

Cipbersecurity in Healthcare

Zero-Knowledge Proof (ZKP)

Zero-Knowledge Proof (ZKP)

A zero-knowledge proof (ZKP) is a cryptographic method that allows one party to convince another party that a statement is true, without revealing any information beyond the truth of the statement itself.

Case 1

Someone says: “**I live in Tampa!**”

What can we infer?

- They are **likely from Florida** (Tampa is in Florida).
- They **may also be a U.S. citizen**, but we don't know for sure.

The person hasn't revealed **any document or ID** to **prove citizenship or state identity**.

Case 1

Question: What if that person could prove they are from Florida, or even prove they are a U.S. citizen, **without** ever saying 'I live in Tampa' or showing their address or passport?



Zero Knowledge Proof

Case 2

A patient visits a hospital for a medical procedure that is only allowed for patients above 18 years old

Normally, the hospital would ask for:

- ID card
- Full date of birth
- Insurance details, etc.



This **reveals a lot of personal data** that isn't necessary for just proving date birth.

Summary:

Zero-Knowledge Proof allows the users to prove themselves without exposing private data.

It protects user privacy and reduces data leakage risks

Demo 1

Me:

**A ZKP for proving knowledge of a
secret number $x = 4$,**

Random number $r = 3$

$r = 3$

You:

Generate a random value $c = 5$

Send to me!

Me:

$$s = r + c * x = 3 + 5 * 4 = 23$$

SEND 23 to you.

YOU:

s=23

x=4

r=3

c=5

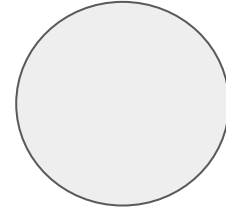
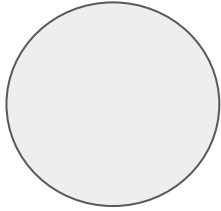
$r+x*c=23$

So you understand me that I know it is 4.

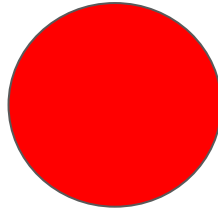
Demo 2: string

Question: If it is not numbers?

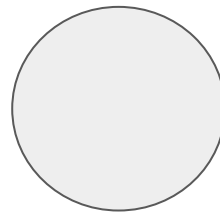
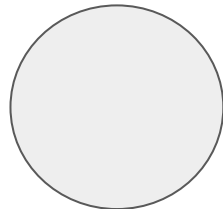
I know a secret (or
password), and its hash is
 $H(x)$



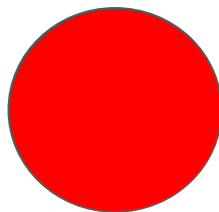
Listening



I know HOW OLD ARE
YOU. and its hash is $H(x)$



Listening



Question: Can we send this?

Applications of Zero-Knowledge Proof (ZKP) in Hospitals

Patient Identity Verification (Without Revealing Sensitive Information)

- Patients can prove they are **registered users, insured, or eligible for treatment**, without disclosing:
 - Name
 - ID number
 - Insurance details
- ✓ "I am a verified patient in the system."

Vaccination Status Proof (Without Revealing Medical History)

- Patients can prove they are **vaccinated**, without revealing:
 - When
 - Where
 - By whom
- Applicable in:
 - Admission protocols
 - Pre-surgery checks
 - Hospital staff onboarding

Consent Proof for Data Access (Digital Authorization + ZKP)

- Patients can issue **digital consent proofs** saying:

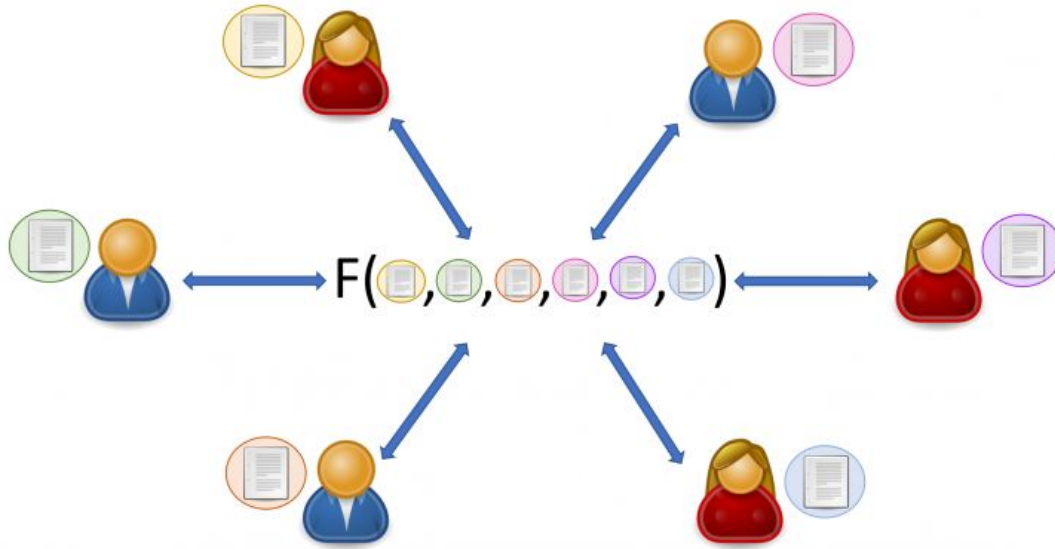
"Doctor A can access my data for the next 7 days,"
without disclosing the full content of the consent
form.
- Helps protect patient rights while maintaining access
control.

Applications of Zero-Knowledge Proof (ZKP) in finance

Private Transactions (Confidential Payments in Blockchain)

- ZKP enables **transactions to be verified without revealing amounts or parties involved.**
- Used in **privacy coins** like **Zcash**, where:
The network verifies the transaction is valid, but no one knows how much was sent or who the sender/receiver is.
- Applicable in:
 - Private asset transfers

Secure Multi-Party Computation (SMPC)



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.

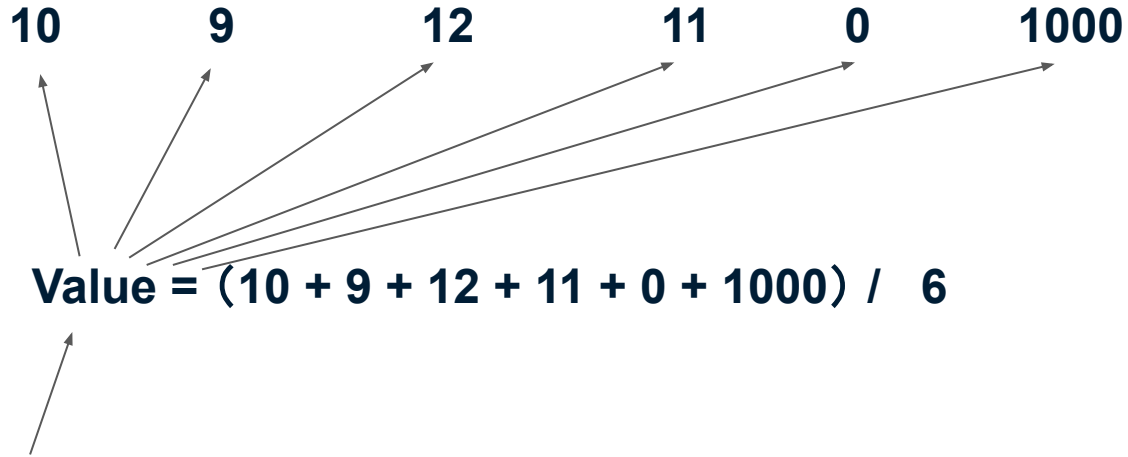
Case 1

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

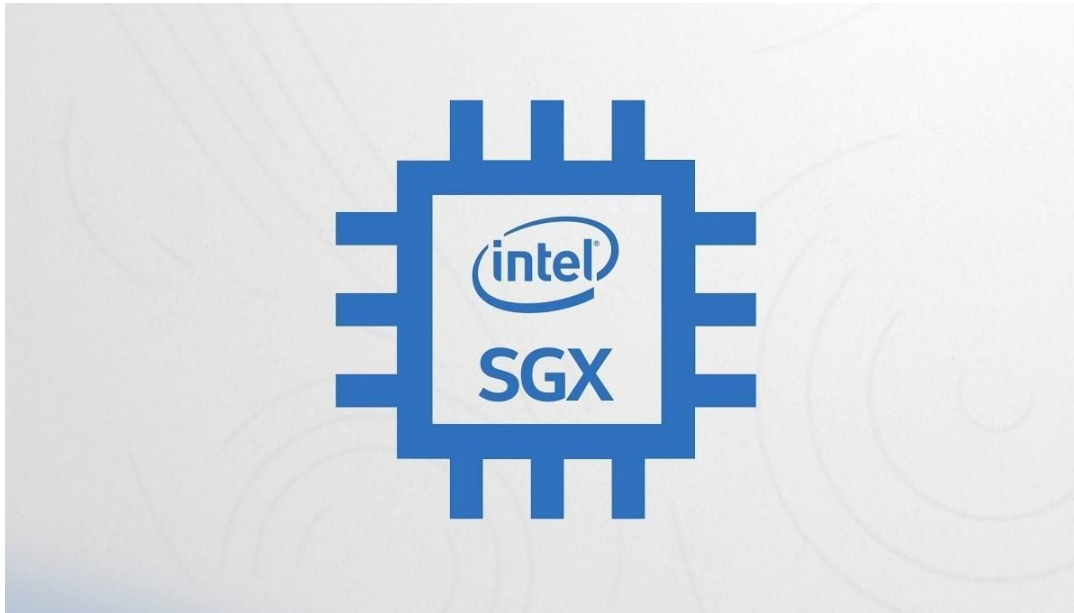


Do you trust this value ?

What is a Trusted Execution Environment (TEE)?

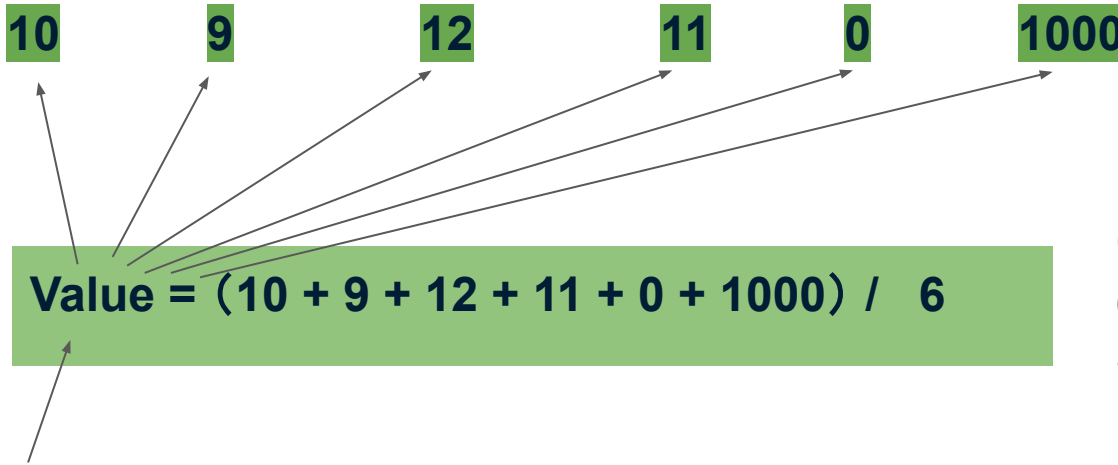
A **Trusted Execution Environment (TEE)** is a **secure area** inside a computer's processor (CPU) that runs code **separately from the main operating system**. It protects sensitive data and computations even if the main system is hacked or compromised.





Intel® Software Guard Extensions

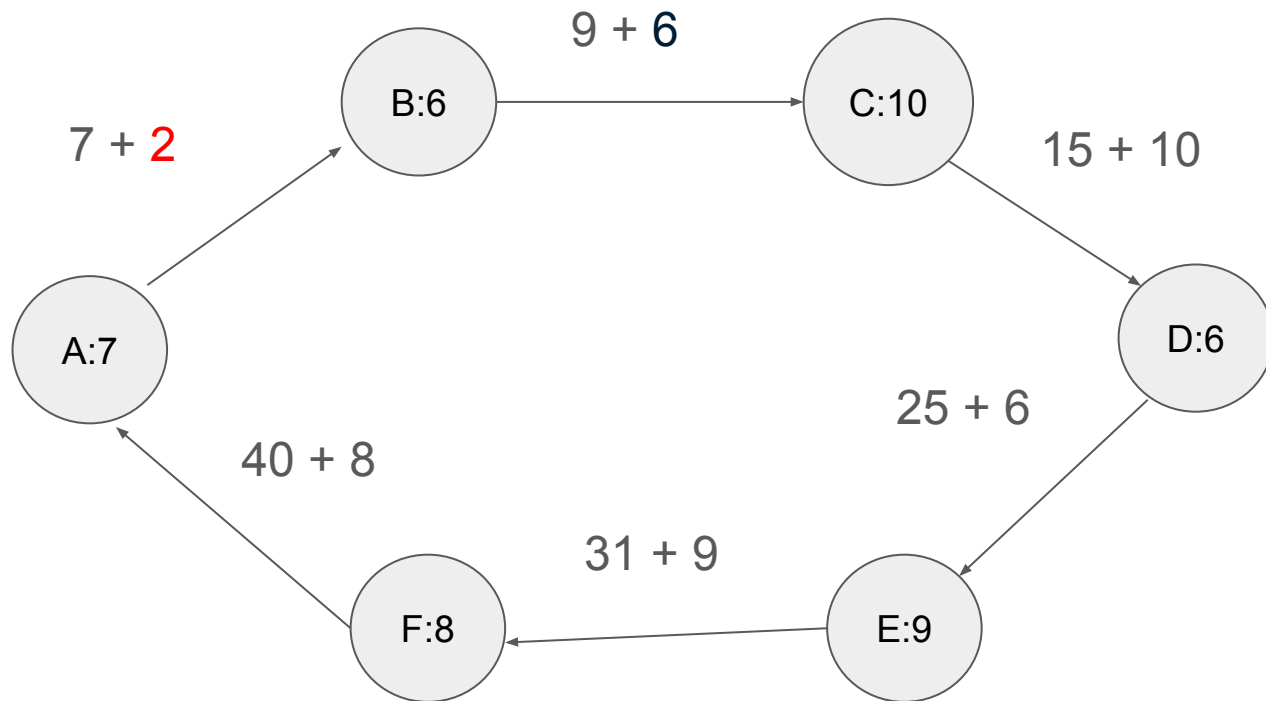
Case 2



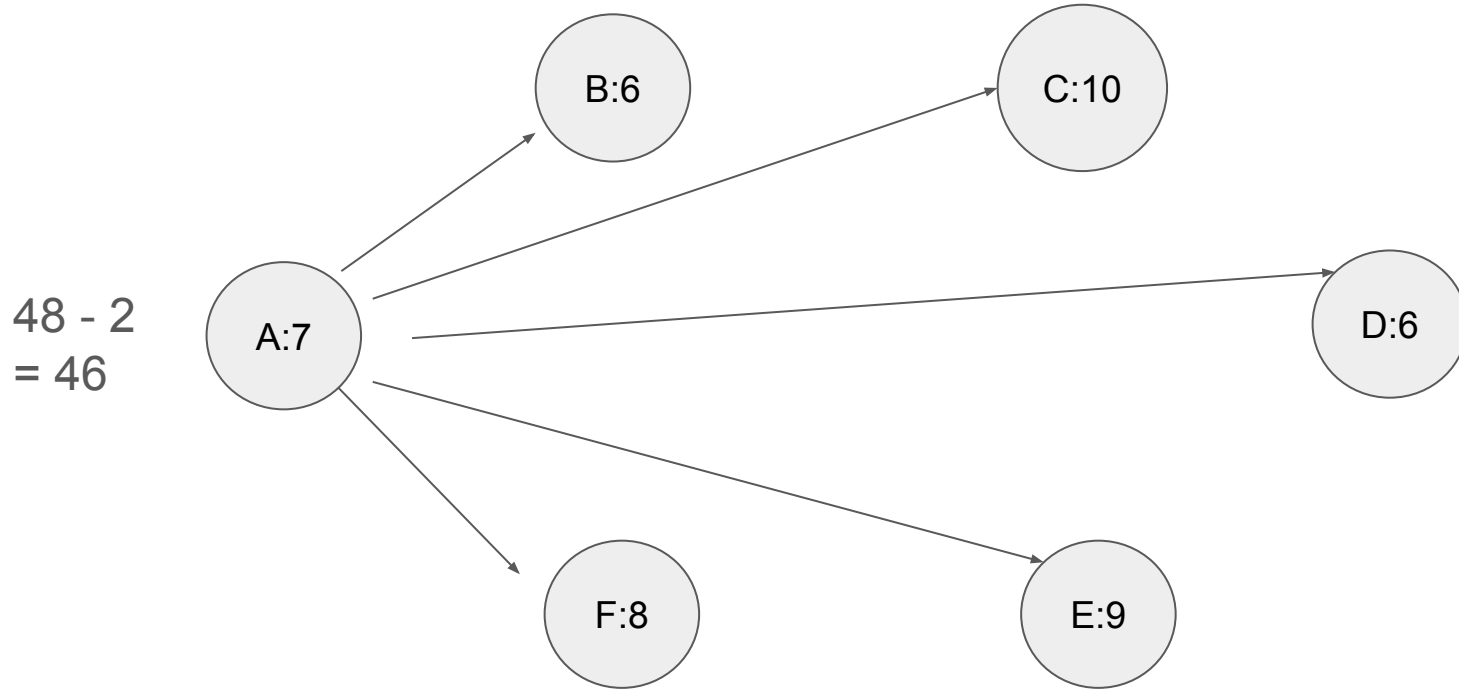
Only SGX can
commuicate with
SGX

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

Case 3: If there is no neutral server

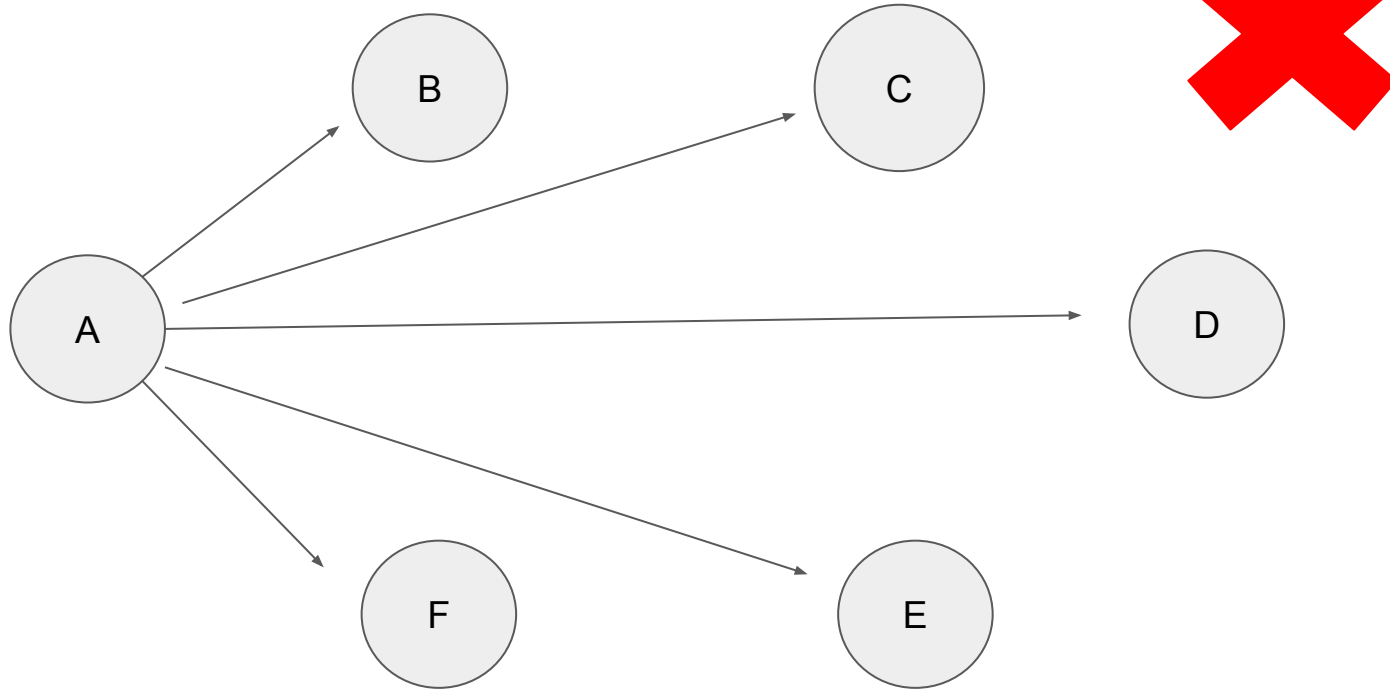


Case 3: If there is no neutral server

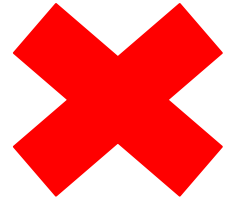


Case 4:

Game: How to protect user A?

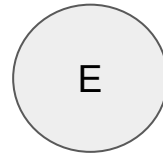
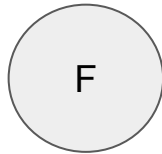
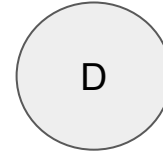
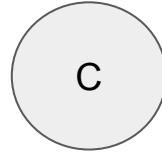
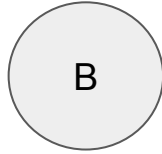
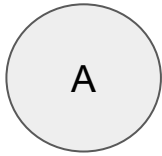


Case 4: How to protect User A?



Case 4: How to protect User A?

This course is
boring~



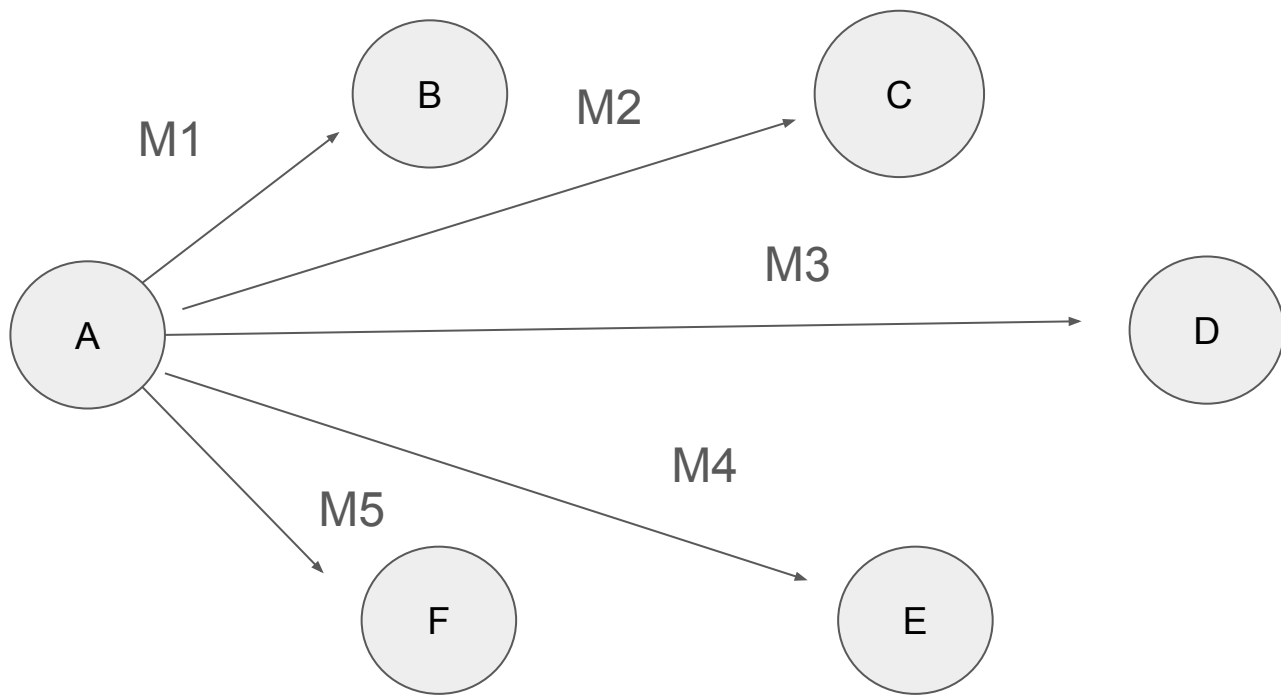
Scenario: Anonymous Reporting with 5 People

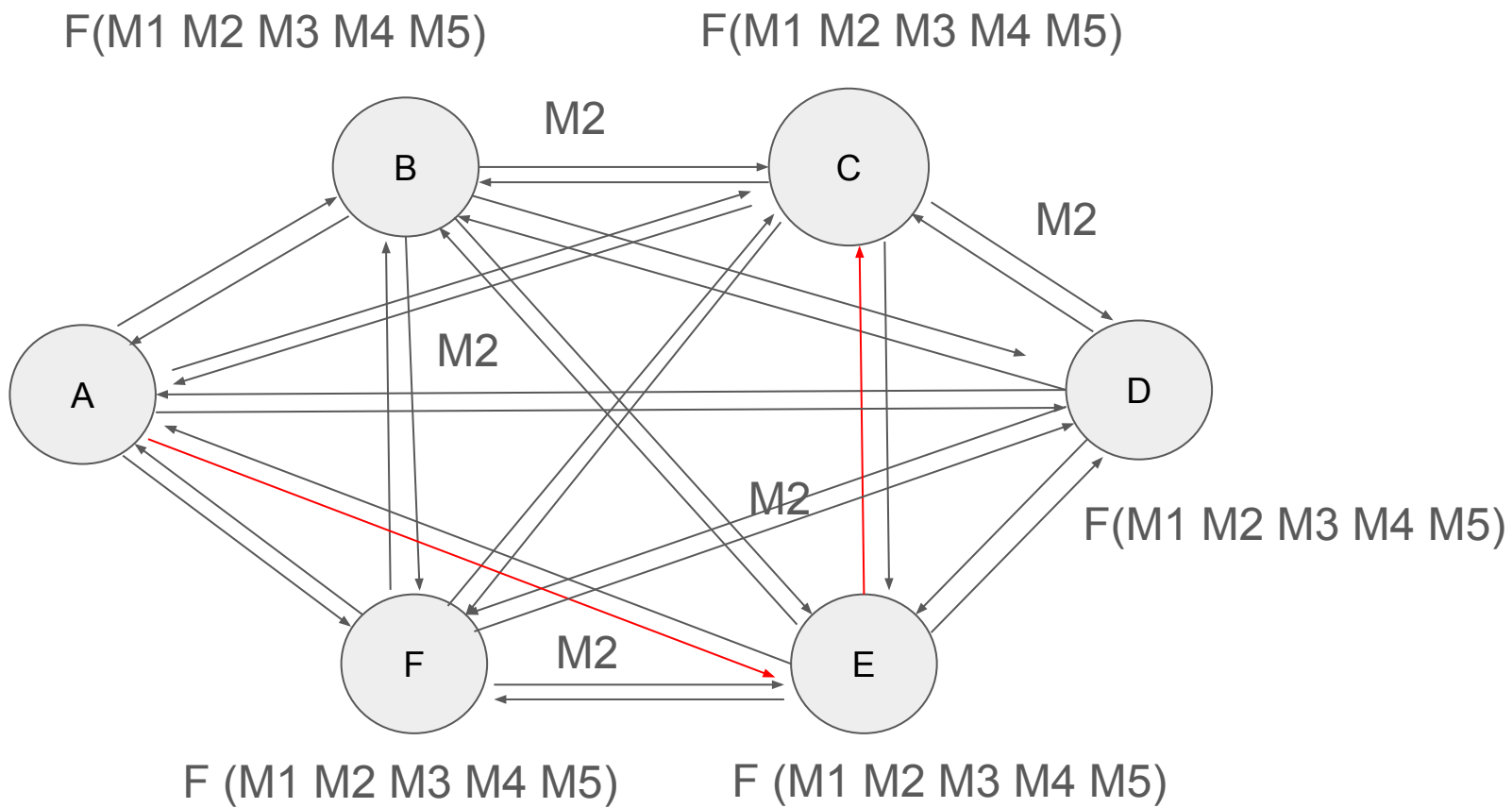
Imagine:

- 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

- **Splitting 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**.

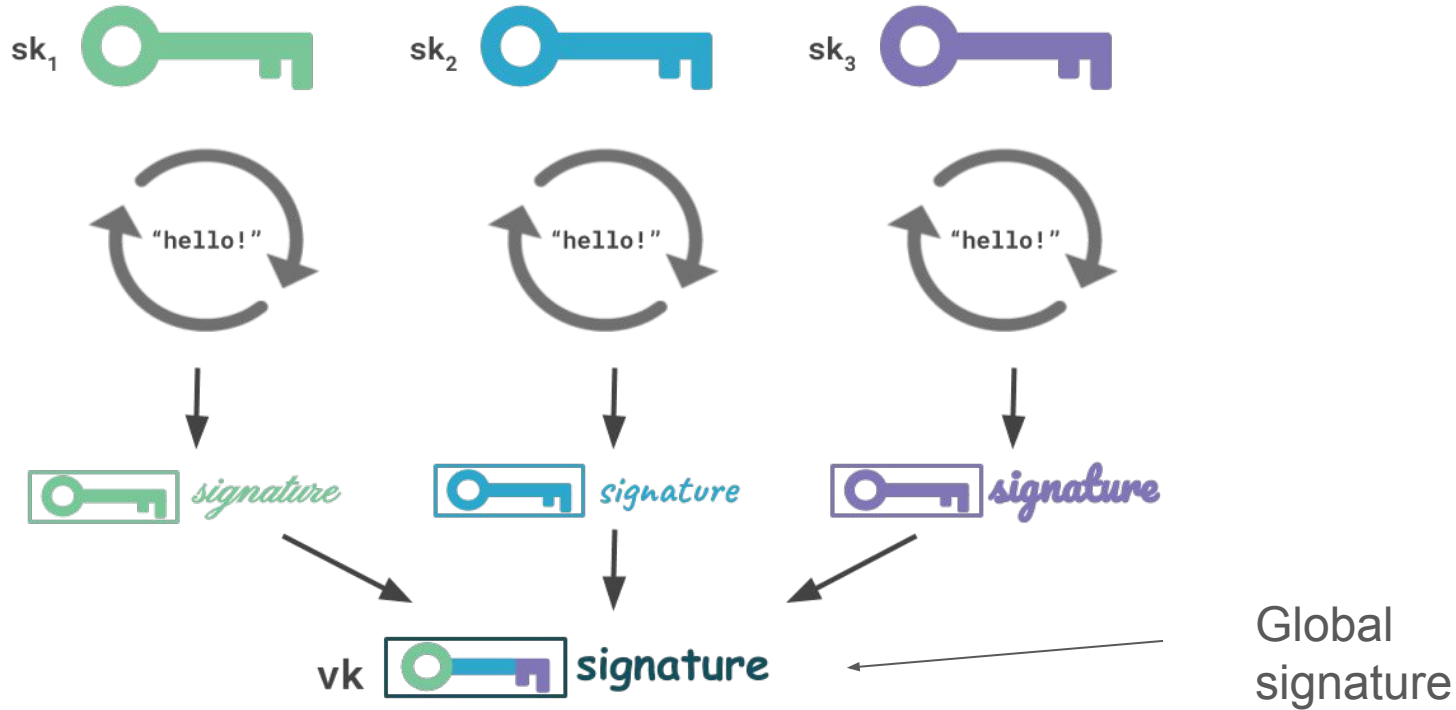




Threshold Signatures

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.

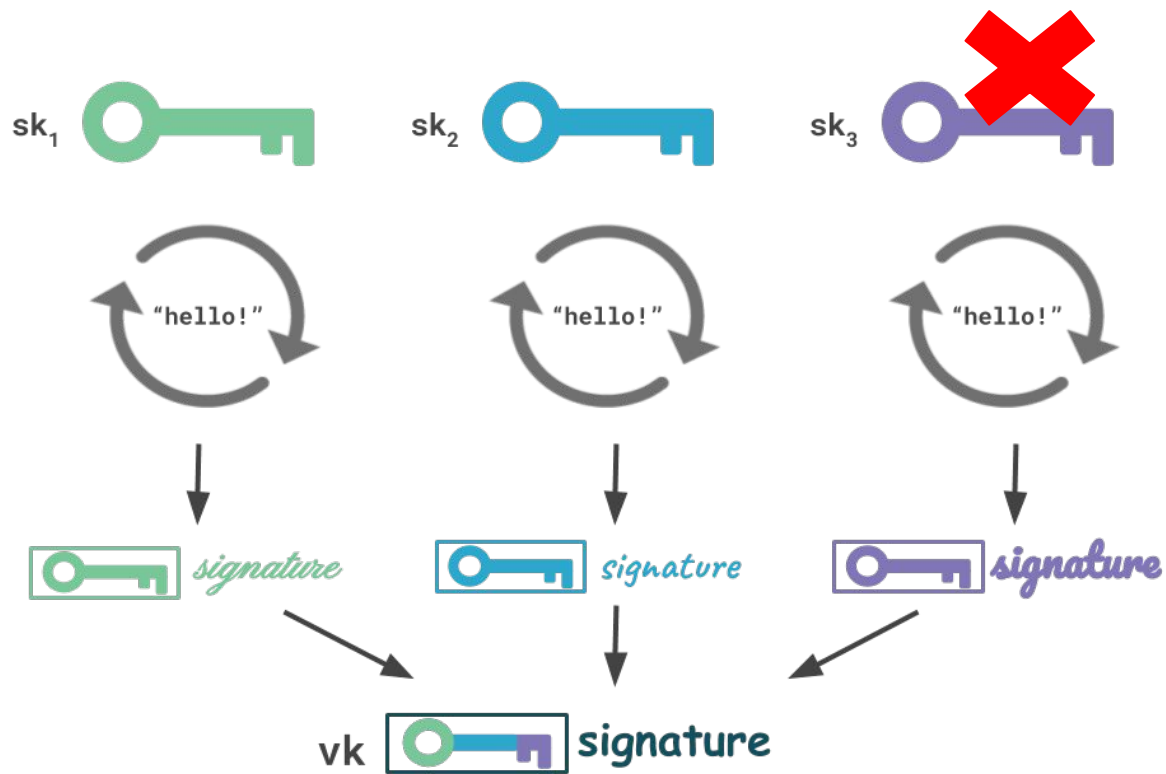


Threshold is dynamical: majority of the users or half

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



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.**

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]

$$\text{Average} = (23 + 25 + 30 + 40 + 35) / 5 = \mathbf{30.6}$$

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

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

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

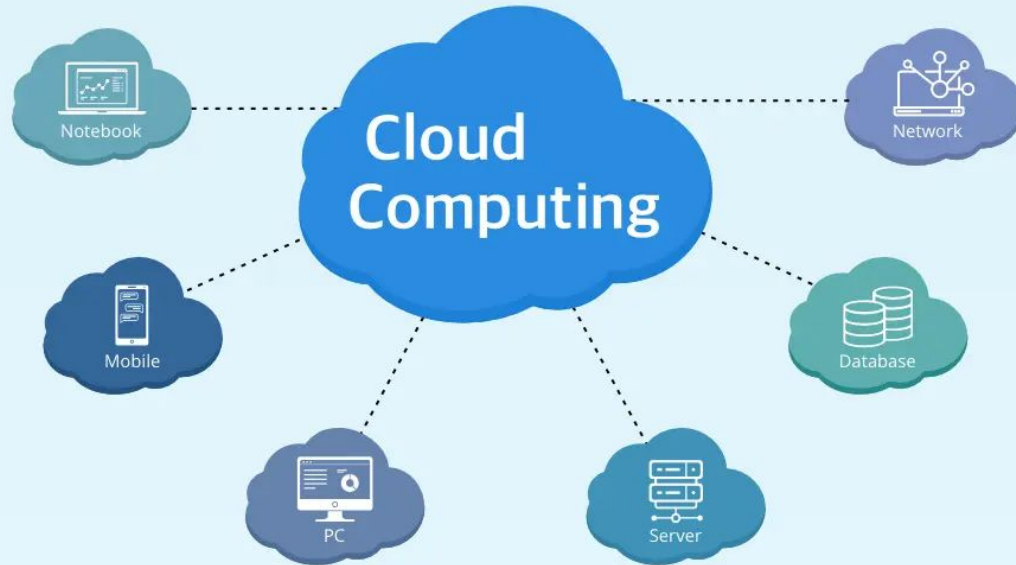
Homomorphic Encryption

Homomorphic Encryption is a type of encryption that allows computations to be performed directly on encrypted data **without decrypting it first.**

Why it's important:

- Enables **privacy-preserving computation.**
- Useful in **cloud computing**, where a server can compute on encrypted data without learning the actual data.
- Vital for **secure data analysis, machine learning, finance,** and **healthcare** where sensitive data is involved.

Cloud data is not secure

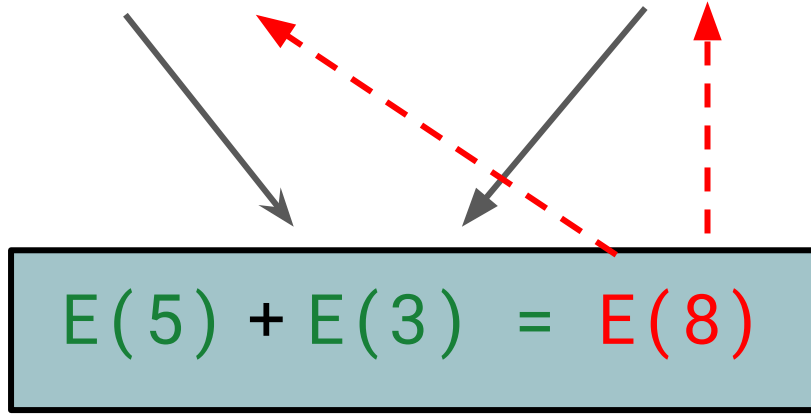


Example:

Imagine you encrypt the numbers 5 and 3. A server can **add the encrypted values**, and when you decrypt the result, you get 8 — **without the server ever knowing what 5 or 3 were.**

Encrypt(5)

Encrypt(3)

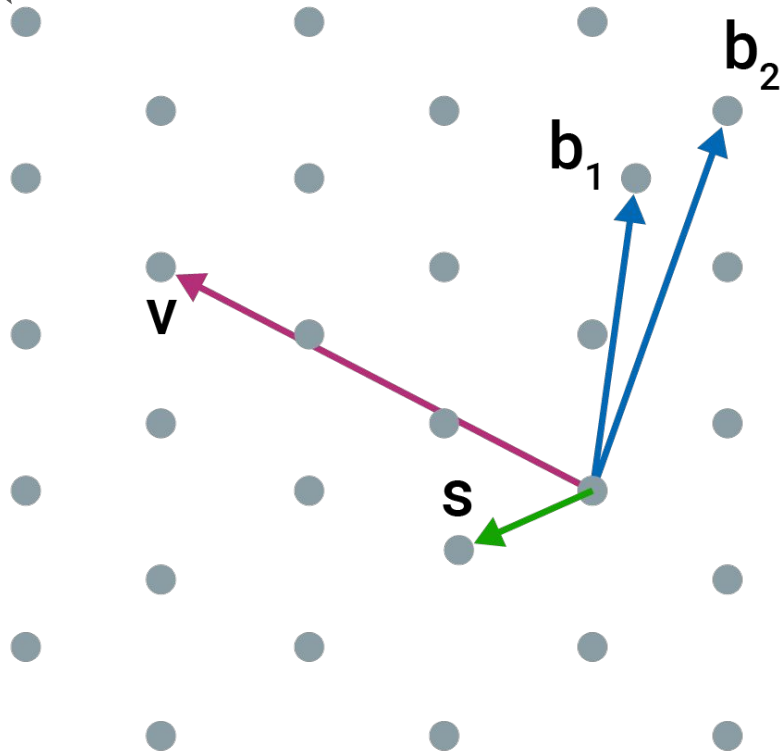


Remote server

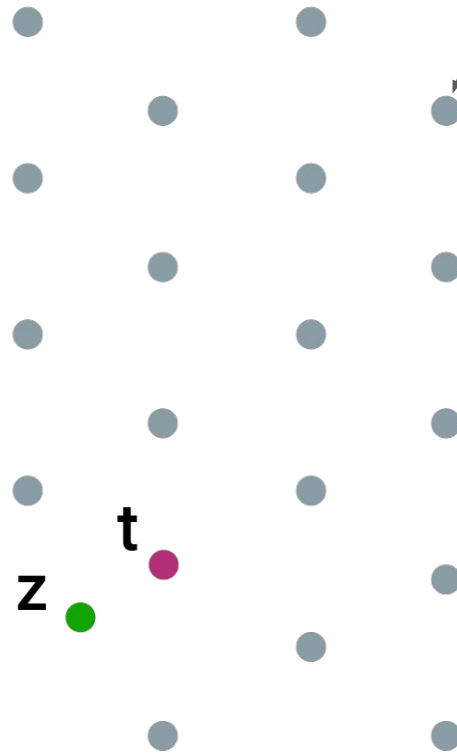
Lattice-based Cryptography

Lattice-based cryptography is a type of encryption that uses complex geometric structures called **lattices** to secure data. It is considered one of the most promising forms of **post-quantum cryptography** — meaning it's designed to be secure even against **quantum computers**.

Lattice



Lattice



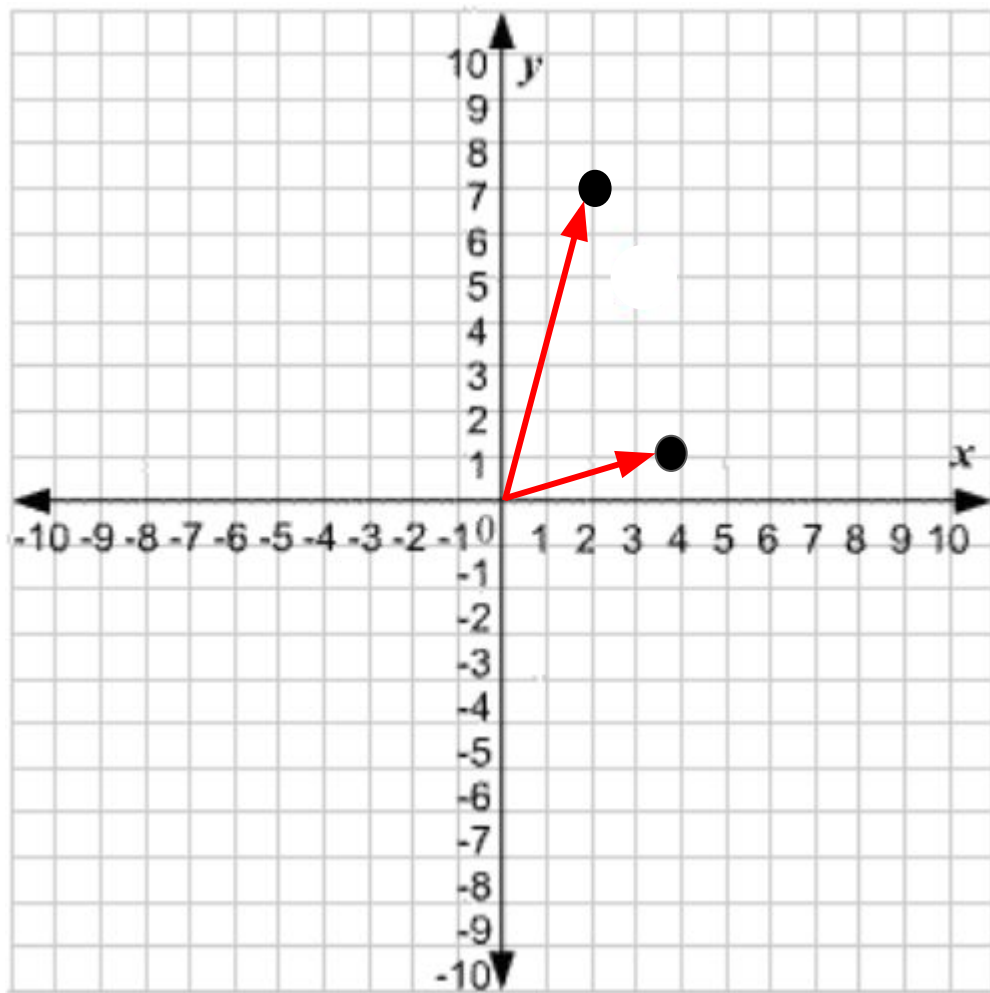
Private key = [3, 5]

$S = a_1 * s_1 + a_2 * s_2 + \text{error (differential privacy)} \bmod 11$

Random A = [a1, a2]

$A_1 = [2, 7] \quad A_1 = 2 * 3 + 7 * 5 + 1 = 42 \bmod 11 = 9$


$A_2 = [4, 1] \quad A_2 = 4 * 3 + 1 * 5 - 1 = 16 \bmod 11 = 5$



$$A1 = [2, 7] \quad A1 = 2 * 3 + 7 * 5 + 1 = 42 \bmod 11 = 9$$

$$A2 = [4, 1] \quad A2 = 4 * 3 + 1 * 5 - 1 = 16 \bmod 11 = 5$$

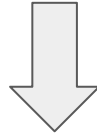
Pr Keys= [3, 5] Mod = 11


"The remainder after
division."

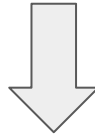


Public keys= ([2, 7], 9), ([4, 1], 5)

Public keys= ([2, 7], 9), ([4, 1], 5)

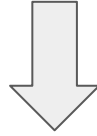


Public key= ([2+4 , 7+1], 9+5 mod 11)

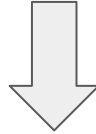


Public key= ([6 , 8], 3)

Public key= ([6 , 8], 3)

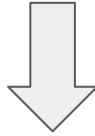


$$6 \times 3 + 8 \times 5 = 18 + 40 = 58 \bmod 11 = 3$$

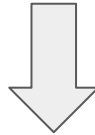


$$3 - 3 = 0$$

Public key= ([6 , 8], 8)



$$6 \times 3 + 8 \times 5 = 18 + 40 = 58 \bmod 11 = 3$$



$$8 - 3 = 5$$

Good: 

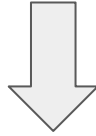
Not good: 

Mod = 11 (0 - 10)

Number close to 0 means 0

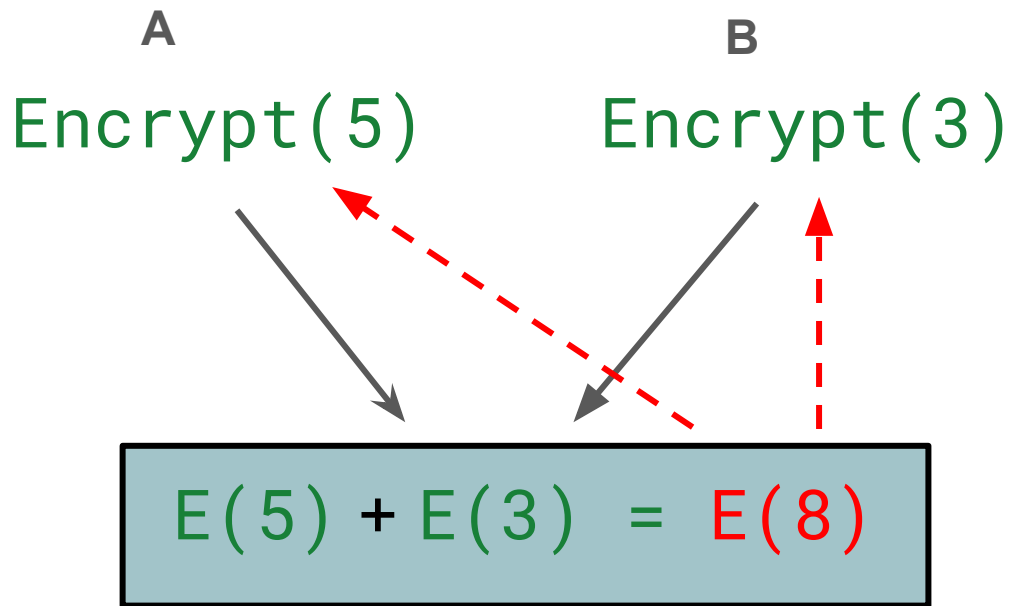
Number close to 5 means 1

Only encrypt message for 0 and 1



(like 200) into binary: **11001000**

Why it can be used in HE?



Remote server

$$\begin{array}{rcl}
 + & 0 & = 0 \\
 & 1 & = 1
 \end{array}$$

1

$([6,8],3)$

$([6,8],8)$

$([6,8],3) + ([6,8],8) = ([12,16],11) \bmod 11$

$([12,16], 0)$  **1**

$$1 + 1 = 0$$

$$1 + 0 = 1$$

$$0 + 1 = 1$$

$$0 + 0 = 0$$

$$3 * 12 + 5 * 16 = 116 \bmod 11 = 6$$

$6 - 0 = 6 \bmod 11 = 6$ is close to 5, so,
it means 1

HE features:

Enables computation directly on encrypted data.

- No need to decrypt before processing.

Supports Arithmetic Operations

- Can perform **addition** and/or **multiplication** on ciphertexts.

Privacy-Preserving

- Sensitive data can be processed in the cloud **without revealing it**.

Noise Management Required

- Each operation adds **noise** to ciphertext.

HE limitations:

Only numerical data.

Slow computation

Limited operations, $+$ $-$ $*$ $/$: Only good for Addition, Subtraction, Multiplication, Division.

Why hospitals need HE? (only numerical data)

- **Complies with HIPAA** (U.S. privacy law)
- Avoids patient data leaks when outsourcing to cloud or AI companies
- Enables collaboration across hospitals without revealing raw data

HE can absolutely be used in hospitals, especially for privacy-preserving AI, analytics, and research collaboration.

But it's still emerging in practice due to **performance and integration challenges.**

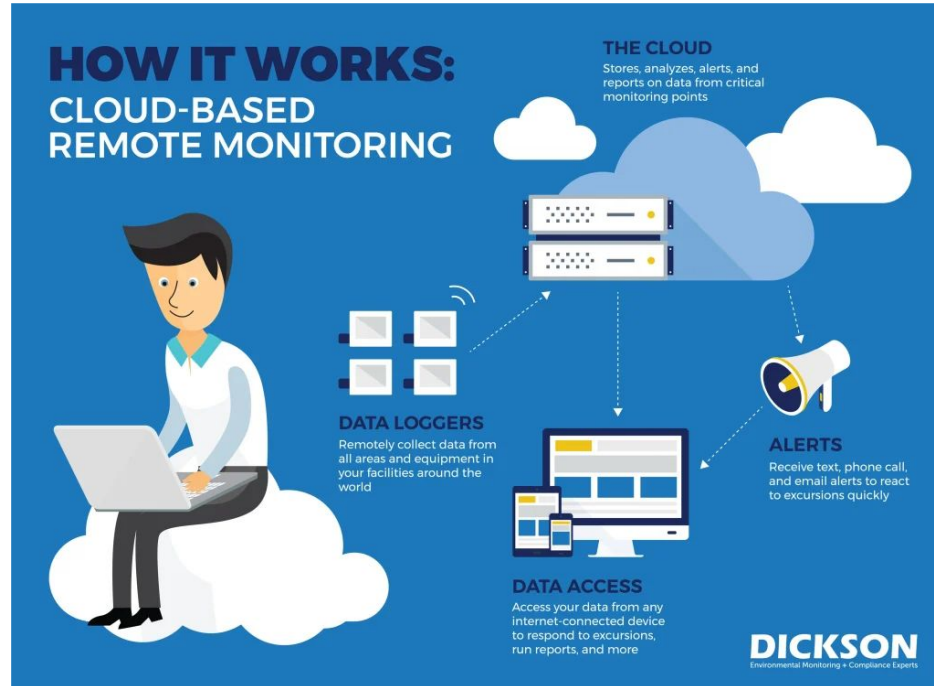
C	FD	FE	FF	FG	FH	FI	FJ	FK	FL	FM	FN	FO	FP	FQ	FR	FS	FT	FU	FV	FW	FX
E_e4	lowput_ex	DATSCAN	DATSCAN	con_caud	ips_caud	mean_cau	DATSCAN	DATSCAN	con_putan	ips_putan	mean_put	con_striat	ips_striat	mean_stri	Stage_part	Stage_sub	Stage_PDT	Stage_S	Stage_D	Stage_G	
1	0.422884	2.15	1.53	1.53	2.15	1.84	1.37	0.79	0.79	1.37	1.08	2.32	3.52	1.46	7	22	0	1	1		
1	0.48624	2	1.39	1.39	2	1.695	1.18	0.9	0.9	1.18	1.04	2.29	3.18	1.3675	5	29	0	1	1		
1	0.413708	1.79	1.43	1.43	1.79	1.61	0.89	0.76	0.76	0.89	0.825	2.19	2.68	1.2175	5	34	0	1	1		
1															4	42	0	1	1		
1	0.386278	1.89	1.12	1.12	1.89	1.505	1.11	0.7	0.7	1.11	0.905	1.82	3	1.205	5	42	0	1	1		
0	0.240906	1.19	1.43	1.19	1.43	1.31	0.48	0.58	0.48	0.58	0.53	1.67	2.01	0.92	7	24	0	1	1		
0	0.273404	1.3	1.32	1.3	1.32	1.31	0.64	0.54	0.64	0.54	0.59	1.94	1.86	0.95	7		1	1	1		
0															6		1	1	1		
0															9		1	1	1		
1	0.217016	1.21	1.46	1.21	1.46	1.335	0.45	0.78	0.45	0.78	0.615	1.66	2.24	0.975	6	15	0	1	1		
1	0.243108	1.32	1.54	1.32	1.54	1.43	0.5	0.64	0.5	0.64	0.57	1.82	2.18	1	7	16	1	1	1		
1															6	13	1	1	1		
0	0.321371	1.86	2.36	1.86	2.36	2.11	0.72	1.38	0.72	1.38	1.05	2.58	3.74	1.58	11	34	0	1	1		
0	0.341553	1.82	2.41	1.82	2.41	2.115	0.76	1.21	0.76	1.21	0.985	2.58	3.62	1.55	10	32	1	1	1		
0	0.280867	1.65	2.09	1.65	2.09	1.87	0.62	1.01	0.62	1.01	0.815	2.27	3.1	1.3425	17	30	1	1	1		
0															9	27	1	1	1		
2	0.17646	0.86	1.2	0.86	1.2	1.03	0.33	0.48	0.33	0.48	0.405	1.19	1.68	0.7175	2	20	0	1	1		
2	0.167254	0.93	1.23	0.93	1.23	1.08	0.31	0.4	0.31	0.4	0.355	1.24	1.63	0.7175	3	18	1	1	1		
2	0.152435	0.86	1.34	0.86	1.34	1.1	0.28	0.42	0.28	0.42	0.35	1.14	1.76	0.725	2	37	1	1	1		
2															3	32	1	1	1		
0	0.448237	1.91	2.5	1.91	2.5	2.205	0.94	1.28	0.94	1.28	1.11	2.85	3.78	1.6575	7	18	0	1	1		
0	0.389459	1.6	2.38	1.6	2.38	1.99	0.81	1.35	0.81	1.35	1.08	2.41	3.73	1.535	7	17	1	1	1		
0	0.459581	1.72	2.35	1.72	2.35	2.035	0.95	1.09	0.95	1.09	1.02	2.67	3.44	1.5275	7	21	1	1	1		
0															16	28	1	1	1		
1	0.359569	1.8	1.47	1.47	1.8	1.635	0.87	0.79	0.79	0.87	0.83	2.26									

Use case

Remote monitoring involves observing and tracking data from a distance, often using technology like sensors and networks, to gain insights into systems, equipment, or even people's health.

Doctor has Keys to decrypt remote patient datasets.

Patients' datasets are encrypted by HE, send to cloud for computation and results will be sent to doctors.



Python Libraries

Homomorphic Encryption (HE)

<https://github.com/Lab41/PySEAL>

Zero Knowledge Proof

<https://github.com/sdiehl/zkp>

Zero Knowledge Proof

<https://github.com/sdiehl/zkp>

Threshold signatures

<https://cryptography.io/en/latest/>

`pip install cryptography`

The most popular python library for encryption

<https://pypi.org/project/cryptography/>

Encryption/Decryption:

- Symmetric (e.g., AES)
- Asymmetric (e.g., RSA, ECC)

Hashing (e.g., SHA-256)

Digital Signatures (e.g., RSASSA-PSS)

Certificates, PKI tools