A Price Negotiable Transaction System

H. Zhu

January 19, 2003

Department of Information Science and Electronics Engineering, Zhejiang University, PR. China
Email: zhuhf@zju.edu.cn

Abstract

We present a practical protocol that allows two players to negotiate price over the Internet in a deniable way so that a player A can prevent another player B from showing this offer P to a third party C in order to elicit a better offer while player B should be sure that this offer P generated by A, but should C be unclear whether P is generated by A or B itself, even C and B fully cooperated. Our protocol is a standard browser-server model and uses a trusted third party, but only in a very limited fashion: the trusted third party is only needed in the cases where one player attempts to cheat or simply crashes, therefore, in the vast of majority transactions, the third party is not to be involved at all. In addition, Our price negotiable transaction system enjoys the following properties:

(1) It works in an asynchronous communication model.

(2) It is inter-operated with existing or proposed scheme for electronics voting system;

(3) The two players need not sacrifice their privacy in making use of the trusted third party;

(4) The deniable property can be proved secure in the random oracle paradigm, while the matching protocol can be proved secure in the standard intractable assumption.

Contents

1 Introduction

As more business is conducted over the Internet, the price negotiable problem assumes increasing importance. For example, suppose player A is willing to order good from player B. Typically A makes a price offer P to B and creates an authenticator of P. At the end of negotiation phase, A and B makes the same decision about the price. The problem are those:

- How can a player A prevent B from showing this offer P to another party C in order to elicit a better offer while player B should be sure that this offer P generated by A, but should C be unclear whether P is generated by A or B itself, even C and B fully cooperated.
- How to prevent one player attempting to cheat or simply crashing when both parties have agreed on the negotiated price P.

Of course, one could use an on-line trust third party in every transaction to act as mediator: each player sends his conversation to the third party, who upon verifying the correcting of the both conversations, forwards the conversation to the other player. This is rather straightforward solution and has been discussed in the papers [CTS, DGLW, FR].

In this paper, we present a practical solution to price negotiable transaction system. Our protocol is a standard browser-server model and uses a trusted third party, but only in a very limited fashion: the third party is only needed in the cases where one player attempts to cheat or simply crashes, therefore, in the vast of majority transactions, the third party is not to be involved at all. This idea of making use of a trusted third party in minimal is not new. It has been used for the construction of optimistic fair exchange of digital signatures by Asokan, Schunter and Waidner [ASW97] and Asokan Shoup, and Waidner[ASW98].

Our protocol can be used to prevent coercion from electronic voting system. Let party A be a voter and party B be be a tallying authority. Suppose a party C compels the voter to select a predetermined candidate but A is unwilling to select this candidate. The voter A should send his ballot P, together with its authenticator to the tallying authority B, so that B makes sure that this ballot is generated by A not by anyone else. Thus it is desirable for A that B cannot prove to third party C that this ballot P is from A even if B and C cooperated fully. That is even if there is full cooperation, but C may be skeptical of the evidence provided by B.

Our price negotiable transaction system enjoys the following properties:

(1) It works in an asynchronous communication model: there is no need for synchronized channel, and one player cannot force the other to wait for any length of time;

(2) To use it, one need not modify message format at all. Thus, it will be inter-operated with existing or proposed scheme for electronics voting system;

(3) The two players need not sacrifice their privacy in making use of the trusted third party;

(4) The deniable property can be proved secure in the random oracle paradigm, while the matching protocol can be proved secure in the standard intractable assumption.

2 Technique overview

In this section, we discuss some of the more subtle issues arise in the design of a price negotiable transaction system, and give a high level sketch of our protocol.

2.1 Subtle issues

Our price negotiable transaction system is standard browser-server model. A party A browses the items advertised by a party B over Internet. A standard transaction mainly consists of two stages: price negotiation stage and contacting signing stage.

Price negotiation phase In this price negotiation phase, we have two players A and B, the trusted third party T need not to be involved in this stage. We assume that the players make use their public key infrastructure to negotiate and exchange price back and forth. The subtle issue we are considering is this: how can a player A prevent B from showing this offer P to another party C in order to elicit a better offer while player B should be sure that this offer P generated by A, but should C be unclear whether P is generated by A or B itself, even C and B fully cooperated.

A key idea behind our protocol is to make use of deniable ring authentication scheme. As digital signatures enable authenticating messages in a way that disallows repudiation, we need a cryptographic primitive so that it is possible to convince a verifier that a member of an ad hoc subset of participants is authenticating a message m without revealing which one, and the verifier V cannot convince a third party that message m was indeed authenticated, that is there is no trail of the conversation, other than what could be produced by verifier alone.

Contract signing phase In this setting we have two players A and B, and a trusted third party T acting as a mediator. The trust party T receives request from each party, updates its internal state, and generate its response. We stress that both A and B know $T's$ public key. The problem we are considering is this: how to prevent one player attempting to cheat or simply crashing when both parties have agreed on the negotiated price P.

The key idea to solve the problem is that we make use of matching protocol which works in the deniable authentication channel. That is each party generates a pair of random variables, then interacts with each other so that only A and B can compute a final value related to the determined price P. Finally, each party delivers the correspondent value to the trusted third party, if the two values are equal, then T delivers the message "accept" to each party, otherwise, T delivers "reject" to each party.

2.2 Our protocol

Our price negotiable transaction system ensures timely termination, without make any assumption about the network, other then the assumption that a player can eventually the trusted third party. Indeed, can unilaterally terminate at any point in the time, either by simply terminate, or by contacting the third party and then terminating.

We sketch the high level logic of our protocol.

Let party A be a buyer and party B be a seller. Both parties hold correspondent certificated public keys.

1. Party A sends a login request to party $B's$ server using a ring authentication scheme, i.e., $LogReq = Rsig_{(A,B)}(Cert_A, k, GID)$, where $Rsig_{A,B}$ is two-party deniable ring signature scheme, k is a session key and GID is the encoding string of some good A is willing to buy;

2. Party B verifies the signature scheme of the login request message $\sigma = (Cert_A, k, GID)$. If σ is a valid message, then it creates a message called login acknowledgement denoted by $LogAck$;

3. Party A and party B interactive with each other back and forth until a price P is agreed by both parties. We set $P \leftarrow null$ if there is no agreement at the end of interactions;

4. Party A generates a pair of random variables, and sends it to B. These random strings are authenticated by a ring authentication scheme;

5. Upon receiving a valid strings from A, party B generates another random pair and sends random strings back to A by means of a ring authentication scheme.

6. Finally, both parties send the final encrypted value to T, if these values are matched, then T delivers the message "accept" to each party, otherwise, T delivers "reject" to each party.

Clearly, our protocol is differently from any previous published in the literature [ASW97, ASW98, CTS, ST] and it heavily relies on matching protocol and ring signature scheme [RST].

This protocol makes use of two sub-protocols: an ring authentication protocol and a matching protocol. The deniable ring authentication protocol is a request that a player A prevents B from showing this offer P to another party C in order to elicit a better offer while player B should be sure that this offer P generated by A, but should C be unclear whether P is generated by A or B itself, even C and B fully cooperated. A matching protocol is a request that it prevents one player attempting to cheat or simply crashing when both parties have agreed on the negotiated price P.

Our approach to define security model for a price negotiable transaction system is that of modern theoretical cryptology, based on complexity theory. In this setting, one explicitly states the assumptions made about the communication network, and the power of the attacker or adversary- its computational power as well how it may interact with the system. Additionally, one explicitly states the security goal, i.e., what it means to break the system. We define security in terms of completeness and deniable property.

Deniable property captures the intuition that there is full cooperation between prover and verifier, but the adversary may be skeptical on the truth of the evidences provided by the verifier. Therefore if the verifier himself can simulate all transcripts between a real prover and a real verifier, then the protocol is deniable.

Completeness means that if neither player is corrupt, and no messages are lost, then the exchange is successful.

3 Matching protocol

3.1 Basic matching protocol

In this subsection we are considering a practical matching scheme that works in the deniable authentication channel. That is given a deniable authentication channel, we are willing to build a matching

protocol. Let p be a large prime number such that the discrete logarithm problem defined in Z_p^* is hard. Let $G \in Z_p^*$ be a cyclic group of prime order q. Denote g be a generator of G (we assume that G is prime order, e.g., p=$2q+1$ and $ord(g)$=q, through out this report). We describe a practical matching protocol below:

1. Party A sends $u = g^r, v = g^{r^2}$ to party B;

2. Upon receiving (u, v), party B computes $c = g^{x_1}u^{x_2}$, $d = g^{y_1}u^{y_2}$, and sends (c, d) to A;

3. A computes $W_A = c^r d^{\alpha r}$ and sends the cipher-text of W_A to the trust third party T. Meanwhile B computes $W_B = u^{x_1+\alpha x_2}v^{y_1+\alpha y_2}$ and sends the cipher-text of W_B to T;

4. T decrypts the two cipher-text and recover the W_A and W_B, if both $W_A = W_B$, then T delivers the message "accept" to each party, otherwise, T delivers "reject" to each party.

Since our basic matching protocol works in the deniable authentication channel, security of the basic matching protocol means that an adversary other then A and B can compute W_{Adv} such that $W_{Adv} = W_B$ with negligible probability. Notice that the adversary can enquire the corresponding oracle queries from time to time to capture the definition of adversary to adaptive chosen cipher-text attack. However the adversary is restricted not to rewind the internal coin tosses generated by A and B, where A and B are defined to be a pair of interactive, probabilistic polynomial time Turing machines.

Reduce basic matching protocol to Diffie-Hellman problem The proof of security of basic matching problem relies on the square of Diffie-Hellman problem [DH], we therefore sketch related problems below. Let p be a large prime number such that the discrete logarithm problem defined in Z_p^* is hard. Let $G \in Z_p^*$ be a cyclic group of prime order q. Denote g be a generator of G (we assume that G is prime order, e.g., p=$2q+1$ and $ord(g)$=q, through out this report). Computational Diffie-Hellman assumption (CDH assumption) is referred to as the following statement:

- Computational Diffie-Hellman problem (CDH): On input g, g^x, g^y, computing g^{xy};
- Square computational Diffie-Hellman problem (SCDH): On input g, g^x, computing g^{x^2};
- Inverse computational Diffie-Hellman problem (InvCDH): On input g, g^x, outputs $g^{x^{-1}}$;
- Divisible computation Diifie-Hellman problem (DCDH problem): On random input g,g^x, g^y, computing $g^{y/x}$.

Note that all variations of computational Diffie-Hellman problem are equivalent, i.e., CDH $\Leftrightarrow$ SCDH $\Leftrightarrow$ InvCDH $\Leftrightarrow$ DCDH.

Similarly, one can define the decisional cases of the above variations.

Decisional Diffie-Hellman assumption-DDH Let G be a large cyclic group of prime order q defined above. We consider the following two distributions:

- Given a Diffie-Hellman quadruple g, g^x, g^y and g^{xy}, where $x, y \in Z_q$, are random strings chosen uniformly at random;

- Given a random quadruple g, g^x, g^y and g^r, where $x, y, r \in Z_q$, are random strings chosen uniformly at random.

An algorithm that solves the Decisional Diffie-Hellman problem is a statistical test that can efficiently distinguish these two distributions. Decisional Diffie-Hellman assumption means that there is no such a polynomial statistical test. This assumption is believed to be true for many cyclic groups, such as the prime sub-group of the multiplicative group of finite fields.

Square decisional Diffie-Hellman assumption-SDDH: Let G be a large cyclic group of prime order q defined above. We consider the following two distributions:

- Given a square Diffie-Hellman triple g, g^x and g^{x^2}, where $x \in Z_q$, is a random string chosen uniformly at random;
- Given a random triple g, g^x and g^r, where $x, r \in Z_q$, are two random strings chosen uniformly at random.

An algorithm that solves the square decisional Diffie-Hellman problem (SDDH for short) is a statistical test that can efficiently distinguish these two distributions. Square decisional Diffie-Hellman assumption means that there is no such a polynomial statistical test.

Inverse decisional Diffie-Hellman assumption -InvDDH Let G be a large cyclic group of prime order q defined above. We consider the following two distributions:

- Given a inverse Diffie-Hellman triple g, g^x and $g^{x^{-1}}$, where $x \in Z_q$, is a random string chosen uniformly at random.;
- Given a random triple g, g^x and g^r, where $x, r \in Z_q$, are random strings chosen uniformly at random.

An algorithm that solves the Inverse decisional Diffie-Hellman problem (InvDDH for short) is a statistical test that can efficiently distinguish these two distributions. Inverse decisional Diffie-Hellman assumption means that there is no such a polynomial statistical test.

Divisible decision Diffie-Hellman assumption-DDDH Let G be a large cyclic group of prime order q defined above. We consider the following two distributions:

- Given a divisible Diffie-Hellman quadruple g, g^x, g^y and $g^{x/y}$, where $x, y \in Z_q$, are random strings chosen uniformly at random.;
- Given a random quadruple g, g^x and g^y and g^r, where $x, y, r \in Z_q$, are random strings chosen uniformly at random.

An algorithm that solves the divisible decision Diffie-Hellman problem (DDDH for short) is a statistical test that can efficiently distinguish these two distributions. Divisive decision Diffie-Hellman assumption means that there is no such a polynomial statistical test.

It is easy to show that: InvDDH $\Rightarrow$ SDDH, SDDH $\Rightarrow$ InvDDH, SDDH $\Rightarrow$ InvDDH and SDDH $\Rightarrow$ DDH. Unfortunately, we are not able to prove that DDH $\Leftrightarrow$ SDDH. We believe that the decisional Diffie-Hellman problem is equivalent to square decisional Diffie-Hellman problem in our setting, i.e., we assume that $G \in Z_p^*$ is a cyclic group of prime order q, g is a generator of G and G is prime order, where p=$2q + 1$ and $ord(g)$=q. This leaves an interesting conjecture.

The proof of security Suppose there is a probabilistic polynomial time adversary, so that it can compute $W_{Adv} = W_B$ with non-negligible probability, then we use adversary as a subroutine to break square decisional Diffie-Hellman problem. More details:

Now given a pair (u, v), which comes either from a random pair or a square Diffie-Hellman pair, we build up a simulator as follows (also the simulation works in the deniable authentication channel):

We choose a random strings $x_1, x_2, y_1, y_2 \in Z_q$, and compute $c = g^{x_1}u^{x_2}$,$d = g^{y_1}u^{y_2}$, the adversary then obtains (u, v) and (c, d), finally, the adversary computes W_j and get a bit from the simulator. We consider the following two cases:

Case1 if (u, v) comes from a random pair, due to the fact that $W_B = u^{x_1+\alpha x_2}v^{y_1+\alpha y_2}$, i.e., $logW_B = r_1(x_1 + \alpha x_2) + r_2(y_1 + \alpha y_2)$, it is easy to show that the success probability is at most k/q, where k is total amount of queries to the simulator oracle and k/q is an negligible amount.

Case2 If (u, v) is a square Diffie-Hellman pair. The simulator is perfect from the point views of the adversary. therefore by assumption, the adversary will have a negligible probability to get a exact value.

Based on the above simple argument, it is easy for one to build a square decisional Diffie-Hemman distinguisher as follows.

- The input to subroutine is (u, v),
- The output of subroutine is $W_1, \cdots, W_k$, where k is a polynomial of security parameter,
- If there exists j such that $W_j = W_B$, then it outputs a bit 1, otherwise, it outputs 0.

Notice that our basic scheme is defined over deniable authentication channel. To define the matching protocol over a public channel, we need further reduction. Fortunately, we are able to apply the modulo design approach to improve this basic scheme.

3.2 Matching protocol working in public channel setting

Since our matching protocol defined over deniable authentication channel, we need further reduce the protocol so that it can work in public channel.

Deniable authentication protocol Recall that a deniable authentication channel enables a receiver to make sure of the source of a given message but cannot prove to a third party of the identity of the sender. Basically, there are three players in the security model: a prover P, a verifier R and an inquisitor INQ. The communications between the prover P and the verifier R are connected by an insecure link. INQ is a person in the middle, sitting on the link between P and V, intercepting the traffic between them and injecting messages of his own and INQ can later compel P and V to reveal all the security data. Hence all information shared between the sender and the receiver is available to the inquisitor. However, it is assumed that the communication between P and V is such that listening to the transmission does not identify P. Moreover, the inquisitor and the receiver fully cooperate with each other. Namely, the receiver gives the inquisitor any information for which she asks. Thus, the right formulation is that there is full cooperation, but the inquisitor may be skeptical on the truth of the evidences provided by the receiver. Therefore if the receiver himself can simulate all transcripts between a real prover and a real verifier, then the protocol is deniable.

One notable deniable authentication scheme has been developed by Dwork, Naor and Sahai as an application of the concurrent zero-knowledge proofs [DNS]. Recently, Rivest, Shamir and Tauman proposed the notion of ring signatures to prove that a message is authenticated by a member of group. Their scheme is very efficient and can be proved secure in the random oracle paradigm [BR]. Following their beautiful works, Naor [Na] also proposed several interesting protocols, all these protocols can be proved secure against active attack in the standard intractability paradigm.

Rivet-Shamir-Tauman based matching protocol Very recently, Rivest, Shamir and Tauman [RST], formalize the notion of ring signature, which makes it possible to specify a set of possible signers without revealing which member actually produced the signature. A ring signature has no group managers, no setup procedures, no revocation procedures and no coordination: any user can choose any set of possible signers that including himself, and sign any message by using his secret keys and the other's public key, without getting their approval or assistance. RST's ring signature scheme is unconditionally signer-ambiguous, provably secure in the random oracle model, therefore the scheme is suit for the construction of practical matching protocol that works in the public channel setting.

The description of Rivet-Shamir-Tauman based matching protocol. Let p be a large prime number such that the discrete logarithm problem defined in Z_p^* is hard. Let $G \in Z_p^*$ be a cyclic group of prime order q. Denote g be a generator of G (we assume that G is prime order, e.g., p=$2q+1$ and $ord(g)$=q, through out this report). Also each party has a RSA public key n_A and n_B, and the corresponding secret key p_A, q_A such that $p_A q_A = n_A$ and p_B, q_B such that $p_B q_B = n_B$ respectively, where E, a symmetric encryption scheme behaves as a random oracle. Based on Rivet-Shamir-Tauman's protocol(Please refer [RST] for more details), we are able to present a practical matching protocol below:

Protocol 1

1. Party A generates a random string $u = g^r, v = g^{r^2}$, and computes ${x_1}^2 mod \mathrm{n_A} = E_{h(u,v)}({y_1}^2 \mathrm{mod n_B})$, and sends (u, v, x_1, y_1) to the party B;

2. Upon receiving (u, v, x_1, y_1), party B checks the validation of receiving message, if it is valid then it computes $c = g^{x_1} u^{x_2}$, $d = g^{y_1} u^{y_2}$, and ${x_2}^2 \mathrm{mod n_A} = E_{h(u,v)}({y_2}^2 \mathrm{mod n_B})$ and sends (c, d, x_2, y_2) to A;

3. A checks the validity of receiving message, if it is valid then it computes $W_A = c^r d^{\alpha r}$ and sends the cipher-text of W_A to the trust third party T. Meanwhile B computes $W_B = u^{x_1 + \alpha x_2} v^{y_1 + \alpha y_2}$ and sends the cipher-text of W_B to T;

4. T decrypts the two cipher-text and recover the W_A and W_B, if both $W_A = W_B$, then T delivers the message "accept" to each party, otherwise, T delivers "reject" to each party.

The functions of steps 1-2 ensure the information exchanging protocol works in a deniable authentication channel, by applying the results studied by Rivest, Shamir and Tauman. Based on the above observation, we claim that:

Theorem 1 The Rivet-Shamir-Tauman based matching protocol 1, is deniable in the public channel setting.

4 Joint encryption and deniable signature protocol

The problem we are considering is this: how to prevent one player attempting to cheat or simply crashing when both parties have agreed on the negotiated price P. In this setting we have two players A and B, and a trusted third party T acting as a mediator. The trust party T receives request from each party, updates its internal state, and generate its response. We stress that both A and B know $T's$ public key. Our approach to solve this problem is that: Let each party generates the random value encrypted using $T's$ public cryptosystem, then it authenticates the cipher-text in a deniable way.

4.1 Variation of Cramer-Shoup's encryption scheme

Asymmetric encryption scheme is a building block to design more complex cryptographic protocols. One beautiful scheme based on RSA [RSA] is due to Bellare and Rogaway [BR] which is provably secure in the random oracle paradigm, another beautiful work is Cramer-Shoup's asymmetric encryption scheme which is provably secure in the standard intractability model [CS]. Our first step is to provide an efficient variation of Cramer-Shoup's public key cryptosystem, which is suitable for our purposes. We sketch this CS-like encryption scheme as follows:

- Key generation: Let p, q be large primes such that $p = 2q + 1$ and G be a sub-group of Z_p^* with order q. Let H be a collision free hash function and $g_1 \in Z_p^*$ with order q. We choose $w, x, y, z \in Z_q$ at random and compute $g_2 = g_1^w \mathrm{mod} p$, $c = g_1^x \mathrm{mod} p$, $d = g_1^y \mathrm{mod} p$ and $h = g_1^z \mathrm{mod} p$. The private keys are (w, x, y, z). The public keys are (g_1, g_2, c, d, h, H).

- Encryption: To encrypt a message $m \in G$, it computes $u_1 = g_1^r \mathrm{mod} p$, $u_2 = g_2^r \mathrm{mod} p$, $e = mh^r \mathrm{mod} p$, $\alpha = H(u_1, u_2, e)$ and $v = c^r d^{r\alpha} \mathrm{mod} p$. The cipher-text is (u_1, u_2, e, v).

- Decryption: Given a putative cipher (u_1, u_2, e, v), it computes $\alpha = H(u_1, u_2, e)$, and tests whether the conditions $u_2 = u_1^w \mathrm{mod} p$ and $u_1^{x+y\alpha} = v \mathrm{mod} p$ hold. If both conditions hold, then the decryption algorithm outputs $m = e/u_1^z \mathrm{mod} p$, Otherwise it outputs $reject$.

Comparisons One can see that the scheme reserves the same algebraic structure as the basic Cramer-Shoup's encryption [CS]. The private keys are $w, x, y, z \in Z_q$ and the public keys of this scheme are g_1, $g_2 = g_1^w$, $c = g_1^x$, $d = g_1^y$, $h = g_1^z$. Since the private keys are $x_1, x_2, y_1, y_2, z \in Z_q$ and public keys are g_1, g_2, $c = g_1^{x_1} g_2^{x_2}$, $d = g_1^{y_1} g_2^{y_2}$, $h = g_1^z$ in the basic Cramer-Shoup's encryption scheme, this variation scheme reduce the key sizes of the basic Cramer-Shoup's scheme actually. It is clear that the computational costs of the decryption algorithm in this scheme is equivalent to that in the basic Cramer-Shoup's encryption however our decryption algorithm is more efficient to reject any invalid cipher-text. This property is desirable if the scheme is applied to construct a matching protocol.

Security analysis We consider the following game: first the encryption's key generation algorithm is run, with a security parameter as input. Next, the adversary chooses to a decryption oracle. Then the adversary chooses two messages m_0 and m_1 and sends them to the encryption oracle. The encryption oracle chooses a bit b at random and encrypts the message m_b. The correspondent cipher-text, called the target cipher-text is given to the adversary. The adversary is given the access the decryption oracle and it queries the decryption oracle polynomial sizes of the cipher-text of his/her own choices except for the target cipher-text. We say that a public key encryption scheme is secure

against adaptive chosen cipher-text the target cipher-text, if the adversary's advantage to guess the bit b is negligible. According to the Rackoff and Simon's security definition [RS], we define the game G_0 as follows.

- Key generation: Let G be a sub-group of prime order q. Let H be a collision free hash function; We choose $g_1 \in G\backslash\{1\}$ and $w, x, y, z \in Z_q$ at random and compute $g_2 = g_1^w$, $c = g_1^x$, $d = g_1^y$ and $h = g_1^z$. The private keys are (w, x, y, z). The public keys are (g_1, g_2, c, d, h, H).

- Encryption: Given two messages m_0, m_1, the encryption oracle chooses a bit $b \in \{0, 1\}$, $r \in z_q$ at random, and computes $u_1 = g_1^r$, $u_2 = g_2^r$, $e = m_b h^r$, $\alpha = H(u_1, u_2, e)$ and $v = c^r d^{r\alpha}$. The cipher-text is (u_1, u_2, e, v).

- Decryption: Given a putative cipher (u_1, u_2, e, v), it computes $\alpha = H(u_1, u_2, e)$, and tests whether $u_2 = u_1^w$ and $u_1^{x+y\alpha} = v$ hold. If the both conditions hold, then the decryption algorithm outputs m_b=e/u_1^z, Otherwise it outputs $reject$.

Claim 0 The game G_0 is secure against adaptive chosen cipher-text attack.

Proof: Suppose we are given (g_1, g_2, u'_1, u'_2), then we build a simulator G_2 as follows with the help of the adversary: the input to the simulator is (g_1, g_2, u'_1, u'_2), which comes from either Diffie-Hellman quadruple or a random quadruple, then the adversary chooses two messages m_0 and m_1 and sends them to the encryption oracle in the game G_2 which is defined below. Finally the encryption oracle chooses a bit $b \in \{0, 1\}$ at random, encrypts the message m_b and gives the adversary the correspondent cipher-text $\sigma = (u'_1, u'_2, e', v')$ (This cipher-text is called the target cipher-text). We describe the simulator, i.e., the game G_2 as follows.

- Key generation: Let G be a sub-group of prime order q. We chosen $x_1, x_2, y_1, y_2, z_1, z_2 \in Z_q$ at random and computes $c = g_1^{x_1} g_2^{x_2}$, $d = g_1^{y_1} g_2^{y_2}$ and $h = g_1^{z_1} g_2^{z_2}$. The private keys are $(x_1, x_2, y_1, y_2, z_1, z_2)$ and the public keys are (g_1, g_2, c, d, h, H), where H is a collision free hash function.

- Encryption oracle: Given two messages m_0, m_1, the encryption oracle chooses a bit b at random, then we compute $e' = m_b u'^{z_1}_1 u'^{z_2}_2$, $\alpha' = H(u'_1, u'_2, e')$ and $v' = u'^{x_1+y_1\alpha'}_1 u'^{x_2+y_2\alpha'}_2$. The cipher-text is (u'_1, u'_2, e', v').

We denote this target cipher-text by $\sigma = (u'_1, u'_2, e', v')$. The simulator answers the decryption queries according to the decryption algorithm in the game G_1, which is stated below.

- Key generation: Same as that in the game G_2;

- Encryption: Given two messages m_0, m_1, the encryption oracle chooses a bit $b \in \{0, 1\}$, $r \in z_q$ at random, and computes $u_1 = g_1^r$, $u_2 = g_2^r$, $e = m_b h^r$, $\alpha = H(u_1, u_2, e)$ and $v = c^r d^{r\alpha}$. The cipher-text is (u_1, u_2, e, v).

- Decryption: Given a putative cipher-text (u_1, u_2, e, v), it computes $\alpha = H(u_1, u_2, e)$, and tests whether $u_1^{x_1+y_1\alpha} u_2^{x_2+y_2\alpha} = v$, if this condition does not hold, the decryption algorithm outputs $reject$; otherwise, it outputs $m_b = e/u_1^{z_1} u_2^{z_2}$.

We say a cipher-text (u_1, u_2, e, v) invalid if $\log_{g_1} u_1 \neq \log_{g_2} u_2$. The rest work in this section is to show the following facts:

- If (g_1, g_2, u'_1, u'_2) comes from a random quadruple then the adversary has NO advantage guessing the bit b chosen at random by the simulator in the game G_2;
- If (g_1, g_2, u'_1, u'_2) comes from the Diffie-Hellman quadruple then the adversary has non-negligible advantage guessing the bit b chosen at random by the simulator in the game G_2 if the adversary has non-negligible advantage broken the crypto-system G_1;
- The adversary's advantage in game G_1 differs from that in the game G_0 is negligible if (g_1, g_2, u'_1, u'_2) comes from the Diffie-Hellman quadruple, that is the game G_0 is equivalent to the game G_1 from the point view of the adversary.

Claim 1 Suppose the target cipher-text $\sigma = (u'_1, u'_2, e', v')$ is computed from the random quadruple in game G_2, then the decryption oracle in game G_2 can reject any invalid cipher-text except for negligible probability.

Proof: Suppose the target cipher-text (u'_1, u'_2, e', v') is computed from the random quadruple, i.e., the input (g_1, g_2, u'_1, u'_2) to the encryption oracle in game G_2 comes from the random quadruple. With the same argument as Lemma 2 presented in [CS], one knows that the decryption algorithm in game G_2 can reject any invalid cipher-text (u_1, u_2, e, v) chosen adaptively by the adversary except for negligible amount.

Claim 2 Suppose the target cipher-text $\sigma = (u'_1, u'_2, e', v')$ is computed from random quadruple, then $\lambda' := u'^{z_1}_1 u'^{z_2}_2$ in the game G_2 is a random element in G from the point of the view of the adversary.

Proof: Suppose the target cipher-text $\sigma = (u'_1, u'_2, e', v')$ is computed from the random quadruple, that is $\log_{g_1} u'_1 \neq \log_{g_2} u'_2$. According to Claim 1, one knows that the decryption algorithm in game G_2 can reject any invalid cipher-text except for negligible amount. With the same argument as Lemma 2 in [CS], one knows that $\lambda' = u'^{z_1}_1 u'^{z_2}_2$ is a random element in the group G from the point of the view of the adversary.

Claim 3 Suppose the target cipher-text $\sigma = (u'_1, u'_2, e', v')$ is computed from the Diffie-Hellman quadruple in game G_2, then the adversary's advantage in the game G_1 differs from the advantage in the game G_2 is negligible.

Proof: By assumption, $\sigma = (u'_1, u'_2, e', v')$ is the target cipher-text computed from the Diffie-Hellman quadruple, i.e., the input (g_1, g_2, u'_1, u'_2) to the encryption oracle comes from the Diffie-Hellman quadruple in game G_2. Since the key generation algorithm in the game G_2 is the same as that in the game G_1, and the encryption oracle in game G_2 is the same as encryption algorithm in the game G_1 if the input (g_1, g_2, u'_1, u'_2) to the simulator comes from the Diffie-Hellman quadruple, also, the decryption algorithm in the game G_2 is the same as that in the game G_1, it follows that the target cipher-text $\sigma = (u'_1, u'_2, e', v')$ can be viewed as it computed from the encryption algorithm in the game G_1. Since the decryption algorithm in the game G_1, as well as the decryption oracle in the game G_2, rejects any invalid cipher-text except for negligible amount, as the same argument as Lemma 1 presented in [CS], it follows that the distribution of any valid cipher-text variable (u_1, u_2, e, v) in game G_1 is the same as that in game G_2 from the point of view of the adversary. Hence the adversary's advantage in the game G_1 is the same as that in the game G_2 except for the negligible amount.

Claim 4 Suppose the target cipher-text $\sigma = (u'_1, u'_2, e', v')$ is computed from the Diffie-Hellman quadruple in game G_1, then the adversary's advantage in game G_0 differs from that in the game G_1 is negligible.

Proof: According to the game G_0, the decryption oracle can reject any invalid cipher-text definitely. By assumption $\sigma = (u'_1, u'_2, e', v')$ is the target cipher-text computed from the Diffie-Hellman quadruple in game G_1, that is (g_1, g_2, u'_1, u'_2) comes from the Diffie-Hellman quadruple according to the game G_1. With the same argument as Lemma 1 presented in [CS], one knows that the decryption oracle in game G_1 can reject any invalid cipher-tex except for the negligible amount. It follows that the distribution of the variable of the valid cipher-text (u_1, u_2, e, v) in game G_0 is the same as that in game G_1 from the point of view of the adversary. Hence the adversary's advantage in game G_0 differs from that in the game G_1 is negligible, that is if the game G_0 is NOT secure then the adversary has non-negligible advantage breaking the game G_1.

With help of the above claims, we are able to construct a Diffie-Hellman distinguisher by considering the following two cases:

- If (g_1, g_2, u'_1, u'_2) comes from Diffie-Hellman quadruple then the adversary's view in game G_2 is equivalent to that in the game G_1, and the adversary's advantage in game G_0 differs from that in game G_1 with negligible amount. That is the adversary has non-negligible advantage correctly guessing the exact value b in game G_2 provided the adversary has non-negligible advantage broken the game G_0 in Rackoff-Simon's sense.

- If (g_1, g_2, u'_1, u'_2) is a random quadruple, then $\lambda' := u'^{z_1}_1 u'^{z_2}_2$ is a random element in the group G from the point of the view of the adversary according to the Claim 2. The fact implies that the adversary's advantage correctly guessing the bit b is negligible.

We now define the distinguisher as follows: we choose a bit b in the game G_2 at random. The distinguisher outputs 1 if the adversary's output bit b' is equal to b, and outputs 0 otherwise. This distinguisher can tell the Diffie-Hellman quadruple from the random quadruple with non-negligible amount provided the adversary has non-negligible advantage breaking the game G_0 in Rackoff-Simon's sense, which contradicts the hardness assumption of the Diffie-Hellman problem.

4.2 Joint deniable encryption and signature scheme

Suppose a honest party A, agreed on a price P, wants to make a contract signing. Fix a particular deniable ring signature scheme Σ, and consider the following game. The message P is encrypted and then the party runs Σ to authenticate the source of the cipher-text. That is we run the protocol 2 as follows:

Protocol 2

- A running the ring signature scheme defined above to authenticate the source of message m, the signature of message m, denoted by σ.

- A running the encryption scheme defined above to encrypt σ, the cipher-text is denoted by c;

- T checking the validity of the receiving cipher-text, and update its internal record if it is valid; otherwise, it rejects the received message.

To prove the security of the scheme, we need to show that the protocol is deniable in the private channel setting. That is, given a message c, T is able to generate transcripts so that it is indistinguishable from the actual transcript prescribed by the protocol. Since T holds a decryption key, it is easy to ensure this property as that doing in the simulator defined above. The deniable property follows from the characters of the ring signature scheme.

Since our encryption is provably secure against active attack in the deniable authentication setting, therefore the protocol defined above is provably secure in the public channel setting by applying the modular approach first studied by Bellare, Canetti and Krawczyk [BCK]. We claim that:

Theorem 2 Protocol 2 defined above is a jointly deniable encryption and signature scheme.

5 A practical model for negotiable transaction system

5.1 Models and definitions

We have two players A and B, and a trusted third party T that acts as a server: it receives request from a client, updates its internal states, and sends a responds back to the client however, T is restricted to be active only in the contract signing stage.

The two players agree upon the signature they want to exchange, and then exchange messages back and forth, so that the conversations between A and B are ambiguous from the point views of an adversary.

We define security in terms of deniable and complete properties. As already mentioned in the section 2, deniable property means that there is full cooperation between prover and verifier, but the adversary may be skeptical on the truth of the evidences provided by the verifier. Therefore if the verifier himself can simulate all transcripts between a real prover and a real verifier, then the protocol is deniable.

Completeness means that if neither player is corrupt, and no messages are lost, then the exchange is successful.

We now make the above notions more precise.

Behaviors of T. T is a polynomial interactive Turing machine that follows the program prescribed for it by the protocol. T acts a server, repeatedly accepting a request, updating its internal state and generating a response. T has a public key/private key pair (PK_T, SK_T) that is generated by a key generation algorithm prescribed by the protocol.

Behaviors of an honest player. An honest player is an interactive polynomial time Turing machine that follows the program prescribed for it by the protocol. It interacts with its environment through a sequence of rounds; in one round it receives a message, updates its internal states and generates a response. An honest player A has a public key/private key pair (PK_A, SK_A) that is generated by a key generation algorithm prescribed by the protocol.

Definition of source hiding Fix a particular deniable ring signature scheme Σ, and consider the following game. The players in the game are adversary, called Adv, which is an interactive polynomial time Turing machine, two honest parties, called A and B. Given a message m, the ring master chooses a bit b at random, if $b = 0$, then the party A is doing the authentication and B is one running it, we say a protocol is source hiding if the probability the adversary guesses correctly which case it is should be negligible.

Definition of deniable property Fix a particular deniable ring signature scheme Σ, and consider the following game. The players in the game are adversary, called Adv, which is an interactive polynomial time Turing machine, two honest parties, called A and B. We say a protocol is deniable if all transcript between the communication participants can be simulated by an honest party.

5.2 Main results

We are able to describe a practical price negotiable transaction system now. We assumes that the protocol is cumulative, that is the internal data in each party is update when one receives a message from the network or asks for an external request to other party. In fact, the model we are considering is this: the adversary controls the network, that is network is absorbed by the adversary, all participants in network are completely passive.

Protocol 3

- The adversary actives the protocol 2, if a price is agreed between two participants then goes to the second round; otherwise, it outputs a Null string indicating the termination of the protocol;
- The adversary actives a protocol 1, if T obtains a pair of matched values then goes to the third round; otherwise, it outputs a Null string indicating the termination of protocol;
- The adversary actives the protocol 2, if the two decrypted values are matching, then sending an "accept" notice; otherwise sending a "reject" notice to two parties.

Completeness: obvious.

To show the protocol is deniable, we consider the following cases:

Case 1: The protocol 3 terminates in the first round. Since protocol 2 is joint encryption and signature scheme, there is now information leaked even the adversary mounts an active attack. In fact, if the adversary has non-negligible advantage to distinguish party who authenticates from another party running the protocol, then there must be two cases:

case 1.1: The information is leaked from the encryption scheme. In this case, we are able to construct a Diffie-Hellman distinguisher by making use of the adversary as a subroutine. This is a contradiction.

case 1.2: The information is leaked from the ring signature scheme. In this case, we are able to construct an RSA inverter by making use of the adversary as a subroutine. This is a contradiction.

Case 2: The protocol 3 terminates in the second round. Since protocol 1 is a deniable matching protocol works in the public channel setting, we consider the two cases also:

case 2.1 Given a quadruple (u, v, c, d) as that prescribed by the protocol 1, if the adversary can forge a response value for a given quadruple W_A or W_B with non-negligible probability, then we are able to construct a Diffie-Hellman distinguisher by making use of the adversary as a subroutine. This is a contradiction.

case 2.2 The information is leaked from the ring signature scheme. In this case, we are able to construct an RSA inverter by making use of the adversary as a subroutine. This is a contradiction.

Case 3: The protocol 3 terminates in the final round. In this case, we also consider the two cases as that in the case 1.

Based on the above argument, we have the following conclusion:

Theorem 3 The price negotiable transaction protocol is deniable and it is secure against active attack (in the sense of the security definitions above).

6 Conclusions

In this report, we present a practical price negotiable transaction system that allows two players to negotiate price over the Internet in a deniable way so that a player A can prevent B from showing this offer P to another party C in order to elicit a better offer while player B should be sure that this offer P generated by A, but should C be unclear whether P is generated by A or B itself, even C and B fully cooperated. The protocol has several nice properties: it works in an asynchronous communication model: there is no need for synchronized channel, and one player cannot force the other to wait for any length of time; To use it, one need not modify message format at all. Thus, it will be inter-operated with existing or proposed scheme for electronics voting system; The two players need not sacrifice their privacy in making use of the trusted third party; And the protocol can be proved secure in the standard intractability paradigm. One issue,we do not specified is whether the protocol withstands concurrent attack or not, this leaves an interesting research problem.

References

[**ASW97**] N. Asokan, M. Schunter, M. Waidner. Optimistic protocols for fair exchange. In 4th ACM conference on computer and communication security, page 6-17, 1997.

[**ASW98**] N. Asokan, V. Shoup, M. Waidner. Optimistic Fair Exchange of Digital Signatures, In advances in Cryptology-Eurocrypt'98, 1998.

[**BCK**] M. Bellare, R. Canetti, and H. Krawczyk. A modular approach to the design and analysis of authentication and key exchange protocols. Proceedings of 30th Annual Symposium on the Theory of Computing, ACM, 1998.

[**BR** M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. Proceedings First Annual Conference on Computer and Communications Security, ACM, 1993.

[**CS**] R. Cramer, V.Shoup. A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attack. Crypto '98, 13-25.

[**CTS**]B. Cox, D. Tygar, and M. Sirbu. Net-billing security and transaction protocol, in First USENIX workshop on Electronics commerce, page 77-88, 1995.

[**DGLW**]R. H. Deng, L. Gong, A.A. Lazar and W. Wang. Practical protocol for certificated electronic mail. Journal of Network and System Management, 4(3), 1996.

[**DH**]W. Diffie and M. E. Hellman. New Directions in Cryptography. In IEEE Transactions on Information Theory, IT-22(6):644-654, November 1976.

[**DNS**]C. Dwork, M. Naor and A. Sahai. Concurrent Zero-knowledge. Proceedings of 30th ACM STOC'98, 409-418, 1998.

[**FR**] M.K. Franklin and M. K. Reiter. Fair exchange with a semi-trusted third party. In 4th ACM conference on computer and communication security, page 1-5, 1997.

[**Na**]Moni Naor: Deniable Ring Authentication. CRYPTO'02: 481-498.

[**RS**]Rackoff, C., Simon, D.: Non-interactive Zero-knowledge Proof of Knowledge and Chosen Ciphertext Attacks. Cryptology-Crypto'91. 433-444, Springer-Verlag, 1992.

[**RSA78**]R.L. Rivest, A. Shamir, and L.M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2): 120-126, February 1978.

[**RST**] Ronald L. Rivest, Adi Shamir, Yael Tauman: How to Leak a Secret. ASIACRYPT 2001: 552-565.

[**ST**] Marvin Sirbu, J. D. Tygar: NetBill: An Internet Commerce System Optimized for Network Delivered Services. COMPCON 1995: 20-25.