Secure computation

With material from Matthew Green, Elaine Shi, CS Unplugged, others
• P3 due Friday

• Exam 2 next Tues
  • Material through today’s lecture
  • Exam review Thurs
• Secure computation
  • Zero-knowledge proofs
  • Commitment schemes
  • Multiparty computation
Zero-knowledge proofs
• Goal: P proves to V that some statement is true
  • *Without* conveying additional information

• In general, probabilistic
  • Repeat a bunch of times as proof
Example 1: Hallway password

- Does Peggy have the key?
- Both stand in the entrance.
  - When Victor isn’t looking, Peggy picks one hall
  - Victor then yells “GREEN” or “ORANGE”
  - Peggy must come back via the chosen color
- Repeating many times “proves” Peggy has password
  - With high probability
Example 2: Two baseballs

- Peggy has two baseballs: One red, one green
  - Otherwise identical

- Victor is color-blind, thinks they are the same
  - Peggy’s goal: To prove she can distinguish

- Peggy places them in Victor’s hands
  - Victor puts them behind his back, may switch
  - Peggy tells whether he switched
  - As before, repeat many times
Security properties

• Complete: Honest V will be convinced by honest P
• Sound: Honest V can’t* be convinced by cheating P
• Proves nothing to outside observers either way
  • Peggy and Victor can **collude** by **precomputing**
• Peggy could cheat with a time machine
  • Victor gets the same info either way
  • Implies that real protocol does not leak
Burning questions

• Why is this crypto?
• Does everyone have to be in the same place?
• Why do we care in real life?
Commitment schemes

COMMITMENT
The chicken is involved. The pig is committed.
Commitment schemes

- Commit to a value but do not show it
  - *Open* it later and prove it hasn’t changed
- Analogy:
  - I pick a number between 1 and 100
    - Write it down and seal it in an envelope
  - You pick odd or even
  - If you’re right, I pay you; else you pay me
  - *Why did I have to write it down?*
Required properties

- Hiding: Commitment reveals nothing about value
- Binding: Can’t open to a different value
Remote coin-flip

- Goal: Flip coin over the telephone
  - Alice flips, Bob chooses heads or tails
- Requires Alice to commit her output
  - In essence, need a one-way function
- Example/activity: Using and/or circuits
Heads = Even input parity
Tails = Odd input parity
Try it! (Small groups)

• “Bob” draws a circuit

• “Alice” commits to an outcome

• “Bob” chooses odd or even parity

• Declare a winner

• Can either of you cheat? How?
Cheating

- Alice can cheat IFF she has two opposite-parity inputs that produce the same output
- Bob can cheat IFF he can predict the input from the output
Commitment via hash

- Alice, Bob pick a random numbers X, Y
  - Alice publishes H(X); Bob publishes H(Y)
- Bob chooses odd or even
  - Reveal X, Y and add them; check sum parity
- Collision resistance: Can’t fake X or Y
- Pre-image resistance: Can’t calculate X or Y
Multiparty computation
• Everyone has a private input
• Together, we compute some related result
• No one’s private input is given away
Example 1: How old are we?

- Goal: Find our average age
  - Without anyone giving away their own age
- Activity: Need five volunteers
  - And five sheets of paper
Setup

- Alice, Bob are **honest but curious**
  - Don’t lie, follow protocol correctly
  - But try to learn from available info

- Security equivalent to **fully trusted** third party
Defining leakage

- Learning $f(a,b)$ gives some information
- What if $f(a,b) = (a + b)$?
- Final security property:
  - Alice learns only info computable from $f(a,b)$, $a$
  - Bob learns only info computable from $f(a,b)$, $b$
Example 2: Truth in dating

• After meeting and chatting, Alice and Bob want to find out whether they want to date each other

• If Bob says no, Alice doesn’t reveal her answer
  • And vice versa

• Essentially secure AND

<table>
<thead>
<tr>
<th>Alice</th>
<th>Bob</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO DATE</td>
<td>NO DATE</td>
<td>NO DATE</td>
</tr>
<tr>
<td>NO DATE</td>
<td>DATE</td>
<td>NO DATE</td>
</tr>
<tr>
<td>DATE</td>
<td>NO DATE</td>
<td>NO DATE</td>
</tr>
<tr>
<td>DATE</td>
<td>DATE</td>
<td>DATE</td>
</tr>
</tbody>
</table>
Solution using 5 cards

- Alice and Bob each get two emoji cards: ❤️,💣
  - Plus one public ❤️

- Place cards face down on table as follows:

  | A | A | ❤️ | B | B |

- Using this chart:

<table>
<thead>
<tr>
<th>DATE</th>
<th>BOB</th>
</tr>
</thead>
<tbody>
<tr>
<td>❤️️</td>
<td>🍀️</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NO DATE</th>
<th>ALICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>🍀️</td>
<td>🍀️</td>
</tr>
<tr>
<td>🍀️</td>
<td>🍀️</td>
</tr>
</tbody>
</table>
Solution, ctd.

• Each gets to privately cyclic-shift the cards X times

• Final results: 3 hearts in a row = match

Equivalent under cyclic shift!
• Exercise: Try to come up with a $\leq 4$ card solution
  • Share results in class or on Piazza
Other sample problems

- Two reporters compare confidential sources
  - To see if they are the same person
- Check for secret society password
- Find out who bid more without revealing your bid
- etc.
Desired properties

- **Resolution**: Find out desired outcome
- **Privacy**:
  - No involved party learns anything else
  - No third party learns anything
- **Security**: No one profits by cheating
  - Can’t know outcome unless other party does
- **Simplicity**: Easy to implement, understand
- **Remoteness**: Don’t need to be co-located
Example: Who is richer?

Yao’s millionaire’s problem (1982)

- Alice (i) and Bob (j) have $1 \leq i, j \leq 6$
  - Assumption for simplicity
  - Generalizable to more people, more numbers
  - Later improvements in efficiency
- Also has security limitations
  - For conceptual purposes only
1. Bob’s turn

- Bob chooses a large random number $x$
- Bob computes $m = E(PK_A, x)$
- Bob sends to Alice: $B = m - j + 1$

*Example: $j = 5, B = m - 4$*
2. Alice’s turn

- Alice generates \( y_u = D(SK_A, m - j + u) \) for \( u = 1:6 \)
  - \( y_u = D(SK_A, B + u - 1) \)

- Alice picks a prime \( p \) and generates \( z_u = y_u \mod p \)
  - Ensure all \( z \)'s at least 2 apart or try again

- Example:
  - \( z_3 = D(SK_A, m - 2) \mod p \)
  - \( z_5 = D(SK_A, m) \mod p = x \mod p \)
2.5 Still Alice’s turn

- Alice sends $p$ to Bob

- Alice sends 6 numbers to Bob as follows:
  - $z_1 .. z_i$
  - $z_{i+1} + 1 .. z_6 + 1$

- **Example:** $i = 2$
  - $z_1, z_2, z_3 + 1, z_4 + 1, z_5 + 1, z_6 + 1$
3. Bob’s turn

- Bob looks at the jth number in Alice’s list
  - If it equals \( x \mod p \) then \( i \geq j \)
  - If not, then \( i < j \)

- Bob tells Alice the answer

- Example: 5th number = \( z_5 + 1 \)
  - \( z_5 + 1 = (x \mod p) + 1 \neq x \mod p \)
Security caveats

• Brute force: Bob looks for q s.t. $E(q) = m - j + 2$
  • Can figure out whether $i \leq 2$

• What if Bob lies to Alice?

• Lots of extensions, generalizations, etc.