SECURE COMMUNICATION PROTOCOL FOR PRESERVING E-TENDERING INTEGRITY

Rong Du, Ernest Foo, Colin Boyd, Brian Fitzgerald

ISRC Queensland University of Technology

r.du@qut.edu.au, e.foo@qut.edu.au,boyd@isrc.qut.edu.au,bf.fitzgerald@qut.edu.au

ABSTRACT

This paper presents a secure communication protocol which can be used as the framework for an e-tendering scheme. This protocol is focused on securing the integrity of tendering documents and ensuring that a secure record of document generation is kept. Our protocol provides a mechanism to manage e-tendering contract evidence as a legal record in a unique and effective manner. It is the starting point of reliable record keeping. To a certain extent, it also addresses existing security problems in the traditional tendering processes.

Key Words: 9 E-business, 49 Information Security, 18 Information Technology

1 INTRODUCTION

Tendering is a process for entering into a sales contract (Thorpe and Bailey, 1996). Legally enforceable contract evidence is the primary means of assessing both the validity of a contract and fair trading practice. Traditionally, tendering is conducted using paper based documents. Many organisations are looking to use electronic documents for tendering to increase the efficiency and ease-of-use of the process.

A contract is formed through interactive communication in the tendering process, in which offers, acceptances and intention of entering contract are sent to the other contracting party for consideration. Because of its legal significance, electronic communication and its record keeping become primary targets for attackers. Illegal activities form threats that could significantly compromise the legitimacy of the contracting process in e-tendering. Identified common threats are masquerade, eavesdropping, time and signature repudiation, and integrity violation (Du et al., 2004). Protocol related attacks are also threats to the contracting process and may result in unfair trading.

For example, the deliberate repudiation of digital signatures or timestamps could result in contracting evidence being deemed invalid and business liabilities unenforceable. Integrity violation could also invalidate the contract. People would be unable to distinguish liabilities and obligations, therefore unable to enforce justice. We believe that securing electronic communication environment is the key issue for a secure e-tendering system.

Considering the legal significance and existing common threats, we need to ensure that the communication is conducted within legal frameworks, and the communicated contracting evidence is managed as a legal record with special consideration to its integrity.

This digital legal record has to be publicly verifiable at a later time to demonstrate that it contains the complete and uncorrupted set of evidence of the e-tendering process.

Without proper protection, business faces potential threats and legal costs for a non-legally compliant e-tendering system. It also faces potential financial damages caused by computer criminal activities such as identity theft, hacking, breaches of information privacy or user abuse of the system.

Therefore in this paper, we present a secure communication protocol as the security framework (foundation) for our first layer of a secure e-tendering scheme. This protocol is focused on securing the communication related to the contracting process in e-tendering. Our protocol considers how to (1) ensure that an e-contract communication or negotiation is conducted within a legal framework, (2) preserve sequentially generated e-contract evidence integrity effectively by using hash chain technology.

In section 2 we discuss the tendering scenario and hash chain applications. In section 3 we discuss the specific security requirements of communications in e-tendering. In section 4 we discuss the proposed protocol to be used in the contracting process in the e-tendering scheme. In section 5 we analyze the security of the protocol against the specified requirements.

2 BACKGROUND

2.1 E-tendering Scenario

As a special contracting process, the traditional tendering procedure is designed to facilitate contracting parties forming a valid contract as an outcome. In general, a principal initiates a tendering process to select the main contractor for a project. The principal invites a small group of qualified potential contractors (tenderers) to tender for the project. The contracting process in tendering practice starts when the principal generates a Tender Document (TD) or Request for Tender (RFT) and sends it to all potential tenderers with an invitation for tender. The subsequent stages involved are Tender Document clarification, tender preparation, tender submission, tender assessment, post tender negotiation and signing of the final contract. The communication for contracting could take any such form: an invitation, offer, clarification, amendment, acceptance or signing contract.

After potential tenderers receive the Tender Document and the formal invitation from the principal, they can each send anonymous requests to the principal for clarification of the Tender Document. The request and clarification or amendment are distributed among all tenderers. Tenderers will prepare their tender according to the guidelines or requirements from the Tender Document. After tenderers submit their tenders, the principal will undertake assessment to choose the first preferred tender for post tender negotiation. A successful negotiation will lead to awarding the contract and signing the final contract.

The contracting legal framework discussed by Christensen (Christensen, 2001) has indicated that not only the communicated contracting evidence needs to be kept in the form it was when first generated, but also in correct order to reflect the sequence of offers, acceptances, and counter offers. Considering business and legal requirements, hash chain algorithm will be discussed as the most suitable technology to be integrated into the electronic communication for preserving e-tendering integrity.

2.2 Related Technical Applications

The contracting process in e-tendering is a sequential process. Chaining contract negotiations and agreed terms have been identified as one of the advanced essential security services for an effective legal record management in a reliable e-tendering process (Du et al., 2004). We believe that the hash chain is the most suitable technique for providing this type of integrity service. The hash chain is constructed with a one way hash function. One way hash function means technically it is impossible to derive the original text from the string generated by this one way hash function.

In a general form hash chain can be represented as follows:

$L_0, L_1, L_2, ..., L_n,$
where $L_i = h(L_{i-1})$, for $i = 1, 2, ..., n,$
$h()$ is the underlying one-way hash function,
L_n is the individual hash chain node.

The generation of the current node requires the input from the previous node. Each node recursively depends on its previous node. This dependency forms a strong security property to protect the order of events generated through sequential processes such as contract negotiations in e-tendering. The difficulty in altering a node of this hash chain without affecting its subsequent nodes is equivalent to the difficulty in finding a preimage of the one way hash. In other words, the security strength of the hash chain depends on the one way hash function.

However, without effective integrity protection of a hash chain, it cannot be a reliable check sum to provide integrity confirmation service for the original information. Some applications such as time-stamp (Haber and Stornetta, 1991), payment systems (Anderson et al., 1997; Rivest and Shamir, 1996), audit log system (Kelsey and Schneier, 1999), and mobile agent (Karjoth, 1998) have demonstrated that a hash chain can be used with other algorithms (eg. digital signature) to provide desired security services for a particular business situation. Each application of hash chains has a different chain forming process according to the business process it is emulating. The chain protecting algorithms (normally signature schemes) are also constructed in a different way to protect against particular attacks related to the business process.

Among all these hash chain applications, the time-stamp scheme by Haber and Stornetta (Haber and Stornetta, 1991) emulates the closest business process with contracting process in e-tendering. They have combined signature and one way hash techniques to prevent the time-stamping service from issuing a fake time stamp. In their scenario, a trusted third party provides a time-stamping service (notary system). Their basic idea is to link a current time-stamping request (hashed message) with a previous request by using one way hash functions to form a linking node. This node is also signed by the time-stamp service and sent to each requester.

Their scheme is designed for a random client to make a request to a trusted third party which provides the time-stamping service. In e-tendering, communications are complex and every contract related communication bears evidential weight. It is a very expensive exercise for the contracting parties to make time-stamping requests on every one of their communications.

For example, the principal and all potential tenderers could use one of this type of time-stamp service through their possibly half-year long contracting process. Their contracting evidence would be mixed with all other requests. The verifying process would

involve all intermediate parties cooperation to confirm the integrity of their evidence on the chain. It would be very difficult to reconstruct the e-tendering contracting evidence set when a dispute occurs.

A complete adoption of time-stamp scheme faces expensive interaction with a third party. Additionally it needs complicated verifying algorithms to reconstruct legal evidence for a complex contracting process in e-tendering.

We will concentrate on simulating traditional contracting procedures to avoid exposing our protocol to traditional attacks.

3 SECURITY PROPERTIES

In an abstract form the following security properties need to be achieved to ensure that contracting and business requirements are properly considered for a secure communication protocol in an e-tendering scheme. The achievement of these security properties can also eliminate corresponding attacks.

3.1 Confidentiality

Confidentiality means only the e-contracting parties are privy to the information related to the contract negotiation, and is needed to ensure their privacy. Providing a confidentiality service is a general requirement throughout the whole tendering practice. It is to prevent unauthorized release of the tenderer's tender strategy, design and other private information. This type of information leak could lead to a criminal offense in trading practice. This property eliminates any opportunity for an eavesdropper to gain access to any confidential information communicated between contracting parties.

3.2 Data Origin Authentication

Data origin authentication means that the party receiving a message can confirm the identity of the party originating the message in a contracting process. This property also implicitly provides an assurance of message integrity (Menezes et al., 1997). The data origin authentication ensures that the e-contracting evidence (the proof of a valid and binding contract) is trustworthy when it is first generated. It involves binding the message originating party to the message to prevent repudiation, message replay, and impersonation attacks. It also provides confirmation of the message's original integrity.

The validity of an e-contract requires that it demonstrate that it has been formed within the legal framework. As indicated in a Crown Law report by Ann Fitzgerald, one party must have made an offer, that offer must have been accepted by another party in unequivocal terms, and that acceptance must be communicated to the first party. Both parties are required to have the intention to create this legal relationship under contract law principles.

Traditionally, a valid and binding contract evidence lies on the assumption that parties can visually identify each other through the signing process. In contrast, this assumption does not exist while people communicate through the Internet. Therefore, this property is required to be in place to provide a functional equivalent practice in the digital world. It is aimed against masquerading attacks, and eliminates signature repudiation.

3.3 Original Integrity Confirmation

Original integrity confirmation means that original contract evidence integrity can be confirmed and any alteration can be detected. This service is to ensure that the original integrity of information related to a contract is effectively preserved and this integrity is verifiable at any given time. Fitzgerald's report further indicates that collected contract evidence has to meet two key evidentiary requirements. These requirements can be stated thus: that a business have a systematic approach to record keeping, and an adequate security infrastructure for its electronic records. In abstract form, we can say that preserving the original integrity is the fundamental evidentiary requirement for electronic contract evidence.

Contracting evidence is generated chronologically through parties' communication. The original integrity confirmation property is used to prove whether or not the contracting evidence can be used in a chronological reconstruction of the original e-contracting negotiations. This property requires the scheme to provide a security service to prevent the contracting evidence from being deleted, altered or re-sequenced after generation, and to prevent the insertion of fake evidence. The achievement of this security property can also create a barrier for deliberate time and signature repudiations.

3.4 System Reliability

A new protocol is a unique set of cryptographic procedures for a special business process. It is constructed with cryptographic primitives or building blocks to provide conventional security services as discussed above. There are always unavoidable protocol disruptions related to the business process, which the protocol is simulating. These disruptions can be protocol attacks, or simply that business itself can not proceed.

The protocol disruption protection will discuss alternative solutions for identified protocol disruptions. There are two types of disruptions in the contracting process (1) normal termination (terminate negotiation) from one party; (2) abnormal behavior from either one or both parties. This property ensures that normal disputes can be easily resolved and that one party cannot use the protocol algorithm to gain advantage over the other party.

A protected chain forming process is still open to some protocol attacks, such that both parties can manipulate the last node of the chain. In this situation, trust is an issue when parties attempt to use dishonest protocol operation to gain an advantage over the other party.

4 SECURE ELECTRONIC COMMUNICATION PROTOCOL FOR E-TENDERING SCHEME

This section is organized into four subsections to discuss notation, system setup and the cryptographic scheme. All contracting stages require some special security properties and consideration other than the underlying communication security properties. The following scheme is the framework for the e-tendering process, and only considers providing the security service for secure e-contracting communications and its related evidence.

4.1 Notation

All the notations used in this paper are listed in Table 1.

Table 1: Notation

SYMBOL	NAME
$\longrightarrow$	One party sends another party a message eg. $B \longrightarrow A$ B send A
$\|$	Concatenation
K_{ID}^{-1}	Private key of party ID eg. K_B^{-1} B's private key
K_{ID}	Public key of party ID
C_{ID}	Certificate of party ID
h	One way hash function
Sig_x	Signature generation function, x represents input key
V_x	Signature verifying function, x represents input key
E_x	Asymmetrical encryption function, x represents input key
D_x	Asymmetrical decryption function, x represents input key
m_n	nth communicated message of a contract negotiation
TD	Tender Document by principal
HC_{ID}	One party's entire hash chain or TTP roll backed hash chain
DS	Start of dispute
TM	Start of termination
RTM	Respond of the termination
m_f	Final contract signed by both parties
m_{cf}	Confirmation message generated by TTP
m_{er}	Error message generated by TTP

4.2 System Setup

Parties involved in the e-contracting are the principal(A), tenderers group ($\mathcal{B}$) or tenderer (B), and trusted third party TTP. All parties have their private, public key pairs and certificates for their public keys. All private keys are secure and have not been compromised. We assume that the local area Public Key Infrastructure is in place for tendering process. In order to improve the efficiency, DHIES will be used when large messages need to be transfered.

All the communication methods used in contracting (phone, email, web submission...) are logged.

TD is the Tender Document. L_0 is always generated by the principal A with TD.

The one way hash function is to be SHA-1 or equivalently well established one way hash function. The asymmetrical encryption refers to the public key systems (Diffie and Hellman, 1976) either based on the problems of computing logarithms over finite fields, factoring large integers, or elliptic curve. The signature scheme is referred to relatively standard schemes such as ElGamal (ElGamal, 1985), DSA or RSA (Rivest et al., 1978) since their security have been widely discussed in the past. If either party's key pair is compromised or revoked during the contract negotiation period (an exceptional condition), all the hash chain nodes have to be re-signed by both parties.

4.3 Cryptographic Protocol of Secure Communication for E-tendering Scheme

The protocol contains seven sub-protocols showed in Figure 1. The Initial Stage, Negotiation Stage, and Final Stage Sub-Protocols are main communication protocols for e-tendering. The Dispute I, Dispute II, Termination and Engage TTP are error control

sub-protocols. Their relationships are also shown in Figure 1. The arrows in the Figure 1 represent directions of connectivities.

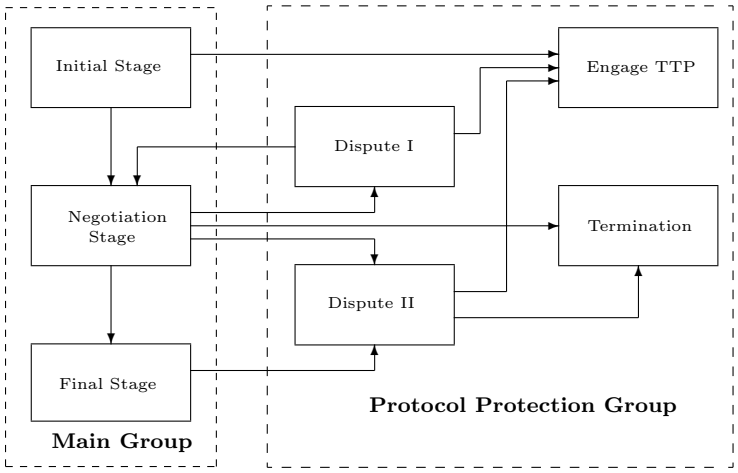

Figure 1: Protocol Diagram

During the e-tendering process, the principal will start the Initial Stage communication protocol and the parties will proceed with the communication using the Negotiation Stage protocol. When the final contract is formed, both parties will enter the Final Stage protocol to close their e-tendering communication with the trusted third party as witness.

If there is a dispute over the hash chain integrity, any party can connect to the dispute sub-protocols. If the responding party agrees with the hash chain integrity from the initiating party, these parties will enter into the Dispute I Sub-Protocol. From the Dispute I Sub-Protocol, the communicating parties can switch back to the Negotiation Stage Sub-Protocol or connect to the Engage TTP Sub-Protocol depending on the parties responses. If the responding party disagrees with the hash chain integrity from the initiating party, these parties will enter into the Dispute II Sub-Protocol. From the Dispute II Sub-Protocol, the parties can formally terminate the communication with the Termination Sub-Protocol or the Engage TTP Sub-Protocol.

The Termination Sub-Protocol can be connected to through the Negotiation Stage directly, to stop the parties communication formally with the TTP, before the Final Stage. This modular architecture is designed such that it can link to other security modules in future work.

4.3.1 Initial Stage

The Initial Stage is for the principal to initialise the record keeping hash chain by generating the root node. The principal first creates the Tender Document. This document includes a project description, tender specification, and weighting system. Principal A calculates the root node L_0 at the start of the contracting process with its Tender Document as the first message, and makes it available to the potential tenderers group $\mathcal{B}$. This stage's protocol is listed in Figure 2.

The principal uses TD as the input of h hash function to generate L_0 as the first step. It then uses its private key K_A^{-1} and signature function Sig to produce its signature σ_A over root node L_0. The principal also encrypts the $(TD \| \sigma_A)$ using the public key K_B

A		$\mathcal{B}$
$L_0 = h(TD)$		
$\sigma_A = Sig_{K_A^{-1}}(L_0)$		
$M_0 = E_{K_B}(TD \| \sigma_A) \| C_A$	$\xrightarrow{M_0}$	

Figure 2: Initial Stage Sub-Protocol

and concatenates C_A of a tenderer to produce the M_0 for sending to a tenderer over an insecure network.

4.3.2 Negotiation Stage

The Negotiation Stage handles all the communicated messages after the tenderers receive the Tender Document to the point when the final contract is formed. The messages can be of any type: tender clarification, tender submission, tender assessment result, post tender negotiations and signed final contract.

For example, during the post tender negotiation stage the process of nth negotiation message from B to A is listed in Figure 3.

A		B
		$L_n = h(m_n \| L_{n-1})$
		$\sigma_{nB} = Sig_{K_B^{-1}}(L_n)$
	$\xleftarrow{M_n}$	$M_n = E_{K_A}(m_n \| \sigma_{nB}) \| C_B$
$(m_n \| \sigma_{nB}) = D_{K_A^{-1}}(M_n)$		
$L_n = V_{K_B}(\sigma_{nB})$		
$L'_n = h(m_n \| L_{n-1})$		
if $L'_n = L_n$		
$\sigma_{nA} = Sig_{K_A^{-1}}(L_n)$		
$RSP_n = E_{K_B}(\sigma_{nA}) \| C_A$	$\xrightarrow{RSP_n}$	
if $L'_n \neq L_n$		
do not send RSP trigger M_n resend		

Figure 3: Negotiation Stage Sub-Protocol

In this stage, the sender B concatenates $m_n \| L_{n-1}$ as input of h to generate the nth node L_n. B then signs the node with its private key K_B^{-1} to produce its signature σ_{nB} over L_n. It also binds its commitment to the message.

B encrypts $m_n \| \sigma_{nB}$ with A's public key K_A, concatenates C_B and sends the encrypted message M_n to A.

On receiving the message M_n, A decrypts the message M_n with its private key K_A^{-1}, extracts the nth node L_n using B's public key K_B, and checks B's certificate C_B. A calculates the nth node L'_n itself by concatenating m_n and the previous node L_{n-1} as the input of the one way hash function h. At this stage A can compare whether $L'_n = L_n$ to confirm the messages integrity. If $L'_n = L_n$, A will sign the L_n and send RSP to B. If $L'_n \neq L_n$, A will not send RSP. It will trigger B to re-send M_n to A. Both parties also have the option to start Protocol Disruption Protection Protocol.

4.3.3 Final Stage

When the final node L_f is generated after both parties have signed the final formal contract, they need to send the final node to the TTP for long term preservation of the contracting evidence. The TTP normally will have a more secure environment than the communicating parties. The TTP can have longer lived keys, less vulnerabilities in the system, and better facilities for key updates and upgrades to resign the documents.

This stage is to confirm that both parties have agreed on the final contract. The third party is the witness. This stage's protocol is listed in Figure 4.

A	TTP	B
$\sigma_{fA} = Sig_{K_A^{-1}}(L_f)$		$\sigma_{fB} = Sig_{K_B^{-1}}(L_f)$
$M_{fA} = E_{K_{TTP}}(\sigma_{fA} \| B) \| C_A$		$M_{fB} = E_{K_{TTP}}(\sigma_{fB} \| A) \| C_B$
$\xrightarrow{M_{fA}}$		$\xleftarrow{M_{fB}}$
	$L'_f = V_{K_A}(\sigma_{fA})$	
	$L''_f = V_{K_B}(\sigma_{fB})$	
	if $L'_f = L''_f$	
	$\sigma_{TTP} = Sig_{K_{TTP}^{-1}}(\sigma_{fA} \| \sigma_{fB} \| m_{cf})$	
	$M_{fTTPA} = E_{K_A}(\sigma_{TTP}) \| C_{TTP}$	
	$M_{fTTPB} = E_{K_B}(\sigma_{TTP}) \| C_{TTP}$	
$\xleftarrow{M_{fTTPA}}$		$\xrightarrow{M_{fTTPB}}$
	if $L'_f \neq L''_f$	
	$\sigma_{TTP} = Sig_{K_{TTP}^{-1}}(\sigma_{fA} \| \sigma_{fB} \| m_{er})$	
	$M_{fTTPA} = E_{K_A}(\sigma_{TTP}) \| C_{TTP}$	
	$M_{fTTPB} = E_{K_B}(\sigma_{TTP}) \| C_{TTP}$	
$\xleftarrow{M_{fTTPA}}$		$\xrightarrow{M_{fTTPB}}$
*** any party can start dispute sub-protocols ***		

Figure 4: Final Stage Sub-Protocol

Both parties A and B have to sign the final node L_f. Signatures σ_{fA} and σ_{fB} are encrypted with the public key K_{TTP} of TTP, then the encrypted messages M_{fA} and M_{fB} are sent to TTP.

On receiving messages (service requests from A and B), TTP decrypts the message with its private key K_{TTP}^{-1}, extracts the parties signature, identifies the corresponding party's ID. Use each parties' public key to extract and compare whether both parties' L_f are equal. If they are equal, the TTP will send confirmation and preserve the last node for the contracting parties. If the results are not equal, TTP will sign the concatenation of σ_{fA}, σ_{fB} and error message m_{er}. Encrypted error messages M_{fTTPA} M_{fTTPB} will be sent to party A and B correspondingly.

4.3.4 Protocol Disruption Protection

This stage of protocol has fall back functionality to increase system reliability. It initiates early involvement of the TTP when trust becomes an issue. The secure communication protocol can be operated dishonestly, the last one or two hash chain nodes being vulnerable to this type of attack. 1)One party receives a message but does not send RSP and

claims either, to have not received the message, or to have technical problems. 2)The party, further more, can add its own node and send it, such that there is no evidence that it ever received the other party's message. 3)One party can add a message to its hash chain but does not send it to the other party.

Such problems and attacks will result in contracting parties having an unsynchronized hash chain, meaning they no longer have identical contracting evidence. In this situation, any party can initiate this set protocol to engage the TTP.

The TTP will act in such a role: witness the dispute between two parties on the hash chain and roll back the hash chain to a common point to re-start the negotiation. Parties also have choices of either terminating or restarting the negotiation.

The protocol disruption protection contains four sub-protocols: Dispute I and II Sub-Protocols, Termination Sub-Protocol, and Engage TTP Sub-Protocol. This set of sub-protocols will resolve most common situations providing both the reliability of the communication protocol and the flexibility for freedom of contract.

The protocol disruption protection for the Negotiation Stage protocol is demonstrated in figures 5, 6, 7, 8. In this case, party B initiates the protocol to engage TTP for communication with party A.

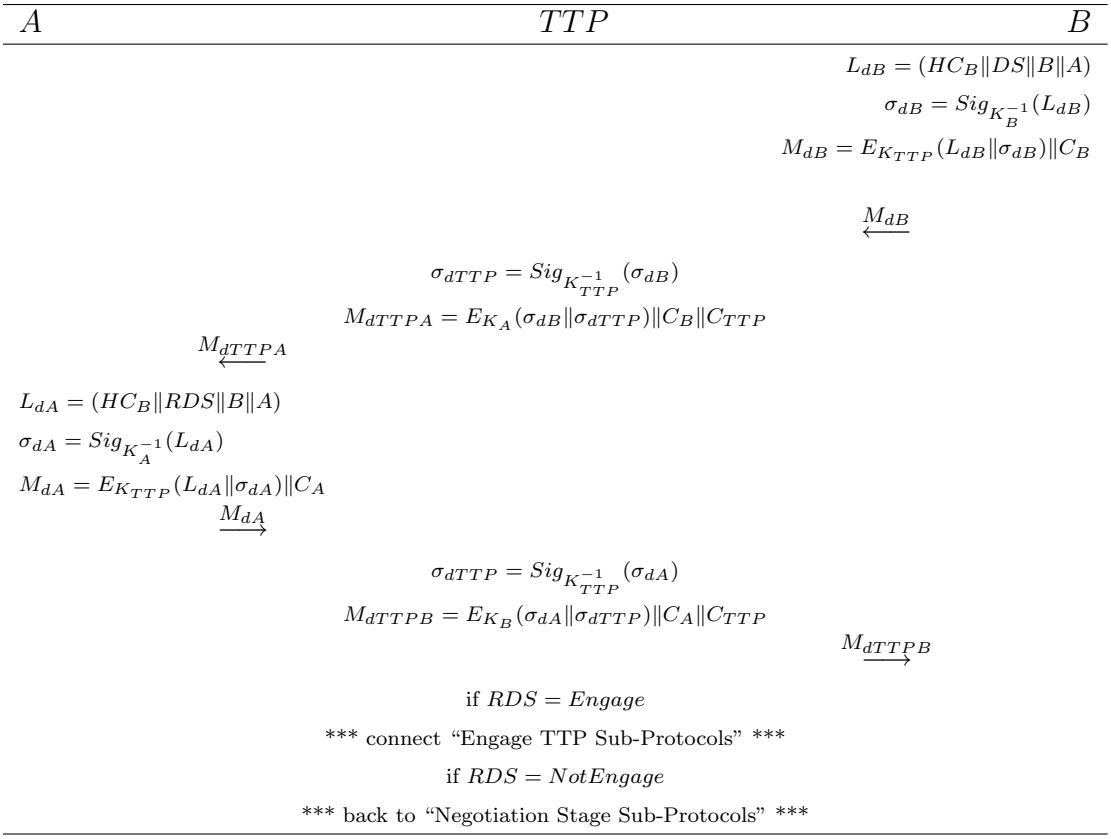

Figure 5: Dispute I Sub-Protocol

Dispute I Sub-Protocol In Dispute I Sub-Protocol (Figure 5), party B raises the dispute and A agrees that party B is right. Party A has two options after it accepts B's hash chain integrity. In this protocol, B first sends its own hash chain HC_B along with $DS \| B \| A$ to TTP. This message indicates to the TTP that there is a hash chain

integrity dispute between parties A and B. The party B is the initiator. TTP signs the σ_{dB} and passes it to party A. A will sign B's hash chain as proof of the agreement on hash chain integrity, and sends back RDS along with other information to indicate to the TTP what it prefers to do next. TTP signs and passes the message from A to B. According to A's response RDS, parties can connect to "Engage TTP Sub-Protocol" or back to "Negotiation Sub-Protocols".

A	TTP	B
		$L_{dB} = (HC_B \| DS \| B \| A)$
		$\sigma_{dB} = Sig_{K_B^{-1}}(L_{dB})$
		$M_{dB} = E_{K_{TTP}}(L_{dB} \| \sigma_{dB}) \| C_B$
		$M_{dB} \xleftarrow{}$
	$\sigma_{dTTP} = Sig_{K_{TTP}^{-1}}(\sigma_{dB})$	
	$M_{dTTPA} = E_{K_A}(\sigma_{dB} \| \sigma_{dTTP}) \| C_B \| C_{TTP}$	
$M_{dTTPA} \xleftarrow{}$		
$L_{dA} = (HC_A \| RDS \| B \| A)$		
$\sigma_{dA} = Sig_{K_A^{-1}}(L_{dA})$		
$M_{dA} = E_{K_{TTP}}(L_{dA} \| \sigma_{dA}) \| C_A$		
$M_{dA} \xrightarrow{}$		
	$\sigma_{dTTP} = Sig_{K_{TTP}^{-1}}(\sigma_{dA})$	
	$M_{dTTPB} = E_{K_B}(\sigma_{dA} \| \sigma_{dTTP}) \| C_A \| C_{TTP}$	
		$M_{dTTPB} \xrightarrow{}$
	if $RDS = TM$	
	*** connect "Terminator Sub-Protocol" ***	
	if $RDS = Engage$	
	$\sigma_{TTP} = Sig_{K_{TTP}^{-1}}(HC_{RBP})$	
	$M_{TTPB} = E_{K_B}(HC_{TTP} \| \sigma_{TTP} \| A) \| C_{TTP}$	
	$M_{TTPA} = E_{K_A}(HC_{TTP} \| \sigma_{TTP} \| B) \| C_{TTP}$	
$M_{TTPA} \xleftarrow{}$		$M_{TTPB} \xrightarrow{}$
	*** connect "Engage TTP Sub-Protocol" ***	

Figure 6: Dispute II Sub-Protocol

Dispute II Sub-Protocol This Dispute II Sub-Protocol (Figure 6) starts in the same manner as "Dispute I Sub-Protocol". The difference is that A disagrees with B's hash chain integrity. Instead of signing HC_B, A signs its own hash chain HC_A along with a response message and sends it to the TTP. The party A has two choices; either terminate the communication or roll back to a common point, then resume the communication under TTP's supervision. TTP passes the response to B. If A's response is termination, parties connect to "Termination Sub-Protocol", otherwise TTP rolls back to a common point HC_{TTP}, and connects to "Engage TTP Sub-Protocol".

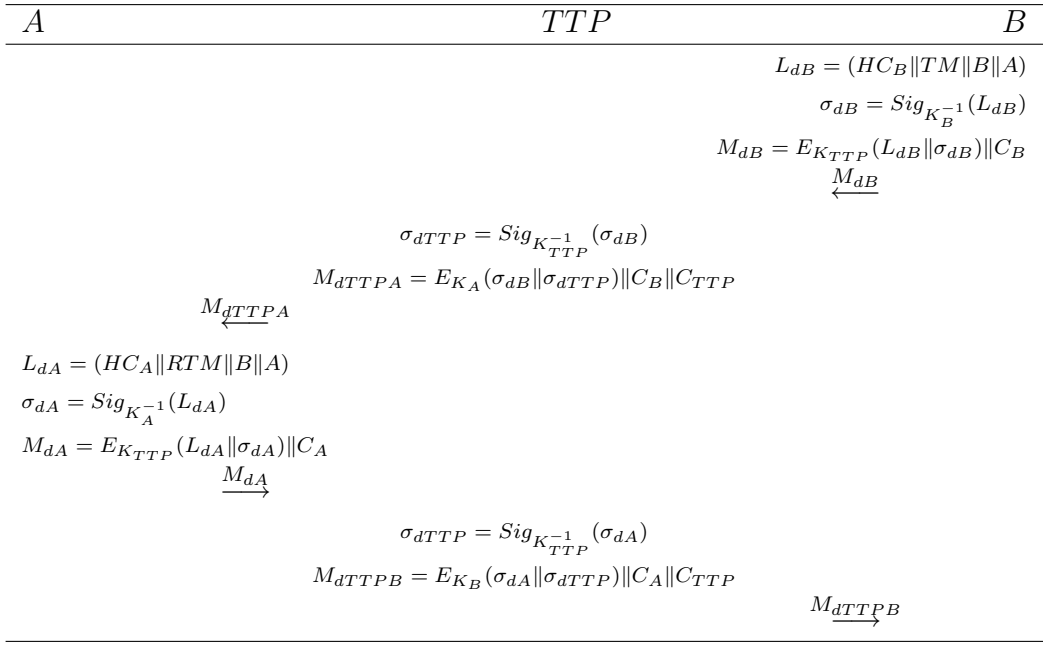

Figure 7: Termination Sub-Protocol

Termination Sub-Protocol The "Termination Sub-Protocol" (Figure 7) has the same format of dispute sub-protocols, with TTP passing and witnessing the parties' termination process. The "Termination Sub-Protocol" can be executed from any stage of the protocol instead of only from dispute sub-protocols. In Figure 7, B initiates termination by sending its signed hash chain (HC_B) to TTP. TTP signs B's signature (σ_{dB}) and passes the message to A. Within the response of B's termination request, A signs its own hash chain (HC_A) and sends it to TTP for witnessing. TTP also signs A's signature and passes the response to B.

Engage TTP Sub-Protocol The "Engage TTP Sub-Protocol" (Figure 8) is similar to "Negotiation Stage Sub-Protocol" with extra information for TTP to build a hash chain that both parties agree upon. Figure 8 demonstrates that the message originator B encodes the original message with A's public key to block the TTP from viewing the message content. B then encodes the rest of the information with M_n using TTP's public key, and sends M_{nTTP} to TTP. TTP signs the $(L_n \| \sigma_{nB})$ and sends all information to A. When A receives the message, it can calculate L_n with original message and compare the L_n with the extracted L_n from the two signatures. A then signs the hash chain node and sends RSP_n back to TTP. When TTP confirms that both parties agree on the same L_n, TTP passes the RSP_{TTP} to A.

5 ANALYSIS

Our secure communication protocol for e-tendering is based on a one way hash function and asymmetrical encryption, and also combines signature and hash chain algorithms to achieve the specified security properties. This protocol uses a hash chain forming process to emulate traditional contracting procedure for e-tendering, which protects the protocol from traditional business attacks (forgery). Our protocol uses asymmetrical encryption to provide communication confidentiality. The signature algorithm provides data origin

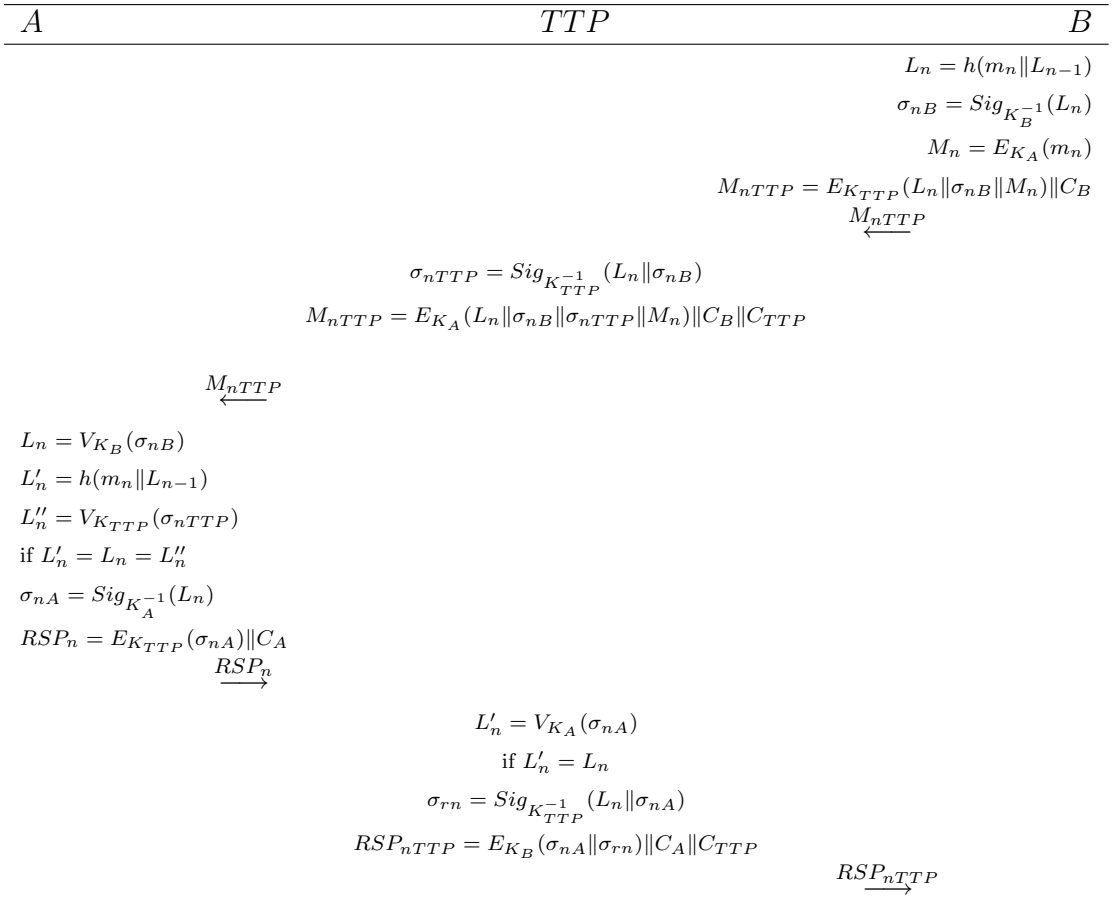

Figure 8: Engage TTP Sub-Protocol

authentication, and the hash chain provides original integrity confirmation. The detailed technical security analysis is discussed against each security property stated in section 3.

5.1 Confidentiality

The confidentiality is achieved by using asymmetrical encryption. In practise, this asymmetrical encryption will be used jointly with DHIES for efficiency. For example, party A sends party B a message, the message will be encrypted with party B's public key using a secure asymmetrical encryption. If an eavesdropper intercepts the message, this secure asymmetrical encryption will block extraction of any information from the cypher text without B's private key. The communication confidentiality relies on the security of asymmetrical encryption.

5.2 Data Origin Authentication

Signature scheme has been used to achieve the Data Origin Authentication security property. The main advantage of this protocol is that the message originator is in charge of generating a check value (hash chain node) which is used to confirm this message's original integrity at a later time. This check value is also digitally signed by the message originator using their own private key, which demonstrates their commitment to the transaction.

A secure digital signature algorithm will bind the message to the originator. It prevents the message originator from denying the transaction (message and hash chain node). This will achieve the data origin authentication. For example, if later on, party A denies that it sent the above message to B, the challenger can use verifying algorithm to provide evidence. If the message can hash back into the corresponding chain node in a hash chain, and this node is equal to the chain node extracted from A's signature (only A can generate its own signature), party A cannot deny the transaction.

5.3 Original Integrity Confirmation

Hash chain is the check value that confirms the input message has remained unchanged from its original form. Hash chain integrity itself is protected by the signature over each node during the contracting period, and by the TTP for the end node of the chain.

This hash chain prevents both contracting parties from altering the set of contracting evidence (messages and chain nodes). If B alters a message and corresponding hash chain node which were generated by A, B cannot compute A's signature over this altered node, because B does not know A's private key. If B alters a message and node generated by itself with a new signature, A can prove that the new signature does not match the one it received before. According to this evidence, A can prove that B is dishonest.

Under a protocol attack, the hash chain can be recalculated to cover up any trace of interference (deletion, insertion, resequencing, alteration of content), after a contracting party has varied a chain node. This recalculation can occur during and after the contracting process. Personal digital signature has the functionality to prevent this recalculation during the contracting process by protecting chain nodes (demonstrated in the above example). But after the contracting process, the key pairs may be revoked in the short term.

TTP is introduced to preserve the last node of the hash chain after a successful contract negotiation in the e-tendering process. In a technical sense the TTP is incorporated to further protect the hash chain's integrity, by eliminating the short term key revocation of digital signature, which is used to protect the hash chain during the contracting process.

For example, after the final contract is formed, party B obtains A's private key (worst case condition). B's intention is to vary the contract evidence, recalculate the hash chain, and use this key to re-sign each node created by A. The result is that A has evidence which is different from B's evidence. Theoretically, both parties should possess one identical set of contracting evidence. Without the TTP to hold the last node, a challenger may not be able to distinguish which party has recalculated the entire hash chain to suit their own purpose.

In contrast, any recalculated hash chain is unable to match the last node that is held by the TTP. Our protocol prevents the hash chain from being recalculated both during, and subsequent to, the contracting process. Contracting parties' interaction with the TTP can also confirm that both parties agreed on one set of agreements and possess identical contracting evidence. The assurance of contracting parties holding identical contracting evidence set is not provided by traditional tendering systems.

TTP preserves the integrity of the last node of the hash chain and in turn protects the original integrity of contract evidence over the term of the contract. In a legal sense the TTP is acting as the witness of the final contracting result.

Because the chain integrity is preserved the original integrity of contract evidence can

be confirmed by the hash chain as the check value at any later time.

5.4 System Reliability

If trust is an issue, any party can invoke fall back protocol with TTP. The Dispute I, II, Termination and Engage TTP Sub-Protocols are constructed to solve common disputes. Therefore the communication protocol can proceed to a more secure termination or to the final stage of e-tendering. With this set of sub-protocols in place, parties are able to collect and maintain a complete set of communicated evidence regardless of whether a contract is awarded or negotiation is terminated prematurely.

In Dispute I and Dispute II Sub-Protocols, the TTP's intervention will force party A to respond and make decisions about how the rest of the e-tendering process will be conducted with party B. Dispute I covers human errors and technical problems causing unmatching hash chain integrity. A agrees B's hash chain integrity and resumes the communication. Dispute II covers dishonest operation causing unmatching hash chain integrity. In this situation, trust is an obvious issue. If parties still want to proceed, the rest of communications need to be monitored. Otherwise, parties can formally terminate the communication with the involvement of TTP.

6 CONCLUSION

Our secure communication protocol has set up a framework for preserving the integrity of the e-tendering process. It requires logging all types of communication techniques and produces integrity check values for the entire contracting process in e-tendering. These check values have been effectively preserved by cryptographic algorithms to provide reliable confirmation checking of original integrity for contracting evidence. It is the first step for legal compliance of e-tendering scheme.

The security property of the protocol also eliminates the existing problems in the tendering process such as logging telephone negotiations. It eases the long existing problem of how to provide reliable digital evidence in court, which has put risks in e-commerce.

The e-tendering protocol is modeling a complex business process. For quality insurance, we intend to verify the protocol's security with formal verification tools in future work.

7 ACKNOWLEDGMENTS

We thank Ann Fitzgerald, Principal Lawyer for the Crown solicitor, for providing a valuable analysis report on Electronic Transaction Act 1999, Australia and electronic tendering issues.

References

Anderson, R., Manifavas, C., and Sutherland, C. (1997). Netcard – a practical electronic cash system. In *Fourth Cambridge Workshop on Security Protocols*, pages 49–57.

Christensen, S. (2001). Formation of contracts by email-is it just the same as the post? *Queensland University of Technology Law and Justice Journal*, pages 22–38.

Diffie, W. and Hellman, M. E. (1976). New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654.

Du, R., Foo, E., Boyd, C., and Fitzgerald, B. (2004). Defining security services for electronic tendering. In *The Australasian Information Security Workshop (AISW2004)*, volume 32, pages 43–52. Australian Computer Society Inc and ACM.

ElGamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472.

Haber, S. and Stornetta, W. S. (1991). How to time-stamp a digital document. *Journal of Cryptology*, 3(2):99–111.

Karjoth, G., A. N. G. C. (1998). Protecting the computation results of free-roaming agents. In Rothermel, K., H. F., editor, *Proceedings of the 2nd International Workshop on Mobile Agents (MA '98)*, volume 1477 of *Lecture Notes in Computer Science*, pages 195–207. Springer-Verlag, Berlin Heidelberg New York.

Kelsey, J. and Schneier, B. (1999). Minimizing bandwidth for remote access to cryptographically protected audit logs. In On-line proceeding:*Recent Advances in Intrusion Detection (RAID 99)*, http://www.raid-symposium.org/raid99/PAPERS/Kelsey.pdf

Menezes, A., van Oorschot, P., and Vanstone, S. (1997). *Handbook of Applied Cryptography*. CRC Press, Boca Raton, Florida 33431.

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

Rivest, R. L. and Shamir, A. (1996). Payword and micromint: Two simple micropayment schemes. In *Security Protocols Workshop*, pages 69–87.

Thorpe, C. and Bailey, J. (1996). *Commercial contracts, A practical guide to deals, contracts, agreements and promises*. Woodhead, Cambridge England.