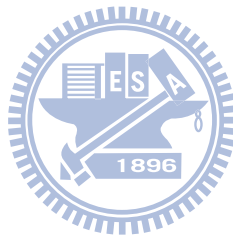


Chapter 1 Introduction

1.1 Research Motivation

An electronic business (e-Business) application supports an automatic, efficient execution of the business processes understood as a set of logically related tasks executed to accomplish a well defined business outcome [1]. Hence, a firm keeps e-Business applications (such as e-Commerce, Supply Chains, and e-Services) running to keep cost down and efficiency. From the current literature[2-4], e-Business concepts fall in many categories such as:

- Business to Consumer (B2C)
- Consumer to Business (C2B)
- Business to Business (B2B)
- Government to Citizen (G2C)
- Citizen to Government (C2G)
- Business to Employ (B2E, Intra Business)



In all these categories, the users of an e-Business application must execute it with online business processes and online intra-services. This means there are no face-to-face operations and all transactions are executed electronically using a communication network.

Organizations usually base electronic business applications on two main systems:

electronic document management and workflow. These systems work together to perform business processes automatically. The electronic document management system controls the document life cycle (such as creation, approval, distribution, and storage) and the workflow system parses tasks from the documents and dispatches the tasks to the participants based on business logic.

Without face-to-face contact among participants in an e-Business environment, the participants need three basic security mechanisms to create trusted relationships:

1. Encryption: provides confidentiality for each document life cycle in the electronic document management system.
2. Hash Functions: ensures the correctness of content of a piece of information. Hash Functions usually integrates with other cryptographies to ensure that no data should be corrupted in an electronic business application.
3. Digital Signatures: prevents unauthorized modification and repudiation. Digital signatures are also related to legitimate pattern of operations in data access in a business process.

The three components above reduce the risk of e-Business application modules. Generally, four major modules make up e-Business applications: logging, document verification, document processing, and document dispatch. A conceptual layout of e-Business applications functions is given in Figure 1-1.

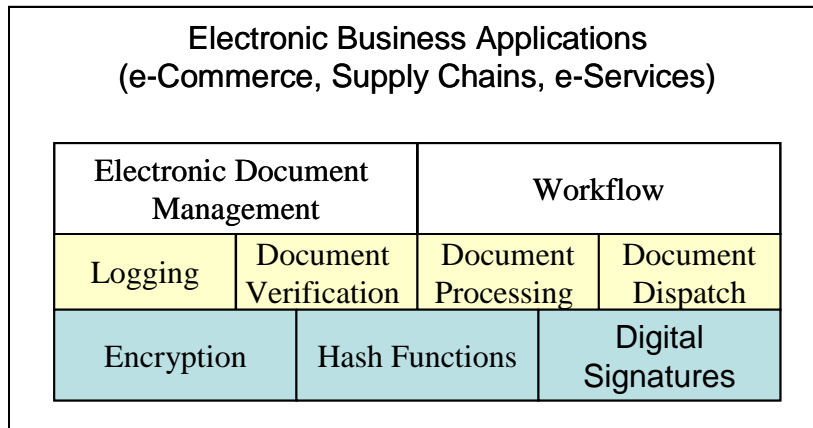


Figure 1-1: Conceptual layout of e-Business applications

The logging module and the document verification module support user (i.e. the participants of e-Business Applications) authentication and document authentication, and can be one of the seven classes of authentication mechanisms that defined in the IAB (Internet Architecture Board) [5, 6]. The seven classes of authentication mechanisms are as follows:



- Password in the clear
- One-time passwords
- Challenge/response
- Anonymous key exchange
- Zero-Knowledge password proofs
- Server certificates plus client authentication
- Mutual public key authentication

A discussion about various authentication mechanisms of the IAB[5] indicates all authentication mechanisms need the three security components listed earlier to guarantee the

following security properties:

- The identity of participants (Authentication)
- The document transmitted has not changed (Integrity)
- The confidentiality of the information in transit (Confidentiality)
- Protection against denial of transaction by one of the participants (Non-repudiation)

The Document Processing module and the Document Dispatch module provide the functions such as [7]:

- The workflow manipulates confidential or sensitive document.
- The document moves between a set of participants or agents.
- Enforces authorization procedures for different resources in the workflow.
- Store the documents remotely and securely.

To offer the security properties mentioned above for the workflow, the Document Processing and the Document Dispatch module also require the three basic security components listed earlier.

Researchers have proposed solutions such as the following to achieve the security properties mentioned above:

- Lazaro et al. proposed authenticated encryption schemes for encryption and the hash functions, to speed up storing and accessing documents remotely [2, 8, 9]. In [8] and [9],

an authenticated encryption can use block ciphers, such as AES [6], with HMAC (The Keyed-Hash Message Authenticated Code)[10]. In these authenticated encryption schemes, a nonce is generated as an Initial Vector (IV) and computes a HMAC for a plaintext block. The plaintext is then encrypted and stored. By contrast, an authenticated encryption based on incremental encryption [1, 11, 12] provides document integrity without additional computation or other cryptographic primitives. In an incremental mode of encryption, updating the cipher of a document is much faster than re-encryption of the entire document.

- For the authentication, SET[13-15] and Cyberspace Center [16] proposed flexible authentication and authorize procedures based on a dual signature scheme for e-Commerce and remote file management. However, the dual signature itself limits the number of participants in the procedure.

Although these schemes solved some issues, researchers must further investigate other issues to meet the needs of increasingly complex business logic.

- In block ciphers, designers often ignore the security of encrypting multiple blocks and padding the last block (the security of encrypting one block is well known).
- The dual signature scheme itself limits the number of participants in authentication and authorization procedures.

Therefore, this dissertation focuses on the following issues: unforgeable incremental

encryption, information leakage in padding last block, and flexible authentication and authorization procedures.

.1.2 Concepts of e-Business

Research indicates an electronic business is a set of procedures, mechanisms and computer programs not only to authenticate the participants and the source of information (documents), but also to authorize resources for different participants [3, 17]. Studies categorize e-Business applications as Business to Business (for electronic trading), Business to Consumer (online retailing), Business to Employment (management of business logic within businesses or organizations), Consumer to Business (for electronic submission of individual tax returns), and Consumer to Consumer (online auctions) [4, 7]. Figure 1-2 depicts a typical e-Business application framework.

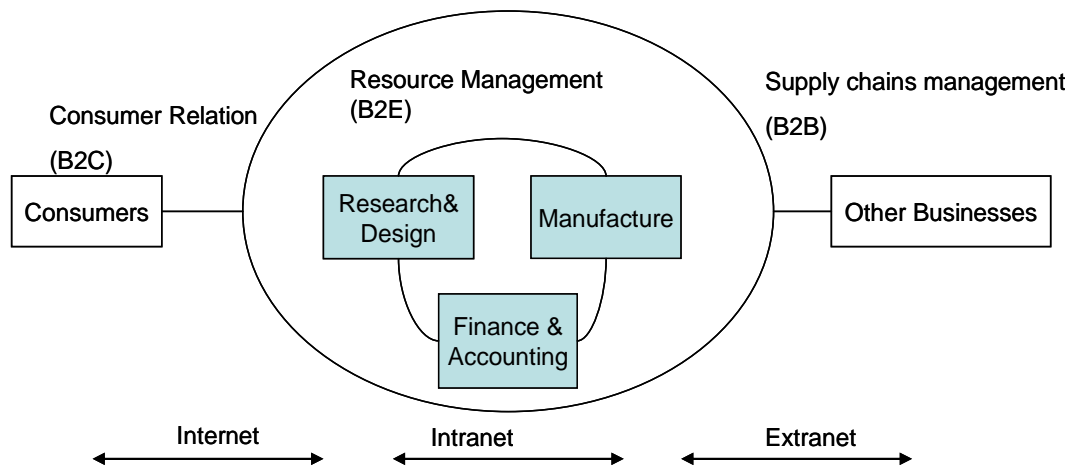


Figure 1-2 A Typical e-Business Framework

As previously mentioned, the following characteristics distinguish an e-Business environment from the original business environment:

- No face-to-face operations without established interpersonal trust among participants,
- All e-Business transactions that are performed electronically with the use of communication networks,
- The participants involved in through Intranet, Extranet, and Internet, and
- An encryption storage system in which the underlying data is constantly changing yet encrypted versions must be stored.

Unfortunately, current solutions do not fit all e-Business environment conditions.

Therefore, this dissertation develops the following new schemes for various characteristics mentioned above:

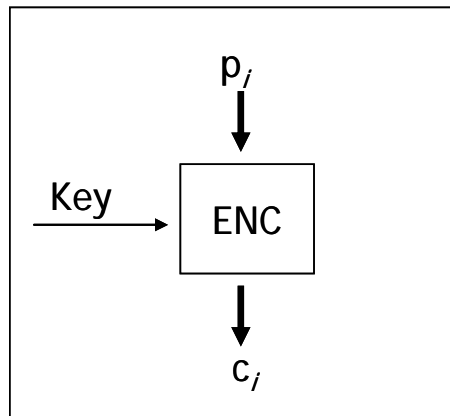
- An unforgeable incremental encryption
- A low information leakage padding scheme
- A flexible authentication and authorization scheme.

1.3 Concepts of Block Ciphers

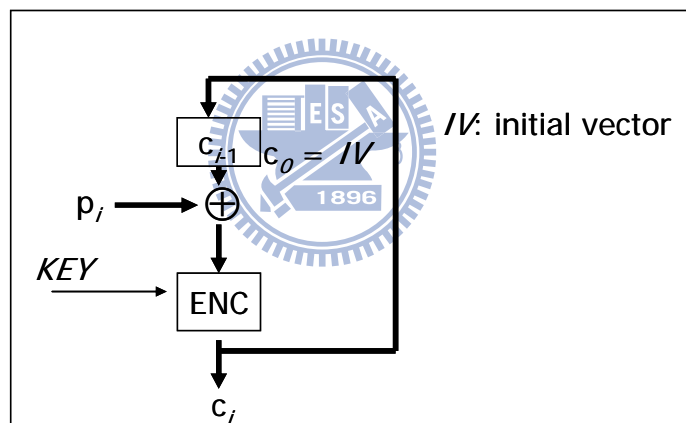
A block cipher is a secret-key cryptosystem that encrypts plaintext to generate ciphertext. The plaintext must be divided into fixed-size n -bit blocks, where n is the block-length and the generated ciphertext is a sequence of n -bit blocks as well. This means the input plaintext must be processed before encrypting. The processes include partitioning the input plaintext into a sequence of n -bit blocks, encrypting each separately, and appending a padding string to the last block if needed.

In fact, the former one is called a mode of encryption operation. The four most common modes are ECB (Electronic Codebook), CBC (Cipher-Block Chaining), CFB (Cipher FeedBack), and OFB (Output FeedBack) [6]. ECB mode divides a plaintext into a sequence of n -bit blocks p_i . An algorithm, ENC, encrypts each block p_i with the same key KEY . By contrast, CBC modes x-ors each cipher block c_i with the next plaintext block p_i before being encrypted with same key KEY . They are depicted in figure 1-3 (a) and (b) respectively. In OFB and CFB, a keystream is generated, and then is x-ored with each plaintext block p_i which

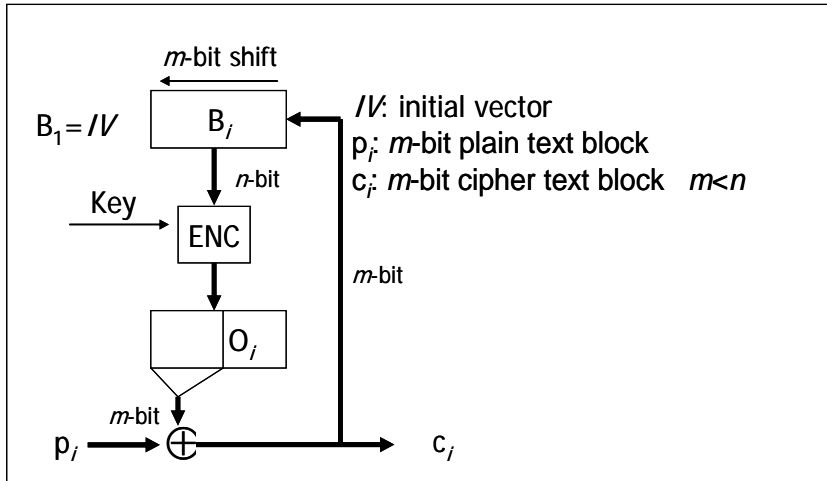
block-length is m , where $m \leq n$. They are depicted in 1-3 (c) and (d) respectively.



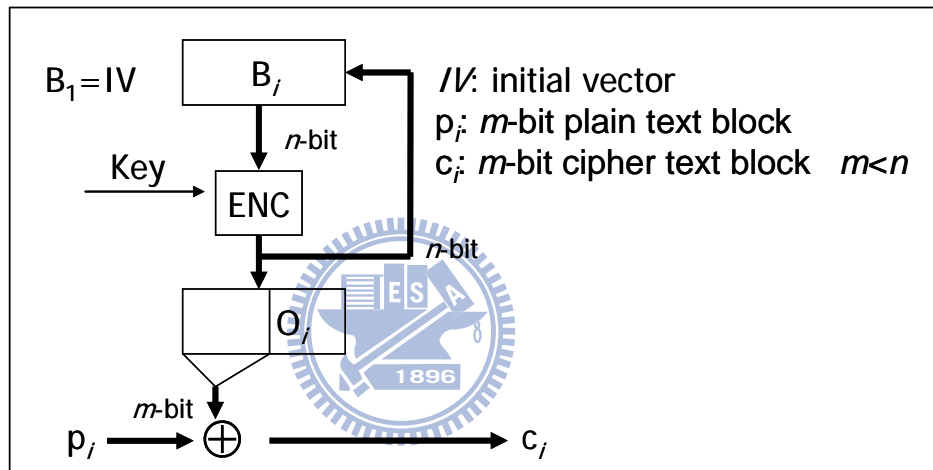
(a) Electronic Codebook (ECB) mode



(b) Cipher-block Chaining (CBC) mode



(c) Cipher-feedback (CFB) mode



(d) Output feedback (OFB) mode

Figure 1-3: Four common encryption modes

In ECB and CBC modes, the ENC algorithm directly inputs the plaintext. This is not done in CFB and OFB modes. Therefore, ECB and CBC encryption modes require a multiple of the block length as plaintext input. This results in a padding scheme. The most common padding The most common padding scheme is defined in PKCS#5 [18, 1999 #5] and RFC2630 [19]. Usually, the CBC-mode requires padding of its block ciphers, and takes the

padding rule from RFC2630. The padding rule is taken from RFC 2630 and depicted as follows.

The padding rule says the input shall be padded at the trailing end with $n - (\text{LEN} \bmod n)$ octets all having value $n - (\text{LEN} \bmod n)$, where LEN is the length of the input. One of the following strings pads the input at the trailing:

01 -- if $(\text{LEN}) \bmod n = n - 1$

02 02 – if $(\text{LEN}) \bmod n = n - 2$

...

$n n \dots n n$ -- if $(\text{LEN}) \bmod n = 0$



The padding string can be removed unambiguously since all input is padded, including input values that are already a multiple of the block size, and no padding string is a suffix of another. Because CFB and OFB modes operate as a stream cipher, they do not need a padding scheme.

Example 1 Suppose that the encryption algorithm is DES, and so block-length = 8. Then

ASCII plaintext: Hello

HEX plaintext: 48656C6C6F

Input block: 48656C6C6F030303

Example 2 Suppose that the encryption algorithm is AES-128, and so block-length=16. Then

ASCII plaintext: Hello

HEX plaintext: 48656C6C6F

Input block: 48656C6C6F0B0B0B0B0B0B0B0B0B0B



1.4 Concepts of Incremental Encryption

The results of incremental cryptography algorithms can be quickly updated for a modified document, rather than having to re-compute the algorithm from scratch. Very efficiency improvements can be achieved in settings where cryptographic algorithms, such as encryption or signatures are frequently applied to changing documents. One such setting is authentication tags for an encrypted file system in which the underlying data is constantly changing.

Incremental encryption views , a document, D , as a sequence of blocks p_1, p_2, \dots, p_n .

In the incremental encryption setting, a document D is viewed as a sequence of blocks $p_1,$

p_2, \dots, p_n . Researchers have considered various modification operations for incremental cryptography. A generic modification operation, M , can be one of the following types of modifications operations:

- $M = (\text{delete}, i)$ delete block i of the document D .
- $M = (\text{insert}, i, p)$ insert block p between the i^{th} and $(i+1)^{\text{th}}$ blocks of the document D .
- $M = (\text{replace}, i, p)$ change the i^{th} block of the document to p .

Document D can be modified as follows:

$$D\langle M_1, M_2, \dots, M_k \rangle = (\dots((D\langle M_1 \rangle)\langle M_2 \rangle)\dots)\langle M_k \rangle,$$

where M_i is one of the types of modification operations, $i=1,2, \dots, k$.

Incremental encryption operates each block independently like ECB mode, but the incremental encryption needs more complicated cryptographic schemes; The following session discusses the details.

1.5 Concepts of Dual Signature

The SET (Secure Electronic Transaction) protocol develops a dual signature is to guarantee the integrity of transaction data and the privacy of customers both [13-15]. The dual signature is illustrated in figure 1-4. The SET protocol was defined to ensure the security of credit card payment through Internet, and so a customer generates a purchase request including order

information (OI) and a payment authorization including payment information (PI) sending to a merchant and an acquirer. The order information but the payment information is provided to the merchant, and the payment information is encrypted with the public key of the acquirer. Thus, the payment information but not the order information is available to the acquirer, and the merchant cannot get the customer's financial information. Nevertheless, the dual signature provides the non-repudiation and integrity of a transaction. The customer computes the message digests $H(OI)$ and $H(PI)$ independently. The dual signature is the signed message digests of $H(OI)$ and $H(PI)$. Since the message digests can not retrieve any information about OI or PI, this approach protects customer privacy.

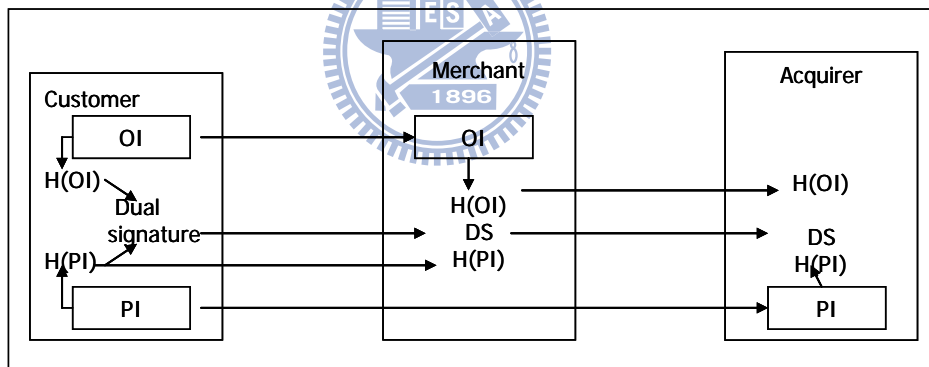



Figure 1- 4: The dual signature links the order information and authorization while protecting privacy in SET

1.6 Problems Statements

Without face-to-face interaction, electronic business applications participants encounter

certain issues. First, a virus or a malicious attack may tamper with the encrypted files on remote hosts. In order to avoid the risks, the participant checks the integrity of the encrypted files before executing. This means the participant must re-compute some authentication tags, e.g. MAC (Message Authentication Code) or hash values of the encrypted files, for each file, and so incurs inefficiency when the local memory is not large enough. In fact, an incremental encryption scheme (for block cipher), which can guarantee ciphertext integrity, has been designed to solve the issues. There are two kinds of such schemes. One uses an incremental mode together with an incremental MAC (Message Authentication Code) of ciphertext [12] to achieve the goals. This kind of schemes is attractive for its simplicity, but it is inefficiency because incremental MAC algorithms are inefficient in practice. The other is a new incremental encryption scheme which can guarantee the cipher integrity at the same time [20].



In [20], the authors proposed the RPC authenticated encryption scheme supporting the properties of incrementality and confidentiality. Although this second approach is more efficient than others are, it still has integrity problem. More precisely, this dissertation will describe a forgery attack on the RPC scheme and proposed two methods to strengthen the PRC scheme.

The signature verification process of signed files creates the burden of certificate verification or network delay. The two steps of the signature verification process, signature value verification and certificate validation, create the burden. Certificate validation includes

two phases, certificate path creation and validation. Certificate path creation generates a chain of cross-certificates and CA certificates running from the trust anchor of the relying party to the certificate of end-entity. The process can dynamically create a certificate path each time one is needed or construct the path once and store it. Certificate path validation is the process of investigating each certificate that constructs the certificate path and consulting CRLs (Certificate Revocation List) issued by CAs (Certificate Authority) to determine the validity status of each certificate. It is desired that all CRLs are stored in local and all the certificates in the certificate path are validated in real-time at the beginning of each transaction, yet this is difficult for some types of mobile device due to the limit memory. This implies that the way to on-line check the status of certificates is suitable for mobile user and so network delay will follow the burden of certificate verification. Thus, this study proposes a scheme in which the signature verification can be done without plaintext by trusted third party to eliminate the network delay.

Finally, secure transmissions are weak because of padding oracle attacks [21, 22] on Transport Layer Security (TLS) used in HTTP connections (*i.e.* https, or authenticate the remote user, for example, an EAP-TLS like authentication protocol). The attack assumes a padding oracle receives a ciphertext and answers whether or not the corresponding plaintext is correctly padded. The attack works because of the information leakage in CBC padding schemes. Several CBC-mode encryptions in well-known standards are potentially vulnerable

to this attack [22]. Therefore, this study proposes a new padding scheme in which padding strings are key-dependent and almost random so the CBC encryption mode can defeat padding oracle attacks.

1.7 Research

Due to the characteristics discussed in section 1.2, e-Business applications need security components, depicted in figure 1-1, to ensure security properties such as authentication, integrity, confidentiality, and non-repudiation. Current solutions for security components of e-Business applications do not meet all conditions. Hence, this dissertation proposes schemes to improve the barriers. The proposed schemes include the following components:

- For block ciphers, this study proposes two schemes for encryption modes and padding the last block. The encryption mode scheme analyzes and improves the unforgeable encryption of [15]. In contrast to the current authenticated encryption schemes such as [13-15], the unforgeable encryption more readily meets the condition of frequent and small document changes. The padding scheme proposes a new padding with low information leakage. The padding scheme prevents padding oracle attacks. Such attacks are useful for the SSL/TLS (Secure Socket Layer/Transport Layer Security) protocol, used for building a secure channel, such as hypertext transfer protocol secure (https) https, and for some authentication protocols such as EAP-TLS.

- For authentication procedures, the current solutions, such as SET and Cyberspace Center [16], use a dual signature to satisfy integrity, authentication, non-repudiation, confidentiality, and relationship requirements [15]. However, this restricts the number of participants to two. To avoid this restriction, this dissertation proposes an orthogonal signature scheme that works for multiple parties, each having more than two participants, and a flexible authentication procedure based on an orthogonal signature.

1.8 Thesis Organization

Chapter 2 of this dissertation surveys previous studies and briefly reviews SSL/TLS protocols.

Chapter 3 describes a forgery attack on the RPC incremental unforgeable encryption scheme

and proposes two methods to strengthen the RPC scheme for defeating the attack. Chapter 4

proposes an orthogonal signature scheme and applies it to a secure file transferring system.

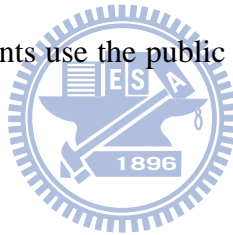
Chapter 5 proposes a new random padding scheme for symmetric key block encryption. In the

scheme, padding is key-dependent and almost random. Finally, chapter 6 draws conclusions.

Chapter 2 Related Work

2.1 Secure Electronic Transaction

The SET (Secure Electronic Transaction) is a standard protocol, which secures credit card transactions over insecure networks, for e-Commerce. A SET system includes six participants such as a cardholder, merchant, issuer, payment gateway, and certification authority (CA) depicted as Figure 2-1. The SET protocol assumes the presence of certification authorities. Each of the participants must possess a certificate, which contains the participant's public key, issued by a CA. Then all participants use the public key and the corresponding private key to do secure transactions.



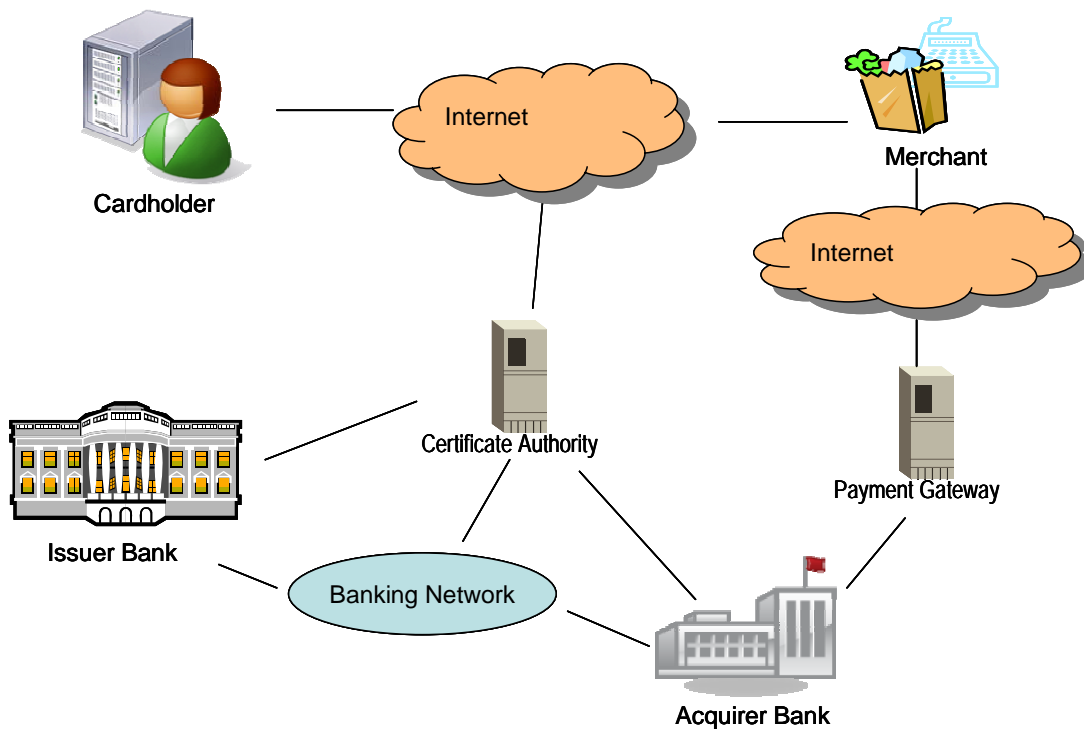


Figure 2-1 SET System Participants

The SET protocol employs the standard PKCS#7 [19] and dual signature to design an authentication procedure satisfying the following requirements:

- Integrity: The SET protocol ensures the integrity of all transaction data using dual signatures.
- Authentication and Nonrepudiation: Dual signatures and certificates are used to verify that a cardholder is a legitimate user of a valid account.
- Confidentiality: The SET protocol protects the cardholder account and payment information without transport security mechanism.
- Relationship: The SET protocol provides authentication that a merchant can accept credit

card transactions through its relationship with a financial institution.

The format of the authentication procedure is depicted as Figure 2-2.

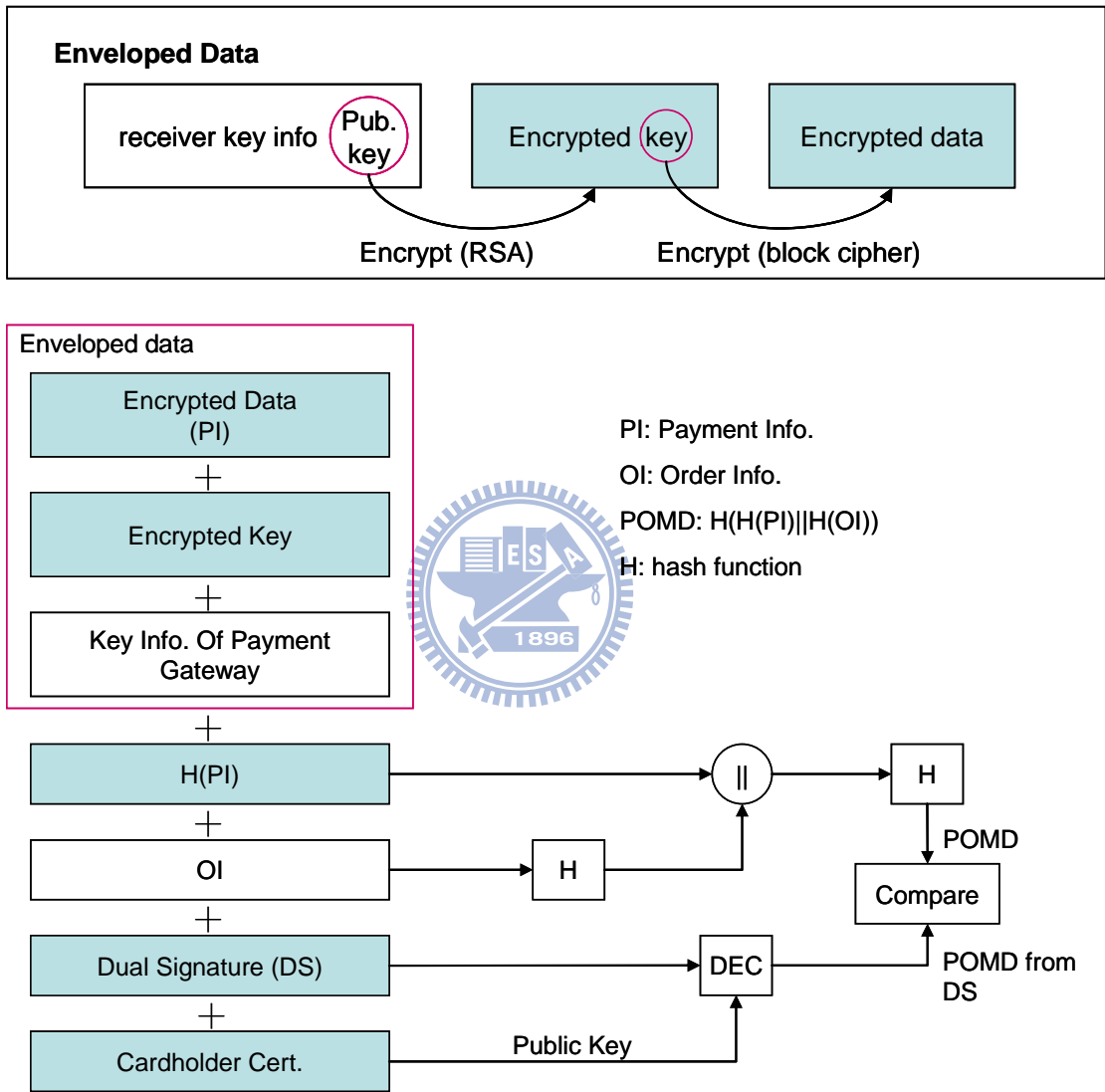


Figure 2-2 Authentication Procedure of the SET

From figure 2-2 and the discussion of session 1.3, the authentication procedure of the SET protocol preserves the privacy and the confidentiality at same time.

2.2 A Secure Database Encryption Scheme

The cryptographic technique in early stage is not often used in database because that time cost involved in encrypting and decrypting data items can greatly degrade database system performance. However, most common database systems have leak-sensitive data no matter what the degree of security. An attacker can penetrate a database, since a database system processes large amount of data in complex ways. Encrypting sensitive data in a database system is a compromise solution achieving both efficiency and security.

The authors in [23, 24] provided solutions to the security issues of field based protection and proposed a multilevel database project *i.e.* encryption at various levels such as table, attribute and field levels. The authors in [23, 24] tried to solve database integrity issue using cryptographic checksum. Differing from the above schemes, the solution of [25] is an integral solution that includes structure, key management and implementation procedures.

The proposed scheme of [25] adopts a two-level relational database system, where users are assigned to either of two levels, low and high. Users in low level can only access the non-sensitive data of database, while those in high level can access both non-sensitive and sensitive data of database. The semantic structure of their model is depicted as figure 2-3 and is divided into three layers: The first layer is the user interface layer containing two blocks, one for low level users and the other for high level users. The second layer is the database management layer which contains two blocks, one that implements the mandatory access

control (MANAC) to the database and other that includes tamper-free controller.

The functions of the controller (KC):

1. Encrypt sensitive data before storing in the database.
2. Decrypt ciphertext in response to user queries.
3. Perform integrity-check on returned data.

The bottom layer includes the database. In order to facilitate the fast retrieval of data, the database system stores non-sensitive data in clear while sensitive data are stored in encrypted form. The first field of every record uniquely identifies the record, and serves as its primary key. The primary key should not be confused with cryptographic keys used to decrypt ciphertext.

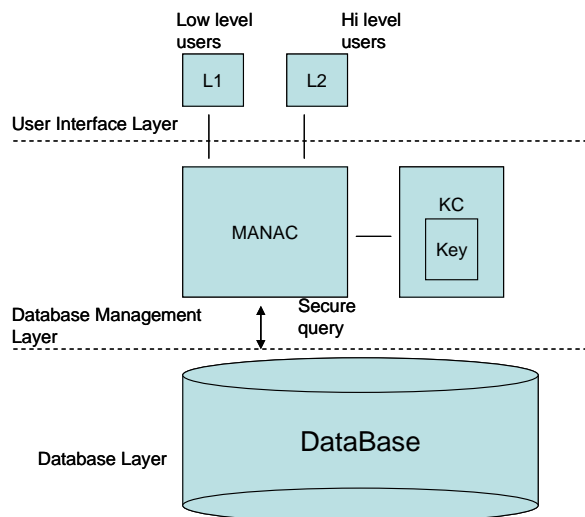


Figure 2- 3 Secure Database

In [25], the sensitive data elements are encrypted/decrypted using block cipher technique. Encryption is performed in CBC mode if elements block size. The system incurs inefficiency whenever there are frequent, yet small, changes in sensitive data elements. Intuitively, this issue can improve using the un-forgeable encryption mentioned above. The details are discussed in the following.

2.3 A Secure File Server

A secure file server generally has the following features:

1. Stores and forwards the secure files from sender to recipient.
2. Transfers fully encrypted highly sensitive data between any two locations.
3. Transfers safely data in compliance with required levels of authentication including digital certificate.
4. Sends and receives files using a variety of clients and protocols that support industry standards, including web browsers, FTP over SSL [26, 27].

Secure file sever features always contain the signature verification process. The process includes:

1. Verify signature value.
2. Validate the certificate chain.
3. Check certificate revoking list.

Constrain mobile device resources and unstable network results in the certificate verification burden process consuming more power and network delay for users. Thus, this work proposes a scheme [16] which allows the server to capably verify signature without plaintext. The semantic structure is depicted as figure 2-4.

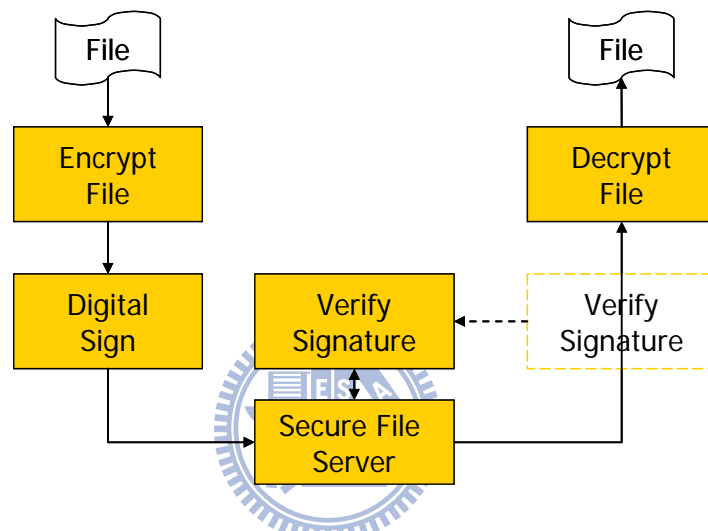


Figure 2-4: Secure file server

We note that their scheme guarantees the end-to-end data confidentiality and real-time signature verification on the secure file server.

2.4 Attacks on SSL/TLS with block ciphers in CBC-mode

The SSL/TLS (Secure Sockets Layer/Transport Layer Security) [26, 27] are design to run in a user-level process, and used to establish a secure connection between a client and a server,

initiated by the client. In addition to secure the connection, the TLS is also used in wireless authentication protocols, *e.g.* EAP-TLS, EAP-TTLS and PEAP [28, 29], recently. The TLS consists of six sub-protocols:

The handshake protocol, the change cipher spec protocol, the application protocol, alert protocol, and record layer protocol.

The handshake protocol is used to negotiate the cryptographic parameters and establish a master secret KEY used to derive other session keys. It consists of nine messages:

1. ClientHello: this message contains a random value R_{Client} .
2. ServerHello: this message contains a random value R_{Server} .
3. ServerHelloDone: this message indicates that the server is finished sending handshake messages.
4. ClientKeyExchange: this message contains a pre-master secret S_{Client} encrypted with the server's public key.
5. ServerKeyExchange: this message contains the signature (signed with a long term key which is used for signing only) of server's public key which using to encryption.
6. CertificateRequest: this message is sent by server to request that the client send a certificate and authenticate.
7. Certificate: this message contains one or more certificates.

8. CertificateVerify: this message is sent by the client to prove it owns its private key.
9. HandshakeFinished: this message ensure integrity of the exchange and proving knowledge of the key.

The change cipher spec protocol indicates that all messages following this will protect with the cipher agree upon in the handshake protocol. It consists of a single message ChangeCipherSpec.

The application data protocol handles the transmission of data between the application and TLS.

The alert protocol handles SSL-related alerts and error-messages.

The record layer protocol provides confidentiality and message integrity for the SSL/TLS connections.

Figure 2-5 depicts the simplified SSL/TLS protocol.

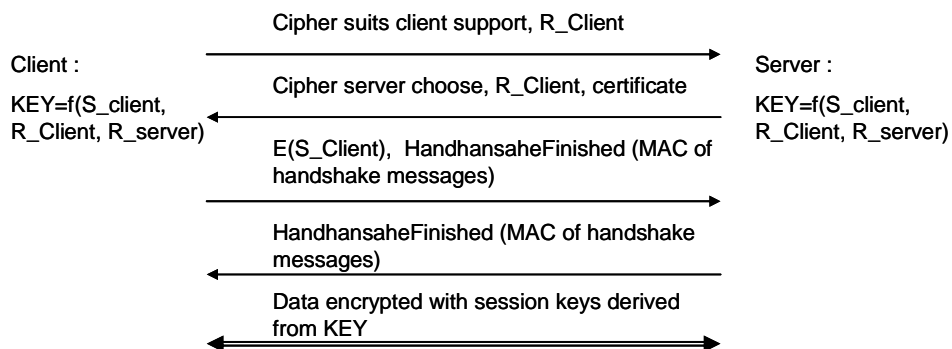


Figure 2- 5: Simplified SSL/TLS protocol

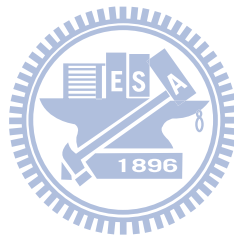
The attacks on SSL/TLS themselves can be distinguished in several different types. Here, we present an overview of side channel attacks on SSL/TLS. A side channel attack is any attack based on information gained from the physical implementation of a cryptosystem, rather than theoretical weakness in the algorithm.

Block ciphers in SSL/TLS are encrypted in CBC-mode. For the encryption with a block cipher, the plaintext must have the multiple of the block size. Otherwise, a padding string will be appended to plaintext to make it fitting the requirement needed by block cipher. The padding scheme used in SSL is the scheme of PKCS#5 [18, March 1999 #5].

When the server receives the cipher text and then answers whether or not the corresponding plaintext is correctly padded. In the negative case, the server sends an alert message and terminates the connection. Hence, in [22], the authors present a side channel attack based on the alert message. They use the server as a query oracle to check if a modified cipher text has a correct padding. By querying to the oracle with appropriate modified initial vector and the cipher text, an attacker is possible to invert the underlying block cipher.

Specifically, the attacker sends a CBC mode cipher text block (C) including the initial vector (IV) to the padding oracle. If the answer indicates that the padding string P is correct. Consider the formula: $IV \oplus D(C) = P$ where $D()$ is the decryption function. That is, $D(C) = IV \oplus P$. Since IV is known by the attacker. Thus, if the attacker also knows the padding string P , then the plaintext of C ($=D(C)$) is discovered. Therefore, simple CBC mode

encryption with the padding schemes that use constant or regular padding string is vulnerable to this attack.



Chapter 3 Un-forgearable Incremental Encryption Scheme

The investigation describe a forgery attack on the RPC (Related Plaintext Chaining) incremental unforgeable encryption scheme of [20]. The attack allows an adversary to forge a new ciphertext with probability $1/2$ using $2^{r/2}$ incremental update queries, where r is the parameter of random values used in the RPC scheme and is at most, half the block length of the block cipher used. However, the original analysis claims that the order of 2^r queries are needed. When applying the attack to the scheme using a block cipher with 128-bit block length and assuming $r = 48$ as suggested in the original article of the RPC scheme, the adversary can obtain a forgery with probability $1/2$ after 2^{24} update queries. Even in the case of 256-bit RPC scheme with $r=64$, the required number of queries is only 2^{32} . We also propose two methods to strengthen the RPC scheme for defeating the proposed attack.

3.1 RPC incremental unforgeable encryption

The notion of incrementality of cryptographic functions was proposed by Bellare et al. [1, 11, 12]. The concept is that for a document P and the corresponding cryptographic value C , having once applied modifications (e.g. insert, delete and replace) to P , the time to recompute C should be proportional to the amount of modifications done to P and less than the time

required to re-compute C from scratch. Thereby, one obtains much faster cryptographic primitives for environments where closely related documents are undergoing the same cryptographic transformation. Incremental encryption is one topic in the field of incremental cryptography. Buonanno et al. extended the notion to include the unforgeability property and proposed the RPC incremental unforgeable encryption scheme in [20]. That is, RPC is an encryption mode, in the symmetric-key setting, supporting the properties of incrementality, confidentiality, and unforgeability. Other researches on authenticated (unforgeable) encryption had been proposed [1, 11, 12, 28, 29], although they did not support the property of incrementality. The paper [20] was the first to give security definitions for incremental unforgeable encryption. Readers can refer to the paper for these definitions in detail.

The encryption algorithm of RPC can be described briefly as follows:

Algorithm $\text{RPCE}_k^{b,r}(P)$

For $i = 0$ to n

$r_i = \text{an } r\text{-bit random value } \in \{0,1\}^r$;

$C_0 = F_k(r_0 \parallel \text{start} \parallel r_1)$;

For $i = 1$ to $n-1$

$C_i = F_k(r_i \parallel p_i \parallel r_{i+1})$;

$C_n = F_k(r_n \parallel p_n \parallel r_0)$;

$r^* = \bigoplus_{i=1}^n r_i$;

$$C^* = F_k(r^* \oplus r^0 \parallel 0^{b-2r} \parallel r^*);$$

Return $C_0 \dots C_n C^*$.

In the above algorithm, $F_k()$ is the underlying block cipher with b -bit data block, and k is the secret key. The notation r denotes the amount of random padding. The document P is parsed into a sequence of $(b-2r)$ -bit blocks p_1, \dots, p_n . The notation \parallel means concatenation.

When inserting a new block B into the document P in position j , the algorithm chooses a new random value r'_j and then computes

$$C'_j = F_k(r'_j \parallel B \parallel r'_j),$$

$$C'_{j-1} = F_k(r_{j-1} \parallel p_{j-1} \parallel r'_j) \text{ and}$$

$$C'_{i+1} = C_i$$

for i from j to n .



Furthermore, new r_0 and r_1 are chosen at random, and the checksum r^* is recomputed. Thus, six new ciphertext blocks $C'_0, C'_1, C'_{j-1}, C'_j, C'_{n+1}, C^*$ have to be computed. A similar process occurs during a delete modification and a replace modification. We do not describe the operations for the two modifications (delete and replace) here because only the insert modification is used in the attack.

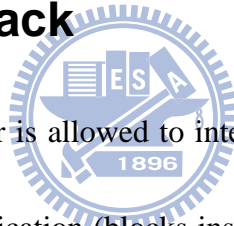
As described in the original paper [20] (Theorem 6 in [20]), the upper bound on the probability of getting a forgery ciphertext is

$$\frac{1}{2^{2r}} + O\left(\frac{q_{\text{total}}^2}{2^{2r}}\right) \quad (1)$$

where q_{total} is the amount of queries to an RPC oracle. The oracle receives a query containing an encryption operation or an incremental modification (insert, delete, and replace) operation to a document and then returns the corresponding updated ciphertext.

According to equation (1), the attacker obtains a forgery ciphertext with success probability $1/2^r$ after $2^{r/2}$ queries. However, using the proposed attack method, the attacker can get a forgery ciphertext with a success probability larger than $1/2$ after only $2^{r/2}$ queries. The success probability is far larger than the one claimed in [20].

3.2 The Forgery Attack



In the attack scenario, the attacker is allowed to interact with an RPC oracle which supports encryption and incremental modification (blocks insert, delete and replace) operations. The attack begins from an arbitrarily chosen document $P(= p_1, \dots, p_n)$ in length of n blocks (Each block has $b-2r$ bits as described in the previous section.) and the corresponding ciphertext $C(= C_0 \dots C_n C^*)$ gotten by querying to the RPC oracle. The attacker then randomly chooses a data block p_i from P . Let the content of p_i be X . In each attack step, the attacker sends a query to the RPC oracle for inserting an X into the document in a specific position. In fact, the position is $i+k$ in step k . That is, to insert X into the document P in position $i+1, i+2, i+3, \dots$, in sequence. Let $C_i^{(j)}$ denotes the $(i+1)^{\text{th}}$ block in the ciphertext sequence after the j^{th} attack step. The attacker can observe the ciphertext blocks $C_i^{(0)}, C_{i+1}^{(0)}, C_{i+2}^{(0)}$ to find possible

repetition. If the attacker finds that $C_{i+j}^{(j)} = C_{i+k}^{(k)}$ for some j and k ($k > j$), he can generate

valid cipher texts by the following formula without querying to the RPC oracle:

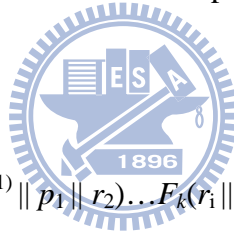
$$C_0^{(k)}, C_1^{(k)}, \dots, [C_{i+j}^{(k)}, \dots, C_{i+k-1}^{(k)}]^N, \dots, C_{n+k}^{(k)}, C^{*(k)}, N \geq 2 \text{ and } N \text{ is even.}$$

$[C_{i+j}^{(k)}, \dots, C_{i+k-1}^{(k)}]^N$ means to replicate $C_{i+j}^{(k)}, \dots, C_{i+k-1}^{(k)}$ N times.

More specifically, let us observe the ciphertext sequence: (The attacker does not know all the random padding values $\{r_i\}$.)

$$C^{(0)} = F_k(r_0 \parallel \text{start} \parallel r_1) F_k(r_1 \parallel p_1 \parallel r_2) \dots F_k(r_i \parallel p_i \parallel r_{i+1}) F_k(r_{i+1} \parallel p_{i+1} \parallel r_{i+2}) \dots C_n C^*.$$

After the first attack step, (an X is inserted in position $i+1$), the new ciphertext sequence will be as follows:



$$C^{(1)} = F_k(r_0^{(1)} \parallel \text{start} \parallel r_1^{(1)}) F_k(r_1^{(1)} \parallel p_1 \parallel r_2) \dots F_k(r_i \parallel p_i \parallel g^{(1)}) F_k(g^{(1)} \parallel X \parallel r_{i+1})$$

$$F_k(r_{i+1} \parallel p_{i+1} \parallel r_{i+2}) \dots C_{n+1} C^{*(1)},$$

where $r_0^{(1)}$, $r_1^{(1)}$ and $g^{(1)}$ are new random values generated by the RPC oracle in the first attack

step. By observing $C^{(0)}$ and $C^{(1)}$, the attacker obviously can get the two ciphertext blocks:

$$C_i^{(0)} = F_k(r_i \parallel p_i \parallel r_{i+1}) = F_k(r_i \parallel X \parallel r_{i+1})$$

and

$$C_{i+1}^{(1)} = F_k(g^{(1)} \parallel X \parallel r_{i+1}).$$

If the result is $C_i^{(0)} = C_{i+1}^{(1)}$, the attacker knows that $r_i = g^{(1)}$. The ciphertext $C^{(1)}$ can be

rewritten as follows:

$$C^{(1)} = F_k(r_0^{(1)} || \text{start} || r_1^{(1)}) F_k(r_1^{(1)} || p_1 || r_2) \dots F_k(r_i || p_i || r_i) F_k(r_i || X || r_{i+1}) \\ F_k(r_{i+1} || p_{i+1} || r_{i+2}) \dots C_{n+1} C^{*(1)}.$$

The attacker now can construct valid ciphertexts from $C^{(1)}$ by inserting two copies of $C_i^{(1)}$ into the position just behind it. The forgery ciphertext will be as follows:

$$C^{\text{forgery}} = F_k(r_0^{(1)} || \text{start} || r_1^{(1)}) F_k(r_1^{(1)} || p_1 || r_2) \dots F_k(r_i || p_i || r_i) F_k(r_i || p_i || r_i) F_k(r_i || p_i || r_i) \\ F_k(r_i || X || r_{i+1}) F_k(r_{i+1} || p_{i+1} || r_{i+2}) \dots C_{n+1} C^{*(1)}.$$

Obviously, in the new ciphertext C^{forgery} the chain of random padding values ($\{r_i\}$) is still consistent and the checksum value (r^*) will not be changed. It is a valid ciphertext although it is constructed by the attacker. Furthermore, the attacker can replicate $C_i^{(1)}$ any even times to generate valid cipher texts as many as possible without querying to the RPC oracle.

However, if $C_i^{(0)} \neq C_{i+1}^{(1)}$, the attacker repeatedly insert X into positions $i+2, i+3, \dots$ until some two blocks with equivalent value are found in the set $\{C_i^{(0)}, C_{i+1}^{(1)}, C_{i+2}^{(2)}, C_{i+3}^{(3)}, \dots\}$. Without loss of generality, let $C_{i+j}^{(0)} = C_{i+k}^{(1)}$ for some j and k ($k > j$), the attacker can generate forgery cipher texts as follows:

$$C_0^{(k)}, C_1^{(k)}, \dots, [C_{i+j}^{(k)}, \dots, C_{i+k-1}^{(k)}]^N, \dots, C_{n+k}^{(k)}, C^{*(k)}, N \geq 2, N \text{ is even.}$$

3.3 Efficiency of the Attack

From the birthday paradox [30, 31] we obtain a collision of the random padding values $\{r_i\}$ in the range from 0 to 2^r-1 after an expected number of $2^{r/2}$ queries. That is, after $2^{r/2}$ steps in the attack the probability to get two equivalent random padding values is larger than $1/2$. This means that the attacker can get a forgery ciphertext with probability larger than $1/2$ after $2^{r/2}$ queries to the RPC oracle. Compared to the conclusion in the original paper [20]. The probability to get a forgery ciphertext by equation (1) is $1/2^r$ after $2^{r/2}$ queries. Our attack method is obviously efficient.

For instance, in the case of 128-bit block RPC scheme, block size $b = 128$ and random padding size $r = 48$ as suggested in [20], the attacker can get forgery ciphertext with probability larger than $1/2$ after 2^{24} queries to the RPC oracle by applying our attack method. Even in the case of 256-bit block RPC scheme, $b = 256$ and $r = 64$ (another instance suggested in [20]), the number of queries needed to get a forgery ciphertext with probability larger than $1/2$ is 2^{32} . The computation is feasible by current computing power.

The theorem 6 in [20] gives the equation and proof of the upper bound of forgery attack complexity for the RPC scheme. However, as shown by the attack above, the theorem is not correct. Obviously, more complicated conditions have to be considered while trying to figure out the possible upper bound. We are not able to give the exact answer here.

3.4 Discussion

The current study develops two methods to improve the RPC scheme. First, the length of data can be combined into the tail block of ciphertext C^* . For instance, change C^* from $F_k(r^* \oplus r_0 \parallel 0^{b-2r} \parallel r^*)$ to be $F_k(r^* \oplus r_0 \parallel L \parallel r^*)$, where L is the length of plaintext. The attack does not for inconsistent data length in this way.

The second method applies mask value. Each data block is combined with a random mask value M_i before been encrypted. The ciphertext for a data block $(r_i \parallel p_i \parallel r_{i+1})$ is extended to be $F_k(M_i) \parallel F_k((r_i \parallel p_i \parallel r_{i+1}) \oplus M_i)$. Consider two encrypted data blocks $F_k(M_i) \parallel F_k((r_i \parallel X \parallel r_{i+1}) \oplus M_i)$ and $F_k(M_j) \parallel F_k((g^{(1)} \parallel X \parallel r_{i+1}) \oplus M_j)$. The attacker knows $r_i = g^{(1)}$ when $M_i = M_j$ and $r_i = g^{(1)}$. According to the birthday paradox, the probability is larger than 1/2 after $2^{(m+r)/2}$ queries, where m is the bit length of a random mask value. Usually, it is equal to the length of the underlying block cipher ($m = b$). Applying the improvement method to 128-bit block RPC scheme, ($b = 128$ and $r = 48$), the attack complexity will arise to 2^{88} queries for getting forgery ciphertext with probability larger than 1/2. The data expansion (ciphertext / plaintext) arises to be 8. (The original case has data expansion 4.) In the case of doubling the block size ($b = 256$) with the same manner of RPC scheme and extending random value size (r) to be 112 (data expansion = 8), the attack complexity becomes 2^{56} with successful probability 1/2. Obviously, our method has better improvement, although it may be inefficient for requiring additional block cipher encryptions of the mask values.

Chapter 4 Low Information Leakage Padding Scheme

The work proposes a new random padding scheme for symmetric key block encryption. In the padding scheme, a padding string is key-dependent and almost random. Thus, the padding string causes extreme low information leakage to the adversary with cipher text-only manner. Collecting plaintext-ciphertext pairs relating to the underlying secret key from padding strings becomes very difficult. We also show that with the padding scheme the simple CBC encryption mode becomes strong to defeat the padding oracle attacks.



4.1 Current Padding Schemes

Block cipher encryptions, such as AES [32] in encryption modes of simple Electronic Code Book (ECB) and Cipher Block Chaining (CBC) [33], require their input to be a multiple of the block size. Otherwise, a padding string will be appended to the plaintext to make it fitting the requirement. The padding string should be removed unambiguously at the time of decryption.

Several conventional padding schemes are used. Some use simple and constant padding string, called them Constant-Padding (CP) scheme for instance as follows.

- (1) CP1: Pad with zero characters.

- (2) CP2: Pad with zero characters and fill the last byte with the number of padding characters.
- (3) CP3: Pad with bytes of the same value as the number of padding bytes. The method was recommended in PKCS#5 [18, March 1999 #5] and RFC2630 [19].
- (4) CP4: Pad with 0x80 followed by zero bytes.

Constant-Padding schemes are easy to implement but leak vast amounts of information about padding plaintext. The leaked information supports the adversary with high advantage to collect pairs of plaintext and cipher text relating to the underlying secret key. Constant padding string is also favorable for the padding oracle attacks proposed in [22] as discussed above. The strategy to reduce the possible information leakage is by random padding. That is, random values are included in the padding string. There are two Random-Padding (RP) schemes as below.

- (1) RP1: Pad with randomly selected characters and fill with last byte with the number of padding bytes.
- (2) RP2: Pad with randomly selected characters X and Y by form XY^n where $X \neq Y$. [34]

This paper proposes a new random padding scheme which causes very low information leakage about the padding string.

4.2 Information Leakage of a Padding String

A block cipher with good pseudo random property makes it difficult for an adversary, who does not know the secret key, to guess the corresponding plaintext for a given ciphertext. Ideally, all possible plaintexts are distributed uniformly. That is, given n bits ciphertext, the success probability to guess the corresponding n bits plaintext would be 2^{-n} . However, when the plaintext distribution space is not uniform, the adversary may get higher advantage to guess the plaintext. In the extreme case that the plaintext is a known constant, the probability to guess the plaintext is obviously 1. Plaintext information is useful for analyzing the underlying block cipher. Even the block cipher is strong to defeat known-plaintext attack, it would be better to hide all information of plaintext for reducing the probability of attack that is unknown currently. Here, we define the information leakage of the plaintext corresponding to a given ciphertext. The definition can be used to evaluate the information leakage of a padding string. For simplicity, the plaintext and ciphertext mentioned below, and at the rest of the paper, are assumed to be computed by an ideal block cipher.

Definition 1: (Entropy [35])

Let X be a random variable which takes on a finite set of values, x_i , with $1 \leq i \leq n$, and has probability distribution $p_i = p(X=x_i)$. The entropy of X is:

$$H(X) = - \sum_{i=1}^n p_i \log_2 p_i .$$

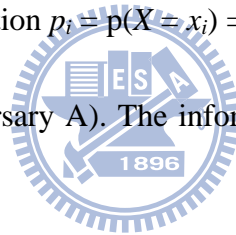
Definition 2: (Information leakage)

The random variable X , the same as in Definition 1, has information leakage, said $L(X)$, as follows. ($|X|$ stands for the average length of X)

$$L(X) = |X| - H(X) = \log_2 n + \sum_{i=1}^n p_i \log_2 p_i$$

Definition 3: (Information leakage of a plaintext)

Given an s -bit ciphertext C , the corresponding plaintext P is an element in the set $\{0, \dots, 2^s - 1\}$ as the secret key is chosen at random. Let X be a random variable on the set $\{x_i \mid 0 \leq x_i \leq 2^s - 1\}$ with the probability distribution $p_i = p(X = x_i)$ (the probability that x_i is the plaintext of C from the viewpoint of an adversary A). The information leakage of P to the adversary A , denoted as $L_A(P) = L(X)$.



The above definitions obtain the following results. To guess an s -bit plaintext P for which 2^s possible elements have the same probability, i.e. $1/2^s$, from the viewpoint of adversary, the information leakage of P is 0. However, if the plaintext is determined, the information leakage is s . So that it is reasonable to believe that the plaintext with low information leakage is more difficult to be guessed than the one with high information leakage, assuming that the two plaintexts have equal length.

When estimating the information leakage of a padding string, consider that the adversary knows padding string length. This is reasonable in the case that the adversary is able to

observe communication system information. For example, an encrypted message may be decrypted and then be saved to a file. Although the adversary is not authorized to open this file, he/she may be able to read its attributes. Thus the file size attribute may leak the length of the plain message to the adversary. Then padding string length can be derived.

Known padding schemes mentioned in section 1 have information leakage listed in Table 4-1, according to above information leakage definitions. For simplicity, we calculate that the block cipher has 64-bit data block, the padding is byte-oriented, and the encryption mode is simple ECB mode.

Table 4- 1: Information leakage of known padding schemes

$L_A(PS)$	CP1	CP2	CP3	CP4	RP1	RP2
(pad= 1 byte)	8	8	8	8	8	NA
(pad= 2 byte)	16	16	16	16	8	~0
(pad= 3 byte)	24	24	24	24	8	8
(pad= 4 byte)	32	32	32	32	8	16
(pad= 5 byte)	40	40	40	40	8	24
(pad= 6 byte)	48	48	48	48	8	32
(pad= 7 byte)	56	56	56	56	8	40
(pad= 8 byte)	64	64	64	64	8	48

4.3 The Proposed Random Padding Scheme

The proposed padding scheme uses a secret value as a mark word to construct a padding string. Let *MARK* be the mark word. A padding string *PS* is constructed as follows. (The notation \parallel denotes concatenation)

$$PS = MARK \parallel r_1 \parallel r_2 \parallel \dots, \text{ where } r_i \text{ is a random word and } r_i \neq MARK.$$

The mark word *MARK* is used as a distinguishable symbol for unambiguously removing the padding string from the plaintext. It can be an extended part of the secret key shared by the message sender and receiver. Let the length of a word be *w* bits. The padding string

$$PS = MARK \parallel r_1 \parallel r_2 \parallel \dots \parallel r_t$$


has information leakage, according Definition 3, as follows.

$$L_A(PS) = \log_2 2^{tw} + \sum_{i=1}^t \frac{1}{(2^w - 1)^t} \frac{1}{(2^w - 1)^t} \log \frac{1}{(2^w - 1)^t}$$

$$= t(w - \log_2(2^w - 1)).$$

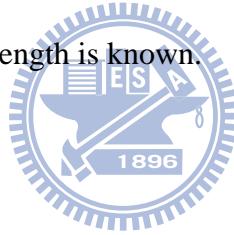
Since *MARK* is fixed, the property of *PS* is $\frac{1}{(2^w - 1)^t}$.

Consider the same arguments as described in Section 4.2. That is, the word size is a byte, and the block size is 64 bits. Information leakage of the proposed random padding scheme is shown in Table 4-2.

Table 4- 2: Information leakage of the proposed random padding scheme

$L_A(PS)$	Pad =1byte	Pad =2byte	Pad =3byte	Pad =4byte	Pad =5byte	Pad =6byte	Pad =7byte	Pad =8byte
	0	0.0056	0.0113	0.0169	0.0226	0.0282	0.0339	0.0452

To the best our knowledge, the proposed scheme has smallest information leakage as shown in Table 4-2 when padding string length is known.

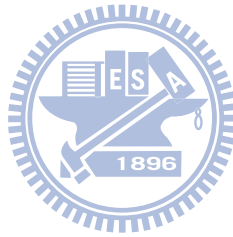


4.4 The security against the padding oracle attacks

Vaudenay [22] introduces the notion of padding oracle attacks on CBC mode encryption with CBC-PAD padding scheme. This attack assumes that a padding oracle which receives a ciphertext and then answer whether or not the corresponding plaintext is correctly padded. By querying to the oracle with appropriate modified initial vector and the ciphertext, an attacker is possible to invert the underlying block cipher. Several CBC mode encryptions in well-known products and standards are shown potentially vulnerable to this attack. [22, 36]

This simple CBC mode encryption with the proposed random padding scheme is strong

for defeating the padding oracle attacks under ciphertext-only manner. Assume that a padding oracle answers that the padding string is correct or not by checking whether the mark word exists or not. By applying a padding oracle attack, the attacker can control IV to confirm that $IV \oplus D(C) = MARK$ where $MARK$ is the mark word. But he/she can not know the extract value of $MARK$. That is, he/she can not obtain the $MARK$ information by ciphertext-only because the mark word can be key-dependent and is hidden from the eavesdropper. Thus, the plaintext of C will not be derived out.



Chapter 5 Flexible Authentication Scheme Based On Orthogonal Signature

E-business has recently become even more popular among enterprises. A verifier must obtain certain information in every stage to assure the authorization and integrity of each part of a signed e-document. However, the verifier generally knows nothing else except the very necessary information. For example, the merchant will keep banks away from acquiring what customers purchase; on the other hand, the merchants are not allowed to know customers' detailed finance status in banks. This research proposes an orthogonal signature that can work within multiple parties more than with two parties supported by the Dual Signature proposed by SET. This work also proves that the orthogonal signature satisfies those requirements proposed by SET and proposes a flexible authentication scheme based on the Orthogonal Signature to eliminate the burden of certificate validation for multiple users.

5.1 A Flexible Authentication Scheme Based On Dual Signature

5.1.1 The data flow of Dual Signature

Accessing the Internet has recently become very popular. Twenty-four hours online non-stop shopping without boundary limitation contributes to e-Commerce rapid growth. On the other hand, many e-Business tools like Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), Supply Chain Management (SCM), and etc., play an increasingly important role in today's business management.

Completing every e-Business procedure, especially for ERP and SCM, is necessary to obtain the signature from different participants of different operation departments to serve as official authorizations. Take the process of purchasing a PC online for example. The procedure normally begins with Internet order confirmation between customer and merchant. The merchant also operates through Internet to take over verification of stock control system, package and delivery with enterprise logistic network. The procedure ends with the customer signature, physical or electronic, to assure that he/she has received the merchandise. Using the verification/signature and different steps from the above procedure, it is possible to trace back exactly where the merchandise has been sent.

Nevertheless, we encounter serious threat today against the security issues upon transaction and privacy resulting from malicious attack of hackers or viruses. To avoid risk, the verifier/signer of each step will hope to sign on certain necessary part of a document without cognizing excess information. Similarly, when we need to check out some questionable part of a document, we would like to verify a limited part of it without reviewing

the whole document. For this purpose, the Secure Electronic Transaction (SET) protocol [13-15] developed by Visa and MasterCard proposed the dual signature scheme.

The concept of original dual signature is a method to secure the transaction which is paid with credit cards over public networks and whose information consists of order information and payment instructions. The assumptions of SET protocol are that the merchant and the acquirer can read order information and payment instructions respectively. Furthermore, both of them can verify the relationship between order information and payment instructions by adopting dual signature to link both pieces of information. The data flow is shown in figure 5-1.

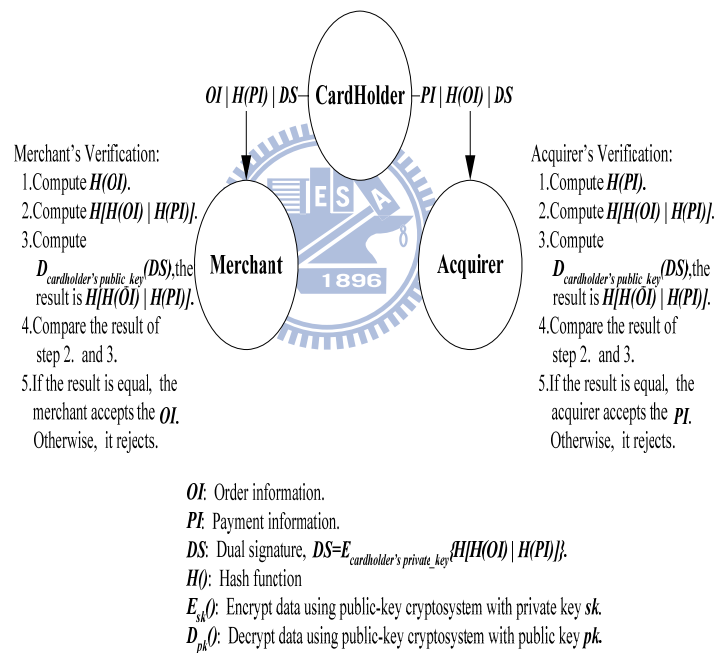


Figure 5- 1: The conceptual data flow of the dual signature scheme

A dual signature is generated by concatenating the two messages digests together, computing the message digest of the result and encrypting this digest with the signer's private key. Unfortunately, the dual signature can handle merely the situation of two parties. Therefore, this work extends the dual signature scheme by proposing an orthogonal signature

scheme. The orthogonal signature scheme can handle more than two individual parties of one document and verify the integrities and relationship between each party of the partial or whole document. In addition, if necessary, the information of this document concerning one unique party can be verified without revealing the other ones'. We use the modular multiplication to compute the message digest of the whole document instead of the concatenating operation in this scheme.

5.1.2 Flexible Authentication Scheme

In [16], the authors proposed a secure file server depicted as figure 2-2. They use dual signature to achieve the following advantages:

- Signature verification can be done by trusted third parties.
- Eliminate the burden of certificate validation.
- The trusted third party can verify the signature without plaintext.

Their process is described briefly as follows:

- Sender generates the dual signature with private key

$$DS = \text{Sign}(H(H(E(\text{file}))|H(\text{file}))), \text{ where}$$

$\text{Sign}()$: sign function

$H()$: hash function

$E()$: symmetric cipher function

- Trust third party (Secure File Server) checks DS using the public key of sender and

certificate status.

- If trust third party validates DS , then forward $Enc(\text{file})$ and $Hash(\text{file})$ to receiver.
- Receiver validate receiving the file by checking $Hash(\text{file})$.

We note that the secure file server can does real-time signature verification when a file is submitted and the receiver need not do the process of signature verification described in section 2.3.

5.2 The Orthogonal Signature scheme

5.2.1 Notations

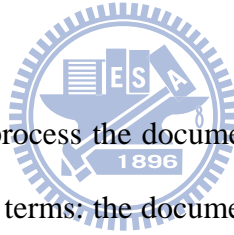


Table 5-1: Orthogonal Signature Notation

M	A document
N	The number of users
U_i	The i^{th} user who owns M_i ; where $M=M_1 M_2 \dots M_n$, and $ $ denotes concatenation
H	A hash function
(s_k, p_k)	The (private key, public key) pair for the

	trusted third party
E	Encryption algorithm
D	Decryption algorithm
S	Signing algorithm
V	Verifying algorithm
P	A large prime positive number

5.2.2 Conceptual model



The n users want to collectively process the document M and each user owns his/her part of the document. We will define two terms: the document message integrity code ($DMIC$) of the document and the exclusive message integrity code ($EMIC$) as following:

Definition 2.1:

$$1. DMIC (M) = H \left(\prod_{i=1}^n H(M_i) \bmod p \right), \text{ for all users.}$$

$$2. EMIC (M_i) = \prod_{j=1, j \neq i}^n H(M_j) \bmod p, \text{ for } i^{\text{th}} \text{ user.}$$

We use the modular multiplication and the one-way hash function (e.g. SHA-serials [37]) to compute a $DMIC$ of the document M and all the $EMIC$ s for each party of the document. The orthogonal signature of the document can be generated by applying any public-key algorithm (e.g. RSA [38], ElGamal [39] or elliptic curve cryptosystem [40]) on the $DMIC$ with a

signer's private key (often is the trusted third party). In our model, we also encrypt each *EMIC* with singer's private key. For each user, the orthogonal signature and the encrypted *EMIC* are the evidences for the whole document itself or for any individual part of document.

Concerning the verification, we describe the process of the i^{th} user as the example. He/She decrypts the orthogonal signature and the encrypted *EMIC* to get the $DMIC(M)$ and the $EMIC(M_i)$, and computes the message digest of his/her part of the document, $H(M_i)$.

According to the definition 2.1, we know the following:

$$DMIC(M)=H(H(M_i)\bullet EMIC(M_i) \text{ mod } p). \quad (1)$$

We use $DMIC(M)$, $EMIC(M_i)$ and $H(M_i)$ to check whether the Eq. (1) is held. If they satisfy the Eq. (1), the integrity of his/her part of the document is clear and the relationship between his/her part and the document is confirmed [41].

If any malicious attacker wants to compromise the document, it must find another document M' such that $DMIC(M')=DMIC(M)$. In fact, $DMIC$ is generated by hash function; it is computationally infeasible to find another document with the same $DMIC$. If any user maliciously or carelessly modifies the content of his/her part of the document, the message digest of the modified content must be equal to the original one. It is also computationally infeasible to do so.

5.2.3 The Proposed Scheme

Our scheme needs a trustworthy center to manage and distribute the document. Figure 5-2 shows what are the messages sent to each user, how this trusted center manipulate and distribute the document, and how each user verifies his/her own receiving messages. And,

we describe each step as follows.

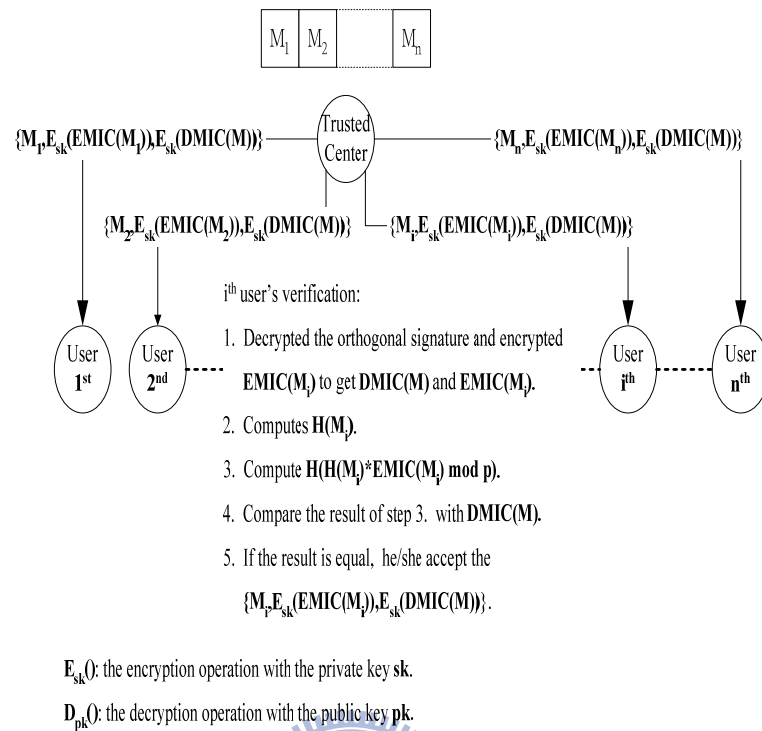


Figure 5- 2: The conceptual data flow of the orthogonal signature scheme

(1) Document structure:

We assume that the document consists of n parts to be distributed to n users. The trusted center knows which user should own which part. Let D be the document and $D=D_1|D_2| \dots |D_n$. The D_i is owned by i^{th} user where $i=1, \dots, n$. For the sake of easy reconstruction, the trusted center indexes all these parts of the document. M is the indexed document and let $M=(1|D_1)|(2|D_2)| \dots |(n|D_n)$; where $M_i=i|D_i$ and $M=M_1|M_2| \dots |M_n$.

(2) The generation of a $DMIC$ and all the $EMICs$ by trusted center:

Let $DMIC(M)$ is a $DMIC$ of the indexed document M . We use a one-way hash function H and modular multiplication to compute it, where

$$\Delta DMIC = \prod_{i=1}^n H(M_i) \text{ mod } p \quad (2)$$

and

$$DMIC(M) = H(\Delta DMIC). \quad (3)$$

In our scheme, we use the multiplication modulo p to compute the $DMIC$. The concatenation must operate in order and the complexity of the space is the $O(n)$. The size of the prime number p is larger than the size of the output size of one-way hash function.

Let the $EMIC(M_i)$ is an $EMIC$ of the i^{th} individual part of the document, where

$$EMIC(M_i) = \prod_{j=1, j \neq i}^n H(M_j) \text{ mod } p \text{ where } i=1, \dots, n. \quad (4)$$

According to the Eq. (2) and Eq. (4), we have

$$\Delta DMIC = H(M_i) \bullet EMIC(M_i) \text{ mod } p \quad (5)$$

and

$$EMIC(M_i) = \Delta DMIC \bullet H(M_i)^{-1} \text{ mod } p \quad (6)$$

The trusted center uses Eq. (2), Eq. (3) and Eq. (6) to compute all the $EMIC$ s and the $DMIC$. It uses a public-key algorithm to encrypt the $DMIC$ and all the $EMIC$ s with its private key s_k . The encrypted $DMIC$, $E_{s_k}(DMIC(M))$, is the orthogonal signature of the document M . For each user, the encrypted $EMIC$ s of his/her part of the document and the orthogonal signature are the necessary evidences that we mentioned in section 2.2.

The trusted center must send messages $\{M_i, E_{s_k}(EMIC(M_i)), E_{s_k}(DMIC(M))\}$ to n users respectively. We use the digital envelope scheme to transmit the message in secure.

(3) Verification of the orthogonal signature of the document M:

After receiving the message $\{M_i, E_{sk}(EMIC(M_i)), E_{sk}(DMIC(M))\}$, the i^{th} user will do the following steps to verify the signature:

- 1.) Use the trusted center's public key p_k to decrypt the encrypted $EMIC(M_i)$ and the orthogonal signature to get the $EMIC(M_i)$ and the $DMIC(M)$.

$$DMIC(M) = D_{pk}(E_{sk}(DMIC(M)))$$

and

$$EMIC(M_i) = D_{pk}(E_{sk}(EMIC(M_i))).$$

- 2.) Compute the message digest of his/her own part of the document, $H(M_i)$.
- 3.) Compute T_1 , $T_1 = H(H(M_i) * EMIC(M_i) \text{ mod } p)$.
- 4.) Compare T_1 with $DMIC(M)$.
- 5.) If T_1 equals to $DMIC(M)$, the orthogonal signature is valid. Otherwise, it is invalid.

If the orthogonal is valid, it means that the i^{th} user also verifies the integrity of his/her own part of the document and the evidences (orthogonal signature and the encrypted $EMIC$) from the trusted center. He/She also confirms the relationship between his/her own part of the document and the document. So the i^{th} user can be confident to continue further process with the sufficient evidences to protect himself/herself.

(4) Verification of the $DMIC$ and the $EMIC$:

In our scheme, the i^{th} user can further verify the $DMIC(M)$ and $EMIC(M_i)$ by requesting the $H(M_j)$ and the $EMIC(M_j)$ where $j=1, \dots, n$ and $j \neq i$ from all the other users.

(4.1) Verification of the $DMIC$:

- (1) Compute the message digest of his/her own part of the document, $H(M_i)$.

(2) Compute the $\Delta DMIC$,

$$\Delta DMIC = \prod_{i=1}^n H(M_i) \text{ mod } p \quad (7)$$

(3) Compute $H(\Delta DMIC)$.

(4) Decrypt the orthogonal signature to get $DMIC(M)$, $DMIC(M) = D_{pk}(E_{sk}(DMIC(M)))$.

(5) If $H(\Delta DMIC) = DMIC(M)$, he/she accepts; otherwise, he/she rejects.

(4.2) Verification of the $EMIC(M_i)$:

(1) Compute the $\Delta EMIC$,

$$\Delta EMIC (M_i) = \prod_{j=1, j \neq i}^n H(M_j) \text{ mod } p \quad (8)$$

(2) Decrypt the encrypted $EMIC$, $EMIC(M_i) = D_{pk}(E_{sk}(EMIC(M_i)))$.

(3) If $\Delta EMIC = EMIC(M_i)$, he/she accepts; otherwise, he/she rejects.



5.2.4 .Flexible authentication scheme based on orthogonal signature

A secure file server has the same structure depicted as figure 2-4. This work uses orthogonal signature scheme, instead of dual signature scheme using in [16], as the kernel of its authentication scheme. The secure file server process based on orthogonal signature scheme is described as follows:

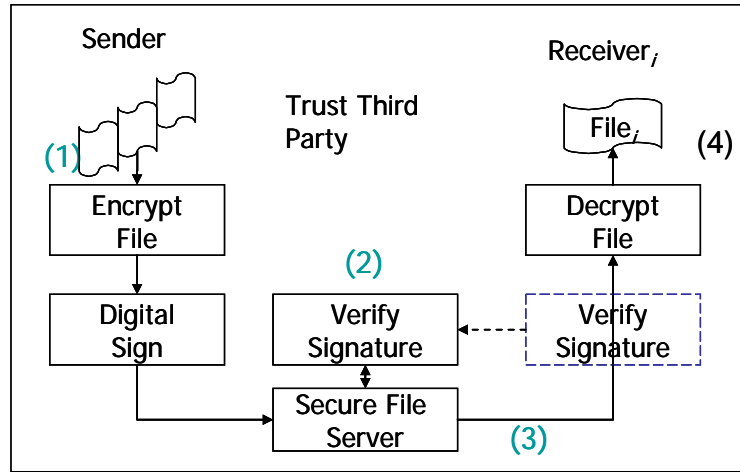
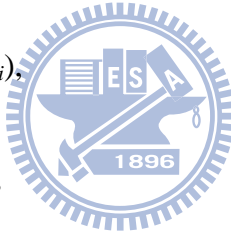


Figure 5- 3: Secure File Server Based on Orthogonal Signature

(1) A sender wants to send n files to n receivers. The sender generates

$$m_i = E_{sk_i}(\text{file}_i) | H(\text{file}_i),$$

$$M = m_1 | m_2 | \dots | m_n | m_s,$$



$$E_{sk}(\text{DMIC}(M)), \text{ and}$$

$$E_{sk}(\text{EMIC}(m_s)) \text{ and } E_{sk}(\text{EMIC}(m_i))$$

, where (sk, pk) is the public/private key pair of sender,

m_s is the document the sender send to secure file server,

(sk_i, pk_i) is the public/private key pair of receiver $_i, i = 1 \dots n,$

$E_{sk}()$ means that encrypts using key sk .

(2) Secure file server decrypts $E_{sk}(\text{EMIC}(m_i))$ and $E_{sk}(\text{DMIC}(M))$. Check if

$$DMIC(M) = H(H(m_s)*EMIC(m_s)\text{mod } p).$$

(3) If the test above holds, Secure file server forward m_i , $DMIC(M)$, and $EMIC(m_i)$ to receiver_{*i*}

(4) Receiver_{*i*} check if $DMIC(M) = H(H(m_i)*EMIC(m_i)\text{mod } p)$. if the test holds, receiver accepts.

The secure file server based on the orthogonal signature scheme will have advantages:

- The sender can send securely multiple files to multiple users.
- The receiver can verify the relationship.

These advantages will be discussed in next section.



5.3 Discussions

This work shows that the orthogonal signature can satisfy those requirements proposed by SET as follows.

(1) Integrity

The proposed scheme uses the $EMIC$ and the $DMIC$ to verify the integrity of the individual part of the document and the whole document itself. If the malicious attacker wants to compromise an individual part of the document M_i' ; the hash value of the multiplication modulo p of $EMIC(M_i')$ and $H(M_i')$ must be equal to the $DMIC$. But the other users can verify the $EMIC(M_i')$ by checking if the following equation holds

$$EMIC(M_i') = \prod_{j=1, j \neq i}^n H(M_j) \bmod p.$$

If the result is not equal, they realize the malicious attacker has modified an individual part of the document. Furthermore, in case that the malicious center compromises the *DMIC* or the *EMIC*, all users can verify the *DMIC* by Eq. (7) and the *EMIC* by Eq. (8).

(2) Authentication and Nonrepudiation:

This work using public-key algorithm encrypts *DMIC* and all the *EMICs*. As a result, the orthogonal signature provides of authenticity and nonrepudiation features.

(3) Confidentiality:

In this scheme, the i^{th} user can compute $T_1 = H(H(M_i) * EMIC(M_i) \bmod p)$ mentioned in Section 5.2.3; and compare T_1 and $DMIC(M)$ to verify the orthogonal signature. Therefore, the i^{th} user needs not to reveal any other individual part of document to verify the orthogonal signature.

(4) Relationship:

The relationship between the individual part of the document and the whole document by comparing $H(H(M_i) \bullet EMIC(M_i) \bmod p)$ with $DMIC(M)$ in this scheme. If the result is equal, the user can be sure of the correct relationship.

Chapter 6 Conclusions and Future Work

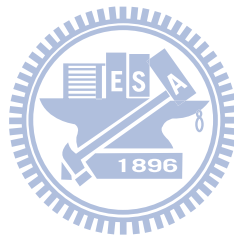
In Chapter 3, we show that the success probability of getting a forgery ciphertext is far larger than the conclusion in the original RPC paper [20]. Furthermore, we suggest two variants of the RPC scheme to defeat the proposed attack.

The proposed random padding scheme in Chapter 4 has very low information leakage. The appended padding string is almost random and hidden from the eavesdropper. The recent famous padding oracle attacks are extremely difficult to be applied on the simple CBC mode encryption with the proposed padding scheme ciphertext-only manner.

Chapter 5 discusses how a document needs many critical parts to process, while at the same time, each part needs the evidence to prove the integrity, belonging, authenticity and non-repudiation in an e-commerce environment. Processing the individual part of the document must additionally not reveal the contents of the whole document. The dual signature of SET proposal can only handle the document within two parties; therefore we extend the dual signature to be an orthogonal signature scheme which can protect each part from being betrayed. Consequently, this scheme can be more widely used in today's e-Business environment where the contract usually concerns more than two parties.

Current solutions for security components of e-Business applications do not meet all conditions. Hence, this dissertation proposes schemes to improve the barriers. The proposed schemes include: unforgeable incremental encryption, information leakage in padding last

block, and flexible authentication and authorization procedures. Mobile internet is a new development trend resulting in the fact that more and more e-Business applications are running on mobile devices. Mobile devices have the restricted processing capacity as well as the bandwidth. Thus our future work is to reduce the computing costs of cryptographic algorithms.



References

- [1] M. Bellare, O. Goldreich, and S. Goldwasser, "On the Security of Cipher Block Chaining," in *Advances in Cryptology - Crypto'94*, Y. Desmedt, Ed. Santa Barbara, California: LNCS 839, 1994, p. 216 ~ 233.
- [2] F. Hou, D. Gu, N. Xiao, and Y. Tang, "Secure Remote Storage through Authenticated Encryption," in *International Conference on Networking, Architecture, and Storage*, 2008.
- [3] S. Jones, M. Wilikens, P. Morris, and M. Masera, "Trust Requirements in e-Business," *Communications of the ACM*, vol. 43, pp. 81-87, 2000.
- [4] T. Tsiakis and G. Sthephanides, "The concept of security and trust in electronic payments," *Computer & Security*, pp. 10-15, 2005.
- [5] E. Rescorla, "A Survey of Authentication Mechanisms," *Internet Architecture board (IAB), working in progress, draft-iab-auth-mech-02.txt*, 2003.
- [6] B. Schneier, *Applied Cryptography*, Second ed.: John Wiley & Sons, 1996.
- [7] P. C. K. Hung and K. Karlapalem, "A Secure Workflow Model," in *Australasian Information Security Workshop (AISW2003)* Adelaide, Australia, 2003.
- [8] IEEE, "P1619, .1, .2, .3," in <http://ieee-p1619.wetpaint.com/>; IEEE Security in

Storage Working Group.

- [9] **J. Lazaro, A. Astarloa, U. Bidarte, Jimenez, and A. Zuloaga, "AES-Galois Counter Mode Encryption.Decryption FPGA Core for Industrial and Residential Gigbit Ethernet Communications," *ACR 2009*, vol. LNCS 5453, pp. 312-317, 2009.**
- [10] **RFC2104, "HMAC, Keyed-Hashing for Message Authentication," IETF, Ed., 1997.**
- [11] **M. Bellare and D. Micciancio, "A new paradigm for collision-free hashing: incrementality at reduced cost," in *Advances in Cryptology - EUROCRYPT 97*. vol. LNCS1233, W. in Fumy, Ed. Konstanz, Germany, 1997, pp. 163-192.**
- [12] **M. Bellare and C. Namprempre, "Authenticated Encryption: Relations Ammong Notions and Analysis of Generic Composition Paradigm," in *Asiacrypt 2000*, 2000.**
- [13] **SET1(SETCo), "SET Specification Version 1.0, Book 1: Business Description," 1997.**
- [14] **SET2(SETCo), "SET Specification Version 1.0 Book 2: programmer's Guide," 1997.**
- [15] **SET3(SETCo), "SET Specification Version 1.0, Book 3: Formal Protocol**

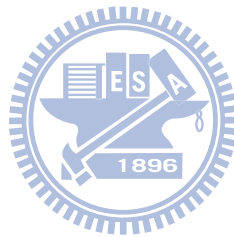
- Definition," 1997.
- [16] C. Center, "Secure File Transfer Project," 2003.
- [17] C. S. Jutla, "Encryption modes with almost free message integrity," *Advances in Cryptology-EUROCRYPT 2001*, vol. LNCS2045, pp. 529-544, 2001.
- [18] PKCS#5, " Password-Based Cryptography Standard ", PKCS ed: RSA Laboratories, 1999.
- [19] RFC2630, "Cryptographic Message Syntax," IETF, Ed., 1999.
- [20] E. Buonano, J. Katz, and M. Yung, "Incremental Unforgeable Encryption," in *FSE 2001*. vol. LNCS 2355, M. Matsui, Ed. Yokohama, Japan, 2002, p. 109 ~ 124.
- [21] H. Krawczyk, "How Secure Is SSL? ," *Advances in Cryptology - CRYPTO'01*, p. 310 ~ 331, 2001.
- [22] S. Vaudenay, "Security Flaws Induced by CBC Padding - Application to SSL, IPSEC, WTLS..." *Advances in Cryptology - Eurocrypt'02*, vol. LNCS 2332, pp. 534 - 545, 2002.
- [23] D. Denning, "Field Encryption and Authentication," *Advances in Cryptology - CRYPTO'83*, 1983.
- [24] D. Denning, "Cryptographic checksums for multilevel data security," in *Symp. on Security and Privacy*, 1984, pp. 52-61.

- [25] Z. Yang, S. Sesay, J. Chen, and D. Xu, "A Secure Database Encryption Scheme," *American Journal of Applied Sciences*, vol. 1, pp. 327-331, 2004.
- [26] RFC3447, "The TLS Protocol Version 1.0," 1999.
- [27] RFC4346, "The Transport Layer Security (TLS) Protocol Version 1.1," 2006.
- [28] RFC2284, "PPP Extensible Authentication Protocol (EAP)," IETF, Ed., 1998.
- [29] RFC2716, "PPP EAP TLS Authentication Protocol," IETF, Ed., 1999.
- [30] C. P. Pfleeger. and P. S. L, *Security in Computing*, 4 ed.: Prentice Hall, 2007.
- [31] D. R. Stinson, *Cryptography Theory and Praticce*: CRC Press, 1995.
- [32] FIPS197, "Advanced Encryption Standard (AES)," U. S. D. o. C. N. I. o. S. a. Technology, Ed., 2001.
- [33] FIPS81, "DES Modes of Operation," U. S. D. o. C. N. I. o. S. a. Technology, Ed., 1980.
- [34] J. Black and H. Urtubia, "Side-Channel Attacks on Symmetric Encryption Schemes: The Case for Authenticated Encryption," in *Proc. of 11th USENIX Security Symposium*, San Francisco 2002, pp. 327-338.
- [35] N. Smart, *Cryptography: An Introduction*: McGraw-Hill, 2003.
- [36] A. K. L. Yau, K. G. Paterson, and C. J. Mitchell, "Padding Oracle Attacks on

- CBC-Mode Encryption with Secret and Random IVs," *FSE 2005*, pp. 299-319, 2005.**
- [37] **FIPS180-2, "Secure Hash Signature Standard (SHS)," U. S. D. o. C. N. I. o. S. a. Technology, Ed., 2002.**
- [38] **R. L. Rivest, A. Shamir, and L. M. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Communications of the ACM*, vol. 21, pp. 120-126, 1978.**
- [39] **T. ElGamal, "A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," *Advances in Cryptology - CRYPTO 84*, pp. 10-18, 1985.**
- [40] **P1363, "Standard Specifications for Public-Key Cryptography, Version D13," IEEE, Ed., 1999.**
- [41] **W. Stallings, *Cryptography and Network Security: Principle and Practice*, 3 ed.: Prentice Hall, 2003.**

Vita

- Aug. 2009 Doctor of Philosophy, Computer Science,
National Chiao Tung University, Taiwan
- July 1996 Master of Electrical Engineering,
National Cheng Kung University, Taiwan
- July 1992 Bachelor of Mathematics,
National Central University, Taiwan



Publication List

■ Journal paper

- [1] C.-C. Wang, **Min-Chih Kao** and Ya-Shiung Yeh, “Forgery attack on the RPC incremental unforgeable encryption scheme”, IEE Proc.-Inf. Secur., Vol. 153, No. 4, December 2006, pp. 143-145. (SCIE, EI)
- [2] C.-C. Wang, **Min-Chih Kao** and Ya-Shiung Yeh, “Low Information Leakage Random Padding Scheme for Block Encryption”, Journal of Discrete Mathematical Sciences & Cryptography.
- [3] **Min-Chih Kao** and Chen-Yu Lee(交大學生), “Efficient Share Renewal Protocol Design for Mobile Ad Hoc Networks using Perfect Hash Families”, IET Information Security. (submitted) (SCIE, EI)

■ Conference paper

- [1] Ya-Shiung Yeh (指導教授), **Min-Chih Kao**, “New Threshold Proxy One-Time Signature Scheme”, International Computer Symposium 2006 (ICS 2006)
- [2] **Min-Chih Kao**, Ya-Shiung Yeh (指導教授) and C.-C. Wang(清雲助理教授) “Untraceable Identity Management Framework for Mobile Access”, Intelligent System Design and Applications Nov. 26-28, 2008 (ISDA 2008) (EI)

■ Pattern

- [1] **高銘智**, 陳彥學(工研院工程師), 崔文(工研院工程師), 楊文新(工研院工程師), 鄭仁傑(工研院組長), “Data access control system and method”, No. 6,748,084 (owned by ITRI) (美國國專利)
- [2] **Min-Chih Kao**, YA-WEN LEE(工研院工程師), Ya-Shiung Yeh (指導教授), and CHEN-HWA SONG(工研院經理), “無線網路安全認證方法與系統”, No. I305462 (owned by ITRI) (中華民國專利)
- [3] **Min-Chih Kao**, YA-WEN LEE(交大學生), Ya-Shiung Yeh (指導教授) “網路訊息認證方法與應用其方法之網路系統”, 申請號 096107464 (owned by ITRI) (中華民國專利) (審查中)