Analysis of a PKI-Based Secure Infrastructure for Mobile E-Commerce

Anirudh Kumar Srivastava#1, Aasha Singh*2, Awadhesh Kumar#3

[1]Computer Science & Engg. , KNIPSS Sultanpur, INDIA

[3]*Computer Science & Engg. , KNIT Sultanpur, INDIA*

[1]ani.cse2005@gmail.com

[2]awadhesh@knit.ac.in

[2]*Computer Science & Engg. , KNIT Sultanpur, INDIA*

[2]aashapranay@gmail.com

Abstract— **The development of mobile devices and the public key infrastructure （PKI） have improved the rapid development of mobile e-commerce. However, there exits some challenges such as limit computing capacity for PKI-based secure transactions. This paper presents a new system architecture which includes client, mobile operator, service provider, certificate authority and so on. On this basis, the protocols for authentication and key exchange that is suitable for the mobile e-commerce environment are designed to support some applications.**

Keywords— **public key infrastructure (PKI), mobile e-commerce, key exchange protocols**

I. INTRODUCTION

The rapid advances in wireless mobile communication technology and the growth of electronic commerce have naturally led to the development of electronic commercial services on the wireless medium through mobile phones. For business transactions conducted on electronic means, security is a major concern. Both the Internet and the wireless network are public networks and considered to be insecure, where messages can be eavesdropped, captured, modified and inserted by intruders. Intruders may also impersonate as legitimate parties for personal gain. Therefore, some mechanism is needed to guarantee the confidentiality, authenticity and integrity of the transmitted messages [1]. Internationally, the Public Key Infrastructure (PKI) is accepted as an effective means to tackle the above security problem. Our objective is to develop a PKI-based open infrastructure that supports end-to-end secure electronic transactions through mobile phones. Besides the security concerns, efficiency and availability of supporting hardware products are also important. The main challenge is the resources on current SIM cards are not sufficient to perform general PKI-based authentication [2]. Moreover, the wireless network is error-prone and slow compared to wired networks. We have modified a set of key exchange and authentication protocols that can run on a thin client model.

II. PAGE LAYOUT

Due to the scarce resource for both memory space and computational power, the Mobile Equipment (ME) is incapable of verifying a X.509 digital certificate to authenticate a service provider [3]. We have developed a server called the User Authentication Server (UAS) to act as a trusted third party to assist the mobile client to authenticate and exchange keys with the service provider, which is named the PKI End-to-end Secure Module (PESM).The diagram is depicted in Fig. 1.

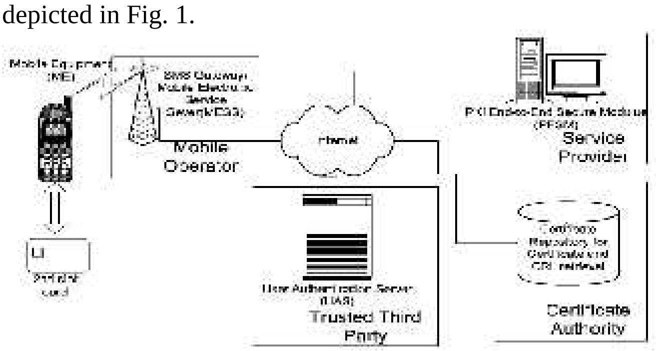

Figure 1. The system architecture

A. An Mobile Client

The mobile client is a portable device, which in our case consists of a dual slot GSM phone and a smart card with cryptographic functionality. Each user is required to have his own digital certificate issued by a valid Certification Authority (CA). The corresponding private key is stored in the user's second slot smart card. Moreover, we require the UAS's public key be pre-loaded on the card as well. In subsequent sections, the mobile client is abbreviated as ME (Mobile Equipment) for simplicity. In this case, the personal information is directly stored in the PK-SIM card.

B. SMS Gateway and Mobile Electronic Service Server (MESS)

The SMS Gateway and MESS together act as an interface between the wireless and wired networks. MESS interprets the header of message packets and routes the packets to the proper MEs and servers. It is unable to read the message contents since they are encrypted at source.

C. User Authentication Server (UAS)

The UAS is a centralized server that should be operated by a trusted third party. Its role is to help the ME to authenticate the party it is communicating with. First, mutual authentication is performed between the ME and UAS. Then, the UAS authenticates the PESM on behalf of the ME. Following that, a PESM session key is exchanged between the ME and PESM to establish an end-to-end secure communication channel.

D. PKI End-to-end Secure Module (PESM)

PESM is a server operated by the service provider. It is responsible for ensuring security at the application level, includes authentication, confidentiality and integrity. For authentication, it performs the handshake protocol to authenticate the UAS or optionally authenticate the mobile client and establishes a session key. For confidentiality, it encrypts and decrypts messages sent and received from the mobile client using the established session key. Furthermore, it verifies the Message Authentication Code (MAC) of each message to guarantee integrity. For non-repudiation, it verifies the digital signature of a message if it is present.

E. Certificate repository

The certificate repository is a service provided by the CA, which allows the public to access the issued digital certificates. It is usually implemented by a LDAP server on which object records can be searched by subjects. The UAS and PESMs will access this server from time to time to retrieve digital certificates for verification purposes.

III. DESIGN OF PROTOCOLS

The protocol is based on a 3-tiered model involving the ME, UAS and PESM. With the assistance of the UAS, the ME and PESM establish a session key [4]. Prior to key establishment, authentication is required between the ME and UAS, and then the UAS and PESM. Therefore, the protocol is divided into 2 phrases, namely: UAS Session Establishment and PESM Session Establishment. The notations in describing the protocol are presented in Table 1.

TABLE I. SYMBOLS USED IN PROTOCOL DESCRIPTION

Symbol	**Description**
ID*p*	A unique identifier of entity *P*
E*p{x}*	Encrypt *x* by *P*'s public key
Cert*p*	Digital certificate of *P*
Hash*{x}*	Hash the value *x*
f(x)	Some kind of one-way function for session key Diversification
$N \in R\{0,1\}_k$	Randomly generate *k* bits of binary data *N*
A ‖ B or A, B	*A* concatenates with *B*
Ver	Version of the protocol
Na	A random number generated by ME
Seq	A random number generated by ME as the starting sequence number of this session
Nb	A random number generated by UAS
USKey	UAS Session Key calculated from *f (Na‖Nb)*
KeyPolicy	A value defining the lifetime of *USKey*
ESKEY*{x}*	Encrypt by symmetric key block cipher in CBC mode (3DES) with the key "KEY"
MachAttr	Configuration attribute of the ME (e.g., language)

A. UAS Session Establishment

The session is established between the ME and UAS based on a general challenge–response authentication mechanism (Fig. 2). The ME initiates the establishment of a secure with the UAS by performing the following operations:

- Randomly generates *Na* and *Seq*.
- Encrypts (hash {*Cert* UAS}, *IDME, Na, Seq*) using UAS's public key.
- Composes and sends *ukey_session_req* to UAS.

When UAS receives the *ukey_session_req* message, it should:

- Decrypt the message using its own private key.
- Check if hash {Cert UAS} is the fingerprint of its current certificate. If the check fails, the protocol cannot be continued since the ME does not have the correct public key of the UAS.
- Randomly generate Nb and calculate *USKey* (UAS session key) from *f (Na||Nb).*
- Determine the lifetime of the session key and specify it in the value of *KeyPolicy*.
- Compose and send ukey_session_resp to ME.

On receiving the ukey_session_resp message the ME verifies the validity of the message by generating its own value of Hash {*Ver, IDME, USKey, KeyPolicy*} and comparing it with the one in the message Since only the valid UAS can decrypt ukey_session_req to get the value of Na, ME can authenticate UAS by checking the correctness of the ukey_session_req message. If the message is correct, ME accepts US Key and *Key Policy*.

In the above protocol, only one-way authentication of UAS is achieved. Adversaries can impersonate the ME by creating its own ukey_session_req message. Therefore, the UAS does not accept this newly established session yet. Instead, it stores the state parameters of the session as a pending state and switch to the current state only after the ME has further authenticated itself in the PESM Session Establishment protocol that follows.

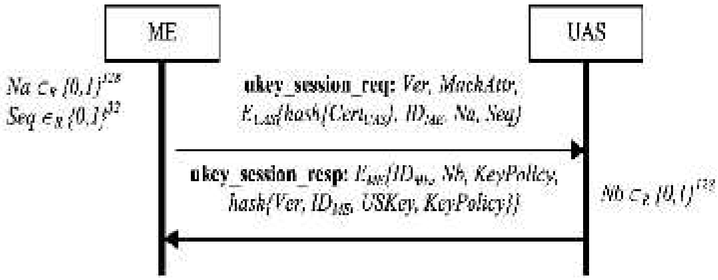

Figure 2. UAS Session Establishment

B. PESM Session Establishment

After the UAS session has been established (either in pending state or current state), the ME may start the PESM Session Establishment protocol in order to establish a secure communication session with the service provider. A request is sent by the ME to the UAS specifying which PESM it would like to talk with [5]. The UAS then communicates with the target PESM on behalf of the ME. Before the UAS can start the key exchange protocol with a PESM, it may have to interact with the PESM to find out the key exchange mode required and exchange the related certificates. If the information is already known then this step can be skipped. The PESM can choose from two authentication modes of session key establishment when it receives an enquiry: Server Authentication and Client Authentication. Server authentication means the PESM does not need to authenticate the ME. Otherwise, Client Authentication mode is used.

1) *Server Authentication.* If Server Authentication mode is selected, the protocol runs as in Fig. 3.

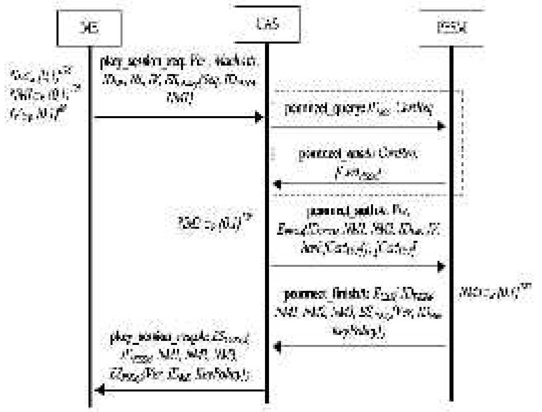

Figure 3. Protocol for PESM Session establishment

The ME initiates the protocol with the following actions:

- Randomly generates *NM1* and *Nx* and calculates *UEKey* from *f (USKey||Nx)*.
- Increments *Seq*.
- Encrypts (*Seq, IDPESM, NM1*) using *UEKey*.
- Composes and sends pkey_session_req to UAS. When UAS receives the pkey_session_req message, it
- Computes *UEKey* using the received *Nx* and its own *USKey*.
- Decrypts the message using *UEKey*.
- Checks if the value of *Seq* is valid. UAS will only accept *Seq* if it is larger than the last accepted *Seq* but falls within a certain predefined range. This mechanism is to avoid intruder's attack by replaying the pkey_session_req message.
- If this message is valid, UAS switches session state from "pending" to "current".
- If UAS has no information about the mode of authentication or the certificate of PESM, a pconnect_query message is sent to the PESM. After receiving the pconnect_ansA response, UAS checks if the PESM's certificate was issued by one of the CAs listed in the non-empty *CertReq*. If it does not check out, the session cannot be established.
- Randomly generates *NM2*.
- Encrypts the elements in the pconnect_authA message using PESM's public key and sends the message to PESM.On receiving the pconnect_authA message, the PESM Decrypts the message using its private key.
- Checks if the UAS certificate fingerprint in the message matches that of the certificate.
- Randomly generates NM3 and computes the *PSKey* by *f (NM1||NM2||NM3)*.
- Determines the lifetime of the *PSKey* and assigns the value of *KeyPolicy*.
- Encrypts (*Ver*, *SRN*, *KeyPolicy*) using *PSKey*.
- Composes the pconnect_finishA message, encrypts it using the public key of UAS and sends to UAS.

On receiving the message pconnect_FinishA, the UAS can authenticate the PESM by checking if the values of *NM1* and *NM2* are the same as what were sent in the pconnect_authA message [6]. This is, again, a simple challenge–response mechanism since the values of *NM1* and NM2 can only be obtained by the holder of PESM's private key. After authenticating the identity of PSEM, the UAS forwards the needed data to ME needed to calculate the PESM session key.

And among the variables, *Nx* means A random number generated by ME; *NM1* means A random number generated by ME; *NM2* means A random number generated by UAS; *NM3* means A random number generated by PESM; *IV*, means Initialization vector for CBC mode encryption; *UEKey* means UAS Encryption Key which is calculated from *f(USKey_Nx)*; *PSKey* means PESM Session Key which is calculated from *f(NM1_NM2_NM3)*; *CertReq* means A list of CAs which is recognized by the sender. If this list is empty, it means certificate is not requested.

IV.CONCLUSIONS

A secure architecture is important for the development of mobile e-commerce. And a PKI-based secure architecture involves three parties, that is to say, the mobile client, the service provider, and a trusted third party. Among them there must be key exchange protocols to protect the application of infrastructure. The application is being used in real-life for purchase and payment, which includes the credit card number transmitted from the second slot smart card on the mobile client to the payment server of the merchant's acquirer bank.

With the development of mobile e-commerce, more and more attentions are being paid to the security. Therefore, the application of PKI-based secure infrastructure will be more popular.

V.LIMITATION AND FUTURE WORK

Paper clearly leaves a scope of improvement because it just discusses about a one way authentication and not mutual authentication in details. So it does not provide strong authentication. Further, the authentication is between a mobile device and a server whereas in general practice, the best case scenario would be to do it between 2 mobile devices, exploiting the computation power of the intermediate servers. Hence, this also has a very good scope for further improvement and research.

Another aspect that has not been explored is the battery drain if this mechanism is applied. Yet another practical and significant area of research in terms of performance improvement.

REFERENCES

[1] A. Aziz and W. Diffie, "Privacy and authentication for wireless local area networks," IEEE Personal Commun.1, 1994, pp. 25–31.
[2] S. Blake-Wilson, D. Johnson, and A. Menezes, "Key agreement protocols and their security analysis," in Sixth IMA Internat. Conf. on Cryptography and Coding, December 1997.
[3] Xiong L, Liu L, Peer Trust: Supporting reputation-based trust for peer-to-peer electronic communities. IEEE Trans Knowl Data Eng 2004, 16(7), pp. 843–857.
[4] Wang Y, Lin FR, "Trust and risk evaluation of transactions with different amounts in peer-to-peer ecommerce environments". In: Proceedings of the IEEE International Conference on e-Business Engineering (ICEBE2006), Shanghai, China, 2006, pp. 102–109.
[5] K. H. Lee and S. J. Moon, "AKA protocols for mobile communications," in Proc. of the 5th Australasian Conf. on Information Security and Privacy (ACISP 2000), 2000, pp. 400–411.
[6] C. H. Lim and P. J. Lee, "Several practical protocols for authentication and key exchange," Inform. Process.Lett. 53, 1995, pp. 91–96.
[7] H.-Y. Lin and L. Harn, "Authentication protocols with nonrepudiation services in personal communication systems," IEEE Commun. Lett., 1999, 3(8), pp. 236–238.

