SCIS2000 The 2000 Symposium on Cryptography and Information Security
Okinawa, Japan, January 26-28, 2000

Another Improved SET : Slim SET

Moonseog Seo *
msseo@icu.ac.kr

Kwangjo Kim *
kkj@icu.ac.kr

Abstract— In order to improve the efficiency of the Secure Electronic Transaction (SET) protocol, Hanaoka *et al.* proposed a light weight version of the SET protocol, called "LITESET"at ACISP '98. They claimed that LITESET has about 50% reduction in computational time and about 80% reduction in communication overhead by using a signcryption scheme. We propose another way to construct an efficient SET protocol, called "Slim SET(SSET)". For acheiving the same security level and efficiency, SSET uses the domain-verifiable signcryption scheme that only predetermined n participants within the domain of protocol participants can decrypt their own part of message and verify the whole transaction. SSET is as efficient as LITESET, but doesn't require recipients' private keys for detection of dishonest acts.

Keywords: SET, signcryption, cryptographic protocol.

1 Introduction

There is no question that electronic commerce, as examplified by the popularity of the Internet, is going to have an enormous impact on the financial services industry. Payment systems and their financial institutions will play a significant role in the electronic commerce field.

The Secure Electronic Transaction (SET) protocol provided by MasterCard and VISA is one of a large number of proposals for electronic payments over open networks like the Internet. The SET specification[6, 7, 8] will enable greater payment card acceptance, with a level of security that will encourage consumers and businesses to make wide usage of payment card products in this emerging market.

The security requirements of SET transaction are confidentiality of information, integrity of data, cardholder account authentication, merchant authentication and interoperability. To meet these needs, the SET protocol use the well-known cryptographic algorithm such as DES, RSA signature algorithm and so on. Aso, these algorithms must be implemented almost simultaneously in transactions of the SET. However the implementation of SET is considered to impose heavy computation and overhead on a system that employs SET, primarily due to use of the RSA digital signature and encryption scheme.

Efficiency is one of important key factors that must be fulfilled in the SET transaction processing. The efficiency is achieved by applying signcryption scheme to SET protocol. The signcryption first proposed by Zheng[9], is a new cryptographic primitive so called *"catch two birds with single stone"* scheme. This simultaneously fulfills both the functions of signature and encryption in a single logical step, and reduces computational cost which is significantly lower than that required by the traditional signature-then-encryption paradigm [1, 4].

In [2], Hanaoka *et al.* proposed a light weight version of the SET protocol, called "LITESET"to improve SET's efficiency. They had been achieved reduction of the computational time and communication overhead by the use of the original signcryption scheme. But in LITESET, to detect dishonest acts, recipients' private keys are required. Recipients' key must be revoked and then reissued. More computational costs need to be invested to detect dishonest acts.

In this paper, we propose SET protocol based on Domain-verifiable signcryption proposed in [5], and then we call this scheme as SSET, meaning Slim SET. The Domain-Verifiable Signcryption Scheme (DVSS) is that only predetermined n participants within the domain of protocol participants can decrypt their own part of message and verify the whole transaction. By comparing SSET with LITESET, we claim that the compu-

* School of Eng., Information and Communications Univ., 58-4 Hwaam-dong, Yusong-gu, Taejon, 305-348, Korea

tational cost and message overhead of our scheme is almost the same as those of LITESET. Also we can detect dishonest acts without recipients' secret keys.

The rest of the paper is organized as follows. In Section 2, we describe the overview of SET and DVSS in brief and propose SSET. Section 3 provides the comparison of our scheme SSET with LITESET. In Section 4, we show the security considerations of SSET. Finally concluding remarks are given in Section 5.

2 SSET : SET based on DVSS

In this section, we will show how to improve SET in terms of efficiency: specifically, how to adapt Domain-verifiable signcryption for SET. The most important part of this work is how to link a message to another message like the Dual signature. We also construct messages in SSET. Before the detailed description of SSET, we first briefly review the SET and DVSS.

2.1 Overview of SET

SET would only come into play after the customer had decided what to buy, for how much, and that she wanted to pay with a payment card. An overview of the payment process is shown in Figure 1. The cardholder initiates a payment with the merchant using SET. The merchant then uses SET to have the payment authorized. The payment gateway acts as the front end processor to the existing financial network [3].

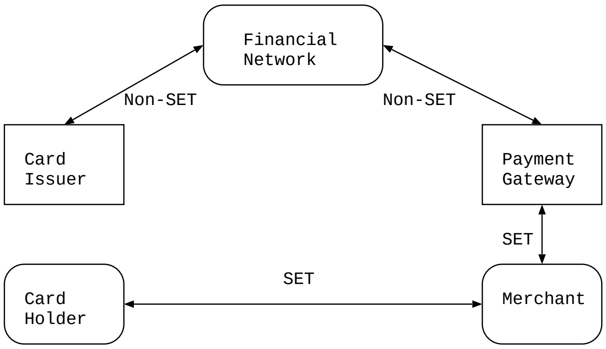

Figure 1: Phases of a credit card payment addressed by SET standards

The SET protocol messages consist of request/ response message pairs. The main transaction of SET is a purchase transaction. The messages needed to perform a complete purchase transaction usually include initialization request and response(PInitReq/PInitRes), purchase order request and response(PReq/PRes) and authorization request and response(AuthReq/AuthRes). The flow of these six messages is shown in Figure 2.

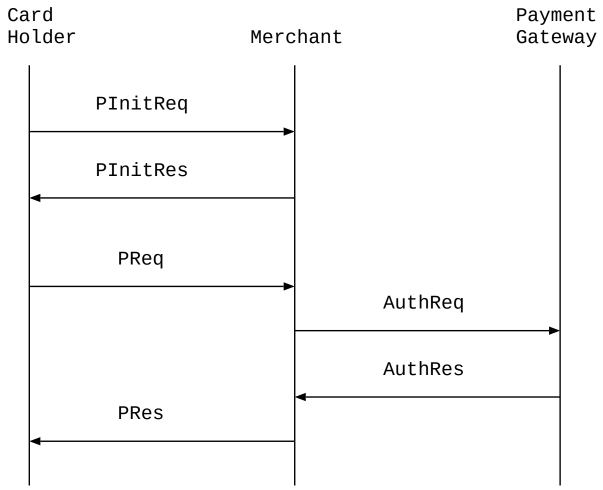

Figure 2: Flows of SET messages

In SET, the optimized dual signatures is applied to link order information and payment information. This provides a link between a message and an identity, to prevent other party from seeing the content of the message. As the name implies, it is used in applications where two related messages are being sent. In protocols involving three parties, such as a credit card transaction, the dual signature is employed [6].

2.2 DVSS

Within domain of protocol participants, each participant wants to be maintained his own message included in transaction secretly even against any other participants. All participants also require to authenticate the transaction that consists of participants secret partial information.

In [5], the Domain-verifiable Signcryption Scheme, in short, DVSS satisfies these requirements. Each participant can decrypt just his own message and all participants can verify the whole transaction. This scheme could be applied to SET protocol that need to be kept participant's partial information secret and for the total transaction message to be authenticated by merchants and payment gateways simultaneously.

Scheme for Domain Verification

We use the following notations throughout this paper.

p : a large prime

q : a large prime factor of $p-1$

g : an element of Z_p^* of order q
$hash$: a one-way hash function
KH : a keyed one-way hash function
(E, D) : the encryption and decryption algorithms of a symmetric key cipher.
Alice's key :
$x_a \in Z_q^*$: Alice's private key
$y_a = g^{x_a} \bmod p$: Alice's public key.
Recipient B_i's Keys Within domain of n participants ($i \in \{1, \ldots, n\}$)
$x_{b_i} \in Z_q^*$: B_i's private key
$y_{b_i} = g^{x_{b_i}} \bmod p$: B_i's public key.

Signcrypting : Alice randomly chooses $x \in Z_q^*$ then sets
$k_1 = hash({y_{b_1}}^x \bmod p), k_2 = hash({y_{b_2}}^x \bmod p), \ldots, k_n = hash({y_{b_n}}^x \bmod p)$
$k = hash(g^x \bmod p)$
$c_1 = E_{k_1}(m_1), c_2 = E_{k_2}(m_2), \ldots, c_n = E_{k_n}(m_n)$
$r_1 = KH_k(m_1||c_2||\cdots||c_n), r_2 = KH_k(c_1||m_2||\cdots||c_n), \ldots, r_n = KH_k(c_1||c_2||\cdots||m_n)$
$s = x/(r_1 r_2 \cdots r_n + x_a) \bmod q$.
Alice sends $(c_1, c_2, \ldots, c_n, r_1, r_2, \ldots, r_n, s)$ to B_n.

Unsigncrypting : Recipient B_i computes
$t = (y_a g^{r_1 r_2 \cdots r_n})^s \bmod p$
$t_i = t^{x_{b_i}} \bmod p$
$k = hash(t)$
$k_i = hash(t_i)$,
$m_i = D_{k_i}(c_i)$ to obtain B_i's own plaintext message, then checks whether
$KH_k(c_1||\cdots||m_i||\cdots||c_n) = r_i$ for signature verification.

Later when necessary, B_i may forward $(c_1, c_2, \ldots, c_n, r_1, r_2, \ldots, r_n, s)$ to any other participants, who want to decrypt his own message and can be convinced that it came originally from Alice by executing through this unsigncrypting.

2.3 Proposed Scheme

In SET protocol usage, there exist 3 participants for one transaction. A transaction consists of secret information to be processed by each participant. Each participant requires confidentiality for his own secret information. All participants also need authentication of that whole transaction.

The most important part of our work is how to link a message to another message. In LITESET, Hanaoka *et al.* suggested two kinds of linking operations : LinkedData and CoupledData. Also they modified CoupledData for the purpose of applying when a sender, S wants to encrypt partial message m_2. We propose **Link** operations corresponding to LITESET's each linking operation.

To sign the message m_1, encrypt it with the recipient R's public key and show the relationship between m_1 and m_2, we run Link-I operation.

Link-I$_{(\mathbf{S},\mathbf{R})}(\mathbf{m_1}, \mathbf{m_2})$ operation
Generation : A sender, S randomly chooses $x \in Z_q^*$ then sets
$k_1 = hash({y_b}^x \bmod p)$
$k = hash(g^x \bmod p)$
$c_1 = E_{k_1}(m_1), m_2$
$r_1 = KH_k(m_1||m_2)$
$s = x/(r_1 + x_a) \bmod q$.
S sends (c_1, m_2, r_1, s) to R.
Verification : R computes
$t = (y_a g^{r_1})^s \bmod p$
$t_1 = t^{x_b} \bmod p$
$k = hash(t)$
$k_1 = hash(t_1)$,
$m_1 = D_{k_1}(c_1)$ to obtain it's own plaintext message, then checks whether $KH_k(m_1||m_2) = r_1$ for signature verification.

To sign the messages, m_1 and m_2, encrypt only m_1, send m_1 and m_2 to R2, let R2 send m_1 to R1 with keeping m_1 unread, and show the relationship between m_1 and m_2, we run Link-II operation.

Link-II$_{(\mathbf{S},\mathbf{R1})}(\mathbf{m_1}, \mathbf{m_2})$ operation
Generation : S randomly chooses $x \in Z_q^*$ then sets
$k_1 = hash({y_{b_1}}^x \bmod p)$
$k = hash(g^x \bmod p)$
$c_1 = E_{k_1}(m_1), m_2$
$r_1 = KH_k(m_1)$
$r_2 = KH_k(c_1||m_2)$
$s = x/(r_1 r_2 + x_a) \bmod q$.
S sends (c_1, m_2, r_1, r_2, s) to R2
Verification of R1: R1 computes
$t = (y_a g^{r_1 r_2})^s \bmod p$
$t_1 = t^{x_{b_1}} \bmod p$
$k = hash(t)$
$k_1 = hash(t_1)$,
$m_1 = D_{k_1}(c_1)$ to obtain it's own plaintext message, then checks whether $KH_k(m_1) = r_1$ for signature verification.
Verification of R2: R2 computes
$t = (y_a g^{r_1 r_2})^s \bmod p$
$k = hash(t)$
and then checks whether
$KH_k(c_1||m_2) = r_2$ for signature verification.

If the verification succeed, R2 sends (c_1, r_1, r_2, s) to R1.

If S wants to encrypt m_2 for R2, S should send Link-III as follow :

Link-III$_{(\mathbf{S,R1,R2})}(\mathbf{m_1, m_2})$ operation
Generation : S randomly chooses $x \in Z_q^*$ then sets
$k_1 = hash({y_{b_1}}^x \bmod p), k_2 = hash({y_{b_2}}^x \bmod p)$
$k = hash(g^x \bmod p)$
$c_1 = E_{k_1}(m_1), c_2 = E_{k_2}(m_2)$
$r_1 = KH_k(m_1||c_2)$
$r_2 = KH_k(c_1||m_2)$
$s = x/(r_1 r_2 + x_a) \bmod q.$
S sends (c_1, c_2, r_1, r_2, s) to R1 and R2.
Verification of R1: R1 computes
$t = (y_a g^{r_1 r_2})^s \bmod p$
$t_1 = t^{x_{b_1}} \bmod p$
$k = hash(t)$
$k_1 = hash(t_1)$,
$m_1 = D_{k_1}(c_1)$ to obtain it's own plaintext message, then checks whether $KH_k(m_1||c_2) = r_1$ for signature verification.
Verification of R2: R2 computes
$t = (y_a g^{r_1 r_2})^s \bmod p$
$t_2 = t^{x_{b_2}} \bmod p$
$k = hash(t)$
$k_2 = hash(t_2)$,
$m_2 = D_{k_2}(c_2)$ to obtain it's own plaintext message, then checks whether $KH_k(c_1||m_2) = r_2$ for signature verification.

Messages in SSET
Whereas original SET protocol is based on dual signature scheme using hash and RSA signature algorithm, messages in SSET through above $Link\text{-}i(i \in \{I, II, III\})$ operations are shown in Table 1.

The abbreviated key words in Table 1 have the following meaning :

- RRPID : UniqueID for one pair of request and response.
- LID-C : LocalID of cardholder's transaction.
- Chall_C : Cardholder's challenge.
- BrandID : Brand of card.
- BIN : Cardholder's account number.

Table 1: Six main messages of SSET

message	message factor
PInitReq	{RRPID,LID-C,Chall_C,BrandID,BIN}
PInitRes	$\{Sig_M(PInitResData)\}$
PReq	$\{Link\text{-}II_{(C,P)}(PIData, OIData)\}$ If OIData is encrypted, $\{Link\text{-}III_{(C,M,P)}(PIData, OIData)\}$
AuthReq	$\{Link\text{-}I_{(M,P)}(AuthReqData, PIpartofPReq)\}$
AuthRes	$\{Link\text{-}I_{(P,M)}(AuthReqData, CapToken)\}$
PRes	$\{Sig_M(PResData)\}$

3 SSET v.s. LITESET

In the original SET protocol, dual signature is used for linking two messages whose recipients are different. Thus, although one recipient can only see the contents of the message m_1 he receives, he can be confident of the digest $H(m_2)$ of the other message m_2. Figure 3 shows how a dual signature is constructed.

In LITESET, they realized the function of dual signature by applying original signcryption scheme. They called this linked message as $CoupledData$. $CoupledData$ is constructed as follows:

$CoupledData_{S,R1}(m_1, m_2) = \{CSC_{S,R1,m_2}(m_1), CSig_{S,m_1}(m_2)\}$

- $CSig_{S,m_1}(m_2) = (m_2, r_1, s_1, H(m_1))$
  S randomly chooses $x_1 \in Z_q^*$ then sets
  $r_1 = hash(g^{x_1}, hash(m_1), hash(m_2))$
  $s_1 = x_1/(r_1 + x_a) \bmod q.$
  If verification succeed, R2 accepts m_2, and sends $CSC_{S,R1,m_2}(m_1)$ and $H(m_2)$ to R1.

- $CSC_{S,R1,m_2}(m_1) = (c_2, r_2, s_2)$
  S randomly chooses $x_2 \in Z_q^*$ then sets
  $(k_1, k_2) = hash({y_{R1}}^{x_2} \bmod p)$
  $r_2 = KH_{k_1}(hash(m_1), hash(m_2)[, etc])$
  $s_2 = x_2/(r_2 + x_a) \bmod q.$
  $c_2 = E_{k_2}(m_1)$
  R2 sends (c_2, r_2, s_2) to R1.
  If verification succeed, R1 accepts m_1.

In our implementation, **Link-II** operation works the equivalent function.

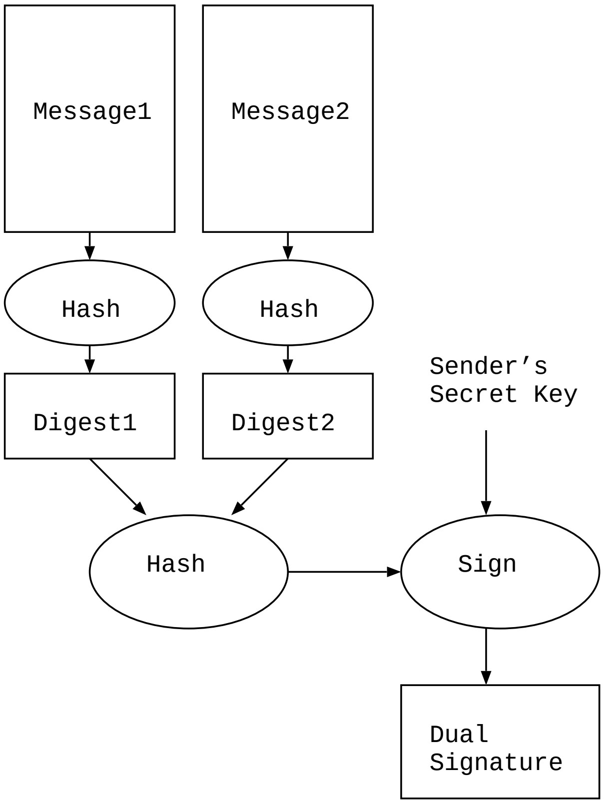

Figure 3: Construction of a dual signature on a pair of message

Table 2: Computational costs for message generation/verification

message	SET	LITESET	SSET
PInitReq	-/-	-/-	-/-
PInitRes	384/384	240/280	240/280
PReq	768/384	480/280	480/280
AuthReq	768/1536	240/560	480/1040
AuthRes	1536/768	480/280	480/520
PRes	384/384	240/280	240/280
Total	3072/3456	1440/1600	1920/2400

Table 3: Message overhead

message	SET	LITESET	SSET
PInitReq	-	-	-
PInitRes	1024bit	320bit	320bit
PReq	2008bit	720bit	400bit
AuthReq	4056bit	640bit	640bit
AuthRes	4256bit	480bit	480bit
PRes	1024bit	320bit	320bit
Total	12368bit	2480bit	2160bit

3.1 Computational costs

We estimate the number of modulo multiplications by using "square-and-multiply"and "simultaneous multiple exponentiation"used in [2]. They assumed that $|q| = 160$bit and $|p| = 1024$bit to achieve the same security level. We see that the computational costs of our scheme are little higher than that of LITESET, but saved about 35% compared with the conventional scheme. Computational costs for each scheme are shown in Table 2.

3.2 Message overhead

For the evaluation of message overhead, like LITESET, we assume that the message overhead is digital signature and encrypted session key. We see that the message overhead of main messages is saved over 82% on an average from Table 3.

3.3 Detection of dishonest acts

In LITESET, since they used the original signcryption scheme, recipient's private keys are required for detection of dishonest acts, and recipient can not be confident of the signature which is received by the other recipient. So in this scheme, more computational costs are required to detect dishonest acts and to maitain user's keys. But in SSET, we don't require the recipient's private keys, and we can be confident of the signature because SSET uses the DVSS that don't require recipient's private keys to generate signature verification keys.

4 Security Considerations

In DVSS, it can be done only within a domain of protocol participants to unsigncrypt message, because a participant B_i having his own secret key x_{b_i} can obtain partial information m_i within a domain and the only person who gets m_i can try to check if $KH_k(c_1||\cdots||m_i||\cdots||c_n) = r_i$ for signature verification. Any other persons that have not secret x_{b_i} will not be able to take part in unsigncryption.

In [5], they provided the security analysis of DVSS. Namely, if a signer, S chooses the integer x uniformly and randomly, then two values are (pseudo) independent as both g and $y_b = g^{x_b} \bmod p$ are generators in Z_p^* of order q which is a prime. This ensures that the signature verification and partial recovery of bits does not leak information that can be used in an attack on breaking message confidentiality or signature forgery.

According to this security analysis, if S chooses the integer x uniformly and randomly, then two values in **Link** operations of SSET such as ${y_{b_1}}^x \bmod p$ and $g^x \bmod p$ are (pseudo) independent. This guarantees that **Link** operations of SSET have message confidentiality and signature unforgeability.

In **Link-II** operation, if an attacker modify message, (c_1, r_1, r_2, s) which R2 sends to R1, R1 cannot accept his own message m_1. In this case, R1 has to send the false result of the verification to R2, because R2 already has accepted his own message m_2. According to the result from R1, R2 must revoke the previously received message m_2 for the completion of transaction.

5 Concluding Remarks

In this paper, we proposed SET protocol based on DVSS that only predetermined n participants can decrypt and verify within a domain. Our scheme has about 35% reduction in computational time and about 80% reduction in communication overhead by using DVSS.

In the side of efficiency, our scheme SSET is little higher than LITESET. But SSET doesn't require recipient's private keys for detection of dishonest acts. So user's keys can be easily and efficiently maintained. For the more efficiency and secrecy, the detailed designs of existing SET specification using the DVSS can be considered carefully.

References

[1] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms,"IEEE Transactions on Information Theory, IT-31(4):469-472, 1985.

[2] G. Hanaoka, Y. Zheng and H. Imai, "LITESET : a Light-Weight Secure Electronic Transaction Protocol,"Information Security and Privacy-ACISP'98, LNCS Vol. 1438, Springer-Verlag, pp. 215-226, 1998.

[3] D. O'Mahony, M. Peirce and H. Tewari, Electronic Payment Systems, May 1997

[4] C.P. Schnorr, "Efficient identification and signature for smart cards,"Advances in Cryptology- CRYPTO '89, LNCS 435, Springer-Verlag, pp. 239-251,1989.

[5] M. Seo and K. Kim, "Electronic Funds Transfer Protocol based on Domain-verifiable Signcryption Scheme,"Pre-Proc. of ICISC'99, Springer-Verlag, pp. 263-271,1999.

[6] Visa International and MasterCard International, Secure Electronic Transaction(SET) Specification book 1:Business Description, May 1997.

[7] Visa International and MasterCard International, Secure Electronic Transaction(SET) Specification book 2:Programmer's Guide, May 1997.

[8] Visa International and MasterCard International, Secure Electronic Transaction(SET) Specification book 3:Fomal Protocol Definition, May 1997.

[9] Y. Zheng, "Digital signcryption or how to achieve cost(signature and encryption) $<<$ cost(signature)+cost(encryption),"Advances Cryptology-CRYPTO'97, LNCS 1294, Springer- Verlag, pp. 165-179, 1997.