The following paper was originally published in the
Proceedings of the 3rd USENIX Workshop on Electronic Commerce
Boston, Massachusetts, August 31–September 3, 1998

NetCents: A Lightweight Protocol for Secure Micropayments

Tomi Poutanen, *University of Toronto*
Heather Hinton, *Ryerson University*
Michael Stumm, *University of Toronto*

For more information about USENIX Association contact:

1. Phone: 1 510 528-8649
2. FAX: 1 510 548-5738
3. Email: office@usenix.org
4. WWW URL: http://www.usenix.org/

NetCents: A Lightweight Protocol for Secure Micropayments

Tomi Poutanen
University of Toronto
poutanen@embanet.com

Heather Hinton
Ryerson University
hhinton@ee.ryerson.ca

Michael Stumm
University of Toronto
stumm@eecg.toronto.edu

Toronto, Canada

Abstract

NetCents is a lightweight, flexible and secure protocol for electronic commerce over the Internet that is designed to support purchases ranging in value from a fraction of a penny and up. NetCents differs from previous protocols in several respects: NetCents uses (vendor-independent) *floating scrips* as signed containers of electronic currency, passed from vendor to vendor. This allows NetCents to incorporate decentralized verification of electronic currency at a vendor's server with offline payment capture. Customer trust is not required within this protocol and a probabilistic verification scheme is used to effectively limit vendor fraud. An online arbiter is implemented that will ensure proper delivery of purchased goods and that can settle most customer/vendor disputes. NetCents can be extended to support fully anonymous payments. In this paper we describe the NetCents protocol and present experimental results of a prototype implementation.

1 Introduction

This paper introduces NetCents, a novel protocol for electronic commerce transactions. NetCents was designed to support micropayment transactions, payments that range in value from a fraction of a penny to several dollars, and was extended to handle larger value purchases. A number of systems capable of supporting micropayments have been proposed and implemented, but none have established themselves as a de facto standard. The lack of a standard payment protocol that handles the full range of payments has arguably limited the growth of consumer driven electronic commerce on the Internet. Electronic payment protocols in general require a tradeoff between transaction security and transaction cost. NetCents effectively bridges that gap and satisfies the requirements of a universal payment mechanism. The work builds on the *NetBill* [CTS95], *DigiCash* [Ch95] and *Agora* [GaSi96] protocols and in particular the *Millicent* [Man95] protocol.

The key innovation of NetCents is its use of *floating scrips*. A NetCents floating scrip is a signed container of electronic currency passed from one vendor to another, such that it is active at only one vendor at a time. A scrip has a monetary value or balance associated with it, as well as a unique public and private key. Public key cryptography is used for increased security and privacy. It also enables NetCents to provide the anonymity of cash, yet the buyer can dispute a purchase effectively to an online arbiter.

NetCents functions in an offline fashion, without the need to contact a verification server in the common case. Floating scrips are vendor-independent and are passed from vendor to vendor without the need for central authorization. NetCents is scalable and supports multiple currencies and issuing authorities (or mints) and it provides adequate security, privacy and non-repudiation.

NetCents was designed to minimize operational costs. The bank is only required for offline batch processing and customer services. Vendor overhead is carefully minimized by offloading the majority of the computation load onto the purchaser; in the common case, payment verification is reduced to two modular multiplications. Vendor communication costs are reduced by distributing the scrips in a LRU fashion, in an attempt to exploit the users' tendency to shop repeatedly at the same locations

In the following section we discuss some current electronic commerce protocols. The third section describes the NetCents payment protocol in detail. We then discuss the security properties of the protocol in section four. Section five describes our implementation of the NetCents. Finally, we compare NetCents with other micropayment protocols.

2 Related Work

2.1 Basic Protocol Properties

In a computerized transaction system, a transaction should have four characteristics: atomicity, consistency, isolation and durability [GR93], which are commonly referred to as the ACID properties. The ACID properties have recently also been used to evaluate electronic payment systems [CTS95,

CHTY96]. The atomicity property has been further subdivided in the context of electronic commerce to include the delivery of purchased goods as part of the evaluation of a payment transaction [Ty96]. This is especially relevant in the realm of Internet commerce, where communication channels are not reliable.

Money atomicity refers to the atomic transfer of money, where a transaction either fully completes or it does not complete at all and does not create or destroy money. A transaction is **goods atomic** if it is money atomic and if the customer will receive the goods purchased if and only if the merchant is paid. A goods atomic transaction provides an atomic swap of electronic goods for funds. If a protocol is goods atomic and allows a consumer and merchant to prove exactly what was delivered, it satisfies **certified delivery** requirements. In case of a dispute, this evidence can be shown to a trusted arbiter to prove exactly what was delivered.

Other implementation and usability issues in Internet payment systems include anonymity, scalability, divisibility, transferability, interoperability, and non-repudiation. Anonymity implies that the identity of the buyer is not revealed in the course of a transaction. Most protocols achieve anonymity against vendors and snoopers (partial anonymity) but not against the buyer's bank (full anonymity) which can track the user's shopping habits. System scalability is of paramount importance in order to accommodate a worldwide user base without bottlenecks and single points of failure. Divisibility of currency is desired in order to pay amounts of arbitrary value. Transferability is the ability to transfer funds to other users or financial institutions. Interoperability requires the system to function with multiple financial institutions and in many currencies. Finally, non-repudiation provides proof of integrity and origin of an online transactions that can be verified by any party – a requirement for effectively policing online fraud.

2.2 Hybrid Payment Systems

Currently, most online purchases involve the entry of credit card information over a secure connection. However, there are security concerns in the form of the delivery and storage of credit card information, and the existence of Trojan sites [TW96] that mimic legitimate vendors in an attempt to gain credit card numbers. Also, not only is manual entry cumbersome but there is also a high minimum transaction charge. This charge can be amortized over several payments at one vendor by using accounts. Account-based payments systems have further problems: the cumbersome registration process discourages "shopping around". There is no policing of promised service from the online service, and the customer may have a difficult time withdrawing the balance in her account.

2.3 Online Payment Protocols

With an online protocol, a central payment authority is contacted to authorize each transaction. In general, online systems (if designed and implemented properly) secure the merchant and the bank against customer fraud, since every payment is approved by the customer's bank. Customers, however, may counter theft or loss of their electronic money, and they may be cheated by merchants via misrepresentation of goods or failed delivery. The primary disadvantage of online authorization is the associated per transaction cost, imposed by the requirement for a highly reliable and efficient clearing system at the customer's bank.

Notable online protocols include CyberCoin (http://www.cybercash.com), DigiCash [Ch95], NetBill [CTS95], and, SET [SET97]. With CyberCoin, a client makes a payment by signing a fund transfer request to the merchant. The merchant submits this signed request to the bank for authorization of payment. Depending on the availability of funds in the buyer's account, the bank will reply to the merchant with a signed authorization or refusal. The scalability of the CyberCoin protocol is questionable since it relies on the availability of a single online bank. The protocol is not fully anonymous in that it allows the issuing bank to track every purchase. Finally, CyberCoin is not inexpensive: the system restricts payments to multiples of 25 cents, and charges a minimum authorization fee of 8 cents.

NetBill extends the above payment mechanism by supporting goods atomicity and certified delivery. This is accomplished with the use of encrypted delivery and signed price quotes and sales agreements, which can later be used by an arbiter to resolve disputes. The drawback of the protocol is the addition of extra messages, and the significant increase in the amount of encryption used.

DigiCash uses blind signatures to provide a fully anonymous coin-based payment system [Ch82]. Unfortunately, the need for sophisticated cryptographic functions for each coin requires added computational resources for the bank to validate the purchase. Flaws in the DigiCash protocol have been shown to lead to an inconsistent state violating the money atomic definition [CST95]. If transfer of DigiCash tokens from customer to merchant is interrupted, then it is possible that both or neither party may believe that it has legitimate access to the tokens. The protocol is also not goods-atomic, nor do the messages support non-

repudiation.

Easily the most sophisticated protocol is the SET protocol, which was designed to facilitate credit card type transactions over the Internet. SET is secure, scalable and robust. The protocol is based on the RSA encryption protocol and utilizes 1024 bit public keys, and a 2048 bit root key. The protocol makes extensive use of cryptographic primitives that include hashes, public and private key encryption, digital signatures, dual signatures and certificates. The trust relationship is hierarchical, stemming from one root key and supported through digital certificates. This enables multiple certification authorities, issuing authorities (buyer banks), and acquirers (merchant payment agencies), facilitating wide scalability. Much of this security hierarchy is also preserved in the NetCents protocol.

SET security comes at considerable computation and communication cost. In particular, the system relies on traditional *interchange networks* when clearing payments between acquirers and issuers. Currently, interchange network services charge between 5 to 10 cents for timely and secure message passing between acquirers and issuers, whether it be a credit card or a debit card transaction [GS97]. This means that SET does not scale to the micropayment range. SET, unlike other simpler online protocols, does not offer full anonymity, non-repudiation or certified delivery of purchased goods.

2.4 Offline Payment Protocols

In an offline protocol, the merchant verifies the payment using cryptographic techniques, and commits the payment to the payment authority later, in an offline batch process. Offline systems were designed to lower the cost of transactions by delaying the clearing to a batch process. Offline systems, however, suffer from the potential of double spending, whereby the electronic currency is duplicated and spent repeatedly. Thus, offline protocols concentrate on detecting and limiting fraud, and in catching the fraudulent party. They are generally suitable only for low value transactions where accountability after the fact is sufficient to deter abuse.

Two notable offline protocols include Agora [GaSi96], Mini-Pay [HeYo96]. Agora allows its payment protocol to piggyback on existing HTTP Get-Request messages without additional communication overhead. Agora also adds non-repudiation and an online arbiter that functions much like NetCents'. Agora's method of fraud control by means of customer revocation messages may not scale to a worldwide reach, however. Mini-Pay is similar to Agora but does not use revocation messages. Instead, Mini-Pay requires a daily signed authorization from the customer's bank with a defined credit limit. As such, Mini-Pay is vulnerable to double spending up to the credit limit at multiple vendors. Other offline protocols include PayWord and MicroMint [RiSh96], Micro Payment Transfer Protocol (MPTP) [HalB95], and micro-*i*KP [HSW96]

2.5 Millicent

Digital Equipment's Millicent [DEC95, Man95] does not fall into either the online or the offline category, but rather is a distributed allocation of funds to merchants, who locally authorize payments. Grossly simplified, Millicent is an automated account based system. It introduces a scrip, which is a digital money that is honored by a single vendor. In contrast to NetCents, Millicent scrip is specific to only one vendor and the scrip must be cleared by the broker when relocating the scrip elsewhere.

A Millicent customer will have their electronic credit distributed at various site accounts on the Internet, with a common management interface - in effect, prepaying for access to a vendor, as in an account based scheme. On the first visit to an online merchant, the customer requests her broker (bank) for a signed scrip specific to the merchant. The scrip is transported to the merchant, who will subsequently authorize payments from the customer against that scrip. A scrip is analogous to a prepaid calling card, or a debit card specific to one merchant.

Once the scrip is fully used, the customer asks the broker to transmit additional (scrip) funds to the merchant. If the broker does not have sufficient available funds, then it can redeem the balance of a scrip from another vendor. Thus, while most small purchases can be made directly against scrip at the vendor, the broker is required to transfer funds between vendors. The protocol was designed on the assumption that online consumers, as in the real world, tend to shop repeatedly at the same merchants. If this assumption holds, then indeed the protocol is inexpensive to the broker due to its limited and mainly offline involvement. However, if a customer does not tend to revisit Internet merchants, then this protocol reverts to an online protocol – the broker is required in every transaction to shuffle funds from vendor to vendor. As with online payment protocols, this will result in longer payment latencies and a single point of failure. The movement of funds is especially troublesome when a customer balance nears zero and the number of scrips approaches one. At this point, scrip transfers become common, requiring broker

involvement to redeem existing scrip and sign new ones.

Millicent does not use public key encryption, but uses shared keys with cryptographic hashes. Though this operation is substantially faster than public key cryptography, it implies that non-repudiation cannot be assured, and an online arbiter cannot be implemented. Shared keys also introduce communication overhead between the issuing authority and the account holder when creating and re-issuing scrips. Furthermore, the protocol is not fully anonymous, since the broker handles the customer's scrip transfers.

3 NetCents Protocol

The NetCents protocol was designed as a low cost, scalable electronic payment mechanism with the security properties of hard currency. It is based on the assumption that online shoppers tend to frequent the same vendor sites repeatedly on the Internet. With NetCents, as with Millicent, money is transferred in the form of scrip. A scrip consists of a public and a private portion. The public portion is called vendor scrip and consists of a short public key and a monetary balance. The vendor scrip is signed by the issuing authority (customer's bank) and distributed to vendors upon customer request. The private half of the scrip, the customer scrip, contains the corresponding private key and is concealed by the customer. Unlike Millicent, a NetCents scrip is not vendor-specific. This allows a customer to transfer a scrip between vendors directly without the involvement of a broker. Scrips are active at only one vendor at a time, thus enabling double-spending detection over a distributed network.

A scrip relocation algorithm minimizes scrip movement by drawing from the user's expected shopping behavior. Currently, a least recently used (LRU) policy is used in determining scrip movement. There is a fine balance between how much value a scrip can hold and how many scrips there are. A customer tending to spend all his money at only a handful of sites should have a small number of large valued scrips. A customer making many small payments to a broad range of vendors would benefit from a large number of low-value scrips. The allocation of scrips and scrip values can be fine-tuned over time by a customer software agent.

A purchase is executed with a signed electronic payment order (EPO). The EPO holds a snapshot of the scrip signed by the private key in the customer scrip. Successive payments are made against a scrip by presenting EPOs at declining balances. The EPO identifies the payer, payee, purchased item identifier, balance remaining in the scrip after the current transaction, and the time of the transaction. The EPO is encrypted within a single encryption block with the scrip private key.

The signing algorithm can be any one of a number of public key signature algorithms chosen to minimize the verification cost performed by the vendor and the bank. Our implementation uses the RSA public-key algorithm [RSA79]. By keeping the public key short, verification by the vendor and the bank is very fast [Sch96]. We chose a public key of 3, which reduces the EPO verification computation to two modular multiplications. The signing process performed by the customer is a more computationally expensive exponentiation. This mechanism effectively distributes the computational load of public key cryptography from the critical vendor server to the many buyers.

3.1 Protocol Participants

NetCents protocol participants include the Customer, Vendor, NetCents Root Certificate Server, Issuing Authority, Acquirer, Arbiter, and Blinding Site. Each participant is issued a digital certificate.

The **NetCents Root Certificate Server** periodically signs certificates for issuing authorities and acquirers and provides notification of revoked certificates. All participants have certificates that have been signed by either the root server or entities trusted by the root server.

An **issuing authority**, or issuer, is analogous to an online bank or mint that sells scrip, provides detailed transaction records, and guards against misuse and double spending. The issuing authority must possess a valid certificate signed with the NetCents root key.

An **acquirer** is an online financial institution that serves as a vendor's clearing center. The typical flow of money is from the customer to the issuing authority to the vendor's acquirer and finally to the vendor's bank account.

The **customer** creates a long term relationship with an issuing authority for the purchase of scrip. The issuing authority also provides a signed certificate that allows the customer to initiate future secure communications. A customer agent is the software interface that interacts with the system. A **vendor** is an online store that accepts NetCents scrip as a form of payment. A vendor has a long term relationship with an acquirer and holds a certificate signed by that acquirer.

Arbiters are mutually trusted, independent

observers to transactions. Their function is to guarantee delivery of purchased goods and to provide for dispute resolution. **Blinding sites** are void, non-functional sites that are used to hide the previous location of a scrip.

Digital certificates are issued to every member in the NetCents community. All digital certificates have expiry dates, an IP address where applicable, and the member's (RSA) public key. The vendor certificate also includes a liability amount, used in vendor fraud detection. The customer certificate can optionally also include terms such as address, nationality, age or memberships that can be presented to vendors for proof of eligibility or discounts.

3.2 NetCents Transactions

In this section we outline the basic NetCents protocol. A more detailed description can be found in [Pou97].

The purchase of NetCents currency by a customer is performed via a more traditional and costly mechanism, such as an electronic transfer directly from the customer's bank to the issuing authority. The funds are released from the NetCents account in the form of scrips. The private customer scrip is transferred to the customer agent over a secure SSL connection.

The base NetCents purchase involves an exchange of messages between a customer and vendor. The transaction begins with the user requesting a quote for an item on a Web site (M0). The vendor returns the VendorID, ProductID, price quote, date and the vendor certificate, in a cleartext message (M1). The customer agent verifies the VendorID and vendor IP address against the vendor certificate. Once the vendor is verified, the customer agent will prompt the user to accept or decline the purchase from the vendor. If the customer has sufficient amount of money in a scrip at the vendor, then she will generate a signed EPO with the proper (post purchase) balance for transmission to the vendor (M2). The vendor verifies the ScripID, VendorID, Balance and Date in the EPO. If verified, the EPO will be stored for a later offline transaction with the bank, and the customer is supplied the goods and an acknowledgment (M3). As in Agora, the four messages can be piggybacked on current HTTP Get requests, with no additional messages needed for payment.

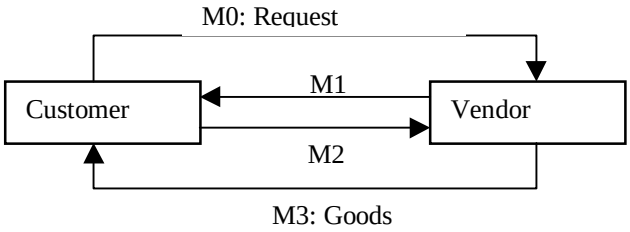

Figure 1: Basic NetCents purchase transaction

When a customer wishes to make a purchase at a vendor at which she does not have (enough) scrip, the customer agent instructs the vendor to fetch additional scrips, identifying the source – either another vendor or the issuing authority. The fetch request is accompanied with a signed EPO at its current balance identifying the new vendor. This transfer EPO serves two purposes. First, it provides the base balance of the scrip against which future payments are made. Second, the signed transfer EPO serves as proof to the vendor currently holding the scrip that the customer requested the transfer. This prevents fraudulent scrip transfer requests.

In an attempt to minimize scrip migration costs, the customer agent uses a least-recently used algorithm to select the vendor or issuer from which to request additional scrip. Prior to releasing the scrip to another vendor, the current vendor must first sign the scrip and its balance. This digital signature is used to guard against double spending initiated by a fraudulent vendor, as described later.

Some transactions may require a debit against more than one scrip. It is not sufficient to simply repeat the relocation and payment algorithm until enough scrip has been transferred to the merchant, as this can break the money atomic properties of the protocol. The vendor, acquirer, arbiter and issuing authority all require that a payment is one unbreakable unit. Hence, a multiple scrip payment procedure is as follows. Once all the scrips have been successfully transferred to the vendor, the post-purchase EPOs are sent in one atomic transaction. Atomicity is achieved by extending the EPOs to include a 128 bit MD5 hash of the concatenated scrip identifiers and ending balances. The individual signed EPOs are only useful if all signed EPOs can be presented to the issuing authority or an arbiter. This scheme is similar to SET's method of using dual signatures to link an order message with the payment instructions.

The distributed allocation of funds is efficient for micro-level payments, but becomes a burden for large value transactions. A large payment may require

the system to fetch several scrips from other vendors in order to fulfill the payment. This translates to high payment latency and a higher probability of failure. Fortunately, intelligent fund allocation policies can overcome such inefficiencies. The issuer is advised to keep a sizeable balance available at the bank and only release an appropriate amount of the customer account as floating scrip for micropayments. For example, a customer that deposits $100 may receive only 10 one-dollar floating scrips and one $90 scrip that remains at the issuer. A large value transaction involves requesting the large value scrip from the bank, a payment against the scrip, and returning the scrip to the issuer – a process no more expensive than an online transaction. Thus, NetCents minimizes the cost of low value transactions by using floating scrips but it reverts to an online protocol for higher value transactions where per transaction costs are less of an issue to security.

Vendors collect their money from the issuing authority in offline batch processing at the end of the business day. For each scrip, the vendor presents to the issuing authority two signed EPOs at differing balances. In order to better model the banking process, the EPOs are first sent to the vendor's acquirer (clearing center) which then forwards them to the issuing authority. The highest and the lowest balance scrips are decrypted and verified. These may encompass multiple purchases by the customer at that vendor. The verification process is inexpensive at two modular multiplications per EPO. Once verified, the difference in balance represents the amount owing to the vendor, and the funds are transferred from the issuer to the vendor's account at the acquirer. The electronic funds transfer between the broker, issuing authority and the acquirer is handled independently from the NetCents protocol. The intermediary EPOs can be stored for transaction tracking, if desired.

Fund transfer between users is desirable but is complicated by the lack of trust and the lack of fraud control placed on users. The transfer of scrip must pass through an issuing authority in order to prevent double spending. This is the same as purchasing scrip at one issuing authority with someone else's scrip. The cost of involving a central server does impose a minimum per transaction cost.

3.3 Online Arbitration

In case of disputes, an online arbiter can use the signed EPOs in an audit. The product identifier within the EPO can be used to ensure delivery of a product to the buyer. The NetCents system requires that the customer can reload the purchased electronic goods by presenting the vendor with the EPO. That is, if M3 is lost and the goods are not delivered properly, then the customer can reinitiate the download by presenting the EPO. If the site fails to transmit the goods, then an online arbiter is contacted with the original EPO (M4 in Figure 2). The arbiter verifies the EPO, and contacts the vendor with the same EPO (M2'). If the vendor supplies the arbiter with the requested item, then the arbiter delivers the goods to the customer (M3'). However, if the arbiter is denied the goods, and the vendor refuses to roll back the transaction, then the issuing authority is notified (M5). The vendor is flagged, and the arbiter and customer are advised (M6) that the transaction has been cancelled. At the time of offline batch processing with the flagged vendor, the bank will request a full EPO history for the scrip. If the vendor attempts to use the discredited EPO, then the money is credited to the customer and not the vendor.

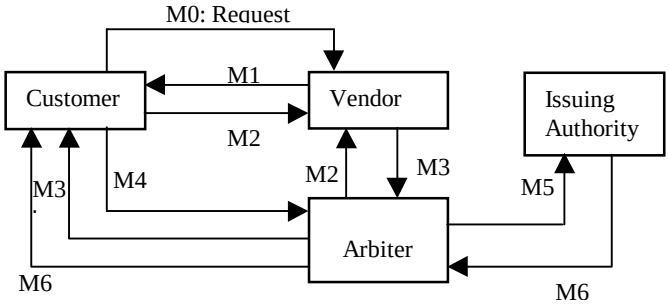

Figure 2: Online Arbiter Transactions

3.4 Anonymity and Privacy in NetCents

The protocol, as described, is not fully anonymous as the issuing authority can track the purchases against a scrip purchased by a certain client. An anonymous payment mechanism based on blind signatures has been developed as an "add on" and is described in length elsewhere [Pou97]. Like Chaum's DigiCash [Ch82], a customer can purchase scrip from an issuer by personally creating the scrip, digitally blinding it, and requesting the bank to sign it with a signature of an agreed value. When the customer is ready to make a purchase from a vendor, the unblinded, signed scrip is passed with the EPO and sent on to the issuer.

The method of moving scrip from one vendor to another presents a privacy problem by revealing the customer's past shopping behavior. It is possible to hide the details of previous vendors by passing scrips from one vendor to another via a blinding site. The

customer agent will first contact the blinding site, and request the scrip to be fetched from its current site. The vendor will then retrieve the scrip from the blinding site, and will receive no information about where the scrip was prior to the blinding site.

3.5 Vendor Fraud Control

We have shown how customers can be prevented from double spending, even when colluding. However, floating scrips place their trust on vendors holding and transmitting scrip. A malicious vendor is able to reissue scrip to multiple vendors at the base scrip value. These funds can then be spent repeatedly at each of the vendors. In contrast to offline protocols that guard against buyer fraud, vendor fraud control is a much more realistic proposition. NetCents employs a probability based verification system to detect a malicious vendor before they are able to "profit" from their crime. A vendor only benefits when he successfully double spends funds. However, the criminal vendor is caught when the issuer receives two payment notifications with overlapping balances against the same scrip.

NetCents relaxes the policing requirements to take into effect the transaction size and the vendor's liability. Consider charging a new vendor a $200 setup fee. In this case, vendor crime will only be profitable if the vendor is able to gross more than $200 for its crime. By extending the model to take into effect a vendor's accounts receivable (from the acquirer) and its expected future income (as a product of its monthly cash flow and the number of months it has been in service) a vendor liability amount can be determined and stored in the vendor certificate. It is then only necessary to police a vendor to an extent where a vendor is better off continuing honestly rather than trying to cheat.

A scrip can travel via many vendors without central notification. As a scrip moves from one vendor to another an associated liability value, L, is set to the minimum of the scrip's liability and the current vendor's liability. The value of L thus corresponds to the liability of the least trusted vendor visited. This value is used to determine how often the issuing authority should monitor the scrip and vendor. The issuer is notified of the purchase at a probability of KP/L, where P is the purchase price and K is a constant.

K must be chosen such that the probability of a vendor benefiting from the crime is less than some desired fraction. Using probability theory as the base and selecting a probability of success at less than 50%, the following inequality is used to determine K:

$$0.5 \geq (1 - K*P/L)^{L/P} + K(1 - K*P/L)^{L/P-1}$$

Depending on the price, and the scrip liability, K must be chosen such that the right-hand side of the equation is always less than 0.5. To solve for K as an equality, we are left with a multivariate function that requires an iterative search to solve. Instead, we have arbitrarily chosen the value 2 as K, since this yields a fast answer, and the formula equates to below 0.5 for any P or L. The risk is the highest at low P/L (i.e. small valued transactions against well-trusted scrip), where the maximum probability of breaking even is 0.46. For larger P/L, the function is strictly decreasing.

Consider a vendor that has requested a scrip from a site with a liability of $1000 to satisfy a 5 cent transaction. The certifying bank would be contacted at a probability of 2*0.05 / 1000, or 0.0001. Since the probability of central authentication is so low for the scale of such transactions, the aggregate cost is negligible to the banks as well as the vendors. In order for a criminal vendor to benefit in the crime, it would have to succeed in making at least 20,000 five cent purchases without two separate vendors verifying the integrity of the scrip with the central authority. Using basic probability theory, a malicious vendor, in this scenario, would have a less than 40% chance of profiting from the crime (i.e. the crime does not pay).

When an issuing authority detects double spending, then the NetCents root server is presented with the evidence. The root server verifies the evidence and contacts the vendor and the vendor's acquirer. If the evidence is found to be conclusive, then a full broadcast is made to all acquirers. The acquirers, in turn, inform all of their merchants of the malicious vendor. The vendors will refuse to transfer scrips from the malicious vendor, unless it is passed via the scrip's issuing authority.

The onus is on the acquirer to manage vendor liability, policing and punishment. In the event of a malicious vendor, the acquirer is responsible for the double spent currency up to the liability amount of the vendor. If, against odds, the double spent currency exceeds the vendor liability then the claiming parties will not be fully compensated. This limit ensures that a colluding set of vendors that together have circumvented the probability central notification scheme cannot profit from colluding.

In practice, policing can be relaxed for online content that has no palpable value associated with it – such as an online article. Whereas access to an online article to a mass of colluding customers (as is possible in an offline protocol) does have an economic value associated with it, repeated downloads by one malicious vendor fails to gain from the crime.

However, content with a tangible value, such as a query against costly data stores, needs to be guarded closely.

3.6 Currency Conversion

The importance of divisibility and the ability to handle fractions of pennies is exemplified in currency conversion. Unlike coin based payment systems that struggle to provide divisibility, NetCents' use of signed scrip balances supports payments to arbitrary precision. A NetCents scrip has a specific currency attached to it and a transparent online currency exchange mechanism is defined to handle cross-currency transactions. Since a currency conversion will typically not translate to an even number, it is important to be able to support fractions of pennies. For example, an item listed at two U.S. cents would translate to 3.52 German pfennings (@ 1.7664 marks/dollar). For a vendor to convert between currencies, the acquirer provides a periodically signed currency conversion table. The currency conversion table serves as an agreement between the acquirer and vendor where the acquirer agrees to the exchange rates for a set period of time. The acquirer would presumably set the conversion rate at a level that would protect it against unfavorable currency fluctuations.

The desired currency is included in the initial quote request message. The vendor performs the conversion on the fly and returns the quote in the customer's native currency. For the above example, a German account holder will be asked to accept a payment of 3.52 pfennings (U.S. 2 cents). The vendor, in turn, can claim the U.S. 2 cents from the acquirer by presenting the EPO spread, and referring to the currency conversion table.

4 NetCents Properties

In this section we discuss the properties of the NetCents protocol.

4.1 Atomic Properties of Transactions

The NetCents protocol can be shown to be money and goods atomic. It does not support Tygar's notion of certified delivery because of the initial clear-text transfer of item description, identifier and price from the vendor to the customer. The protocol specification can be extended to support certified delivery by requiring a signing operation by the vendor [Pou97]. However, this is a computationally expensive process that is not suitable for a low-cost payment scheme such as NetCents.

4.2 Security

NetCents is secure against an adversary who can intercept, destroy, modify and replay messages. Replay and double-spending attacks are ineffective because of the inclusion of the post-transaction balance and vendor identifier within the signed EPO and because of the vendor policing algorithm described earlier. Vendors cannot double charge a customer because the EPO contains a scrip balance, instead of a payment amount. A vendor cannot alter the amount charged for goods as they cannot generate a valid, signed EPO (this requires the customer's private key).

A man-in-the-middle can effectively raise the quote given by a vendor to a customer. However, if the customer agrees to pay this amount, the signed receipt will made out to the vendor and is of no use to the man-in-the-middle.

Only the legitimate owner of a scrip can cause it to be transferred within the system since the relocation process requires a signed EPO containing the balance and destination vendor. Malicious vendors, issuing authorities, etc therefore cannot cause a denial-of-service attack based on clogging the system with bogus messages. Only a customer can clog a system, and this will occur with legitimate EPOs generated by the customer. The cost of detecting a bogus message is the cost of an EPO signature verification, which is inexpensive by design.

The initial registration of a user with an issuing authority is vulnerable to attack. This process requires the transfer of sensitive information and should therefore pass through a secure, protected channel. We recommend that the registration procedure be hosted by a dedicated server, separated from the operational services of the issuing authority.

4.3 Cost

From an issuing authority's point of view, the system is inexpensive to implement and oversee. The issuer sells NetCents scrips to customers and distributes these scrips to vendors in response to customer requests. The issuer is updated offline when the vendors cash in their receipts. Beyond the minimal error correction and vendor policing overhead, the issuer's mission-critical workflow is independent of the transaction size. Thus, there is little incentive for an issuer to impose a minimum transaction fee. Acquirers are required only for offline batch processing and possible vendor revocation.

4.4 Scalability

The NetCents system (and protocol) was designed to be scalable. The SET protocol was followed in creating a

trusted, interoperable network of issuing authorities and acquirers. We expect the system to be truly scalable to a global reach given its distributed nature, the low vendor verification costs, and even lower bank involvement in transactions. The only central server is the NetCents root certificate server, which signs certificates for issuing authorities and acquirers. Fortunately, this server is not part of a payment or clearing process, and thus, high availability and fast response is not a necessity. All other parties, including issuing authorities, acquirers, arbiters and blinding sites, can be arbitrarily added in response to market forces.

In a payment transaction, the only possible single point of failure comes from issuing authority downtime. Only the funds already distributed to vendors are accessible when the issuing authority is down. Unlike Millicent, however, funds can be moved between vendors to facilitate payments even when the issuing authority is inaccessible. Online arbiters can be added to the system as needed, and the system can continue to function even with complete arbiter downtime, since a complaint is not time sensitive. Similarly, blinding sites can be added as needed. If all blinding sites are unavailable, then the scrip should be passed via the bank in order to ensure customer privacy.

5 NetCents Implementation

5.1 Implementation

A prototype implementation of the NetCents protocol was built in order to test the system security and performance. The prototype runs on Intel PCs using Microsoft software technology. Off-the-shelf products were favored in order to ease development time and to make the system less proprietary.

The Client Agent evolved from a simple iterative *ping*, to test round trip latency and server throughput, to a scripted sequence of calls, and finally to a script based analysis tool that gathered detailed statistics at specified usage loads. The scripting capability allowed the development of a set of predetermined calls to NetCents vendors, issuing authorities and acquirers, and facilitated the modeling of a functioning NetCents payment environment. Transaction system practice has taught us that maximum server throughput alone is insufficient to determine acceptable loads in lieu of latency [LZGS84]. Requests are unlikely to arrive at a constant frequency, and the system must be able to gracefully handle bursts of calls. The roundtrip latency is thus calculated as the average latency over a period of time, on requests that arrive at a specific average frequency.

The NetCents issuing authority and vendor servers are built on Windows NT server technology – Windows NT Server 4.0, Internet Information Server 3.0 (IIS), and Microsoft SQL Server 6.5. The NetCents service itself is designed as an ISAPI extension. This is specific to IIS and differs from a CGI call by the fact that an ISAPI extension is a library loaded into the same address space as the server, whereas a CGI program is a fully separate process.

The issuing authority server handles requests for initializing a customer, issuing and signing scrip, and transmitting scrip to vendors when requested. For the purposes of this prototype, all NetCents requests are conducted as extended URL calls with the EPOs appended in hexadecimal notation after the URL, and replies are read as files. The vendor was given the functionality to respond to the following requests: scrip fetch process from another vendor or a bank; scrip release to another vendor; and the payment request.

5.2 Experimental Setup

The prototype NetCents system was set up on a 10 Mbits/s Ethernet network running Windows NT networking. One bank and two vendors were set up on three Intel Pentium Pro/180 servers on Windows NT 4.0 and IIS. The prototype system was controlled by requests from one client on a Pentium 166 also running NT 4.0. Unfortunately, the network setup was not sufficiently fast to test high load server throughput, as is evident from the low server CPU utilization. Customer requests could not be initiated on the vendor server since the client application is resource intensive and would have interfered with the server performance. The SQL server was found to consume the majority of the processing time.

5.3 Experimental Results

The first experiment is a simple test to measure round-trip latencies for various NetCents system calls. Its purpose is to demonstrate the base performance of the test environment. Then, the individual NetCents services are called to test the computational cost of various functions.

Table 2 shows the round trip latencies for the tests. The three first measurements – file request, CGI call and ISAPI call – demonstrate the effectiveness of the ISAPI extensions over CGI calls. However, the low server load for file request and ISAPI calls shows that throughput was not limited by the server CPU but by the test environment, possibly the network, network

cards, or protocol processing at the client. Thus, Table 2 does not reveal effective throughput times.

	Round-trip latency (ms)	Server load
HTTP request of 256 byte file	9.9	30%
CGI request with 256 byte reply	44.6	84%
ISAPI DLL call with 256 byte reply	17.3	38%
Payment against existing scrip	18.2	34%
Scrip fetch from a bank/merchant	31.3	55%
Payment with scrip fetch	63.9	31%

Table 1: NetCents vendor server performance

Raw EPO signing and verification costs are very efficient when running on a stand-alone system. The RSA based signature scheme, with a 512-bit modulus and a public key of 3, yielded a signature speed of verification rate of 0.3 ms on a Pentium Pro/180 server. The most expensive component of vendor operation is the task of signing outgoing scrips, which was measured at 34 ms on the same server. Assuming a 90% hit rate (percentage of payments that do not require scrip transfer), a vendor, running on the same system, requires on average 3.7 ms per payment (not including communication and database overhead). Customer performed receipt signing is sufficiently fast at 180 ms on a Pentium/75.

However, when put into practice in an ISAPI extension, the overhead of the slow network, TCP/IP packet handling, the Internet server, and SQL database calls greatly effect the round-trip latency. Table 2 shows the observed latencies for a payment against an existing scrip (18.2 ms), a scrip fetch from an issuing authority or another vendor (31.3 ms), and a payment which requires a scrip to be fetched (63.9 ms). Again, the server load is far from its peak, and thus the maximum throughput is higher than the experiment would suggest.

Figure 3 depicts the observed average round-trip latency for purchases at varying server loads, and for different hit rates. The hit rate has a large impact on the server throughput, since a miss would result in a scrip fetch procedure involving three message exchanges: the first to inform the vendor from where to fetch the scrip; the second to actually fetch the scrip; and the last to pay against the transferred scrip. Furthermore, a fourth message is sent at a random time to fetch another scrip from the vendor, in order to keep the number of scrips in equilibrium, for the sake of our analysis.

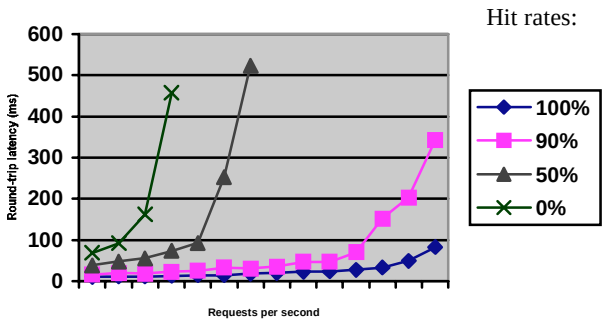

Figure 3: Average payment latency compared to the frequency of requests for various hit rates.

The graph shows the importance of modeling expected purchasing behavior in scrip relocation policies. When all payments are against scrip that reside on the vendor, then the vendor can support up to 120 payments per second (with an average latency of 371 ms) with our experimental hardware setup. For hit rates of 90% the system can comfortably handle 65 requests per second (at 200 ms). The system handles 45 and 30 payments per second for hit rates of 75% and 50% respectively. A vendor that never receives repeat buyers can only support 18 purchases per second.

6 Discussion

In this section we discuss some of the implementation issues with NetCents and put the NetCents protocol in the context of several existing payment protocols. Section 6.1 describes the integration of NetCents and SET. Section 6.2 compares the NetCents protocol with many popular protocols, including Millicent.

6.1 Integration with SET

In designing NetCents, the SET security and banking hierarchy was followed in order to allow for possible future SET interoperability. SET is an open protocol, and it can be built on top of the NetCents protocol. SET supports multiple brands, such as a VISA or Mastercard brands. NetCents can be built as another brand extension, with differences in the underlying payment mechanism. The two protocols could exist in unison, with NetCents handling all transaction of small monetary value, and SET with a credit card organization covering the higher end of the spectrum. The same public keys can be shared and the account services can be combined into one.

6.2 Putting NetCents in Context

In Section 2.1 we identified several desirable properties of electronic payment protocols. In this section, we evaluate NetCents against these properties. Table 2 presents the results of this comparison of NetCents with respect to the following properties and attributes.

We begin by evaluating the size of payments supported by a payment protocol, where *large payments* value exceed five dollars, *small* (to medium) *payments* range from 25 cents to five dollars, and *micropayments* range from 25 cents to fractions of a penny. The larger the range of payments that can be handled the more flexible, and real-world applicable, a protocol is.

The speed of operation of the protocol depends in large part on the speed of vendor validation. A protocol should support *fast validation* of payment deposits at a vendor. This in turn implies that public key message signing and decryption at the vendor should not be allowed due to their computational load. Given our implementation results, we postulate that a typical Pentium Pro 200 server should be able to validate at least 20 transactions per second.

One thing that limits the use of a protocol is its reliance on *customer specific hardware*. A payment system that requires users (customers) to have additional hardware will find limited deployment among potential customers.

An electronic payment system should also satisfy the ACID properties identified in section 2.1. Minimally, the protocol should be *money atomic*, in that a transaction either completes fully or does not complete at all. A protocol will be strengthened by being *goods atomic*, so that a purchase including both goods and money transfer will complete fully or not all. Finally, *certified delivery* will ensure that the protocol is goods atomic and allows proof of *what* was delivered. These attributes of a protocol will allow increased customer and vendor trust in the protocol itself.

Another attribute required for trust in the protocol is *protection from double spending*. This provides the vendor with additional assurance that they will be paid for purchases made. This also provides a level of assurance to the bank that the cost of fraud within the system will be minimized.

We also consider several characteristics of the implementation of the protocol. A good protocol will be *scalable* to a worldwide implementation, and *distributed*, to remove its dependence on a single central server. The payment means should be *divisible* to support multiple denominations and payment values. As with hard currency, we desire that electronic currency be *transferable* between users. The protocol itself should be *interoperable*, meaning that it supports multiple currencies and payment institutions.

Payment protocols may also ensure *partial anonymity*, where a buyer's identity is hidden from the vendor but not the bank, or *full anonymity*, where the buyer's identity cannot be associated with a purchase by either the vendor or the bank.

Finally, a protocol should implement *non-repudiation*, so that neither a vendor nor a customer can repudiate a transaction after the fact.

Property	CyberCash	NetBill	DigiCash	SET	MiniPay	Agora	PayWord	Anonymous offline	Smart Card	Millicent	**NetCents**
Large-payments	✓	✓	✓	✓					✓		✓
Small-payments	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓
Micro-payments					✓	✓	✓	✓	✓	✓	✓
Fast vendor valid'n.	✓		✓		✓		✓	✓	✓	✓	✓
No customer HW	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓
Money atomic	✓	✓		✓	✓	✓	✓	✓	✓		✓
Goods atomic		✓									✓
Certified delivery		✓									
Double spending protection	✓	✓	✓	✓					✓	✓	✓
Scalable				✓	✓		✓	✓	✓	✓	✓
Distributed					✓	✓	✓	✓	✓	✓	✓
Divisible	✓	✓	✓	✓					✓	✓	✓
Transferable			✓						✓		✓
Interoperable				✓	✓						✓
Partial anonymity	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓
Full anonymity			✓					✓	✓		✓
Non-repudiation		✓				✓			✓		✓

Table 2: Comparative analysis of payment protocol properties

7 Conclusions

We have introduced the NetCents micropayment protocol and demonstrated our claims that it is low-cost, scalable and secure. Previous payment protocols have traded off security for efficiency, introducing vulnerabilities to double spending, and lacking non-repudiation and anonymity. NetCents meets the requirements that we identified in Section 2.1, that is, it allows a full range of payment sizes, implements fast vendor validation, does not require additional hardware, is money and goods atomic, provides protection against double spending, implements non-repudiation, is scalable and distributed, supports multiple denominations of scrip and payment values, supports the transfer of funds between users, supports the use of multiple currencies and allows for partial and full anonymity of customer purchases.

With the NetCents protocol we have introduced the notion of floating funds that are passed from one vendor to another, following a customer's shopping tendencies. This distributed movement of funds removes the reliance on central servers in the course of a payment, making the protocol both fast and inexpensive. NetCents incorporates cryptographic functionality that makes the system secure against customer fraud; information in a transaction conclusively shows proof of payment. This proof can be used by an online arbiter for minor dispute resolution and for ensuring proper delivery of purchased goods. NetCents includes options for anonymous funds using blind signatures. Since the protocol treats money in bulk, NetCents offers a faster anonymous payment mechanism than individually signed coin based schemes.

NetCents' unique treatment of funds, in the form of floating scrips, enables distributed operation and payments without the involvement of a third party. The use of floating scrips has many benefits. First, the cost of handling a payment is limited to the computational load on part by the vendor. Since the bank is not required in the course of a transaction, then it need not impose a minimum per transaction cost to pay for the verification service. This enables vendors to sell wares at arbitrarily low costs ranging down to fractions of pennies. Second, by eliminating the bank from the transaction, the payment round-trip latency is reduced, and a central bottleneck is removed. Third, unlike offline payment protocols, NetCents is secure against customer fraud, and in particular, prevents

double spending of funds.

Bibliography

[Br95] S. Brands, "Proposal for an Internet cash system", Proceedings of the Internet Society; Symposium on Network and Distributed System Security, San Diego, CA, 1995.

[CHTY96] J. Camp, M. Harkavy, J. D. Tygar, B. Yee, "Anonymous Atomic Transactions", The Second USENIX Workshop on Electronic Commerce, Oakland, CA, 1996.

[CST95] J. Camp, M. Sirbu, J. D. Tygar, "Token and Notational Money in Electronic Commerce", The First USENIX Workshop on Electronic Commerce, New York, NY, 1995.

[Ch95] D. Chaum, "An Introduction to ecash", DigiCash, http://www.digicash.com, 1995.

[Ch82] D. Chaum, "Blind signatures for untraceable payments", Advances in Cryptology: Crypto '82 Proceedings. Plenum Press, 1983.

[CFN88] D. Chaum, A. Fiat, M. Naor, "Untraceable electronic cash", Advances in Cryptology: Crypto '88, Lecture Notes in Computer Science, no. 403, Springer-Verlag, 1988.

[CTS95] B. Cox, J. D. Tygar, M. Sirbu, "NetBill Security and Transaction Protocol", The First USENIX Workshop on Electronic Commerce, New York, NY, 1995.

[DEC95] Millicent, http://www.Millicent.com

[GaSi96] E. Gabber, A. Silberschatz, "Agora: A Minimal Distributed Protocol for Electronic Commerce", The Second USENIX Workshop on Electronic Commerce, Oakland, CA, 1996.

[GR93] J. Gray, A. Reuter, "Transaction Processing: Concepts and Techniques", Morgan Kaufmann Publishers, San Francisco, CA, 1993.

[HalB95] P. M. Hallman-Baker, "Micro Payment Transfer Protocol (MPTP) Version 0.1", W3C Working Draft, November 22, 1995, http://www.w3.org/pub/WWW/TR/WD-mptp.

[HSW96] R. Hauser, M. Steiner, M. Waidner, "Micro-Payments based on iKP", August 1996, http://www.zurich.ibm.com/iKP_references.html

[HeYo96] A. Herzberg, H. Yochai, "Mini-Pay: Charging per Click on the Web", http://www.ibm.net.il/ibm_il/int-lab/mpay.

[LZGS84] E. D. Lazowska, J. Zahorjan, G. S. Graham, K. C. Sevcik, "Quantitative System Performance; Computer System Analysis Using Queueing Network Models", Prentice-Hall, New Jersey, 1984.

[Man95] M. S. Manasse, "The Millicent protocol for electronic commerce," The First USENIX Workshop on Electronic Commerce, New York, NY, 1995.

[Pou97] T. J. Poutanen, "Metcents Protocol for Inexpensive Internet Payments", 1997. http://www.ee.ryerson.ca:8080/~hhinton/netcents

[Sch96] Bruce Schneier, Applied Cryptography; Second Edition", John Wiley & Sons, USA, 1996.

[RiSh96] R. L. Rivest, A. Shamir, "PayWord and MicroMint: Two simple micropayment schemes", 1996, http://theory.lcs.mit.edu/~rivest/RivestShamir-mpay.ps.

[RSA79] R. L. Rivest, A. Shamir, L.M. Adleman, "On Digital Signatures and Public Key Cryptosystems", MIT Laboratory for Computer Science, Technical Report, MIT/LCS/TR-212, Jan 1979.

[SET97] Mastercard, Visa, "SET 1.0 – Secure Electronic Transaction Specification", May 31, 1997, http://www.mastercard.com/set.html.

[Ty96] J. D. Tygar, "Atomicity in electronic commerce", Proceedings of the Fifteenth Annual ACM Symposium on Principles of Distributed Computing, pages 8-26, May 1996.

[TW96] J. D. Tygar, A. Whitten, "WWW Electronic Commerce and Java Trojan Horses", The Second USENIX Workshop on Electronic Commerce, Oakland, CA, 1996.