Secure Computation - I
(Introduction and Definitions)

Lecture 13
Data Privacy, so far

How can Alice send $x$ privately to Bob?
Data Privacy, so far

Public-key Encryption

PK, x → SK
Data Privacy, so far

Public-key Encryption

PK, x → x → SK
Data Privacy, so far

Public-key Encryption

- If Bob has the secret-key, then he learns **entire** $x$
Data Privacy, so far

Public-key Encryption

• If Bob has the secret-key, then he learns entire $x$
• Else, he learns nothing about $x$
“All-or-nothing” Privacy
(either learn the entire secret or nothing about it)
Today’s Lecture

“Controlled” Privacy

(release partial information about the secret)
Example from earlier

Zero-Knowledge Proofs

(x, w) \quad x \in L \quad (x)

- Bob learns that $x \in L$ but nothing else about $w$
Matchmaking
Matchmaking
Matchmaking

Problem: Tinder not only learns that the players matched, but also their entire profiles
Problem: Tinder not only learns that the players matched, but also their entire profiles
Matchmaking

**Want**: The only information revealed is that there was a match, no more
Correlation between Smoking and risk of early onset Alzheimer’s disease
Law Enforcement

Can we perform DNA matching without violating privacy of individual?
Feb 10, 2009: Two satellites, Iridium 33 and Kosmos-2251, collided

Unlikely that Governments will share Location and Trajectory of Military Satellites

How can governments compute “safe” trajectories without sharing private data?
General Problem

Common Input: $f$
General Problem

**Goals:**
- **Correctness:** Both parties learn $f(x, y)$

**Common Input:** $f$

$x$

$y$
General Problem

Goals:
- **Correctness**: Both parties learn $f(x,y)$
- **Security**: Each party only learns $f(x,y)$

Common Input: $f$
Remarks

Wlog, we will consider:

- **Symmetric functions**: \( f(x,y) = (z_1, z_2) \) where \( z_1 = z_2 \)
Remarks

Wlog, we will consider:

• **Symmetric functions**: \( f(x,y) = (z_1,z_2) \) where \( z_1 = z_2 \)

• **Think**: Asymmetric functions?
Remarks

Wlog, we will consider:

- **Symmetric functions**: \( f(x,y) = (z_1, z_2) \) where \( z_1 = z_2 \)
- **Think**: Asymmetric functions?
- \( g((x,r),(y,s)) : (z_1,z_2) = f(x,y) \). Output \( z_1 + r, z_2 + s \)
Remarks

Wlog, we will consider:

- **Symmetric functions**: $f(x,y) = (z_1, z_2)$ where $z_1 = z_2$

  - **Think**: Asymmetric functions?

- $g((x,r),(y,s))$: $(z_1, z_2) = f(x,y)$. Output $z_1 + r$, $z_2 + s$

- **Deterministic functions**
Remarks

Wlog, we will consider:

- **Symmetric functions:** \( f(x,y) = (z_1, z_2) \) where \( z_1 = z_2 \)
  - **Think:** Asymmetric functions?

- \( g((x,r),(y,s)) : (z_1, z_2) = f(x,y) \). Output \( z_1 + r, z_2 + s \)

- **Deterministic functions**
  - **Think:** Randomized functions?
Remarks

Wlog, we will consider:

• **Symmetric functions**: \( f(x,y) = (z_1,z_2) \) where \( z_1 = z_2 \)

• **Think**: Asymmetric functions?

• \( g((x,r),(y,s)) : (z_1,z_2) = f(x,y) \). **Output** \( z_1+r, z_2+s \)

• **Deterministic functions**

  • **Think**: Randomized functions?

  • \( g((x,r),(y,s)) : \) Output \( f(x,y;r+s) \)
General Problem

$x$

$y$
General Problem

\[ z = f(x, y) \]
General Problem

Problem: Where to find this trusted party?
Secure Computation
[Yao82, Goldreich-Micali-Wigderson-87]

Goal: Compute $f(x, y)$
Secure Computation

[Yao82, Goldreich-Micali-Wigderson-87]

Algorithmically Emulate the Trusted Party

\[ x \quad \rightarrow \quad y \]
Secure Computation

[Yao82, Goldreich-Micali-Wigderson-87]

Algorithmically Emulate the Trusted Party
Secure Computation

[Yao82, Goldreich-Micali-Wigderson-87]

\[ z = f(x, y) \]
Secure Computation

[Yao82, Goldreich-Micali-Wigderson-87]

\[ z = f(x, y) \]

Protocol \( \Pi \) securely computes \( f \) if adversary learns same information in left and right worlds.
Secure Computation
[Yao82, Goldreich-Micali-Wigderson-87]

Security defined formally via simulation

\[ z = f(x,y) \]
Secure (Two-Party) Computation

[Yao82, Goldreich-Micali-Wigderson-87]

**Def (Secure Computation):** Protocol securely computes $f$ if for every PPT adversary $A$, there exists a PPT simulator $S$ s.t. for all inputs $(x,y)$ to $f$, and all auxiliary information $z$,

$$\text{View}_{\text{real}}(x,y,z) \sim \text{View}_{\text{ideal}}(x,y,z)$$

where,

- **View$_{\text{real}}$:** = everything seen by $A$ (including input, random tape, aux input and protocol messages) and output of honest party
- **View$_{\text{ideal}}$:** = output of $S$ and output of honest party
Remarks

• Passive vs Active adversaries
  • Passive adversaries follow the protocol. Active adversaries may use arbitrary strategy

• Must modify ideal world to capture active adv
  • $s$ can send any $y^*$ to trusted party
  • $s$ can tell trusted party whether honest party should get output or not