A Proposed Architecture for Secure Two-Party Mobile Payment

J. E. Rice and Y. Zhu
Dept. of Math & Computer Science
University of Lethbridge
4401 University Dr. W., Lethbridge, AB, Canada
{j.rice, yunpu.zhu}@uleth.ca

Abstract

The evolution of wireless networks and mobile devices has resulted in increased concerns about performance and security of mobile payment systems. In this paper we propose a new secured architecture for two-party mobile payments, e.g. *mobile banking. The proposed architecture employs a lightweight cryptography system that combines public key and symmetric key cryptography systems (ECDSA and AES), as well as a multi-factor authentication mechanism. These are coupled with a transaction log strategy to satisfy the properties of confidentiality, authentication, integrity and non-repudiation. Compared to some existing mobile payment platforms the proposed architecture is a lightweight secured mechanism that is more suitable for two-party banking transactions over resource-limited mobile devices.*

1. Introduction

Wireless networks and mobile devices are becoming more and more widely used. At the same time, in part due to the prevalence of non-protected public transmissions over wireless systems, security issues are becoming more and more problematic. In this paper we propose a a new secured architecture for mobile banking/payments. We propose use of a lightweight cryptography system along with a multi-factor authentication mechanism and a transaction log strategy to ensure all security requirements are fulfilled. We argue that compared to various architectures and models such as [7], [9], [18] and [1], the architecture we propose is simpler and better suited to two-party mobile payment transactions over resource-limited mobile devices.

2. Background

Some background in this area may be of benefit in understanding the proposed architecture.

In our research, mobile devices are recognized as handheld devices generally with internet browsing capability and other basic computational capabilities. A mobile device can be viewed as an identifier for a particular individual, in that each individual generally has one's own mobile device which is not usually shared with others.

Mobile payment can be defined as any payment transaction which involves a mobile device [2]. According to Gao *et al.* existing mobile payment systems can be classified into two types: mobile POS payment systems that enable customers to purchase products on vending machines with their mobile devices, and account-based payment systems which can be mobile phone-based, smart card or credit card-based [5]. It is on this latter type that we focus. The two-party payment model of mobile payments is the simplest type of mobile payment model. The two parties involved are assumed to be a customer and a financial service provider.

There are two common channels that can be recognized as wireless networks: the wireless local area network (WLAN) and the mobile phone network [17]. The mobile phone network, on which our research focuses, is a radio network which consists of a number of cells, each of which is served by one or more fixed transmitters [16].

In order to provide security for two-party transactions an implementation is generally expected to reside either on the transport layer or on the application layer [7]. Our architecture is proposed on the application layer. The application layer's security architecture is independent of the lower layers' security protocols, and is designed such that the application handles all the security-related functions. Furthermore, implementing an application layer security architecture does not require modifications to the current wireless network's infrastructure and protocols.

[11] details a number of concerns held by customers regarding security in mobile payments. Table 1 summarizes the requirements resulting from these concerns, and technologies recommended to address them. The third column of Table 1 describes the specific solutions proposed in this

paper for addressing each of the concerns.

Security Requirement	Technology	Solution
Authentication	Possession	mobile device
	Knowledge	PIN
	Property	userid/password
	Digital Signature	ECDSA
Integrity	Digital Signature	
Non-repudiation	Digital Signature	
	Log	Business Transaction Log
Confidentiality	Encryto/Decrypto	AES

Table 1. Security requirements for mobile transactions, along with the technologies recommended for these requirements and solutions to address them.

3. Related Work

In this section we give an overview of some related work.

3.1. J2ME Application-Layer End-to-End Security Architecture

Itani and Kayssi present JASA, an application-layer security architecture based on the Java 2 Platform, MicroEdition (J2ME) [7] to ensure end-to-end security for m-commerce. JASA uses pure Java components to provide end-to-end security between a wireless J2ME-based client and J2EE-based servers. As in our proposal, this solution also does not require any modification to the underlying protocols or wireless network infrastructure.

JASA consists of a client side and server side. The client application consists of a Mobile Information Device Profile (MIDP) on top of the Connected Limited Device Configuration (CLDC) profile. These provide the necessary methods needed for encapsulating an HTTP connection, and the result is that JASA can be easily employed in the current wireless network environment. AES is employed to provide encryption and decryption. On the server side the application is specified for the Java 2 Platform Enterprise Edition (J2EE) and intended to be deployed on an J2EE application server such as IBM WebSphere or Sun Glassfish.

One disadvantage of JASA is that AES is a symmetric ciphering algorithm, meaning that the server and the client share the same key. AES can not guarantee non-repudiation in the transaction between two parties.

3.2. Lightweight Security for Mobile Commerce Transactions

K. Lam *et al.* proposed a lightweight security mechanism (LSM) [9] for protecting electronic transactions over handheld devices. The concept of a wireless protocol gateway was introduced in this proposal. A wireless protocol gateway serves as a fixed-line agent for the handheld devices. Handheld devices are connected to the gateway via the mobile phone network and the gateway is connected to the application server via a fixed line network.

Transactions in LSM are implemented by a combination of the wireless protocol gateway and an end-to-end security mechanism. The mechanism presumes that the mobile handheld device supports plug-in or applet implementations in an Internet browser environment. In communications between the handheld device and the wireless protocol gateway an authentication protocol is established through the sharing of a symmetric secret key. The fixed line network communication between the wireless protocol gateway and the application server consists of a combination of public key cryptography and simple password authentication. A tamper-resistant hardware device is suggested in order to ensure non-repudiation.

While LSM meets the security requirements of mobile commerce in authentication, confidentiality, integrity and non-repudiation, there are some disadvantages of this proposal. One disadvantage is that there is a security gap at the wireless protocol gateway. The wireless gateway receives the traffic from handheld devices, decrypts the traffic with the symmetric key and encrypts them again using public key cryptography, then sends the data to the application server. This can result in exposure of the data.

Another disadvantage is the way that the mobile transaction is implemented on the mobile phone network. The mobile phone network is provided by mobile network operators, while the mobile application service is offered by application providers such as banks. Application providers may not want mobile network operators to be involved in their security planning; however in LSM it is essential for both the client and server to keep their cryptography key pairs away from third parties, and so some joint collaboration to achieve this may be required.

3.3. iKP and SET

iKP (Internet Keyed Payment Protocols) is a group of secure payment protocols developed by IBM Research Division [8, 1]. All iKP protocols are based on RSA public key cryptography. However, the number of public keys (which

is sometimes referred to by the i in iKP) is varied according to the particular business requirement. This number is mirrored in the name of the individual protocols: 1KP, 2KP, and 3KP. The simplest protocol, 1KP, only asks for one of the three parties involved to hold a public key.

The Secure Electronic Transaction (SET) specification is an open encryption and security specification designed to protect credit card transactions on the Internet. Various big-name companies collaborated in the development of SET and SET is currently supported by major corporations such as VISA Inc. and MasterCard. As a standard protocol SET has been defined to ensure the security of credit card payments on the Internet, but its transaction flow and implementation of security can also be applied to wireless networks. Details of the SET specification can be found in [18].

Both SET and iKP are credit-card payment protocols. Although they have been successfully implemented for e-commerce on a wired network, they are too heavy-loaded to operate on resource constrained devices such as mobile devices and wireless networks. SET and iKP are comprehensive architectures, not specifically designed for two-party payment transactions; this makes SET and iKP too complex for efficient and realistic implementations of two-party mobile payment transactions.

4. Proposed Architecture

We propose a new secured architecture for two-party mobile payment (SA2pMP) based on JASA and LSM. SA2pMP implements a digital signature module and a transaction log strategy to solve the problem of non-repudiation, which was not ensured in JASA. SA2pMP's proposed structure overcomes the security gap at the wireless protocol gateway that is found in LSM, and as well the physically distributed transaction log strategy lowers the wireless network operator's involved sensitivity and increases its capability as a third trusted auditor.

4.1. Network Module

The architecture proposed for SA2pMP is designed for banking transactions to be carried out on mobile devices. In a normal banking transaction there are two parties involved: the customer and the bank. The customer taking part in our banking transaction is also the mobile device's holder and owner. The customer's mobile device communicates with the bank's mobile server via HTTP/HTTPs. As illustrated in Figure 1, the mobile device connects with a network gateway through the mobile network provided by a network operation provider. For example in Canada Rogers provides its users data service which enables users' mobile devices to access HTTP/HTTPs applications through their mobile network. Wired networks connect the banking system and the network gateway. Except for constraints in network bandwidth and mobile devices, this physical network architecture is transparent to the mobile banking platform.

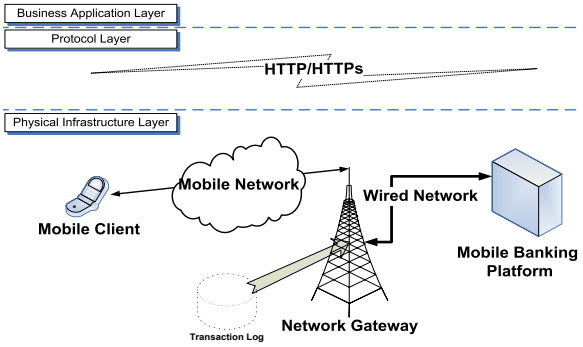


Figure 1. Network module for SA2pMP.

4.2. Security Mechanism

As Figure 2 illustrates, data are transferred in a open, public wireless network environment. This is not a secured network environment. During a transaction the system combines the transaction message and the digital signature's public key and both are transferred over this unsecured network link. It is not necessary to encrypt a public key used in a digital signature system, and so we can transfer the public key over an open wireless network. The transaction message (*msg*) must be protected from third party eavesdropping, and so we use the signature layer and the encryption layer to process *msg*. This means that in our architecture the digital signature layer is not independent of the encryption layer. Another reason for crossing the signature layer and the encryption layer is to ensure the message is sent from a specific client to a specific server. For these reasons, we combine the *SIM* (Subscriber Identifier Module Number), *PHID* (mobile phone serial number, or PHone IDentifier), and *ACCID* (user's bank account number, or ACCount IDentifier) as the identifier ID_C (client ID), then sign this together with the message. This is described by Equation 1 where E refers to the encryption of the message and S refers to the digital signature signing operation.

$$Client \rightarrow Bank : E(S(msg, ID_C)) \quad (1)$$

Cryptography The Elliptic Curve Digital Signature Algorithm (ECDSA) is one approach to implementing the Digital Signature Algorithm (DSA). ECDSA has been applied to wireless networks because of its low computational cost and short key size, both of which reduce the overhead

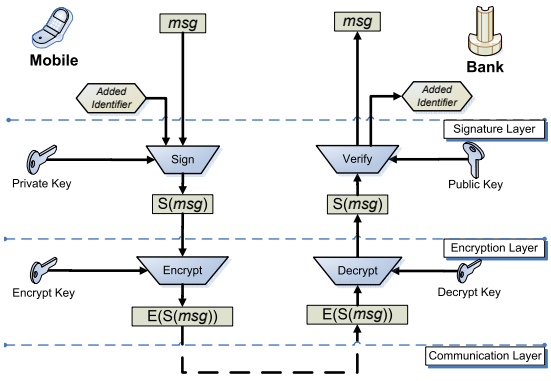


Figure 2. Security scheme for SA2pMP.

in a wireless and resource constraint environment [19]. Details of ECDSA can be found in [4]. The key size is 192 bits which is chosen based on [10] and [15]. The algorithm uses SHA-1 for message digesting. AES is employed for encryption and decryption.

Multi-factor Authentication Strategy Authentication is concerned with ensuring that a communicating entity is the one that it claims to be [14]. To meet recommendations in [3], our proposal should provide strong authentication. To satisfy this we chose these factors from [12]:

- Something you have: A mobile device is an physical object, thus the user's possession of this physical object can be one factor in ensuring authentication.
- Something you know: The mobile banking platform is an integrated banking system. As such it is a part of banking infrastructure and each user has his/her own userid/password for this system.
- Something you have: Mobile devices need the wireless connection services offered by telecommunication providers, and a mobile device must possess a mobile phone number offered by wireless network services. This number plays the role of the Personal Identity number.
- Something the user is or does: Digital signature is an important technology which also provides authentication of a message.

Transaction Log Strategy We suggest a new cooperative relationship between the financial service provider and the wireless network operator. Along with the digital signature a transaction log strategy is used to ensure non-repudiation. The transaction log server is a security mechanism to protect a bank from repudiation between users and bank. If a user refuses to admit participating in a mobile transaction, the transaction log server can provide the transaction records as proof. The definition of the network protocol gateway in LSM [9] is used to realize our business transaction log server. Illustrated by Figure 1, the network gateway is provided by the mobile network operator, which is not in the same business unit with the bank. The mobile network operator objectively takes the role as the third trusted auditor. In that, the transaction log server is logically a part of the mobile banking platform, but in our architecture we suggest that it be physically positioned on the network gateway.

4.3. Key Management

Key management is concerned with the secure generation, distribution, and storage of keys [13]. Secure methods of key management are important to a secured mobile payment system. When the key is randomly generated the system must prevent impersonation. In practice most attacks on public key systems will be aimed at the key management, rather than at the cryptographic algorithm(s)) [13]. In our architecture the two key pairs required are used both for the digital signature and in encryption. Figure 3 illustrates our proposed key management strategy for the digital signature.

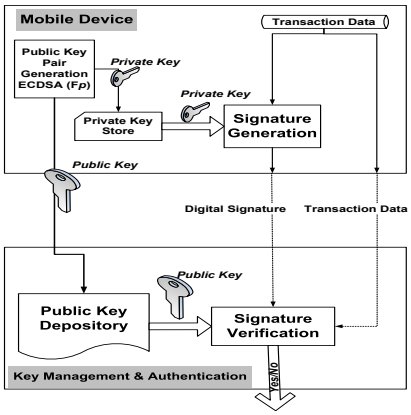


Figure 3. Digital Signature Key management for SA2pMP.

Digital Signature Key Management We propose the use of a digital signature to ensure non-repudiation during the transactions. Public key cryptography does not share any secrets between the two parties involved. A private key contributes to generating the digital signature on one side, and on the other a public key is used to verify the digital signature. To keep the private key secret we propose to generate the key pair in the mobile device. When a user starts to use the mobile payment application no key exists yet in the system, and a Key Generation function is called to generate a

key pair before processing the transaction. The keys must then be distributed and stored.

The private key is stored in the mobile device, either using File Stored in JAR or Record Stored in RMS. File stored in JAR refers to storage of the private key in the same jar package as the application program. This would rely on the fact that in the process of compilation of the Java application, class files in machine language are generated, and this process would obscure any details of the private key incorporated into the JAR. Record Stored in RMS (Record Management System)refers to the use of a subsystem of the MIDP in the Java ME standard [6]. The RMS APIs provide the ability to manipulate records in a record store and share records within an application, but sharing of records between different applications is prohibited (in MIDP2.0).

The public key is transferred to the authentication server in the banking server and then stored in a public key depository. Security on the server side is not within the scope of this proposal, and so we do not propose any details for this depository.

This will only take place once unless a renewed key pair is requested. This process can run off-line, not requiring any communications, and not competing with other transaction processing for computational resources. A key pair will eventually expire, and renewal of a key pair must initiated by the banking server. Once the server detects that renewal is needed, a notice (such as SMS) must be sent to the mobile device to generate a new key pair.

Encryption Key Management Generation of the encryption key takes place on the server. Encryption and decryption share the same secret key, which clearly cannot be transferred over the open wireless network due to the risk of interception. We propose that this key be stored in the program application jar package; then users would be issued the key along with the application package when they register for mobile banking services with their bank. On the server side we assume that the bank's security measures are sufficient.

5. Implementation

For space reasons, full implementation details cannot be included in this paper. Figures 4, 5(A) and 5(B) illustrate the proposed designs for the banking module, client architecture and server architecture for SA2pMP.

5.1. SA2pMP Mobile Client Architecture

The client portion of the proposed SA2pMP is intended to be built on Java ME enabled mobile devices. Figure 5(A) illustrates the architecture of the mobile client. The mobile client system is composed of the four modules described below.

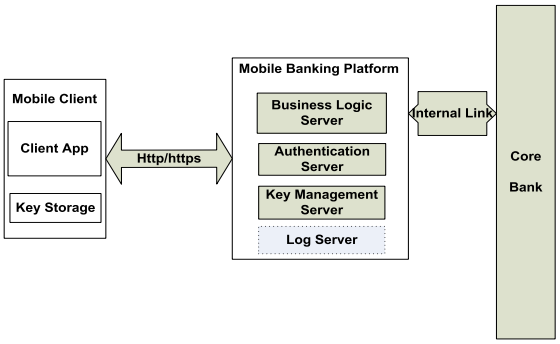
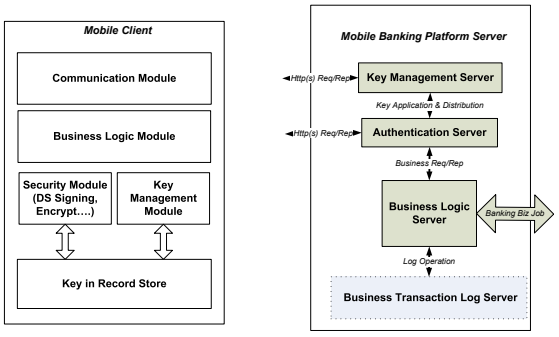


Figure 4. Mobile banking module design for SA2pMP.



(A) (B)

Figure 5. (A) Mobile client architecture for SA2pMP. (B) Mobile banking server architecture for SA2pMP.

Business Logic Module The BLM is in charge of all particular business functions between bank and user. These business functions will generated different entities, and the user can choose individual entities' composition according to his/her individual requirements.

Security Module The SM is to be responsible for security issues. After the user's request has been processed by the BLM, an information message is generated. This message will be processed with digital signature and encryption in the SM. Following the rules of ECDSA we read the private key from the KLM, and then affix the digital signature to the original message, along with the encryption. The encrypted message is the communication entity which will be sent by the CM to the mobile banking server.

Communication Module The CM is the module in charge of the network link. We propose the use of HTTP. The CM handles information exchange between the mobile client application and the mobile banking server.

Key Management Module The KMM is in charge of key management as described in Section 4.3.

5.2 Server Architecture

The mobile banking platform server consists of the components described in the following paragraphs. Figure 5B) illustrates the proposed architecture for the server platform.

Key Management Server KMS deals with receipt and storage of the public key, as well as initiating messages notifying the client the renew the key pair currently in use. Details of this are given in Section 4.3.

Authentication Server The AS provides authentication service for the mobile banking platform. The following security items should be implemented in the AS:

1. checking for userid/password legality during log on,
2. verification of the digital signature, and
3. decryption of message(s).

The message which has passed through AS is a legal business message and so it is next interpreted as to the original business request.

Business Logic Server The BLS handles all legal business requires. The BLS interacts with the regular bank, processing business logic and exchanging business information. When BLS finishes processing a business job it responds by sending the result information to the user's mobile device.

Transaction Log Server The TLS creates and maintains the log files for transactions taking place in the mobile banking platform. The LS is a security mechanism to protect the bank from repudiation between users and the bank. Once users could deny having taken actions in a mobile banking platform, so the LS prevents this by providing the transaction record as proof.

6. Conclusion

We have proposed a secure architecture for two-party mobile payments. While other architectures and protocols have been proposed, they either are not well suited for mobile (and thus resource-constrained) devices or they do not satisfy all of the parties' concerns regarding security in carrying out mobile transactions. Our architecture, referred to as SA2pMP, is intended to be implemented in Java ME on a mobile client, with a mobile banking server supporting it, likely implemented in Java EE. The proposed architecture employs a digital signature and a transaction log strategy to meet the security requirement of non-repudiation, and its lightweight client design is intended to be well-suited to mobile devices.

Work is continuing on implementation, with a goal of simulating mobile transactions and comparing the results to existing mobile payment systems.

References

[1] M. Bellare, J. A. Garay, R. Hauser, A. Herzberg, H. Krawczyk, M. Steiner, G. Tsudik, E. V. Herreweghen, and M. Waidner. Design, implementation, and deployment of the iKP secure electronic payment system. *IEEE Journal on Selected Areas in Communications*, 18(4):611–627, April 2000.

[2] P. C. Deans. *E-Commerce and M-Commerce Technologies*. IRM Press, 2004.

[3] Federal Financial Institutions Examination Council. *Authentication in an Internet Banking Environment*. http://www.ffiec.gov/pdf/authentication_guidance.pdf.

[4] A. Fernandes. Elliptic curve cryptography. *Dr. Dobb's Journal*, December 1999.

[5] J. Gao, J. Cai, K. Patel, and S. Shim. A wireless payment system. In *Proceedings of the 2nd International Conference on Embedded Software and Systems*, volume 16-18, page 8, 2005.

[6] E. Giguere. *Databases and MIDP, Part 1: Understanding the Record Management System*. iAnywhere Solutions, February 2004.

[7] W. Itani and A. Kayssi. J2ME application-layer end-to-end security for m-commerce. *Journal of Network and Computer Applications*, 27:13–32, 2004.

[8] P. Janson. Internet keyed payment protocols (ikp). Technical report, IBM Zrich Information Technology Solutions, 2007.

[9] K.-Y. Lam, S.-L. Chung, M. Gu, and J.-G. Sun. Lightweight security for mobile commerce transactions. *Computer Communications*, 26:2052–2060, 2003.

[10] E. R. Lenstra, A. K. Verheul. Selecting cryptographic key sizes. *Lecture Notes in Computer Science*, 1751:446–465, 1999.

[11] K. Linck, K. Pousttchi, and D. G. Wiedemann. Security issues in mobile payment from the customer viewpoint. In *Proceedings of the 14th European Conference on Information Systems (ECIS 2006)*, Ljungberg, 2006.

[12] E. Maiwarld. *Fundamentals of Network Security*. McGraw-Hill Professional, 2004.

[13] RSA Laboratories. *RSA Laboratories' Frequently Asked Questions About Today's Cryptography, Version 4.1*. RSA Security Inc., 2000.

[14] W. Stallings. *Cryptography and Network Security: Principles and Practice*. Pearson Prentice Hall, 3rd edition, 2006.

[15] Sun Microsystems. *JSR 177: Security and Trust Services API for J2METM*.

[16] A. S. Tanenbaum. *Computer Networks*. Prentice Hall, 1996.

[17] U. Varshney and R. Vetter. Emerging mobile and wireless networks. *Communications of the ACM*, 43(6):73–81, 2000.

[18] VISA & Mastercard. *SET Secure Electronic Transaction Specification*, 1997.

[19] X. Zhong, D. Guanzhong, and Y. Deming. An efficient ECDSA-based signature scheme for wireless networks. *Wuhan University Journal of Natural Sciences*, 11(6):1707 – 1710, November 2006.