6.857 Computer and Network Security October 23, 2001

Lecture Notes 13 : Payment Schemes (E-Cash)

Lecturer: Ron Rivest *Scribe: Nelson/Newsom/Okunseinde/Walker*

1 Lecture Outline

1. Properties
2. Anonymous Cash (Chaum)
3. Detecting Double-Spending
4. Micropayments

2 Properties

To purchase items over the Internet, people currently use credit cards as the prevailing form of payment. For years, however, people have asked "Why not use Electronic Cash?" These are the properties that would be necessary for such a scheme:

1. Financial Infrastructure: There is a big difference between bits and atoms. People have used atoms, gold, bills, etc. as money. Behind the bits in E-cash, there must be financial infrastructure that the money represented by the instructions in the bits from one account to another. A transaction is an instruction to move money from a consumer's account to a merchant's account.
2. No Double-Spending and Non-forgeability: Bits can be easily duplicated but atoms cannot. So copies of cash should not be spendable. Nor should one be able to forge or create e-cash and spend it.
3. Security: Account information should be kept secure. Transfers should be kept secure.
4. Immediate Verifiability that Payment is OK Online vs. Offline systems: Every time you receive a payment, you could instantly relay it to the bank to verify it. Or there could be an intrinsic property that lets you know the money is good if a bank is not readily available.
5. Persistence: Atoms stick around better than bits do. If your computer crashes, you should not be bankrupted. A backup of your wallet to record your wealth should not be spendable.
6. Exclusive Ownership

7. Anonymity: There are different types of anonymity, payer, payee, and even bank anonymity. The merchant may accept money without knowing who the payer is. Also you should be able to deposit money without the bank knowing where the transaction comes from. There are issues of money laundering. People can transfer money without the bank knowing.

8. Transferability: `A` can pass money to `B` and then to `C` easily and anonymously.

9. Amounts: It would be nice to support a variety of "coin sizes".

10. Traceable to issuer: We should know who backs each bit of money. E.g., We can tell by inspection that U.S. money is issued by the U.S. treasury.

11. Divisibility and Combination: If you have an instrument worth 1 dollar, you should be able to divide it into two instruments each worth 50 cents.

12. Compatibility with existing systems: An electronic payment system should interface smoothly with existing payment systems. (To what extent a monetary system actually depends on others is an interesting open question; can you have e.g., a Galactic System with no central Government authority?)

13. Efficient for small amounts

14. Scalability

15. Competition between Issuers: Free banking before the Gold Standard

2.1 Types of Systems

1. Checks: This is the simplest scheme. Each check is numbered to prevent duplication.

   Q: With anonymous checks, how do we ensure they do not bounce?
   A: For each check, you need a key that the bank does not recognize. You could use this key to do blinded withdrawal of money from the account associated with that key.

2. Credit Cards:

   This is the most popular payment method on the Web. It has very dangerous security implications because a single number gives access to an account. However, the credit card companies have taken this risk to be competitive and numerous protocols have been developed (e.g., SSL) to help secure this medium.

3. Anonymous Cash (see Chaum in lecture 8): Use a blinded signature process to make withdrawals. that way the bank does not know what it is signing. Each merchant can check the bank signature upon receipt of payment and deposit in the bank that confirms its own signature. Double spending is a severe problem in this scheme. The Bank could refuse the deposit and the merchant would stop accepting the form of payment.
   Q: What kind of anonymity do we want?
   A: Good point. We could remove the anonymity of each coin so that the User is anonymous to the Merchant but not anonymous to the Bank.

   Q: Wouldn't that destroy our "Transferability" property?
   A: Yes. But most schemes do not support transferability anyway.

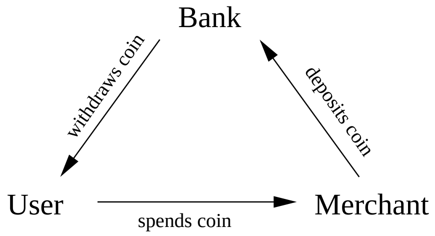

Figure 1: The circulation of an electronic coin.

2.2 Detection and Prevention of Double-Spending

Electronic cash is based on bits and bits can be easily copied. How can we let people own the bits but prevent them from being able to copy them? A simple solution might be to store the bits in a smartcard that resists tampering. Many prevention schemes involve physical devices that the user cannot control.

Q: Are there any examples of hardware that is widely distributed and truly physically secure? More important, if one instance of the hardware is broken, does it break the entire system?
A: Consider Set top boxes. Also, note that there are business models that require these types of devices where users should not be able to crack them. Whether the hardware meets those requirements is based on the situation. These are implementation problems that involve designs predicated on the belief that the users cannot completely control the operation of the hardware.

Q: Is this analogous to Intellectual Property?
A: Certainly, if one rents a movie, they presume you will not copy it and give it to your friends. This is a question of controlled usage. You want to design hardware that helps the distributor enforce these rules.

Q: Does the bank have to know about every piece of money it issues?
A: Yes, another scheme is for the bank to have databases of coins. They then cross off each coin from a "list" as it's spent. This is a non-trivial cost of doing business.

For detection, we need a scheme that reveals the user when they attempt double-spending. Each user is anonymous as long as they do not double-spend. This solution eliminates the possibility for transferability and also assumes honest merchants that do not try and cheat users by claiming their money was double spent. The result is a scheme where a bank can figure out who a user is with two instances of the same coin. Obviously, these coins cannot be copies of each other. Instead, we use a challenge-response protocol that forces the user to reveal some information. Two responses, from two different challenges, result in information that can identify the user.

Q: Is detection a viable deterrent?
A: Maybe, maybe not.

The following is Stefan Brands' scheme to detect Double-Spending where two spendings give away a spender's identity:

Public parameters published by bank:

1. large prime p
2. $q|(p-1)$
3. generators g_1, g_2, h of order q
4. Public Key: $g_1^x, g_2^x, h = g^x$ where x is the bank's secret key

1. User Secret: u_1
2. User ID $= I = g_1^{u_1} \pmod{p}$

Let $Z = I * {g_2}^x$ where $x =$ bank's secret.

We claim that the following payment protocol reveals the identity of anyone who double spends a coin.

Payer		Merchant
chooses random s		
A $= I * {g_2}^s = {g_1^{u_1}}^s g_2^s$	$\xrightarrow{coin=A,B,sign(A,B)}$	
B $= g_1^{x_1} g_2^{x_2}$		
	$\xleftarrow{d=\texttt{random challenge}}$	
$r_1 = du_1 s + x_1$	$\xrightarrow{r_1, r_2}$	$g_1^{r_1} {g_2}^{r_2} \stackrel{?}{=} A^d B$
$r_2 = du_2 s + x_2$		

To see that the user's identity is revealed if she answers two challenges: Suppose that a second challenge/response is also performed, giving the corresponding variables (using primes to denote them). Then

$$r_1' = d'(u_1 s) + x_1 \tag{1}$$
$$r_2' = d'(s) + x_2 \tag{2}$$
$$\frac{r_1 - r_1'}{r_2 - r_2'} = \frac{u_1 s(d - d')}{s(d - d')} = u_1 \tag{3}$$
$$g_1^{u_1} = I \tag{4}$$
$$\tag{5}$$

Q: Should we put an expiration on each coin?
A: Yes, this puts a limit on our storage space. Coins expire and are recredited to the bank.

Q: Should we have money that can only be spent at certain sites?
A: Probably not because that would destroy anonymity. This always reduces the convenience of E-cash.

The withdrawal protocol is a modification of the blind signature protocol so that when a user requests a signature, they embed their identity in the data. The bank can still sign it blindly, but the protocol needs to assure the bank that the identity is indeed buried in the coin.

2.3 MicroPayments

Is it possible to economically support a small payment scheme? Consider the instance of a Web site which charges perhaps 2 cents to serve each page. Is there a scheme that supports this with a low overhead cost?

Payword

This scheme uses hash chains to make micropayments cheaper.

$$\overset{x_4}{\bullet} \xrightarrow{h} \overset{x_3}{\bullet} \xrightarrow{h} \overset{x_2}{\bullet} \xrightarrow{h} \overset{x_1}{\bullet} \xrightarrow{h} \overset{x_0}{\bullet}$$

The Payer constructs a stack of coins by taking the hash of the first coin, then the hash of this value and so on. The Payer sends the merchant a stack with the base (x_0) being the first value the merchant can access. The payer signs the entire stack as opposed to each coin. Then, with each successive penny spent, the user reveals the previous element of the chain (which can be verified easily by the merchant, by hashing to get the previously received element). This scheme presumes that the payer spends a fair amount at each merchant.

Lottery Tickets

In this scheme, the Payer gives the merchant a check with the stipulation that it is only good with the probability of a certain event.

For Example:

- One penny = 10 dollars with a probability of a 1/1000
- Alice sends Bob a message "Pay Bob 10 dollars if the last 10 bits of the x hashing to $H(x)$ are 0110101101" where $H(x)$ is a value Bob gave to Alice.

This scheme works for a very large number of transactions, in the sense that everyone receives approximately what they expect to receive.