Digital Cash

Cryptography – CS 507
Erkay Savas
Sabanci University
erkays@sabanciuniv.edu
Esoteric Application

- Secure Elections
- Secure multi-party computations
- Anonymous Message Broadcast
- Digital Cash (Digital Wallet)
- Electronic Notary
- Data (content) distribution
  - Conditional access TV
  - Software distribution via CD-ROM
  - Information Bulletin Board
Digital Cash

- We do not want to carry cash
- It spreads germs
- It is not secure to carry it
- Credit cards, personal checks, and money orders violate the anonymity of financial transactions
- Anonymity in financial transactions is essential to protect privacy
- Digital cash protocols meet this social need with complicated protocols employing cryptography
6 Commandments of Digital Cash

1. The cash can be sent securely through computer networks
2. The cash cannot be copied and reused
3. The spender of the cash can remain anonymous
   • If the cash is spent legitimately, neither the recipient nor the central bank can identify the spender
4. The transaction can be done offline
   • No communication with the central bank
5. The cash can be transferred to others
6. A piece of cash can be divided into smaller amounts
Participants

- The Bank
- The Spender (Alice)
- The Merchant
A basic scheme: Protocol #1

1. Alice prepares 100 anonymous money orders for $1000 each.

2. Alice puts each of the money order and a piece of carbon paper into 100 different envelopes. She gives them to Bank.

3. The bank randomly chooses 99 envelopes, opens them and confirms that each money order is legitimate.

4. The bank signs the one remaining unopened envelope. The signature goes through the carbon paper to the money order. Alice is given this money order and $1000 is deducted from her account.
A basic scheme: Protocol #1

5. Alice opens the envelope and spends the money with a merchant.

6. The merchant checks for the bank’s signature to make sure the order is legitimate.

7. The merchant takes the money order to the bank.

8. The bank verifies its signature and credits $1000 to the merchant’s account.
The Protocol #1: Security

- Alice remains anonymous
- Alice has 1% (1 in 100) chance of cheating the bank
- The penalties are severe, so Alice thinks that it is not worth to try cheating
- Alice or the merchant can photocopy the money order and spend it twice
- This is what is called **double spending problem**
Protocol #2: no double spending

1. Alice prepares 100 anonymous money orders for $1000 each. **On each money order she also includes a different random uniqueness string.**

2. Alice puts each of the money order and a piece of carbon paper into 100 different envelopes. She gives them to the bank.

3. The bank randomly chooses 99 envelopes, opens them and confirms that each money order is legitimate.
Protocol #2:

4. The bank signs the one remaining unopened envelope. The signature goes through the carbon paper to the money order. Alice is given unopened envelope and $1000 is deducted from her account.

5. Alice opens the envelope and spends the money with a merchant.

6. The merchant checks for the bank’s signature to make sure the order is legitimate.

7. The merchant takes the money order to the bank.
Protocol #2:

8. The bank verifies its signature and **checks its database** to make sure a money order with the same uniqueness string has not been previously deposited. If it hasn’t, it credits $1000 to the merchant’s account.

9. The bank records the uniqueness string in the database.

10. If it has been previously deposited, the bank won’t accept the money.
The Protocol #2: Security

- No double spending.
- The bank cannot identify the cheater: Alice or the merchant?
Protocol #3: Tracing The Cheater

1. Alice prepares 100 anonymous money orders for $1000 each. On each money order she also includes a different random *uniqueness string.*

2. Alice puts each of the money order and a piece of carbon paper into 100 different envelopes. She gives them to the bank.

3. The bank randomly chooses 99 envelopes, opens them and confirms that each money order is legitimate.
Protocol #3: Tracing The Cheater

4. The bank signs the one remaining unopened envelope. The signature goes through the carbon paper to the money order. Alice is given unopened envelope and $1000 is deducted from her account.

5. Alice opens the envelope and spends the money with a merchant.

6. The merchant checks for the bank’s signature to make sure the order is legitimate.

7. The merchant also asks Alice to write a random identity string on the money order.

8. Alice complies.
Protocol #3: Tracing The Cheater

9. The merchant takes the money order to the bank.

10. The bank verifies its signature and checks its database to make sure a money order with the same uniqueness string has not been previously deposited. If it hasn’t, it credits $1000 to the merchant’s account. The bank records the uniqueness string in the database.

11. If the uniqueness string is in the database, the bank won’t accept the money order.
Protocol #3: Tracing The Cheater

12. Then, the bank compares the identity strings on the money order with the one stored in the database.
   1. If they are different, Alice is the cheater
   2. If they are the same, the merchant is the cheater

- We assume that the merchant cannot change the identity string on the money order
- The merchant needs to keep a database in order to prevent Alice giving the same identity string twice
How to do it with cryptography

- Recall the blind signatures with RSA
- Bob wants Alice to sign a message $m$, without her knowing the contents of the message.
- Alice’s public key is $(e, n)$.
- **Blinding the message**: Bob chooses a random integer $k < n$ with $\gcd(k, n) = 1$ and computes $t \equiv k^e m \pmod{n}$. He sends $t$ to Alice.
- Alice signs $t$ by computing $s \equiv t^d \pmod{n}$. She returns $s$ to Bob.
Blind Signatures

- *Unblinding*: Bob computes $s/k \pmod{n}$. This is the signed message $m^d \pmod{n}$ since

$$s / k \equiv t^d / k \equiv (k^e m)^d / k \equiv k^{ed} m^d / k \equiv m^d \pmod{n}.$$ 

- $k$ is a random integer, so is $k^e \pmod{n}$. Therefore, $k^e m \pmod{n}$ gives no information about the message $m$. Alice knows nothing about what she is signing.
Digital Cash: Cryptographic Protocol

- Alice prepares $n$ anonymous money order for a given amount
- Each money order, for example, look like this:
  
  \[
  \{ \\
  \text{Amount} \\
  \text{Uniqueness String:} \quad X \\
  \text{Identity Strings:} \\
  \quad I_1 = (I_{1L}, I_{1R}) \\
  \quad I_2 = (I_{2L}, I_{2R}) \\
  \quad \vdots \\
  \quad I_n = (I_{nL}, I_{nR})
  \}
  \]
Identity Strings

- Alice creates a string that contains, for example, her name, address, any piece of identifying information that the bank wants to see.
- Alice splits it into two pieces using $n$ different ways; hence coming up with $n$ pairs of identity strings, i.e. $(I_{1L}, I_{1R}), (I_{2L}, I_{2R}), \ldots, (I_{nL}, I_{nR})$
- It is impossible to extract identity string from one half, or halves of different splittings.
- Alice bit-commits to these strings. When asked, she opens and the bank can easily verifies.
Digital cash: blinding and unblinding

- Alice blinds all $n$ money orders, using a blind signature protocol. She gives them all to the bank.
- The bank picks up $n-1$ money orders at random and asks Alice to unblind them.
- Alice complies; the bank checks the amount, the uniqueness string, and asks Alice to reveal all of the identity strings.
- If the banks sees everything is in order, it signs the one remaining blinded money order. It hands it with signature to Alice and deducts the amount from her account.
Digital cash: Spending it

- Alice unblinds the money order and spends it with a merchant.
- The merchant verifies the bank’s signature to make sure the money order is legitimate.
- The merchant asks Alice to randomly reveal either the left half or the right half of each identity string on the money order. In effect, the merchant gives Alice a randomly chosen \(n\)-bit selector string, \(b_1, b_2, \ldots, b_n\). Alice opens the left or right half of \(I_i\), depending on whether \(b_i\) is a 0 or a 1.
Digital cash: Depositing It

- Alice complies.
- The merchant takes the money order to the bank.
- The bank verifies the signature and checks its database to make sure a money order with the same uniqueness string has not been previously deposited. If it hasn’t, the bank credits the amount to the merchant’s account. It records the uniqueness string in the database.
Catching the cheater: The merchant

• If a money order with the same uniqueness string in the database, the banks refuses to accept it.
• It compares the identity string on the money order with the one stored in database. If they are the same, the bank figures out the merchant is the one who photocopied the money order.
Catching the cheater: The spender

- If the uniqueness string is in the database and the identity strings are different, then the bank knows the spender is the cheater. Furthermore, it can even reveal the identity of the spender. Here how it does:
- Alice must have spent the money order with two different merchants who probably have picked up two different selector string.
- Since these two binary string are selected at random and we know that for some bit position they do not coincide.
Catching the cheater: The spender

- The non-coinciding bit position of two strings reveal the identity of the spender, since one merchant had Alice open the left half of the identity string while the other merchant had Alice open the right half.
- The bank XOR the two halves to get the identity.
Can Alice Cheat?

- Alice can cheat by trying to pass a fake money order; but her chances are one in \( n \). The penalty if she get caught is severe.
- She can try to use a legitimate money order twice, her identity is revealed.
Can the Merchant Cheat?

- The merchant may try to deposit a money order twice. But the bank immediately can figure out the merchant is trying to cheat.
- If the merchant tries to use random identity strings, then the bank understands the money order is trying to be spent twice. But the double spending does not reveal Alice’s identity. The bank decides the merchant is the cheater.
Can the Bank Trace Alice?

- As long as Alice does not try to spend the money order twice, the bank learns nothing about Alice’s identity from one half of the identity strings.
- Alice is protected by the blind signature protocol.
- Even if the bank and the merchant get together, they cannot reveal Alice’s identity.
Can Eve Cheat?

- Yes, she can.
- If she steals the money order from Alice, she can spent it herself.
- She can try to spend it twice. Alice is identified as the cheater.
- She can eavesdrop on the communication between Alice and the merchant; and if she gets to the bank before the merchant does she can deposit the money order to her own account.
- But, this is usually what happens when your money is stolen.
Requirements

- Our protocol satisfies 4 out of 6 requirements:
  1. The cash can be sent securely through computer networks.
  2. The cash cannot be copied and reused
  3. The spender of the cash can remain anonymous.
  4. The transaction can be done offline.
- But it fails to fulfill the last to
  5. The cash can be transferred to others.
  6. A piece of cash can be divided into smaller amounts
Protocols

- A protocol satisfies 1 through 4 was first proposed in D. Chaum, A. Fiat, and M. Naor, “Untraceable Electronic Cash” Advances in Cryptology – CRYPTO ’88 Proceedings, pp. 319-327
- Okamoto and Ohta proposed a protocol fulfills all the requirements in T. Okamoto and K. Ohta “Universal Electronic Cash”, Advances in Cryptology – CRYPTO ’91 Proceedings, pp. 324-337.
- Each payment requires data transfer of 20 kilobyte.