A Secure LITESET Scheme *

Jau-Ji Shen[†] Iuon-Chung Lin[‡] Min-Shiang Hwang$_{Member}$[†]

Graduate Institute of Networking and Communication Engineering[†]
Chaoyang University of Technology
Wufeng, Taiwan 413, R.O.C.
Email: mshwang@cyut.edu.tw

Department of Computer Science and Information Engineering[‡]
National Chung Cheng University
Chaiyi, Taiwan, R. O. C.

April 11, 2003

*This research was partially supported by the National Science Council, Taiwan, R.O.C., under contract no.: NSC91-2213-E-324-003.

A Secure LITESET Scheme

Abstract

Recently, a new light-weight version of the secure electronic transaction protocol was proposed. The protocol can achieve two goals. One goal is that the security level is the same as the SET protocol. The other goal is to reduce the computational time in message generation and verification, and reduce the communication overhead. However, the protocol has a weakness, which is that non-repudiation is acquired, but confidentiality is lost. In this paper, we point out the weakness of the protocol. We also propose an improvement to the protocol to overcome this weakness.

Keyworks: Cryptography, LITESET, SET, signcryption

1 Introduction

During the past few years, the use of computer networks has grown spectacularly. More and more transactions are completed increasingly through computer networks. In 1996, the MasterCard and VISA Corporations united to develop a unified Secure Electronic Transaction (SET) protocol [8, 9]. The SET protocol is a credit card-based payment system that allows making payments across computer networks. The overview of the payment process is that the cardholder can make secure electronic payments to the merchant over computer networks and the merchant can authorize a payment using the SET protocol.

To produce a secure electronic payment system, the following requirements must hold: authentication, confidentiality, integrity, and non-repudiation [6, 7].

The SET protocol satisfies these requirements using RSA [3, 11] and DES cryptosystems [12]. In the SET payment process, dual signatures are an important feature, which provide information segregation and protect the privacy of the cardholder. However, the RSA scheme within SET requires relatively large computational resources and produces a large message overhead.

Recently, a light-weight secure electronic transaction protocol, called "LITESET", was proposed by Hanaoka, Zheng, and Imai [4, 5]. The LITESET protocol is an improvement over SET, which solves the inefficiencies of SET. It uses a new cryptographic technology called signcryption [1, 13, 14]. The protocol can achieve two goals. One goal is that the security level is the same as the SET protocol. The other goal is to reduce the computational time in message generation and verification, and reduce the communication overhead.

At the same security level, LITESET can reduce the computational costs by 53.1%, message generation and verification by 53.7% and reduce message overhead by 79.9% [4, 5]. LITESET uses the same payment process as the SET protocol and it is more efficient. However, in Petersen and Michels's scheme [10], they point out a weakness in the signcryption scheme. The scheme acquires non-repudiation, but the confidentiality is lost. In this paper, we show the same weakness in the LITESET protocol based on the signcryption scheme, and propose an improved scheme to solve the confidentiality problem.

The rest of this paper is organized as follows. Section 2 reviews the LITESET protocol. Section 3 provides a cryptanalysis of LITESET. We improve the LITESET protocol and describe the security analysis in Sections 4 and 5. Section 6 concludes this paper.

2 Overview of the LITESET

The LITESET protocol [4, 5] consists of three mail entities: cardholder, merchant, and payment gateway. The cardholder makes payments with a credit card to the merchant. The payment gateway helps the merchant verify the payment. The LITESET protocol is based on the signcryption cryptosystem [1, 13, 14], which is different from SET, especially in the six frequent processes. In this paper, we only describe these processes. The transmittable messages in these processes are specified as follows.

- PInitReq: purchase initialization request.
- PInitRes: purchase initialization response.
- PReq: purchase request.
- PRes: purchase response.
- AuthReq: authorization request.
- AuthRes: authorization response.

In this protocol, the LITESET parameters include the following: Pv_c/Pb_c, Pv_m/Pb_m, Pv_p/Pb_p where $Pb = g^{Pv} \bmod p$ are the private/public keys of the cardholder, merchant, and payment gateway, respectively. $H(\cdot)$ is a one-way hash function. $E_k(\cdot)/D_k(\cdot)$ is an encryption/decryption function using a key k. p is a large prime; q is a large prime factor of $p-1$; g is an integer in $[1, \cdots, p-1]$ with the order q modulo p. The $PReq$ message includes PID, OID, and the dual signature. PID is a cardholder's payment instruction data and OID is a cardholder's order information data.

When cardholder sends a $PReq$ to the merchant, the cardholder must:

1. sign the messages, PID and OID, called dual signature.

2. encrypt the PID, and send $\{r_1, s_1, OID, H(PID)\}$, $\{r_2, s_2, c_2\}$ to the merchant.

Here we choose x_1 and x_2 which are two secret random numbers, and then compute:

$$\begin{aligned}
r_1 &= H(g^{x_1}, H(PID), H(OID)),\\
s_1 &= \frac{x_1}{r_1 + Pv_c} \bmod q,\\
e &= Pb_p^{x_2} \bmod p,\\
(k_1, k_2) &= H(e),\\
r_2 &= H(k_1, H(PID), H(OID)),\\
s_2 &= \frac{x_2}{r_2 + Pv_c} \bmod q,\\
c_2 &= E_{k_2}(PID).
\end{aligned}$$

When the merchant receives these messages, merchant verifies it as follows:

1. The merchant recovers g^{x_1} from Pb_c, g, r_1, s_1, and p.

$$g^{x_1} = (Pb_c \times g^{r_1})^{s_1} \bmod p.$$

2. $H(g^{x_1}, H(OID), H(PID))$ by using g^{x_1}, OID and $H(PID)$ is then computed. If the product equals r_1, the merchant accepts OID, and forwards $\{r_2, s_2, c_2\}$ and $H(OID)$ to the payment gateway.

The payment gateway verifies $\{r_2, s_2, c_2\}$ as follows:

1. The payment gateway recovers e from Pb_c, g, r_2, s_2, Pv_p, and computes (k_1, k_2), where $e = (Pb_c \times g^{r_2})^{s_2 \times Pv_p}$ $mod\ p$, and $(k_1, k_2) = H(e)$.

2. PID is decrypted, $PID = D_{k_2}(c_2)$.

3. The payment gateway checks $r_2 = H(k_1, H(PID), H(OID))$. If the product is equal, the payment gateway accepts the PID, and sends a message $AuthRes$ to the merchant.

The payment gateway proves the correctness of r_2, s_2, c_2, PID, and OID. In this protocol, the OID is known only to the merchant, while the PID is known only to the payment gateway.

Furthermore, if a dispute arises, such as the cardholder repudiates the transaction, the mechanism would need judge's cooperation to judge the validity of the transaction [14]. Therefore, in a dispute transaction, for the LITESET protocol to acquire non-repudiation, the judge must confirm that the signature, PID and OID were issued by the cardholder. A simplest solution is that the payment gateway directly presents Pv_p to the judge. However, the judge may be corrupt or less trusted. If the judge holds Pv_p, the judge can do everything on behalf of the payment gateway. This is very dangerous. Thus, the payment gateway cannot directly present Pv_p to the judge. A practical solution is that the payment gateway reveals $e = Pb_p^{x_2} \bmod p$ to the judge [14]. After that the judge can prove

$$\log_{(Pb_c \times g^{r_2})^{s_2}}(e) = \log_g Pb_p.$$

using the 4-move zero-knowledge proof protocol in [2].
With the parameter e, the judge can compute $(k_1, k_2) = H(e)$, and checks whether $r_2 = H(k_1, H(PID), H(OID))$. If the product is equal, the judge is convinced that the PID and OID are correct. Thus, the judge can correctly determine the origin of the transaction message.

3 Weakness of the LITESET Protocol

The LITESET provides a non-repudiation protocol to prevent the cardholder from being denied a transaction. In the case of a dispute, the protocol requires a judge to determine the validity of the transaction. The payment gateway reveals e to a judge, then, the judge determines if $\log_{(Pb_c \times g^{r_2})^{s_2}}(e) = \log_g Pb_p$ using a zero-knowledge interactive protocol [2]. If it is equal, the judge computes k_1, k_2 and checks $r_2 = H(k_1, H(PID), H(OID))$ and $c_2 = E_{k_2}(PID)$. However, after deciding a disputed case, the judge will know e, r_2, s_2, and Pb_p. The judge could easily compute K_{DH} as follows.

$$K_{DH} \equiv e^{s_2^{-1}} \times Pb_p^{-r_2} \equiv g^{Pv_c Pv_p} \bmod p. \tag{1}$$

Afterward, if this cardholder makes payment with the same credit card for any transaction, the judge have enough information to compute every transaction's parameter e'. Where $e' = K_{DH}^{s_2'} \times Pb_p^{r_2' s_2'} \bmod p$. According to parameter e', the judge can easily determine k_1', k_2' from $(k_1', k_2') = H(e')$.

Therefore, the judge can decrypt PID from $PID = D_{k_2}(c_2)$ for any transaction. The cardholder's privacy is lost and the protocol's security is compromised. The judge can easily determine PID and make forged transactions using PID. The LITESET protocol can gain non-repudiation, but confidentiality is lost.

4 An Improvement of LITESET

In this section, we improve the LITESET protocol to solve the confidentiality problem in the LITESET protocol. The judge receives e, and can then compute K_{DH} using some public information. The weakness of the LITESET protocol is that it cannot insure confidentiality. In our improvement protocol, we added a modular exponentiation operation that sends a new parameter to the judge to solve this weakness. Our solution as follows:

We define G to be a finite group of the order p. α is a generator of G and computes $K = \alpha^e \bmod p$. Now, we compute K_1, K_2 using $H(K) = (K_1, K_2)$ to replace $H(e)$. The payment gateway recovers e using the same method as in the LITESET protocol, $e = (Pb_c \times g^{r_2})^{s_2 \times Pv_p}$, and computes $(K_1, K_2) = H(\alpha^e \, mod \, p) = H(K)$. In a disputed transaction, the payment gateway reveals parameter K to the judge as a replacement parameter, and the judge proves $\log_{(Pb_c \times g^{r_2})^{s_2}}(\log_\alpha K) = \log_g Pb_p$ using zero-knowledge proof protocol [2] to verify K.

Next the judge computes (k_1, k_2), as in the conventional LITESET protocol, $(K_1, K_2) = H(\alpha^e) = H(K)$. The judge then checks r_2, c_2 and verifies the validity of this transaction. In the LITESET improvement, the judge cannot compute parameter K_{DH} or any other information. This improvement gains non-repudiation and protects confidentiality.

5 Discussions

5.1 Security Analysis

In our LITESET improvement protocol, we use parameter K to replace parameter e. In the non-repudiation protocol, the payment gateway reveals K, rather than e, to the judge. Similar to the LITESET, our improved scheme is also based on the signcryption cryptosystm [13, 14]. Thus, the security level is the same as the conventional LITESET protocol. Signcryption cryptosystem provides message confidentiality and unforgeability with lower computational and communication overhead. The formal proofs for the confidentiality and unforgeability of signcryption can refer to [1].

However, in Section 3, we have shown that the non-repudiation of LITESET is acquired, but the confidentiality is lost. In order to overcome this weakness, the LITESET improvement reveals K to the judge, where $K = \alpha^e \bmod p$. The judge cannot recover e from K, as it is based on the discrete logarithm problem. Any other user in the system has no means to derive K_{DH} by using Equation 1 without the knowledge of parameter e. Thus, it is impossible for a judge attempting to derive the parameter e' of following transactions, where $e' = K_{DH}^{s'_2} \times Pb_p^{r'_2 s'_2} \bmod p$, because the judge will find it hard to get K_{DH} from K.

In addition, the judge uses Zero-Knowledge to prove the correctness of parameter K and determine the validity of the transaction. Thus, the cardholder cannot repeat this transaction. If someone attempts to forge parameter K, the judge can perform zero-knowledge proof protocol [2] to see if the discrete logarithms of $\log_\alpha K$ to base $(Pb_c \times g^{r_2})^{s_2}$ and Pb_p to g are equal. Thus, the improved LITESET not only gains non-repudiation, but also ensures confidentiality.

5.2 Performance Analysis

LITESET can fulfill the all functions of SET and indeed can improve the efficiency of SET by using the *signcryption* cryptosystem [1, 13, 14]. With the help of signcryption, LITESET provides a 56.2% reduction in computational time in message generation, a 51.4% reduction in computational time in message verification, and a 79.9% reduction in communication overhead. The details of performance analysis and comparisons can refer to the [4] and [5]. In the improved LITESET, the processes are almost the same as the conventional LITESET protocol. We only add a modular exponentiation operation to compute the parameter K, $K = \alpha^e \bmod p$, in message generation and verification. However, the improved LITESET can gain non-repudiation without loosing confidentiality. Although the improved LITESET requires more computational overhead than the conventional LITESET protocol, the improved LITESET still more efficient than SET protocol. Furthermore, the improved LITESET also provide a 79.9% reduction in communication overhead.

6 Conclusion

The LITESET protocol reduces computational time and communication overhead using a new signcryption cryptosystem. This protocol cannot ensure confidentiality. In the solution to this problem, we added a modular exponentiation operation to the message generator. This operation increases operational costs minimally. This improvement is more efficient than SET, but less than the conventional LITESET protocol.

Acknowledgements

The authors wish to thank many anonymous referees for their suggestions to improve this paper.

References

[1] J. Baek, R. Steinfeld, and Y. Zheng, "Formal Proofs for Security of Signcryption," *Proceedings of Public Key Cryptography (PKC 2002)*, Lecture Notes in Computer Science, vol.2274, pp. 80–98, Springer-Verlag, 2002.

[2] D. Chaum, "Zero-knowledge undeniable signatures," *Advances in Cryptology, Eurocrypt'90*, Lecture Notes in Computer Science, vol.473, pp. 458–464, Springer-Verlag, 1991.

[3] C. C. Chang and M. S. Hwang, "Parallel computation of the generating keys for RSA cryptosystems," *IEE Electronics Letters*, vol.32, no.15, pp.1365–1366, 1996.

[4] G. Hanaoka, Y. Zheng, and H. Imai, "LITESET: A light-weight secure electronic transaction protocol," *Proceedings of Third Australasian Con-*

ference on Information Security and Privacy, ACISP'98, Lecture Notes in Computer Science, vol.1438, pp. 215–226, Springer-Verlag, 1998.

[5] G. Hanaoka, Y. Zheng, and H. Imai, "Improving the Secure Electronic Transaction Protocol by Using Signcryption," *IEICE Transactions on Fundamentals*, vol.E84-A, no.8, pp.2042–2051, August 2001.

[6] M. S. Hwang, I. C. Lin, and L. H. Li, "A simple micro-payment scheme," *Journal of Systems and Software*, vol.55, no.3, pp.221–229, 2001.

[7] M. S. Hwang, E. J. L. Lu, and I. C. Lin, "Adding timestamps to the secure electronic auction protocol," *Data & Knowledge Engineering*, vol.40, no.2, pp.155–162, 2002.

[8] MasterCard and VISA Corporations, *Secure Electronic Transaction (SET) Specification Book 1: Business Description*, MasterCard and VISA Corporations, June 1996.

[9] MasterCard and VISA Corporations, *Secure Electronic Transaction (SET) Specification Book 2: Programmer's Guide*, MasterCard and VISA Corporations, June 1996.

[10] H. Petersen and M. Michels, "Cryptanalysis and improvement of signcryption schemes," *IEE Proc.-Comput. Digit Tech.*, vol.145, no.2, pp.149–151, 1998.

[11] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public key cryptosystems," *Communications of the ACM*, vol.21, no.2, pp.120–126, Feb. 1978.

[12] B. Schneier, *Applied Cryptography*, 2nd Edition, John Wiley & Sons, New York, 1996.

[13] Y. Zheng, "Digital signcryption or how to achieve cost(signature & encryption) << cost(signature) + cost(encryption)," *Advances in Cryptology, Crypto'97*, Lecture Notes in Computer Science, vol.1294, pp.165–179, Springer-Verlag, 1997.

[14] Y. Zheng, "Signcryption or how to achieve cost(signature & encryption) << cost(signature) + cost(encryption)," *http://www.pscit.monash.edu.au/yuliang/pubs/signcrypt.ps.Z*, 1999.