2006 Conference on Parallel and Distributed Computing, Applications and Technologies, Taipei, Taiwan, December 4-7, 2006, © IEEE 2006

Off-line Micro-payment Protocol for Multiple Vendors in Mobile Commerce

Xiaoling Dai[1] Oluwatomi Ayoade[1] and John Grundy[2, 3]

School of Computing Information and Mathematics Science
The University of the South Pacific, Laucala Campus, Suva, Fiji[1]
dai_s@usp.ac.fj
s11035778@student.usp.ac.fj

Department of Electrical and Computer Engineering[2] *and Department of Computer Science*[3]
University of Auckland, Private Bag 92019, Auckland, New Zealand
john-g@cs.auckland.ac.nz

Abstract

Micro-payment systems have the potential to provide non-intrusive, high-volume and low-cost pay-as-you-use services for a wide variety of web-based applications. Previously we have developed NetPay, an off-line micro-payment protocol for E-commerce. In this paper we describe a set of extensions, Mobile-NetPay, optimised for mobile E-commerce. Mobile-NetPay provides high performance and security using one-way hashing functions for e-coin encryption. Each mobile user's transaction does not involve any broker and double spending is detected during the redeeming transaction. We describe the motivation for Mobile-NetPay and describe three transactions of the Mobile-NetPay protocol in detail to illustrate the approach. We then discuss future research on this protocol.

1. Introduction

In the next few years, the market for very low value, very high frequency purchase items such as web pages, ring tones or multimedia clips, online content (such as music and videos), is expected to grow substantially [8, 11]. In recent times, mobile phones have become platforms of choice for delivering rich digital data, used for recording and downloading photos, video and music, Internet access, Podcasts, and transmitting payments.

Mobile E-commerce ("m-commerce") involves the usage of the Internet for purchasing goods and services and also for transmitting messages using wireless mobile devices. Mobile computing enables internet-enabled cell phones, PDAs, and other wireless computing devices to access digital information on the Internet from any location and at anytime. This is otherwise known as **wireless e-commerce**. There are many wireless e-commerce scenarios in existence and the potential role of using wireless devices to access the web is enormous. However, there is no commerce without payment and the key to innovation in wireless services is payment [9]. Therefore, wireless e-commerce demands an appropriate payment method and in order to allow "pay-per-use" of such content, micro-payment systems (payment for high volume, low value transactions) are expected to play an important role.

With the advent of wireless communications technology, the two essential elements (mobility and accessibility) need to be enhanced for existing micro-payment approaches. Also, with the proliferation of wireless networks, mobile devices and customers' increasing desire for more purchasing power and convenience, demand for micro-payment schemes is expected to rise [10, 13].

This paper focuses on the use of a micro-payment scheme as a means of payment for wireless e-commerce. We have been investigating approaches to apply our earlier NetPay micro-payment protocol for the mobile information content applications. This is achieved by providing phone and PDA-hosted micro-payment applications with a client side e-wallet storage on the mobile device. A Mobile-NetPay-enabled application needs to provide HTML (web browser) and Wireless Markup Language (mobile) user interfaces and support a wide range of diverse input devices. This paper presents the main functional characteristics of our new proposition of wireless micro-payment systems for multiple vendors that provide more flexible, mobile and accessible mobile micro-payment solutions.

In this paper, we briefly describe a protocol [8] and the NetPay micro-payment protocol with the client-side e-wallet in the client-server networks. We then proposal an off-line micro-payment protocol called Mobile-NetPay to provide high performance and security in wireless as opposed to a wired network. We conclude with an outline of our further plans for research and development in this area.

2. Motivation

With the growth of mobile computing technologies, the popularity of mobile devices such as mobile phone, PDAs has increased over the past few years. A wide range of software applications can be deployed on these mobile terminals and can communicate with other applications or information systems through a wireless network. Consider a mobile device user carrying out the following tasks using a mobile device: (1) Reading both free and pay-per-click web sites on the device browser (2) Purchasing images, videos, music clips and ring-tones (3) Making available pictures and videos for others to use and possibly purchase (4) Accessing various information sources for weather, shopping, tourism etc.

There are many types of m-commerce payment such as the use of operator billing, digital wallets (E-Cash) Techniques, credit card and direct Payments. However, we will focus more on the digital wallets (e-cash) technique as we are investigating approaches of using NetPay for information content micro-payment application with a client side "e-wallet" storage by the mobile device. For the purpose of this research, we will consider transaction payment for mobile information content such as ringtones, music, video, games and wallpapers. The e-wallet stored by the mobile device will involve the use of payment tokens in the form of "e-coins" for payment of low valued items. We will take the two major key issues involve in micro-payment into consideration for designing our new framework. These are ***low value (use of hash function to reduce cost overhead)*** and ***high volume.*** Therefore, the ability to buy inexpensive items conveniently (at anywhere and at anytime) would eliminate the need for buyers to pay large subscription fees for the entire sets of material when they only want selected pieces of content [11].

There are a number of micro-payment systems for wired client-server networks in various stages of development from proposals in the academic literature to systems in commercial use [1, 7]. Micro-payment systems can be used to support payment of vendors from customers in client-server networks. In a mobile communication system, the low computing power of mobile devices and a lower bandwidth and higher channel error rate than wired networks should be considered to design a micro-payment system. Zhu protocol is an example off-line payword-based micro-payment system designed for wireless networks [8].

3. A micro-payment protocol in mobile commerce

The micro-payment system was proposed by Zhu, et al. [8] called Zhu micro-payment protocol. In Zhu protocol, a mobile user (MU) attaches to the network through an access network operator (NO). The connection may pass through one or more other network operators before reaching the destination vendor. The MUs generate their own "coins," or paywords, which are sent to NOs and vendors and then verified by brokers. Fig. 1. shows key Zhu protocol interactions.

1. PayWord chain commitment request: In the beginning of the transaction, MU generates hash chain by applying one-way hash to root Wn. MU randomly picks a payword seed Wn and then computes a payword chain by repeatedly hashing Wn:, Wi-1 = h(Wi) where i=1,2,..,n. MU then sends M1 = {macro-payment, W0, length n, desired total value of the payword chain, ID of his NO} to a broker. M1 is digitally signed by broker's public key. Broker commits the payword chain by digitally signing M2 = Comm-w = {W0, n, chain_value, NO, expiry} and sends it to MU. Comm.-w ensures SPs that the MU's paywords are redeemable by the broker

2. Pricing Contract: To get some services from a Vendor the MU sends M3 = the request details = {Vendor, service type, Quality of service (QoS) requirement, Comm-w} to the NO first. The NO generates an endorsement hash chain commitment (Comm-E) for each MU. Comm-E is signed by the NO and sent to the vendor. The vendor generate signed Pricing Contract (PCv) in order to allow verifiable dynamic tariffs; fix the starting hash; decide the value per payment hash and create a record for MU; and link a single payment commitment to multiple vendors for MU. PCv = {Transaction ID, ID of NO and V, Charge, Comm-w, Wstart, Strat, W_value, Comm-E, Broker} is signed by the vendor and agreed by the MU.

***3. Payment Processes**:* To make m cents payment, the MU releases W1 through Wm where m is

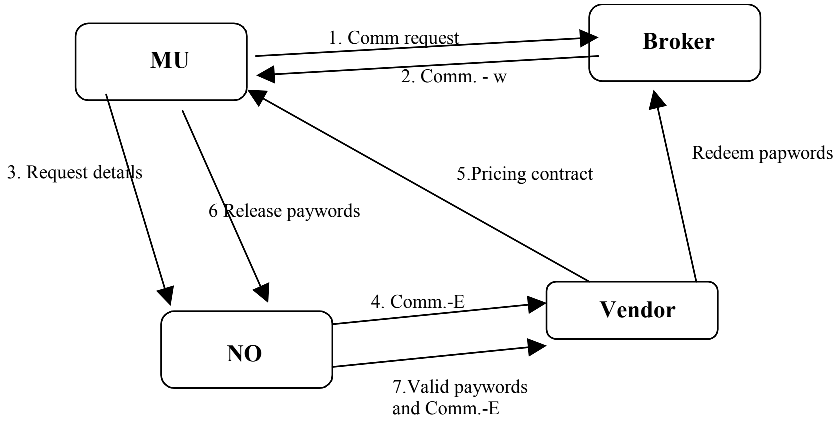

Fig. 1. Zhu protocol participant interactions [based on 8]

the number of the paywords the UM wish to spend and the requirement of the information goods to the NO. The NO can easily verify the paywords by hashing Wm m times until he reaches W0 and forward the paywords and his endorsement hash E1 through Em to the vendor. The vendor verifies both the paywords and the endorsement hash. The UM could download the **downloadable media** from the vendor.

4. Redeeming: At the end of each day, the vendor sends the highest payword spent, a corresponding endorsement hash and pricing contract to the broker. The broker verifies the paywords using the root w_0 and knows how much to pay the vendor from the contents of the pricing contract.

Zhu protocol is an off-line system. The MU only needs to contact the broker at the beginning of each commitment lifetime in order to obtain a new-signed commitment. The system aims to minimize the computation time of public key operations required per payment using hash operations instead whenever possible. Every payment needs to be authorized by the NO in order to prevent double spending from MU. The NO also needs to generate a corresponding endorsement hash and sends them to the vendor in every payment. The e-coin (paywords) in the system is MU and vendor specific and the paywords in the chain have no value to another vendor.

4. NetPay in client-server networks

We developed a protocol called NetPay that provides a secure, cheap, widely available, and debit-based protocol for an off-line micro-payment system [1]. We have developed NetPay-based systems for client-server broker, vendor and customer networks [3], [4]. We have also designed three kinds of "e-wallets" to manage e-coins in our client-server NetPay systems [3, 4, 5]. In one model the E-wallet is hosted by vendor servers and is passed from vendor to vendor as the customer moves from one site to another. The second is a client-side application resident on the client's PC. The third is a hybrid that caches E-coins in a web browser cookie for debiting as the customer spends at a site.

The client-side e-wallet is an application running on the client PC that holds e-coin information. Customers can buy article content using the client-side e-wallet at different sites without the need to log in after the e-wallet application is downloaded to their PC. Their e-coins are resident on their own PC and so access to them is never lost due to network outages to one vendor. The e-coin debiting time is slower for a client-side e-wallet than the server-side e-wallet due to the extra communication between vendor application server and customer PC's e-wallet application. In a client-side e-wallet NetPay system, a Touchstone and an Index (T&I) of a customer's e-wallet are passed from the broker to each vendor. We designed that the broker application server communicates with vendor application servers to get the T&I to verify e-coins. The vendor application servers also communicate with another vendor application server to pass the T&I, without use of the broker. The main problem with this approach is that a vendor system cannot get the T&I if a previous vendor system goes down.

5. Mobile-NetPay protocol in mobile commerce

Based on the client-side e-wallet NetPay protocol, we have developed an adaptation to a Mobile-NeyPay protocol that is suitable for wireless network environments. Our Mobile-NetPay protocol uses touchstones that are signed by the broker and an e-coin index signed by vendors. The signed touchstone is used by a vendor to verify the electronic currency – paywords, and signed Index is used to prevent double spending from a Mobile User (MU) and to resolve disputes between vendors. In this section, we describe the key transactions in Mobile-NetPay protocol in wireless networks.

In this section, the details of a mobile micro-payment NetPay model are discussed. Consider a trading community consisting of a mobile user (MU), a network operator (NO), vendors (Vs), and Broker (B). Assume that the broker is honest and is trusted by the NOs, Vs and MUs. The MUs and Vs may be or may not be honest. The MUs open accounts and deposit funds with the broker. The payment involves Vs, NOs, MUs and Broker. Broker is responsible for the registration of MUs and for crediting the V's account and debiting the MU's account. In a Mobile-NetPay system, there are three transactions which are MU-broker, MU – Vendors, and Vendors - Broker transactions. How the Mobile-NetPay protocol works in each transaction will now be described in more detail. We adopt the following notations:

IDa --- pseudonymous identity of any party A in the trade community issued by the broker

PK-a --- A's public key

SK-a --- A's digital signature

{x}SK-a --- x signed by A

{x}PK-a --- x is encrypted by A's public key

{x}SAK- a --- x signed by A using A's asymmetric key

5.1 Transaction 1: Mobile User – Broker

Before a Mobile User (MU) asks for service from the Vendor1 (V1) she has to register and send an integer n (M1), the number of paywords in a payword chain the MU applied for, to the broker (Fig. 2). The broker completes two actions:

- Debits money from the account of MU and creates a payword chain W_0, W_1, W_2,…,W_n, W_{n+1} which satisfies $W_i = h(W_{i+1})$, where i = n, …, 0. (here h(.) is a one way hash function). Root W_0 is used to verify the validity of the paywords W_1, W_2, …, W_n by peers and the broker. Seed W_{n+1} is kept by the broker to be used to prevent the peer1 from overspending and forging paywords in that chain. The peers only receive ID_e (e-coin ID) and paywords W_1, W_2,…,W_n that are encrypted by MU's public key from the broker (M2) as shown in Fig. 2. **M2** = $\{ ID_e, W_1, W_2, \dots ,W_n \}_{PK\text{-}MU}$ The broker computes the touchstone for the payword chain. T = $\{IDe, W_0\}_{SK\text{-}broker}$
- Save IDe, W_0, W_{n+1}, and amount to the broker database.

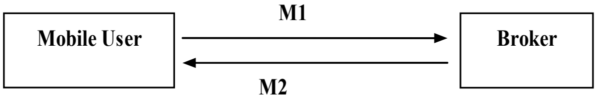

Fig. 2. Mobile User buys e-coins transaction

For example, the MU sends n=50 to the broker who generates the IDe=1 and payword chain $\{W_0, W_1, W_2, \dots ,W_{50}, W_{51}\}$. The MU's e-wallet which resides on the MU's mobile phone thus contains $\{$IDe, $W_1, W_2, \dots ,W_{50}\}$ and T. The broker saves IDe, W_0, W_{n+1}, and 50 to its database.

The MU - broker transaction guarantees no overspending and forging. The broker selects the seed W_{n+1} to create the payword chain which satisfy $W_n = h(W_{n+1})$, $W_{n-1} = h(W_n)$, …, $W_1 = h(W_2)$, $W_0 = h(W_1)$ and keep the seed W_{n+1} secretly. It is impossible to forge the paywords in that chain by MUs and attackers, since they do not have the seed W_{n+1}, i.e. it is impossible to generate other paywords in a chain by knowing some of them in the chain since h() is a truly one-way hash function [12].

5.2 Transaction2: mobile user - vendor

The following sequence of messages describes a transaction between a MU and Vendors (V_1, V_2) in the course of a downloadable media from vendors to MU. The MU knows the price of a downloadable media from vendors' site.

When a MU attempts to purchase downloadable media from V_1, V_1 sends a host and port (M3) to the MU's e-wallet. The e-wallet compares the host and port in M3 with the previous host and port. If different, the e-wallet sends a message M4 back to V_1.

M4 = { IDe, paywords, B's host and port, identity of MU's NO } where paywords = $\{W_1, W_2, \dots, W_m\}$. For example, to make a 2cs (m=2) payment, the customer sends Paywords = $\{W_1, W_2\}$ to the V_1.

If the MU fist time makes a purchase with a vendor using the e-coin, V_1 sends the IDe (M5) to the broker for requesting the touchstone. The broker sends following transmission message:

Index = $\{IDB, 1\}_{SK\text{-}B}$ along with the payword chain touchstone T, and transmits them to V_1 (M6). The touchstone and Index authorise V_1 to verify

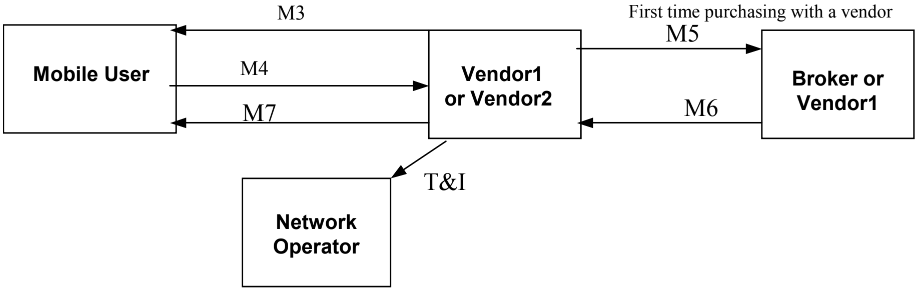

Fig. 3. Mobile User buys downloadable media transaction

the paywords using root W_0 and redeems the paywords with the broker as shown in Fig. 3.

The paywords are verified by taking the hash of the paywords in the order W_1 first, then W_2, and so on. It is hard for V_1 and attackers to create W_1 even though he knows W_0 since the generation of a value that would hash to W_0 is computationally infeasible due to the nature of the one-way hash function. If the paywords are valid, V_1 will be stored for a later redemption with the broker. The MU could download the downloadable media from V_1 (M7). The V_1 signs the current Index = $\{IDv_1, i\}_{SK\text{-}v1.}$ T and current Index are sent to NO by V_1 in order to prevent that a vendor cannot get the T&I if a previous vendor system down. The MU could continue to buy other downloadable media with the V_1.

When a MU wishes to make a purchase at a different vendor V_2, the vendor2 requests the current e-coin index and the touchstone from vendor1 to verify the e-coins when the MU changes purchasing from the vendor1 to a vendor2 as shown in Fig. 3. When the MU wishes to purchase information goods at V_2, V_2 sends a price, host and port (M3) to the e-wallet. The e-wallet compares the host and port in M3 with the previous host and port. If different, the e-wallet sends a message: **M4** = {IDe, paywords, V_1's host and port, identify of NO} to V_2. V_2 transmits the IDe (M5) to V_1 requiring touchstone and index.

The V_1 signs the following transmission message: **Index** = $\{IDv_1, i\}_{SK\text{-}v1}$ along with the payword chain touchstone, and transmits them to V2 (M6), where i is the index of the last payword V_1 received. The Index may be used for disputes between the vendors, and the touchstone is used to make future transactions with C and to redeem the paywords from the broker. After V_2 verifies the paywords using the touchstone and the index, the MU downloads the downloadable media form V_2 (M7). If V_1 system is down, V_2 sends message M5 to the NO and gets T and Index from the NO. The MU could continue to buy other downloadable media with the V_2. This transaction has two advantages: firstly, the transfer of the message M6 from V_1 to V_2 does not involve the broker, it reduces the communication burden of the broker; secondly, the message M6 includes the index of the paywords, it prevents the MU from double spending when the MU purchases from another vendor.

5.3 Transaction3: Vendor – Broker Offline Redeem Processing

At the end of each day (or other suitable period), for each payword chain, all vendors need to send all paywords that they received from MUs to the broker and redeem them for real money. To do this a vendor must aggregate the paywords by each e-coinID and send the following message to the broker. **M8** = {IDv, IDe, Payments}. The broker needs to verify each payword received from the vendor by performing hashes on it and counting the amount of paywords. If all the paywords are valid, the broker deposits the amount to the vendor's account, and then sends an acknowledgement. **M9** = {Balance Statement of the vendor's account} to the vendor as shown in Fig. 4.

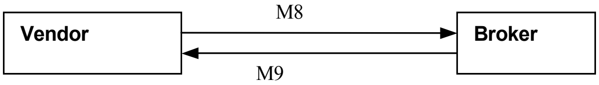

Fig. 4. Vendor-redeem transaction

6. Discussion

As we discussed in Section 3, existing mobile payword-based micro-payment protocols like Zhu protocol is almost an on-line micro-payment system for the Network Operator (NO). The NO needs to generate a corresponding endorsement hash for every payword chain, which is sent by a mobile user. Then NO sends the valid paywords (W1, W2) and the corresponding endorsement paywords (E1, E2) to the vendor in every transaction. The e-coin (paywords) in the system is user and vendor specific. This greatly limits the portability of the paywords and may very often require the user to over-purchase credit.

We have presented a real off-line from a broker and network operators and debit-based protocol suitable for micropayments in mobile networks. The protocol prevents mobile users from double spending using an e-coin Index and any internal and external adversaries from forging, so it satisfies the requirements of security that a micropayment system should have. The protocol is economical since it does not involve public-key operations per purchase. Netpay can easily handle multiple transactions between vendors. The paywords in our Mobile-NetPay protocol are not user-spcific and vendor-specific, allowing a single e-wallet to provide payment across a wide range of vendors of mobile content. The major thrust of our Mobile-Netpay protocol is that it shifts the communication traffic bottleneck from the broker and distributes it among the vendors, thus placing some processing burden on the vendors when a mobile user wishes to purchase from a vendor. The advantages for the vendor is that the Mobile-NetPay Broker guarantees redemption of valid e-coins for credit to the vendor. It also allows the vendor to focus on content provision and the Broker to provide cash-handling functionality with a bank. This makes the Mobile-NetPay protocol suitable for new and emerging vendors with a low adaoption cost for the micro-payment protocol.

Unlike the wired NetPay micro-payment protocol Mobile-NetPay attempts to minimise network traffic and data exchange between the mobile user's device, the vendor and the broker. This is in order to account for much greater network latency and reduced bandwidth on mobile networks. In addition, Mobile-NetPay supports a "previous vendor off-line" scenario by allowing the mobile user's network operator to cache limited information about the mobile user's e-wallet, allowing a validation against the broker database. As in our previous protocol, Mobile-NetPay requires a trusted broker to manage generation of e-coins and redeeming of e-coins for credit by vendors. Replicated vendor and broker servers can be used to provide load balancing and failure tolerance for the architecture. We are developing a prototype implementation of Mobile-NetPay to enable us to purchase mobile device content (music clips, tourist information and news) using a micro-payment approach across multiple vendors. We will carry out both performance (load) testing of the application architecture and response time assessment for the mobile device end user.

7. References

[1] Dai, X. and Lo, B.: NetPay – An Efficient Protocol for Micropayments on the WWW. Fifth Australian World Wide Web Conference, Australia (1999).

[2] Dai X. and Grundy J.: Architecture for a Component-based, Plug-in Micro-payment System, In Proceedings of the Fifth Asia Pacific Web Conference, LNCS 2642, Springer, April 2003, pp. 251-262.

[3] Dai, X., Grundy, J.: Architecture of a Micro-Payment System for Thin-Client Web Applications. In Proceedings of the 2002 International Conference on Internet Computing, Las Vegas, CSREA Press, June 24-27, 444--450

[4] Dai X. and Grundy J., Three Kinds of E-wallets for a NetPay Micro-payment System, The Fifth International Conference on Web Information Systems Engineering, November 22-24, 2004, Brisbane, Australia. LNCS 3306, pp. 66 - 77

[5] Dai X. and Grundy J., Three Kinds of E-wallets for a NetPay Micro-payment System, The Fifth International Conference on Web Information Systems Engineering, November 22-24, 2004, Brisbane, Australia. Lecture notes in Computer Science 3306, pp. 66 – 77

[6] Gabber, E. and Silberschatz, A.: Agora: A Minimal Distributed Protocol for Electronic Commerce, Proceedings of the Second USENIX Workshop on Electronic Commerce, *O*akland, California, November 18-21, 1996, pp. 223-232

[7] Rivest, R. and Shamir, A.: PayWord and MicroMint: Two Simple Micropayment Schemes. Proceedings of 1996 International Workshop on Security Protocols, Lecture Notes in Computer Science, Vol. 1189. Springer (1997) 69—87

[8] Zhu, J., Wang, N. and Ma, J.: A Micro-payment Scheme for Multiple-Vendor in M-Commerce. Proceedings of the IEEE International Conference on E-Commerce Technology for Dynamic E-Business (CEC-East'04), 2004

[9] Costelllo, D.: Mobility and micro-payments. 2003 http://www.epaynews.com/downloads/zafion_WP.pdf

[10] Lesk, M. Micropayments: An idea whose time has possed twice? *IEEE Security & Privacy* (2:1), 2004, pp. 61-63

[11] Geer, D.: E-Micropayments Sweat the small stuff. Journal of Industry Trends. August 2004 http://csdl2.computer.org/comp/mags/co/2004/08/r8019.pdf

[12] Rivest, R.: The MD5 Message-Digest Algorithm. RFC 1321, Internet Activities Board, 1992

[13] Wilson, T. Micro-payments rise out of the trash can. 2000 http://www.internetweek.com/columns00/bits030600.htm