An in depth survey on the security features of NetCents micro-payment protocol

Spyridon Antakis

Eindhoven University of Technology
Department of Mathematics & Computer Science
Den Dolech 2, P.O Box 513, 5600 MB Eindhoven
Email: s.antakis@student.tue.nl

April 16, 2009

Abstract

As new e-commerce services made their appearance to our every day life, a necessity for securing small value electronic payments came on surface. The traditional heavy-weight cryptographical protocols seemed unable to fulfill the expectations at acceptable cost and thus new light-weight protocols were introduced in order to satisfy such transactions. NetCents, is such a light-weight flexible protocol that aims to secure micro-payments over the internet and defend against possible attacks. In this paper, we perform an in depth survey on the security problems that NetCents solves and we make a comparison with the MilliCent micro-payment protocol, focusing mainly on the security properties.

Keywords: *NetCents, micro-payments, floating scrips, vendor-independent, double-spending, Electronic Payment Order (EPO).*

1 Introduction

During the last decades the needs of e-commerce changed and a different model of payment transactions appeared, called *micro-payments*. A micro-payment is a small transfer of money, usually below one euro and in the range of fractions of cents. The evolution of internet and the new services that the marketplace offered to users, demanded low value transactions. Services with extremely low prices like downloading music files and accessing documents on a *pay per view* billing, were introduced in the internet community. The existing payment systems were unable and sometimes impractical to handle such transactions as the functionality cost was not acceptable and the risk involved was not sufficient to be processed by credit cards or other traditional electronic mechanisms. [1]

The way in which a payment protocol treats the transactions has significant impact to its cost and to the necessity for security mechanisms. Considering the common money that we are using in every day life, it make us realize that there are essential differences in comparison to electronic payments. Real coins are protected from duplication since they are created through an expensive procedure and bills contain watermarks and holograms which they cannot be easily counterfeited. On the other hand, an electronic coin remains vulnerable to *double spending*, which refers to the ability of an attacker to repeatedly duplicate and re-spend its electronic currency. This security threat constitutes a great obstacle for electronic payment systems. In order for a system to defend against the double spending attack, every payment must be authorized by the party that issued the coin and thus an *online* payment scheme is needed since the issuing bank must be involved in every transaction. Nevertheless, this induces serious drawbacks with respect to the scalability, usability and reliability of the payment system.

A payment protocol such as NetCents, is desired to fulfill several essential security requirements. In fact, the most important security requirement that a payment protocol should fulfill is to guarantee that is capable of defending against the double-spending threat. Furthermore, the property of *anonymity*

must be supported, aiming to keep the buyer's identity secret during payments. Usually, protocols are able to support only *partial anonymity* and not *full anonymity*. In practice, that means that vendors are not aware of the customers identity, but the bank can easily disclose their shopping habits. Moreover, *non-repudiation* is also important and it should be supported as it provides proof of integrity and origin of a transaction and can be really helpful for tracing potential frauds.

According to T.J. Poutanen [2] a transaction must be *goods atomic*, which means that:

1. *It must be fully completed or not be completed at all.*
2. *It must not create or destroy money.*
3. *It it must guarantee that the customer will receive the goods purchased, if and only if the purchase is paid.*

This paper focuses on an in depth description of NetCents micro-payment protocol, mainly from a security perspective. In section 2, a detailed analysis of NetCents is presented and all the important security features that is able to provide, are described. Finally, in section 3 a comparison between NetCents and Millicent micro-payment protocol is performed and the paper concludes about the efficiency and the general performance of NetCents.

2 NetCents Protocol

NetCents is able to support secure transactions by including the following participants: *Root certificate server*, *Customer*, *Vendor*, *Issuing authority*, *Acquirer*, *Arbiter* and *Blinding site*. A brief description for the role that each participant plays inside the NetCents protocol, is given below. [2] [3]

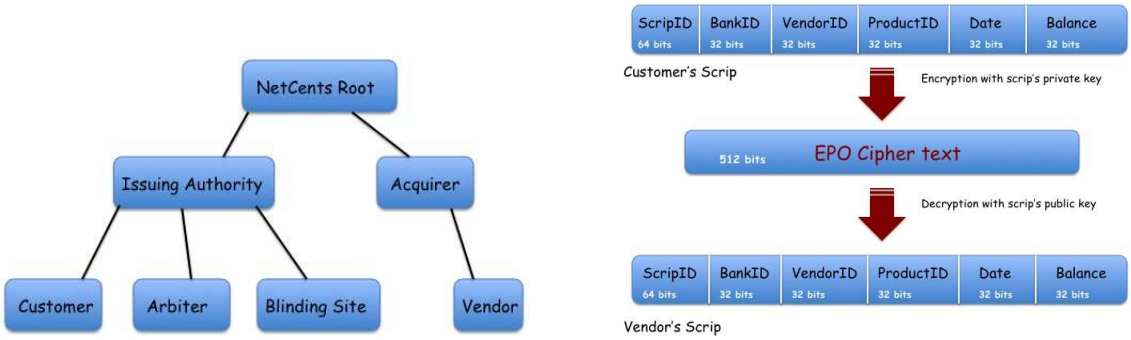
- *Root certificate server* is responsible for signing periodically certificates for the issuing authorities and must also inform the acquirers for the revoked certificates. However, it is not participating in any online transaction.
- *Customer* is the entity who buys the scrips from issuing authorities. A certificate must be provided to the customer by his issuing authority in order to be able to accomplish secure transactions and communication with the bank.
- *Vendor* refers to any online shop that accepts NetCents scrip as a valid payment method. Each vendor co-operates with an acquirer and holds a certificate signed by that acquirer.
- *Issuing authority* refers to any financial institution like a bank that sells to the customer scrips. It provides detailed transaction records that can be used for guarding against misuse and double spending. In order to be trusted, the issuing authority must possess a valid certificate signed by a root key.
- *Acquirer* represents any online financial institution that serves as vendor's clearing center.
- *Arbiter* is a mutually trusted, independent observer to transactions and he is responsible for guaranteeing that the purchased goods are delivered.
- *Blinding site* constitutes a non-functional site that has as a goal to hide the previous location of the scrip.

NetCents is a micro-payment protocol that treats funds in a unique way and it was originally designed for securing Internet payments. It provides the necessary requirements for an effective online economy such as *low cost*, *sufficient level of security*, *scalability*, *non-repudiation* and *anonymity of cash*. Money are transferred to vendors in the form of a *floating scrip*, which is a signed container of electronic currency. Floating scrips are *vendor-independent* and they enable a distributed operation that has no need for a third party. Unlike *offline* protocol payments, NetCents defends against customer fraud and it is able to successfully prevent *double spending*, since a floating scrip remains active at only one vendor at a time. Furthermore, the bank is only required to handle offline batch processing and thus the

distributed operation of the protocol is able to reduce the central bottleneck and transaction latency. In fact, the majority of expensive computations is performed by the purchaser, therefore the efficiency is significantly increased. NetCents security is built on *public key cryptography* and the main idea lies on the fact that in order to make a purchase with a scrip, the buyer needs to present to the vendor an *electronic payment order (EPO)* which is signed by the private key that is included in his scrip. In comparison to other existing micro-payment protocols, NetCents seems to fulfill efficiently most of the fundamental requirements. In the following subsections, a complete analysis of NetCents is made, the transaction mechanisms are described and an evaluation with respect to the provided security features is performed. [2] [3]

2.1 NetCents Analysis

The protocol completes a purchase with the usage of a signed *Electronic Payment Order (EPO)*. EPO is a *scrip signature* that is able to identify the vendor's location, the balance and all the important transaction information. It contains the *ScripID*, *BankID*, *VendorID*, *ProductID*, *Balance* and *Date*. The ScripID field refers to the customers identity, the Balance field is the remaining money in the scrip after a purchase and the Date field signifies the date of the transaction. The signing procedure is performed by the use of a public key signature algorithm and usually RSA is used. The choice of the algorithm is independent and focuses mainly on minimizing the verification cost between the vendor and the bank. Here, the bank is only responsible for sending out the scripts when they are requested and it is later updated offline, after vendors cash their persuaded receipts.



(a) Hierarchy of trust

(b) Electronic Payment Order (EPO)

Figure 1: *NetCents Analysis*

In practice, a hierarchy of trust is built between the members of NetCents network (fig. 1(a)) and *digital certificates* are issued to every participant. These certificates are containing *a public key*, *an expiration date* and if applicable an *IP address*. Moreover, in order to prevent vendor frauds an amount of liability that shows the level of vendor trust, is included inside the vendor certificate. Likewise, the customer certificate can include additional information such as *customer's nationality*, *address*, *age* or *memberships* that can be presented to the vendors as a proof of eligibility or discount.

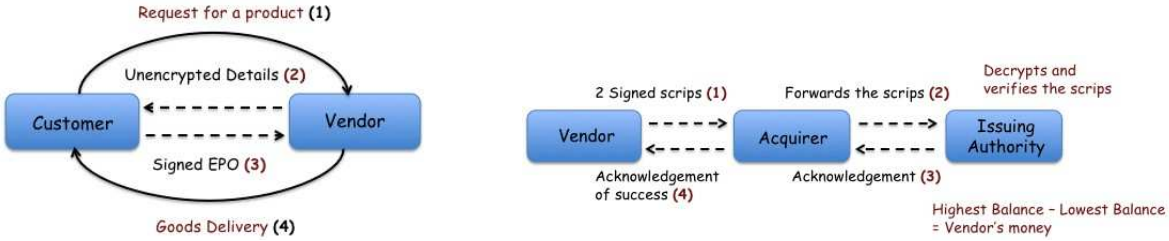
A scrip structure is divided into two main parts: i)*the vendor scrip* and ii) *the customer scrip*. The vendor scrip contains a *public key* and the customer scrip a *private key*. Each scrip has an associated balance that is decreased every time that a payment is made using the scrip. More precisely, the balance is set to a monetary value and then a payment can be made by presenting to the vendor a signature of the scrip at the balance after the purchase. The vendor scrip is signed by the issuing authority (e.g. customer's bank) and it is distributed to vendors upon customer request. Respectively, the customer scrip contains the corresponding private key. The Electronic Payment Order (EPO) concept is used and all the information fields that EPO includes are encrypted with the customer's private key (fig. 1(b)). If we suppose that RSA algorithm is used, then the items can be represented in less than 512 bits and thus a single RSA block is enough to encrypt the data. The validation is performed from the vendor with the usage of the public key and more specific with an exponentiation procedure. After decryption, a comparative validation is possible with the expected EPO values.

2.2 Basic NetCents Transaction

In NetCents, in order a customer to make a purchase is necessary first to contact an online issuing authority, like a traditional bank. The bank will generate the scrip and will return it to the customer through a secure channel. The currency of the scrip can be purchased by a traditional way, such as a cheque delivery or a direct electronic transfer from buyer's bank.

In a basic NetCents transaction a customer will submit a request for a product to a vendor. The vendor will return unencrypted the VendorID, ProductID, price of the product, date and his certificate. Then, the customer will verify the received vendor's certificate with the VendorID and the vendor's IP address and if verification succeeds, the customer will be asked to accept or decline the purchase. If the purchase is accepted and the balance of the customer's scrip is sufficient then a signed *electronic payment order (EPO)* will be generated containing the proper balance and it will be sent to the vendor. Upon the arrival of the EPO, the vendor will decrypt it and he will verify its contents. If the contents is as it is expected, the EPO will be stored and the goods together with an acknowledgement will be sent to the customer. After the transaction is completed successfully, the customer and the vendor will subtract the value of the purchase from the balance of the scrip (fig. 2(a)).

The vendors are able to get their money through a later offline transaction with the bank. Initially, they send to the acquirer two signed EPOs for each scrip. These two EPOs have different balances and they are the EPOS that had been stored after each successful transaction with customers. The acquirer forwards these EPOs into the issuing authority, which decrypts and verifies the scrips. When verification procedure is completed successfully, the lowest and highest balance are subtracted and the difference represents the amount of money owing to the vendor. Then, the funds are transferred from the issuer to the vendor's account with the involvement of the acquirer and an acknowledgement of success is returned to the vendor. The electronic funds transfer that takes place between the issuing authority and the acquirer is performed independently from the NetCents protocol and through existing bank networks (fig. 2(b)).



(a) Basic NetCents Transaction.

(b) Offline batch processing

Figure 2: *Transactions*

However, taking a closer look to the previous described transaction a problem is introduced. If the last message of vendors is not delivered, then the customer will not receive the purchased goods. In an unstable communication environment like the internet, this is very common to happen. In this case, NetCents is still able to fulfill the expectations of a reliable transaction by the means of an *online arbiter*. An online arbiter is responsible for solving any disputes based on the signed EPOs. Inside an EPO there is the ProductID and thus it can be used to ensure the delivery of the product to the buyer.

More precisely, if the goods are not delivered to a customer, then the customer first presents the same EPO again to the vendor and if the vendor refuses to retransmit the goods, an online arbiter is contacted. The customer sends the EPO to the arbiter and after the arbiter verifies its content, he sends it back to the vendor. Then, the vendor replies to the arbiter with the requested items and the arbiter delivers the goods to the customer. Nevertheless, if the vendor denies also the goods to the arbiter, then an issuing authority is notified and the customer is finally informed that the transaction will be canceled (fig. 3). In order to prevent any possible vendor fraud, at the time of the offline batch processing the bank will request a full EPO history for the scrip and if the vendor tries to use the rejected receipt, then the money will be credited to the customer. [3]

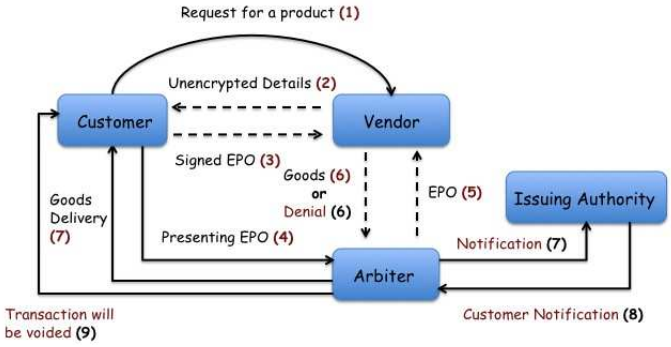


Figure 3: *Online Arbitration.*

2.3 Scrip Relocation & Larger Payments

The NetCents floating scrips introduce an extra novelty to the field of micro-payment protocols, as they are not vendor-specific. A customer is allowed to transfer a scrip between vendors directly and thus a payment can be successfully completed without anyone's else involvement. Moreover, the scrips can be active at the same time only to one vendor, therefore a double-spending detection is provided. The customer agent is responsible to select which scrip to fetch and from where. In fact, a *Least Recently Used (LRU)* algorithm is used by the customer agent in an attempt to minimize the scrip's migration cost. If the scrip is still unused, it can be found at the issuing authority or else at a vendor. The customer agent calculates the electronic payment order (EPO) for the scrip and the *new vendor identifier*. After that, the signed EPO and the *old vendor identifier* are sent to the new vendor. The new vendor checks the revocation list against fraudulent vendors and then transmits the receipt to the old vendor. As only the holder of the private key is capable of generating a valid singed EPO, it is not possible for an attacker to initiate false scrip transfer requests. After that, the old vendor verifies the receipt, signs the scrip with the current balance and transmits it to the new vendor. The new vendor performs a verification check and if it is successful he stores the scrip and the receipt. Later, the scrip can be used to solve any disputes and the receipt can be used as a common receipt for the customer. Finally, the customer is informed that the scrip was transfered and the payment transaction can start (fig. 4).

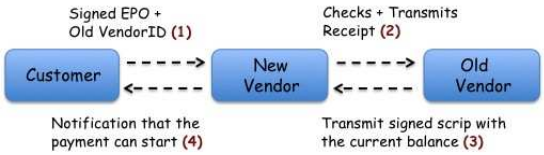


Figure 4: *Scrip Relocation*

Nevertheless, when larger payments are needed, distributed allocation of funds seems insufficient to handle such transactions. The system will have to fetch a number of scrips from other vendors in order to complete a payment and this introduces several unwanted impacts with respect to efficiency. Furthermore, repeating the relocation payment procedure can break the goods atomic properties of the protocol. Thus, in order to support larger payments NetCents uses a multiple scrip payment mechanism. When all the scrips are successfully transferred to the vendor, are sent through a single atomic transaction. Atomicity is ensured by extending EPO to include a 128-bit MD5 hash of all the ScripID's used to fulfill the transaction. Moreover, the hash is the same on each signed EPO in the transaction and it is easily reproducible by the bank. [2] [3]

2.4 Scrip Anonymity & Blinding Sites

By default, NetCents supports only *partial anonymity* since the issuing authority is able to track the purchases of a customer based on information provided by his scrip. However, NetCents delivers an optional component that an issuing authority may use to support *full anonymity* on payments. A customer creates a scrip, including the public and private keys and a random ScripID. Then, he calculates the message digest, blinds it with a random blinding factor and he sends it to the issuing authority. The issuing authority signs the blinded message digests with private keys of defined monetary value and returns the signed blinded message to the customer. The customer receives the message, removes the blinding factor and appends the signature into the scrip. When the customers initiates a purchase from a vendor, the signed vendor scrip is transferred instead of fetching relocation information. After that, the vendor passes the scrip to the issuing authority, which verifies and decrypts the signature. If everything went according to the procedure, the issuing authority will accept the scrip as valid and will store the scrip into its database of active scrips. In the end, acknowledgment messages will be sent to vendor and customer (fig. 5).

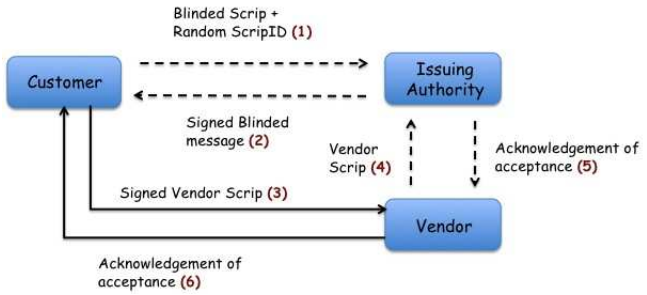


Figure 5: *Anonymity mechanism*

Another important mechanism that NetCents provides is the *blinding sites*, which are able to support a form of privacy. When the method of scrip relocation from one vendor to another is used, there is a potential threat of revealing the customer's past shopping behavior. In practice, a blinding site is a service of the issuing authority that serves as a transport of the scrip. A customer agent will contact the blinding site and will request the scrip to be fetched from its current site. Thereafter, the new vendor will be instructed to fetch the scrip from the blinding site and thus there will not be any information leakage about the previous location of the scrip.

2.5 Evaluating Security

NetCents is a micro-payment protocol that is able to defend against of a variety of security threats. The electronic payment order (EPO) combined with a public key cryptography algorithm such as RSA, is the basic key for secure payment transactions. *Double spending* and *replay attacks* are ineffective and if an attacker attempts to reuse an old EPO for a new purchase the attack will fail. The signature includes the balance within the scrip and the vendor will accept only an EPO that contains the proper new balance. Moreover, as an EPO includes a scrip balance instead of a payment amount, the vendors are not able to *over-charge* or *double-charge* a customer. Beside this, an alternation of the charged amount by a vendor is not feasible, since the customer is the only holder of the private key that signs the EPO containing the scrip balance.

An attacker can perform a man-in-the-middle attack between the communication of a customer and a vendor. A potential quote can be raised from an attacker pretending that is a valid quote from a vendor given to a customer. However, this attack is insignificant because even if the customer agrees to pay this amount, the signed receipt that will be produced for the vendor is not useful to the attacker. Another potential attack to the NetCents system would be to cause a *denial of service attack* by introducing bogus messages over the network. Vendors and issuing authorities are not capable of producing bogus messages, since the attack needs a legitimate owner of the scrip to sign the EPOs containing the vendor identifier and the balance field. However, customers can generate legitimate EPOs and then this kind of attack can

be applied to a NetCents system. The cost of detecting a bogus message is really inexpensive and requires only an EPO signature verification, thus the threat of this attack is not of such a high-importance for the system.

A very vital point with respect to security inside the NetCents system, is the initial registration of a user with an issuing authority. The sensitive information that the transfer process handles are vulnerable to attacks and therefore a secure channel must be used. The protocol's authors recommendation for securing the registration procedure, is to use an independent server separately from the one that issuing authority uses and also a TLS protected channel. This kind of weakness is significant for the system's security, however with a careful system design and by using IP monitoring mechanisms, the existing vulnerability can be taken under control.

The scrip relocation that was described in section 2.3, induces also a serious threat to the NetCents protocol. The protocol is left open for vendor fraud and a malicious vendor can reissue a scrip to multiple vendors with the basic scrip value. Thus, the trust of a vendor is challenged as a malicious vendor and a customer can co-operate in order to accomplish double-spending of a scrip at the other vendors. NetCents proposes two different methods for defending against this kind of attack. In the first method, the solution is to pass always the scrip through the issuing authority in order to detect double-spending. Unfortunately, this method is not so efficient as it introduces a service bottleneck and an increased latency, which are not unacceptable for a light-weight micro-payment protocol. Moreover, the costs of a transaction are based on the spending behavior of a customer and therefore a bank can be easily tempted to impose a minimum transaction amount. The second method applies a softer policy and takes into account the effect of the transaction size and the liability of vendors. A verification system based on probabilities is used and the detection of malicious vendors can be performed before they receive any profit from their actions. In fact, a malicious vendor can be caught when the issuer receives two payments notifications that have overlapping balances for the same scrip. [2] [3]

Another important mechanism that NetCents is designed to use for ensuring security is *non-repudiation*. Through the transaction history that is kept during the payments, an entity like arbiter can be contacted to decide about any disputes and guarantee that everything happened according to the protocol's specification. Finally, NetCents is able to support *full anonymity* as an optional feature and through the *blinding sites* it brings a certain level of privacy with respect to the customers' shopping behavior.

3 NetCents & Millicent

Millicent is a light-weight micro-payment protocol such as NetCents and it was also designed for securing electronic payments over Internet. It exists before the appearance of NetCents and it was the first protocol that represented electronic money with the form of a scrip. Millcent supports purchases with cost less than a cent and its function is based on distribution of funds. However, in comparison to NetCents, it is not able to provide larger payment transactions. A customer, can make a payment to a vendor by requesting from a broker (e.g. bank) a signed scrip specific to that vendor. After the scrip is received and verified by the vendor, a payment is authorized against that scrip. When the electronic currency of the scrip is fully used, the customer asks the broker to transmit additional funds to the vendor. In order for the broker to redeem the balance of a scrip, has to transfer funds between the vendors and thus a certain level of inflexibility appears. [2] [4]

The scrips that Millcent introduces are *vendor-dependent*, which means that can be used only at particular vendors. On the other hand, NetCents uses *floating scrips* and allows the customer to spend the electronic money designated for a specific vendor to any other vendor. Millicent uses shared keys with cryptographic hashes instead of public key cryptography and therefore operations are faster and computationally cheaper. As a comparison, the NetCents protocol applies asymmetric cryptography for customer authentication and authorization of scrip floating, which is rather computationally expensive for such micro-transactions. Millicent is also able to defend against the *double-spending* attack, since a double-spending can be easily checked by a database lookup at vendors and brokers. Furthermore, it provides only *partial anonymity* and it has not any optional components that could be enabled in order to support *full anonymity*, like NetCents. Therefore, the customer's identity is hidden from the merchants, but it does not remain secret to the bank.

Moreover, Millicent is not able to support *non-repudiation*, since the implementation of a mechanism such as *an online arbiter* is not feasible. Another important difference with NetCents protocol is that the

Property	NetCents	Millicent
Large-payments	✔	✗
Small-payments	✔	✔
Goods Atomic	✔	✗
Double-Spending protection	✔	✔
Distributed	✔	✔
Partial Anonymity	✔	✔
Full Anonymity	✔	✗
Non-Repudiation	✔	✗

Figure 6: *Basic Supported Properties*

requirement for *goods atomic* transactions is not fulfilled by Millicent. A vendor scrip can be generated by a vendor and thus when a customer spends this scrip at the vendor, the vendor may refuse the scrip by claiming that it has been already spent. In this case, the customer's secret that is used for customer authentication of a payment to a vendor is not only known to the vendor but also to the customer. Therefore, a vendor can sell these scrips to the broker without delivering the purchased goods to the customer. It is obvious that Millicent breaks the goods atomic property and only if the vendors are fully trusted can ensure the delivery of goods (fig. 6).

Finally, the direct interaction between the customers and the brokers that Millicents implements, can lead to a bottleneck of the system. In order a customer to complete a purchase to a vendor must first acquire a vendor scrip from a broker, therefore a number of communications is involved between the customer and the broker. Similarly, in the design of NetCents the usage of the floating scrips introduces a lot of communications, especially when the scrips are floating among vendors. Concluding we can say that NetCents micro-payment protocol is an extended version of Millicents protocol that successfully achieves to include a variety of additional security features by maintaining equivalent performance and fulfilling the most important payment transaction requirements. [4] [5]

4 Conclusions

The required trade-off between *transaction security* and *transaction cost* is more than obvious, when we are referring to a payment protocol design. *Security requirements* are closely related with *implementation requirements* and a micro-payment protocol design is always trying to balance between *security*, *efficiency* and *cost*. In this paper, we presented and evaluated the security features of NetCents, performing also a comparative assessment with Millicent.

In general, NetCents is a micro-payment protocol that was based on Millicent and introduces new additional features for fulfilling the expectations of a micro-payment protocol. In comparison to other protocols succeeds to provide an adequate level of security by maintaining at the same time the implementation cost at acceptable levels. It defends against the double spending attack as it provides detection mechanisms and it supports *partial anonymity* by default. Moreover in contrast to Millicent, it has an optional mechanism for implementing *full anonymity* and an online arbitration mechanism that can ensure the delivery of goods to the customer. NetCents supports the non-repudiation feature and it prevents customer fraud by the means of public key cryptography and the concept of the electronic payment order (EPO). It has a decentralized nature and with the usage of floating scrips has no need for a third party between the customer and vendor communication.

Nevertheless, even if NetCents successfully supplies a satisfying level of functionality, it demonstrates also some unwanted and maybe discouraging drawbacks for a real implementation. In my opinion, controlling vendor fraud is still a challenging and critical point, inside the NetCents design. Putting trust on vendors is a difficult decision and needs a special treatment as it can lead to vendor frauds. NetCents uses two basic countermeasures in order to control vendor fraud, but both of them include a disadvantage. Passing the scrip through an issuing authority introduces a service bottleneck and increases latency and applying a probabilistic verification procedure can not guarantee complete security. Furthermore, the

involved risk is very low since most of the transactions are referring to micro-payments in the range of a couple of euros or even less. Thus, using asymmetric instead of symmetric cryptography is not always acceptable because it introduces a computational cost for the payment system.

In conclusion, we can say that NetCents is a micro-payment protocol ready for commercial use. However, its adoption from the industry is something difficult and needs time, as people usually hesitate to trust new methods of electronic payment.

References

[1] *B.Weger* "Cryptography II - Cryptographic Systems", Eindhoven University of Technology, Department of Mathematics & Computer Science, v1.2, (December 2008).

[2] *Tomi J. Poutanen* "Netcents Protocol for Inexpensive Internet Payments", University of Toronto, Department of Electrical and Computer Engineering, (1997).

[3] *T. Poutanen, H. Hinton and M. Stumm* "NetCents: a lightweight protocol for secure micropayments", Proceedings of the 3rd conference on USENIX Workshop on Electronic Commerce, Vol. 3, (1998).

[4] *S. Glassman, M. Manasse, M. Abadi, P. Gauthier and P. Sobalvarro* "The Millicent Protocol for Inexpensive Electronic Commerce" `http://www.w3.org/Conferences/WWW4/Papers/246/` (Accessed on: 12/04/2009)

[5] *U. Yuwathiticharoenwong and Y. Permpoontanalarp* "An Agent-Based Approach to Micropayment system" In Proceedings of International Conference on Information Technology, Thummasat University Press, (2001).