A Novel Fair GSR Contract Signing Protocol Against Earnest Money

Debajyoti Konar [†] **and Chandan Mazumdar**[††],

[†]Institute of Engineering & Management, Saltlake City,Kolkata-700 091, India
[††]Department of Computer Science & Engineering, Jadavpur University, Kolkata- 700 032, India

Summary
In this paper we propose a novel Fair Gradual Secrete Release (GSR) protocol for E-contract signing against earnest money between an originator and a responder involving their banks as transacting parties. We provide a security analysis of the protocol and analyze some important and pertinent properties of the protocol, which include money atomicity, validated contract and the fairness in true sense. The protocol involves originator, originator's bank, responder, responder's bank as transacting parties without using an additional trusted third party. We also formally prove the existence of all the said properties.

Key words:
Contract Signing, Fairness in true sense, Gradual Secrete Release, Money atomicity, Validated contract.

1. Introduction

Low cost and high efficiency drive the world towards E-commerce, E-business, E-governance etc. Services of E-business or E-commerce can be categorized in two classes, namely, E-contracting and E-trading. Signing a contract, where the parties are distributed geographically, is one of the major activities in today's commerce, business and governance. E-contracting is a major activity of E-commerce or E-business or E-governance. But the success of E-commerce (either E-contracting or E-trading) faces a major challenge of information security.

In this context, non-repudiation is a security service that creates, collects, validates and maintains the cryptographic evidences to support settlement of possible dispute among the transacting parties [22]. Non-repudiation services for transmission of messages are defined in terms of non-repudiation of origin (NRO) and non-repudiation of receipt (NRR).The service NRO is to protect the recipient from originator's repudiation regarding the origin of the message and NRR is to protect from the denial of the receipt of the message [5,7,11,15,16]. An important requirement in these non-repudiation protocols is fairness which assures that neither party can gain an advantage by quitting prematurely or otherwise misbehaving during a transaction [11,18,21,25,26]. The fairness of these fair exchange protocols which is in essence a non-repudiation protocol is the way that guarantees that either all the parties obtained what they want or none do [5].

Along with fairness, a Contract Signing protocol should ensure that no single participant of the protocol has enough scope to create or destroy the money (pay-order or payment token or digital draft etc.) within the scope of protocol. To ensure this, a fair e-contracting protocol should hold the *money atomicity* property. In E-commerce, while executing a Contract Signing protocol a responder requires to have a scope to verify that s/he is going to sign the valid contract for which s/he is going to pay the earnest money. This leads to a situation where the Contract Signing protocol requires the *validated contract* property. Unlike the E-trading protocol, a Contract Signing protocol has to reveal the identities of participating parties. Thus, E-contracting protocols should not allow anonymity of the parties. In the current scenario of E-commerce to provide mutual guarantees to the participants, Contract Signing protocols for E-contracting protocols are required to ensure *fairness*, *money atomicity* and *validated contract* properties. A straight forward approach for solving fair exchange problem used in several ISO proposal is to use a third party within the scope of protocol to ensure the fairness, which obviously can be named as Trusted Third Party (TTP) protocol [14,25,27,28,29]. This trusted third party (TTP) can be of two types, viz., Off-line TTP and On-line TTP. Besides the two-party fair exchange protocols, the multi-party fair exchange protocols also use an additional trusted third party to ensure the fairness [23,28]. The cost for subscribing the third party is a major issue in implementing the TTP fair exchange protocol in E-commerce. However there are several protocols where TTP is not being used and in these cases 'gradual release of secretes' is being used to achieve the fairness [2,4]. These protocols can be named as Gradual Secrete Release (GSR) protocols.

In this paper we briefly review some published works in section 2. Section 3 proposes a novel Fair GSR Contract Signing Protocol against Earnest Money by building the assumptions, defining the most required terms and providing symbols and notations. This protocol involves originator, originator's bank, responder and responder's bank as transacting parties. The proposed contract signing

Manuscript received November 5, 2007
Manuscript revised November 20, 2007

protocol does not use an additional trusted third party to achieve *fairness in true sense*, *validated contract* and *money atomicity* properties. Then we present a security analysis for confidentiality, integrity, authentication and non-repudiation services provided by the protocol in section 4. In section 5, we formally define the *validated contract, money atomicity* and *fairness in true sense* properties and prove the theorems on existence of these properties within the protocol.

2. Previous Work

In this section we discuss briefly some related work in providing fair-exchange in E-commerce, particularly in E-contracting. The idea of using a trusted third party in on-line mode to obtain non-repudiation of origin and delivery of an email message was proposed by Deng et al. [3] and Zhou and Gollmann [5]. In essence, these protocols are similar. In these protocols, the dispute resolution is outside the scope of the protocol. However, the protocols specify the evidences which are to be stored and the way of collection of these evidences for the dispute to be resolved in a fair manner. Franklin and Reiter [10] proposes a set of fair-exchange protocols that verify the consistency of a document before the exchange takes place. These protocols require a semi-trusted third party. A semi-trusted third party is one that can misbehave on its own but will not collude with any of the participating parties. But maintaining on-line third party makes the protocol costly in implementation and use. The protocol requires an active involvement of the semi-trusted third party for all scenarios and the information that principal parties are trying to exchange is never revealed to the third party. The protocols use a one-way function which has the additional property that there exists another efficiently computable function F such that F(x, (f(y)) =f(xy). The function, f, is known by both the parties, and F is known by the third party.

There are several fair exchange protocols that use third party in off-line mode, when it is required and hence they are optimistic fair exchange protocol. These protocols are designed either to sign a contract [14, 17] or to purchase a digital product [23,28,29]. An Optimistic Contract Signing Protocol [10] has been designed by Asokan, Shoup and Waidner to provide a service to Originator and Responder for obtaining each other's commitment on a previously agreed content. The protocol consists of three interdependent sub-protocols, viz., Exchange sub-protocol, Abort sub-protocol and Resolve sub-protocol. This asynchronous protocol, in essence, a fair exchange protocol involves three participating parties, viz., originator (O), Responder (R) and trusted third party (T). As it is a contract signing protocol, the protocol does not consider the anonymity property for any transacting party. An Abuse-free Optimistic Contract Signing Protocol [17] is designed by Garay, Jakobsson and MacKenzie involving the same role to guarantee abuse-freeness in addition to fairness and third party accountability. This protocol also consists of three interdependent sub-protocols, viz., Exchange sub-protocol, Abort sub-protocol and Resolve sub-protocol. The protocol relies on the cryptographic primitive called private contract signature. An Anonymous Fair Exchange E-commerce Protocol [23] by Ray and Ray uses customer, customer's bank, merchant, merchant's bank and an additional offline TTP as transacting parties to achieve fairness, correctness of the product and customer's anonymity properties. An Optimistic Anonymous Protocol with Validated Receipt *I. Ray et al* [28] also involves customer (C), merchant (M) and customer's bank (B), along with an additional offline TTP to achieve fairness and *validated receipt* properties. Both the protocols are for E-trading..

The GSR protocols have rather high communication requirements. On the other hand the cost of maintaining third party is nil, which makes these protocols cost effective in implementation for E-commerce. The GSR protocol presented by Blum [2] can be used in conjunction with digital signatures to sign contracts and send certified emails. This protocol provides a mechanism to exchange secrets between two parties. Even et al. [1] propose randomized protocols for signing contracts, certified mail and flipping a coin. The protocols use a notion of a 1-out-of-2 oblivious transfer protocol. The authors define a message to be a "recognizable secret message" if, although the receiver cannot compute the message, he/she can authenticate it once received. The 1-out-of-2 oblivious transfer protocol allows the sender to transfer exactly one secret out of two recognizable secrets. To motivate the participants to behave fairly in the transaction Sandholm and Lesser use game theory in their work [4]. The authors propose a contracting protocol, which is in essence a fair exchange protocol. To ensure fairness in contracting, the protocol allows any player to pay a penalty and withdraw from a contract during the execution. This game theoretic approach in the protocol assumes that all the participants behave rationally. In a technical report H. Pagnia and F.C. Gartner [19] showed that it is impossible to provide strong fair exchange between two parties without a trusted third party. In their model the notion of strong fairness and only two communicating parties have been used.

In this paper we propose a Fair GSR Contract Signing Protocol with Earnest Money. Within the scope of this proposed protocol the responder has to enter into the originator's website before signing the contract to have the details of the contract and then has to decide regarding the contract signing. The protocol engages an originator (O), a responder (R), originator's bank (OB) and responder's bank (RB) as participants. A significant contribution of

this paper is that the proposed protocol does not use an additional trusted third party to achieve *fairness in true sense,* a concept defined in section 5.3. The model involves multiple communicating parties and adopts the concept of pre signature and post signature. Theory of Cross Validation [23, 28] which assumes a RSA-like cryptographic scheme has been used for the development of the protocol. The protocol also achieves two other pertinent properties viz. *validated contract* and *money atomicity* properties.

3. The Fair GSR Contract Signing Protocol against Earnest Money

The proposed protocol has been designed to provide the e-commerce services for signing a contract against earnest money, with *fairness in true sense, validated contract* and *money atomicity* properties. The protocol does not use an additional trusted third party for which either originator or responder has to pay the subscription. At the outset, it is assumed that originator has full trust to his/her bank (OB) and vice-versa. This is also true for the responder (R) and his/her bank (RB). It is also assumed that the technical infrastructure is robust enough to cater the communication requirements for the proposed GSR Contract Signing Protocol. The protocol begins when the responder enters into the originator's website to have the details of the contract, like, draft of the contract agreement, the earnest money to be paid to sign the contract, etc. and being satisfied with this contract details decides to sign the original contract. The protocol adopts the concept of pre-signed and post- signed contract. The proposed Contract Signing protocol is based on the theory of cross-validation and provides the scheme to check the validity of the pre signed contract without decrypting it. Before presenting the protocol, we describe the symbols and notations, used in the protocol, define some important terminologies and also mention the assumptions made for the protocol.

3.1 Symbols & Notations

Symbols	**Interpretation**
B_{acct}	B's bank account
H( )	A collision-resistant one way hash function.
N_A	Random Number generated by A, used as authenticator
Sig_x (..)	The message signed by party X
c	The digital contract, already signed by originator and of the form (Sig_o [contract text, $H(N_O)$], N_O)
CI	Intimation to sign a contract
T_i	Transaction for signing a particular contract
A_{prv}, A_{pub}	A's private and public keys
A_{iprv}, A_{ipub}	A's private and public keys for T_i
A → B:X	A sends X to B
[X,K]	Encryption of X with key K
CC(X)	Cryptographic checksum of X
MTI	Money Transfer Instruction
ack	Acknowledgement of message (particularly payment)
pay-info	Payment information

3.2 Definitions

A Fair Contract Signing protocol: It is a exchange protocol to sign a contract agreement over internet in which contracting parties exchange items of value in such a manner that no party can gain an advantage over the others by misbehaving, misrepresenting or by prematurely aborting the protocol.

Money Transfer Instruction (MTI): An instruction issued by any transacting party of the protocol to his/her bank consisting the information regarding the amount to be transferred, the account which is to be debited and the account in which the amount is to be credited.

Intimation for contract (CI): It can be defined as a message containing the information regarding the particular contract, the responder intends to sign, the amount of the earnest money, and the identity of the responder.

Digital Demand Draft or Pay-order (P): In this protocol 'Digital Demand Draft or Pay-order' can be defined as a message consisting of the information regarding the amount and currency that is to be credited, the account in which the payment is to be credited and a nonce to prevent the replay.

3.3 Assumptions

The assumptions behind the protocol are as follows:

1. that the originator hosts his/her pre-signed contract (c), encrypted with a key (say K_1) along with all of its details, like, draft of the contract agreement, the earnest money to be paid to sign the contract, etc. in its own website so that the responders can download it.
2. that the responder has an account with the responder's bank and the originator has an account with the originator's bank. It is also assumed that the responder and originator have

full trust on their respective banks and vice-versa. Here the Banks are being used for financial transactions only.

3. that the signed contract and its execution are much more valuable to the originator than the earnest money.
4. that the scheme of encryptions is strong enough to provide the integrity of messages and signatures and it is same for all transacting parties.
5. that the technical infrastructure is strong enough to handle the communication requirements for the message exchanges in the protocol and is fail-safe to handle the log records corruption in any site.
6. that the mode of payment of earnest money, i.e., digital demand draft or pay-order is acceptable by the related banks.
7. that the key distribution scheme for the proposed protocol is secure.
8. that the identity of any party cannot be revealed only by the IP address.
9. that the fixed period for time-out is known to all concerned.
10. that each party keeps a copy of each message, s/he is sending.

3.4. Protocol Description

The protocol starts when for a particular transaction, say, T_i, the responder enters into the originators website to have the details of the contract and being satisfied with the draft of the contract decides to sign it. After that the protocol may be described as follows:

1) O → R: [c, K_1], O_{ipub}; */*R selects contract c from originator's website*/*
2) R → O: IC [CC(IC), R_{iprv}] [R_{ipub}, O_{ipub}]; */*R sends the intimation to O for signing the contract */*
3) O → R: [Abort, M_{iprv}]; */* O aborts*/*
Or
O →R:[CC(IC), O_{iprv}] [c.r, K_1xK_2] [CC([c.r, K_1xK_2]), O_{iprv}] [r, K_1] [CC([r, K_1]), O_{iprv}] [O_{acct}, OB_{pub}] [CC([O_{acct}, OB_{pub}]), O_{iprv}] [CC(R_{ipub}), O_{iprv}];
*/*Accepting the intimation for contract, O sends encrypted pre-signed contract and account information including responder's public key encrypted with his/her private key*/*
4) R → RB: [[MTI, R_{prv}], RB_{pub}]; */* R instructs RB to prepare pay-order and to send it to OB*/*
5) RB → OB: [[P,B_{cprv}],OB_{pub}]; */*RB sends the pay-order to OB*/*
Or
RB → R: [Failure, R_{pub}]; */*RB fails to send pay-order and informs R*/*
6) RB → R: [pay-info, RB_{prv}]; */*RB sends a copy of payment details to R*/*
7) R → O: [pay-info, R_{iprv}]; */* R forwards the copy of payment details to O*/*
Or
R→ O: [Abort, R_{iprv}]; */*R aborts if RB fails to send pay-order */*
8) OB → RB: [ack, OB_{prv}];
*/*OB sends acknowledgement of payment-clearance to RB*/*
9) RB → R: [[ack, OB_{prv}], RB_{prv}];
*/*RB forwards a copy of acknowledgement of payment-clearance to R*/*
10) OB → O: [ack, OB_{prv}];
*/*OB sends a copy of acknowledgement of payment-clearance to O*/*
11) O → R: [K_2^{-1}, R_{ipub}] [CC(K_2^{-1}), O_{iprv}] [r^{-1}, R_{ipub}] [CC(r^{-1}), O_{iprv}];
*/*O sends decryption key to R so that R can get the pre-signed contract*/*
12) R → O: [Sig_R [c, $H(N_R)$], O_{ipub}] [N_R ,O_{ipub}] [R_{acct}, RB_{pub}], [CC([R_{acct}, RB_{pub}]), R_{iprv}];
*/*R puts his/her signature and sends complete contract to O*/*
13) O → OB: [[MTI, O_{prv}], OB_{pub}]; */* O instructs OB to prepare pay-order for returning earnest money and to send it to RB*/*
14) OB → RB: [[P,B_{cprv}], OB_{pub}]; */*OB sends the pay-order to RB*/*
Or
OB → O: [Failure, O_{pub}]; */*OB fails to send pay-order and informs O*/*
15) OB → O: [pay-info, OB_{prv}]; */*OB sends a copy of payment details to O*/*
16) O → R: [pay-info, O_{iprv}]; */* O forwards the copy of payment details to R*/*
Or
O → OB: Retry: message 13
17) RB → OB: [ack, RB_{prv}];
*/*RB sends acknowledgement of payment-clearance to OB*/*
18) RB → R: [ack, RB_{prv}];
*/*RB forwards a copy of acknowledgement of payment-clearance to R*/*

In this protocol, first message exchange is to download ([c, K_1], O_{ipub}) from the website of the

originator (O).Then responder creates an intimation for signing the contract by including a nonce that forestalls a replay of the contract signing intimation. Message 2 allows the responder (R) to send the contract signing intimation for a desired contract to the originator (O) whereas by message 3, the originator (O) either aborts the transaction or responds by signing the contract signing intimation on acceptance and sending the encrypted (with cross-key K_1xK_2) contract and his /her account information, encrypted with his/her bank's (OB) public key. In this message the originator (O) also includes the responder's public-key under his/her private-key ($[CC(R_{ipub}), O_{iprv}]$). Thus, the customer can detect the man-in-middle attack. In message 4 responder (R) issues a money transfer (MTI) to his/her bank (RB) for paying the earnest money for signing contract mentioning the originator's account information (O_{acct}, OB_{pub}), exact amount to be paid to the originator's bank account. Through message 5, responder's bank (RB) sends the digital demand draft or pay-order (P) to originator's bank (OB) and sends copy of payment details to the responder (R) by message 6 or sends a failure message to the responder (R). Then the responder (R) forwards the copy of that payment details to originator (O) just to say that payment has been sent to the specified bank account. The responder (R) aborts the transaction, if from message 5 s/he receives a failure message through message 7. After clearance of the pay-order or demand draft, sent by responder's bank (RB), the originator's bank (OB) sends the acknowledgement against the payment of earnest money (credited to the originator's account) to the responder's bank (RB) and responder's bank (RB) forwards this acknowledgement to the responder (R) to inform that the payment has been credited to his/her account consecutively by messages 8 and 9. In message 10, originator's bank (OB) also sends the acknowledgement to the originator (O). As the originator (O) already knows that the specified payment has been credited to his/her account, s/he sends the decryption key, which will be used to get the pre-signed contracts (c), by the responder (R) through message 11.

Then by message 12, the responder (R) puts his/her signature on the pre-signed contracts (c) and sends that to the originator (O). In this message the responder (R) also sends his /her account information, encrypted with his/her bank's (OB) public key. The originator (O) instructs his/her bank (OB) to prepare pay-order for returning earnest money and to send it to responder's bank (RB) and accordingly sends the digital demand draft or pay-order to responder's bank (RB) or sends the failure message to originator (O) through message 13 and 14 respectively. Through message 15, the originator's bank (OB) sends the payment information to the originator (O) signing by own private key.

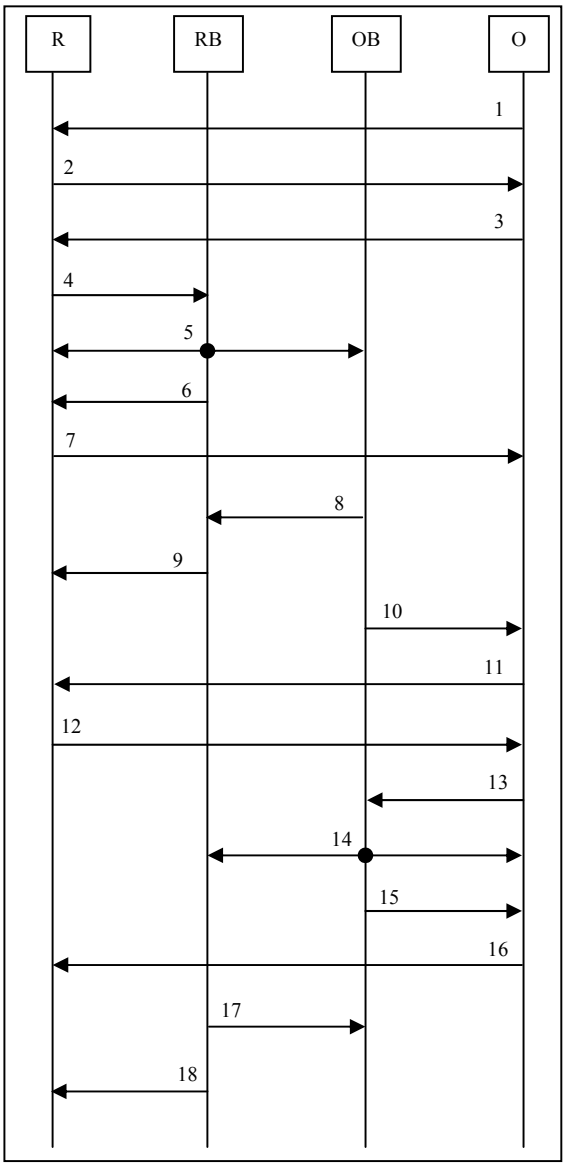

Fig. 1: Message Exchanges in the Protocol.

In message 16, the originator (O) forwards the same payment information to the responder (R) or retry message 13 if originator's bank (OB) fails to send pay-order in message 14. After crediting the payment in the responder's account (R_{acct}) the responder's bank (RB) sends the signed payment acknowledgement to the originator's bank (OB) and as well as to the account holder, i.e., the responder (R) by last two consecutive messages 17 and 18.

4. Security Analysis

Within the scope of protocol, we use the scheme of encryption to provide adequate security services. Here we discuss different aspects of the security mechanism of our protocol. In this protocol, through message 5, the responder's bank (RB) sends the digital draft or pay-order to originator's bank (OB) and sends copy of payment details to the responder (R) by message 6. After receiving the copy of payment details responder (R) sends the same to the originator (O) just to say that the digital draft or pay-order has been sent to the specified bank account through message 7. So, if any employee of the responder's bank (in essence, a malicious bank) creates the pay-order, it will go to originator's bank and be credited to originator's account. After getting the payment acknowledgement the originator will send the decryption key to the responder directly. So, for a malicious bank, it's not possible to have the decryption key. This provides security from a *malicious bank attack*.

This protocol also provides the *intruder detection* facility. The originator (O) either aborts the transaction or includes the responder's public-key encrypted by his/her private-key ([CC(R_{ipub}), O_{iprv}]), while sending message 3. Here, the originator (O) is signing the public key of the responder (R) with its own private key and sending it to the responder. If there is any intruder, the responder can not get its own public key R_{ipub} signed by the originator's private key (O_{iprv}). So the responder can detect the intruder.

In the scheme of encryption, if e is chosen small and a responder (R) can guess e, which leads to a security problem that a responder can avoid the payment of earnest money. Note that this attack is similar to the low exponent attack on the RSA cryptosystem [16]. This mode of attack requires the attacker to try all possible primes less than e. Also, this is an infeasible mode of attack when e is sufficiently large.

We also provide a mechanism using which the security will not be compromised even if the responder (R) can guess e correctly. The responder (R) downloads the pre-signed contract [c, K_1] from originator's website. The originator (O) chooses a random number r such that r is relatively prime to n_2. Rather than sending [c, K_1 x K_2] to the responder (R), the originator (O) sends the following: [c.r, K_1 x K_2], [r, K_1], where c.r is the product of c and r. To validate the contract, the responder (R) multiplies [c, K_1] with [r, K_1] and the resulting product is compared with [c.r, K_1 x K_2]. If both match, the responder (R) gets confidence about the contract for which he is going to pay the earnest money. Finally, instead of sending K_2^{-1}, the originator now sends K_2^{-1} and r^{-1} where r^{-1} is the multiplicative inverse of r modulo n_2. Using K_2^{-1}, the customer obtains c.r mod n_2. Multiplying this by r^{-1} the responder can retrieve c.

Within the scope of the protocol, the originator receives the contract signing intimation, forwarded copy of payment details originated by the responder and then collects the confirmation report for payment clearance signed by its own bank before sending the decryption keys. The originator also receives the signed contract from the responder. On the other hand, the responder collects encrypted pre-signed contract, originator's account information and the decryption key to decrypt the pre-signed contract form originator (O). The responder also receives the payment details sent to originator and the payment clearance report signed by its own bank (RB). The responder's bank sends the payment to the originator's bank only after getting the *Money Transfer Instruction* from the responder and the originator's bank also sends the payment for earnest money only after getting the *Money Transfer Instruction* from the originator. The above situation protects all the participant of the protocol from the repudiation regarding the origin of the message and also from the denial of the receipt of the message, which provides both the *NRO* and *NRR* services.

5. Analysis of the Properties

As per the design, our proposed Contract Signing protocol engages an originator (O), a responder (R), originator's bank (OB) and responder's bank (RB) as participants. Significantly the protocol does not use an additional trusted third party to provide *the fairness in true sense* without offering any advantage to either the originator or the responder. This proposed Contract Signing protocol also achieves two other pertinent properties viz. *validated contract* and *money atomicity* properties. Here we formally define the *validated contract, money atomicity* and *fairness in true sense* properties and prove the theorems on existence of these properties within the protocol.

5.1 Validated Contract:

Definition: Validated contract is a property of an E-contracting protocol to ensure that the pre-signed contract the responder is about to receive from an originator, is the same as that the responder intended to sign, before the responder pays the earnest money.

Theorem 5.1: *The Fair GSR Contract Signing Protocol against Earnest Money satisfies the' validated contract' property.*

Proof: As described in the protocol the responder initially downloads $[c,K_1]$ from the originator's website. Before issuing the money transfer instruction to his/her bank for paying the earnest money, the responder also receives a copy of encrypted pre-signed contract from the originator in the form of $[c.r, K_1 \times K_2]$, $[r,K_1]$, where c.r is the product of c and r. The responder multiplies $[c, K_1]$ with $[r, K_1]$ and the resulting product is compared with $[c.r, K_1 \times K_2]$. Theory of Cross Validation [23,28] states a theorem: for any two messages m , $m_c < n_1$, n_2, $[m, K_1 \times K_2] \equiv [m_c, K_1] \bmod n_1$ iff $m = m_c$ and $[m, K_1 \times K_2] \equiv [m_c, K_2] \bmod n_2$ iff $m = m_c$. If both match in the above said comparison, the responder is confident that the pre-signed contract s/he is about to receive from the originator, is the same as that s/he intended to sign, before paying the earnest money. Hence the protocol satisfies the validated contract property.

5.2. Money Atomicity

Definition: An E-commerce protocol satisfies the money atomicity property if money is neither created nor destroyed during the execution of the protocol.

Theorem 5.2: *Fair GSR Contract Signing Protocol against Earnest Money satisfies the 'money atomicity' property*.

Proof: We have to show that, within the scope protocol the digital draft or pay-order is neither created nor destroyed during the execution of the protocol. To do so, let us consider the contradiction, i.e., the digital draft or pay-order can be created or destroyed during the execution of the protocol.

To disprove this let us consider the following cases:

Case1: Let the pay-order can be created in two different ways, viz., using the same pay-order to get credited in the bank accounts of originator and responder for multiple times by them and using the same pay-order to pay or return the earnest money for more than one contracts by responder or originator respectively. Both the cases are the pay-order is being replayed. But as described in the protocol, a nonce value is used within the pay-order to forestall these replays. Also in the protocol, the pay-order is prepared by the responder's bank against the instruction of the responder to pay the earnest money and is being sent to the originator's bank for crediting the specified amount to the originator's account. On the other hand, another pay-order is prepared by the originator's bank against the instruction of the originator to return the earnest money and is being sent to the responder's bank for crediting the specified amount to the responder's account. The originator or responder receives only copy of payment details and the pay-order is directly exchanged by their banks. Neither the responder nor the originator gets the pay-order directly in their hand. Thus the pay-order can not be created within the scope protocol.

Case 2: Let the pay-order can be destroyed in two different ways, viz., not using the pay-order by the originator and responder to get credited in their accounts or by loosing the pay-order by them before getting it credited. But as described in the protocol, the responder instructs his/her bank to prepare the pay-order and send it to the originator's bank for crediting the specified amount to the originator's account. Also the originator instructs his/her bank to prepare the pay-order and send it to the responder's bank for crediting the specified amount to the responder's account. Both of them receive only copy of payment details from the other side and the pay-order is directly exchanged by their banks.

So, there is no scope that the pay-order can be destroyed. Thus the above two cases contradict that the pay-order can be created or destroyed during the execution of the protocol. Hence, by Involution Law of propositional logic, the GSR Fair Exchange protocol for E-contracting against earnest money satisfies the money atomicity property.

5.3. Fairness

Definition: An important property of E-commerce protocols is fairness with which neither party can gain an advantage by quitting prematurely or otherwise misbehaving during a transaction. In particular, to hold fairness in true sense an E-commerce protocol is required to ensure the following:

(a) One party is not able to deny to send the digital content what s/he supposed to send
(b) The other party is not able to deny the receipt of the digital content what s/he received
(c) Either party is able to have the correct digital content against his/her own digital content.

Theorem 5.3: *Fair GSR Contract Signing Protocol against Earnest Money satisfies the 'fairness in true sense' property*.

Proof: We have to show that, neither party, participating in the protocol can gain an advantage by misbehaving during a transaction. Let us consider the contradiction, i.e. some parties can gain advantages within the scope of protocol.

To disprove this let us consider the following cases:

Case1: Let the originator misbehaves by denying the receipt of earnest money and its return or by disappearing after receiving the earnest money or by sending an incorrect decryption key after receiving the earnest money. But, in this protocol the responder (R) is getting the information from his/her bank that the exact payment has

been sent to the originator's account (O_{acct}) through message 6. Again, by message 9 s/he (R) is getting signed copy of the acknowledgement from his/her bank (RB) regarding the encashment of the payment into originator's account (O_{acct}), signed by originator's bank (OB), which responder's bank (RB) is getting from originator's bank (OB) through message 8. So the responder (R) have two important documents, viz, [[ack, OB_{prv}], RB_{prv}] & [pay-info, RB_{pub}], which can legally prove that s/he has done the payment to originator's account (O_{acct}) in originator's bank. Moreover, after getting the contract signed by the responder, if the originator denies the return of the earnest money then also responder (R) is legally strong enough with the documents [[ack, OB_{prv}], RB_{prv}] & [pay-info, RB_{pub}] and [Sig_R [c, $H(N_R)$], O_{ipub}] [N_R ,O_{ipub}]. Also as assumed in assumption 3, getting the signed contract and its execution are extremely valuable to the originator than the earnest money. These facts lead to a situation where originator (O) is not in an advantage such that s/he can deny the receipt of earnest money and its return or by disappearing after receiving the earnest money or by sending an incorrect decryption key after receiving the earnest money.

Case2: Let the responder intends to sign the contract and misbehaves by denying the payment of earnest money. But, as described in the protocol the responder (R) is issuing the money transaction instruction (MTI) to his/her bank (RB) and the bank is sending the payment to originator's bank, not to the responder. The responder receives only the copy of a payment details form his/her bank. So, if the responder intends to sign the contract s/he has to get the actual pre-signed contract from originator and for that s/he has to instruct his/her bank to pay and the payment has to be credited in originator's account in originator's bank directly. These facts show that it is not possible to deny the payment by the responder if s/he intends to sign the contract.

Case 3: Let the responder does not receive the correct pre-signed contract but the originator gets the correct payment of earnest money. But, as described in the protocol the responder initially downloads [c, K_1] from the originator's website. Before paying the earnest money for the contract, the responder also receives a copy of encrypted pre-signed contract from the originator in the form of [c.r, K_1 x K_2], [r,K_1], where c.r is the product of c and r. The responder multiplies [c, K_1] with [r,K_1] and the resulting product is compared with [c.r, K_1 x K_2]. If both a match, then only the responder instructs his/her bank to prepare the pay-order and send it to originator's account in originator's bank. Thus within the scope of this protocol this is not possible that the originator gets the correct payment of earnest money but responder does not receive the correct pre-signed contract.

Case 4: Let the originator does not receive the correct payment of earnest money but the responder gets the correct pre-signed contract. This is only possible if the protocol allows the responder to receive the pre-signed contract before paying the earnest money. But, in this protocol, originator sends the pre-signed contract in encrypted form through message exchange 3. To have the actual pre-signed contract the customer must have the decryption key, which is provided by the merchant by the message exchange 11. In between the responder instructs his/her bank to prepare the pay-order and send it to originator's account (O_{acct}) in originator's bank (OB). Then the responder's bank (RB) sends the pay-order directly to the originator's account. After having an acknowledgement that the exact payment has been credited to his/her account the originator sends the decryption key to the responder by message exchange 11. This shows that in this protocol it is not possible that the originator does not receive the correct payment of earnest money but the responder gets the correct pre-signed contract.

Thus the above four cases contradicts that some parties can gain advantages within the scope of protocol. Hence, by Involution Law of propositional logic, the GSR Fair Exchange protocol for E-contracting against earnest money satisfies the fairness property.

6. Conclusion

In the current scenario of E-commerce fair exchange is one of the pertinent issues and it is to be addressed by all type of E-commerce protocol, whether it is an E-trading protocol or E-contracting protocol. Along with the fairness, one of the important objectives contract signing protocols against earnest money is to protect the validity of the contract by ensuring that the pre-signed contract the responder is about to receive from an originator, is the same as that the responder intended to sign, before the responder pays the earnest money. Hence, in this E-commerce scenario also needs an contract signing protocols against earnest money that holds the *validated contract* property. An E-commerce protocol should also ensure that no single participant of the protocol has enough scope to create or destroy the money within the scope of protocol by holding the *money atomicity* property. Majority of the protocols proposed in the literature rely on trusted third party to provide the *fairness*. Whether the protocol uses offline TTP or online TTP, the cost to maintain the trusted third party is a major concern in its implementation.

Keeping these in our mind, in this paper we proposed a GSR Contract Signing Protocol against Earnest Money. The protocol engages an originator (O), a responder (R), originator's bank (OB) and responder's bank (RB) as participants. Our proposed protocol does not use an additional trusted third party to achieve *fairness in true sense*. The properties of the protocol also include *money atomicity* and *validated contract*. Here, we provided a detailed security analysis for confidentiality, integrity, authentication and non repudiation services provided by the protocol. We also formally defined the *validated contract, money atomicity* and *fairness in true sense* properties and proved the theorems on existence of these properties within the protocol.

We plan to check the feasibility of operation of this protocol in conjunction with other protocols. We also plan to study the performance of the protocol by applying different load of transaction, which will help to optimize the protocol.

We believe our work in this paper will extend the area of applicability of Fair Exchange protocol in E-commerce and strengthen the GSR approach to develop the Fair Exchange protocol so that transacting parties can participate in such transaction with more assurance, while stationed in geographically distributed locations.

References

[1] S. Even, O. Goldreich, A. Lempel, "A randomized protocol for signing contracts", Communications of the ACM 28 (6) 1985 (June) 637-647.

[2] M.Bulm "How to exchange (secrete) keys", ACM Transactions on Computer Systems 1 (1993), pp. 175-193.

[3] R.H. Deng, L. Gong, A.A. Lazar, W. Wang, "Practical protocols for certified electronic mail", Journal of Network and System Management, Vol. 4 (3), (1996).

[4] T.W. Sandholm, V.R. Lesser, "Advantages of a leveled commitment contracting protocol", Proc. Of 13th National Conference on Artificial Intelligence, Portland or The MIT Press, Massachusetts, 1996, pp. 126-133.

[5] Jianying Zhou and Dieter Gollmann. "A Fair Non-repudiation Protocol". Proceedings of 1996 IEEE Symposium on Security and Privacy, pages 55--61, Oakland, USA, May 1996.

[6] Jianying Zhou and Dieter Gollmann. "Observations on Non-repudiation". Lecture Notes in Computer Science 1163, Proceedings of Asiacrypt'96, pages 133--144, Kyongju, Korea, November 1996.

[7] Ning Zhang and Qi Shi. "Achieving Non-repudiation of Receipt". The Computer Journal, 39(10):844--853, 1996.

[8] S. H. Low, N. F. Maxemchuk and S. Paul. "Anonymous Credit Cards and Their Collusion Analysis". IEEE/ACM Transaction on Networking, 4(6), Dec. 1996.

[9] ISO/IEC 10181-4. "Information technology - Open systems interconnection - Security frameworks in open systems - Part 4: Non-repudiation framework". ISO/IEC, 1996.

[10] M.K. Franklin, M.K. Reiter, "Fair exchange with a semi-trusted third party", Proceedings of the 4th ACM Conference on Computer and Communications Security, Zurich, Switzerland, Association for Computing Machinery, New York, 1997 (April), pp. 1-6.

[11] N. Asokan, Matthias Schunter, and Michael Waidner. "Optimistic Protocols for Fair Exchange". Proceedings of 4th ACM Conference on Computer and Communications Security, pages 7--17, Zurich, Switzerland, April 1997.

[12] J. Zhou and Dieter Gollmann. "Evidence and Non-repudiation". Journal of Network and Computer Applications, 20(3):267--281, July 1997.

[13] ISO/IEC 13888-1. "Information technology - Security techniques - Non-repudiation - Part 1: General". ISO/IEC, 1997.

[14] N. Asokan, Victor Shoup, and Michael Waidner. "Asynchronous Protocols for Optimistic Fair Exchange". Proceedings of 1998 IEEE Symposium on Security and Privacy, pages 86--99, Oakland, USA, May 1998.

[15] Steve Schneider, "Formal Analysis of a Non-repudiation Protocol" Proceedings of 11th IEEE Computer Security Foundations Workshop, pages 54--65, Rockport, USA, June 1998.

[16] J. Zhou and Dieter Gollmann. "Towards Verification of Non-repudiation Protocols". Proceedings of 1998 International Refinement Workshop and Formal Methods Pacific, pages 370--380, Canberra, Australia, September 1998.

[17] J.A. Garay, Mjackobsson and P. MacKenzie, "Abuse-free Optimistic Contract Signing", Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology, Springer-Verlag, 1999, pp. 449-466.

[18] Kwangio Kim, Sangjoon Park, and Joonsang Baek. "Improving Fairness and Privacy of Zhou-Gollmann's Fair Non-repudiation Protocol". Proceedings of 1999 ICPP Workshops on Security, pages 140--145, Aizu, Japan, September 1999.

[19] Pagnia and Gartner, "On the Impossibility of Fair Exchange without a Trusted Third Party" Department of Computer Science, Darmstadt University of Technology, Technical Report TUD-BS-1999-02.

[20] Jianying. Zhou, Robert Deng, and Feng Bao. "Some Remarks on a Fair Exchange Protocol". Lecture Notes in Computer Science 1751, Proceedings of 2000 International Workshop on Practice and Theory in Public Key Cryptography, pages 46--57, Melbourne, Australia, January, 2000.

[21] Els Van Herreweghen. "Non-repudiation in SET: Open Issues". Lecture Notes in Computer Science 1962, Proceedings of 2000 Financial Cryptography, pages 140--156, Anguilla BWI, February 2000.

[22] Jianying Zhou. "Non-repudiation in Electronic Commerce". ISBN 1-58053-247-0, Computer Security Series, Artech House, 2001.
[23] Indrakshi Ray and Indrajit Ray. “An Anonymous Fair-Exchange E-Commerce Protocol.” Proceedings of the First International Workshop on Internet Computing and E-Commerce, San Francisco, CA, April, 2001.
[24] Nicolas Gonzalez-Deleito and Olivier Markowitch. "An Optimistic Multi-party Fair Exchange Protocol with Reduced Trust Requirements". Lecture Notes in Computer Science 2288, Proceedings of 4th International Conference on Information Security and Cryptology, pages 256--267, Seoul, Korea, December 2001.
[25] Jianying Zhou. "Achieving Fair Non-repudiation in Electronic Transactions". Journal of Organizational Computing and Electronic Commerce, 11(4):253--267, December 2001.
[26] Feng Bao, Guilin Wang, Jianying Zhou, and Huafei Zhu. "Analysis and Improvement of Micali's Fair Contract Signing Protocol". Lecture Notes in Computer Science 3108, Proceedings of 2004 Australasian Conference on Information Security and Privacy, pages 176--187, Sydney, Australia, July 2004.
[27] Q. Zhang, K. Mayes and K. Markantonakis. "APractical E-Payment Protocol To Realize Fair-Exchange". Proceedings of Workshop on Information Security Application, Jeju Island, Korea, August 23-24, 2004.
[28] Indrajit Ray, Indrakshi Ray, and N. Natarajan. "An Anonymous and Faliure Resilient Fair-exchange E-commerce Protocol". Decision Support Systems, 39(2005):267-292, 2005.
[29] Yusuke Okada, Y. Manabe and T. Okamoto. "Optimistic Fair Exchange Protocol for E-Commerce". Proceedings of Symposium on Cryptographic and Information Security, SCIS 2006 , Hiroshima, Japan, January 17-20, 2006.

Debajyoti Konar M.Sc. (Mathematics) from Jadavpur University, Kolkata, India in 1996. Author is teaching in Institute of Engineering & Management as Assistant Professor and has 8 years teaching experience. The author also worked in IT industry for 3 years. Major area of Author’s research interest is Information Security. Author is also interested in the field of Mathematical Modeling. Author has two international published paper and three more have been communicated.

Prof. Chandan Mazumdar is teaching Computer Science & Engineering in Jadavpur University for over 21 years. He is the Coordinator of the Centre for Distributed Computing. He has supervised five Ph.D.s. He has to his credit over 52 publications in National and International Journals and Conferences. He has copyrighted two software – one, on Forensic Identification of Fire Arms and another on Enterprise Information Security Management. He was the Program Co-chair of the First International Conference on Information Systems Security, 2005 and General Co-chair of the same conference in 2006. His present research interests are Information Security, Disaster Management Information System and GIS.