Some Odd Problems in Crypto

Viet Tung Hoang
1. The dating problem

2. Telephone coin flipping
The Dating Problem

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

**Want**: Each person only knows:

- His/her choice & the final outcome
- Whatever **can be inferred** from the above
Bob’s Privacy for the Dating Problem

Alice knows Bob’s input = “agree”

In those cases Bob’s privacy is moot

Alice knows Bob’s input = “disagree”
Bob’s Privacy for the Dating Problem

Must reveal **no information** about Bob’s input
How to Solve the Dating Problem: 5-card Trick
How to Solve the Dating Problem: 5-card Trick

Alice’s cards

Agree

Disagree

Bob’s cards

Agree

Disagree
How to Solve the Dating Problem: 5-card Trick

Alice’s cut

Each takes turn to make a private cut
How to Solve the Dating Problem: 5-card Trick

Bob’s cut

Each takes turn to make a **private** cut
How to Solve the Dating Problem: 5-card Trick

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.
But There Are Two Groups When Wrap Around
The Initial Place

Date: Group 1

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

Circular shift
Why Is the Solution Private?

Your Exercise
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

$M \in \{0, 1\}$
A Physical Solution
A Physical Solution
A Physical Solution

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?

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

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

\[M \in \{0, 1\}\]

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

\[M' \in \{0, 1\} \cup \{\perp\}\]

How to put a bit in a locked box

![Diagram of a locked box with a committer and a decommitter](image)
We Actually Need a **Bit Commitment Scheme**

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

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

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

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

How to open

\[\begin{array}{c}
\text{M} \\
\text{C} \\
\text{K} \\
\text{DeComm} \\
\text{M}
\end{array}\]
Security Requirements of Bit Commitment

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

Alice 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

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

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

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

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

Committal: \( N = pq \)

Key: \( (p, q) \)
Implementing Decommitment

Your Exercise