Electronic Commerce Research and Applications Journal, Vol. 3, No. 4, Winter 2004, pp. 381-8

PayCash: A Secure Efficient Internet Payment System

Jon M. Peha[1] and Ildar M. Khamitov
Cyphermint Inc., Marlborough, MA, USA

ABSTRACT
This paper describes PayCash, an Internet payment system that was designed to offer strong security and privacy protection. This system is based on the concept of electronic cash, extended to support a flexible anonymity policy so as to accommodate privacy and security laws that differ from nation to nation. PayCash includes novel techniques to generate trustworthy records of all transactions, making it possible to detect many forms of fraud. This system also allows users to send a variable number of "electronic coins" in a single message, so both large and small amounts of money can be transferred efficiently.

Keywords
Payment System, Electronic Cash, Privacy, Security.

1. INTRODUCTION

Despite the many inherent security risks of the Internet, it has become an essential tool for commerce and financial services. This has created a tremendous need for secure and efficient payment systems that can operate over unsecure networks. Today's payment systems routinely undermine the security and privacy of their users. Moreover, many consumers are unable to perform transactions over the Internet at all because they lack access to computer technology, suitable financial instruments, or both. This paper describes Cyphermint's novel and effective new payment system called *PayCash*, which has quickly emerged as the leading Internet payment system in five nations of eastern Europe, and has begun expansion to top e-commerce merchants in the US [4]. Its uses include business-to-consumer electronic commerce, peer-to-peer funds transfers among consumers and among businesses, and transfers from one agent of a licensed international funds transfer company to another.

PayCash uses novel algorithms to advance traditional objectives of Internet payment system design, such as security, privacy, and efficiency. More specifically, this system creates verifiable records of all transactions that cannot be forged or undetectably altered by the party sending funds, the party receiving funds, or even by the operator of the payment system. Such records are essential to protect all parties from many forms of fraud [10,11]. Moreover, this is accomplished without sacrificing privacy of either sender or receiver, and without imposing a heavy processing burden on the payment system's servers. However, advancing these traditional objectives is not enough. An effective payment system must be consistent with laws and policies of all nations where it operates, which requires that some flexibility on issues of privacy and security be built into the technology. Not only do the laws vary from nation to nation, but in the US, policies have changed to address new security concerns in the wake of the September 11, 2001 attacks. The PayCash design has evolved accordingly.

Section 2 briefly addresses the state of payment systems today. Section 3 discusses the design objectives for a new payment system. Section 4 presents an overview of the electronic cash approach originally proposed by Chaum [2]. Section 5 presents Paycash, which builds on the electronic cash concept, with significant extensions to achieve the design objectives from Section 3. Finally, the paper is summarized in Section 6.

2. THE STATUS QUO

Today, many financial transactions use mechanisms that offer little security or privacy protection, such as credit cards or simple password schemes. Most on-line purchases use credit cards. In the process, consumers often reveal credit card numbers and

[1] Corresponding author: Chief Technical Officer at Cyphermint, Inc. Professor at Carnegie Mellon University (CMU), and Associated Director of the CMU Center for Wireless and Broadband Networks.

personal information to unknown merchants, and often to anybody who cares enough to watch the traffic pass from consumer to merchant over the Internet or through an exposed wireless connection. Anyone observing credit card information can use it to make additional purchases. It is no wonder that fraud and identity theft are rising at a tremendous rate [5]. Even if they are not victims of fraud or theft, consumers who reveal personal information compromise their own privacy, and may be rewarded with an avalanche of spam and telemarketer calls. In addition, many banks, merchants, and payment systems allow their customers to log in over the Internet to access personal information and initiate financial transactions. Such sites are often "protected" with passwords. Thieves can access a significant fraction of these sites using password-guessing software that is readily available over the Internet.

Security problems aside, many consumers cannot enjoy the e-commerce opportunities because they have no credit cards. Transaction costs are also an issue. For example, the market for inexpensive digital products, such as individual magazine articles or digitized songs, has been slow to emerge in part because the cost of transferring a payment can exceed the cost of the product itself. International funds transfers are particularly expensive, as anyone who has made a wire transfer knows. Most international money transfer companies have not yet reaped the benefits of secure Internet payment systems.

3. DESIGN GOALS FOR AN EFFECTIVE PAYMENT SYSTEM

To protect security and privacy, PayCash was designed to achieve the following.

- **Tamper-proof records**: As described in Section 1, every financial transaction must produce a record that cannot be undetectably altered by sender, receiver, or operators of the payment system. In Paycash, digitally signed records are a byproduct of transactions, so trust among these parties is not required.
- **Privacy Protection**: To protect privacy and combat identity theft in e-commerce, consumers must be able to send funds without revealing any personal information to the recipient, and receive funds without revealing information (other than an account number) to the sender. They reveal only what they choose to reveal.
- **Flexible anonymity policies**: In countries where privacy is greatly valued, such as Russia, Paycash users demand the ability to send and receive money without revealing personal information to anyone, including the operator of the payment system. In other countries, this level of anonymity is unacceptable, because it prevents law enforcement agents from observing transactions that might be linked to crime or terrorism. The US moved decisively into the latter camp after the attacks of September 11, 2001, when the US government began requiring more companies to monitor financial transactions and report suspicious behavior to government authorities. It would be inconvenient to deploy different systems in different countries, and painful to completely change systems every time a nation changes its policy. To succeed in the global Internet, the payment system must offer users the level of privacy and anonymity that is currently appropriate in their country, whatever that might be.
- **Protection from password guessing**: To send or receive money, a PayCash user must have software known as a *wallet,* which manages the user's encryption keys. Users can place their wallet on their own computer, so it is more difficult for thieves to break in by guessing passwords over the Internet. In this configuration, most password-guessing schemes require physical access to the user's computer. This safe option is not available with many payment systems.
- **Protection from outside observers**: Because it is easy to observe traffic over the Internet and many wireless networks, all messages must be encrypted.

To support a wide variety of uses, PayCash was designed to achieve the following.

- **Support for disconnected users**: There are cases where the sender and receiver of a payment are not both connected to the Internet, at least not at the same time. For example, a consumer may be connected to a merchant through a wireless local-area network, but the consumer has no direct Internet connection. Unlike many payment systems, Paycash is designed to work if the device sending funds can connect with the recipient through any communications link, or the sender can connect with an agent of the payment system called a *Payment Authorizer* that operates on the Internet. (In this paper, we focus on the former case, which is shown in Figure 1.) Both connections are not required. As a result, 802.11-equipped laptops can use PayCash to pay for Internet access in commercial 802.11 LANs, and transmitters can use PayCash to pay for access to licensed spectrum through a real-time secondary market [12] or a band manager [13,14].
- **Wide range of payments**: To support the sale of inexpensive digitized products, the system should even handle payments of less than a cent.
- **Multiple currencies**: The system must handle multiple currencies. Some of those currencies will be created for a specific merchant or for groups of merchants to support a loyalty program, like those developed by airlines for frequent flyers. PayCash currently supports four billion currencies, and the ability to limit who can use a given currency and how.
- **Scalability**: The system must scale easily to a large number of users, while maintaining a low cost per transaction.

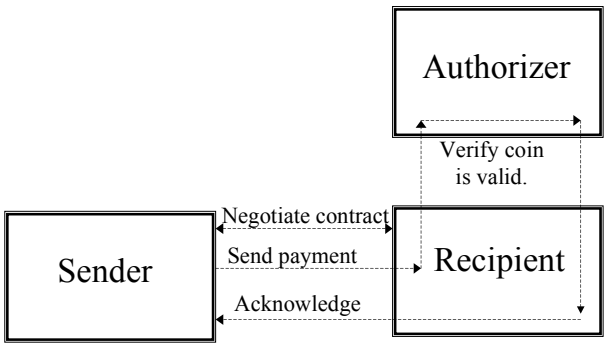

Figure 1: Payment block diagram

4. THE SUITABILITY OF CHAUM'S ELECTRONIC COINS

As described in the previous section, the PayCash system is designed to provide a user with a level of privacy and anonymity that is appropriate to the country where that user resides. When a user adds or removes money from a PayCash account, she usually reveals personal information. For example, if the user requests that a check be mailed out, or that funds be wired to her account in a real bank, she must specify the name on the check or the name associated with the bank account, respectively. Nevertheless, even if the payment system operator can associate one name with account A and another name with account B, the payment system does not necessarily know whether funds were transferred from A to B. Thus, for users in nations where the policy favors anonymity, the technical challenge is to prevent the operator of the system from identifying both parties in any funds transfer. In nations like the US where transfers must be monitored, the opposite is true; such information must be captured and analyzed.

Chaum [2] was the first to demonstrate how anonymity could be supported in a payment system by using *electronic coins*: digital strings that can be transferred anonymously from person to person just like cash. The payment system facilitates the transfers, and makes it impossible for users to counterfeit coins, but the payment system never knows who owns a digital coin until an owner wants to redeem that electronic coin for real cash. We review this scheme in this section, and borrow ideas from it in the next.

In this system, a coin with serial number *X* is defined by $\{ X, g^{-1}(f(X)) \}$, where f(.) and g(.) are functions that are easy to calculate and hard to invert. Anyone can check whether a coin {X,Y} is valid by determining whether f(X)=g(Y). Only payment system's agent (which we call the *Payment Authorizer)* can "mint" a coin because only this agent can apply the function $g^{-1}(.)$, which is the inverse of g(.). (No one can invert f.) To preserve anonymity, the agent must mint the coin with serial number X without learning X or f(X). To accomplish this, the user applies a blinding function such as [3,9] before requesting that the agent apply the $g^{-1}(.)$ function. The user unblinds the result, and produces the coin $\{ X, g^{-1}(f(X)) \}$. The payment system's agent does not know f(X), because it never saw f(X) without the blinding function. The agent deducts enough money from the user's account to pay for this newly minted coin.

No one can counterfeit a coin in Chaum's system without learning to invert g(.), but additional protection is needed to prevent a user from spending the same coin multiple times. The payment system's agent must record the serial numbers of all coins that have been spent. The recipient of a coin typically checks immediately with the agent to see if that coin has already been spent before accepting the coin. The agent checks by searching this list. If the serial number is not already in the list it is added, thereby invalidating the coin, and a new coin is minted for this user, or the coin's value is added to the recipient's account which is maintained by the payment system agent.

Chaum's approach has important merits, although it does not meet all the objectives described in Section 3. Beginning with the positive points, the information flow in a funds transfer is exactly that shown in Figure 1: the coin travels from sender to receiver, and the receiver contacts the payment system's Authorizer to make sure the coin has not already been spent. The sender need not communicate directly with the Authorizer, so the scheme is suitable for disconnected users as described in Section 3. Furthermore, there is nothing in this system that forces sender and receiver to reveal any information to each other, wallets can run locally at the sender and receiver's computers to combat password guessing, and all messages can be encrypted. Multiple currencies can be supported by using different functions f(.) and g(.) for each currency.

Chaum's scheme provides strong anonymity for all. Although flexible anonymity policies are not supported, it would be easy to relax anonymity for some users and not for all, simply by disabling the blinding mechanism for users that should not have full anonymity, and recording serial numbers when those users mint new coins. When those coins are redeemed, the agent can observe the details of the transfer. Alternatively, an intermediate level of anonymity could be supported if the keys to this blinding function were held in escrow where they can be retrieved [6].

A serious limitation of this scheme is the absence of tamper-proof transaction records. If there is a dispute, the sender cannot prove that he transferred funds to the recipient, and he certainly cannot prove that it was part of another transaction, such as an e-commerce purchase. The system also provides no way to resolve disputes between users and the payment system operator. For example, the payment system's agent may claim that a coin has already been spent and reject it when the coin has not been spent, or a user may spend a coin twice and deny it. There are no records to reliably determine who is right.

Supporting a wide range of payments is also problematic. If all coins represent a value of one cent, then transfers of many thousands of dollars could be impractical. A typical solution is to create coins of large and small denominations. As a result, a recipient of funds may have to make change, which complicates a transfer, and then both sender and recipient need the ability to contact the agent.

One final limitation of this scheme is that a list of all spent coins must be maintained, and frequently searched. The list can grow large. To prevent the list from growing without bound, an expiration date must be added to coins so that spent coins that have expired can eventually be removed from the list, but this means that coins belonging to users also expire, which is inconvenient. It would help if the list of expired coins grew more slowly.

5. THE PAYCASH APPROACH

5.1 Producing Tamper-Proof Records

Like Chaum's electronic coins, the PayCash system is based on the electronic currency concept. The first innovation of the system is to digitally sign all transaction records, and to integrate this signature into the payment system itself to create tamper-proof records. Instead of an arbitrary serial number X, the customer generates a pair of public and private keys, P and S, which will be used for this signature. Let Sign(S,X) be the digital signature function that uses the private key S, and Verify(P,X) be its easy-to-calculate inverse that uses public key P, so Verify(P,Sign(S,X))=X.

Similar to the Chaum scheme, a coin is { P, $g^{-1}(f(P))$ }, where P is both serial number and public key. To send one coin, the user transfers the four-tuple

{*record*, Sign(S,*record*), P, $g^{-1}(f(P))$ },

where *record* is a description of the transaction, including recipient of the funds, timestamp, and any other information needed for a contract between sender and receiver, or at least a hash of such information. The payment {A, B, C, D} is valid if the following three conditions are met.

1. a payment has not already been made with serial number C,
2. the coin has been properly minted with the g(.) function, i.e. f(C)=g(D),
3. the digital signature is correct, i.e. Verify(C, B) = A, and
4. the recipient of the funds transfer corresponds with the one listed in *record* A.

The first two conditions are analogous to the Chaum scheme, and the latter two are new. This third condition proves that the creator of the payment four-tuple knows the secret key S, so it authenticates the sender.

The extra signature provides some added security. Chaum's scheme can be broken if an inverse $f^{-1}(.)$ can be found to f(), because { $f^{-1}(g(X))$, X }would be accepted as a valid coin for any value of X. With PayCash, even if someone can somehow invert f(), they must still find the secret key that would correspond to a public key of $f^{-1}(g(X))$.

More importantly, thanks to the third and fourth conditions, any attempt to spend the same serial number P more than once will leave clear evidence. Consider the case where a user makes two payments to two different recipients with the same P. If the *record* fields are identical in both cases, then it is easy to demonstrate that condition 4 fails for at least one of the recipients. If the *record* fields are not identical, then the payment system's agent can produce two dissimilar payments with the same serial number P. The agent could not have faked these payment records, because only the sender has the secret key S needed to produce both digital signatures.

With the addition of one more step, we can also address the problem of settling disputes between sender and recipient. The payment already includes a transaction record that has been digitally signed by the sender. If the important fields within *record* were signed by the recipient before it was signed by the sender, then neither party could undetectably alter a transaction record. This leads us to the protocol described below. For example, it is used to create a "contract" between consumer and merchant in a typical e-commerce transaction.

1. Consumer sends information to merchant to be placed in contract.
2. Merchant composes contract, digitally signs it, sends result back to consumer.
3. Consumer includes a hash of the signed contract in *record*, constructs payment as described above, and sends it to merchant.
4. Merchant sends message to the Payment Authorizer to make sure the payment is valid.
5. Payment authorizer checks the signature, makes sure that the serial number has not been spent already, updates records, and informs the merchant that the payment succeeded.
6. The merchant informs the consumer that the payment succeeded.

5.2 Making Payments of Different Amounts

Alas, not all payments are exactly one coin. Another important property of the PayCash system is that a payment of any amount can be made without sending multiple coins, and without requiring change. For each serial number P, the payment system agent keeps track of the total amount of money m(P) that has been spent so far. A user can spend k coins of value c simultaneously simply by proving that the number N of coins that he has received so far (including those already spent) is large enough that he has at least k left, i.e. $N \geq k + m(P)/c$. The payment system agent can then update m(P) to reflect the money that has been spent. All the user needs for this to work is an efficient method of demonstrating N.

This is achieved in part by allowing the same serial number to be "minted" with $g^{-1}(.)$ multiple times, similar to the hash chain approach [15,7,1,8], thereby putting the value of multiple "coins" in a single data structure. Instead of the single coin of the form

{P, $g^{-1}(f(P))$ }, we define a PayBook(N,P) of N coins associated with serial number P in the following structure:

Paybook(N,P) = {N, P, $g^{-N}(f(P))$ },

where N is a non-negative integer, $g^{-0}(X) = X$, and $g^{-N}(X) = g^{-1}(g^{-(N-1)}(X))$ for any integer N: N>0. This has several advantages. First, as described in Section 4, a list of serial

numbers associated with the spent coins must be maintained and searched regularly. Minting the coin multiple times on the same serial number greatly reduces the size of that list, so it is not necessary to take coins out of circulation so often. Second, this eliminates the need to generate a public/private key pay for each coin. Third, it greatly reduces the size of multi-coin payment messages.

Any customer can create a paybook with no funds (i.e. N=0) without help from the payment system. An empty paybook is simply {0, P, f(P) } for some P. Moreover, if a user has a PayBook with N coins P(N,P) = {N, P, Z}, it is easy to generate a Paybook with less money, such as {N-1, P, g(Z)} which has N-1 coins. However, adding a coin to produce {N+1, P, $g^{-1}(Z)$} is impossible without the help of the payment system agent, because only that agent can apply the minting function $g^{-1}(.)$ to $g^{-N}(f(P))$. As described in Section 4, this can be done with or without a blinding function, depending on whether anonymity is supported for this customer.

A payment would work as follows. Consider a user with a paybook containing N coins, i.e. N coins have previously been deposited. He wants to make a payment of amount q, and has previously spent m from this paybook, where each coin is worth c. He will prove to the payment system that at least n coins have been deposited, where $(q+m)/c \le n \le N$. As shown above, from the paybook with N coins, it is trivial to construct a paybook(n, P) with just n coins. The user then makes a multi-coin payment with the following set:

{*record*, Sign(S,*record*), PayBook(n,P) } =

{*record*, Sign(S,*record*), n, P, $g^{-n}(f(P))$ }

where the transaction *record* includes the amount q of the payment.

A payment {*record*, *sign*, n, P, Y} of amount q is valid if the following conditions are met.

1. The Payment Authorizer verifies that the paybook is valid, i.e. $f(P) = g^n(Y)$. If this condition is not met, or if the paybook is empty (n=0), then the payment is rejected.
2. The payment Authorizer verifies that the digital signature is correct, i.e. Verify(P, *sign*) = *record*. If not, the payment is rejected.
3. The Payment Authorizer checks its table to determine the amount of money m(P) associated with this paybook that has already been spent. If no paybook has been seen before with serial number P, then a new one is created with m(P)=0.
4. If there are insufficient funds, i.e. $nc < q+m(P)$, then the payment is rejected. Otherwise, the payment is authorized, and m(P) is increased by q.

A mechanism like this is useful when transferring a large number of coins. For example, where Chaum's scheme would require a user to send 1000 coins, and ultimately add 1000 serial numbers to that list of used coins, PayCash achieves the same thing in one simple message. Such a mechanism is also useful when transferring a fraction of a coin. If a user can demonstrate that he has 5 coins, the system can easily allow him to spend 4.5 coins, and adjust the amount spent m(P) accordingly. Our PayCash implementation can support payments of a hundredth of a cent, even though deposits and withdrawals must be an integral number of cents.

The PayCash wallet software allows a user to create multiple PayBooks. Thus, a user who wants to make two purchases from the same merchant, without revealing any connection between these purchases, can easily do so from separate paybooks.

6. SUMMARY

Millions of people enjoy the convenience of transferring money and shopping on the Internet, but at great risk. Privacy goes unprotected. Personal information obtained on the Internet facilitates identify theft. Fraud is common; many transactions generate no credible records that can be used to resolve disputes. Some systems are vulnerable to password-guessing attacks launched from across the Internet. Effective methods are needed to protect the privacy and security of users. We have presented the design of a new Internet payment system called Paycash that meets these needs.

The security problem is even more challenging because a strategy that is effective in one country may be inappropriate or even illegal in another. In some countries, it is essential to protect anonymity, whereas in countries like the United States (after September 11, 2001), complete anonymity is inappropriate, and an effective payment system must allow authorized law enforcement agencies to monitor suspicious activity. (Even where users are not allowed to hide their identity from the payment system, they should still be able to hide identity from each other.) The PayCash system provides protection that can be tailored to fit different national policies.

PayCash is based on the concept of electronic currency. However, unlike competing systems, PayCash produces credible records of all transactions to deter fraud and resolve disputes. This is accomplished by requiring users to digitally sign transaction records, and by integrating these signatures into the payment system itself. PayCash also supports transfers that are equivalent to many electronic "coins" through use of paybooks. This greatly decreases transactions costs, and allows the system to efficiently support a wide range of payments.

An important benefit of Internet payment systems like PayCash is that they make electronic commerce accessible to people who do not have credit cards. Of course, there are also consumers without easy access to computers. To bring e-commerce to individuals who do not own computers or do not know how to use computers, the next challenge was to create a wallet that was specifically designed to run on a publicly-accessible user-friendly kiosk, which might look similar to today's ATM machines. These kiosks also make it easy to deposit money into a PayCash account, and to shop on line using cash as well as credit cards. In 2003, this software was deployed in one thousand kiosks in convenience stores across the US, and more will be deployed in 2004. Future publications will describe the formidable technical challenges of designing these kiosk systems so that they are secure enough to handle large amounts of cash, manageable enough to operate with no on-site support staff whatsoever, and efficient enough to operative effectively even over low-bandwidth connections.

7. REFERENCES

[1] Anderson, R. Manifavas, C. and Sutherland, C. "NetCard - A Practical Electronic Cash System," *Fourth Cambridge Workshop on Security Protocols*, April 1996.

[2] Chaum, D., Fiat, A. and Naor, N. "Transaction Systems To Make Big Brother Obsolete," *Communications of the ACM,* Vol. 28, No. 5, Oct. 1985, pp. 1030-44.

[3] Chaum, D. "Blinding for Unanticipated Signatures," *Advances in Cryptology EUROCRYPT '87*, Springer-Verlag, pp. 227-33.

[4] Cyphermint merchants, www.cyphermint.com/support/frameset2/index2.htm

[5] Federal Trade Commission, "On Identity Theft: The FTC's Response," Howard Beales, Before the Subcommittee On Technology, Terrorism and Government Information of the Judiciary Committee, United States Senate, March 20, 2002.

[6] Gemmell, P. S. "Traceable E-cash," *IEEE Spectrum*, Vol. 34, No. 2, Feb. 1997, pp. 35-37.

[7] Hauser, R., Steiner, M. and Waidner, M. "Micro-Payments based on *i*Kp," *Research Report 2791*, IBM Research, Feb. 1996.

[8] Pedersen, T. P. "Electronic Payments of Small Amounts," *Fourth Cambridge Workshop on Security Protocols*, April 1996, pp. 59-68.

[9] Khamitov, I. M., Moshonkin, A. G. and Smirnov, A. L. "Blind Unanticipated RSA-Signature Schemes," *Cyphermint White Paper*, www.cyphermint.com

[10] Peha, J. M. "Making the Internet Fit for Commerce: new policies to enforce tax laws, protect privacy, deter fraud, and prevent illegal sales," *Issues in Science and Technology,* National Academy Press, Winter 1999-2000, pp. 72-9.[1]

[11] Peha, J. M. "Making Electronic Transactions Auditable and Private," *Proceedings of Internet Society (ISOC) INET '99*, June 1999.[1]

[12] Peha, J. M. and Panichpapiboon, S. "Real-Time Secondary Markets for Spectrum," *Proceedings of 31st Telecommunications Policy Research Conference* (TPRC), Sept. 2003.[1]

[13] Peha, J. M. "Spectrum Management Policy Options," *IEEE Communications Surveys*, Vol. 1, No. 1, Fourth Quarter 1998.[1]

[14] Peha, J. M. "Wireless Communications and Coexistence for Smart Environments," *IEEE Personal Communications*, Vol. 7, No. 5, Oct. 2000, pp. 66-68.[1]

[15] Rivest, R. L. and Shamir, A. "Payword and Micromint: Two Simple Micropayment Schemes," *CryptoBytes*, vol. 2, no. 1, pp. 7-11, 1996.

[1] Available at www.ece.cmu.edu/~peha/papers.html