The 2006 International Arab Conference on Information Technology (ACIT'2006)

OFFLINE E-CHECK PAYMENT PROTOCOL

Suba Jarrar
Information and Communication Technology Center-ICTC
Al-Quds Open University
sjarrar@qou.edu

ABSTRACT

In this paper we propose an offline electronic check payment protocol, which offers payer anonymity over payee. In our protocol, we adopt the scenario of traditional check payment system: We follow the general steps series of the check payment process, satisfying all its requirements/aspects, or at least the security and functionality goals behind them, with a careful consideration to the characteristics of electronic check (eCheck), as well the anonymity of the payer. This true adaptation allows keeping up the advantages of traditional check system besides the new features offered by its electronic counterpart. In our protocol, payee will have the ability to verify the correctness and primary-validity of an eCheck, and will be provided with guarantees in order to trust and thus accept the payment, without affecting payer's anonymity. A correct eCheck is considered as a guarantee for a later deposit of the enclosed amount of money. In order to encourage payees to trust and accept such system, we offer different verification and security aspects which lead to a trusted and high-assurance eCheck payment with respect to payer anonymity. The proposed protocol will provide users with additional alternatives for anonymous electronic payments, nevertheless allowing a wider usage of eCheck.

Keywords: *e-commerce, anonymity, security protocols.*

1. INTRODUCTION

The rise of information technologies involves the dematerializing of exchanges and the increasing computerization of the means of traditional payment systems. Electronic Payments start to be essential requirements in the international financial landscape. In this paper we propose an offline eCheck payment system, which offers payer anonymity over payee.

1.1 BACKGROUND

Electronic check links the idea of traditional check with electronic payment systems. As we adopt the scenario of the paper check payment, in what follows, we briefly introduce the traditional check payment process.

Traditional check payment system can differ from one country to another, but generally, it has a particular procedure, as shown in Fig.1: the payer first issues a check (formal written order) to the payee, who in turn deposits it into his bank. The check then is cleared through a clearing process that starts when the payee's bank passes the check along with a payment request onto an intermediary bank (i.e. clearinghouse) for verification. The intermediary bank identifies the paying bank by the check's routing number, which is an international number consists of (commonly) 9 of digits and uniquely identifies the payer bank, and then presents the check to the paying bank along with a payment request. If payer's bank agrees to pay, then the check is verified to be executed, otherwise the check is rejected and returned to the payee through his bank. After check verification and the payment request is accepted, payer's bank debits his checking account and transfers the requested amount to the payee's bank (through agreed payment method). At the end of this process, the payee has a full access to the transferred money. And finally, statements/reports are sent to both payer and payee.

In this paper, we are adopting the concept of traditional check payment, with respect to payer anonymity over payee. In addition to anonymity, this true adoption allows drawing advantages of the traditional system, such as deposit later, installment check payments, etc. likewise, users can be more familiar with the eCheck payment.

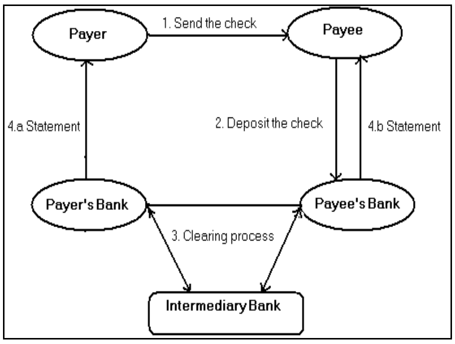

Fig.1. Traditional check payment process.

1.2 RELATED WORK

The Financial Services Technology Consortium (FSTC)[1] worked out a largely diffused and recognized financial tool "eCheck", which has a similar appearance with its counterpart paper. The customer will thus lay out a booklet of electronic checks that could be delivered through a web site or be attached to electronic mail. The manual signature will be replaced by the electronic signature, that many regards as being more reliable and the concept of safety would be thus satisfied. Authorize.Net[2] is a company that applies the work of (FSTC), through eCheck.Net process.

We observe that the FSTC project appears as a transposition on Internet of the means of payment by traditional check. Whereas its applications, such as authorize.net, utilize an intermediary in its systems and this is not a true adaptation of the check. Although, eChecks, as in the previous systems, do not reveal private information, they are not anonymous payment instruments, since the activity of the payer can be simply traced.

In some proposed protocols for offline electronic check such as [2, 3] the payer's bank is involved in issuing the echeck (e.g. adding bank's signature, etc.) but not at the time of communication between the payer and payee. The payer creates some eCheck candidates in a special way, and then the bank chooses randomly some of them to be revealed by the payer in order to be sure that they were created in a proper way. And then the bank signs the rest, in which they include a specified amount of money; contrariwise in our protocol, the bank has no role in the payment process but in later stage for final validation. Also the payer has the ability to specify the needed payment amount at the time of issuing the check without the need of his bank approval as well they offer the property of refund which contradicts the concept of the check.

Unlike the proposed approaches, our protocol offers true adaptation of the traditional check system, with all its properties. In our protocol there is no involvement of any third trusted party (bank, intermediary, etc.). Because of payer anonymity and in order for the payee to trust and accept such system, the payee has the ability to primary-validate the eCheck without any connection with the bank.

1.3 MOTIVATION OF OUR WORK

In traditional payment systems, customers prefer writing checks, thus millions of checks are processed every business day, and keeping track of all these papers is a complex procedure. Electronic payments encountered but meet with difficulties generally, especially at the present times (e.g. with credit cards, billing errors or Late Payment Fees, etc. in e-cash, not suitable for large amount payments…). Hence, for an easy, fast and effective electronic deposit alternative, eChecks become more and more essential in the electronic payment world wide. In addition, eCheck payments are favored in different business types over other payment systems, especially in B2B[3] and C2B[4]. On the other hand, because of the lack of anonymity in the current electronic check payment systems, they are having the lower score or lagging position in the contest with other anonymous electronic payment systems.

In many cases, payers prefer to keep their electronic payment activities private w.r.t payees. Payers not only prefer to avoid the payees (shops, libraries, etc.) from knowing their identities, but also from tracking their behaviors and history. E.g. in competing life, companies may want to keep anonymous in order to cover business deals that may affect its relations with other competitors. Or a payer wants to buy a large amount of some product and wants to keep anonymous so no one can know that he is the owner of the product, or the payer may want not to be bothered by the seller commercials in the future, etc.

As well, electronic payments lack the property of installment, which allows the payer the possibility to split the needed payment amount into n payments, along specific periods of time, and our protocol offers such concept, even regardless anonymity.

Some offline and online e-payment systems, propose untraceable transactions, such as e-cash [2], anonymous credit cards [7] and NetCash [8, 9]. But, although payers are looking for anonymity in e-payments, these services still do not satisfy all their needs. (E.g. in credit cards, payer is not anonymous on the billing company, while with e-cash and other small payment systems, payer cannot pay a large amount of money for large investments). Because of the need of anonymity and the few alternatives for anonymous electronic payment, in order to allow a wider base in the usage of eCheck payment, but also because some businesses can really develop only when clients can profit from a dose of

[1] http://www.fstc.org/
[2] http://www.authorizenet.com/

[3] B2B: abbreviation of Business to Business commerce.
[4] C2B: abbreviation of Customer to Business commerce.

anonymity, we propose a new alternative anonymous offline eCheck payment system.

1.4 CONTRIPUTION AND OUTLINES

Anonymity can be categorized into four main types: sender anonymity over receiver, sender anonymity over all, receiver anonymity over sender and receiver anonymity over all.
In our proposed protocol, we satisfy the first type of anonymity, such that the payer will keep anonymous over payee. In addition, our main contribution is the adaptation of the concepts of traditional check payment in addition to favor features of the eCheck. The proposed anonymous offline eCheck payment protocol will be used just like paper check. In general, the protocol will follow the steps series used in traditional check payment, and previously explained. As well, it will satisfy trust and confidence with respect to payer anonymity. This true adaptation of paper check will give our protocol the property of acceptance and familiarity. On the one hand, this will give benefit of the advantages of check payment system properties that are not yet offered by other electronic payment systems.

The previous eCheck payment systems provide audit track (i.e. financial account tracing) for each involved party to enable assured and trusted transactions. In our work, we will keep this assurance and trust, not by tracking payer’s identity, but by allowing the payee to verify the correctness of the eCheck and to be guaranteed that payer's identity will be revealed in case of deceit, keeping in mind that the new protocol is an offline payment system.

As we previously showed in the clearing process, a check cannot be verified before the clearing process, and the payee has to take the risk of the check to be returned (invalid check). Some of the reasons why the check can be rejected and returned are: insufficient funds, invalid account or routing number, closed or frozen account, etc. In our protocol we will work on assuring some of these requirements in order to give the payee more guarantees to accept such a system and verify the correctness of the eCheck with a minimum level of risks.

In the following section we introduce our proposed protocol. In 2.1 we give a short description of the cryptographic framework used in our protocol. In 2.2, we define protocol notations and abbreviations. In 2.3 we introduce our protocol, the Anonymous Offline eCheck Payment. And finally, we analyze and discuss the security, privacy and functionality requirements of our protocol in section 3.

2 OUR PROTOCOL

2.1 CRYPTOGRAPHIC FRAMEWORK

Group Signature: Group signature was first introduced by Chaum and van Heyst [4] with a basic property that allows a group member to sign anonymously on behalf of the group, while no one can reveal the signer identity but a trusted entity, in case of dispute/deceit; we consider this entity to be the group manager GM. A group manager is the one who initiates the group and generates a valid group signing keys for each member when joining the group. For more details, there are different proposed group signature schemes such as [4, 11, 16].

In addition to the basic property of signing anonymously on behalf of the group, a group signature should satisfy the following properties:

- Correctness: A verifier must be able to verify and accept a valid group signature produced by a group member.
- Unforgeability: Only a group member can generate signatures on behalf of the group.
- Unlinkability: It is computationally hard to everybody but the group manager to decide whether two valid signatures are produced by the same group member.
- Traceability: The group manager GM is always able to open a valid signature and identify the signer.
- Exculpability: Neither a group member (or a coalition of group members) nor a group manager can generate signatures as generated from another group member. This means a group member cannot be blamed to have generated a signature that he did not generate.
- Coalition-resistance: A colluding subset (even all but GM) of group members cannot generate a valid group signature that the group manager cannot link to one of the colluding group members.

With all these properties, there were still some limitations such as the problems of exposure of group signing key (i.e. the key may become vulnerable by an attacker), and the efficient exclusion of group member (a group member may leave the group). In 2001, D.X. Song proposed forward secure group signature schemes [16], which offer, in addition to all group signature properties, new properties that solve these problems.

Forward group signature, Scheme II, supports a strong level of forward security, which means that if an attacker given a set of group signing keys Φ he cannot generate a valid group signing key not in $\Psi(\Phi)$, where

$\Phi = \{k_i, t_i\}$ $1 \le i \le L$[5], k_i, t_i represent the group signing key k_i of member i for time period $t_i \subseteq T$[6], and $\Psi(\Phi)$, called the *span*[7] of Φ, represents the set of group signing keys $\{k_i, w_i\}$ $1 \le i \le L$, $t_i \le w_i \subseteq T$. Unlike prior group signature, with this concept, even when a group signing key is exposed, all group signatures generated before remain valid and do not need to be signed again. In addition, the group public key stays fixed and a group signing key of a group member evolves over time.

Other properties offered by this scheme are: time-limited membership, retroactive public revocability, and backward unlinkability. Time-limited membership means that the GM can limit a member's group membership by issuing him group signing keys which can only generate group signatures valid for some periods of time. Using scheme II, a group membership can be valid only for a time period t_m specified by GM, and has no ability to regenerate a group signing key after t_m.

With the properties of retroactive public revocability, at a time period i, GM can exclude a group signing key starting at a period j such that it becomes invalid after j and a verifier can easily check whether the signature is revoked. In Backward unlinkability, all signatures generated by the excluded signing key before time j remain anonymous and unlinkable to all but GM.

Anonymous and secure communications: We assume that all the communications between payer and payee, in our protocol, are done anonymously, such that the payee cannot identify the payee/sender, neither any link back to him. One of the possible approaches for anonymous communication that can be used is the one proposed in [1], which untraceable electronic mail by which the payer can send his eCheck as a document in an electronic mail, and the payee cannot know the sender's identity or trace the email back to him. Other alternatives for anonymous communication are presented in, [14, 17, 5]. In our protocol the payer may need a slight interactive, to have a response from the payee. Therefore we assume the anonymous communications to support the concept of anonymous response backward to the sender of the message; such in untraceable return address [1], a mechanism allows the receiver to replay a message back to the sender without tracing or identifying his address.

For secure communication, such as when the payee sends the eCheck electronically to his bank, we use public key cryptography. The sender uses the receiver public key to encrypt a massage before sending, so that no one can decrypt the message but the receiver using his private key. For the authenticity of the sender, we use a digital signature such that the sender uses his private key to sign a message that can only be decrypted with his own public key and thus the receiver can be sure of the identity of the sender.

One-Way Hash Function: "A hash function is a function, defines mathematically or otherwise, that takes a variable length input string (pre-image) and converts it to a fixed length (generally smaller) output string (hash value)" [13]. Furthermore, a one-way hash function is a function designed in such a way that it is hard to reverse the process, that is, to find a string that hashes to a given value. various one way hash functions are introduced in [10 ,6], and we consider the collision-free one-way hash functions, by which it is computationally difficult to generate two pre-images x, y with the same hash value, such that $\mathcal{H}(x) = \mathcal{H}(y)$, even a slight change in an input string should cause the hash value to change drastically.

It is computationally infeasible to produce a document that would hash to a given value or find two documents that hash to the same value, therefore, a document's hash can serve as a cryptographic equivalent of the document. This makes a one-way hash function a central notion in public-key cryptography. When producing a digital signature for a document, we no longer need to sign the entire document with a sender's private key (which can be extremely slow). It is sufficient to sign the document's hash value instead. Although a one-way hash function is used mostly for generating digital signatures, it can have other practical applications as well, such as storing passwords in a user database securely or creating a file identification system.

Public Key Infrastructure: Public-key infrastructure (PKI) is the combination of software, encryption technologies, and services that enables enterprises to protect the security of their communications and business transactions on the Internet. PKI integrate digital certificates, public-key cryptography, and certificate authorities into a total, enterprise-wide network security architecture. A typical enterprise's PKI encompasses the issuance of digital certificates to individual users and servers; end-user enrollment software; integration with corporate certificate directories; tools for managing, renewing, and revoking certificates; and related services and support.

Public key cryptography: In public key cryptography, [12, 15], each user has a pair of private and public keys (s, p). In two-parties communication between sender and receiver, before sending a message m, the sender encrypts m by applying the encryption transformation E_p determined by receiver's public key p to obtain what is called ciphertext c, such that $c = E_p(m)$. The receiver then decrypts c by applying the decryption inverse transformation D_s uniquely determined by his private key s which is unknown but to the receiver, such that $m = D_s(c)$.

[5] L is the list of all group members.

[6] T is the set of all time periods, in which the group public key is desired to be valid.

[7] A set of group signing keys in Φ, with an extended time period within T.

Protocol requirements: For efficient and proper use, the proposed protocol should satisfy privacy, security, and functionality requirements in order to be accepted for use. In this section we introduce and define these requirements. Functionality requirements are related to security and privacy requirements, since satisfying the latest leads to the functional requirements to be efficiently filled in the protocol. As we discus in protocol analysis, some security and privacy requirements leads as logical conclusion to some functional requirements.

Security and privacy requirements: *Certifiability*: eCheck is certified by the bank to be correct at the time of issuing, i.e. the bank supports its correctness.
Payer's anonymity over payee: along with the offline eCheck payment system, the payee can never identify the payer or any link back to him.

Unlinkability: payee cannot link any two eCheck payments back to the same payer.

Integrity: the eCheck cannot be modified all over the protocol steps. In other words, the payee as well the bank can verify that the received eCheck is the original one sent by the payer.

Non-repudiation: payer cannot deny that the eCheck was issued by him.

Unforgeability: no one can sign the eCheck with a forged (valid) signature on behalf of another as the signer will be identified by GM (i.e. the bank).

Unduplicatability: payer or payee cannot duplicate the same eCheck, without being detected by the bank.

Functional requirements: *Deposit-later:* as we adopt the concept of traditional check payment, our protocol should support deposit-later, which is an aspect in which the payment amount of money is deposited in a later stage, not at the time of payment, (i.e. the payment instrument is used as a guarantee for the transformation of the money at the timing date or else the payee uses the eCheck to institute legal proceedings against the payer).
*Offline eCheck***:** the payer can use his electronic payment instrument in order to pay an amount to any payee without establishing a communication with a third trusted party (e.g. bank).

Primary-Validity: the eCheck has the quality of logical and legal force to be executed.

Verifiability: the payee has the ability to verify a primary-validity of the eCheck without any communication with a third trusted party.

Correctness: eCheck is agreed and accepted by the bank to be executed, i.e. it is formal bank order.

Installmentability: the ability of payment in parts: the money may be paid at regular intervals within validity time period within which the payer has a valid signing key, allowing him to generate a valid group signature that can be verified and accepted by the payee.

2.2 LIST OF NOTATIONS

Secure communication: we assume that secure communications support confidentiality of transaction, such that no one may access the data except for the specific entity (or entities) intended.

Symbols:

- @→: denotes anonymous and secure communication channels such that payer can send and receive messages securely and anonymously.
- →: denotes secure communication channels.
- ⇒: denotes that when the fact stated before (⇒) is true then it implies, by logical consequence, the fact after to be true.
- Ch_e: Electronic check, eCheck.
- P_{desc}: The description of the purchase order set by the payer.
- O_{id}: Unique large number ∈ **Z***, where **Z*** is a set of positive integers.
- $\mathcal{M}$: amount of money to be specified in the eCheck.
- I_D: Payee account number.
- Ĩ: Payer Identity
- $\mathcal{T}$: eCheck timing/date, the time or date the check is valid to be executed after.
- T_p: Group signature validity time period.
- $\mathcal{A}$: Approval given by the payee denotes his acceptance for the eCheck
- $\mathcal{H}$: One way public hash function, (collision-free).
- h: hash value.
- B_{Gsig}: Payer's bank group signature.
- S: signed hash value using B_{Gsig}.
- $\mathcal{V}$: the upper bound of the amount of money allowed to be issued within the check, such that $\mathcal{M} \leq \mathcal{V}$

Abbreviations:

- **GM:** Group manager (bank, one or set of entities).
- **DB:** Database, records where all bank's clients are saved (local & secure).
- ***Gverif*:** the process of verifying the group signature with the verification algorithm.
- ***Look*:** examine the eCheck enclosed information**.**
- ***Sub*:** the process in which we examine an item belongs to a set of items/interval. (e.g. Sub **(**Monday, { Sunday, Monday,…, Thursday}**)** = true).
- ***OPEN*:** The group signature open process in which the group signature is opened by GM to identify the signer.
- ***Unforg*:** a process to assure whether the eCheck is forged.
- ***Valid*:** a process in which the bank verifies the validity of the eCheck, by checking the sufficiency of funds specified within the eCheck.
- ***Execute:*** a process of executing the eCheck (withdraw from payer's account and deposit into payee's account).
- ***Return*:** a process, in case the eCheck is invalid, in which the payer's identity is revealed and given with the eCheck back to the payee through his bank.

2.4 ANONYMOUS OFFLINE E-CHECK PAYMMENT

In our protocol, we propose an offline anonymous eCheck payment system. Just like with traditional checks, the actual payment is performed at the final stage, when the money is deposited in the payee's bank account, while eCheck electronic instrument is used as a guarantee for the payee to be sure that even if this is a deposit-later system, the payment will be executed, or at least the identity of the payer will be revealed in case of dispute /deceive.

In communications we assume public key cryptography to be applied in order to provide the confidentiality of transactions, in order to keep information secured from any unauthorized access. In addition, we assume using an *anonymous offline fair exchange protocol*, (to be proposed in future work), such that, either both payer and payee receive each other's items (i.e. eCheck and purchase order), or none do. The protocol is assumes to gather enough evidence during execution so that, in case one party behaves unfairly and obtains the other's item without sending his, the misbehaving party can be prosecuted.

For clarification, we categorize the steps of our proposed protocol into three main procedures, **FOUND**, **VERIFY**, **CLOSE**, and below is the description for each of them.

- **FOUND:** in this phase the payer founds the payment process.

Step1: The payer first sends to the payee a description of his purchase order $\mathcal{P}_{desc}$ through anonymous and secure communication channel. Successively, the payee sends back an order number O_{id}, which is a *unique* number given by the payee for each new purchase and linked to $\mathcal{P}_{desc}$, the payment amount of money $\mathcal{M}$, as well the payee account number I_D to be specified later in the eCheck payment .

Payer @$\rightarrow$ payee: $\mathcal{P}_{desc}$
Payee @$\rightarrow$ Payer: O_{id}, $\mathcal{M}$, I_D

Step2: The payer then fills eCheck $\mathcal{C}h_e$ with the necessary (agreed) information for the payee, such as the payee's I_D, the purchase order number O_{id}, paying amount $\mathcal{M}$, and payment timing/date $\mathcal{T}$. When the payer has the eCheck detailed with all needed information, he hashes it using a public one way hash function $\mathcal{H}$. Then the payer signs the hash h using the bank group signature B_{Gsig}. The payer then sends both the original eCheck $\mathcal{C}h_e$ with the signed hash $\mathcal{S}$ to the payee, using anonymous and secured communication channels.

Payer: $\mathcal{H}(\mathcal{C}h_e) \Rightarrow h$, where $\mathcal{C}h_e = (I_D, O_{id}, \mathcal{M}, \mathcal{T})$,
$B_{Gsig}(h) \Rightarrow \mathcal{S}$
Payer @$\rightarrow$ Payee: $(\mathcal{C}h_e, \mathcal{S})$

- **VERIFY:** in this phase the payee verifies the primary-validity of the eCheck and evaluates its correctness and thus decide whether to give his approval $\mathcal{A}$.

Step3: The payee in turn, decrypts the message; hashes the eCheck, $\mathcal{H}(\mathcal{C}h_e) = h$; verifies the signed hash $\mathcal{S}$ with respect to h (and thus verifies the signature), he also compares the check timing/date $\mathcal{T}$, with the group signature validity time period T_p.

Payee: $\mathcal{H}(\mathcal{C}h_e) \Rightarrow h$,
Gverif $(\mathcal{S})$, w.r.t (h).
Look $(I_D, O_{id}, \mathcal{M})$.
Sub $(\mathcal{T}, T_p)$.

Step4: After the eCheck is primary-validated, if the payee is assured that the eCheck is correct and guaranteed, he gives his approval $(\mathcal{A})$. And the payer gets his order trough anonymous offline fair exchange protocol.

- **CLOSE:** in this phase the traditional deposit and clear process is applied, through electronic means. And the protocol ends either by successfully transferring the money into the payee account or returning the eCheck with the payer revealed identity.

Step5: the payee deposits $(\mathcal{C}h_e, \mathcal{S}, \mathcal{A})$ later into his bank (send by electronic means). The eCheck then goes through clearing process exactly as for traditional checks, as previously mentioned.

Payee $\rightarrow$ (payee) bank: $(\mathcal{C}h_e, \mathcal{S}, \mathcal{A})$,

Step6: in the clearing process, the payer's bank re-ensures the primary validation done in step3; if the verification fails, then the bank stops proceeding in the verification process and returns the eCheck, but without the payer's revealed identity, since the payee has approved on the eCheck and thus he takes all responsibility rejecting the eCheck.

Step7: The payer's bank always opens the group signature to identify the signer, and thus can verify the validity of the eCheck.

Step8: The bank checks out its database DB to see whether the eCheck was executed before. If it is has been executed with the same O_{id}, then the payee is the cheater as he is the one who issued O_{id} (which is unique) and he must not give an approval twice to the same O_{id}. In that case the bank refuses to execute the eCheck. Otherwise, the bank examines the amount of money in the payer's account; in case it sufficient then the eCheck will be executed, else if less than $\mathcal{M}$ (which means insufficient fund), the eCheck then cannot be executed, and the bank returns the check with the payer's revealed identity to the payee's bank and in turn to the payee, who can start a legal proceeding against the payer using the eCheck as a prove of deceit.

Payer's Bank: re-ensures:

$\mathcal{H}(\mathcal{C}h_e) \Rightarrow h$,
Gverif $(\mathcal{S})$, w.r.t (h).
Look $(I_D, O_{id}, \mathcal{M})$.
Sub $(\mathcal{T}, T_p)$.

OPEN (B_{Gsig}): Identify the payer $\tilde{I}$
Unforg $(\mathcal{C}h_e, O_{id})$: If in DB $\mathcal{C}h_{e(1)} = \mathcal{C}h_{e(2)} \Rightarrow O_{id(1)} \neq O_{id(2)}$

Else payee is a cheater
Valid ($\mathcal{M}$): account $\geq \mathcal{M} \Rightarrow$ *execute* ($\mathcal{C}h_e$)
If account $< \mathcal{M} \Rightarrow$ *Return* ($\mathcal{C}h_e$, $\tilde{I}$)

In the design of anonymous offline eCheck protocol, some requirements are assumed to be satisfied. To apply this payment system, all involved parties should be outfitted with all necessary infrastructures. Both payer's and payee's banks must support such electronic payment system. Payer's bank can have a secured website, using public key infrastructure, for its clients through which payers can issue electronic checks. On the one hand, the payer and the payee must have the capacity to send and receive emails. The sender's bank must have a system (e.g. Database), in which all the electronic transactions of its clients are recorded. Nevertheless we assume that the proposed eCheck payment system will also be supported by a legal framework as in traditional check payments, in case of bankruptcy or returned eCheck.

3 PROTOCOL ANALYSIS

Our Anonymous offline eCheck payment protocol combines the concepts of traditional check with its electronic counterpart. In the protocol analysis, we map the traditional check characteristics into anonymous offline eCheck-payment features and discuss the protocol security, privacy and functional features.

As we previously mentioned, the main contribution in our protocol lies in adopting the concept of the traditional check payment, with respect to payer anonymity over payee. Therefore we consider the main characteristics of the paper-checks, and *fit* (i.e. Evaluate there importance, make changes, additions to unify them and the characteristics of eCheck w.r.t payer anonymity over payee, nevertheless, maintain the concept of paper-check payment) them to their electronics counterpart, in order to achieve an accepted and trusted as well anonymous eCheck payment system.

Within paper-check, there are mandatory informations that must be provided, otherwise, the payment cannot be accepted. These information are, (*Payer's bank name, address*, and *routing number*; *Paying amount*, *timing date*, *payee name and check number; Payer's name*, *address*, *signature*, and *account number*). In table 1 we briefly evaluate the importance and the effect of this information on the payer anonymity and the trust and acceptance of our protocol.

	Payer Anonymity (cause to break)	**verification, trust and acceptance**
Bank's name, address, and routing number	No effect, No changes	Indirect importance
Paying amount, timing date, payee name, and check number.	No effect, slight changes	Important
Payer's name, address, signature, and account number	High effect, replaced…	Important, (thus replaced by equivalent information maintaining the goals, w.r.t payer anonymity).

Table.1. Evaluation of paper check required data to fit the requirements of the anonymous offline eCheck payment.

3.1 SECURITY AND PRIVACY ANALYSIS:

Payer's bank name, address, and routing number, are informations linked to the payer bank, through which the intermediary bank (clearing house) can identify, and thus goes on the clearing process. In our protocol, the bank group signature can satisfy the goals behind these informations as the group signature public key indicated which group the signature belongs to. And as an additional privilege it certifies the eCheck. In our protocol, we use forward secure group signature scheme II [3] and assume that it is secure and satisfies all its claimed properties, as we introduced before.

With the property of time-limited membership, once the payer joins the bank, the group manager (which is the bank in our case), issues to the payer a valid group signing key in a time period T_p, which indicates that the payer has a valid bank membership during this period of time, and in turn a valid account, without giving any information about the available amount of money, or his account number. The bank will periodically regenerate new signing keys at the end of each time period. As well, the bank has the ability with the properties of *retroactive public revocability*, to exclude a group signing key starting at a period j, (e.g. starting at the time the payer has no longer a valid account in the bank), such that it became invalid after j and a verifier can easily check whether the signature is revoked. With this privilege, when the payer signs the eCheck with timing date $\mathcal{T}$ within the time interval Tp, the payee will be assured that payer’s bank will agree on the valid signature on the eCheck and logically certifies the eCheck to be correct in the validity time period, according to the property of strong forward security. This fact provides the property of eCheck *certifiability*.

Without the essential data, (payer's name, address, signature, and account number), in paper-check the payee does not accept the payment. This is completely opposite to our protocol main goal; thus we replace these informations by others that keep the same trust and assurance. As agreed in our protocol, the GM reveals the payer identity with assumption of a probable dispute or deceit. Once the bank receives the eCheck, he links it to the signer, and then examines out its validity. In that case the payer needs not to include his account number neither any private information (i.e. name address, etc.), as the payee is assured that the bank can link the eCheck to his signer and thus will return the revealed identity of the signer in case of deceit. These privileges support *payer anonymity over receiver*.

Because of unlinkability of group signature, and as there are no available private information of the payer neither any repeatable information such as the account number, our protocol supports *unlinkability* of eChecks.

Unlike in paper-check, with group signature, payer cannot forge his signature or sign on behalf of a group he does not belong to. As well, by the property of coalition-resistance, no two or more of the group members can forge a valid group signature without identifying at least one of them by the bank. Nevertheless the property of exculpability provides the property of *unforgeability*. As each group member (or subset) has a private group signing key that no one can use but its owner (it is assumed to be secured from any attack), thus the payer cannot deny signing the eCheck, and thus supports the property of *non-repudiation*.

In step1 of the protocol, the payee replays the payer request of purchase order with a unique order number O_{id}: the uniqueness of this number supports the property of *unduplicatability*, since the payee, when verifying the eCheck, does not give his approval $\mathcal{A}$ without checking out that this number O_{id} has never been approved before, in step3. So the only blamed party in a duplicated eCheck is the payee, and it is easy to detect his deceive when the bank examines its database. Similarly, the payer cannot duplicate the eCheck as he has to enclose the purchase order unique number, which indicated the uniqueness (i.e. O_{id} distinguishes the eCheck from another having the same characteristics.) of the eCheck as well, before the payee gives his approval.

In step2, the payer uses a one way hash function to hash the eCheck before signing it with the bank group signature, and sends the signed hash with the original eCheck to the payee: by this, the payee can verify that the signed hash value is the hash of the original eCheck that he accepts and agrees on its included information. Thus the payee can be sure, as well the payer and the bank, that the eCheck cannot be modified or replaced afterwards, during the whole protocol steps, as the verification will be on the signed hash value of the eCheck. And thus one way hash function ensures the *integrity* of the eCheck.

FUNCTIONAL ANALYSIS

In paper check, the Paying amount, timing date, and payee name, are agreed information by both parties to proceed on the payment. With all this, the bank can verify the validity of the check; identifies the payee and thus transfers the amount of money to his bank after the timing date. In our protocol, these informations have the same (in fact more) value in verifying the primary-validation of the eCheck by the payee and later its validity by the bank. In step3, when the payee examines out the information of the eCheck, he finds that it is issued specifically to him by I_D, with the needed amount of money $\mathcal{M}$, and with the same unique order number O_{id} sent by him at an earlier stage, and then examines the validity of the signature and whether the eCheck timing/date $\mathcal{T}$, is within the group signature validity time period T_p. The correctness of these informations in addition to the certifiability provides the *primary-validation* property.

As a logical conclusion we can see the *verifiability* of the eCheck, as the payee can easily verify the primary-validity of the eCheck without the need of a communication with the bank. By the certifiability security property and the functional primary-validity property, the eCheck has the property of *correctness*. In the normal scenario of the protocol (i.e. where there is no deceit), the bank logically agrees and accepts to execute the approved eCheck signed with a valid group signature (i.e. valid group member signing key which was issued to the payer by the bank in the first place). As we can see from the above discussion, the payee need not to communicate the bank, neither to verify the primary-validity of the eCheck nor in any of the protocol steps but when deposit the eCheck which may be performed later, and that leads the eCheck payment to be *offline*. Moreover, with all the guarantees given to the payee from the previous security, privacy and functionality properties the payee will agree on the eCheck as a guaranteed instrument, which implies the property of *deposit-later*.

In the protocol, the deposit-later property also offers a functional property of *installmentability*: since the eCheck payment can be done in a timing date $\mathcal{T}$ within the group signature validity time period T_p, the payer can also pay the requested amount of money at regular intervals, within the validity time period in a series of related eChecks. Each eCheck has a different timing/date $\mathcal{T}$ but the same unique order number O_{id}, that is divided into two parts, where the first part is fixed, indicating the unique purchase order number, and the second indicates the payment part number, x of n, where n is the payment parts number, 1≤x≤n. (E.g. 3 out of 5 indicated the third eCheck payment from 5 eChecks). Also the amount of money $\mathcal{M}$ could be different, such that each eCheck is denoted by $Ch_{e\_n} = (I_D, O_{id\_n1}, m_n, \mathcal{T}_n)$, $1 \leq n \leq z$, we have $\mathcal{M} = m_1 + m_2 + \ldots + m_z$, with *Sub* $(\mathcal{T}_n, T_p)$ = true for each n.

This mechanism keeps the advantages of traditional paper-check, by which the payer can buy in advance, as a debt, but with a guarantee for later repay. Also the payer has the ability to specify the eCheck timing to be at a time he can be assured to have enough money in his account. This facility can be offered by the payee in order to make the payment process easier and thus attracts a larger number of payers.

4 PROBLEMS AND SOLUTIONS

Although the payer identity will be revealed in case of deceit, the eCheck payment system should offer solutions for all possible risks.

One risk the payer can face is insufficient funds in the payer account (as with traditional checks); therefore, in order to reduce this risk, we may propose the eCheck to have an upper bound $\mathcal{V}$ for the amount, agreed by all protocol involved parties. $\mathcal{V}$ is specified by the bank according to the payer financial situation (i.e. according to the fund in his account), such that the payer cannot use eChecks payment with an amount exceeding $\mathcal{V}$. but this is a contradiction to the concept of the traditional check that we adopt, as the check must be open-bounded (i.e. has no limited amount). However with installmentability, and a limited amount at each part, the risk of returned eChecks is reduced.

Another problem that may occur, is that the payee may request the eCheck payment later after T_p. as with traditional check, eCheck is valid to be executed once it is deposited after a timing date $\mathcal{T}$and since the payer will be no longer a valid member in the bank (i.e. has no account) after T_p, the payee will not be paid the eCheck value by the bank who will in turn reveal the payer's identity to the payee. But the payer's main goal behind using this eCheck payment system is to keep anonymous, thus the payer may trace all his eCheck payments and before closing his account, he may save the values (amount of money) of all issued and un-withdrawn eChecks into a bank safe (as an agreement between the payer and his bank), from which the bank can pay the payee and hence need not to reveal the payer's identity.

5 CONCLUSION AND FUTURE WORK

Yet, electronic payments on Internet do not gain the confidence of all users. The existing electronic payment systems must fulfill all required security measures. And as users may sometimes want to keep anonymous, confidentiality of transactions is one of the main requirements, which offers restrictions on the knowledge of different kinds of information related to a transaction. In this paper, we proposed an anonymous offline eCheck payment protocol; we adopted the concept of the traditional check system, so we can keep its advantages. The proposed protocol offers new security, privacy and functionality features for the electronic check payment, which allow the payee to verify a primary-validity of the eCheck and thus trust and accept the payment system. The proposed protocol functions efficiently with a minimum level of risk, as the protocol decreases the possibilities of cheating or dispute by all the provided security properties and the ability of primary-validation.

In future works, we will offer the property of transferability, which allows the payee to be in the position of the payer and transfers the same eCheck to another payee following the same scenario. Most probably, our protocol can be generalized and extended to be used in other formats and applied with other payment options by offering the other three types of anonymity: sender anonymity over all, receiver anonymity over sender and receiver anonymity over all. In our protocol, payer anonymity can be optionally applied, with no effect on other properties. As well, we will work on a new anonymous offline fair exchange protocol, which allows a fair exchange between two parties, ensuring their anonymity over each other, with no interfere from a trusted third party but in case of problem or dispute.

REFERENCES

1. Chaum D., "Untraceable electronic mail, return addresses, and digital pseudonyms," *in Proceedings of Communications of the ACM*, 84--88, Feb. 1981.

2. Chaum D., Fiat A., Naor M., "Untraceable Electronic Cash," *in Proceedings of Crypto '88*, LNCS 403, Springer Verlag, pp. 319-327

3. Chaum D., Den Boer B.,van Heyst E., Mjølsnes S., Steenbeek A., "Efficient Offline Electronic Checks, Advances in Cryptology," *in Proceeding Eurocrypt '89*, LNCS 434, SpringerVerlag, 294-301.

4. Chaum D. and van Heyst E., "Group signatures", *in Proceding of Advances in Cryptology EUROCRYPT '91*, D.W. Davies (Ed.), Springer-Verlag, pp. 257-265

5. Goldschlag D., Reed Michael G., and Syverson Paul F., "Hiding Routing Information," *in the proceeding of the Workshop on Information Hiding*, 1996

6. Hirose S. and Yoshida S., "A one-way hash function based on a two-dimensional cellular automaton", *in the proceeding of The 20th Symposium on Information Theory and Its Applications (SITA97)*, Matsuyama, Japan, Dec. 1997, Proc. vol. 1, pp. 213-216.

7. Low S.H., .Maxemchuk N.F., and Paul S., "Anonymous Credit Cards," *in the proceeding of .2nd ACM Conf. Computer and Communication Security*, ACM Press, New York, 1994, pp. 108-117

8. Medvinsky G. and Clifford Neuman B.. "NetCash: A design for practical electronic currency on the Internet," *in the Proceedings of 1st the ACM Conference on Computer and Communication Security*, November 1993.

9. Medvinsky G. and Clifford Neuman B., "Electronic Currency for the Internet, Electronic Markets", 3(9/10):23-24, October 1993 (invited). Also appeared in Connexions 8(6):19-23, June 1994.

10. Merkle R.C., "A fast software one-way hash function," *Journal of Cryptology*, 3(1):43--58, 1990

11. Park S.J., Lee I.S., and Won D.H., "A practical group signature," *in Proceedings of the 1995 Japan-Korea Workshop on Information Security and Cryptography*, pages 127--133, Jan. 1995.

12. Rivest R., Shamir A., Adleman L., "A method for obtaining Digital Signatures and Public Key Cryptosystems," *in the Proceeding of Comm. of ACM*, 21 (2), pp. 120-126, Feb. 1978

13. Schneier B., *Applied Cryptography*. John Wiley & Sons, Inc., 1996, Second Edition.

14. Shields C. and Levine B.N., "A Protocol for Anonymous Communications Over Internet*," in Proceeding of 7th ACM Conference on Computer and Communication Security*, November 2000.

15. Smith P. and Skinner C., "A public--key cryptosystem and a digital signature algorithm based on the Lucas function," *in the proceeding of ASIACRYPT'94*, 298-- 306, Wollongong, 1994.

16. Song D.X., "Practical forward secure group signature schemes," *in the proceeding of ACM CCS '01*, pages 225-234. ACM Press, November 2001.

17. Syverson P., Goldschlag D., and Reed M., "Anonymous Connections and Onion Routing," *in the proceeding of IEEE Symposium on Security and Privacy*, PP44-54, 1997.