Privacy Enhanced Electronic Cheque System

Vijayakrishnan Pasupathinathan Josef Pieprzyk Huaxiong Wang

Department of Computing
Macquarie University
Sydney, NSW 2109, Australia

E-mail: {krishnan,josef,hwang}@ics.mq.edu.au

Abstract

With the introduction of Check 21 law and the development of FSTC's eCheck system there has been an increasing usage of e-cheque conversions and acceptance among retailers, banks, and consumers. However, the current e-cheque system does not address issues concerning privacy, confidentiality, and traceability. We highlight the issues concerning the current electronic cheque system and provide a solution to overcome those drawbacks.

1 Introduction

Due to the convenience of purchasing as well as selling of products over the Internet, there has been tremendous growth in electronic commerce. However, the lack of consumer confidence in security of electronic transactions has been a major issue for wider acceptance. Cheque payments are the preferred method for medium and high value transactions. Cheques provide the payee an assurance of guaranteed payment as the payments are generally made to the payee's account before goods or services are delivered to the payer.

The *Check 21* U.S. federal law [5] became effective on October 2004. The law allows the banks to process cheques faster and more effectively as the paper cheque deposits are converted to an electronic image for processing. But the effectiveness of Check 21 increases when it is combined with Financial Services Technology Consortium's (FSTC) *eCheck* system [1]. The FSTC's *eCheck* system is an electronic payment instrument specifically developed for the Internet and designed to work like paper cheques and with existing checking accounts. The FSTC's *eCheck* is based on the same legal framework that applies to paper cheques and thus all existing legislations and account agreements that apply to paper cheques still hold for electronic cheques.

Our analysis show there are various security issues concerning confidentiality, privacy and tracebililty with the FSTC's *eCheck* system that are yet to be addressed. In this paper we propose a Privacy Enhanced E-Cheque (PEEC) system that can over come those problems. Similar to the conventional paper-based checking system the proposed cheque system is an off-line post-pay method and is based on the discrete logarithm problem [8]. The system provides enhanced privacy by protecting the payment details like payer's account information from merchants by allowing the payer to choose an anonymous identity during a transaction.

Related Work: Various electronic cheque (e-cheque) protocols [2, 13, 10, 11, 7] have been proposed over the years. Systems like FSTC's *eCheck* [11], NetCheque [13] and MANDATE II [2] are based on methods used in traditional paper based checking protocols. Systems like NetBill [10], ECheque and PayNow by CyberCash use a central server. Other e-checking systems are based on modified versions of e-cash protocols [7]. But most promising of all e-cheque system that has the support of major financial institutions and government agencies has been the FSTC's *eCheck* system.

Organisation: First a overview of a e-cheque system is presented (Section 2). We then describe the working FSTC's *eCheck* system and its protocol (Section 3). A security analysis of the FSTC's *eCheck* system is presented (Section 3.1). We then describe our privacy enhanced e-cheque system and its Characteristics (Section 4). We then conclude our work in (Section 5).

2 E-cheque preliminaries

Entities: As most e-cheque systems are modeled on paper based cheque system, entities involved in an e-cheque

Proceedings of the Seventh IEEE International Conference on E-Commerce Technology (CEC'05)
1530-1354/05 $20.00 © 2005 **IEEE**

system resemble the entities in a paper based cheque system. An e-cheque system involves an **Issuer**, who is a bank and is responsible for issuing e-cheques (more likely cheque book) to its account holder/customers. An **Acquirer** is a bank to which the payee is a registered account holder. A **Payer** is an entity registered with the issuer who wishes to issue a e-cheque so as to make a payment to another entity. A **Payee** is an entity who the e-cheque is addressed to, based on payer's instructions. **Trusted Third Parties** (TTP's) are entities who are implicitly trusted by other entities in an e-cheque system. These might include certification authorities for digital signatures, hardware and software manufactures for smart cards and their interfaces, and public key databases required for verification of an entity's public key.

Processing Information: At the very least an e-cheque should have the following necessary information for processing: an unique identifying e-cheque number, an unique account number that identifies the payer, an unique issuing bank identifier, e-cheque date and time stamp of when it was drawn, amount and currency of payment, payee's name and payer's signature.

3 FSTC's *eCheck*

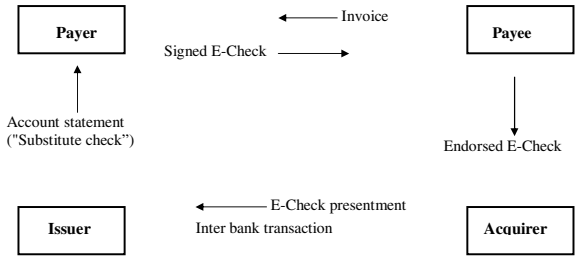

Figure 1. FSTC E-Check

FSTC's *eCheck* consists of two core components: a mark-up language (FSML - Financial Services Mark-up Language) and digital signatures (SDML - Signed Document Mark-up Language). Like Hyper Text Markup Language(HTML) both FSML and SDML are block structured markup languages, but unlike HTML which is used to define how contents are displayed in a web browser, FSML and SDML are used to define how application layer software processes *eCheck* and associated signature data. FSML defines the document data needed for electronic cheques and contains tags which identify cheque-specific data items and the SDML is the signature part of the FSML. The *eCheck* system combines these two core components to create a payment instruction secure enough to use through the Internet. The protocol flow of the FSTC's *eCheck* system is given in Fig. 1.

3.1 Issues in FSTC's *eCheck* systems

The *eCheck* system uses digital signatures for payer and payee authentication. The signature key (secret key) is stored in a tamper proof smart card and requires a secure terminal interface to generate and sign an *eCheck*. Smart card is protected by a PIN (Personal Identifying Number) and also contains the details that are used to generate a unique *eCheck* number during a transaction.

A primary concern with FSTC's *eCheck* is confidentiality of payer's *eCheck* and account information. The FSTC's *eCheck* does provide user authentication and non-repudiation by using digital signatures but does not have any inbuilt mechanism to provide data confidentiality. It relies on other application layer software encryption technologies like SSL [9] for data confidentiality. This reliance on application layer software is a security concern. The SSL protocol is secure but there have been well published browser attacks [12, 6, 4, 3] that render the SSL security ineffective.

The second concern with the *eCheck* system is regarding privacy of payer's *eCheck* and account details. In the protocol the payee not only has unrestricted access to all *eCheck* information but also to payer's account information. The payee may not be able to create new digital signed messages as he does not hold the payer's signature key, but the leakage of payer's account information is a security concern that has to be addressed.

The third concern is non-repudiation of transactional proof and smart card security. All proof of transactions concerning an *eCheck* transaction is stored on the payer's smart card. The FSTC's document specification on hardware interactions specifies that data can not copied from the smart card, that leave the smart card as the only source of proof for transactions. The loss of the smart card by a payer also implies the loss of proofs for all transactions.

Another concern with the *eCheck* system is traceability of payment instructions. As mentioned above the smart card used in the *eCheck* system stores all *eCheck* transaction details. Because of limited memory in the card, they have to be returned to either the issuer or to a trusted third party (TTP). This raises the issue of payer privacy for transaction details. The issuer or TTP will be able to obtain all transaction details including the purchasing details of the payer and will be able to create electronic dossiers on payer's purchasing habits.

4 Privacy Enhanced E-cheque (PEE-Cheque)

The system consists of three phases; (a) an initial registration protocol during which the payer obtains a set of PEE-cheques (an PEE-cheque book) that is digitally signed

by the bank. (b) a payment protocol, in which the payer creates an anonymous identity and signs an PEE-cheque that contains all necessary information for processing from that set and (c) a deposit protocol, during which the payee makes a deposit of the PEE-cheque with his/her bank.

4.1 Initial Setup

Bank $\mathcal{B}$: Bank $\mathcal{B}$ chooses primes p and q such that $|p-1| = \delta + k$ for a specified constant δ, and $p = \gamma q + 1$, for a specified integer γ. A unique subgroup G_q of prime order q of the multiplicative group $\mathcal{Z}_p^*$ and generators g_0, g_1, g_2 of G_q are defined. Hash functions $\mathcal{H}(.)$ from a family of collision intractable hash functions are defined. Bank also generates a secret key $X_B \in_R Z_q$ and corresponding public keys $h = g_0^{X_B}$, $h_1 = g_1^{X_B}$, $h_2 = g_2^{X_B}$. The Bank also chooses a value n that represents the number of PEE-cheques in a PEE-cheque book.

p, q, $\mathcal{H}(.)$, (g_0, g_1, g_2) are published along with h, h_1 and h_2 .

Payer $\mathcal{U}$: Each payer $\mathcal{U}$ has to initially register with the Bank $\mathcal{B}$. The payer generates a public key $\mathcal{I} = g_1^{u_1}$ where $u_1 \in G_q$ such that $g_1^{u_1} g_2 \neq 1$.

Payee $\mathcal{M}$: Similar to the payer, each payee $\mathcal{M}$ initially register with the Bank $\mathcal{B}$ to obtain a certified public key $\mathcal{P} = g_1^{X_P}$ where $X_P \in G_q$.

4.2 Registration Protocol

$\mathcal{U}$: $\mathcal{I} = g_1^{u_1}$
$\mathcal{U} \longrightarrow \mathcal{B}$: $\mathcal{I}$
$\mathcal{B}$: $k, [k_1, k_2, .k_i., k_n], t \in_R \mathcal{Z}_q$
$\forall$ *n* e-cheques indexed by *i* - $E_i^{'} = \mathcal{H}(\mathcal{I} g^{bact} g^i)$
$\forall$ *n* e-cheques indexed by *i* - $S_{E_i^{'}} = E_i^{'} X_B + k_i \ mod \ q$
$y = g_1^t, Y = \mathcal{I}y, S_Y = Y X_B + k \ mod \ q$
$\mathcal{B} \longrightarrow \mathcal{U}$: Y, S_Y, y, t
$[E_i^{'}, ..., E_{i+n}^{'}]$, $[S_{E_i^{'}}, ..., S_{E_{i+n}^{'}}]$
$\mathcal{U}$: $\forall$ *n* e-cheques indexed by *i* :-
VerifySignature($S_{E_i^{'}}$), VerifySignature($S_{Y^{'}}$)

The registration protocol can take place during the initial account creation or later over an Internet banking session. The payer creates identity $\mathcal{I}$ and passes it to the Issuing bank. The bank creates an PEE-cheque book where each PEE-cheque is hashed and digitally signed by the bank and consists of a serial number *i*, account number of the payer and the identity $\mathcal{I}$. The bank sends a digitally signed token Y to the user that would create a link between the identity $\mathcal{I}$ and the anonymous identity created later by the user. The VerifySign() function is a signature verification algorithm as in Schnorr Signature [14]. For post processing of e-cheques the bank needs to maintain a database indexed with token Y for all PEE-cheque books being issued to payer.

4.3 Payment Protocol

$\mathcal{M} \longrightarrow \mathcal{U}$: $\{amount, date/time, MerchantName\}_{S_\mathcal{M}}$
$\mathcal{U}$: $s, w \in_R \mathcal{Z}_q \ A = Y^s$; $A_1 = g_1^{u_1 s}, A_2 = y^s$
$Order$ = $\mathcal{H}(date/time||MerchantName||amount)$
$r = u_1 s^2 t - order.u_1.s$, $r^{'} = r.s$
$S\mathcal{U}_{E_i^{'}} = E_i^{'} g^{amount} u_1 st + w \ mod \ q$
$\mathcal{U} \longrightarrow \mathcal{M}$: $r^{'}, A_1, A_2, A, Order, E_i^{'}, S_{E_i^{'}}, Y, S_Y, S\mathcal{U}_{E_i^{'}}$
$\mathcal{M}$: $Order^{'}$ =
$\mathcal{H}(date/time||MerchantName||amount)$
VerifySignatrue(S_Y), $A \stackrel{?}{=} A_1 A_2$, $A \stackrel{?}{=} A_1^{Order^{'}} Y^{r^{'}}$
VerifySignature($S\mathcal{U}_{E_i^{'}}$)

The Payer on receiving the invoice containing payment details like *Payee name, amount, date/time* creates an anonymous identity $\mathcal{A}$ using the token Y which was issued by the Bank. The payer then sends a digitally signed PEE-cheque which includes all necessary payment information for transaction processing to the payee.

The payee on receiving the PEE-cheque creates a new hash order value ($Order'$) and verifies whether it is same as the received order value ($Order$). The Payee also verifies the bank signature on token, payer signature on the e-cheque and the anonymous identity ($\mathcal{A}$). If all verifications hold the payee accepts the PEE-cheque and sends it to the bank for processing.

4.4 Deposit Protocol

$\mathcal{M}$: $k_3 \in_R \mathcal{Z}_q$, $S\mathcal{M}_{Order'} = Order' X_M + k_3 \ mod \ q$
$\mathcal{M} \longrightarrow \mathcal{B}$: $amount, date/time, MerchantName,$
$Order^{'}, S\mathcal{M}_{Order'}, r', S_Y, Y, S\mathcal{I}_{E_i'}, E_i', A, A_1, A_2$
$\mathcal{B}$: $Order''$ = $\mathcal{H}(date/time||MerchantName||amount)$
$Order''$ $\stackrel{?}{=}$ $Order'$ $\stackrel{?}{=}$ $Order$
VerifySignature(S_Y), VerifySignature($S\mathcal{I}_{E_i'}$)
VerifySignature($S\mathcal{M}_{Order'}$)
$(\mathcal{I}, bact, i)$ = Obtain$\mathcal{I}$dbasenum(Y)
Verify$Y$value($i, Y, \mathcal{I}$), ClearFunds($\mathcal{I}, M$)

The payee's Bank on receiving the e-cheque verifies the signature on the PEE-cheque and the payer token. The Bank also creates a new hash value of the order ($Order''$) and verifies with the order sent by the payee ($Order'$) and the payer ($Order$). It then performs the function Obtain$\mathcal{I}$dbasenum() that retrieves the payer's bank account ($bact$) and his original identity ($\mathcal{I}$), and also the PEE-cheque number (i) from the database indexed by Y. It then verifies the identity of the payer and if the payer has enough funds in his/her account for clearance. If the verification was successful and sufficient funds are available it credits the payee's account and debits the payer's account.

4.5 Characteristics and Advantages of PEE-cheque system

The security of the PEE-cheque system is based on the assumptions that, (a) there exists no polynomial-time algorithm to solve the discrete log problem, (b) Schnorr signatures are unforgeable and (c) hash functions are cryptographically secure.

Privacy protection: By using an anonymous identity the system provides privacy protection to the payer's identity and account information. The anonymous identity $\mathcal{A}$ is created before conducting a transaction with the payee and does not requires communication or interaction with the Issuer. It is also created in a way, such that there is a provable linkage between the original identity and the anonymous identity. The token Y used to create an anonymous identity is cryptographically secure as it is in a form of Elgamal encryption over $\mathcal{I}$. The anonymous identity is guaranteed to be secure as long as the secret linkage value t remains known only to the payer and the bank and secret s is chosen at random before every transaction.

Identity Authentication: Identity authentication in the PEE-cheque system is based on public key verification. All public keys issued are certified by a certification authority except for the public key created that is used as an anonymous identity ($\mathcal{A}$). The proof for anonymous identity is essential a Schnorr identification protocol in a non-interactive setting. The random value here is the randomness provided by the inclusion of date-time variable in $Order$. Thus from the Schnorr identification and the payer's signature on the PEE-cheque presented to the payee, authentication of the payer is guaranteed.

The authentication of the payee towards the payer and the bank is based on verification of the payee's public key identity $\mathcal{M}$. The payment protocol described above also includes payee authentication. The commitment by the payee is digitally signed and can be verified for authenticity. The Bank authenticates the payee by verifying the digital signature on the $Order'$ that is sent by the payee during the deposit protocol.

Unforgeability: Every e-cheque created by the bank uses a cryptographically secure hash function with inputs, payer's identity $\mathcal{I}$, payer's unique bank account ($bact$) and a unique e-cheque number generated by the bank (i). The e-cheque is then digitally signed. For a e-cheque to be forgeable by the payer, the payer must be able to forge the digital signature of the bank and the bank should uses a weak hash function. This contradicts our assumptions that the Schnorr signature is secure and the bank uses cryptographically secure hash function.

5 Conclusion

In this paper we have highlighted the issues concerning the current e-cheque system and proposed a secure e-cheque system. The advantages of the PEE-cheque system over the FSTC's eCheck system are apparent. The PEE-cheque system provides improved privacy to the payer by using an anonymous identity, provides loss-tolerance by including multiple proof for transactions processed. The entities obtain unforgeable proof of transactions authorisation by the payer and acceptance by the payee. The PEE-cheque system also improves data confidentiality as the payer's account information are sealed inside the e-cheque are not available to the payee. The PEE-cheque system is also extensible and can be included to model other checking methods like pre-paid bank cheques, co-signed cheques, certified cheques or co-endorsed cheques.

References

[1] Fstc echeck initiative. http://www.echeck.org.

[2] Mandate. http://www.cryptomathic.dk/mandate.

[3] FBI web spoofing warning. *http://www.fbi.gov/pressrel/pressrel03/spoofing072103.htm*, 2003.

[4] phishing scams reel in your identity. *http://www3.cnn.com/2003/TECH /internet/07/21/ phishing/scam*, 2003.

[5] Check clearing for the 21st century act. http://www.federalreserve.gov/ paymentsystems/truncation/, October 2004.

[6] D. Boneh and D. Brumley. Remote timing attacks are practical. 12th Usenix Security Symposium, 2003.

[7] S. A. Brands. An efficient off-line electronic cash system based on the representation problem. In *246*, page 77. Centrum voor Wiskunde en Informatica (CWI), ISSN 0169-118X, 31 1993.

[8] D. Chaum and H. van Antwerpen. Undeniable signatures. In *Crypto 90*, volume LNCS of *473*, pages 212–216, 1990.

[9] N. Communications. Ssl 3.0 specification. http://wp.netscape.com/eng/ssl3/.

[10] B. Cox, J. D. Tygar, and M. Sirbu. Netbill security and transaction protocol. In *First USENIX Workshop on Electronic Commerce*, New York, July 1995. USENIX.

[11] J. K. (ed.). Financial services markup language version 1.5. Technical report, FSTC, July 1999.

[12] E. Gabrilovich and A. Gontmakher. The homograph attack. *Communications of the ACM*, 45(2):128, 2002.

[13] B. Neuman and G. Medvinsky. Requirements for network payments: A netcheque perspective. In *IEEE COMPCON-95*, pages 32–36, San Francisco, CA, USA, March 5-9 1995.

[14] C. P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4:161–174, 1991.