Anonymous Internet Credit Cards

Yi Mu and Vijay Varadharajan
School of Computing and IT, University of Western Sydney, Nepean,
PO Box 10, Kingswood, N.S.W. 2747, Australia
Email: {yimu,vijay}@cit.nepean.uws.edu.au

Abstract

In credit based payment systems, confidential data such as PIN and credit card number embedded in an electronic credit card must be protected. The authenticity of an electronic credit card in a payment process is normally achieved through an on-line trusted server such as a financial institution that runs the payment system. This paper considers some novel secure electronic credit card based payment schemes for the Internet, where the involvement of the on-line financial institution is reduced to a minimum. Our protocols use the technique of proof of knowledge of discrete logarithms that enables a prover to prove a secret without revealing the secret to the verifier.

1 Introduction

A fundamental requirement for the Internet to really become an open marketplace for goods and services in practice is the need for secure charging and payment mechanisms. Given the explosive growth in the use of the Internet, and in the services and applications being offered over the Internet, the mechanisms enabling secure commercial transactions form the foundation stone of this electronic information revolution. Needless to say security aspects become critical to the successful operation of any of these electronic payment schemes. Broadly speaking, there are two classes of electronic payment schemes. Schemes such as DigiCash [1, 2] and NetCash [3] consider cash-based electronic payments. Others such as iKP [4], NetBill [5], NetCheque [6], CyberCash [7], STT[8], SEPP [9], and SET [10] consider credit and debit based electronic payments. Our focus in this paper is on credit based electronic payment systems.

In general, an on-line credit card payment system involves at least the following parties: *Clients* (Service Users) requesting services from *Merchants* who provide the services, and *Banks* (or Financial Institutions) providing guarantee and transfer of cash and credits between *Clients* and *Merchants*. STT, SEPP, and SET refer to the Financial Institution as the *Acquirer* whereas *i*KP models this using a *Gateway* that acts as an intermediary to the existing financial network.

We use generic terms such as offer, order and slip in describing the payment schemes. The Merchant presents the information about the service in terms of an *Offer*. When the Client wishes to make a purchase, s/he sends an *Order* and a payment *Slip* to the Merchant. The *Offer* is intended for the merchant whereas the *Slip* is passed on to the Financial Institution. The Merchant requests the authorization from the Financial Institution, and upon receipt of the authorization, the Merchant confirms to the Client the conclusion of the payment transactions. This is then followed by the service delivery. Under these systems, the financial institution must be on-line, where it serves as a trusted authentication center for merchants and customers by verifying the authenticity of payment slips. The obvious reason for needing an on-line server is that merchants do not have

the right to access the PINs and credit card numbers of clients. However, the cost of an on-line service has to be taken into account in the overall assessment.

In this paper, we propose three novel credit based systems that present a promise for merchants not to use an on-line banking service during the processing of a payment. In other words, in our systems merchants can ensure the authenticity of the credit cards without any help from the financial institution that provides the credit card service. Merchant can contact the on-line authentication server for a higher level of assurance, but the option is chosen only if merchants want to have a further confirmation from the financial institution before delivering the goods or providing service. The significance of our schemes have not been shared by the existing schemes mentioned above.

Our schemes are based on the methods of proof of knowledge on the equality of logarithms (for convenience, we refer to it as *equality proof of knowledge*[11] and *non-interactive equality proof*[12]). The idea behind our scheme is that when purchases an item from a merchant, a client can prove his PIN and credit card number embedded in his electronic credit card to the merchant without revealing the secrets. In particular, we will also show that using this technique we can make clients in our payment system anonymous. Our first scheme, using the interactive equality proof[11], is somewhat similar to Chuam's electronic wallet, but the scheme is used to a different payment system. However, the second and the third schemes are entirely new: The second scheme, using the non-interactive equality proof, reduces communication flows between client and merchant. The third scheme achieves anonymous credit cards.

The rest of this paper is organized as follows. In Section 2, we introduce the scheme of equality proof of knowledge. In Section 3, we construct our protocol without considering the anonymity of clients. In Section 4, we introduce non-interactive equality proof of knowledge and construct an improved payment protocol. In Section 5, we consider clients anonymity, based on the equality proof of several secrets, and propose a new protocol.

2 Equality Proof of Knowledge

The credit based payment scheme that will be introduced in this paper is based on the scheme of proving knowledge without revealing anything about the content that has been proven. This scheme of proof was initially proposed by Chaum and Pedersen [11] and Verheul and Tilborg [12].

We assume that the prover is P and the verifier is V. The common knowledge includes the the generators $g_i \in G$ $(1 \leq i \leq n)$. The P can prove that she knows a secret x from $h_i = g_i^x \bmod q$ without revealing the secret, where q is large prime. A confidence parameter l is used to specify the level of the confidence of the protocol. The protocol is given below.

In Protocol 1, the prover P chooses a number $r \in_R \mathcal{Z}_p$, computes a_i, in terms of $a_i = g_i^r \bmod q$, $(1 \leq i \leq n)$, and sends it to V. V then chooses a challenge $c \in l$ and sends it to P. Upon receipt of c, P computes $z = cx + r \ (\bmod q)$ as a response and then sends it to V, where q is a large prime number. V checks if the equality $g_i^z = h_i^c a_i$ holds for all i. If it does, the V will accept it. If not, then there is a probability less than $1/l$ that it will still be accepted by V. This protocol proves the knowledge of x to V without revealing x. The verification of completeness, soundness, and security of the protocol is given in [12].

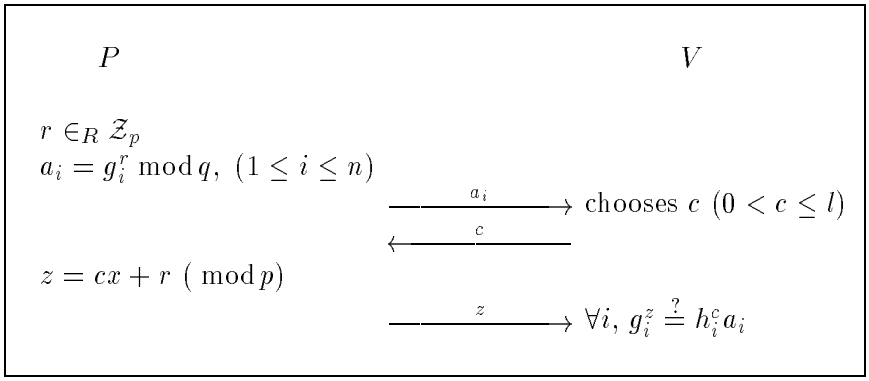
For convenience, we introduce the definition of knowledge of equality proof.

Definition 1 *Given a challenge c and a corresponding response z,*

$$k[x, c, z] \equiv \{g_1, ..., g_n, h_1, ..., h_n, a_1, ..., a_n, z, c\}$$

is the knowledge of equality proof of the secret x from $\{g_i^x\}$.

Protocol 1



With $k[x, c, z]$, the verifier can verify the authenticity of the secret without knowing the secret.

3 Construction of Our Protocol

Our protocol is based on the equality proof of knowledge given above and does not need an on-line authentication server such as a bank that runs the system. The authenticity of the credit cards is based on knowing the secret data embedded in the credit card. The secret data is constructed by concatenating the PIN number, credit number, and salt shared with the financial institution. The client can prove to the merchant that he knows the PIN number and the credit card number without revealing them to the merchant.

The notations that will be used in the description of the protocols are as follows:

- C: Client,
- M: Merchant,
- B: Bank,
- TA: Trusted Certification Authority,
- t_p: Timestamp generated by party P,
- d_p: Private Key of user P,
- e_p: Public Key of user P (associated with d_p),
- $\langle ... \rangle_{e_p}$: Public Key Encryption using e_p,
- $\langle ... \rangle_{d_p}$: Public Key based Signature using d_p.

3.1 Construction of Credit Card and Payment Tokens

An electronic credit card is obtained by a client at the stage of registration with the bank. This can be done either through a secure channel or via physical contact with the bank.

The information embedded in a credit card is as follows:

- client's ID (C),

- confidence level (l),
- $h_i = g_i^x \bmod q$, $i = 1, 2, ..., n$, where g_i are the common generators ($g_i \in G$) and x is the concatenation of PIN number, credit card number and salt,
- expiry date ($\mathcal{E}$), and
- the maximum amount ($\mathcal{A}$) that can be used.

The credit is denoted by symbol $\mathcal{C}$ expressed as follows:

$$\mathcal{C} = \langle C, l, h_1, ..., h_n, \mathcal{E}, \mathcal{A} \rangle_{d_b} \tag{1}$$

The credit card is digitally signed by the bank using its secret key d_b. We have assumed that the number of generators (n) is fixed by the bank. Therefore, it is unique in the system. The salt embedded in the credit card is unique for each client. The owner of the credit card knows the secret x. When using the credit card, the client must prove to the merchant that she knows x without revealing the secret to the merchant.

When a client wants to use the credit card for a purchase, she constructs a payment slip ($\mathcal{S}$). The payment slip contains the credit card information, the ID of merchant (M), order ($\mathcal{O}$), the amount \$ (including currency type) and a timestamp (t_c).

$$\mathcal{S} = \langle \mathcal{C}, M, \mathcal{O}, \$, t_c \rangle_{d_c} \tag{2}$$

is signed by the client. After client has paid for the goods, the merchant sends the client a receipt ($\mathcal{R}$) as a undeniable proof of the receipt of the payment. The receipt contains hashed credit card information, hashed amount that has been paid, hashed order, the challenge, and a timestamp. The receipt for client C is given by

$$\mathcal{R}_c = \langle H(\mathcal{C}), H(\$), H(\mathcal{O}), c, t_m \rangle_{d_m} \tag{3}$$

The receipt is signed by the merchant.

3.2 System Setup

Assume that there exists a trusted certificate authority (TA) such as a bank B that runs the payment system and issues certificates for all parties involved in the system. The TA could be an authorized department in the financial institution. Therefore, it is reasonable to assume that all parties involved in our system have a pair of legitimate public/secret keys (e_p, d_p) and the corresponding public key certificate ($Cert_p$) signed by the TA, where the subscript p denotes a party P. The certificate token for party P is constructed as follows:

$$Cert_p = \langle P, e_p, t, t', B \rangle_{d_b}, \tag{4}$$

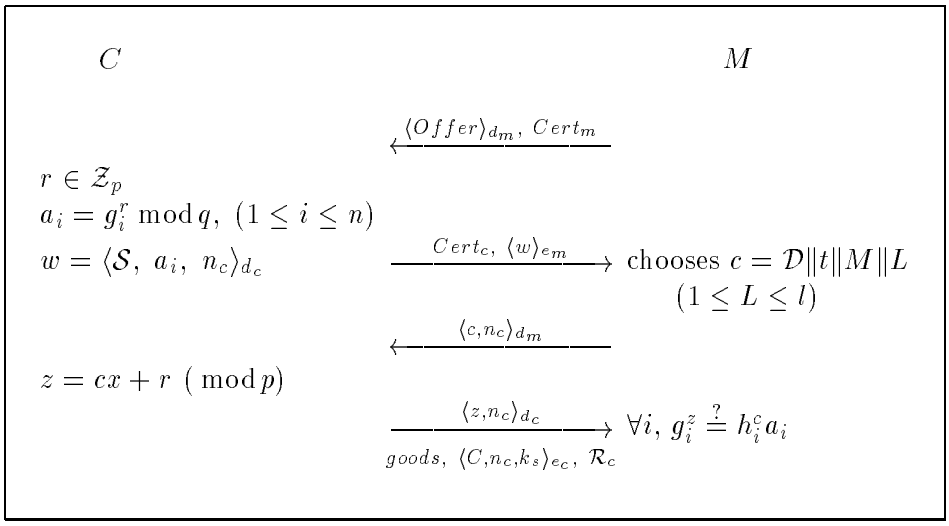
where t and t' denote a timestamp and expiry time respectively.

Consider a client C who has registered with the financial institution that runs the system. In other words, C possesses his electronic credit card issued by the bank. The client and the bank share information on the credit card such as the credit card number, PIN, and salt (we refer to the concatenation of these data as x). However, there could be an option that the bank keeps only $g^x \bmod q$ and deletes x. Once a client can prove that she knows x by using an equality proof of knowledge, she will be allowed to use the credit card.

Consider a merchant M who has registered with the bank B and can provide the credit based payment service. This means that the merchant is ready to act as a verifier of the equality proof of knowledge. As mentioned before, the merchant should not know x, but she will be convinced by the client that the client who makes the payment is a legitimate party and the owner of the credit card.

The bank B is the authority that runs the system. The bank is on-line, but it might not be necessary to get involved in the actual payment process. It could get involved only if it is absolutely necessary. That is, when the merchant asks it to do so. An example of this situation would be the case when the amount of payment is large, thus the bank should check the available amount remaining on the client's account. This situation is the same as the existing non-electronic credit card systems.

Protocol 2



3.3 The Protocol

Protocol 2 describs the payment protocol. The merchant M broadcasts the offers through the Internet. The offers have been signed by the merchant using his private key. When the client decides to purchase an item described in an offer, she sends the merchant his payment slip containing the information on an order and the electronic credit card. Once a payment has been received, the merchant verifies the client's signatures in the slip and implements the equality proof of knowledge for the secret x. The goods and receipt will be sent to the client, provided every check is successful. The provision of goods/service is encrypted with a symmetric session key k_s, which has been protected using C's public key.

Note that the challenge c sent to the client is a bit concatenation of current date, time, the merchant's ID, and a random number R. This construction is important, since it removes the risk of occurrence of two identical challenges in different payments. Increasing the size of the data does not change the confidence level, since only R is randomly chosen.

The system functions in a similar fashion to a non-electronic credit card system such as Visa or MasterCard. The payment received by the merchant can be deposited to the bank later on; for instance at the end of each day. There could be an additional option that the merchant contacts

the bank for the verification of the authenticity of the credit card before delivering the goods. This could happen when the payment is large. However, for a large number of small payments, the merchant might not need to contact the bank for reducing the cost of transactions.

In the deposit phase, the merchant sends the knowledge of equality proof $k[x, c, z]$ along with the payment slip to the bank. The deposit token is constructed as follows:

$$\{Cert_m, Cert_c, \langle\langle w, k[x, c, z], t_m\rangle_{d_m}\rangle_{e_b}\}$$

The bank returns a receipt with respect to the transaction. The receipt is constructed as follows:

$$\mathcal{R}_b = \langle C, M, w, H(\mathcal{S}), t_b\rangle_{d_b}$$

The bank keeps w and c as evidence of the transaction.

4 The Scheme Based on Non-interactive Proofs

In the previous scheme, the merchant needs to send a challenge to the client. In this section, we show an improved scheme that does not require the involvement of the merchant in an equality proof using the non-interactive method of equality proof of knowledge [13].

4.1 Non-interactive Equality Proof

In the non-interactive method, the prover and verifier do not need to interact each other. In other words, the verifier does not need to send a challenge in the process of a proof. The challenge is actually computed by the prover itself.

One again assumes that the prover is P and the verifier is V. The common knowledge is the generators $g_i \in G$ $(1 \leq i \leq n)$. P will prove that she knows the secret x from $h_i = g_i^x \bmod q$ without revealing the secret.

Definition 2 *A pair* (c, z) *satisfying*

$$c = H(g_1\|...\|g_n\|a_1\|...\|a_n\|h_1\|...\|h_n) \tag{5}$$

and

$$z = cx + r \tag{6}$$

is the knowledge of an equality proof of x*, denoted by* $K[1{:}n, g_i, a_i, h_i]$.

H denotes a strong one-way hash function. $\{a_i\}$, c, and z are called commitment, challenge, and response respectively. $K[1{:}n, g_i, a_i, h_i]$ can only be given if the secret x is known, according to equations (5) and (6). With $K[1{:}n, g_i, a_i, h_i]$, one can verify the knowledge of equality proof, by checking $g_i^z = h_i^c a_i$ $(i = 1, ..., n)$.

The idea behind this is the fact that a $K[\,]$ can be forged if the challenge c is known before the computation of the commitment a_i. Since c can only be computed after a group of $\{a_i\}$ has been computed or r has been chosen, the equality proof is equivalent to the one studied previously and cannot be forged.

4.2 The Payment Protocol

The algebraic setting is the same as in Protocol 2. We assume that the client wants to purchase an item from the merchant. To generate the knowledge of equality proof of x, the client executes an initializing run (RUN) that has the following steps:

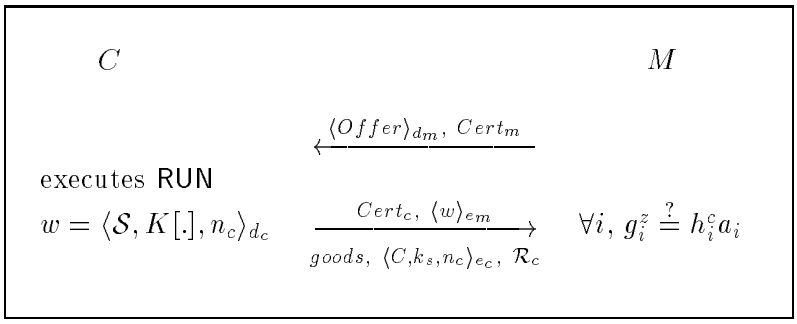
RUN:

The client:

- chooses r at random in $\mathcal{Z}_p$ and defines $r \equiv \mathcal{D}\|t\|r$,
- computes $a_i = g_i^r \bmod q,\ 1 \leq i \leq n$, and
- computes $(c, z) = K[1{:}n, g_i, a_i, h_i]$.

In the initializing run, the client needs to make sure that the r has not been used before. One possible solution is to concatenate the current date and time to r. The payment protocol is illustrated as follows (Protocol 3):

Protocol 3



In Protocol 3, once the client wants to purchase an item listed in the offer, she needs to execute the initializing run to obtain a current $K[.]$. In step two, the client sends the merchant the slip and the knowledge of equality proof on x. These information are signed by the client using his private key and then encrypted using the public key of the merchant. Upon receipt of this information, the merchant verifies all signatures therein and the authenticity of the knowledge of equality proof. If the checks are successful, the merchant sends the client the encrypted goods, encrypted delivery key k_s, and a receipt. Note that $K[.]$ must be signed along with the current payment slip $\mathcal{S}$ by the client. This will remove the risk of a dishonest merchant reusing it. Note that the confidence level defined previously is not applicable in the present system, and hence has been removed from the credit card token.

In the deposit phase, the merchant sends w to the bank. The deposit token is constructed as follows:

$$\{Cert_m, Cert_c, \langle\langle w, t_m\rangle_{d_m}\rangle_{e_b}\}$$

The bank checks the correctness of w and ensures that the set $\{a_i\}$ has not been used before. If the checks are successful, it provides a receipt with respect to the deposit made to the merchant. The receipt is constructed as follows:

$$\mathcal{R}_b = \langle C, M, w, H(\mathcal{S}), t_b\rangle_{d_b}$$

The bank needs to keep w as evidence of the transaction.

5 Anonymous Electronic Credit Card

In this section, we consider client anonymity. This is, clients' IDs are not revealed to the merchant when processing a purchase run. The anonymity of clients is based on the non-interactive equality proof studied in Section 4.

5.1 Anonymous Certificate

The client should obtain an anonymous public key certificate from the bank, the trusted certificate authority. The certificate contains an anonymous ID of the client ($\hat{C} = \{g_1^{C\|s}, g_2^{C\|s}, ..., g_n^{C\|s}\}$), the public key, a timestamp, expiry time, and the ID of the issuer:

$$Cert'_c = \langle \hat{C}, e_c, t, t', B \rangle_{d_b} \tag{7}$$

s is salt that is shared by the client and TA. The authenticity of the anonymous certificate is based on an equality proof of knowledge on C, namely the proof of the owner of the certificate without revealing the ID of the owner to the verifier.

The bank has the mapping between $\hat{C}$ and the real ID of the client. Therefore, the anonymity is only against merchants. This could be sufficient in a credit based payment system. However, if clients are allowed to have anonymous accounts with the bank, it is possible to make the credit card payments entirely anonymous. We will not consider this topic further in this paper.

5.2 Construction of Anonymous Credit Card

When an electronic credit card becomes anonymous, merchants are not able to know the ID of the card holder. The structure of a credit card is similar to the one studied in the previous section; the only difference is that it uses the anonymous ID of the client.

$$\hat{\mathcal{C}} = \langle \hat{C}, h_1, ..., h_n, \mathcal{E}, \mathcal{A} \rangle_{d_b} \tag{8}$$

The authenticity of the credit card relies upon two equality proofs of knowledge of $C\|s$ and x. The first proof ensures that the client is the owner of the credit card. The second proof shows that the client knows the secret data x. Now the payment slip and receipt tokens have $\hat{\mathcal{C}}$ instead of $\mathcal{C}$.

5.3 The Knowledge of Proof of Several Secret Numbers

In this subsection, we define and formalize the building blocks of our scheme. They are based on the non-interactive equality proofs of knowledge. The definition of the knowledge of proof of m secret numbers is as follows:

Definition 3 *A $m+1$ tuple $(c, z_1, ..., z_m)$ satisfying*

$$c = H(g_1\|...\|g_n\|a_1\|...\|a_n\|h_1^{(1)}\|...\|h_n^{(1)}\|...\|h_1^{(m)}\|...\|h_n^{(m)})$$

and

$$z_1 = cx_1 + r, \ ... \ z_m = cx_m + r.$$

is the knowledge of equality proof of $x_1, ..., x_m$, denoted by $K[1{:}n, g_i, a_i, h_i^{(1)}, ..., h_i^{(m)}]$.

Each response z_i corresponds to a secret number x_i, while the challenge c and the commitment r are unique. With the knowledge of the proof, one can verify the authenticity of $x_1, ..., x_m$, by checking $g_i^{z_j} \stackrel{?}{=} (h_i^{(j)})^c a_i \quad (i = 1, ..., n; j = 1, ..., m)$

5.4 Anonymous Protocol

The anonymous protocol is similar to the non-anonymous one, except that the client now needs to prove to the merchant both the ID and the secret x using the proof given in Definition 3.

RUN:

The client:

- chooses r at random in $\mathcal{Z}_p$ and defines $r \equiv \mathcal{D}\|t\|r$,
- computes $a_i = g_i^r \bmod q,\ (1 \leq i \leq n)$, and
- computes $(c, z_1, ..., z_m) = K[1{:}n, g_i, a_i, h_i^{(1)}, ..., h_i^{(m)}]$.

The protocol is given as follows:

Protocol 4

C		M
	$\xleftarrow{\langle Offer\rangle_{d_m},\ Cert_m}$	
executes RUN $(m=2)$		
$w = \langle \mathcal{S}, K[.], n_c\rangle_{d_c}$	$\xrightarrow{Cert'_c,\ \langle w\rangle_{em}}$	$\forall i,\ g_i^{z_1} \stackrel{?}{=} (\hat{C})^c a_i$
		$\forall i,\ g_i^{z_2} \stackrel{?}{=} h_i^c a'_i$
	$\xleftarrow{goods,\ \langle C,k,n_c\rangle_{ec},\ \mathcal{R}_c}$	

The client's signature is verified using his public key e_c embedded in the anonymous public key certificate. In the deposit phase, the merchant sends w to the bank. The deposit token is constructed as follows:

$$\{Cert_m, Cert'_c, \langle\langle w, t_m\rangle_{d_m}\rangle_{e_b}\}$$

The bank needs to check all the information in w including the authenticity of the client and the validity of the credit card and then returns a receipt with respect to the deposit made to the merchant. The receipt is constructed as follows:

$$\mathcal{R}_b = \langle \hat{C}, M, w, H(\mathcal{S}), t_b\rangle_{d_b}$$

6 Conclusion

We have introduced three methods for the establishment of credit based electronic payment systems over Internet, using the techniques of equality proof of knowledge. In particular, we have presented a construction of payment that allows anonymity of clients. The major feature of the system lies in the purchase phase where the merchant involved can ensure the authenticity of the credit card information without any help from the financial institution that runs the system. This has given a greater flexibility to the payment system as it gives a further choice to the merchants whether to use the on-line bank or not.

References

[1] D. Chaum, A. Fait, and M. Naor, "Untraceable electronic cash," in *Advances in Cryptology – CRYPTO '88 Proceedings*, pp. 319–327, 1990.

[2] D. L. Chaum, "Achieving electronic privacy," *Scientific American*, pp. 96–101, August 1992.

[3] G. Medvinsky and B. C. Neuman, "Netcash: A design for practical electronic currency on the internet," in *Proceedings of the ACM Conference on Computer and Communication Security*, November 1993.

[4] G. Medvinsky and B. C. Neuman, "ikp - a family of secure electronic payment protocols," 1995. <http://www.zurich.ibm.com/Technology/Security/extern/ ecommerce/>.

[5] M. Sirbu and J. D. Tygar, "Netbill: An internet commerce system optimized or network delivered services," <http://www.ini.cmu.edu/ netbill>.

[6] B. C. Neuman and G. Medvinsky, "Requirements for network payment: The netcheque^{tm} perspective," in *Proceedings of the IEEE CompCon'95*, March 1995.

[7] CyberCash, "The cybercash^{tm} system - how it works," <http://www.cybercash.com/cybercash/cyber2.html>.

[8] STT, "Secure transaction technology," 1995. <http://www.visa.com/Visa-stt/ Stt-os.html>.

[9] SEPP, "Secure electronic payment protocol," 1995. <http://www.mastercard.com/Sepp/>.

[10] "Secure electronic transactions," in *VISA amd MasterCard*, 1996. <http://www.mastercard.com/SET/>.

[11] D. Chaum and T. P. Pedersen, "Wallet databases with observers," in *Advances in Cryptology – CRYPTO '92 Proceedings*, pp. 89–105, Springer-Verlag, 1992.

[12] E. R. Verheul and H. C. A. van Tilborg, "Binding elgamal: A fraud-detectable alternative to key-escrow proposals," in *Advances in Cryptology – EUROCRYPTO '97 Proceedings*, pp. 119–133, Springer-Verlag, 1997.

[13] J. Camenisch, "Efficient and generalized group signatures," in *Adances in cryptology - EUROCRYPT'97, Lecture Notes in Computer Secience 1233*, pp. 465–479, Springer-Verlag, Berlin, 1997.