Special Topics in Cryptography

Mohammad Mahmoody
Logistics

• Most submitted PS3. If you have not you will get delay, but email it to me ASAP.

• Deadline for project reports/drafts + slides : This Thursday 5pm. There will be a collab post for it.

• I will announce the order of presentations. So your talk could be on any of the remaining days.

• You are all anticipated to participate in each others’ presentations.
Last time

• Zero Knowledge Proofs

Today

• Secure computation
Can we ever prove we know something without revealing the details of the secret?

- Alice knows a magic word to open the door inside the cave:
Formal Definition of Zero Knowledge Proofs

- Suppose \( L \in \text{NP} \) meaning, there is poly-time verifier \( V(\cdot, \cdot) \) such that \( x \in L \iff \exists w, V(x, w) = 1 \)

- Examples:
  - An "interactive" protocol between a "prover" \( P \) and a "verifier" \( V \):
    1. Is sound if: for all even malicious prover \( P^* \): \( \Pr[V(x) = 1] \leq \text{negl}(n) \)
    2. Is zero-knowledge if: for all even malicious verifier \( V^* \): \( \exists S \) such that \( S(x) \approx \text{view}(V) \) in interaction with \( P \) on input \( x \)
Suppose we deal with Lang L: \( \{ N \mid N = \text{p.q} \} \)

\( V(N, w) = \text{to make sure N is p.q} \)

\( (p,q) \)

\( \exists \text{ algorithm } T_L(N) \rightarrow G \) such that \( N \in L \) if \( G \) is 3-colorable

\( \& \) \( S_L(w) \rightarrow c \) such that \( N \in L \) such that if \( N \) is p.q \( \rightarrow \) \( c \) is a 3-colorable for \( G \).

Prove \( N \in L \) \( \equiv \) \( T_L(N) = G \in 3\)-Colorable
Another way to see these two properties

- Using a “trusted third party”.

- Real World:

- Ideal World:
Secure Multiparty Computation
Yao’s Billionaires Problem: Who has more money?
In General

- Parties $P_1, ..., P_m$ want to compute $f(x_1, ..., x_m)$ “securely”:
  - Party $P_i$ has input $x_i$ and would learn $f(x_1, ..., x_m)$
  - Nobody should learn beyond what they would from the output.

- Security Models:
  1. **Semi-honest** (aka honest-but-curious): cheating party follows the protocol, but at the end tries to extract information.
  2. **Malicious**: cheating party might deviate from protocol completely.
How to define security in general?

• Real Model:

\[ f(x, y) \]

Protocol output → \[ f_1(x, y) \]

Protocol output

• Ideal Model:

Security for \( P_1 \): \( \exists \text{Sim}_2(.) \) that takes the view of \( P_2 \) in Ideal world and simulates view \( P_2 \) in Real

Security for \( P_2 \): ...
How about fully malicious attackers (who might change their inputs)?

- **Real Model:**

- **Ideal Model:**

  - Ideal model does not allow changing the inputs after they are ‘sent’
Oblivious Transfer: a "complete" functionality

\[ f(x, b) = 1 \]

Security for Alice:
Bob is reading only one of \( x_1, x_2 \)

Security for Bob:
Alice does NOT know \( b \)
Semi-Honest OT from Trapdoor Permutations
Using OT to get 2 party secure computation
Recall: Secure Function Evaluation

- Protocol’s output: $f(x,y)$ where function $f$ is known to both parties.
Yao’s Solution: Garbling of circuits: (Using OT and SKE as building Blocks)

1. Alice writes $f$ as a circuit $C$
2. Convert $C$ into a “garbled” version $G$ where:
   - $G$ “hides” the computation and only reveals the output.
   - Bob can plug in his input only with Alice’s help.
3. Alice sends $G$ (and related keys) to Bob
4. Bob gets right keys for his own inputs using OT protocol.
5. Bob “executes” the circuit and sends the answer back.
Garbling truth table of NAND gate

\[ x \quad y \quad z = \text{NAND}(x, y) \]

Want: if somebody knows
\[
\begin{pmatrix}
    k_{ox} \\
    k_{ly}
\end{pmatrix}
\]
returns \( k \) on wrong rows.
Yao’s garbled circuit

• The basic form is only semi-honest secure

• Can be made maliciously secure:
  • inefficiently: using ZK proofs
  • Efficiently: using “cut and choose”