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.
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”

Agree

Alice knows Bob’s input = “disagree”

Agree

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

Disagree

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

Bob’s cards

Agree

Disagree

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

<table>
<thead>
<tr>
<th>Card</th>
<th>Date</th>
<th>No date</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠️ Ace</td>
<td>♠️ Ace</td>
<td>♦️ Ace</td>
</tr>
<tr>
<td>♠️ 2</td>
<td>♠️ 2</td>
<td>♦️ 2</td>
</tr>
<tr>
<td>♣️ 2</td>
<td>♣️ 2</td>
<td>♥️ 2</td>
</tr>
<tr>
<td>♥️ A</td>
<td>♥️ A</td>
<td>♥️ A</td>
</tr>
</tbody>
</table>

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
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

Sends a random bit $M$ in a locked box
A Physical Solution

Send back another random bit $M'$
A Physical Solution

Sends the key to open the box
A Physical Solution

\[
\text{Output} = M \times M'
\]
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:** $M' \leftarrow \text{DeComm}(K, C)$

- $M' \in \{0, 1\} \cup \{\perp\}$

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

**How to put a bit $M$ in a locked box**
How to Use Bit Commitment

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

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

\(M'\) 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

Your Exercise
Security Of Our Bit Commitment Scheme

**Binding:** For an integer \( N \), there is a unique way to factor it as

\[
N = pq \quad \text{where } p \text{ and } q \text{ are primes, and } p < q
\]

**Hiding:** This seems plausible that unless one can factor \( N = pq \)

one cannot learn whether \( p \equiv 3 \pmod{4} \) or \( p \equiv 1 \pmod{4} \)