A mobile local payment system Bluetooth based

Gianluigi Me, Alex Schuster

Dipartimento di Informatica, Sistemi e Produzione,Università di "Tor Vergata", Roma, 00183, Italy

Fakultät Informatik, Technische Universität München, Germany

***Abstract* — M-Payments mean that the mobile device, becoming a personal trust device (PTD), can be used as a payment instrument in the physical world. A payment capability is embedded in the mobile device, extending different payment methods to the mobile PDA with care of ease of use and overall convenience. The main issue is strict security requirement, which include customer transaction authentication, confidentiality, integrity, non repudiation in an environment composed of heterogeneous wireless networks, such as PAN, LAN, WAN, with different security weaknesses. This paper describes a secure macropayment system prototype, developed with open source and free tools, hosted by a PDA that obtains a virtual cheque from a bank off line or via GPRS and then pays a good/service by transmitting the cheque to a vendor via Bluetooth, with (user option) various levels of interaction in the shop. With appropriate cost analysis on convenience, this prototype can be easily downsized to micropayments systems.**

***Index Terms* — Mobile Networks, E-payments, Ellipitc curce cryptography, Java.**

I. Introduction

The wireless technology Bluetooth and its widespread diffusion on mobile devices (mobile phone, PDA, Smartphone) enables new payment paradigms, called local payment, allowing devices to connect easily to embedded machines and perform transactions [1]. Thus arises the customer need and desire for a wireless payment ([2], [3], [4], [5]) system relying on Bluetooth. This paper aims to propose a secure, device independent, user friendly wireless macropayment system (payments dealing with amounts greater then 10$ / 10 €), based on e-cheque and implemented on a PDA. Since money is transferred, the communication channels have to be secure, with care of well known vulnerabilities [6], [7] of media (Bluetooth, GPRS), of communication layer (e.g. GEA, E3) or new, unpredictable vulnerabilities (e.g. WTLS gap, even if it has been solved in WAP 2.0) of transport layer security. For this reason, the requirement of macropayment system security is met at the application layer [8]. It is accomplished by data encryption and mutual authentication which crosses the anonymity of the user, thus money transfer can be traced to prevent illegal transfer. If needed the system can be downsized in an anonymous one by replacing the mutual authentication with a shop-side one and adding a blind signature [9] to the cheque to avoid its traceability.

II. Architecture

As shown in Fig.1a, where we present the general payment transaction, the PDA runs as client and the bank (for the sake of simplicity, in this prototype we considered the merchant bank as the customer bank) and the shop run as servers.

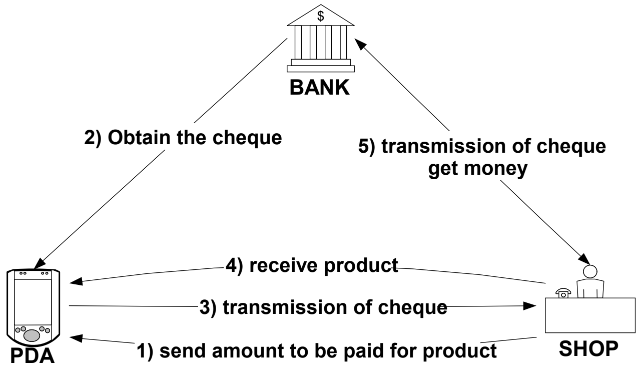

BANK
$
2) Obtain the cheque
5) transmission of cheque get money
4) receive product
3) transmission of cheque
1) send amount to be paid for product
PDA
SHOP

Fig. 1. General e-cheque transaction scheme

We divide the whole transaction in three communication phases: 1) PDA – Shop (via Bluetooth), 2) PDA – Bank (via GPRS) and 3) Shop – Bank (out of the scope of this paper). The PDA-user gets the bill from the shop, downloads a cheque with the corresponding amount of money from the bank and pays by sending that cheque to the shop. This verifies (referring to the same trusted CA) and accepts the cheque. Then the shop transmits the cheque to the bank to complete the transaction. Both PDA-shop and PDA-bank encrypted data exchanges are performed after authentication and session key arrangement (Fig. 2).A signed blank cheque can be used instead of downloading one immediately before payment thus avoiding dependencies on the GPRS net. This blank cheque can be downloaded and saved, completing it when needed with the amount to pay before transmitting it.

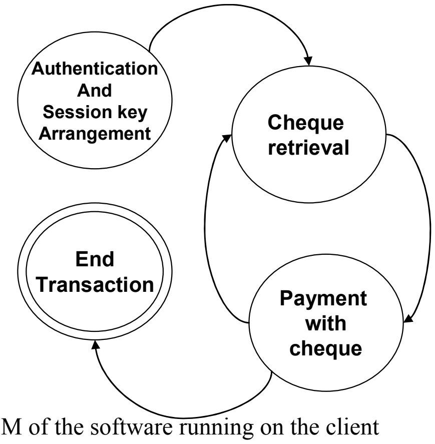


Fig. 2. FSM of the software running on the client

A. *The transaction flow*

At the beginning of each communication a mutual authentication is performed between bank/PDA and shop/PDA: this is arranged by certificates that are signed by a recognized CA (Certification Authority). Therefore each actor (bank, shop, user) is obliged to possess a certificate. Initially the certificates are exchanged and verified, thus guaranteeing the identity of the communication partners and the correct public keys. Then each instance has to attest its proper possession of the corresponding private key with a Challenge/Response procedure. Once the authentication phase succeeds, each actor sends a random number, encrypted with public keys, to the communication partner. They are used to derive a symmetric session key with which the further communication is encrypted.

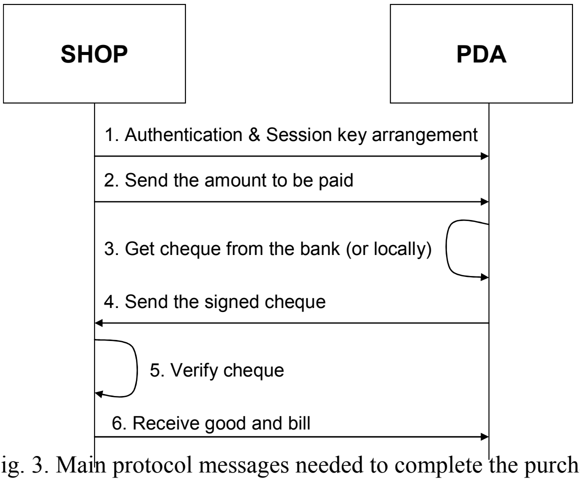

SHOP
PDA
1. Authentication & Session key arrangement
2. Send the amount to be paid
3. Get cheque from the bank (or locally)
4. Send the signed cheque
5. Verify cheque
6. Receive good and bill

Fig. 3. Main protocol messages needed to complete the purchase.

The e-cheque consists of the following 4-tuple (only main information represented here): value of money of the cheque, serial number to identify it, current date and the identity of the user. After the transmission of the required value from shop to PDA (2) and from PDA to bank (3), the bank signs the cheque with its private key and sends it to the PDA (3). This one transfers the cheque without user interaction to the shop (4) which verifies the signature of the bank and the amount of money (5). Finally if each step is passed successfully the user receives a confirmation (6) and the shop can transmit the cheque to the bank to receive the money (if required, the bank can check the cheque validity online, via the proprietary bank network).

B. *Requirements*

To be successful in gaining customer adoption, following payment requirements have to be met:

- *Secure*: Buyers have, and will continue to have, concerns about Internet and wireless transaction security;
- *Convenient and Easy to Use*: New methods must be more convenient than using cash or cards;
- *Universally Available*: User adoption requires that the method can be used across a large range of merchant locations — perhaps the most difficult barrier to new payment methods.

Design main focus has been to shift as much computational effort as possible from a mobile node to the other entities which reside in the wired network, because it is assumed that the mobile node has limited computational power.

We concentrate our proposal on following additional requirements: traceability of payments (to prevent illegal money transfer, [10]) and easily and reusable deployment. On the other hand, a constraint of this system is strictly linked to mobile device computational resources performances managing asymmetric cryptography: due to short message sizes, performance are acceptable on test PDA equipped with old Intel ARM SA 1100, posing great expectations on application performance running on XScale platforms.

III. SYSTEM

A. *Adopted encryption schemas*

The choice of the asymmetric encryption system for the mutual authentication fell on ECIES (Elliptic Curve Integrated Encryption Scheme) with a 192 bit Elliptic Curves (EC). It combines the EC asymmetric encryption with the Advanced Encryption Standard (AES, 128 bit key) and adds a SHA-1 hash for message authentication. In comparison to a 1024 bit RSA key, ECC (Elliptic Curve Cryptography) provides shorter keys, shorter encrypted messages and faster private key operations. As the message has to be split up for encryption in pieces smaller than the asymmetric key, a mechanism that links each piece to the preceding ones is introduced for more

Top

security. This chaining mechanism is called Cipher Block Chaining (CBC) [11]. To sign the transmitted data, and thus avoid its unnoticed modification, the system relies on the Elliptic Curve Digital Signature Algorithm (ECDSA) that adds a 48 byte signature value [12]. A PKCS7 Padding [13] is performed to increase the security by keeping the encrypted message always at a certain length. The certificates are realized as X.509 Certificates that are signed by a 1024 bit RSA public key of a CA and contain its own validity, the identity of the owner and his EC public key.

The symmetric key is derived from two random numbers R by performing a HMAC-SHA256(R_1,R_2) hash function. AES with CBC, PKCS7 and a SHA-1 hash, for message integrity, are performed for symmetric encryption.

Due to the target of this paper, there's not enough room to address a detailed security analysis, which can be found in [14].

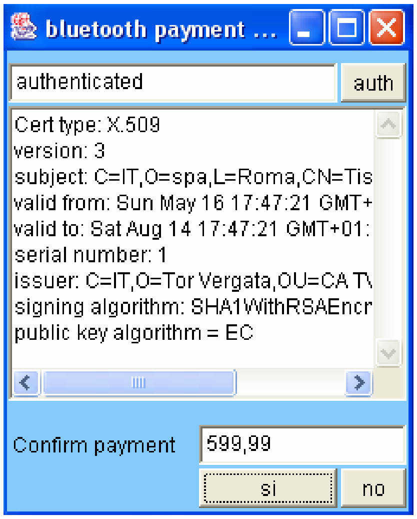
B. *Implementation*

We built the client using a PocketPC 2002 PDA equipped with Bluetooth and Jeode, a Java VM for PocketPC that offers functionality similar to the JDK 1.1.8. The bank and the shop were simulated by a PC equipped with a PCMCIA Bluetooth card. As a development IDE for PC we choose the open source software Eclipse 3.0 in combination with the JDK 1.1.8 to compile the three parts of the system (PDA, shop, bank). Then the compiled PDA java program was transferred to the PDA via Microsoft ActiveSync. As the Sun's JCE (Java Cryptography Extension) didn't satisfy our need for cryptographic algorithms it was replaced by the BouncyCastle JCE that is free for download at [15]. By installing the drivers for a Bluetooth device on a computer, also the corresponding virtual serial ports for this device, i.e. COM6, are installed. With these virtual serial ports it is possible to address and control the Bluetooth device by using normal serial port commands. Therefore Sun offers the Java Comm API [16] for the PC, while for the PDA a special modified Java Comm API is required [17]. In this way the Bluetooth connection between the shop and the PDA is established over their virtual serial ports. The system is implemented with an EC – RSA mixture as asymmetric encryption components. RSA is used to sign a X.509 certificate by a CA, as this solution occupies less time to verify the signature of the CA by the PDA. During authentication, ECC is used to provide data encryption and integrity.

C. *Use Case*

The PDA begins the mutual authentication by pressing the "auth" button in the upper left of the application. The status field changes to "authentication in progress" until the authentication, which begins with the transmission of the certificates, is finished. When the partner's certificate arrives it is verified, the public key is extracted and then it is shown in the application window.

The mutual authentication proceeds as explained in paragraph 2.1. Once completed, the status field changes to "authenticated", as depicted for both actors of payment, with the snapshot in fig. 4 and fig. 5 .



authenticated

```
Cert type: X.509
version: 3
subject: C=IT,O=spa,L=Roma,CN=Tis
valid from: Sun May 16 17:47:21 GMT+
valid to: Sat Aug 14 17:47:21 GMT+01:
serial number: 1
issuer: C=IT,O=Tor Vergata,OU=CA T\
signing algorithm: SHA1WithRSAEncr
public key algorithm = EC
```

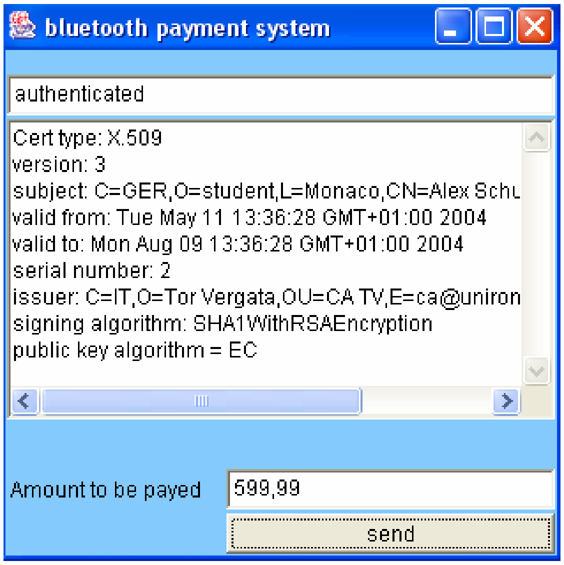
Confirm payment 599,99

Fig. 4.:The snapshot on PDA side when authentication succeeds

Now the shop enters the money for the desired good in the field "Amount to be paid" and sends it via Bluetooth to the PDA. When it appears in the field "Confirm payment", if the "si" button is pressed, the PDA opens a connection to the bank and downloads the cheque with the appropriate value. Since has been received by the PDA, it is decrypted, encrypted and then sent to the shop without interaction of the client. The transaction can be performed even with blank cheques previously stored in PDA, so avoiding transaction failures in case of shop 2/2,5/3 shadowed shops.

Once the cheque arrives at the shop its value and the signature of the bank are checked. If successful, the status field of the shop changes to "cheque verified successfully" and the payment is acknowledged to the PDA. This finishes user side transaction.

Top



authenticated

Cert type: X.509
version: 3
subject: C=GER,O=student,L=Monaco,CN=Alex Schu
valid from: Tue May 11 13:36:28 GMT+01:00 2004
valid to: Mon Aug 09 13:36:28 GMT+01:00 2004
serial number: 2
issuer: C=IT,O=Tor Vergata,OU=CA TV,E=ca@uniron
signing algorithm: SHA1WithRSAEncryption
public key algorithm = EC

Amount to be payed: 599,99

Fig. 5. The snapshot on embedded appliance (Virtual POS) side when authentication succeeds

D. Performance

The duration of the payment transaction depends strongly on the authentication phase and in this way on the performance of the asymmetric encryption. One mutual authentication occupies 38 seconds and ca. 95 % of this time is used to encrypt, decrypt, sign and verify signatures on the PDA. This performance is reached by pre-computing of the random value R2 on the PDA to reduce processing time during authentication.

The other 5% of the time is used for sending the data or executing the mentioned operations on the bank/shop, simulated by a Pentium 4 Notebook. Once the authentication has succeeded, the remaining transaction steps (generation of cheque, transmission of amount, transfer of data and verification of the cheque) don't require noticeable time. The symmetric cryptographic operations (AES) can also be neglected. Altogether the system requires ca. 80 seconds of computational effort consisting of two authentications (2*38secs) and estimated 4 seconds for the rest. If a blank cheque is used, the connection to the bank isn't required anymore and thus only one authentication is needed. In this way the transaction time is reduced drastically to ca. 40 seconds. A further reduction of time could be achieved by replacing the PDA, running with a 206 MHz StrongArm processor, with a newer one equipped with an XScale processor. This 400 MHz CPU promises an increase of at least 50% for CPU intensive operations like our encryption system. The XScale processor however requires an OS like PocketPC 2003 or later that supports the new XScale instructions. Newer OS than the used PocketPC 2002 also provide a better performance themselves. At last a newer and faster Java VM than the Joede JM contribute a decisive improvement of speed.

IV. Security Considerations and Concluding Remarks

Despite the wide plethora of existing payments standards, in this paper we presented a novel mobile macropayment system cheque-based with respect of the following requirements: security, user friendliness, traceable payment (to prevent illegal money transfer), easily and reusable deployment. Considering the fact that the more the complexity of the systems raises the more vulnerability risks exist, a secure, easy-to-use and easily manageable payment application has been needed to face user lack of trust in payment systems. Future works will be focused on security stressing of adopted protocols, testing performance on XScale platforms and different mobile devices (e.g. mobile phones) and to collect potential user feedback to adapt the client on his needs.

References

[1] G.Me, Security overview for m-payed virtual ticketing, Proceedings of 14 IEEE PIMRC, 2003, Pg.844-848

[2] 3-D Secure ™Mobile Authentication Scenarios, VISA,Int'l,http://international.visa.com/fb/paytech/secure/main.jsp

[3] http://www.openmobilealliance.org

[4] Liisa Kanniainen, The Preferred Payment Architecture: Technical Documentation, Mobey Forum, http://www.mobeyforum.org

[5] MeT, MeT Overview White Paper Version 2.0, www.mobiletransaction.org, 2001

[6] Jacobson, Markus Wetzel, Security weaknesses in Bluetooth, http://www.belllabs.com/user/markusj/bluetooth.pdf

[7] R. K. Nichols, P. C. Lekkas," Wireless Security: Model, Threats and Solutions", McGraw-Hill TELECOM, 2001, pp. 402–415.

[8] Michael Peirce, "Multi-Party Electronic Payments for Mobile Communications", http://citeseer.nj.nec.com/peirce00multiparty.html

[9] CiteSeer.IST, blind signature from Chaum, http://citeseer.ist.psu.edu/context/36440/0

[10] P.Bellamare and J.Antoine, Dossiers d'Interpol 2, Editions N1, 1976, pp.301-308

[11] Cipher Block Chaining http://encyclopedia.thefreedictionary.com/Cipher%20Block%20Chaining

[12] Serge Vaudenay, The Security of DSA and ECDSA, http://grouper.ieee.org/groups/1363/WorkingGroup/presentations/ecdsa_pkc03.pdf

[13] RSA Data Security Inc., PKCS#7 Cryptographic Message Syntax Standard, http://www.zone-h.org/files/33/pkcs-7.pdf

Top

[14] G. Me, M.A. Strangio, EC-PAY: An Efficient and Secure ECC-based Wireless Local Payment Scheme, Proceedings of 3 rd IEEE/ICITA 2005, to appear;

[15] The Legion of the Bouncy Castle, free Java Cryptography Extension, http://www.bouncycastle.org :

[16] Sun Microsystems, Java Communications API http://java.sun.com/products/javacomm/index.jsp

[17] RXTX, Java Communications API for PDA http://www.rxtx.org

Top