A Server-aided Signature Scheme Based on Secret Sharing for Mobile Commerce

Chin-Ling Chen[1,*] Ling-Chun Liu[2] Gwoboa Horng[2]

[1] Department of Computer Science and Information Engineering
Chaoyang University of Technology
Taichung 413, Taiwan, ROC
clc@mail.cyut.edu.tw

[2] Department of Computer Science
National Chung Hsing University
Taichung City 402, Taiwan, ROC
0287@sun.epa.gov.tw, gbhorng@cs.nchu.edu.tw

Received 15 October 2007; Revised 1 December2007; Accepted 8 January 2008

Abstract. With the progress of the mobile communication technology and the popularity of the handheld devices, mobile commerce is of great importance today. We can use these devices to conduct business, such as to purchase books, and stocks, and digital goods (videos, audios, codes), and to play games, receive email, and even access various network resources. When the requested services need to be verified, the authentication of users and the non-repudiation of transactions become very important. Completing these tasks in wireless environments is a challenge for mobile devices that have limited computational capabilities. In this paper, we propose a server-aided signature scheme based on secret sharing for mobile commerce. Through one-time password authentication and secret sharing technology, we generate the cooperative signature of the server and the handheld device to satisfy the issues of security, non-repudiation, simplicity, validity, and mobility.

Keywords: Hashing chain, digital signature, secret sharing, server-aided signature, mobile commerce

1 Introduction

With the progress of mobile communication technology, mobile devices have become one of the most popular application tools. Due to convenience and ubiquity, mobile devices are becoming more and more useful tools used to purchase books, stocks, and digital goods (videos, audios, codes), and to play games, receive email, and even conduct business. Such applications include mobile payment systems, remote walk-through systems, electronic wallets, e-ticket systems, image authenticating and exchanging etc. [1]. However, there is no denying that the limited computational capabilities and limited power of mobile devices (almost all of them operate on batteries) make them ill-suited for complex cryptographic computations, such as large number calculations that are required in virtually all public key constructs [2].

Although digital signatures can provide authentication, data integrity and non-repudiation cryptographic services, they are not suitable for mobile devices. There have been many studies [1-4] that have dealt with this problem. For example, Asokan et al. [3] proposed a Server-Supported Signature scheme for mobile communication. They used a lightweight computation of the one-way functions and traditional digital signatures. Signature servers were responsible for generating signature tokens and certification authorities to verify these tokens. Therefore the complex computation depended on the reliability of those servers.

Based on the work of Asokan et al., Ding et al. [2] presented a modified digital signature scheme, called Server Aided Signature. In this scheme, users are involved the generation of the signature token. After that, Lei et al. [1] also proposed a Server Based Signature. In their scheme, the certificate concept is involved in their protocol such that Non-Repudiation of Sender (NRS) and Non-Repudiation of Receiver (NRR) can be achieved. In 2005, Bickakci et al. [4] improved the Asokan et al. scheme. All of the above schemes have these common goals: (1) to achieve the same level of security as the traditional digital signature protocols; (2) to reduce the computation complexity of the mobile devices; and (3) to reduce the communication cost between signer and verifier.

* Correspondence author

Next, we consider another issue— key management. The most common method is to store a secret key in a portable storage media (disk or smart card), and then hand it to a legal user, or to store the secret key in a user's computer. In this case, the artificial carelessness or device factors can lead to the key being lost, damaged, stolen, deleted, etc. A secret key that is disclosed will cause a large amount of damage and inconvenience to its. Therefore key management is an important issue. In view of this, Perlman et al. [5] and Sandhu et al. [6] considered storing the secret key in a key server (or appliance). The key owner can then pre-fetch the secret key via secure wired or wireless network during each transaction. In this concept, the secret key is not only mobile but cannot be forged. However, there is a derivative issue: How a user's identity can be authenticated to allow for the secret key to be downloaded. From Perlman and Sandhu's viewpoint, we can use the Encrypted Key Exchange (EKE) [7,8] or Simple Password Exponential Key Exchange (SPEKE) [9,10] method to solve this problem. On the basis of Diffie-Hellman's [11] communication protocol, the common session key of the EKE and SPEKE is constructed via the other party's public key and his own secret key. Afterward, the participating parties can use the session key to encrypt/decrypt sensitive information and communicate securely with each other.

At the moment, the mobile devices are widely used as a tool for making payments. Any concern for non-repudiation transactions is often requested in terms of a digital signature. With the limited computing power of the mobile devices, digital signatures must be verified via proxy server. Moreover, there are many challenges for the mobile commerce [12-17]. Based on the environment of the current mobile commerce, we consider using a proactive password and lightweight hashing function into the mobile devices to be feasible method for solving the limited computation resource.

It is worthwhile to mention that some studies [18-20] focus on authenticating identity for wireless networks to reduce the computational cost of the mobile devices. In order to build a trusting relationship between a mobile user and server, a secret sharing mechanism is a good idea. A mobile user does not need to give his/her secret key to a proxy server. A mobile user and proxy server should cooperate to generate a secret shadow to create a common signature for a verifier to verify. Such a mechanism not only reduces the computational cost of mobile devices but also dispels a user's doubts. Of course, identity authentication can be verified using a proactive password and hashing function. We think that this is a good mechanism that can be used to meet the requirements of the current mobile environment. The detailed scenarios will be described later in Section 3.

The rest of this paper is organized as follows. In Section 2, we describe the related preliminaries and list the requirements. In Section 3, we explain the notation and propose a server-aided signature scheme based on secret sharing for mobile commerce. In Section 4, we analyze the requirements of the proposed protocol. The paper concludes with some final remarks in Section 5.

2 Preliminaries and Security Requirements

We will introduce the related mechanisms and the requirements in this section.

2.1 Preliminaries

The one-way hash function has been used in computer science for a long time. It takes a variable-length input string (called a pre-image) and converts it to a fixed-length input string (called a hash value). A one-way hash function works in one direction: It is easy to compute a hash value from pre-image, but it is hard to generate a pre-image that hashes to a particular value. For example: a function $h : X \rightarrow Y$ is one way if it is easy to compute $h(x)$ for every $x \in X$, yet is hard for most $y \in Y$ to figure out an $x \in X$ such that $h(x) = y$. A more formal definition of one-way functions can be found in [21]. In our scheme, a mobile user must negotiate one set of hashing values $(a_0, a_1, a_2, ..., a_n)$ in advance. It can be generated via one way hash function $h(\ )$ and a_0, where a_0 is a random seed and $a_1=h(a_0), a_2=h(a_1),...,a_n=h(a_n-_1)$. Thus, a mobile user and proxy server can use them and the password to generate a proactive password to authenticate each other's messages. On the basis of the one-way hash function, we think this mechanism can be used in our scheme to overcome the limited computing power of the mobile devices.

A secret sharing mechanism was proposed by Shamir [22]. In some cases, it may be necessary for a group of people to share a certain set of secret data. Shamir proposed the concept of (t, n) threshold secret sharing to solve this problem. The scheme is designed to encode a secret data set D into n pieces $D_1,...,D_n$ and distribute them to n participants, where any t or more of the pieces makes D easily computable, but where any $t-1$ or fewer D_i pieces leave D completely undetermined. Suppose that we pick a random $t-1$ degree polynomial $f(x)=a_o+ a_1x+, . . .,+a_{t-1}x^{t-1}$ in which $a_o=D$. We also pick a prime p which is bigger than both D and n. The coefficients $a_1,...,a_{t-1}$ in $f(x)$ are randomly chosen from a uniform distribution over the integers in $[0, p]$, and the values $D_1,...,D_n$ are computed modulo p, such that $D_1 = f(1), . . .,D_i = f(i) , . . .,D_n = f(n)$.

Given any subset of t of these D_i values (together with their identifying indices), we can find the coefficients of $f(x)$ by interpolation, and then evaluate $D=f(0)$. Knowledge of just $t - 1$ of these values, on the other hand, does not suffice in order to calculate D. For example, there is a polynomial function $f(x)$ which is generated for embedding the common secret key SK_π, where $f(x) = ax + SK_\pi \pmod{\phi(N_\pi)}$, $a \in [1, \phi(N_\pi)]$. From a practical viewpoint, a mobile user does not need to use his own secret key to make a signature. The proxy server only needs to verify the user's identification and use the secret sharing mechanism to generate the common signature (as explained in section 3). This can solve the problem of the limited computing power of the mobile devices.

2.2 Requirements

In terms of the practicability, a server-aided signature scheme for mobile commerce based on secret sharing should satisfy the following requirements:

1. Security: The proposed scheme should protect against the malicious attacks during communication.
2. Non-repudiation: Non-repudiation services protect transacting parties against any denials that a particular event or action has taken place by providing, collecting, and maintaining evidence to enable the settlement of disputes.
3. Efficiency: The communication and the computation cost should be minimized.
4. Simplicity: Because of the weak computing power of the mobile device, the operations of the mobile devices should be designed to be simple as possible.
5. Mobility: The mobile users can conduct their transactions and access network resource at anytime from anywhere.

Based on the above requirements, a comparison of the Asokan et al. [3], Bicakci et al. [4] and our scheme is given in Section 4.

3 The Proposed Protocol

In this section, we will describe a server-aided signature scheme based on secret sharing for mobile commerce. A mobile user gets an application's service via a trusted proxy server such that the application server can get a verified signature. The protocol still needs the original Wireless Transport Layer Security (WTLS) [23] and Secure Socket Layer (SSL) [24,25] to provide end-to-end security. This protocol is divided into two phases: a negotiation phase and an authentication phase. We illustrate the basic architecture of our scheme in Fig. 1. The reset of the scenarios is described below.

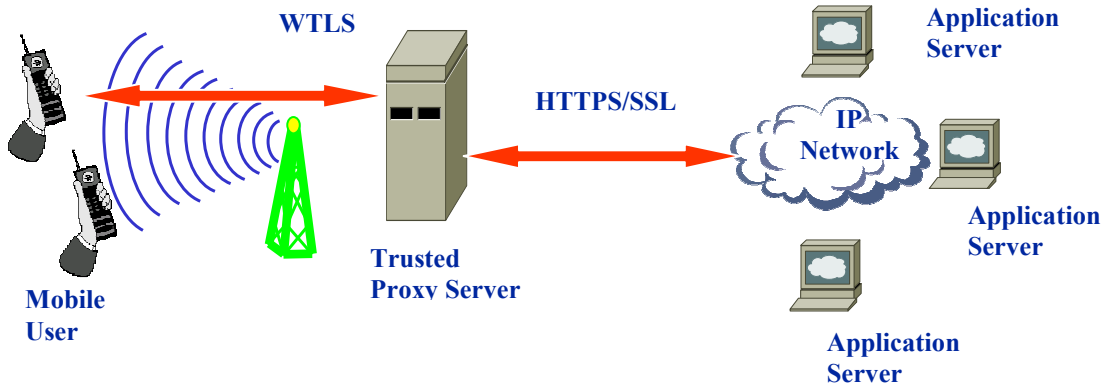

Fig. 1. The basic architecture of our scheme

3.1 Notation

To illustrate our server-aided signature protocol for mobile commerce, the notation used in the scheme is defined as follows:

A	: a mobile user.
B	: the application server.
PS	: a trusted proxy server.
$\|\|$	: concatenate operation.
$+$	: addition operation.
$\oplus$	: exclusive-OR operation.
$h(\ )$	: a one way hash function.
a_0	: a random seed which is negotiated by a mobile user and trusted proxy server in advance such that one set of hashing values ($a_0, a_1, a_2, ..., a_n$) can be generated via the one way hash function $h(\ )$, where $a_1=h(a_0), a_2=h(a_1), ..., a_n=h(a_n\text{-}1)$.
m_{req}	: the request message.
M	: the signed object.
ID_X	: X 's identity.
P_A	: a pre-selected pseudonym of mobile user A.
K	: the symmetric session key.
$E_K(m)$	: use the symmetric key K to encrypt a message m.
$D_K(m)$	: use the symmetric key K to decrypt a message m.
$S_X(m)$	: use X's secret key to sign a message m.
$V_X(m)$	: use X's public key to verify a message m.
PW_i	: the i^{th} password.
(p_x, q_x)	: a pair of large prime numbers.
N_X	: a large number, where $N_X = p_X \cdot q_X$
$\phi(N_X)$	: the Euler totient function, where $\phi(N_X) = (p_X - 1) \cdot (q_X - 1)$
PK_X	: X's public key, where PK_X and $\phi(N_X)$ are relatively prime.
SK_X	: X's secret key, where $PK_X \cdot SK_X = 1 \pmod{\varphi(N_X)}$.

3.2 Negotiation Phase

Since a mobile user (A) and trusted proxy server (PS) want to exchange sensitive data with each other without revealing the information to a third party, they should establish a session key K and pre-defined rules in advance. Afterward, they can use the session key and communication rules to exchange the sensitive data with each other. Because the mobile devices suffer from lack of computing power, we will base on the Diffie et al. scheme [11] and involve the password mechanism to establish session key in advance, and then download the initial parameters into the mobile devices as the communication parameters between the mobile user and the PS. The pre-processing scenarios are depicted in Fig. 2.

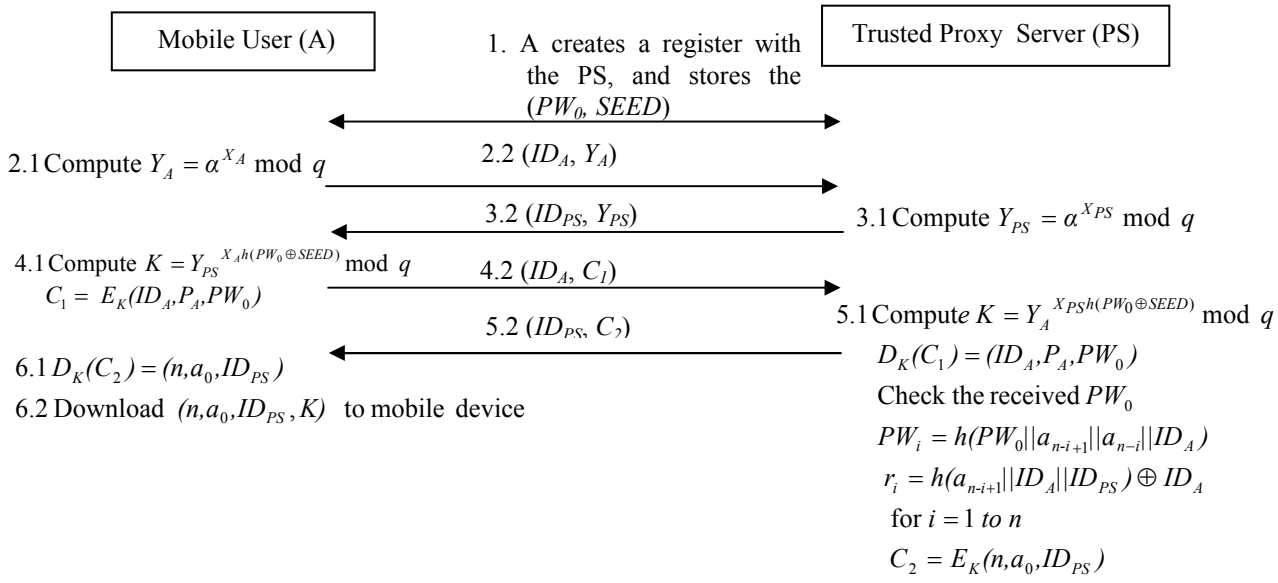

Fig. 2. Protocol of the negotiation phase

Step1: Mobile user A pre-selects an initial password PW_0 and his/her identity ID_A to create a register with the PS. The PS generates a random number *SEED*, and then sends his/her identity ID_{PS} and *SEED* to A.

Step2: We define the global public elements q and α for this phase, where q is a prime number, $\alpha < q$, and α is a primitive root of q. Mobile user A selects a private X_A, $X_A < q$, and calculates public Y_A,

$$Y_A = \alpha^{X_A} \bmod q$$

A sends (ID_A, Y_A) to the proxy server PS.

Step3: The PS selects a private X_{PS}, $X_{PS} < q$, and calculates public Y_{PS},

$$Y_{PS} = \alpha^{X_{PS}} \bmod q .$$

The PS sends Y_{PS} to user A.

Step 4: A computes the session key K as follows:

$$K = Y_{PS}{}^{X_A h(PW_0 \oplus SEED)} \bmod q$$

Afterward, A can use the session key K to encrypt or decrypt the sensitive information.

A pre-selects a pseudonym P_A, and then computes

$$C_1 = E_K(ID_A, P_A, PW_0)$$

Then A sends (ID_A, C_1) to the PS.

Step 5: The PS computes the session key K as follows:

$$K = Y_A{}^{X_{PS} h(PW_0 \oplus SEED)} \bmod q$$

Upon receiving (ID_A, C_1), the PS can use the session key K to reveal the corresponding relationship between ID_A and P_A, and checks whether the initial password PW_0 is correct or not, as follows:

$$D_K(C_1) = (ID_A, P_A, PW_0)$$

If the initial password PW_0 is correct, the PS selects a random seed a_0, then generates and saves one set of hashing values (a_0, a_1, a_2,...,a_n), where $a_1 = h(a_0)$, $a_2 = h(a_1)$,...,$a_n = h(a_{n-1})$. Moreover, the PS also computes and saves the parameters PW_i, r_i and C_2 for the next phase.

$$\left.\begin{aligned} PW_i &= h(PW_0 \,||\, a_{n-i+1} \,||\, a_{n-i} \,||\, ID_A) \\ r_i &= h(a_{n-i+1} \,||\, ID_A \,||\, ID_{PS}) \oplus ID_A \\ C_2 &= E_K(n, a_0, ID_{PS}) \end{aligned}\right\} \text{ for } i = 1,2,\ldots,n$$

The PS sends (ID_{PS}, C_2) to user A.

Step 6: A uses the session key K to decrypt the received message as follows:

$$D_K(C_2) = (n, a_0, ID_{PS})$$

Next, mobile user A downloads (n, a_0, ID_{PS}, K) into his/her mobile device via bluetooth or infrared technology under an off-line model.

3.3 Authentication Phase

Upon establishing the pre-determined parameters, the mobile user A can propose a signature request via mobile device. After verifying A's identity (via P_A to match ID_A, and verify the i^{th} password PW_i), the PS uses A's identity and its own identity to generate the common signature $Sig_\pi = M^{SK_\pi}$ with a secret sharing mechanism via polynomial function *f(x)*. The PS then sends the common signature to the application server B. B uses A and PS's common public key PK_π to verify the request. If the verification is right, B only provides the related service to mobile user A. We will give an example to show the i^{th} request scenarios in Fig. 3.

Step1: Mobile user A inputs the password PW_0' and identity ID_A' to the mobile device. Then the device generates the i^{th} dynamic password PW_i', as follows:

$$a_{n-i} = h(a_{n-i-1})$$

$$a_{n-i+1} = h(a_{n-i})$$

$$PW_i' = h(PW_0' \,||\, a_{n-i+1} \,||\, a_{n-i} \,||\, ID_A')$$

Moreover, mobile user A makes a signature request m_{req}, and computes the following parameters:

$$r_i' = h(a_{n-i} \,||\, ID_A' \,||\, ID_{PS}) \oplus ID_A'$$

$$M_1 = (m_{req} \,||\, i \,||\, r_i' \,||\, P_A \,||\, ID_B \,||\, M)$$

$$X_1 = E_K(M_1, PW_i')$$

A then sends (M_1, X_1) to the PS.

Step2: The PS uses the pre-coordinated session key *K* to decrypt the message (M_1, PW_i') , as follows:

$$D_K(X_1) = (M_1, PW_i')$$

Afterward, the PS also uses the i^{th} hashing values (a_{n-i+1} and a_{n-i}) and the recorded information (A's pseudonym P_A, identity ID_A, and PW_0) to authenticate whether A's identity and password are legal or not, as follows:

$$ID_A \stackrel{?}{=} r_i' \oplus h(a_{n-i} \| ID_A \| ID_{PS})$$

$$h(PW_0 \| a_{n-i+1} \| a_{n-i} \| ID_A) \stackrel{?}{=} PW_i'$$

If the above equalities hold, it means that mobile user A is legal. Therefore, the PS only uses the valid identity ID_A and his own identity ID_{PS} to compute the secret shadows SS_A and SS_{APA} via the following polynomial function *f(x)* which is generated to embed the common secret key SK_π , where $f(x) = ax + SK_\pi \pmod{\phi(N_\pi)}$, $a \in [1, \phi(N_\pi)]$. Let $SK_\pi = SS_A + SS_{PS}$.

$$SS_A = f(ID_A)\frac{-ID_{PS}}{ID_A - ID_{PS}}$$

$$SS_{PS} = f(ID_{PS})\frac{-ID_A}{ID_{PS} - ID_A}$$

The common (A and the PS) signature Sig_π (the Non-Repudiation of Sender) can then be generated as follows:

$$Sig_\pi = (M)^{SS_A + SS_{PS}} = M^{SK_\pi}$$

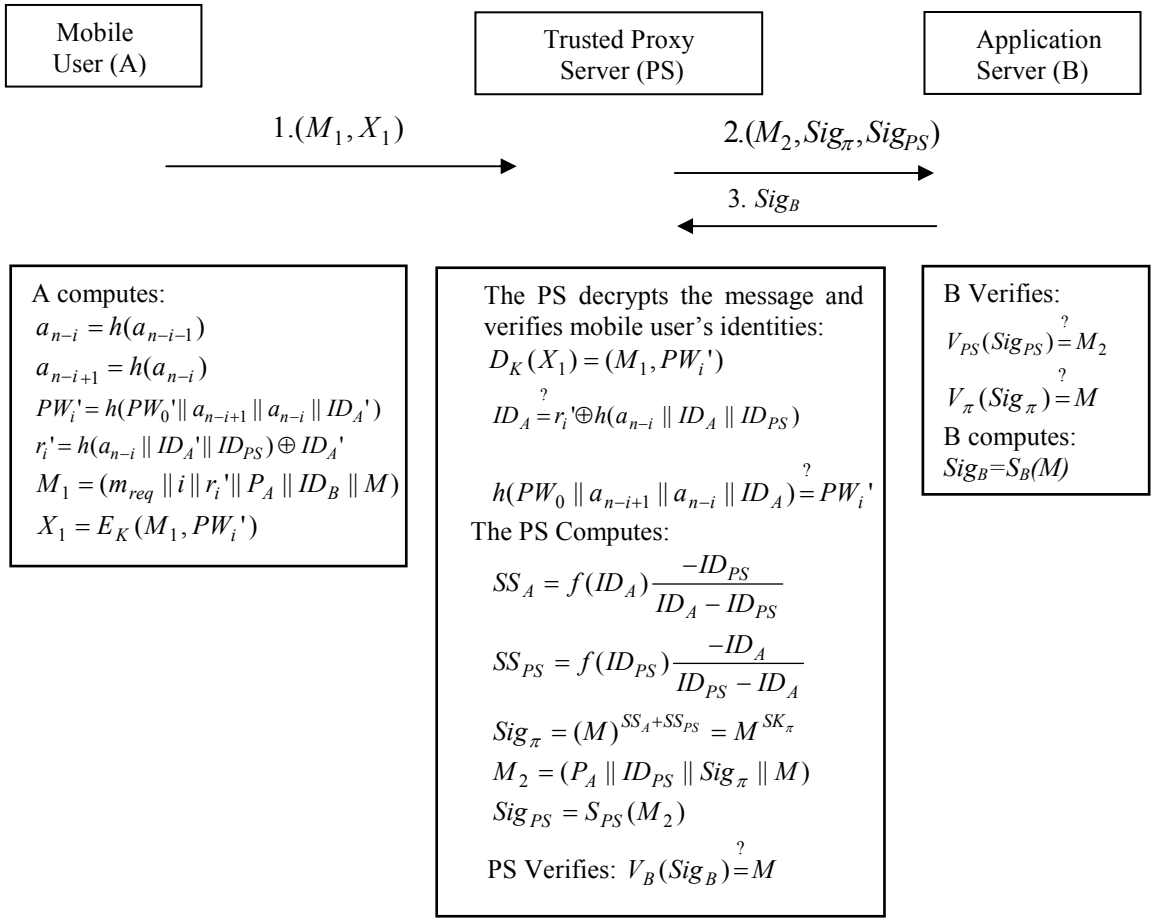

Fig. 3. Protocol of the authentication phase

The PS also computes the M_2 and Sig_{PS}.

$$M_2 = (P_A \| ID_{PS} \| Sig_\pi \| M)$$

$$Sig_{PS} = S_{PS}(M_2)$$

Then the PS sends (M_2, Sig_π, Sig_{PS}) to the application server B.

Step3: Upon receiving the message (M_2, Sig_π, Sig_{PS}), the application server B uses the PS's public key to verify M_2, as follows:

$$V_{PS}(Sig_{PS}) \stackrel{?}{=} M_2$$

The application server B then uses the common (A and the PS) public key PK_π to verify the common signature as follows:

$$V_\pi(Sig_\pi) \stackrel{?}{=} M$$

If the above equalities hold, the application server B only provides the related service to the mobile user A. The application server B computes the signature Sig_B, as follows:

$$Sig_B = S_B(M)$$

The application server B sends Sig_B to the PS as the non-repudiation of the receiver. The PS can verify the correctness as follows:

$$V_B(Sig_B) \stackrel{?}{=} M$$

4 Analysis

We will show that our protocol has met the requirements mentioned in Section two.

4.1 Security Issues

In step 1 of the authentication phase, the i^{th} password $PW_i' = h(PW_0' \| a_{n-i+1} \| a_{n-i} \| ID_A')$, PW_i' is changeable with the hashing values (a_{n-i+1}, a_{n-i}) for each transaction. Even if an attacker intercepts the last password PW_{i-1}', he/she still can not pass the following verifications:

$$ID_A \stackrel{?}{=} r_i' \oplus h(a_{n-i} \| ID_A \| ID_{PS})$$

$$h(PW_0 \| a_{n-i+1} \| a_{n-i} \| ID_A) \stackrel{?}{=} PW_i'$$

Moreover, the password PW_i' is encrypted by the session key K.

$$X_1 = E_K(M_1, PW_i')$$

Such a design can withstand a replay attack and increase the difficulty of a dictionary attack.

4.2 Non-repudiation Issues

Non-repudiation is an important issue in mobile commerce. But the literatures [26-29] can not meet the non-repudiation issues. However, when merchandise price is high or the access information is sensitive, non-repudiation becomes an important issue in transaction. In our scheme, the application server B gets the common signature Sig_π (the proxy server PS and mobile user A's common signature) as the sender non-repudiation. B only provides the relative service for A. B should also send back a signature Sig_B for the PS as the receiver non-repudiation. The mobile user and the application server can not deny this transaction to each other with such a design. We give the non-repudiation proof of the authentication phase in Table 1.

4.3 Efficiency Issues

We show a comparison of Asokan et al.'s scheme, Bicakci et al.'s scheme and our scheme in terms of communication and computational in Table 2 and Table 3 respectively.

Table 1. The non-repudiation proof of the authentication phase

Non-repudiation Evidence	Evidence Issuer	Evidence Holder	Verification Equation
(M_2, Sig_{PS})	PS	B	$V_{PS}(Sig_{PS}) \stackrel{?}{=} M_2$
(M_2, Sig_{π})	A and PS	B	$V_{\pi}(Sig_{\pi}) \stackrel{?}{=} M$
Sig_B	B	PS	$V_B(Sig_B) \stackrel{?}{=} M$

Table 2. Communication comparison of the Asokan et al. scheme, Bicakci et al. scheme and our scheme

	Asokan et al.[3]	Bicakci et al.[4]	Our scheme
Rounds	3	1	2+1 (include 1 round non-repudiation signature for B sends back to PS)
1st round message length	m + h	m + (n+1)h	m + h + k
2nd round message length	m + h + s	m + l+ s	m + 2s
3rd round message length	m +2h + s	—	s

m: length of message, h: length of random numbers and hash values, l: length of server's statement (if it is employed), s: length of signature, n: number of random numbers, k: length of the symmetric encryption.

Table 3. Computation comparison of the Asokan et al. scheme, Bicakci et al. scheme and our scheme

	Asokan et al.[3]	Bicakci et al.[4]	Our scheme
Sender (A)	1H +1V	1H +1M	1N+4H
Proxy Server (PS)	2H +1S	(P+2)H+1M+1S	2H+2V+1N+1EX+2S+2F
Receiver (B)	1V +2H	1V +1H	2V+1S

H: hash computation, S: traditional signing by a public key, V: verification of public key signature,
M: mapping computation (costs less than one hash), P: number of hash computations to verify signature,
N: symmetric encryption/decryption operation, EX: exclusive –OR operation, F: polynomial operation of the secret sharing.

Table 2 gives a comparison between the three protocols. Although our scheme has an extra round of communication cost than [4], there is only one round of communication cost between mobile user A and the proxy server PS. However, the non-repudiation issue is worthy of reconsideration. Our scheme has one extra length of signature in the 2nd and 3rd rounds. But it is designed to meet the non-repudiation issue. The proposed protocol is devoted to handling more complete transaction scenarios in mobile commerce. The current literature often neglects this non-repudiation issue. We adapted symmetric encryption in the first round. Perhaps the communication cost is higher than that of the other schemes, but there is better security.

Table 3 shows a comparison of the Asokan et al. scheme, Bicakci et al. scheme, and our scheme with respect to on-line computational requirements for the participating entities (off-line pre-computations are not included).

From the above analyses, in order to increase trust in the relationship between a mobile user and the proxy server, we use the secret sharing mechanism in our scheme to enhance the security of a business transaction or to access the important resources. The client load will not overload, and the overall performance is still satisfactory.

4.4 Simplicity Issues

Due to the weak computing power of the mobile devices, we pre-process the negotiation parameters (n, a_0, ID_{PS}, K) in advance under an offline model. Afterward, the mobile devices can only perform simple operation (for example: exclusive-OR and symmetric encryption/decryption operation) to carry out any transactions. The proxy server performs the complex operations. In this way, we not only overcome the limited computational capabilities of the mobile devices but achieve the general transaction requirements.

4.5 Mobility Issues

Mobile users can communicate with the proxy server via mobile communication network. Once they pass the server's authentication, they can conduct their transactions and access the network resources at any time from anywhere.

5 Conclusions

To enable mobile users to conduct their business or access the network resource at any time from anywhere, we proposed a practical server-aided signature scheme. Using verification and secret sharing mechanism, this scheme is more secure than prior studies. Based on a one-time password, attackers cannot intercept the last password to generate a valid password and masquerade as the legal user. Our scheme also satisfies the transaction non-repudiation requirement between a mobile user and the application server.

In addition, a mobile device only performs simple operations, while the server executes complex operations. The proposed scheme successfully overcomes the inherent shortcomings of mobile devices which lack the computing resources.

References

[1] Y. Lei, D. Chen and Z. Jiang, "Generating Digital Signatures on Mobile Devices," *Proceedings of the 18th International Conference on Advanced Information Networking and Applications (AINA'04)*, Fukuoka, Japan, Vol. 2, pp.532-535, 2004.

[2] X. Ding, D. Mazzocchi, and G. Tsudik, "Experimenting with Server-Aided Signatures," *Proceedings of 2002 Network and Distributed System Security Symposium (NDSS'2002)*, 2002.

[3] N. Asokan, G. Tsudik, M. Waidner, "Server-supported signatures," *Journal of Computer Security*, Vol. 5, No. 1, pp. 91–108, 1997.

[4] K. Bicakci and N. Baykal, "Improved server assisted signature," Computer Networks, Vol.47, pp.351-366, 2005.

[5] R. Perlman and Charlie Kaufman, "Secure Password-Based Protocol for Downloading a Private Key," *Proceedings of the Network and Distributed System Security Symposium (NDSS '99)*, San Diego, California, 1999.

[6] R. Sandhu, "Password-Enabled Public-Key Infrastructure (PKI) and Role-Based Access Control (RBAC) on the Secure Identity Appliance," *Proceedings of ISC (Information Security Conference)*, Taichung Taiwan, 2002.

[7] S. Bellovin and M. Merritt, "Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks," *Proceedings of the 1992 IEEE Symposium on Research in Security and Privacy*, Oakland, California, pp.72 – 84, 1992.

[8] S. Bellovin and M. Merritt, "Augmented Encrypted Key Exchange: a Password-Based Protocol Secure Against Dictionary Attacks and Password File Compromise," *Proceedings of the First ACM Conference on Computer and Communications Security*, pp. 244-250, 1993.

[9] D. Jablon, "Strong Password-Only Authenticated Key Exchange," *ACM Computer Communications Review*, Vol. 26, No.5, pp.5-26, 1996.

[10] D. Jablon, "Extended Password Protocols Immune to Dictionary Attack," *Proceedings of the WETICE '97 Enterprise Security Workshop*, June 1997.

[11] W. Diffie, M.E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, Vol. 22, No. 6, pp.644–654, 1976.

[12] D. Boneh and N. Daswani, "Experimenting with electronic commerce on the Palm Pilot," *Proceedings of 1999 Financial Cryptography*, pp. 1–16, 1999.

[13] S. S. Grosche and H. Knospe, "Secure Mobile Commerce," *Electronics & Communication Engineering Journal*, Vol. 14, No. 5, pp.228-238, 2002.

[14] A. Raghunathan, S. Ravi, S. Hattangady and J.-J Quisquater, "Securing mobile appliances: new challenges for the system designer, Design, Automation and Test," *Europe Conference and Exhibition*, pp.176 –1 81, 2003.

[15] A. Tsalgatidou and E. Pitoura, "Business Models and Transactions in Mobile Electronic Commerce: Requirements and Properties," *Computer Networks*, Vol.37, pp. 221-236, 2001.

[16] A. Tsalgatidou, J. Veijalainen and E. Pitoura, "Challenge in Mobile Electronic Commerce," *Proceedings of the 3rd Int. Conf. On Innovation through E-Commerce*, UK, 2000.

[17] J. Veijalainen, V. Terziyan and H. Tirri, "Transaction Management for M-Commerce at a Mobile Terminal", *Proceedings of the 36th Annual Hawaii International Conference on System Sciences*, Jan., 2003.

[18] M. Badra, A. Serhrouchni and P. Urien, "A lightweight identity authentication protocol for wireless networks," *Computer Communications*, Vol. 27, pp.1738–1745, 2004.

[19] E. Bresson, O. Chevassut and A. Essiari and D. Pointcheval, "Mutual authentication and group key agreement for low-power mobile devices," *Computer Communications*, Vol. 27, pp.1730–1737, 2004.

[20] H. Y. Lin and Lein Harn, "Authentication Protocols for Personal Communication System," *Proceedings of the 1995 conference on application, computer, communication*, Cambridge, MA, USA, pp. 256-261, 1995.

[21] S. Goldwasser, The search for provably secure cryptosystems, *Proceedings of Symposia in Applied Mathematics*, Vol. 42, pp. 89–113, 1990.

[22] A. Shamir, "How to share a secret," *Communications of the ACM*, Vol. 22, No. 11, pp. 612-613, 1979.

[23] Wireless Transport Layer Security Specification, *WAP Forum, 2001*, http://www.wapforum.org/, accessed March 2007.

[24] A. O. Freier, P. Karlton and P. C. Kocher, "The SSL Protocol Version 3.0," *Internet Draft*, March 1996.

[25] D. Wagner and B. Schneier, "Analysis of the SSL 3.0 protocol," *Proceedings of the Second USENIX Workshop on Electronic Commerce*, USSENIX Press, pp. 29-40, 1996.

[26] B. Ozen and O. Kilic, "Highly Personalized Information Delivery to Mobile Clients," *Wireless Networks*, Vol. 10, No. 6, pp.665 – 683, 2004.

[27] N. M. Sadeh, T. Chan, L. Van, O. Kwon and K. Takizawa, "A Semantic Web Environment for Context-Aware M-Commerce," *Proceedings of the 4th ACM conference on Electronic commerce*, San Diego, CA, USA, pp. 268 – 269, 2003.

[28] G. Shih and S. S.Y. Shim, "A Service Management Framework for M-Commerce Applications," *Mobile Networks and Applications*, Vol. 7, No. 3, pp.199 – 212, 2002.

[29] Z. Trabelsi, S. Cha, D. Desai, C. Tappert, "A voice and ink XML multimodal architecture for mobile e-commerce systems," *Proceedings of the 2nd international workshop on Mobile commerce table of contents*, Atlanta, GA, USA, pp.100-104, 2002.

[30] N. J. A. Sloane and A. D. Wyner (editors), *Claude Elwood Shannon : collected papers*, New York, IEEE Press, 1993.