Lecture 0: Some Odd Problems in Cryptography

Viet Tung Hoang
1. The dating problem

2. Telephone coin flipping
The Dating Problem

Second date?

**Issue:** Embarrassing if one wants a second date while the other doesn’t.
The Dating Problem

Privacy: Each person only knows:
1. His/her choice
2. The final outcome
3. Whatever can be inferred from (1) and (2).

Second date?
Bob’s Privacy for the Dating Problem

Alice knows Bob’s input = “agree”

Alice knows Bob’s input = “disagree”

In those cases Bob’s privacy is moot
Bob’s Privacy for the Dating Problem

Disagree

The protocol must give no information about Bob’s input
How to Solve the Dating Problem: Step 1
How to Solve the Dating Problem: Step 2

Agree

Disagree

Agree

Disagree
How to Solve the Dating Problem: Step 3

Alice’s cut

Each takes turn to make a **private** cut
How to Solve the Dating Problem: Step 3

Bob’s cut

Each takes turn to make a **private** cut
How to Solve the Dating Problem: Step 4

If three ♥️ in a (wrap-around) row then date. Otherwise no date.
Why Is the Solution Correct?

There are ten ways to place 3 ♠ and 2 ♦ in a line.

For wrap-around, there are two groups.
The Initial Place

Date: Group 1

No Date: Group 2
Cutting Doesn’t Change the Group

Circular shift
Why Is the Solution Private?

Homework Exercise
Agenda

1. The dating problem

2. Telephone coin flipping
Alice and Bob want to decide who gets the car (over the phone)

Alice’s proposal:
• Alice tosses a coin and informs Bob of the outcome
• Bob gets the car if the coin lands head
Telephone Coin Flipping

Goal:
- Both Alice and Bob learn the outcome of a fair coin toss
- Nobody can cheat the other
A Physical Solution

Alice sends a random bit $M$ to Bob in a locked box.

Bob sends another random bit $M'$ to Alice.

Alice sends the key to Bob to open the box.

Output = $M \oplus M'$
How to Implement A Digital Locked Box

First attempt:

-A locked box containing a bit $M$ is an encryption $C \leftarrow E_K(M)$

-The key to open the box is the key $K$

What can go wrong?

- Alice can send a **fake** key $K'$ so that $E_{K'}^{-1}(C)$ is another bit of her choice
We Actually Need a **Bit Commitment Scheme**

**Commit:** \((C, K) \leftarrow \text{Comm}(M)\)

- committal
- \(M \in \{0, 1\}\)

**Decommit:** \(M' \leftarrow \text{DeComm}(K, C)\)

- \(M' \in \{0, 1\} \cup \{\bot\}\)

**Correctness:** If \((C, K) \leftarrow \text{Comm}(M)\) then \(\text{DeComm}(K, C) = M\)

**Put a bit** \(M\) **in a locked box:**

- \(M\) \rightarrow \text{Comm}\rightarrow \text{C}
- \text{Comm}\rightarrow \text{K}

\(M\) in a locked box with a key.
How to Use Bit Commitment

\[(C, K) \leftarrow \text{Comm}(M)\]

Alice sends the committal \(C\) of her random bit \(M\)

Bob sends another random bit \(M'\)

Alice sends the key \(K\)

\[M \leftarrow \text{DeComm}(K, C)\]

Output = \(M \oplus M'\)
Security Requirements of Bit Commitment

**Hiding**: Committal $C$ reveals **nothing** about $M$

Bob can’t learn the value in the locked box
Security Requirements of Bit Commitment

**Binding:** It’s **hard** to find $C^*$, $K_0$, $K_1$ such that

$$\text{DeComm}(K_0, C^*) = 0 \text{ and } \text{DeComm}(K_1, C^*) = 1$$

Alice can’t construct a box that she can open to both 0 and 1
A Simple Bit Commitment Scheme

Commit to 0:
- Pick two 1024-bit primes $p$, $q$ such that
  \[
  \begin{align*}
  p &< q \\
  p &\equiv 3 \pmod{4}, \quad q \equiv 1 \pmod{4}
  \end{align*}
  \]

Commit to 1:
- Pick two 1024-bit primes $p$, $q$ such that
  \[
  \begin{align*}
  p &< q \\
  p &\equiv 1 \pmod{4}, \quad q \equiv 3 \pmod{4}
  \end{align*}
  \]

Committal: $N = pq$

Key: $(p, q)$
Implementing Decommitment

Committal: \( N = pq \)

Key: \((p, q)\)

Is the box valid? Does the key open it?

1. Check if \( N \equiv 3 \mod 4 \) and \( N = pq \)

2. Check if \( p \) and \( q \) are primes, and \( p < q \)

Attack when this step is missing: Alice picks \( N = 99 \)

She can open to 0 with \((p, q) = (3, 33)\) or open to 1 with \((p, q) = (9, 11)\)

3. Output 1 if \( p \equiv 1 \mod 4 \), and output 0 otherwise