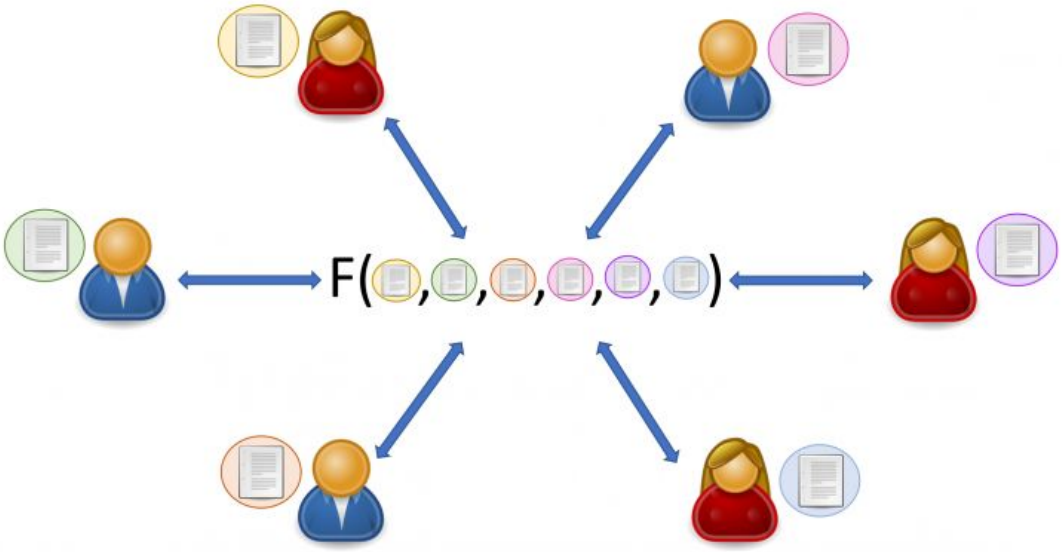
CSC 116
Secure Multi-Party Computation



Secure Multi-Party Computation (SMPC) allows multiple parties to jointly compute a function on their private inputs without revealing those inputs to each other, ensuring privacy and security during the process.

Case 1

Scenario: "Three hospitals want to calculate the average patient age for a study, but they **can't share their patient data** due to privacy laws (HIPAA)."

Problem: Sharing raw data could expose patients' information.

Solution: Secure Multi-Party Computation allows them to **compute together without sharing** individual data.

Case 2

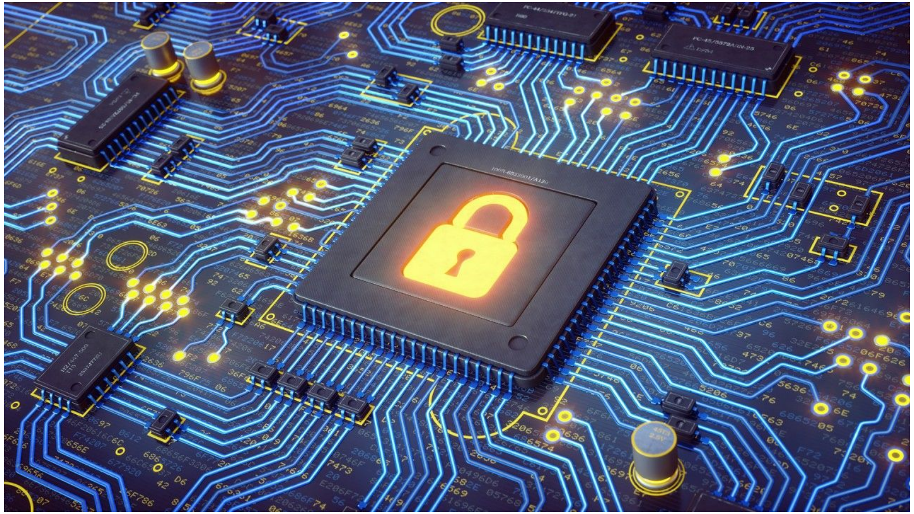
10 9 12 11 0 1000

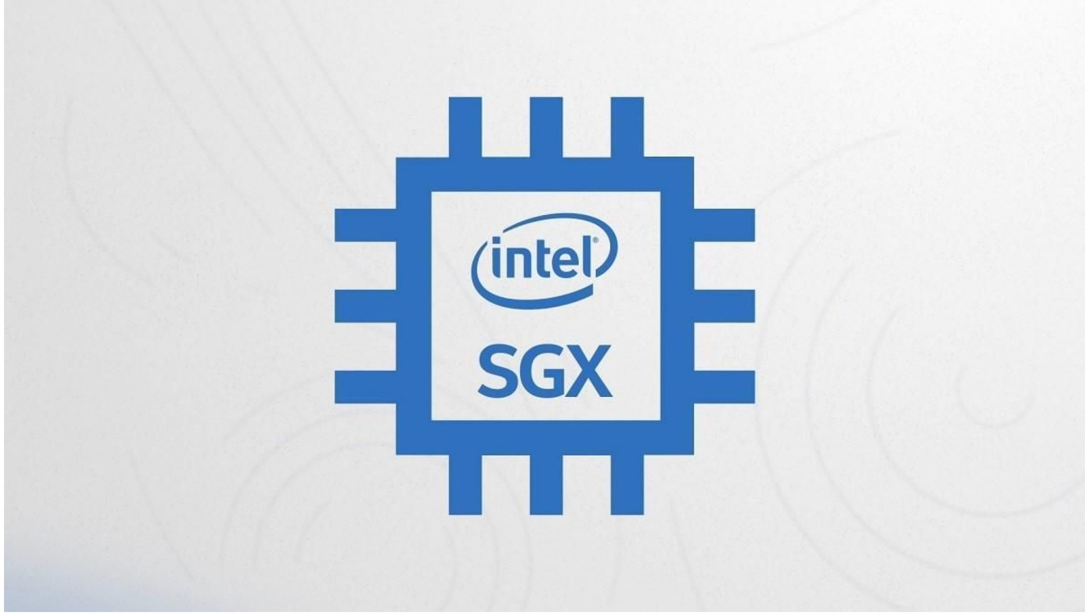
Value =（10 + 9 + 12 + 11 + 0 + 1000）/ 6

Do you trust this value？

What is a Trusted Execution Environment (TEE)?

A **Trusted Execution Environment (TEE)** is a **secure area** inside a computer's processor (CPU) that runs code **separately from the main operating system**. It protects sensitive data and computations even if the main system is hacked or compromised.





Intel® Software Guard Extensions

Case 2

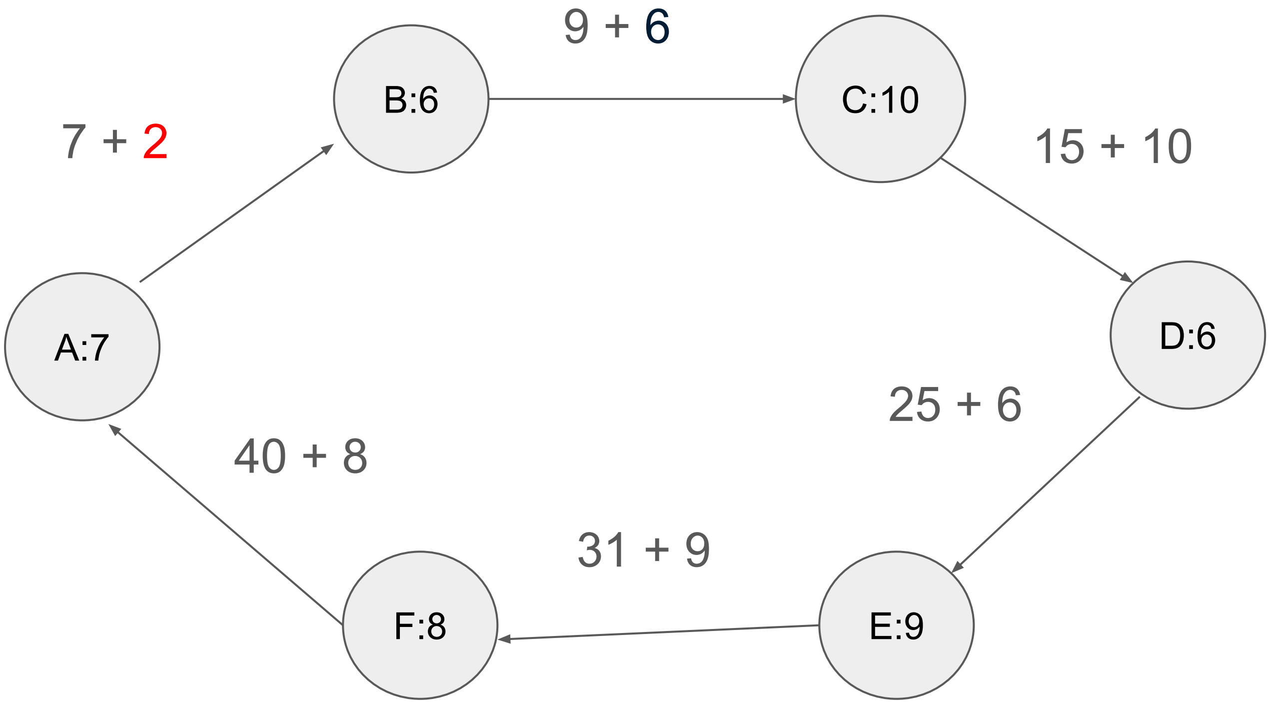
10 9 12 11 0 1000

Value = (10 + 9 + 12 + 11 + 0 + 1000) / 6

Only SGX can commuicate with SGX

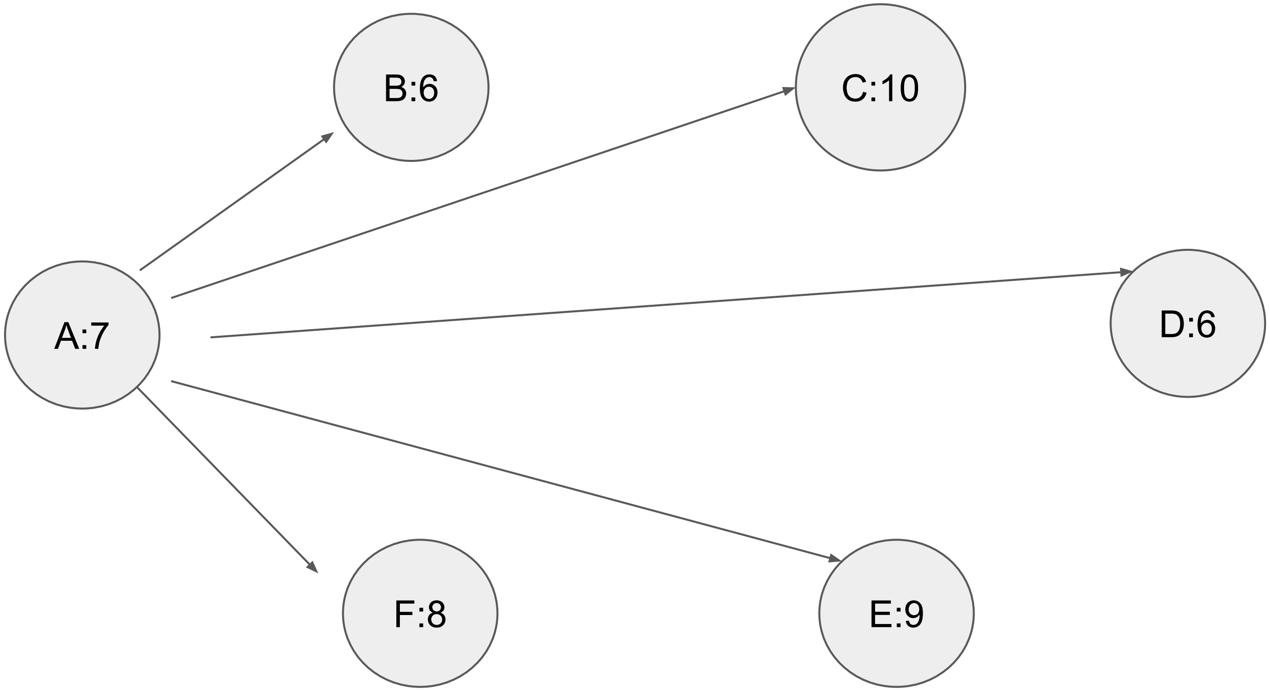
No one understands what is running inside of the SGX, only the programmer knows the codes

Case 3: If there is no neutral server

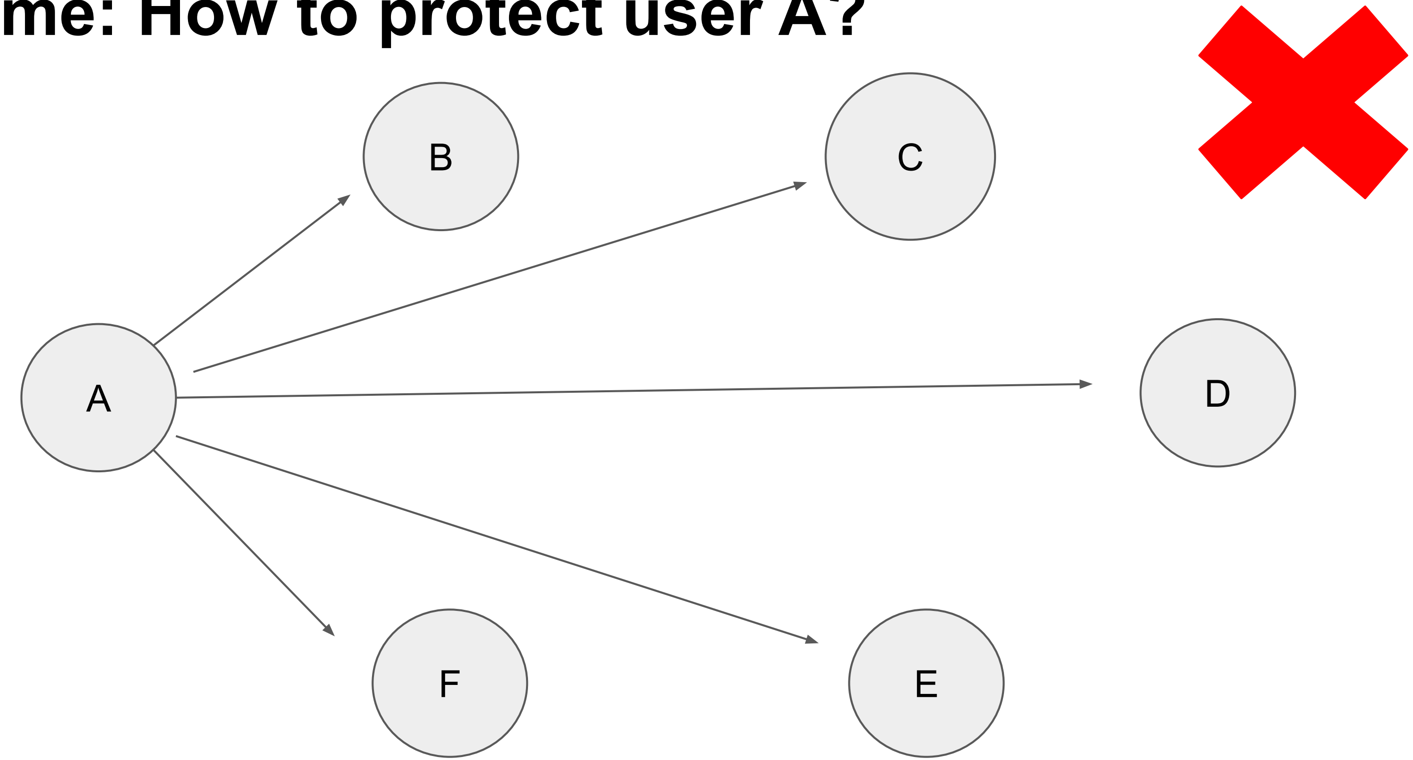


Case 3: If there is no neutral server

48 - 2
= 46

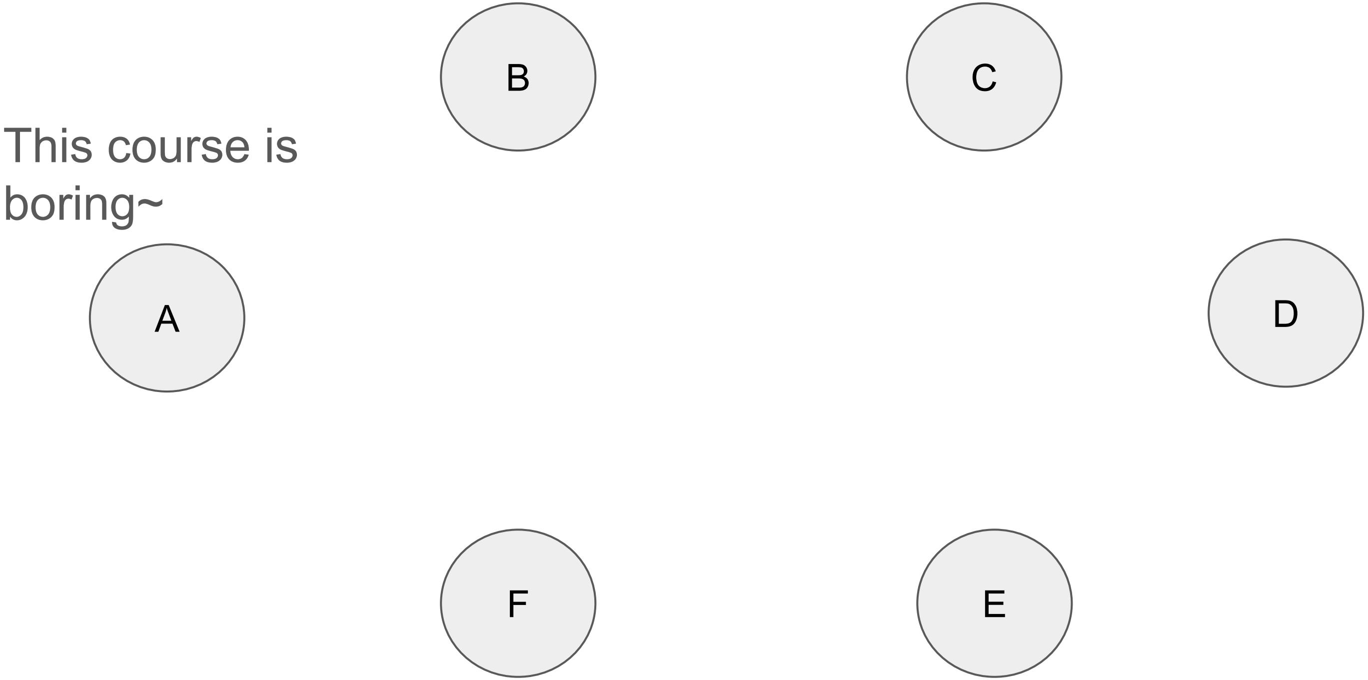


Case 4:
Game: How to protect user A?



Case 4: How to protect User A?

Case 4: How to protect User A?



Scenario: Anonymous Reporting with 5 People

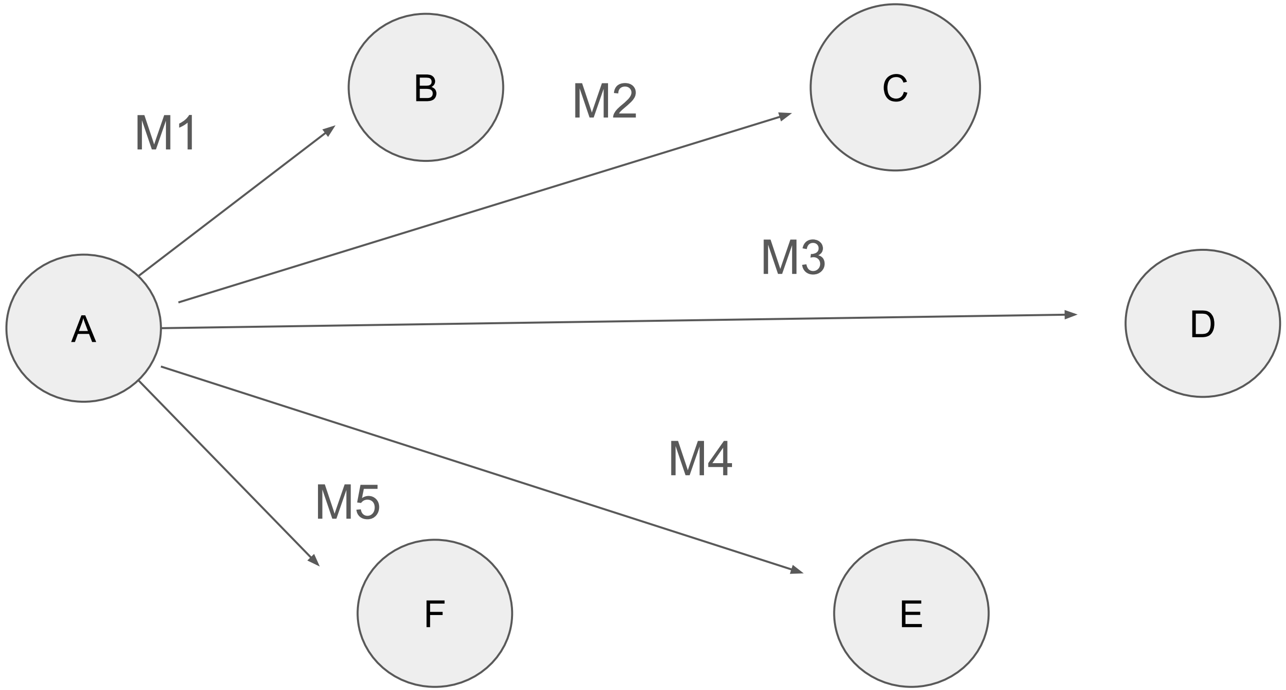
Imagine:

- One person (say, Alice) wants to **report a problem** (like “This course is boring”).
- Alice doesn’t want others to know she said it.
- But everyone should **receive the message**.

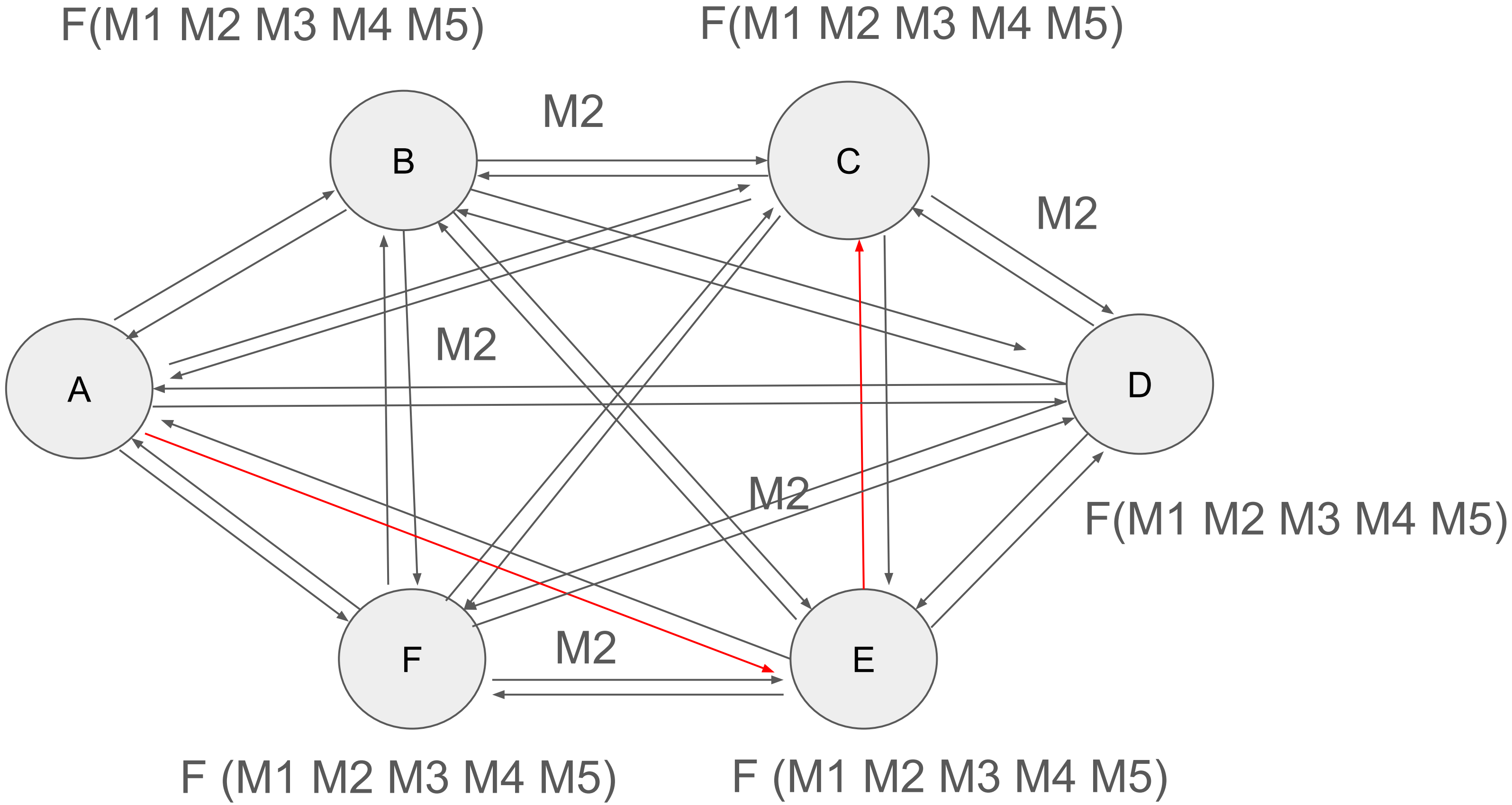
Split the Secret into Pieces

- **Spliting the secret message** into 5 random-looking parts: M1, M2, M3, M4, M5.

- The trick: These parts **add up to the full message**, but **each part alone looks like nonsense**.



B
M1
M2
C
M3
A
D
M4
M5
F
E


F(M1 M2 M3 M4 M5)
F(M1 M2 M3 M4 M5)
M2
B
C
M2
M2
A
D
M2
F(M1 M2 M3 M4 M5)
M2
F
E
F (M1 M2 M3 M4 M5)
F (M1 M2 M3 M4 M5)

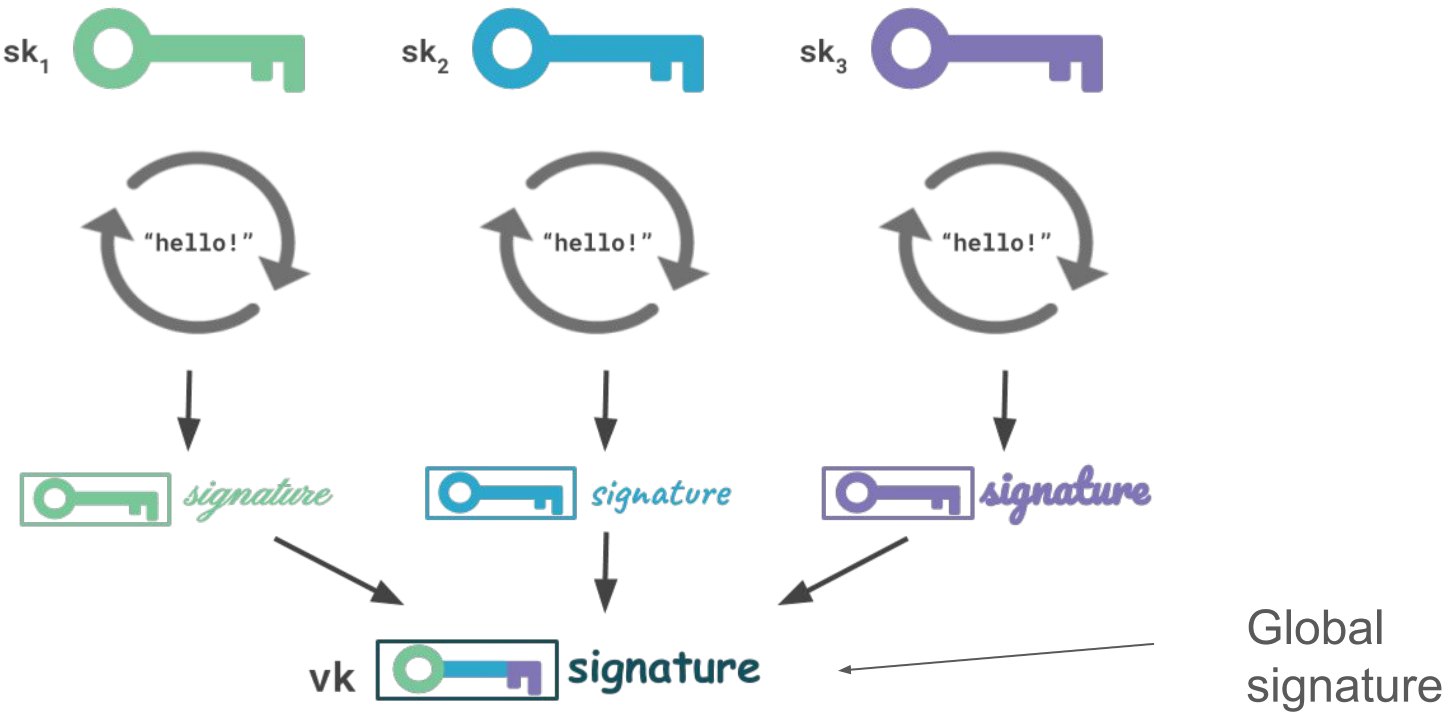
Function 1: split the secret into different parts

Function 2: Combine all the parts and decrypt the message.

Threshold Signatures!!!!

Threshold Signatures!!!!

Threshold signatures are a cryptographic method where a digital signature is generated by multiple parties, requiring a certain number (the "threshold") of them to cooperate, rather than relying on a single private key.

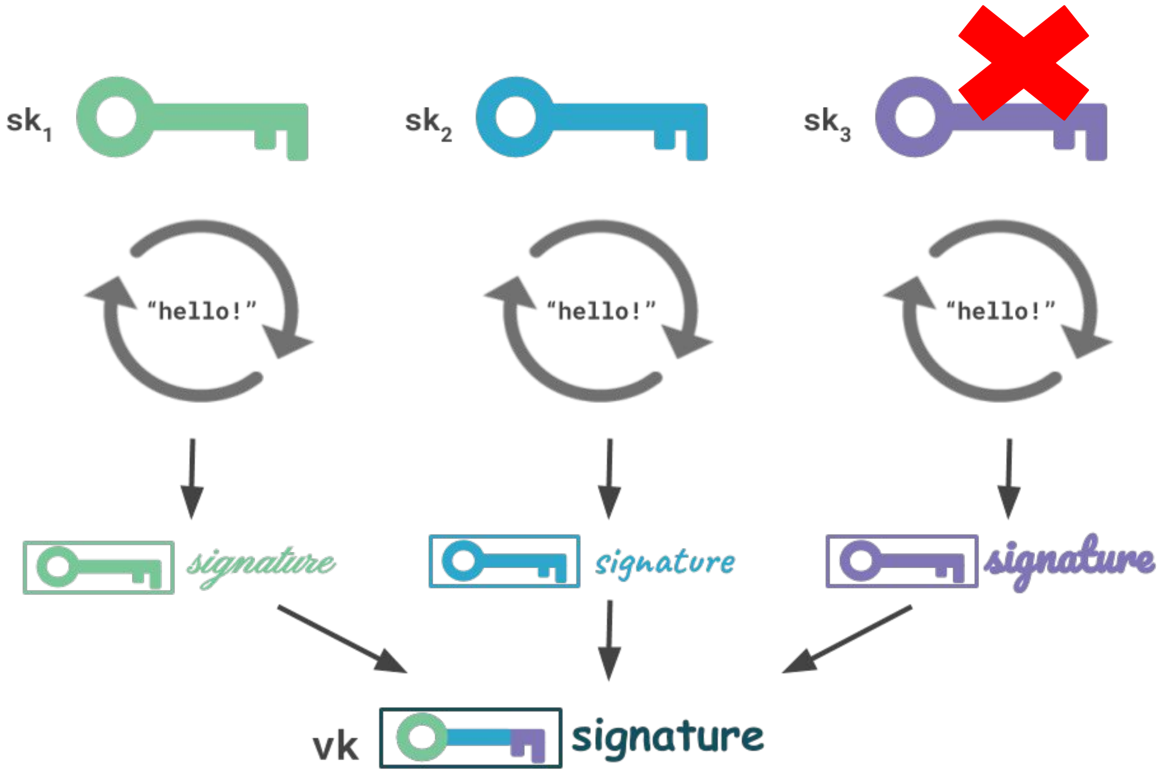


Threshold is dynamical: majority of the users or half

Function 1: Key generations: Private keys, public keys

Function 2: Combine all the signatures into one.

The global signature is the final key to assign an agreement



sk1
sk2
sk3
"hello!"
"hello!"
"hello!"
signature
signature
signature
vk
signature

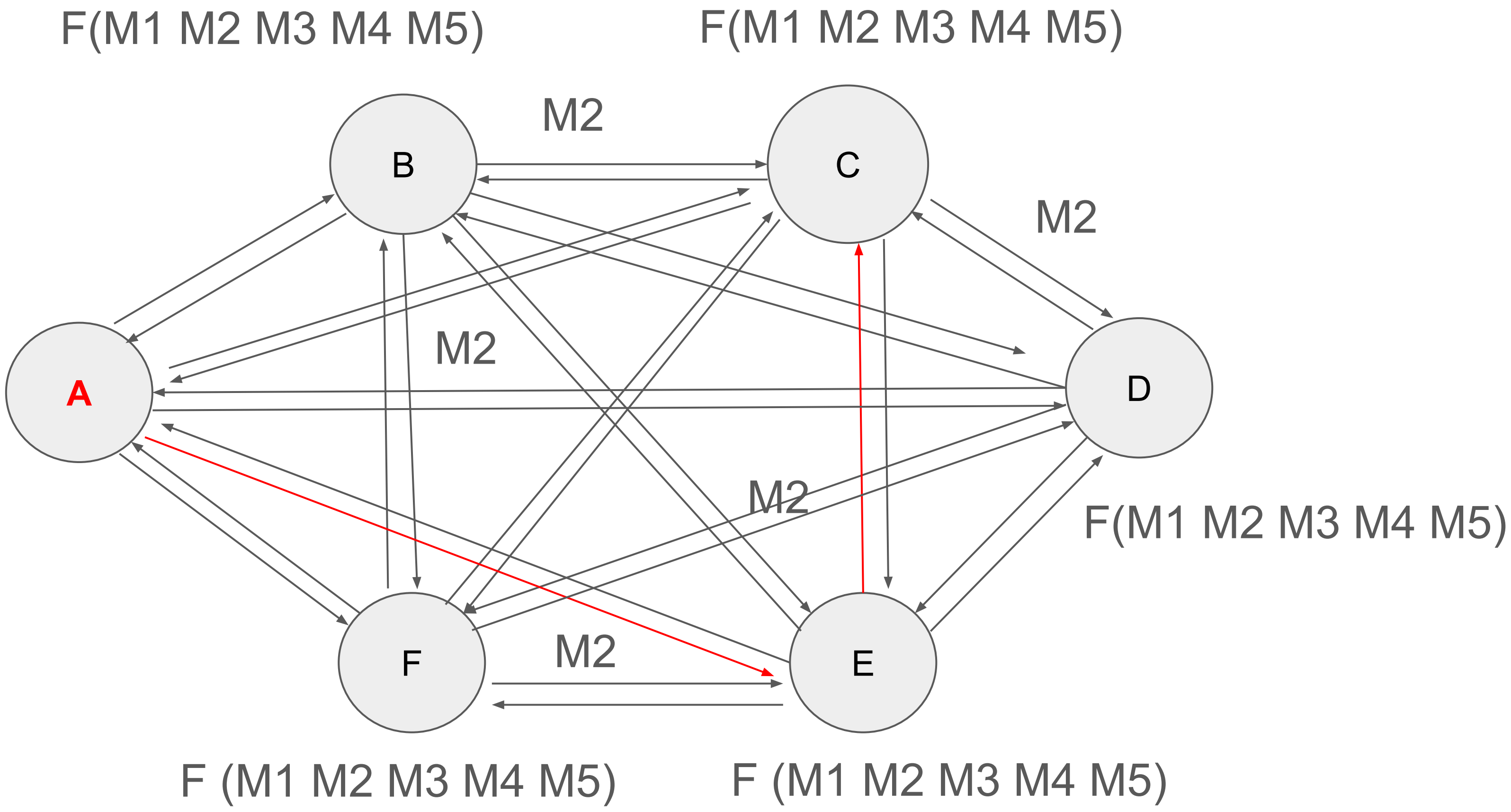
Features:

1. It is **decentralized** signatures
2. Avoids **single point of failure** (no one person has the full private key)
3. It protects the secrets by multiple users: **security and confidentiality.**

Features:

1. **It is decentralized signatures**
2. **Avoids single point of failure (no one person has the full private key)**
3. **It protects the secrets by multiple users: security and confidentiality.**

Case 4


F(M1 M2 M3 M4 M5)
F(M1 M2 M3 M4 M5)
M2
B
C
M2
M2
A
D
M2
F(M1 M2 M3 M4 M5)
M2
F
E
F (M1 M2 M3 M4 M5)
F (M1 M2 M3 M4 M5)

Short summaries

Multi-party computations
1, ask trusted third users to help (new security problem)
2, without third party user.

Threshold Signatures
1, Descentralized signatures.

Short summaries

Multi-party computations
1, ask trusted third users to help (new security problem)
2, without third party user.

Threshold Signatures
1, Descentralized signatures.

Applications

Example 1: Protecting Patient Records (MPC)

Scenario: Three hospitals (A, B, and C) each have part of a patient's medical data. They want to **analyze the patient’s overall health** (e.g., detect early signs of disease) **without sharing their full data** with each other.

Solution with MPC: Using Multi-Party Computation, these hospitals can run a joint program where **no one sees the full data**, but they still get the **same result** as if they had all the data in one place.

Example 2: Accessing Critical Patient Info (Threshold Signatures)

Scenario: A patient's medical record is encrypted for privacy. To unlock it, at least **3 out of 5 senior doctors** need to approve the request.

Solution with Threshold Signatures: Each doctor holds a **piece of the digital key**, and only when 3 or more agree, they can **combine their pieces to unlock** the record.

Example 3: Secure Medication Approval (Threshold Signatures)

Scenario: In an ICU, high-risk medications (e.g., controlled substances) require approval from multiple healthcare professionals before they can be administered.

Solution with Threshold Signatures: The hospital's system uses a digital approval process where **a medication order needs at least 2 out of 3 signatures** (e.g., from a doctor, pharmacist, and nurse supervisor) to proceed.

Example 4: Research Collaboration Without Sharing Raw Data (MPC)

Scenario: Multiple hospitals want to work together on cancer research using patient data — but due to privacy laws, **they can’t share the actual patient records**.

Solution with MPC: Each hospital keeps its data private but runs a joint computation (like calculating cancer rates or risk factors) using MPC. They get useful research results **without ever exchanging patient files**.

https://youtu.be/-1H1Sp-_5YU

Differential Privacy

Differential privacy is a privacy-enhancing technology that protects individual data by adding controlled randomness to data analysis

Scenario: 5 users' ages:

`[23, 25, 30, 40, 35]`

Average = (23 + 25 + 30 + 40 + 35) / 5 = **30.6**

Each user adds a small random noise (between -2 and +2) before submitting:

Example: `[22, 27, 28, 39, 37]`

New average = (22 + 27 + 28 + 39 + 37) / 5 = **30.6**

Application 1: Privacy-Preserving Patient Age Analysis

Scenario:
Three hospitals want to **calculate the average patient age** for diabetic patients to adjust medication guidelines — but **cannot share raw patient data** due to HIPAA.

With Differential Privacy:
Each hospital adds a small random noise to the patient ages **before** sharing.

Example (simplified):

- Hospital A has ages: `[65, 67, 70]` → adds noise: `[66, 65, 69]`
- Hospital B has `[60, 64]` → adds noise: `[61, 66]`
- Hospital C has `[72, 74]` → adds noise: `[73, 72]`

All noisy values are sent to the central server for averaging:

- Average = (66+65+69+61+66+73+72) / 7 = ~67.4

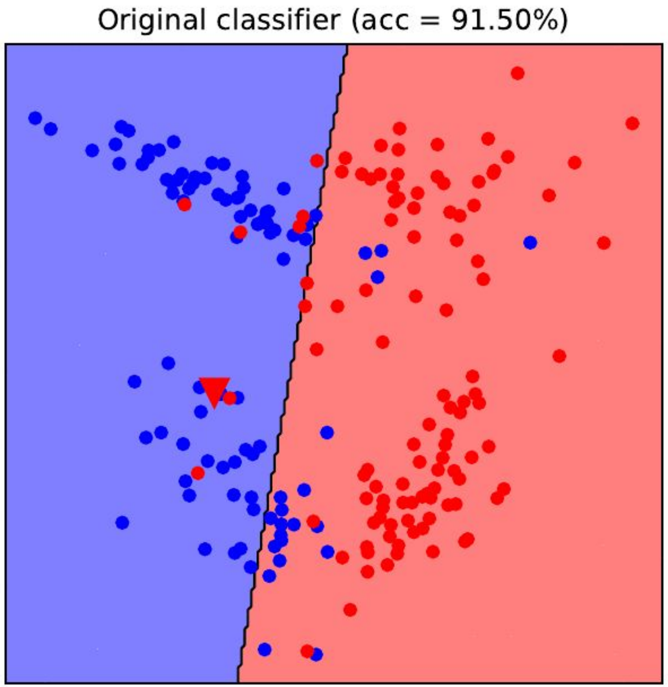
The system learns the average age across all hospitals **without ever seeing real patient ages**.

https://www.youtube.com/watch?v=gI0wk1CXIsQ

Poisoning attacks

Poisoning attacks, involve manipulating the training data of machine learning models to corrupt their behavior and lead to biased or harmful outputs.

Poisoning attacks



Thanks