This Lecture:

- Secure Hardware: Intel SGX, HSM
- Blockchain
Outsourcing Computation by FHE

Client

Data x

Key sk

c_t \leftarrow \text{Enc}(x)

\text{ct, f} \longleftarrow \text{ct, f}

c_t' \leftarrow \text{Eval}(f, c_t)

f(x) \leftarrow \text{Dec}_sk(c_t')
Outsourcing Computation by Secure Hardware

Server

Client

Data $x$

Key $sk$

$ct \leftarrow Enc(x)$

$ct, f$

$ct' = Enc_{sk}(y)$

$y = Dec_{sk}(ct')$
What could go wrong?
Intel Software Guard Extension (SGX)

Server

Intel
(mvk, msk)

Client

Attestation

Provisioning

Memory

CPU

\( m \)

\( \text{VK}, 6, 6' \leftarrow \text{Sign}_{sk}(m) \)

\( 6 = \text{Sign}_{msk}(\text{vk}) \)
Constraints & Attacks

- Trust in hardware
- Trust in Intel
- Limited memory size: 128 MB
- Replay attacks
- Side-channel attacks: memory access pattern

\[ \text{fix: Oblivious RAM (ORAM)} \]

\[ \text{overhead } \Theta(\log N) \]

Size of memory
Hardware Secure Module (HSM)

\[ m \xrightarrow{\text{ct} \leftarrow \text{Enc}_K(m)} \text{ct} \]

\[ m \xrightarrow{\text{m} \leftarrow \text{Dec}_K(\text{ct})} \text{ct} \]

\[ \text{ct} \xrightarrow{\text{ct}' \leftarrow \text{Enc}_{K,k'}(m)} k,k' \]

\[ m \xrightarrow{\text{M} \xleftarrow{\text{V} (t)} \text{Ct} \leftarrow \text{V} (t)} \text{ct} \]

\[ m \xrightarrow{\text{m} \leftarrow \text{Dec}_K(\text{ct})} \text{ct} \]
Key Agreement

Sample $k_1, k_2, k_3$ s.t.

$k_1 \oplus k_2 \oplus k_3 = k$

$k := k_1 \oplus k_2 \oplus k_3$

$k_1 \rightarrow \text{FedEx}$

$k_2 \rightarrow \text{UPS}$

$k_3 \rightarrow \text{USPS}$
Transactions in Real Life

Alice → Starbucks $3

Starbucks → Bob $$$

A trusted party that maintains a private ledger
Blockchain

- Public ledger that everyone can view & verify
- Maintained by "miners" in a distributed way

**Step 1:** Charlie wants to make a transaction
- Charlie broadcasts it to the entire network

**Step 2:** All miners collect all transactions in the network
- Verify validity
  - Initiated by sender
  - Enough balance in sender's account
- Agree on next block

**Step 3:** Repeat
Transaction Authentication

Alice: \((VKA, SKA) \leftarrow \text{KeyGen}(1^\lambda)\)

Bob: \((VKB, SKB) \leftarrow \text{KeyGen}(1^\lambda)\)

Charlie: \((VKc, SKc) \leftarrow \text{KeyGen}(1^\lambda)\)

Starbucks: \((VKS, SKS) \leftarrow \text{KeyGen}(1^\lambda)\)

**Bob \rightarrow Charlie B5:**

\[m_1 = (VKB, VKC, 5) \quad g_2 \leftarrow \text{Sign}_{SKB}(m_1)\]

**Charlie \rightarrow Starbucks B3:**

\[m_2 = (VKC, VKS, 3) \quad g_2 \leftarrow \text{Sign}_{SKC}(m_2)\]
Consensus Protocol

TX1 = \text{Charlie} \rightarrow \text{Starbucks} \oplus 3:

\[ m_2 = (vk_c, vk_s, 3) \quad \sigma_2 \leftarrow \text{Sign}_{sk_c}(m_2) \]

TX2 = \text{Charlie} \rightarrow \text{Alice} \oplus 4:

\[ m_3 = (vk_c, vk_a, 4) \quad \sigma_3 \leftarrow \text{Sign}_{sk_c}(m_3) \]

\text{Miner 1} \quad \text{Miner 2} \quad \text{Miner 3} \quad \text{Miner 4}

\text{WANT:} \quad 1. \text{All miners agree on the same block}

\quad 2. \text{New block is valid}

\text{“permissionless”}
Byzantine Agreement

Byzantine Fault Tolerance (BFT) Protocol:

If \( n \geq 3t+1 \),
then it's possible to reach consensus.

Assume \( t < n/3 \), then agree on a valid block.

Any problem?

"Sybil Attack"

(Guaranteed Output Delivery)

Single Adv.
Proof of Work (PoW)

Miner 1:

\[ H(\square) = h \]

\[ \text{Hash} (\begin{array}{c} h \\ \text{TX1} \\ \text{nonce} \end{array}) = 00 \cdots 01011 \cdots 0 \]

\[ \text{Prob.} = \frac{1}{2^{30}} \Rightarrow 2^{30} \text{ attempts} \]

Find nonce s.t. Hash (block) has \( \geq 30 \) leading 0's.

Consensus Protocol:

Whoever first finds a block that hashes to a value w/ \( \geq 30 \) leading 0's, that block becomes the next block.
Proof of Work (PoW)

Miner 1

Miner 2

Longest Chain Rule: Always adopt the longest chain.

Assuming honest majority of computation power, the longest chain is always valid.
Blockchain

- Efficient verification of sufficient balance: Merkle Tree

- Settlement of a transaction:
  - Included in a block which is ≥ 6 blocks deep (~1 hr)

- Dynamically adjust # leading 0's s.t. each block takes ~10 min to mine
  - Last 1 hr: > 6 blocks: increase # leading 0's
  - < 6 blocks: decrease # leading 0's

- Miners' motivation:
  - transaction fee
  - new coin generated in each block goes to miner

- Extensions
  - Proof of Stake (PoS)
  - Anonymous transactions (zk-SNARGs)
  - Smart Contracts
  - Public Bulletin Board