5. If the validation holds, then the intended receiver R is assured that (es, e_{ll}', f_5) and ms are indeed sent by Sender S.

The following shows why the validation can be used to validate (es, eR, f_5) and m_5.

fs h(ms g^{fs}xYs^{es}x (Y_R)$^{eR\div A}$ mod p)

h(m_5**11** g^{fs}x (g^{xs})^{es}x Y_R^{eR} mod p)

$^{-1}$

h($m_5$11g^{fs}x (gxs)(rs-fs)xs ^{1}$_{x}$ YReR mod p)

h(m511g^{rS}YR$^{eR\ m}$od $^{P)}$

3.2 Security Analysis

Security analysis of DAP-ASP is given below. The sent message {g', (es, ell', ^{f}s)} plays an important role in DAP-ASP. The

unoforgeability of (es, eR, f_5) for a given g' is shown by Lemma 1. Then Lemma 2 is used to

show that {g', (es, e_{12}', f_5)} can be forged by the receiver to provide deniability for senders. Finally Lemma 3 shows that no one can compute A= h((e^R mod p) to validate {g', (es, $e_{12\%}$ f_s), m_s}, except senders and receivers. By Lemma 3, no one can ensure who the sender is.

Because Lemma 1 is based on the decision Diffie-Hellman problem (DDHP for short), the DDHP is stated first.

Decision Diffie-Hellman problem

Let p and q be two large primes such that cap-1). Let g be a generator with order q. Given g^a mod p, g^b mod p, and g^c mod p. Determine whether or not g^c. g^{ba} (mod p).

DDHP assumption

There is no polynomial time algorithm with at least probability c to solve DDHP problem, where c is negligible.

Lemma 1: Unforgeability of (es, eR, fs) for the given g' and ms.

Being based on the DDHP assumption, there is no polynomial time algorithm to forge our triple (es, e_{II}', fs) with at least probability e except the sender S and the intended receiver R, where e is negligible.

Proof: Assume that there is a polynomial-time adversary algorithm F to forge (es, eR, fs) for the given g'= gx mod p and ms with probability co e. The input of the algorithm F is the tuple (Ys, ^{Y}R, e, ms) and the forged output is (es, e_{12}', fs) such that f_s ^{h}(^{m}dg^{fSY}S^{eSY}R^{eR} mod p) (mod q) with e_R=e_R'+A, where A= h(Y_R^x mod p). With the help of the adversary algorithm F, an algorithm I can be constructed to solve DDHP with probability at least e' and at most executing time t_1.

Solve DDHP. Algorithm I takes the input (g^a mod p, g^b mod p, g^c mod p) to determine whether or not g' g^{ba} (mod p). Algorithm I first randomly choose a random integer we Z_q* and computes y = g^w mod p. Algorithm I also randomly generates a message M. Then Algorithm I constructs the input (^{Y}5, ^{Y}R, ^{g}', ^{m}s) = (y, g^a mod p, g^b mod p, M) for Algorithm F. With the help of the adversary algorithm F, the algorithm I obtains the forged tuple (es, ex', ^{f}s) satisfies fs h(MIIg^{fS}y^{eS}e^{eR} mod p) (mod q) with ^{ab} mod p) probability at least e, where e_R= e_R^LFh(g mod q.

Now Algorithm I computes e_1= e_R^LFh(g^c mod p) mod q and uses the validation equation fs h(MIIg^{fs}y^{es}g^{axel} mod p) (mod q) to determine whether or not & g^{axb} (mod p). If fs h(MIIg^{fs}y^{es}g^{axel} mod p) (mod q) holds, the algorithm I determines & gaxb (mod p). Otherwise, the algorithm I given the answer that g^c gXb (mod p).

Consider the situation of the algorithm I outputs wrong answers. The first case is that Algorithm F fails to forge (es, eR, fs) when & g^{ab} (mod p). The probability of this case is (1-w)x(1/q). The second case is that Algorithm F successfully forge (es, e_{II}', fs) but gel $_g$ab (mod p). The necessary successful condition for this case is the collision of h occurs. Therefore, the probability of this case is cox(1/q) x(1-1/q). Therefore, the probability of Algorithm I outputting wrong answer is cox (1/q) x(1-1/q)+ (1-w)x(1/q)= (1/q)- co(1/q^2). Since co is large than 1-e and q is very large, the probability of Algorithm I with wrong answer is also negligible.

Lemma 2: Ambiguity of (es, eR, fs) for the given g' and ms.

Both the sender S and the intended receiver R can construct the triple (es, eR, fs) for a designated message m_s, so other contenders cannot find out who is the real generator of triple (es, eR, f_5).

Proof: The intended receiver R forges the triple (es, e_R", f_R) by the following steps:

Step 1:Choose 3 random numbers es, y & r_REZ$_q$*

Step 2:Compute g" = g^Y mod p and A'=h(Y_R^Y mod p).

Step 3:Compute f_R = ^{h}(^{m}ile^{Y}S$^{eS'}$ mod p), eR = (r_R—f_R)Xs^{-1} mod q, e_R" = e_R"- A' mod q.

Then R successfully constructs a triple (es, en", f_R) for message m' satisfying f_R h(m'llg^{fR}YR$^{eR"}$YS$^{eS'}$ mod p) (mod q). Since R is able to construct (es, e_R", f_R), no one can figure out whether or not (es, ee, f_R) is made by S or R except the sender and the receiver. So the triple (es, e_R", f_R) provides the ambiguity.

Theorem 2: The DAP-ASP provides authentication and deniable properties.

Proof: The triple (es, eR, fs) is unforgeable except the sender and the receiver by Lemma 1, so an adversary cannot impersonate as the sender S to cheat the intended receiver. So the DAP-ASP provides authentication property. On the other hand, the triple (es, e_{II}', fs) is only ambiguous between S and R by Lemma 2, the receiver R cannot uses the triple (es, e_{II}', fs) to prove that S is the source of the message to the third party. The DAP-ASP provides deniable property.

Lemma 3: The difficulty of computing A for

given g' and Y_R is based on DDHP.

The computation of A for given g' and Y_R is based on the difficulty of the DDHP.

Proof: Assume that there is a polynomial-time adversary algorithm E with probability at least c" to compute A for the given $e= g^x \bmod p$ and $Y_R= g^{xR} \bmod p$. The input of the algorithm E is the tuple (g', Y_R) and the output is A such that $A = h(g^{xxxR} \bmod p)$. With the help of the adversary algorithm E, an algorithm J can be constructed to solve DDHP with probability at least c" and taking at most executing time t_2.

Solve DDHP. Algorithm J takes the input ($g^a \bmod p$, $g^b \bmod p$, $g^c \bmod p$) to determine whether or not $g^c = g^{axb} \pmod p$.) Then Algorithm J constructs the input (g', Y_R) = ($g^a \bmod p$, $g^b \bmod p$) for Algorithm E. With the help of the adversary algorithm E, the algorithm J make $A = h(g^{axb} \bmod p)$ with probability at least c".

Algorithm J uses the validation equation $A = h(g^c \bmod p)$ to determine whether or not $g^c = g^{axb} \pmod p$. If $A= h(g^c \bmod p)$ holds, the algorithm J determines that $g^c = g^{aXb} \pmod p$ holds. Otherwise, Algorithm J gives the answer that $g^c \bmod p \# g^{axb} \bmod p$.

Consider the situation of Algorithm J cannot output correct answers. When Algorithm E successfully computes A but $g^c \bmod p \#$ garb mod p, Algorithm J returns that & gaXb (mod p) holds only when the collision of h occurs. So the probability of this situation is cox (1/q)(1-1/q), where co is the successful probability of Algorithm E. When Algorithm E fails to compute A but $g^c \bmod p= gax^b \bmod p$, the probability of this case is (1-w)(1/q). The total probability is wx(1/q)(1-1/q)+(1-w)(1/q)= (1/q)-co(1/q2). Since q is large and co is almost 1, this probability is negligible that Algorithm J cannot output correct answer.

Theorem 3: In DAP-ASP, the anonymity of (es, e_{it}', fs) is provided.

Proof: After obtains (es, eR, fS), an adversary A wants to cheat the receiver R by construct a value of A' which satisfied $g^{fSYSeSYRe"} = g^{fSYAeSYRe"}$, where $Y_A= g^{XA} \bmod p$ is randomly chosen. Then, the value of A' = es((Xs-XA)/XR) + A. However, the computation of A is harder than the DDHP according to Lemma 3. Since that nobody can compute the value of A, the value of A' is also not easy to be found. Therefore, it is infeasible for anyone except the origin sender and the intended receiver to find out their identity. Thus, the DAP-ASP provides the property of anonymity.

4. Performance Analysis and Discussions

The performance analysis among DAP-SP, DAP-ASP, and Lee et al's protocol [11] is given in Table 1. Notation T_E denotes the computational cost of one modular exponentiation and TH denotes the executing time for one hash operation. Both in DAP-SP and DAP-ASP, multi-exponentiation is used. The multi-exponentiation computational costs for $a^{ix1a2x2 \text{ and } aix1a2x2a3x3}$ are about 1.16 TE and $1.25T_E$, respectively [1]. In DAP-SP, the major computational cost paid by the sender S is $2T_H+1.16T_E$ in Step 2 and $T_H+1.25T_E$ in Step 4 by the receiver R. In DAP-ASP, the major computational cost paid by the sender S is $3T_H+3.16T_E$ in Step 2 and $2T_H+2.25T_E$ in Step 4 by the receiver R. In [11], the major computational cost paid by the sender is $2T_H+2T_E$ while the cost paid by the receiver is $2T_H+1.16T_E$. Evidently, the total computational cost of DAP-SP is less than the total cost of Lee et al.'s protocol [11]. The communication cost of the DAP-SP is 34 bits. On the other hand, the communication cost of Lee et al.'s protocol is **IP1±1c11•** Since Ipl is usually larger than Iql, the DAP-SP is close to Lee et al.'s protocol [11] in the communication cost. For the DAP-ASP, the communication cost slightly raise to Ip1+314. The additional communication cost of our protocol with anonymous sender protection is 214 bits. Fortunately, due to **IP1>>1c11,** the additional communication cost is cheap for the anonymous sender protection property.

The security property comparison among DAP-SP, DAP-ASP and Lee et al.'s protocol [11] is given in Table 2. Lee et al.'s protocol [11], DAP-SP, and our protocol conform to the basic properties of deniability and authentication for deniable authentication protocols. These three protocols are all non-interactive. However, both our protocol and DAP-SP provide the sender protection for every session while Lee et al.'s protocol does not. Only the DAP-ASP provides the anonymity for the transmitted message.

The bonus of the anonymous sender protection is that the received deniable evidence can become a formal sender's signature after releasing the keystone. According to the example which is described in Section 1, if a customer decides trading with some merchant after receiving the quotation, he/she claims to the merchant. When the sender who releases his/her keystone, it can complete the unfogeability and make the sent

messages as a legal signature. No additional computation cost is needed to convert sent messages to legal signatures. In the other deniable authentication protocols, senders need generate legal signatures additionally.

5. Conclusions

A deniable authentication protocol DAP-ASP is proposed with not only authentication and deniable properties but also the anonymous sender protection. The DAP-ASP is proposed to protect senders' privacy more completely than DAP-SP. Not only the transmitting message the triple (es, $e_{12\%}$ fs) is ambiguous and unforged, with a special value of A, the sender's anonymity of (es, e_ii, fs) can be also provided.

The bonus of (anonymous) sender protection is that the sender can easily convert the sending deniable evidence to legal senders' signatures. That is the converted senders' signature can be directly used for further application without paying any additional computation cost.

References

[1] Ateniese, G., "Efficient Verifiable Encryption (and Fair Exchange) of Digital Signature," *Proc. of AMC Conference on Computer and Communications Security (CCS'99), ACM Press,* pp. 138-146, New York, U.S.A., 1999.

[2] Aumann, Y. and Rabin, M., "Authentication Enhanced Security and Error Correcting Codes," *Crypto '98, Santa Barbara, CA, USA, LNCS 1462,* New York: Springer-Verlag, pp. 299-303, 1998.

[3] Aumann, Y. and Rabin, M., "Efficient Deniable Authentication of Long Messages," *Int. Conf on Theoretical Computer Science in Honor of Professor Manuel Blum's 60th birthday,* 1998. < http://www.cs.cityu.edu.hk/dept/video.html >

[4] Brown, D. R. L., "Deniable Authentication with RSA and Multicasting," Cryptology ePrint Archive: Report 2005/056, Feb. 24, 2005

[5] Chang, Y. F., Chang, C.C., and Kao, C.L., "An Improvement on a Deniable Authentication Protocol," *ACM SIGOPS Operating Systems Review, Volume 38, Issue 3,* pp.65-74, July 2004.

[6] Chen, L., Kudla, C., and Paterson, K.G., "Concurrent Signatures," *Eurocrypt '04, LNCS 3027,* New York: Spriger-Verlag, pp. 287-305, 2004.

[7] Deng X., Lee C.H., and Zhu H., "Deniable Authentication Protocols", *IEE Proceeding-Computers and Digital Techniques,* Vol.148, No.2, pp. 101-104, 2001.

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

[9] Fan, L., **Xu,** C.X., and Li, J.H., "Deniable Authentication Protocol based on Diffie-Hellman Algorithm," *Electronics Letters,* Vol.38, No.4, pp. 705-706, 2002.

[10] Hwang, S.-J. and Ma, J.-C., "Deniable Authentication Protocols with Sender Protection," *2007 National Computer Symposium,* Wufeng, Taichung, Taiwan, R.O.C., Dec. 20-21, 2007, Vol. 4, pp. 762-767.

[11] Lee, Wei-Bin, Wu, Chia-Chun, and Woei-Jiunn Tsaur, "A Novel Deniable Authentication Protocol Using Generalized ElGamal Signature Scheme," *Information Sciences,* Vol.177, pp.1376-1381, 2007.

[12] Lu, R., and Cao, Z., "Non-Interactive Deniable Authentication Protocol based on Factoring," *Computer Standards and Interfaces,* Vol. 27, pp. 401-405, 2005.

[13] Lu, R., and Cao, Z., "Erratum to Non-Interactive Deniable Authentication Protocol based on Factoring," *Computer Standards & Interfaces,* Vol. 29, p. 275, 2007.

[14] Lu, R, Cao, Z., Dong, X., and Su, R., "Group Oriented Deniable Authentication Protocol," *International Multi-Symposiums on Computer and Computational Sciences (IMSCCS'06),* June 20-24, 2006

[15] Shao, Z., "Efficient Deniable Authentication Protocol based on Generalized ElGamal Signature Scheme," *Computer Standards & Interfaces,* Vol. 26, pp. 449-454, 2004.

[16] Sun, H.M., Wang, K.H., Chang, S.Y., and Wan, L., "An Authentication Protocol Combining Deniability and Forward Secrecy for Resisting Adaptive Attacks," *International Computer Symposium 2006,* Dec 04-06, 2006

[17] Susilo, W., Mu, Y., and Zhang, F., "Perfect Concurrent Signature Scheme," *Information and Communications (ICICS '04),* LNCS 3269, New York: Springer-Verlag, pp. 14-26, 2004.

[18] Shi Y. and Li J., "Identity-based Deniable Authentication Protocol," *IEE ELECTRONICS LETTERS,* Vol. 41, No.5, March 2005

[19] Wang, G., Bao, F., and Thou, J., "The Fairness of Perfect Concurrent Signatures, "

Information and Communications Security (ICICS 06), LNCS 4307, New York: Spriger- Verlag, 2006, pp. 435-451.

[20] Zhu, R. W., Wong, D. S. and Lee, C. H., "Cryptanalysis of a Suite of Deniable Authentication Protocols," IEEE *Communications Letters,* VOL. 10, NO. 6, pp. 504-506, 2006.

Table 1: Computation and Communication Cost Comparison of Lee et al's Protocol, DAP-SP and DAP-ASP

	Lee et al.'s protocol	DAP-SP	DAP-ASP
Sender's computation cost	$2T_H$-F$2T_E$	$2T_H$-F$1.16T_E$	$3\ T_H$+$3\ .16T_E$
Receiver's computation cost	$2T_H$-F$1.16T_E$	T_H-F$1.25T_E$	$2T_H$+$2.25T_E$
Total computation cost	$4\ T_H$+$3.16T_E$	$3T_H$+$2.41T_E$	$5T_H$+$5.41T_E$
Communication cost	11)1·1(41	31c11	1 1 3 1 · 3 1 4

Table 2. **Security Comparison among Lee et al's Protocol, DAP-SP and DAP-ASP**

	Lee et al.'s protocol	DAP-SP	DAP-ASP
Deniability	Yes	Yes	Yes
Authentication	Yes	Yes	Yes
Non-interactive	Yes	Yes	Yes
Sender protection	No	Yes	Yes
Anonymity	No	No	Yes