Digital Payments

Week 8

Schneier: Ch 6.4
Today’s Class

- E-cash
Digital Payments

- Money exists in physical form
  - Difficult to carry
  - Can be stolen
  - Hard to trace

- Credit cards and checks
  - Easy to carry
  - Harder to steal the money
  - Easy to trace

- Can we have the best of two worlds?
  - E-cash
E-Cash

- Money exchanged only electronically
  - Nothing to carry
  - Hard to steal
  - Hard to trace

Alice "the dealer"

Bob "the payer"

Prevent double spending!

Spend $10 token

Spend $10 token

Marco "the merchant"
Solution 1: Blind Signatures: Withdraw

1. Generate RSA key (e,d,n)
2. pk = (e,n)
3. Generate k messages
4. Generate k randoms
5. \( B_i = M_i X_i^e \mod n, i=1..k \)
6. Pick r random from 1..k
7. \( (M_i, X_i), i=1..k, i \neq r \)
8. Verify correctness of \( M_i \) & \( B_i = M_i X_i^e \mod n, i=1..k, i \neq r \)
9. Deduct $1000 from B’s account
10. \( S_T(B_r) = B_r^d \mod n \)
11. \( S_T(M_r) = M_r^d = B_r^d X_r^{-1} \mod n \)

Bob "the payer"

Trent "the Bank"

\( X_1 \quad M_1 = ($1000) \)
\( X_2 \quad M_2 = ($1000) \)
\( \ldots \quad \ldots \)
\( X_k \quad M_k = ($1000) \)
Solution 1: Spend & Deposit

1. **Spend:** $M_r = ($1000), $S_T(M_r)$

2. **Verify $1000**
   - Verify T’s signature

3. **Deposit:** $M_r$, $S_T(M_r)$

4. **Verify $1000**
   - Verify signature

5. **Credit $1000** in Marco’s account
Solution 1: Properties (or lack thereof ...)

- The bank never sees the money it signs
  - Bob spends the money
  - The merchant brings the money to the bank
  - The bank cannot trace it to Bob
- The bank trusts the money are valid
  - Bob has $1/k$ chance of cheating
- Bob can copy the money $M_r^d = B_r^d X_r^{-1} \mod n$
  - Spend them multiple times
  - Double Spending
Solution 2: Double Spending Detection

1. Generate RSA key (e,d,n)
2. pk = (e,n)
3. Generate k messages
4. Generate k randoms
5. \( B_i = M_i X_i^e \mod n \), i=1..k
6. Pick r random from 1..k
7. \((M_i, X_i), i=1..k, i \neq r\)
8. Verify correctness of \( M_i \) & \( B_i = M_i X_i^e \mod n \), i=1..k, i \neq r
9. Deduct $1000 from B’s account
10. \( S_T(B_r) = B_r^d \mod n \)
11. \( S_T(M_r) = M_r^d = B_r^d X_r^{-1} \mod n \)

\[ \begin{align*}
X_1 & \quad M_1 = ($1000, Serial Nr.1) \\
X_2 & \quad M_2 = ($1000, Serial Nr.2) \\
\vdots & \quad \vdots \\
X_k & \quad M_k = ($1000, Serial Nr.k)
\end{align*} \]
Solution 2: Spend & Deposit

1. **Spend**: $M_r = (1000, SN_r)$,
   \[ M_r^d \mod n \]

2. **Verify $1000**
   **Verify T’s signature**

3. **Deposit**: $M_r$, $M_r^d \mod n$

4. **Verify $1000**
   **Verify signature**

5. **Verify $SN_r$ was never used**

6. **Credit $1000$ in M’s account**
Solution 2: Problem

- Trent can detect cheating
- **Cannot detect the cheater**
  - It could be the payer
  - Or the merchant

- We would like to be able to detect a double spending payer
- Use secret splitting
Solution 3: Withdraw

1. Generate k messages

2. \[ M_i = [SN_i, (CMT(R_j), CMT(Id xor R_j), j=1..t] \]

3. Pick r random from 1..k

4. \[ [SN_i, X_i, R_j, CMT(R_j), R_j xor Id, CMT(R_j xor Id) \]
   \[ j=1..t], i=1..k, i ≠ r \]

5. Verify \( R_j xor R_j xor Id = Id \) Bob
   \[ B_i = M_i X_i^e \mod n, i=1..k, i ≠ r \]

6. Deduct $1000 from B’s account

7. \[ S_T(B_r) = B_r^d \mod n \]

8. \[ S_T(M_r) = M_r^d = B_r^d X_r^{-1} \mod n \]

\( SN_i \)
\( R_1 \)
\( Id xor R_1 \)
\( R_2 \)
\( Id xor R_2 \)
\( ... \)
\( R_t \)
\( Id xor R_t \)

CMT = commitment function (see earlier class)
Solution 3: Spend & Deposit

1. **Spend:** \( M_r = [SN_r, CMT(R_j), CMT(Id \ xor \ R_j), j=1..t] \)
   \( S_T(M_r) \)

2. \( b_1, .., b_r \)

3. \( b_i = 0? \ A_i = R_i \)
   \( b_i = 1? \ A_i = Id \ xor \ R_i \)

4. **Deposit:** \( M_r, S_T(M_r) \)
   \( A_i, i=1..t \)

5. If \( SN_r \) has been used before

6. Find index \( i \) such that both
   \( A_i = R_i \) and \( A_i = Id \ xor \ R_i \) have been revealed
   Find Id of Bob – double spender
Solution 3: Discussion

- Can Bob double spend?
  - First time – fine
  - Second time – trouble
  - Chance of succeeding – $1/2^t$

![Diagram of transaction sequence]
Discussion (cont’d)

- Can Bob create shares that do not reveal his identity?
- Chance of succeeding – \(1/k\)
Discussion (cont’d)

- Can Marco cheat?
  - He can try to spend the same money twice
    - The serial number is the same
    - The challenge bits are the same
    - Quite suspicious

- Can Bob and Marco collude?
  - No. For each serial number
  - The bank will provide the money only once
Discussion (cont’d)

- Can Trent (the bank) cheat?
  - Find Bob’s identity
  - No. Bob uses blind signatures
- Can Trent (the bank) and Marco (the merchant) collude?
  - No. If Bob doesn’t double spend, he does not reveal any identity information in the challenge step of Spend
“Rushing” Attack

- Eve listens on the `Spend` protocol between Bob and Marco
- Then takes the deposit values intended for Marco
- And sends them quickly to the bank
- *Before Marco ...*
Micropayments

- Some payments are small in value
- Examples
  - Buying access to music
  - Electronic transactions from your smartphone
- The value of the payment may be smaller than the cost of operating the infrastructure
  - Electricity cost
  - Amortized equipment and administrative costs
- Need efficient (cheap) electronic payment technologies
Micropayments

- Many alternatives
  - Payword
  - Micromint
  - ...
- NOT Anonymous
Payword

**Commitment:** \( M = S_B(\text{Marco}, C_B, w_0, \text{Time}) \)

1. **OpenAccount:**
   - Credit card #,
   - Address \( A_B \),
   - \( pk_B \)

2. \( C_B = S_T(B, A_B, pk_B, E) \)

3. Compute \( w_0, w_1, \ldots, w_n \) worth one cent each

4. **Commitment:**

5. Verify B’s signature on \( S_B \)
   Verify T’s signature on \( C_B \)

\[ w_0 = H(w_1) \]
\[ w_1 = H(w_2) \]
\[ \ldots \]
\[ w_{n-1} = H(w_n) \]

\( E = \text{Expiration time} \)
Payword (cont’d)

Payment: $w_1$, 1
Payment: $w_2$, 2
Payment: $w_k$, k

Verify $w_0 = H(w_1)$
Verify $w_1 = H(w_2)$
Verify $w_{k-1} = H(w_k)$

Deposit: $C_B$, $M$, $w_k$, k

Need only store $w_k$ for k payments!

Verify $T$’s signature on $C_B$
$C_B = S_T(B, A_B, pk_B, E)$

Verify signature of $B$ on $M$
$M = S_B(Marco, $C_B$, $w_0$, Time)

Verify $w_0 = H_k(w_k)$
Bitcoin
What is Bitcoin?

- Global peer-to-peer currency
  - Designed for the Internet
  - Behaves like cash online
  - Can be used by anyone

- No central authority
- Developed by Satoshi Nakamoto

- There will only be a total of 21 Million Bitcoins created or mined until the year 2140
What is Bitcoin?

- The only regulated factor built into the system is the number of coins generated through mining.
- The network allows approximately 1 block to be mined every 10 minutes.
  - Worth 12.5 Bitcoins.
  - This number halves approximately every four years.
- The currency is divisible by 8 decimal places. ($\text{฿}.00000001$)
  - Can be adjusted within the Bitcoin protocol if needed.
Early Days

- Bitcoin network released in January 2009
  - Open source Bitcoin client

- Genesis block
  - Mined by Satoshi Nakamoto
  - 50 bitcoins reward
Why Only 21 Million Bitcoins?

- First block introduced 50 coins in system
- This halves every 210,000 blocks (system parameter)
- Total number of Bitcoins:

\[
\sum_{n=0}^{\infty} \frac{210000 \times 50}{2^n} = 210000 \times 50 \times \frac{1}{1 - \frac{1}{2}} = 21000000
\]
What are they worth?

- Worth what we think they are

- When the first Bitcoin exchanges went online, the price per Bitcoin was about $0.05*.

- October 12, 2016: 1 BTC (Bitcoin) is worth $640
  - October 11, 2017: 1 BTC is worth $4,818
How do you get Bitcoins?

- 4 methods to acquire Bitcoins
  1. Mining
  2. Wire in fiat currency to a BTC exchange
  3. Buy from an individual
  4. Sell items for BTC
Where do you store Bitcoins?

- Bitcoins are stored in wallets
- Wallet: a randomly generated string of numbers
- Consists of two parts
  - the public key
  - the private key
- Example of a public key:
  14GabW85FUMQy62CMWLCToQLo81w7iXL2x
- Only the wallet owner has access to private key
  - If anyone gets ahold of the private key, they will have access to all of the wallet’s Bitcoins
Bitcoins vs. Email

Anyone can send you email if they know your public email address.

But only you can send email from that account with your private email password.

Anyone can send you Bitcoin if they know your public Bitcoin address.

But only you can send Bitcoin from that address with your private Bitcoin key.
What is a Bitcoin wallet?

- **3 types of wallets**

1. **Online Wallet Services**
   - User creates and remembers a password
   - Popular service: Coinbase
   - **Risks:** Susceptible to network failures and hacking

2. **Local Wallets or Offline Wallets**
   - Application downloaded to user’s computer
   - Private keys are stored locally on the hard drive
   - **Risks:** computer failures, theft, hacking

3. **Paper Wallets**
   - Public/private key printed on a physical piece of paper
   - Arguably the most secure way to store your BTC
   - **Risks:** Theft, physical damage
Transactions: How to Send Bitcoins

- Scan QR code with your smartphone
- Scan the receiver QR, enter the number of coins to send

QR Code Generated for the public key:
14GabW85FUMQy62CMWLCToQLo81w7iXL2x
Bitcoin Mining

- The Bitcoin (peer-to-peer) network maintains a Transaction ledger
  - Also called Blockchain
  - String of transactions
  - Organized into blocks
  - Each block stores a few hundred transactions

- Problems that can occur:
  - Attacker removes older transactions
  - Attacker modifies older transactions
  - Attacker inserts transactions that did not occur

- Need to ensure integrity of distributed ledger !!!
Bitcoin Mining

- Every 10 minutes or so
  - Mining computers group existing pending transactions
  - A few hundred transactions per block
  - Transform the block into a cryptographic puzzle

- The first miner to find the solution announces it to others on the network

- The other miners then check
  - Whether the puzzle has been solved
  - If the sender of the funds has the right to spend the money
Bitcoin Mining

- If enough of “verifiers” grant their approval
  - The block is cryptographically added to the ledger
  - The miners move on to the next set of transactions
  - (hence the term “blockchain”)

- The miner who found the solution gets 12.5 bitcoins as a reward
  - After another 99 blocks have been added to the ledger
- Gives miners incentive to participate and validate transactions
Bitcoin Mining

- Integrity of transaction ledger/blockchain:
  - Force miners to solve puzzles in order to add to the ledger

- To double-spend a bitcoin
  - Adversary needs to rewrite the blockchain
  - Need to control more than half of the network’s puzzle-solving capacity

- Such a “51% attack” is expensive
  - Bitcoin miners have 13,000 times more combined number-crunching power than the world’s 500 biggest supercomputers
Genesis Block

Here is a representation of the genesis block[1] as it appeared in a comment in an old version of Bitcoin (line 1613). The first section defines exactly all of the variables necessary to recreate the block. The second section is the block in standard printblock format, which contains shortened versions of the data in the first section.

- Hash of genesis block
  - 0000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f

- 2 more leading hex zeroes than were required for an early block