A Software Framework for Non-repudiation Service in Electronic Commerce based on the Internet

Sung woo Tak*, Yugyung Lee**, and Eun Kyo Park***
School of Interdisciplinary Computing and Engineering
University of Missouri - Kansas City
5100 Rockhill Rd. Kansas City, MO 64110
swt5e1@umkc.edu*, leeyu@umkc.edu**,ekpark@umkc.edu***

***Abstract*—We propose a software framework for non-repudiation service in e-commerce (electronic commerce) on the Internet. The proposed software framework is an explicit security framework for notary service. In the framework we propose a systematic design methodology that provides a security class concept. Our framework can be differentiated from others. First, unlike other frameworks, it is interested in a successful completion of e-commerce transactions by supporting non-repudiation of service. Second, the proposed framework is based on dynamic adaptive mechanism that improves the performance of e-commerce transactions.**

Keywords-non-repudiation; security; electronic commerce;

I. INTRODUCTION

Generally, existing secure transactions do not concentrate on non-repudiation of service except SET (Secure Electronic Transaction) that provides partial non-repudiation of service [1]. The concept of trust in traditional commerce models is different from that of e-commerce models. Under traditional commerce models, we seldom provide proof-positive identification when giving out credit card numbers. In contrast to traditional commerce models, e-commerce models are notoriously insecure. To cope with the rapid growth of e-commerce over the Internet, the issue of e-commerce security should be fully addressed. ISO (International Organization for Standardization) defines five security services as authentication, access control, confidentiality, data integrity, and non-repudiation [2]. Our paper focuses on the non-repudiation security service. Non-repudiation of service is quite different from the other four security services. Non-repudiation of service aims to protect transactions against attacks from outside intruders. One of attacks is a false denial of a particular event or action among transactions. The denial of service needs to be resolved based upon the evidence of a transaction, which is generated, collected, and maintained by non-repudiation of service [3]. Non-repudiation of service is an essential feature of e-commerce security service to establish the legal basis of an electronic transaction. In this paper we propose a software framework for non-repudiation of service that protects both merchants' right and customers' right on an electronic transaction. The proposed framework resolves potential disputes through non-repudiation of service against repudiation of origin, delivery, receipt, and submission. The proposed framework is based on a dynamic mapping mechanism supported by prioritizing security classes. The adaptive mechanism can cope with the change and evolution of network conditions.

II. NOTARY AUTHORITY FRAMEWORK

A. Existing E-commerce Frameworks

There are two well-known e-commerce frameworks. First, SSL (Secure Socket Layer) provides a primitive e-commerce framework without a trusted third party [4]. In SSL, a merchant (or a server) and a customer (a client) directly interact with each other to complete a transaction. This is a bilateral arrangement where the customer exposes sensitive customer information, such as credit card information, to the merchant. Reliable non-repudiation of service does not exist in SSL since there are no trusted thirty party interactions to monitor transactions. Second, SET employs a trusted thirty party called a payment gateway to protect customer's payment information from a merchant. Through the payment gateway, SET only concerns whether the customer's payment is completed or not. SET does not consider non-repudiation of service so that it is difficult for SET to complete a successful transaction between a customer and a merchant.

B. Notary Authority Framework

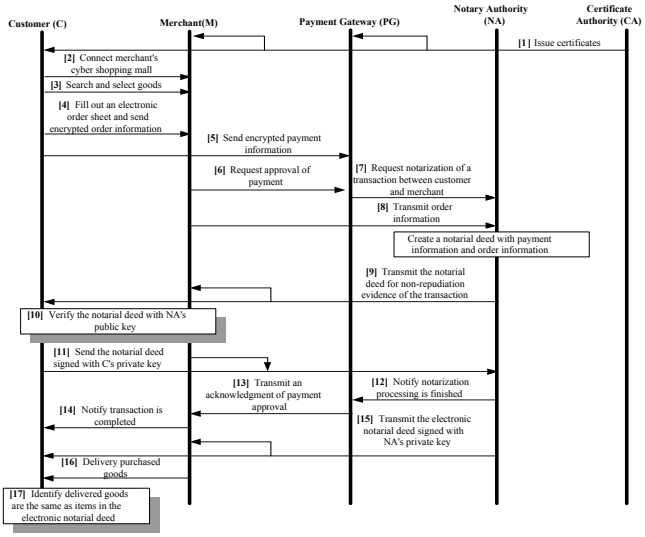

Figure 1. Transaction flows of notarial service

Our framework is based on an explicit model of non-repudiation that generates evidence called a notarial deed. The notarial deed is securely contained in an encapsulated data structure and delivered to anticipatory partners. The proposed framework is called a NA (Notary Authority) framework. The proposed NA framework can be an extension of SET by adding notarial service as non-repudiation of service supported by a trusted third party, notary authority. Figure 1 shows the transaction flows in our NA framework.

TABLE I. SCOPE OF TRANSACTIONS IN NA FRAMEWORK

Step	Transaction	Scope
1	***CA*** issues each certificate to ***C***, ***M***, ***PG***, and ***NA***	NA
2	***C*** connects to ***M***'s cyber shopping mall	Cyber mall
3	**C** searches goods in the cyber shopping mall and selects items to purchase	Cyber mall
4	***C*** sends encrypted order information to ***M***	SET/ NA
5	***C*** sends encrypted payment information to ***PG***	SET/ NA
6	***M*** requests an approval of the payment to ***PG***	SET/ NA
7	***PG*** sends a notarization request of the transaction between ***C*** and ***M*** to ***NA***	NA
8	***NA*** receives the order information of ***C*** from ***M***	NA
9	***NA*** transmits a notarial deed including the payment information and the order information to both ***C*** and ***M*** for non-repudiation of evidence	NA
10	Both ***C*** and ***M*** verify the signed notarial deed with ***NA***'s public key	NA
11	Both ***C*** and ***M*** send the notarial deed signed by their own private key to ***NA***	NA
12	***NA*** notifies ***PG*** that notarization processing is finished.	NA
13	***PG*** sends an acknowledgement of the payment approval to ***M***	SET/ NA
14	***M*** notifies ***C*** that transaction is completed.	SET/ NA
15	***NA*** signs the notarial deed for non-repudiation of evidence.	NA
16	***M*** deliveries the purchased items to ***C***	Delivery
17	***C*** identifies that delivered items are the same as the items in the notarial deed.	NA

Table I shows the scope of our proposed NA framework with respect to the steps of an electronic transaction. There are five participants in our NA framework: Merchant (*M*), Customer (*C*), Payment Gateway (*PG*), Notarial Authority (*NA*), and Certificate Authority (*CA*). The electronic notarial deed plays an important role in generating non-repudiation of evidence. The electronic notarial deed is similar to the receipt in traditional commerce models. *NA* generates and verifies the notarial deed signed by its private key, stores the notarial deed, and securely delivers it to *C* and *M*.

TABLE II. SENSITIVE INFORMATION FLOWS OF E-COMMERCE

R / *S*	*C*	*M*	*PG*	*NA*	*CA*
C		***OI***	***PI***	***ND*** **Signature**	*C* Request
M	Goods		PI Request	**OI** / ***ND*** **Signature**	*C* Request
PG		PI Approval		*ND* Request	*C* Request
NA	***ND***	*OI* Request / ***ND***	*ND* Response		*C* Request
CA	*CERT*	*CERT*	*CERT*	*CERT*	

(*CERT*: Certificate, *ND*: Notarial Deed, *OI*: Order Information, *PI*: Payment Information)

Table II shows diverse sensitivity of information required in e-commerce transactions. The row specifies a sender (*S*) and the column specifies a receiver (*R*). In Table II, the information in bold type represents the most sensitive information while italic type specifies the least sensitive information. Others are in the intermediate level of information sensitivity.

III. DESGIN OF NOTARIAL METHODOLOGY

To overcome the inefficient problem of existing secure software frameworks, we propose a reasonable mechanism to map between a message priority and a level of security. We suggest a modular and methodical approach for building a dynamic mapping mechanism. The mechanism is based on a dynamic mapping function and a security class library. The design strategies and principles are described in a set of rules: (1) Quantification of computational costs and security strength through an extensive empirical study on the performance of the cryptographic techniques and computational overheads, (2) Degree of information sensitivity carried on the messages for non-repudiation of service, and (3) Security classes for reasonable mapping between (1) and (2). The notarial methodology is designed with three primary requirements: efficiency, flexibility, and security. The message generation and delivery is structured as a library with a well-defined interface. As a part of the interface, a mapping function dynamically provides reasonable mapping between a network environment and security classes. The key to flexibility and efficiency is that the notarial methodology incorporates diverse cryptographic techniques and time functions. To simplify the methodology interface, it is necessary to have a pre-defined group of messages. When messages are generated and delivered, detailed policies may be stored in the library.

A. Dynamic Mapping Function

A dynamic mapping function is designed to support the best selection in a situation. The situation is determined according to message sizes, a message sender and a message receiver, a degree of information sensitivity in the message, and a status of network. The dynamic mapping function receives a runtime environment as an input and returns an appropriate security class of a delivered message. We now define the dynamic mapping function (*MF*) and the security class (*SC*) selected by *MF*. *MF* is represented as $SC = MF(M, S, R, SL, DT)$ where *M* stands for "Message", *S* for "Message Sender" $\in \{C, M, PG, CA, NA\}$, *R* for "Message Receiver" $\in \{C, M, PG, CA, NA\}$, *SL* for "Security Level for Information Sensitivity" $\in \{1,2, 3, 4\}$, and *DT* for "Message Delivery Time " $\in \{T1, T2, T3\}$. *MF* returns *SC* to represent a proper level of cryptographic securities. Three time functions are used in *MF*: message generation, delivery, and verification. Message generation time function *(MG-T1)* computes encoding time for a message according to the message sensitivity and the message size. In *MG-T1*, *M* stands for the message to be generated. *SL* is the level of information sensitivity that returns from the function, *Sensitiveof*. *T1* stands for message generation time extracted from a pre-computed lookup table ($Lookup_{encoding/decoding}$) according to the message size and *SL* where *n* is the number of encoding methods.

Time function *MG-T1* for message generation
Input: M
Output: $T1$
$SL \leftarrow Sensitiveof(M)$
$$T1 \leftarrow \prod_{i=1}^{n} Lookup_{encoding}(Sizeof(M), SL)$$

Message delivery time function, *MD-T2*, estimates message delivery time at a network situation. The time to deliver a message is computed based on RTT (Round Trip Time) of the previous message. In *MD-T2*, the message delivery time is measured by two cases. In case of an initial message, a sender uses initial estimated time $T2'$ for the message delivery time. For subsequent messages, a sender estimates delivery time $T2''$ based on RTT of the previous messages. *MD-T2* is designed based on [5-7]. α is the ceiling ratio of a message to be delivered. Message verification time function, *MV-T3*, computes the decoding time of a message. We need to compare the computing capacity of the message receiver system to that of the message sender system for an accurate estimation of the message verification.

Time function *MD-T2* for message delivery
Input: M, M', A, D, g, h, E_i
Output: $T2$

(1) $$T2' = \prod_{i=1}^{n} A_{initial} + E_i D_{initial}$$

(2)
$$Err = RTT - A$$
$$A \leftarrow A + gErr$$
$$D \leftarrow D + h(|Err| - D)$$
$$T2'' = A + 4D$$

(3) $$T2 = \begin{cases} T2' * \alpha, if\ Message = First\ Message \\ T2'' * \alpha, Otherwise \end{cases}$$

$RTT : RTT(Round-Trip\ Time)\ Measurement\ of\ Each\ Message$
$A : An\ Estimator\ of\ the\ Average$
$D : Smoothed\ Mean\ Deviation$
$Err : Difference\ between\ the\ measured\ value$
$g : Gain\ for\ the\ Average$
$h : Gain\ for\ the\ Deviation$
$E_i : Exponential\ Backoff,\ E_i = \{2, 4, 8, 16, 32, 64\},\ 1 \le i \le n$
$M' : Current\ Message\ To\ be\ Delivered$
$M : Previous\ Message\ Delivered$
$$\alpha : \left\lceil \frac{Sizeof(M')}{Sizeof(M)} \right\rceil, \alpha \ge 1$$

The concept of timeout function (*TO)* is a way of specifying the validity of evidence for a particular event or action. It is based on the time of message generation, delivery, and verification. Any message after a timeout period *(Tn > TO*) is ignored. β is a variance factor of *TO*. Dynamic mapping function, *DM,* returns a *SC*. *DM* was empirically defined by an extensive performance test of existing cryptographic techniques. The dynamic mapping rules are as follows. (1) *Rule 1:* A security class is determined based on information sensitivity (2) *Rule 2:* when the information sensitivity level of a message is 3 and the message size is bigger than $Size_{threshold}$ (Message Size Threshold), a dynamic decision rule are applied to map NR-SC3 (Non-repudiation Security Class-3) into NR-SC3-V (Non-repudiation Variation Security Class-3) described in Section III-B (3) *Rule 3:* when network traffic is congested, class degradation rules are applied to adjust the network situation. The class degradation rules are follows: (1) NR-SC2 is degraded into NR-SC1 and NR-SC3 is degraded into NR-SC2 (2) NR-SC1 and NR-SC4 remain at their security level.

Time function *MV-T3* for message verification
Input: M
Output: $T3$
$SL \leftarrow Sensitiveof(M)$
$$T3 \leftarrow \prod_{i=1}^{n} Lookup_{decoding}(Sizeof(M), SL)$$

Timeout function *TO* for message
Input: $T1, T2, T3, \beta$
Output: TO
$$TO = \beta \sum (T1, T2, T3), where\ T1, T2, T3 \ge 0\ and\ \beta \ge 0$$
$T1 : Message\ Generation\ (Encryption)\ Time$
$T2 : Message\ Delivery\ Time$
$T3 : Message\ Verification\ (Decryption)\ Time,$
$\beta :\ TO\ variance\ factor, \beta \ge 1$

Dynamic mapping function
Input: $M, T2$
Output: SC
$SL \leftarrow Sensitiveof(M)$
$if\ (SL = 1)\ SC \leftarrow M_{NR-SC1}$
$else\ if\ ((SL = 2) \wedge (T2 > T_{threshold}))\ SC \leftarrow M_{NR-SC1}$
$\quad else\ SC \leftarrow M_{NR-SC2}$
$else\ if\ (SL = 3)$
$\quad if\ (Sizeof(M) \le Size_{threshold})$
$\quad\quad if(T2 > T_{threshold})\ SC \leftarrow M_{NR-SC2}$
$\quad\quad else\ SC \leftarrow M_{NR-SC3}$
$\quad else\ if\ (T2 > T_{threshold})\ SC \leftarrow M_{NR-SC2}$
$\quad\quad else\ SC \leftarrow M_{NR-SC3-v}$
$else\ if\ (SL = 4)\ SC \leftarrow M_{NR-SC4}$

$M : Current\ Message$
$SL : Levelof\ Information\ Sensitivity$
$NR\text{-}SC : Non\text{-}repudiation\ Security\ Class$
$T_{threshold} = Threshold\ of\ Message\ Delivery$
$Size_{threshold} = Threshold\ of\ MessageSIze$

B. Security Class Library

Non-repudiation security classes have eight attributes: Security Level = {1, 2, 3, 4}, Sender = {*C, M, PG, CA, NA*}, Receiver ={*C, M, PG, CA, NA*}, Cryptographic Technique = {DES (Data Encryption Standard), 3DES (Triple DES), SHA-1 (Secure Hash Algorithm), RSA (Rivest, Shamir, and Adelman), Random number generation}, $TIMEOUT_{period}$ = {$T_1, \ldots, T_n$ }, Performance = {$C_1, \ldots, C_m$}, Encoding, and Description. The security classes define a message with a proper level of security.

1) Design of security classes

The formalization of security classes is determined in terms of the level of message confidentiality, message integration, and message origin authentication. The message confidentiality is provided with secrete key and public key encryption techniques. The level of security techniques is determined based on the number of keys. In our formula, as the number of keys is increased, the security level becomes higher. The message integration and the origin authentication use a message digest technique, SHA-1. These security levels are prioritized according to the order of security level/performance

from lowest/fastest to highest/slowest. In the formula, *X* stands for a message, *MD* for a digested message by a one-way hash function *H*, *secret key* for secret key encryption, *public key* for public key encryption, and *dynamic* for the dynamic decision rule.

$$\text{(1) Message Confidentiality}$$

$$\{X\}_{secrete\ key\ 1} \le \left\{\{X\}_{secret\ key\ 1}\right\}_{secrete\ key\ 2} \} \le \left\{\left\{\{X\}_{secre\ key\ 1}\right\}_{secret\ key\ 2} \}\right\}_{...secrete\ key\ n} \in \text{Secret key Encryption}$$

$$\{X\}^{public\ key1} \le \left\{\{X\}^{public\ key1}\right\}^{public\ key2} \} \le \left\{\left\{\{X\}^{public\ key1}\right\}^{public\ key2} \}\right\}^{.... public\ keyn} \in \text{Public key Encryption}$$

$$\text{(2) Message Integration and Origin Authentication}$$

$$\{MD\}^{private\ key} \le \left\{MD_{dynamic}\right\} \le \{X\}^{private\ key}, \quad MD = H(X),\ X = Original\ Message$$

$$\Rightarrow \text{Security Level Formulation}$$

$$SL1\left[\{X\}_{secret\ key\ 1} \mid \left\{\{MD\}^{private\ key}\right\}_{secret key2}\right] \le SL2\left[\left\{\{X\}_{secret\ key\ 1}\right\}_{secret\ key\ 2} \mid \left\{\{MD\}^{private\ key}\right\}_{secret key1}\right]$$

$$\le SL3\left[\left\{\left\{\{X\}^{private\ key}\right\}_{secret\ key\ 1}\right\}_{secret\ key\ 2}\right] \text{ or } \left[\left\{\left\{MD_{dynamic}\right\}_{secret\ key\ 1}\right\}_{secret\ key\ 2}\right]$$

$$\le SL4\left[\left\{\{X\}^{private\ key}\right\}^{public\ key}\right]$$

In message confidentiality, encryption using multiple different secret keys or different public keys is stronger than single key encryption but the performance of multiple secret keys or public keys is slower than a single key. We prioritize the strength of message confidentiality as described in (1) message confidentiality. In (1), public key encryption is more secure than secret key encryption that shares a single key among participants in terms of non-repudiation service. Multiple message digests ($Md_{dynamic}$) signed with a private key can be stronger than a single message digest signed with a private key. However, an intruder can attack $Md_{dynamic}$ signed with a private key by tricking message digests. Thus, a whole original message signed with a private key is much stronger than $Md_{dynamic}$ because it is impossible for an intruder to modify the whole original message without knowing a private key. We prioritize the strength of message integration and origin authentication as illustrated in (2) message integration and origin authentication. NR-SC2 using 3DES with three different secret keys is more secure than NR-SC1 using DES with a single secret key. NR-SC3 using the method of signing multiple message digests is more secure than NR-SC2 that signs only one message digest of the original message. NR-SC4 signs a whole message with a private key and also encrypts the whole message with a public key. So, NR-SC4 is more secure than NR-SC3 that encrypts the message and signs message digests. Now we define the notation to represent the encoding of security techniques used in secure classes as follows:

- *Cert*[X]: represent a certificate of participants, i.e., a X.509v3-formatted certificate, where $X \in P = \{C, M, PG, CA, NA\}$..
- $RAND_{letters[numbers]}$: represent a random number technique, where letters = {a,…, z } and numbers = {1,2, …, N}
- $K_{[A][B]1}$ and $K_{[A][B]2}$: represent the first and second shared secret keys used in communication between *A* and *B*, where $A, B \in P$.
- $K^{[A]:}$ stands for A's private key where $A \in P$
- $K^{[A]-1}$: stands for A's public key where $A \in P$
- MD = H(X): represent that MD (Message Digest) produced by one-way hash function, H() with an input X
- $\{X\}_K$: represent a message, X, encrypted by a key K.
- ND[P1][P2]: represent an electronic notarial deed for an transaction between *C* and *M* where $P_1, P_2 \in P$.
- $\{X_1 \mid X_2\}$: represent a message containing X_1 and X_2.
- "Message: S → R [X]": represent the flow of a message X from an S (sender) to an R (receiver).

2) Non-repudiation Security Class-1

NR-SC1 (Non-repudiation Security Class-1) represents the lowest security level in the notarial methodology. NR-SC1 is defined by the combination of DES, SHA-1, and RSA. We take advantage of RSA for digital signature and data encryption. However, since RSA requires the most expensive computation overheads, RSA is only used for signing a digested message in this class. The cryptographic technique for digesting a message is one-way hash function, SHA-1. The digital signature of the digested message can support non-repudiation of origin. The random number supports non-repudiation of delivery. Finally, the message encryption is done with DES. NR-SC1 is similar to the cryptographic techniques used by SET except that we use the two different keys instead of one single key in SET and add a random number technique to NR-SC1.

Non-repudiation Security Class-1 (NR-SC1)

Sender	{*PG, NA, C, M*}
Receiver	{*PG, NA, C, M*}
Security Level	Lowest security level (*SL1*)
Performance	Fastest (*SP1*)
$TIMEOUT_{period}$	*T1*
Cryptographic Methods	DES, SHA-1, Random number, and RSA
Encoding	S: MD = H(X) S → R: $\{X\}_{K[S][R]1}$, $\{\{MD \mid RAND_1\}_K^{[S]}\}_{K[S][R]2}$
Description	Message encryption for the least significant information

3) Non-repudiation Security Class-2

Non-repudiation Security Class-2 (NR-SC2)

Sender	{*M, PG*}
Receiver	{*NA*}
Security Level	Second lowest level (*SL2 > SL1*)
Performance	Moderate (*SP2 < 3*SP1*)
$TIMEOUT_{period}$	*T2*
Cryptographic Methods	DES, 3DES, SHA-1, Random number, and RSA
Encoding	S: MD = H(X) S → R: $\{\{X\}_{K[S][R]1}\}_{K[S][R]2}$, $\{\{MD \mid RAND_1\}_K^{[S]}\}_{K[S][R]1}$
Description	Message encryption for moderately significant information

In NR-SC2 (Non-repudiation Security Class-2), 3DES and DES are used to encrypt and decrypt a message. NR-SC2 is identical to NR-SC1 except for using 3DES. Thus, the security level of NR-SC2 is increased. Since 3DES is at least three times slower than DES according to our experiment,

computation overheads for encrypting a message is more expensive.

4) Non-repudiation Security Class-3

In NR-SC3 (Non-repudiation Security Class-3), an original message is signed with a sender's private key instead of signing a digested message of the original message. There is still a chance for sensitive information to be attacked by an intruder by discovering a way of replacing the signed message digest with another message digest. Hence, in case of relatively significant information, it is required to sign an entire message with a private key.

Non-repudiation Class-3 (NR-SC3)

Sender	{*NA, C, M*}
Receiver	{*C, M, NA*}
Security Level	Second highest level (*SL3* > *SL2*)
Performance	Slow (*SP3* << *SP2*)
$TIMEOUT_{period}$	*T3*
Cryptographic Methods	3DES, Random number, and RSA
Encoding	$S \rightarrow R$: $\{\{\{X \mid RAND_1\}_K^{[S]}\}_{K[S][R]1}\}_{K[S][R]2}$
Description	Message encryption for significant information

5) Non-repudiation Security Variation Class-3

NR-SC3 is used to exchange a notarial deed signed by notarial authority between a customer and a merchant. However, NR-SC3 performance is rapidly degraded as the size of messages is increased.

Non-repudiation Security Variation Class-3 (NR-SC3-V)

Sender	{*NA, C, M*}
Receiver	{*NA, C, M*}
Security Level	Second highest level (*SL3* > *SL2*)
Performance	Moderate ($SP3^v >> SP3$)
$TIMEOUT_{period}$	*T3′*
Cryptographic Methods	3DES, Random number, SHA-1, and RSA
Encoding	$(1) S : MD = H(X)$ $(2) Block_{division} = \left\lceil \frac{Sizeof(X)}{MD_{total\_num}} \right\rceil$ $1 \le Block_{division} \le Sizeof(X), \quad MD_{total\_num} \ge 1$ $(3) Block_N = \left\lceil \frac{Sizeof(X)}{Block_{division}} \right\rceil \quad N = 1,\ldots total\_num$ $(4) MD_{dynamic} = \{MD\}K^{[S]} \mid \sum_{i=1}^{total\_num} \{MD_i = H(Block_i)\}K^{[S]}$ $i = 1,..,\ total\_num$ $(5) X = X + MD_{dynamic}$ $(6) S \rightarrow R : [\{X \mid RAND_{a[1]}\}_{[S][R]1}\}K^{[S]}_{[S][R]2}]$
Description	Message encryption for significant information

When the size of a message exceeds 1K bytes, the performance of NR-SC3 clearly degrades. The reason is because RSA is used to encrypt an entire message. NR-SC3-V (Non-repudiation Security Variation Class-3) is a variation of NR-SC3. NR-SC3-V improves the performance of NR-SC3 while preventing the attack of tricking a message digest. The method to generate multiple message digests is as follows: (1) an *S* (a sender) produces the message digest of a message (2) *S* gets the block size ($Block_{division}$) of the message based on the total number of message digests (Md_{total_num}) (3) *S* gets blocks ($Block_n$ *n = 1,…,total_num*) (4) *S* signs the message digests generated from the blocks (5) *S* attaches singed message digests ($MD_{dynamic}$) and sends them to an *R* (a receiver). By generating multiple message digests, we can increase the security level without much affecting the performance of execution time in encrypting and decrypting a message.

6) Non-repudiation Security Class-4

In NR-SC4 (Non-repudiation Security Class-4), only public key encryption (RSA) is used to encrypt and decrypt a message. RSA increases the most expensive computation cost but it has the highest security capabilities. Since the size of a message in NR-SC4 is less than 1K bytes, the performance of NR-SC4 is not affected by expensive computation cost.

Non-repudiation Security Class-4 (NR-SC4)

Sender	{*M, C, PG, NA*}
Receiver	{*M, C, PG, NA*}
Security Level	Highest level (SL4 > SL3)
Performance	Slowest (SP4 < SP3)
$TIMEOUT_{period}$	*T4*
Cryptographic Methods	Random number and RSA
Encoding	$S \rightarrow R$: $\{\{X \mid RAND_1\}_K^{[S]}\}_K^{[R]-1}$
Description	Message encryption for the most significant information

IV. Non-Repudiation of Service

The non-repudiation security defined by the ISO such as non-repudiation of origin, deliver, receipt, and submission are supported by the notarial methodology. The non-repudiation security in the notarial methodology is reviewed.

A. Non-repudiation of origin

Non-repudiation of origin is defined as a security service to protect against the originator's false denial of having sent the message. To provide non-repudiation of origin as proof that a sender transmits a message to a receiver, the sender should encrypt the message containing a sender's certificate with its private key whenever the sender transmits the message to the receiver. On receiving the encrypted message, the receiver can validate that the message originates from the sender as non-repudiation of origin. Using this example, we will illustrate how our notarial methodology provides non-repudiation of origin.

Example 1: A customer sends order information (OI) to a merchant. In the message, the customer produces a message digest (MD) for the input OI. Then, the customer transmits a message that consists of the OI encrypted with a shared secret key $K_{[S][R]1}$, the message digest (MD), its random number generation ($RAND_{a[1]}$) encrypted with the customer's private key $K^{[S]}$ and the other shared secret key $K_{[S][R]2}$. This is an example of proof for non-repudiation of origin from a customer (S) to a merchant (R).

S: MD = H(OI)

$S \rightarrow R$: $[\{OI\}_{K[S][R]1}, \{\{MD \mid RAND_{a[1]}\}_K^{[S]}\}_{K[S][R]2}]$

On receiving the message, the merchant performs three steps: (1) decrypt the message with two shared secret keys, (2) re-compute the message digest, and (3) decrypt the message

digest with the sender's public key. If the recomputed message digest and the decrypted message digest match, the merchant trusts that the customer transmitted the message. Therefore, these three steps can provide proof for non-repudiation of origin.

B. Non-repudiation of delivery

Non-repudiation of delivery is defined as a service to provide message delivery confirmation that protects against attempts by the recipient who falsely denies receiving the message. The verification of whether the contents of message were changed in transit typically counts on a message digest generated by a one-way hash function. The transferred message digest from a sender is identical to the message digest generated by a receiver that receives the message from the sender. Since it is difficult to find the same fixed-size message for variable size inputs, it is safe to use for non-repudiation of delivery. The hash function returns a digested message to the sender so that the digested message can be used as delivery confirmation to the sender, namely non-repudiation of delivery. Using an example of a transaction between a merchant and a customer, we illustrate how our non-repudiation software framework provides non-repudiation of delivery.

Example 2: A merchant (S) sends a payment information request to a customer (R). The customer generates a new random number. The customer signs the new random number and a copy of the random number transmitted by the merchant with the customer's private key. The customer transmits two random numbers to the merchant for an acknowledgment of the previous message. The message contains two components: (1) customer's payment information (PI) encrypted with a shared secret key $K_{[S][R]1}$ (2) a message digest (MD) of PI, $RAND_{a[1]}$ that was previously transmitted by the merchant, and $RAND_{a[2]}$ that is newly generated by the customer. These are signed by the customer's private key ($K^{[R]}$), and then encrypted with the other shared secret key $K_{[S][R]2}$. This is proof for customer's payment information as non-repudiation of origin. After the customer is finished with message encryption and digital signature, the customer transmits it to the merchant.

$$R: MD = H(PI)$$

$$R \rightarrow S: [\{PI\}_{K[S][R]1}, \{\{MD \mid RAND_{a[1]} \mid RAND_{a[2]}\}_{K}^{[R]}\}_{K[S][R]2}]$$

On receiving the message including the previous random number transmitted by the merchant and decrypting it with the customer's public key, the merchant finds out that the previous message was delivered to the customer securely and reliably. Therefore, a copy of the previous random number can provide proof for non-repudiation of delivery.

C. Non-repudiation of receipt

Non-repudiation of receipt is defined as a security service to provide the originator of a message with proof of receipt of the message. It protects against attempts by the recipient who falsely denies receiving the message. Notary authority in our proposed software framework for non-repudiation service generates a notarial deed, and then asks a customer and a merchant to sign a digital signature for the notarial deed. The notarial deed is evidence of non-repudiation. After receiving the notarial deed signed with the customer's private key and the merchant's private key, notarial authority transmits the notarial deed to the customer and the merchant. Therefore, the notarial deed is non-repudiation of receipt provided by the trusted third party, notarial authority. As with non-repudiation of origin, a public key encryption and digital certificates are used to generate the notarial deed with digital signature. Secret key encryption is used to deliver the notarial deed in a form of concealment. The notarial deed plays a role of non-repudiation of receipt.

D. Non-repudiation of submission

Non-repudiation of submission is defined as a service to provide the originator of a message with proof of submission of the message. Non-repudiation of submission protect against attempts to falsely deny that the message was submitted for delivery to the originally specified recipient. Non-repudiation of submission prevents or resolves disagreements by identifying the message submitted by a participant and the message submission time. Messages exchanged among multi-participants in our framework are signed with their own private keys and verified when the messages are processed. Moreover, messages always have a keep-alive time ($TIMEOUT_{period}$) specifying their validity. Thus, non-repudiation of submission is supported by our framework. The assurance that an intruder cannot intercept a message and play it back at some later time can typically be supported by using a random number generation, a sequence counter, or a time stamp [8].

V. EXPERIMENTAL RESULTS

The notarial methodology is implemented by the gnu c compiler, gcc-2.95.2, under a Ultra10 SUN workstation with 64 MB physical memory and 100 clock tick scales in a second. The notarial methodology uses the RSAEURO cryptography library [9]. We assume that no overhead of accessing data is considered since all data are already loaded at physical memory of the workstation.

A. Performance of Security Classes

TABLE III. EXECUTION TIME OF CRYPTOGRAPHIC TECHNIQUES

M Size (Byte)	NR-SC1		NR-SC2		NR-SC3		NR-SC4	
	G (sec)	*V* (sec)	*G* (sec)	*V* (sec)	*G* (sec)	*V* (sec)	*G* (sec)	*V* (sec)
8	0.85	0.05	0.85	0.05	0.85	0.05	0.9	0.9
64	0.85	0.05	0.85	0.05	0.85	0.05	0.9	0.9
117	0.85	0.05	0.85	0.05	0.85	0.05	0.9	0.9
256	0.85	0.05	0.85	0.05	1.85	0.109	1.96	1.96
1K	0.85	0.05	0.85	0.5	7.4	0.43		
8K	0.87	0.07	0.89	0.1	59.5	3.53		
64K	0.99	0.19	1.11	0.31	476.3	28.20		
256K	1.43	0.63	1.9	1.1	1905.2	112.8		
1M	3.1	2.31	5	4.2	7621.0	451.3		
2M	5.3	4.57	9.15	8.35	15242.1	902.6		

(*M*: Message, *G*: Generation, *V*: Verification)

Table III shows the performance of four security classes used in the notarial methodology. We only consider the

performance for messages less than 1K bytes in NR-SC4 due to the performance of NR-SC4. In Table III, the performance of security classes seems to be proportional to the level of security classes in generation and verification of messages. Especially, NR-SC3 shows a significant performance falling-off in signing and verifying a message as the message size is increased. When a message size is greater than 1K bytes, it is required to improve its performance by employing NR-SC3-V.

B. Effects of Dynamic Mapping Rules

In this experiment, one of the parameters is the effect of the dynamic mapping rules. For the measurement of its performance effect, two variables are selected: (1) message size and (2) network situation. The message size is related to the effect of the multiple message digest method in NR-SC3-V. NR-SC3 is the most sensitive class to the increase of message size. First, to model the parameter of the message size, we specified dynamic message digests ($|MD_{dynamic}|$). To show the effect of dynamic mapping rules in terms of message size, we measure the notarial transaction using only NR-SC3 and NR-SC3-V. When the number of message digests ($|MD_{dynamic}|$) varies from $|MD_{dynamic}|=2$ to $|MD_{dynamic}|=4$, the performance of notarial methodology under the dynamic mapping rules (Figure 2-b) is almost 100 times better than that of the original notarial methodology using NR-SC3 (Figure 2-a) where messages range from 8 bytes to 64K bytes. Using the dynamic mapping between an environmental factor (i.e., message size) and the message digest mechanism in NR-SC3-V, we have remarkable improvement.

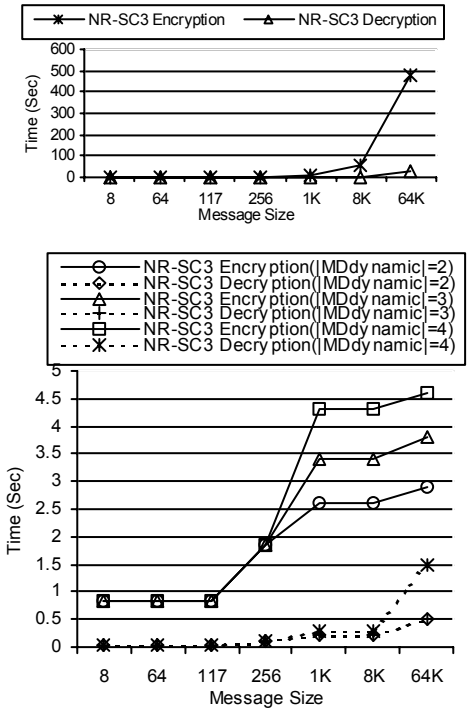

Figure 2. (a) Execution time of NR-SC3 and (b) that of NR-SC3-V

Second, to model the parameter of network traffic, we specify message delivery time (*T2*) as follows: When the message round trip time for the previous message is very slow, the best improvement can be achieved through a dynamic decision rule. In a heavy network traffic situation, security classes may be degraded to the lower level of security if the dynamic decision rule is allowed. In Figure 3, the performance of the notarial methodology under the dynamic decision rule (Figure 3-b) improves almost 100 times than that of the original notarial methodology (Figure 3-a) in the heavy network traffic situation. The performance of the notarial methodology is compared with that of the original notarial methodology where messages range from 8 bytes to 64K bytes.

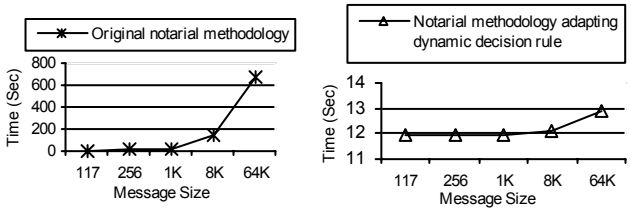

Figure 3. (a) Performance of original notarial methodology and (b) performance of notatial methodology adapting dynamic decision rule

C. Comparison of SET and Notarial Methodology

Now we compare the performance of our notarial methodology with that of SET. We conducted two experiments. First, we have a performance test on the message exchange mechanism of each system by applying it to a notarial transaction process. Among four security classes in our notarial methodology, NR-SC1 is similar to the security technique of SET in terms of a degree of security. Thus, we measure the performance of notarial transaction with NR-SC1 and SET. In details, NR-SC1 is more secure than the security technique of SET because NR-SC1 uses two different shared keys to encrypt and decrypt a message while SET uses one single shared key. Figure 4 shows NR-SC1 is faster than the security technique of SET. This is because a new-shared key and a certificate of participants are required whenever a message is exchanged in SET.

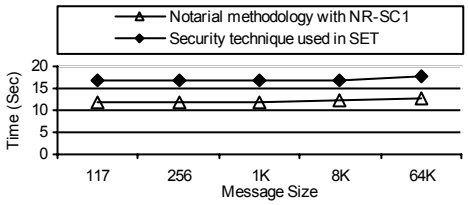

Figure 4. Performance of NR-SC1 and SET

Second, we compared the performance of security technique used in SET with the performance under the dynamic decision rule using NR-SC1 through NR-SC4 in Figure 5. Security levels in security classes are much stronger than the security technique used in SET. In this experiment, we measure the performance of the dynamic decision rule only for the growth of the message size. Given our discussion above on the dynamic decision rule with the multiple message digest method of NR-SC3-V, we expect that the number of message digests

affects the result of the experiment on message encryption and decryption. The performance of the notarial methodology in Figure 5 is almost the same as the security technique in SET. In particular, the notarial methodology with $|MD_{dynamic}| = 2$ is slightly better than in SET. For $|MD_{dynamic}| = 3$, it is almost same as in SET. For $|MD_{dynamic}| = 4$, it is slightly worse than in SET. However, in a degree of security, the performance of our notarial methodology is better than that of SET.

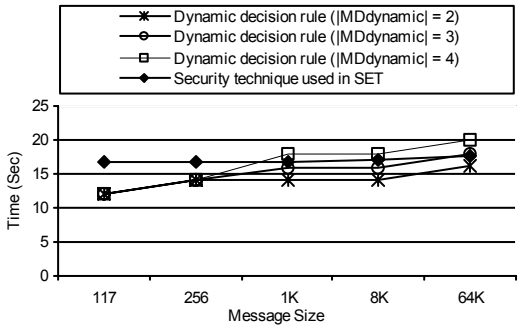

Figure 5. Performance of notarial methodology under dynamic decision rule and SET

D. Comparison with SSL, SET, and NA Framework

TABLE IV. COMPARISON AMONG SSL, SET, AND OUR NA FRAMEWORK

	SSL	**SET**	**NA Framework**
Number of participants	Two (*C*, *M*)	Four (*C*, *M*, *CA*, *PG*)	Five (*C*, *M*, *CA*, *PG*, *NA*)
OSI layer	Transport	Application	Application
Distribution of session key	RSA	RSA	RSA
Message encryption	Yes (DES, IDEA, RC2, RC4, RC5)	Yes (DES)	Yes (DES, 3DES, RSA)
Digital signature	RSA, DSA	RSA	RSA
Hash function	MD5, SHA-1	SHA-1	SHA-1
Certificate Issuer	*M*	*CA* (Trusted third party)	*CA* (Trusted third party)
Non-repudiation of origin	No	Yes (Supported by digital signature)	Yes (Supported by digital signatures)
Non-repudiation of delivery	No	No	Yes (Supported by random numbers)
Non-repudiation of evidence (or receipt)	No	Signed transaction messages	Yes (Signed transaction messages and a notarial deed)
Non-repudiation of submission	No	No	Yes (Supported by digital signatures and keep-alive time)
Prioritized security technique	No	No	Yes (Four security classes)
Adaptive mechanism	No	No	Yes

Table IV compares three frameworks (SSL, SET, and our NA framework) in terms of participants, security techniques, and non-repudiation of service. There are several security frameworks including SSL, S-HTTP (Secure Hyper Text Transfer Protocol), iKP (Internet Keyed Payment Protocol), and SET (Secure Electronic Transactions) [10]. All these frameworks have a common limitation for a successful e-commerce software framework, a lack of non-repudiation security service. Among these frameworks, SET is the only security framework partially addressing non-repudiation. SET partially supports non-repudiation of service due to digital certificates and a dual signature. This is a weak form of non-repudiation of service in contrast to a strong form of non-repudiation of service that our NA framework can provide with a notarial deed. In SET, we need to keep track of transactions to collect evidence. Even if we are able to trace transactions and collect evidence in SET, it is difficult to determine the order of evidence. In SET, we may loose some important evidence and it is difficult to verify the evidence in a timely manner due to the size of search spaces.

VI. CONCLUSION

In this paper, we propose a software framework for non-repudiation services in e-commerce. Our framework is interested in a successful completion of e-commerce transactions. Not only merchants' right but also customers' right on an electronic transaction are protected. Thus, we can prevent any potential disputes among participants. The proposed framework is based on the dynamic mapping mechanism supported by prioritized security classes. The security classes incorporate the security level of cryptographic techniques. The dynamic mapping mechanism improves e-commerce transactions through the security classes and the dynamic decision rules. We have demonstrated that the security classes based upon the dynamic mapping mechanism incorporating a dynamic decision rule improve the performance of our notarial methodology.

REFERENCES

[1] MasterCard and Visa, SET: Secure Electronic Transaction Specification – Book 1: Business Description, Version 1.0 May 31, 1997.

[2] *ISO 10181-4: Information technology – Security frameworks for open systems: Non-repudiation framework*, International Organization for Standardization, 1989.

[3] J. Zhou and D. Gollmann, "Evidence and Non-repudiation," *Journal of Network and Computer Applications*, vol. 28, no. 5, pp. 50-60, 1998.

[4] D. Wagner and B. Schneier, "Analysis of the SSL 3.0 Protocol," Proc. the Second USENIX Workshop on Electronic Commerce, 1996, pp. 29-40.

[5] RFC 793, Transmission Control Protocol Darpa Internet program Protocol Specification, Sep. 1991.

[6] V. Jacobson, "Congestion Avoidance and Control," *Computer Communication Revie*w, vol. 18, no. 4, pp. 314-329, Aug. 1988.

[7] P. Kan and C. Patridge, "Improving Round-Trip Time Estimates in Reliable Transport Protocols," *Computer Communication Revie*w, vol. 17, no. 5, pp. 2-7, Aug. 1987.

[8] W. Ford and M. Baum, *Secure Electronic Commerce*, Prentice Hall, 1997.

[9] N. Barron, *RSAEuro Technical Reference*, RSAEuro Co., Nov. 1996.

[10] A. Mani, "Securing the commercial Internet," *Communications of the ACM*, vol. 39, no. 6, pp. 29-35, June 1996.