CS670: Cryptographic Techniques for Privacy Preservation

Module 0: Introduction

Adithya Vadapalli

Admistrativia

Instructor: Adithya Vadapalli avadapalli@cse.iitk.ac.in

Teaching Assistants

Tufan Singha Mahapatra	tufansm@cse.iitk.ac.in
Indranil Thakur	indra@cse.iitk.ac.in
Rohit Kumar	krohit24@cse.iitk.ac.in
Jogi Yashil Jayesh	yashilj24@cse.iitk.ac.in

where and when?: KD101, Monday and Thursday, 9 AM to 10:15 AM

Office Hours: Monday, 11:30 AM to 12:30 PM

Admistrativia

Acadly for attendance and in-class activities

(Attendance is not compulsory, but the classes will be useful, attendance stats are useful to me)

Hello IITK (and e-mail) for announcements

Course Github Page for course material (will be published soon)

Gradescope (viewing exam and quiz results)

What is Privacy?

Does privacy hinder progress or stop the benefits of the Internet?

Netflix provides you with recommendations, which is great!

But does it know your consumption history?

Can we build a system where Netflix can provide you with recommendations without knowing what you watched?

Can you anonymously post messages on a bulletin board?

Modules of the course

Module 0: Introduction

Module 1: Private Information Retrieval

Module 2: Secure Multiparty Computation

Module 3: Secure Memory Access

Module 4: Zero Knowledge Proofs

Module 5: Secure Systems

Grading Scheme

Quizzes: There will be four quizzes at the end of each module (except Module 0 and 5), *10% of the grade*

(To be done individually, in any language, but should be dockerized)

Programming Assignments: There will be four programming assignments, 40% *of the grade*

Midsem Exam: One Cheat Sheet Allowed, *25% of the grade*

End Exam: One Cheat Sheet Allowed, *25% of the grade*

Private Information Retrieval

A cryptographic primitive that allows a user to download a record from a database held by a remote server without the server learning which record was downloaded.

For example, you want to watch a movie on Netflix without the Netflix servers learning what movie was watched.

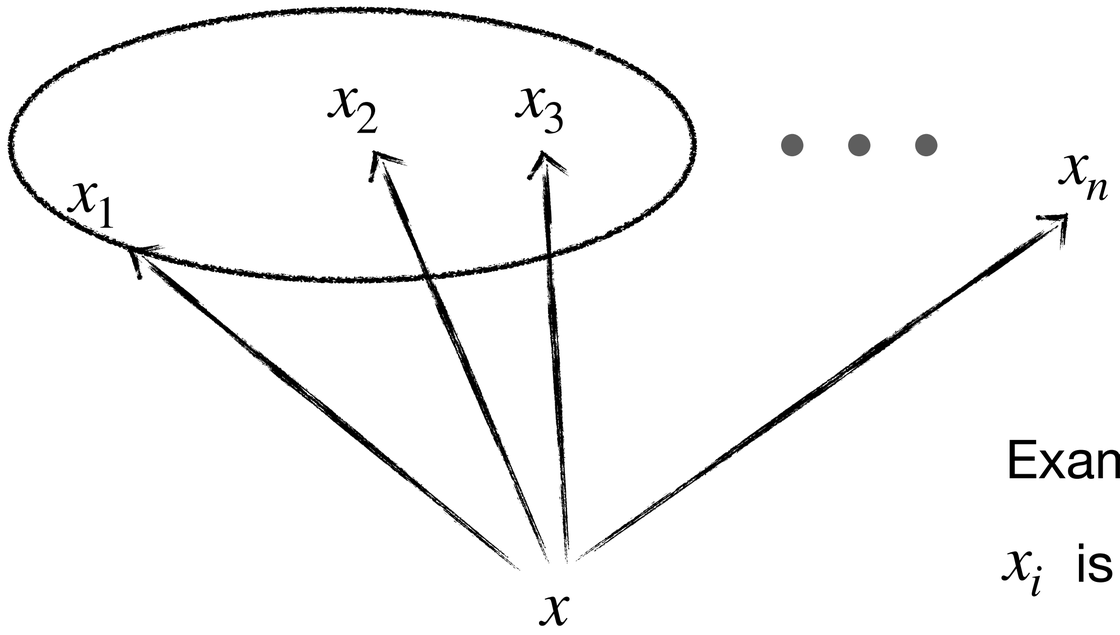
Does not protect your identity!

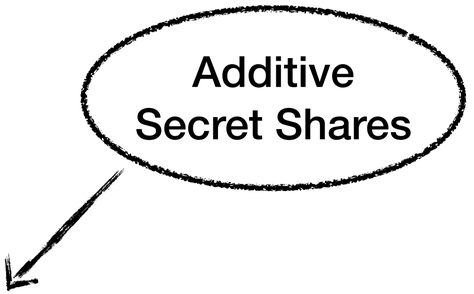
Keeps what was downloaded a secret, not who downloaded it

Secret Shares

A cryptographic technique that allows us to share a secret among multiple parties.

A subset of the parties can reconstruct the secret.





Example: $x = x_1 + x_2 + \ldots + x_n$

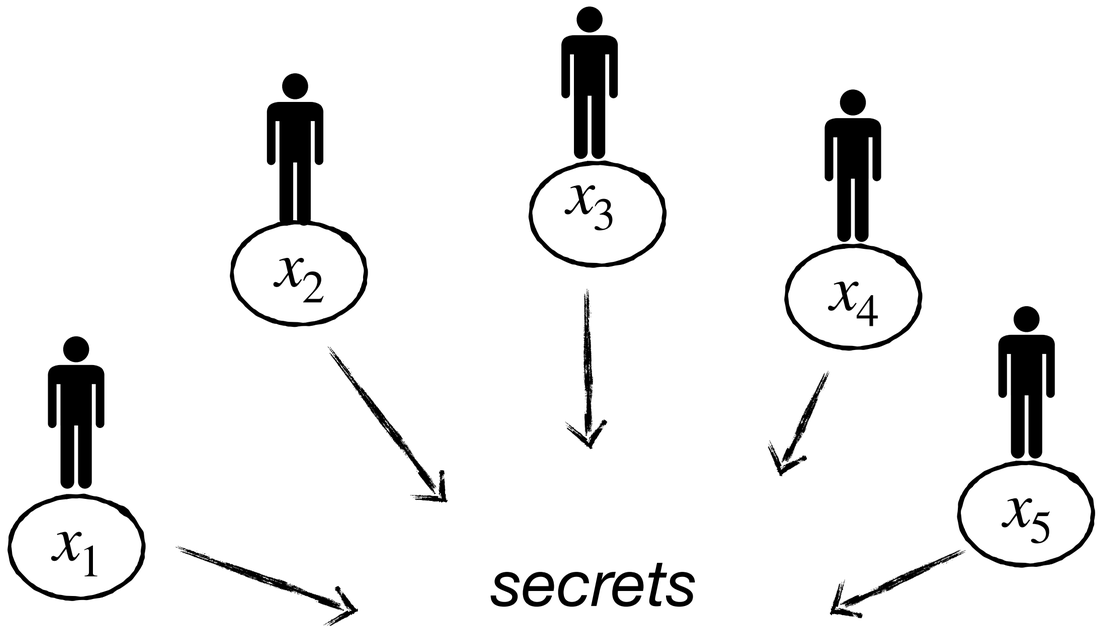
x_i is held by the party P_i

Any subset of size *less than n* cannot learn the secret

Secure Multiparty Computation

MPC is a cryptographic primitive that allows multiple parties to compute a function $f(x_1, x_2, x_3, x_4, x_5)$

(while the parties learn nothing other than the function itself)



Secure Multiparty Computation

Alice and Bob are two extremely wealthy individuals

They want to find out who among them is richer

But, they do not want to reveal to each other their net worth

Distributed ORAMs

A special case of MPC

$\overrightarrow{D}_0 \in \{0,1\}^{m \times n}$

i_0^*

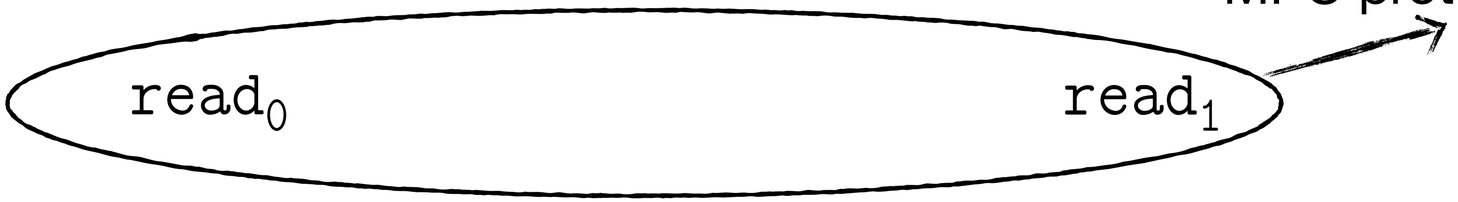
$\overrightarrow{D}_1 \in \{0,1\}^{m \times n}$

i_1^*

$\overrightarrow{D}_0 + \overrightarrow{D}_1 = \overrightarrow{D}$

$i_0 + i_1 = i^*$

MPC protocol computes



$read_0 + read_1 = D[i^*]$

Zero Knowledge Proofs

Allows a prover to prove:

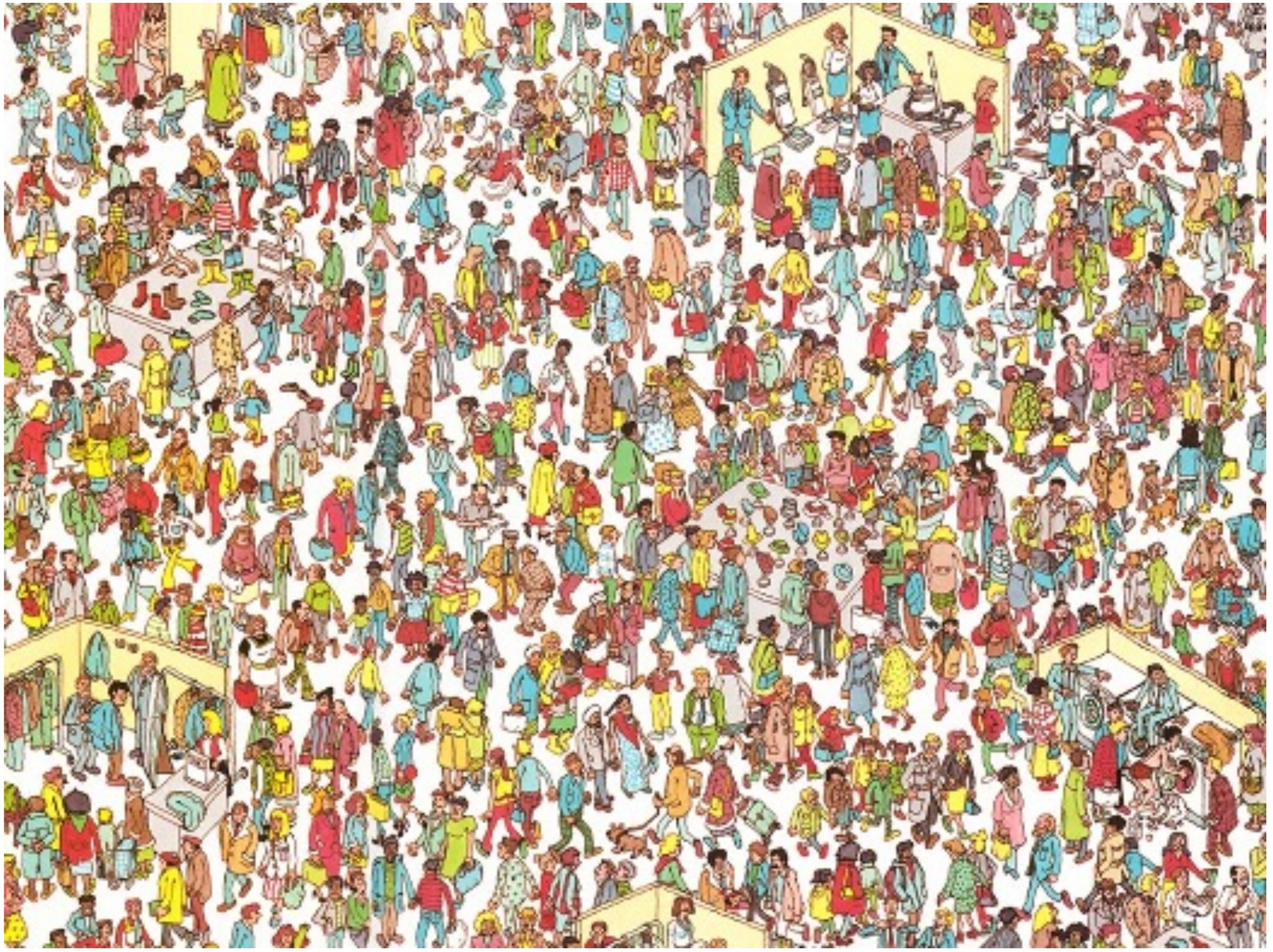
the knowledge of a statement without revealing any other information.

Example: Alice has solved Sudoku.

She wants to prove to Bob that she has solved the puzzle

But … she does not want Bob to learn *any other* information.

the operative word



Abstract Algebra 101

Groups

A group is a set G with one operation $\circ$

Closure • Associativity • Identity Element • Inverse Element

$\forall a, b \in G, a \circ b \in G$ $(a \circ b) \circ c = a \circ (b \circ c)$ $a \circ e = e \circ a = a$ $a \circ a^{-1} = a^{-1} \circ a = e$

Integers under addition

Non-zero real numbers under multiplication

Semigroup

A semigroup is a set S with a binary operation: $\circ$

$$\forall a, b, c \in S, (a \circ b) \circ c = a \circ (b \circ c)$$

No requirement for an identity element

No requirement for inverses

$\mathbb{N}$ (Natural Numbers) under addition is an example of a semigroup

A semigroup with an identity element is called a monoid

A semigroup with an identity and inverse elements is called a group

Abelian Groups

Formally, a group $(G, *)$ is called an *Abelian group* if it satisfies the following *group axioms*, and the operation $*$ is commutative:

Example: Integers under addition Real Numbers under addition

How about Integers under multiplication? That is not a group!

Give an example of a group that is not abelian:

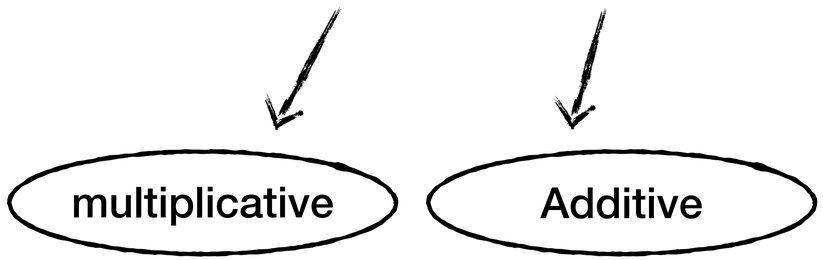
The group of 2x2 invertible matrices under multiplication.

Cyclic Groups

A cyclic group is a group that can be generated by a single element.

A group G is cyclic if there exists an element $g \in G$

Such that every element $h \in G$ can be written as g^n or $n \cdot g$



Rings

A ring is a set R with two binary operations:

Addition +

Multiplication *

Such that:

(R, +) is an Abelian Group

(R, *) is a semi-group

Distributivitiy $a * (b + c) = a*b + a*c$

$(a + b) * c = a*c + b*c$

Rings

Is $(\mathbb{N}, +)$ (positive integers) a ring?

Closed under addition and multiplication, both associative, and multiplication distributes over addition

No additive inverse

Is $(2\mathbb{Z}, +)$ (even integers) a ring?

Additive inverse exists

No multiplicative identity But that is not need for it to be ring

A set of all polynomials whose degree is ≥ 1

Closed under multiplication and addition

No additive identity

Fields

A field is a set F with two operations: say, addition (+) and multiplication (*) satisfying:

(F, +) is an abelian group

(F \ {0}, *) is an abelian group

Distributivity: a * (b + c) = a * b + a * c

No zero divisors: if ab = 0, then a = 0 or b = 0

Every non-zero elemet has a multiplicative inverse

Prove that a field cannot have zero divisors.

Fields

Non-Examples:

Integers $\mathbb{Z}$ under normal addition and multiplication

Integers $\mathbb{Z}_6$ integers modulo 6

Example: $\mathbb{Z}_5$

$\mathbb{Z}_n$ is a field if and only if n is a prime

How about $\mathbb{F}_p[x]$?

$\mathbb{F}_p[x]$ is a set of all polynomials with coefficients in $\mathbb{F}_p$ not a field!

Fields

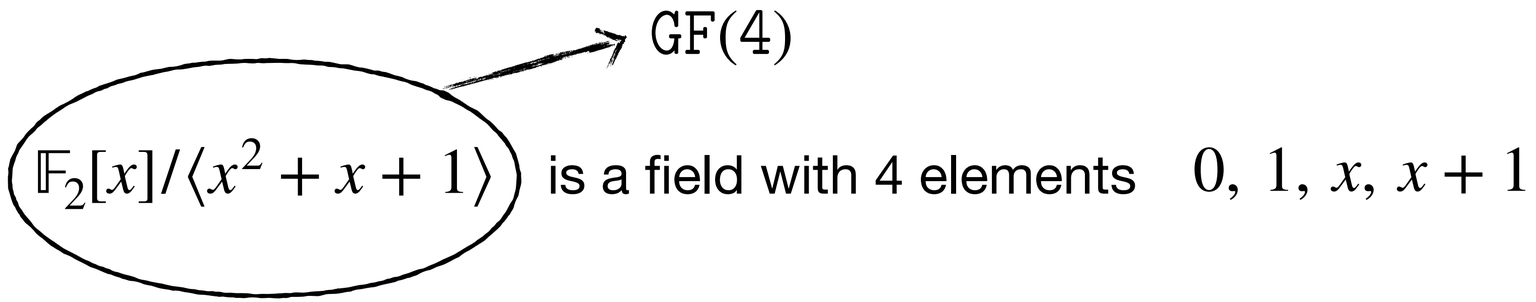
How about $\mathbb{F}_p[x]$?

$\mathbb{F}_p[x]$ is a set of all polynomials with coefficients in $\mathbb{F}_p$

not a field!

Galios Fields

A field F with a finite number of elements



is a field with 4 elements $0,\ 1,\ x,\ x+1$

$\mathtt{GF}(2^{\mathtt{k}})$: a finite field with $2^{\mathtt{k}}$ elements $0,\ 1,\ x,\ x+1$

We can consider bitwise XOR as a Galois Field

Characteristic: 2 (i.e., 1 + 1 = 0)

Each element is represented as a binary polynomial of degree < k

$$(x+1)\ +\ (1) = 1+0$$

$$(x+1)\ +\ (x+1) = 0+0$$

Cryptography

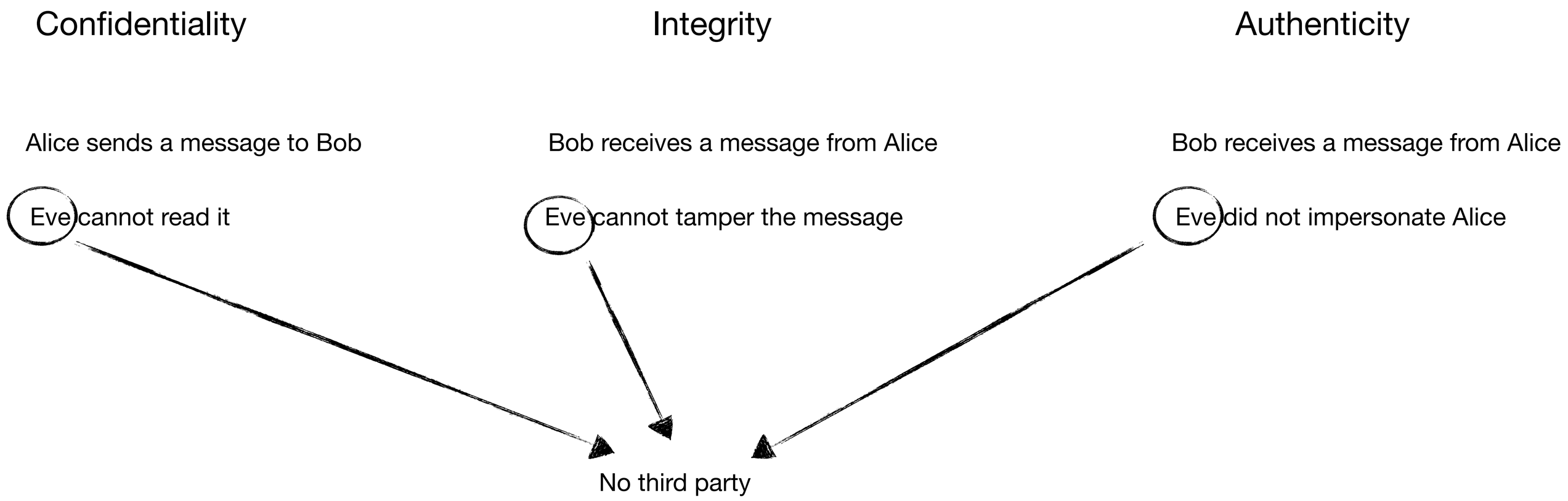
Symettric-key cryptography

- Two parties share a common key
- Anyone with the key can encrypt and decrypt

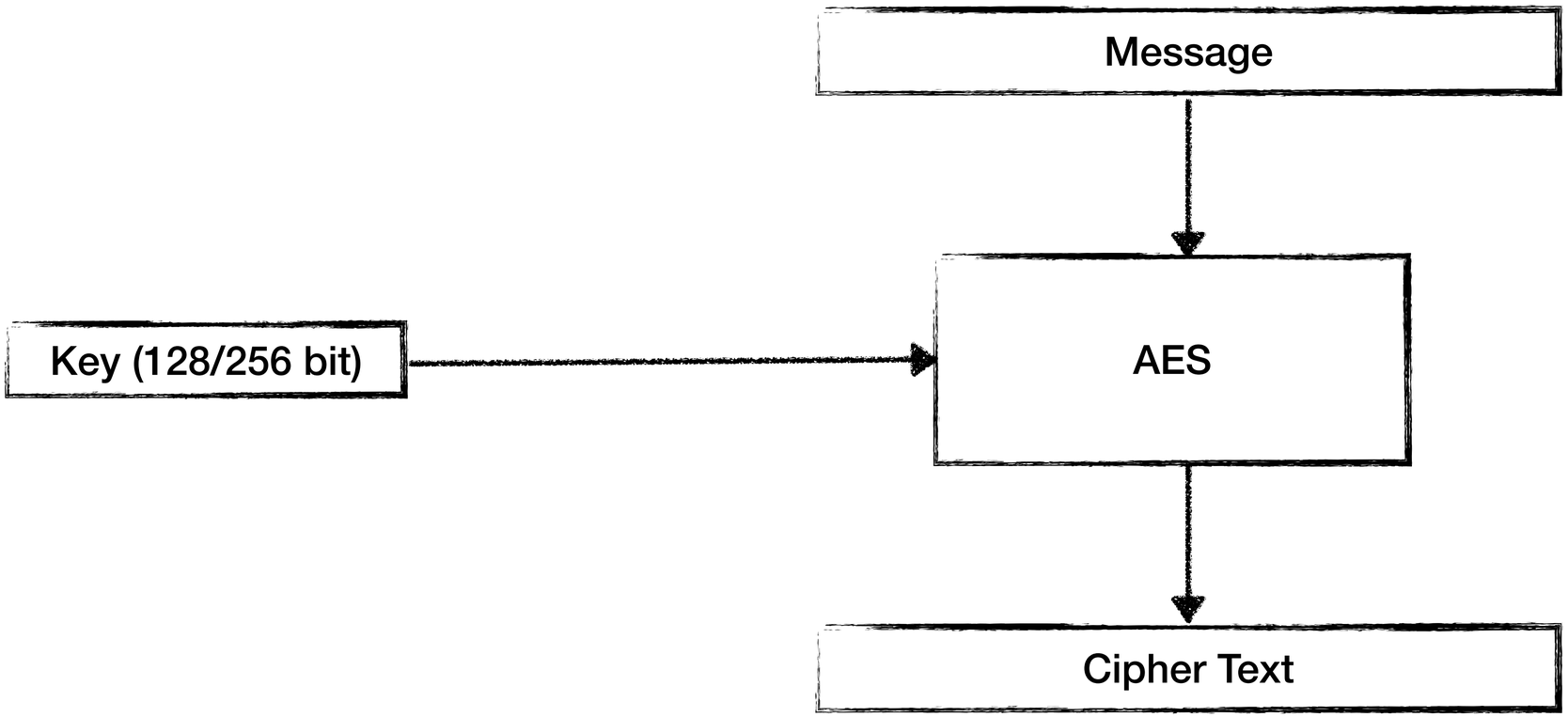
Public-key cryptography

- The key used to encrypt is different from the decryption key
- The encryption key is public (thus, anyone can encrypt the message)
- Decryption key is private (only the recipient can do the decryption)

Confidentiality, Integrity, Authenticity



Cryptography



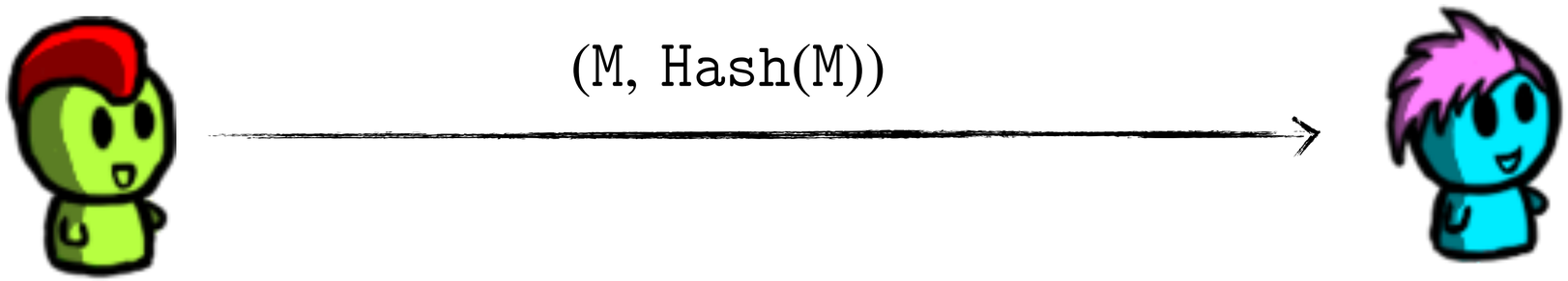
Hash Functions

Given a digest, you cannot find an input that hashes to it.

You cannot find two different inputs with the same digest.

`Hash(HUGE FILE)` = small 256-bit digest

Hash Functions

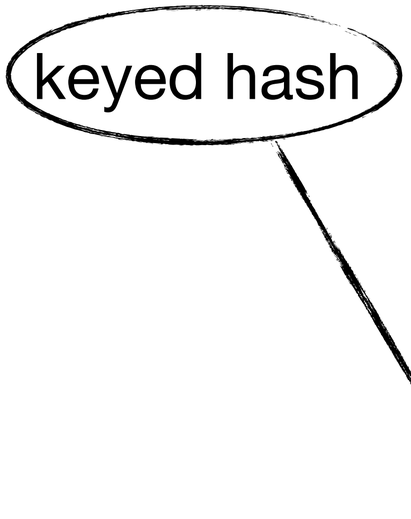


Does this work?

No Eve can change the message and compute the hash.

Message Authentication Codes

A MAC is like a keyed hash



Hash of the secret key shared by Alice and Bob

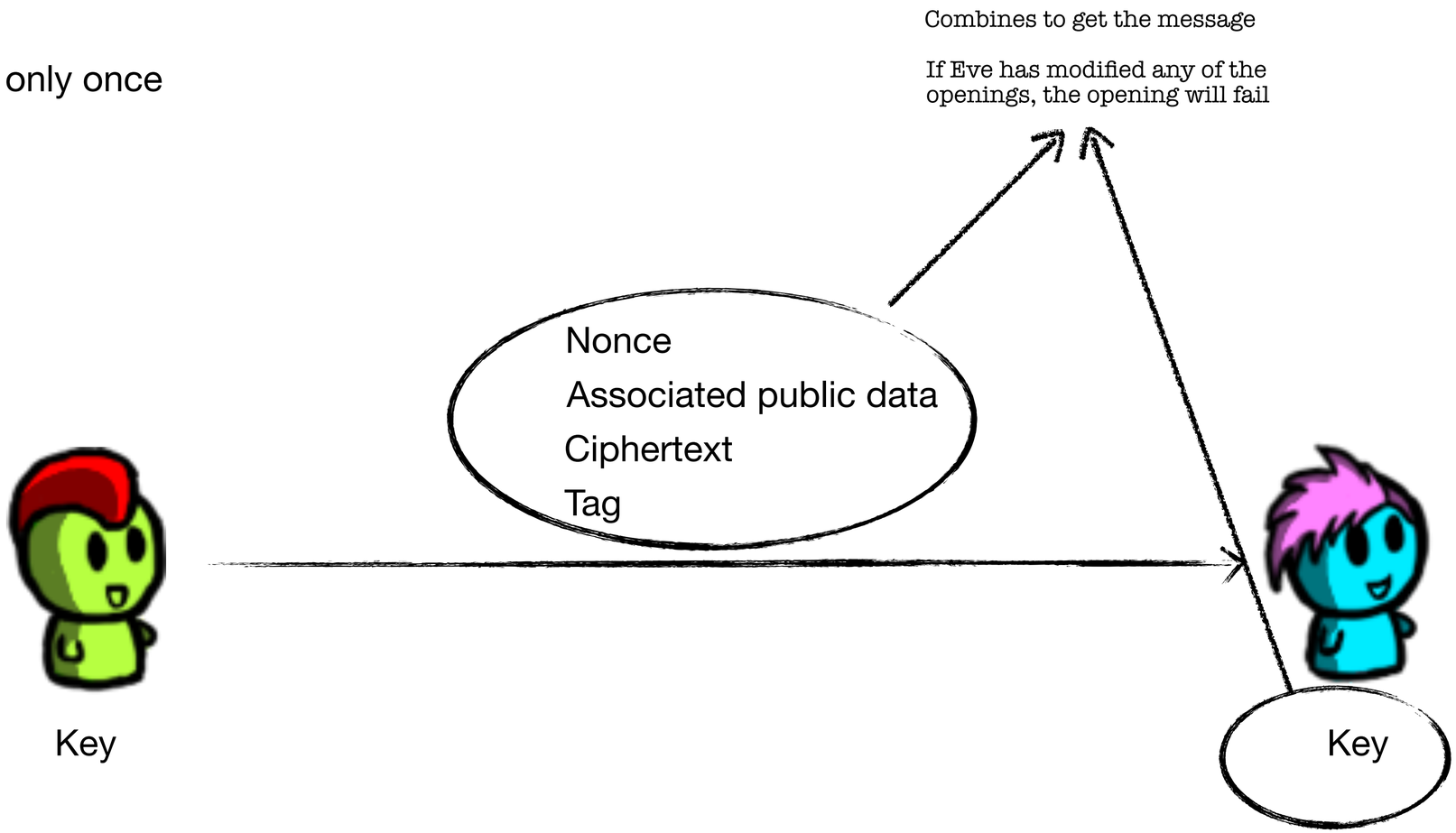
Authenticated Encryption with Associated Data

Inputs:

- Key which is known to Alice and Bob
- A nonce, which is public, but can be used only once
- The plain text to encrypt
- Associated public data

Output:

- Ciphertext
- Tag



Pseudo-random generators (PRGs)

A PRG takes in an input of fixed size and produces an arbitrary length output

Main Security Property:

but not the input key

Given any amount of the output, it does not help us determine the rest of the output

Can be built from AES, hash functions, or MACs

Public Key Cryptography

Problem with Symmetric-Key Cryptography:

Alice and Bob need to share a key in advance.

PKC allows Alice and Bob to create a symmetric key using only public information

Diffie-Hellman

Alice picks a private key x

Alice sends X to Bob $X = xB$

Bob picks a private key y

Alice sends Y to Bob $Y = yB$

$P_A = xY$ (Computed by Alice)

$P_B = yX$ (Computed by Bob)

Commitments

Given a random scalar x, and the output is X = xB

Public key is a commitment to the private key

Given the output X, it is hard to find the input x

The inputs are fixed size

El Gamal Encryption

Do DH to get a shared secret key

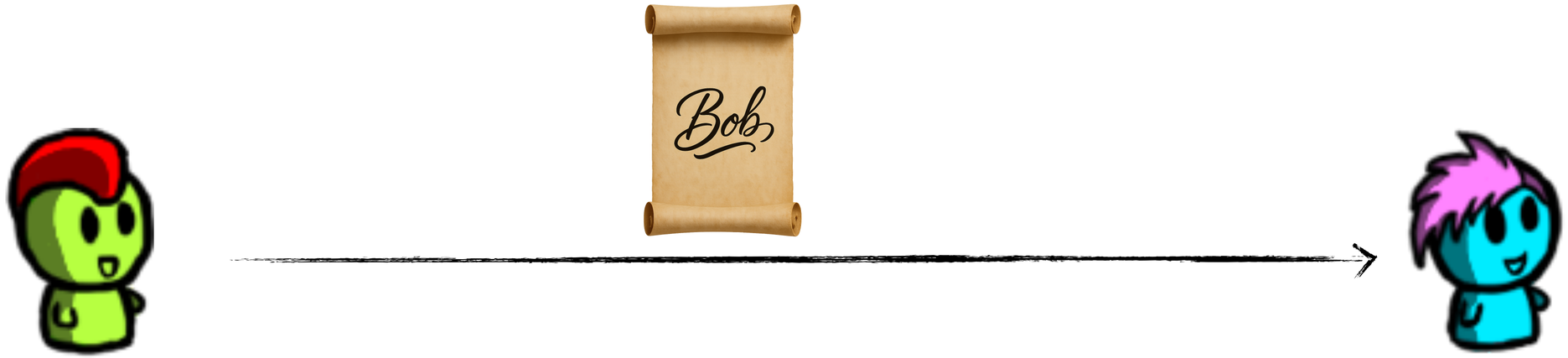
Encrypt the message with the shared key

Bob picks a private key y and publishes $Y = yB$

Alice picks a private key x, compute $X = xB$, $P = xY$, $k = H(P)$

Alice encrypts m with the key k

Digital Signature



Bob sends a message

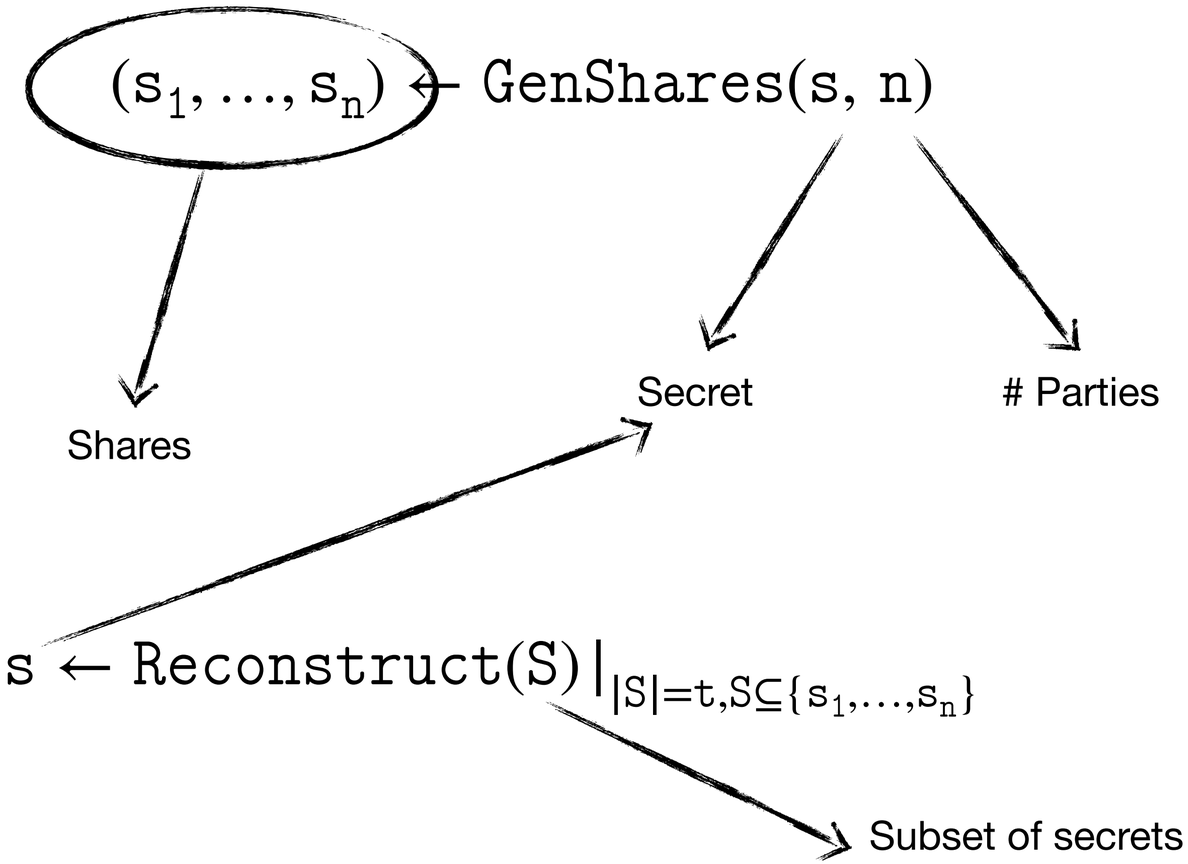
He signs it using a produce a signature: σ

Anyone can verify that it is a valid signature (using some public information)

Only Bob can produce the signature: σ

Secret Shares

A secret sharing mechanism has two primary methods



Additive Secret Shares over a Ring

Consider a secret: S

Consider a ring: $\mathbb{Z}_{64}$

$s_1, \ldots, s_n$ are secret shares over $\mathbb{Z}_{64}$ *iff* $s_1 + \ldots + s_n \bmod 2^{64} = S$

No multiplicative inverse in such a secret sharing!

Additive Secret Shares over a Finite Field

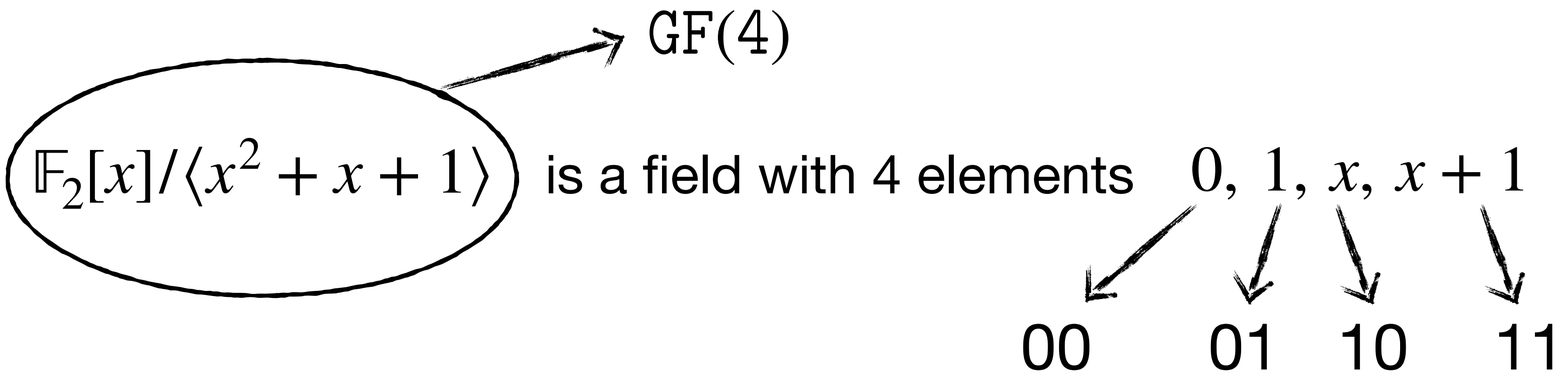
Consider a secret: S

Consider a field: $\mathbb{F}_{17}$

$s_1, \ldots, s_n$ Are secret shares over $\mathbb{Z}_{17}$ iff $s_1 + \ldots + s_n \bmod 17 = S$

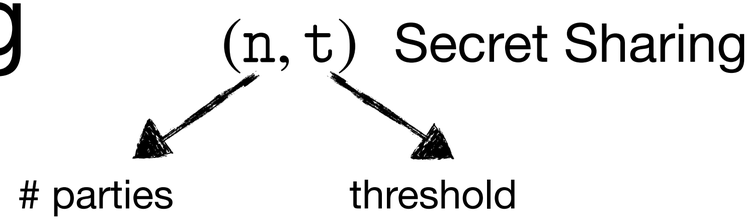
There is a multiplicative inverse in such a secret sharing!

Additive Secret Shares over a Finite Field



This is bitwise XOR.

Threshold Secret Sharing



Downside of additive secret sharing:

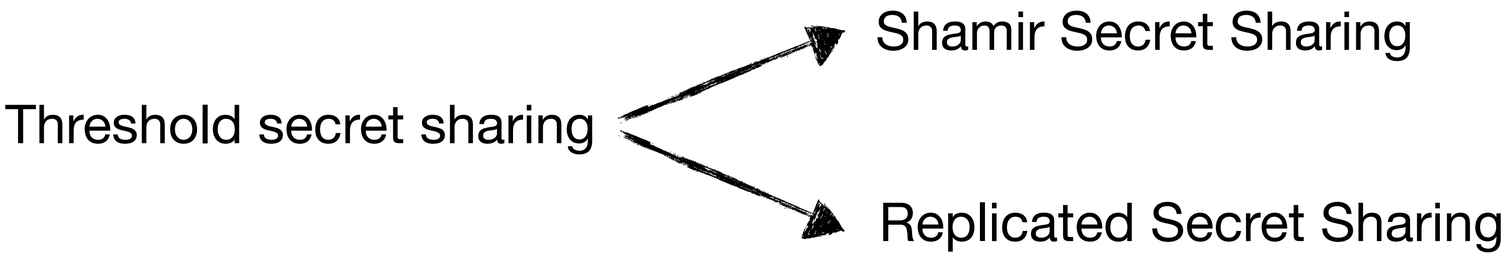
If even one of the parties is unavailable or is crashed, the secret is unusable

Threshold secret sharing:

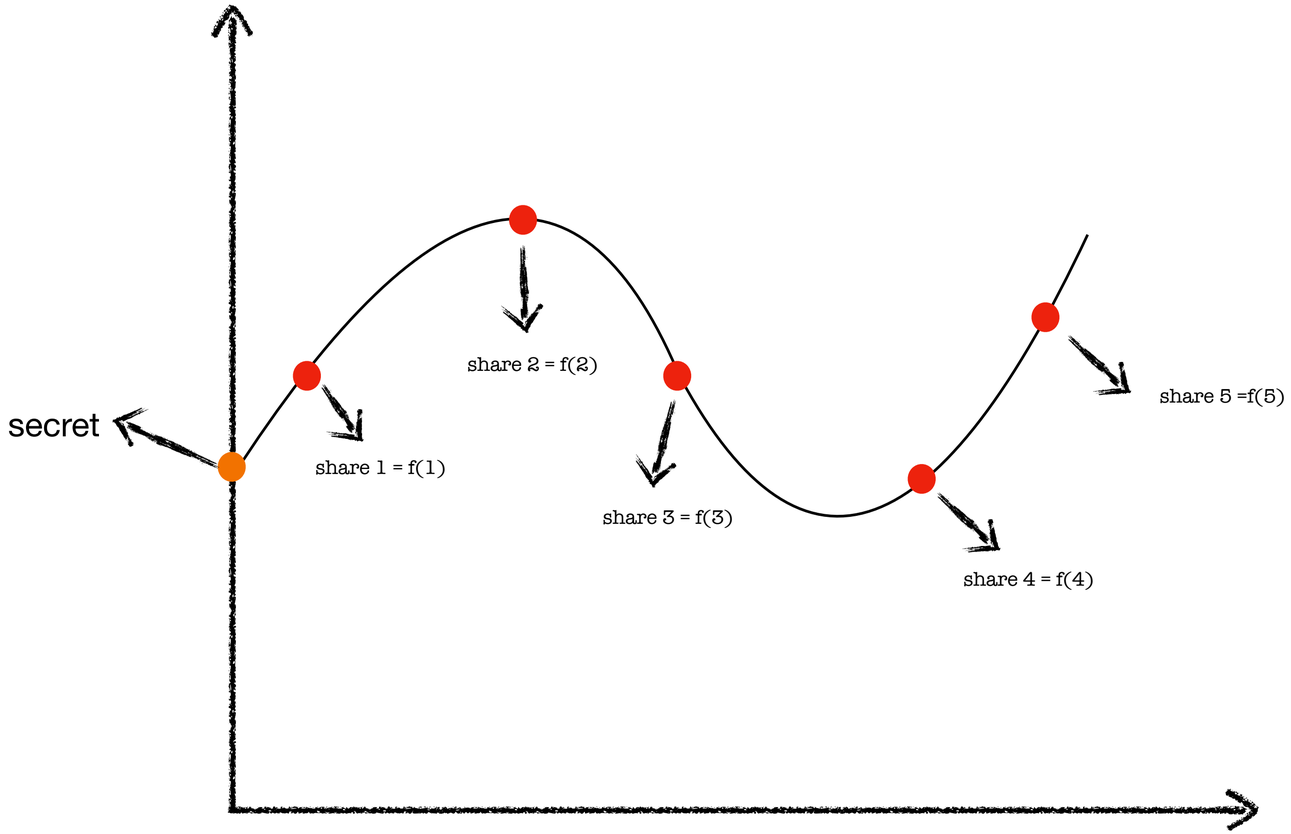
Divide the secret among n parties

However, only *(t+1)* shares have to come together to reconstruct the secret *s*

The threshold of t implies that (t+1) needs to get together to reconstruct the secret



Shamir Secret Shares



$$f(x) = ax^3 + bx^2 + cx + s$$

$$\in \mathbb{Z}_q$$

This is (5, 3)-secret sharing. Why?

Because the number of parties is 5.

And … we need (3+1) parties to come together to reconstruct the secret.

Reconstructing the Secret

Lagrange coefficients

$$\lambda_i = \prod_{j=1, j\neq i}^{t} \frac{x_j}{x_j - x_i} \qquad \text{for } i = 1, \ldots, t$$

$$s = \lambda_1 \cdot y_1 + \lambda_2 \cdot y_2 + \cdots + \lambda_t \cdot y_t$$

Lagrange interpolation

Reconstructing the Secret

Lagrange coefficients, let *n = 7, and t = 3*

Suppose parties 2, 4, and 5 are coming together to reconstruct the secret

$$\lambda_i = \Pi_{j=1,j\neq i}^{t} \frac{x_j}{x_j - x_i} \quad \text{for } i = 1, \ldots, t$$

$$s = \lambda_1 \cdot y_1 + \lambda_2 \cdot y_2 + \cdots + \lambda_t \cdot y_t$$

Lagrange interpolation

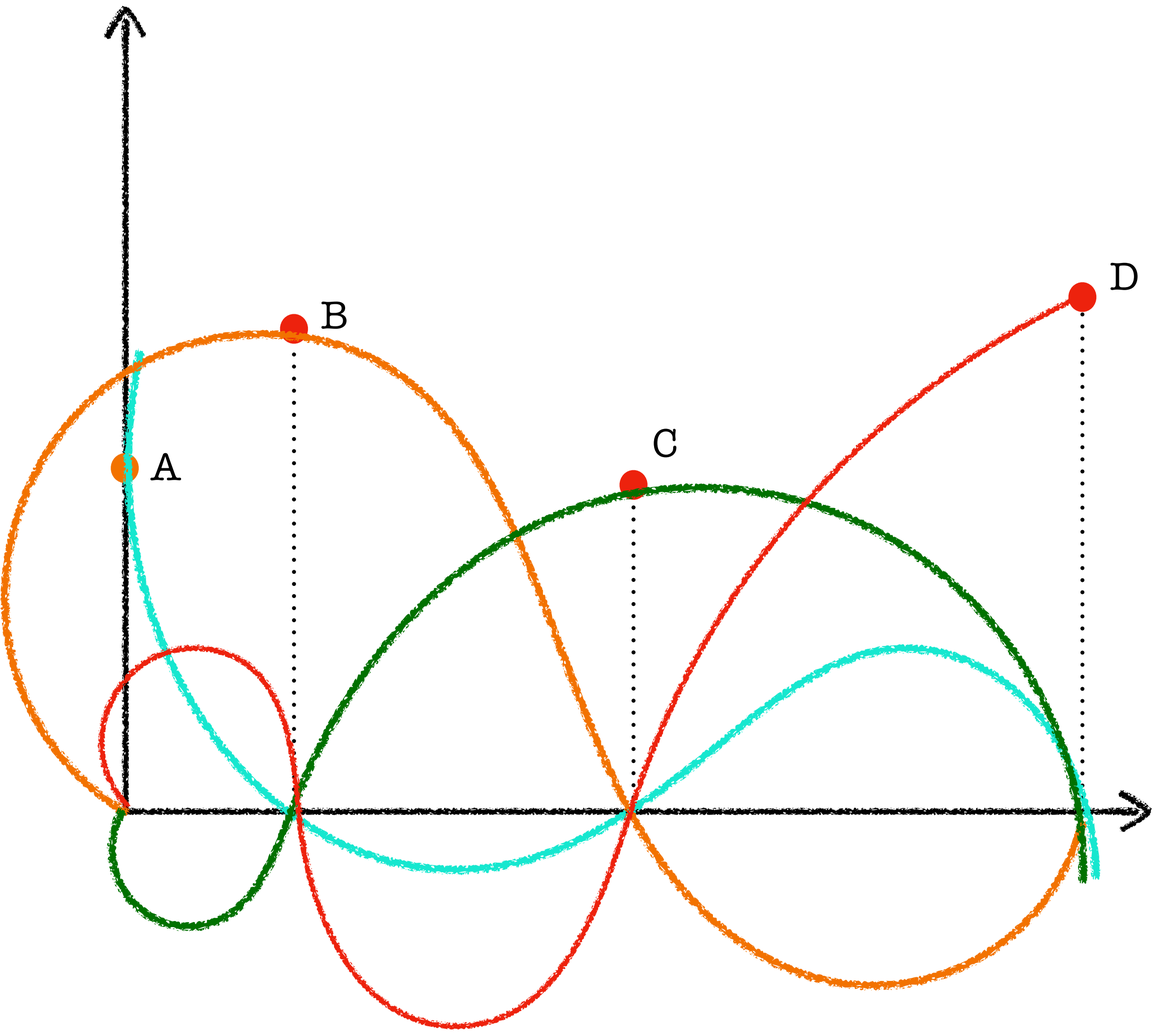
What are the coefficients?

$$\lambda_1 = \frac{4}{4-2} \cdot \frac{5}{5-2} \bmod q$$

$$\lambda_2 = \frac{2}{2-4} \cdot \frac{5}{5-4} \bmod q$$

$$\lambda_3 = \frac{2}{2-5} \cdot \frac{4}{4-5} \bmod q$$

Idea of Lagrange Interpolation



We want to find a polynomial that passes through all the four points

Find a polynomial that passes through A and evaluates to 0 at all other places

Find a polynomial that passes through B and evaluates to 0 at all other places

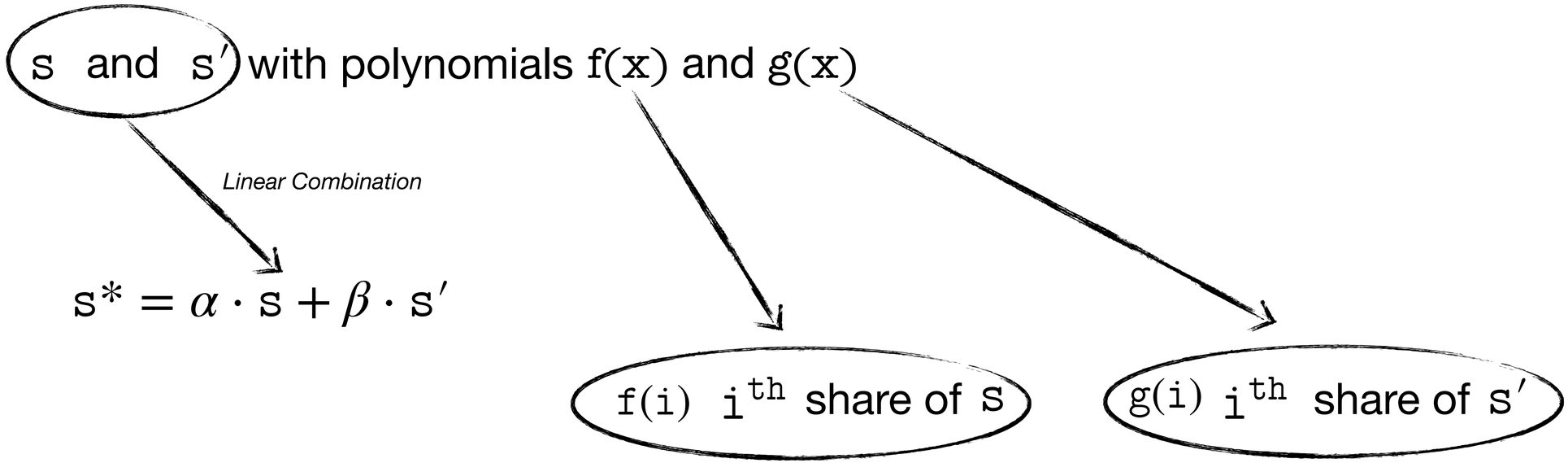
Find a polynomial that passes through C and evaluates to 0 at all other places

Find a polynomial that passes through D and evaluates to 0 at all other places

Add all these polynomials

Linearity of Shamir's Secret Sharing

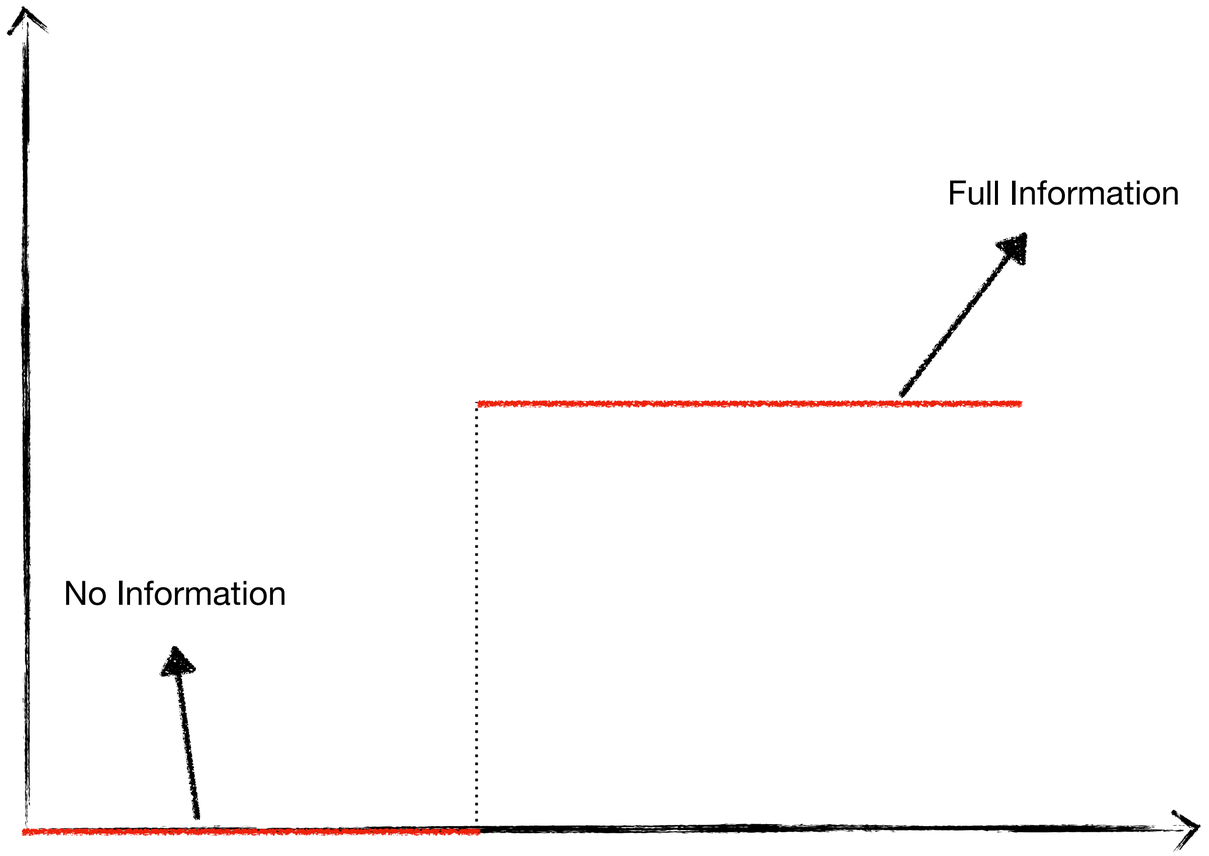
Consider two secrets s and s' with polynomials $f(x)$ and $g(x)$



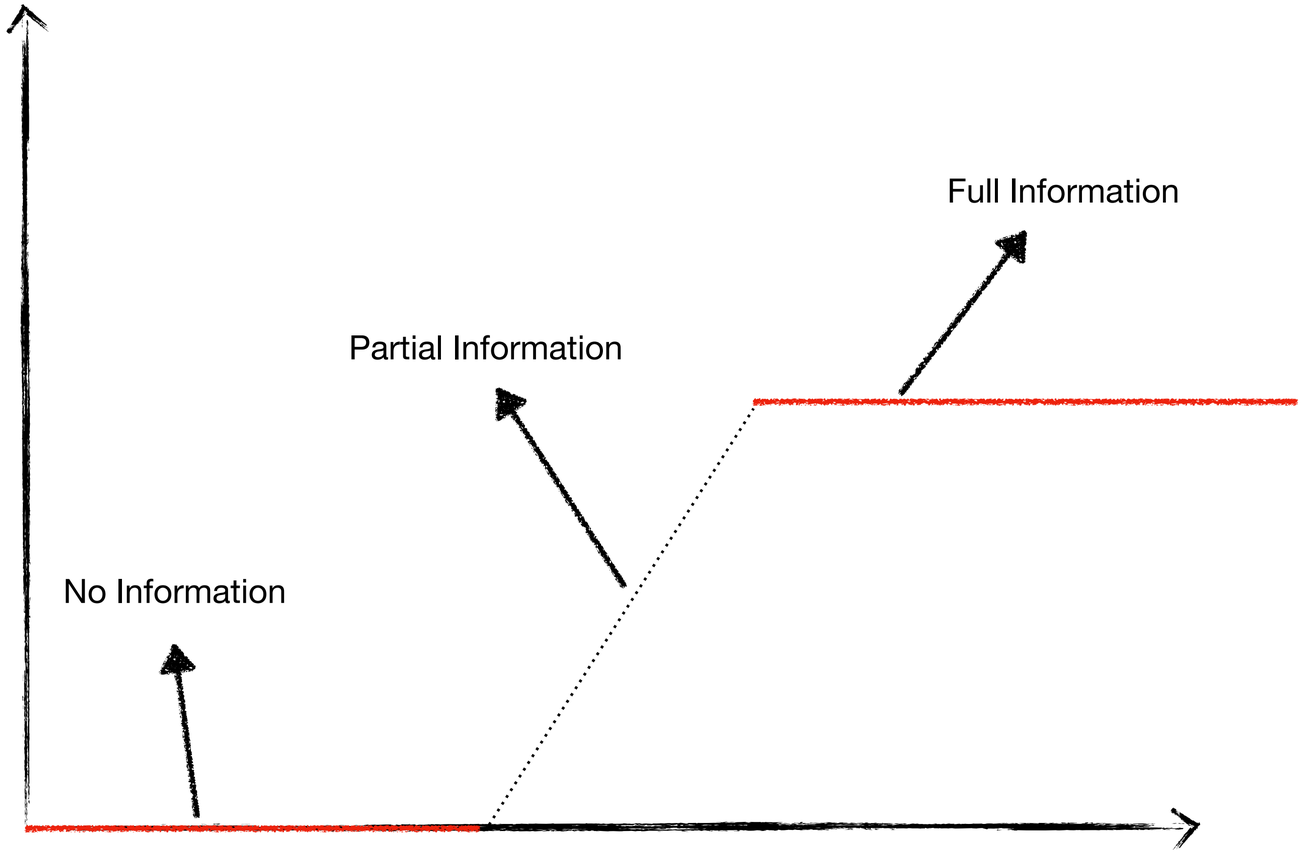
$s_i^* = \alpha \cdot s_i + \beta \cdot s_i'$

Is this a valid Shamir Secret Sharing of s^* ?

Rampified Shamir Secret Sharing



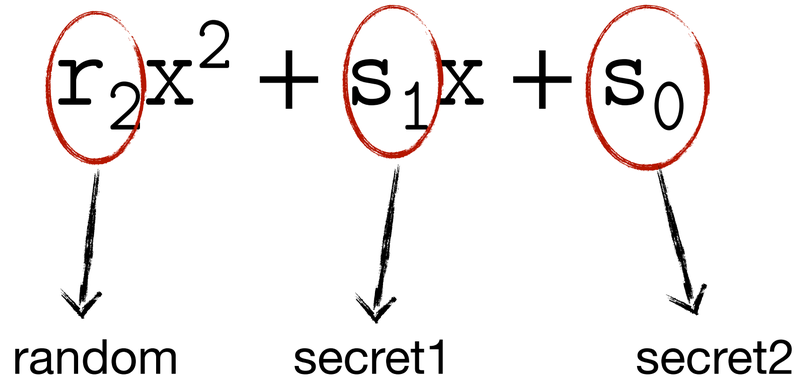
Rampified Shamir Secret Sharing

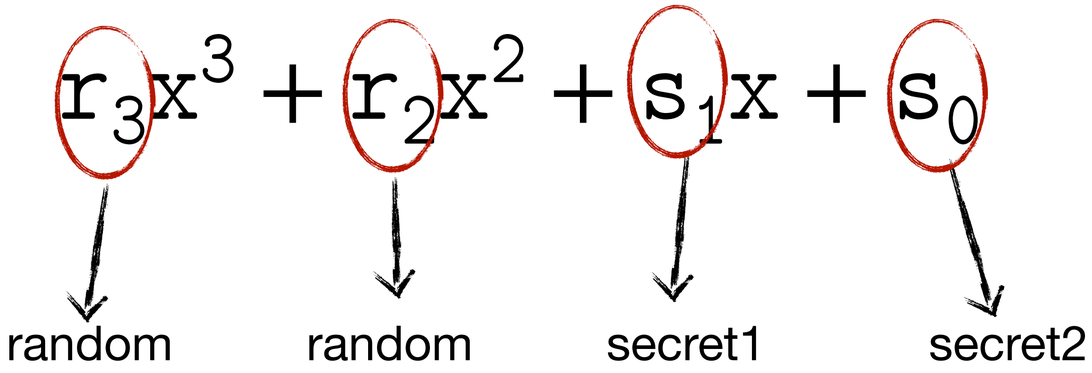


Rampified Shamir Secret Sharing

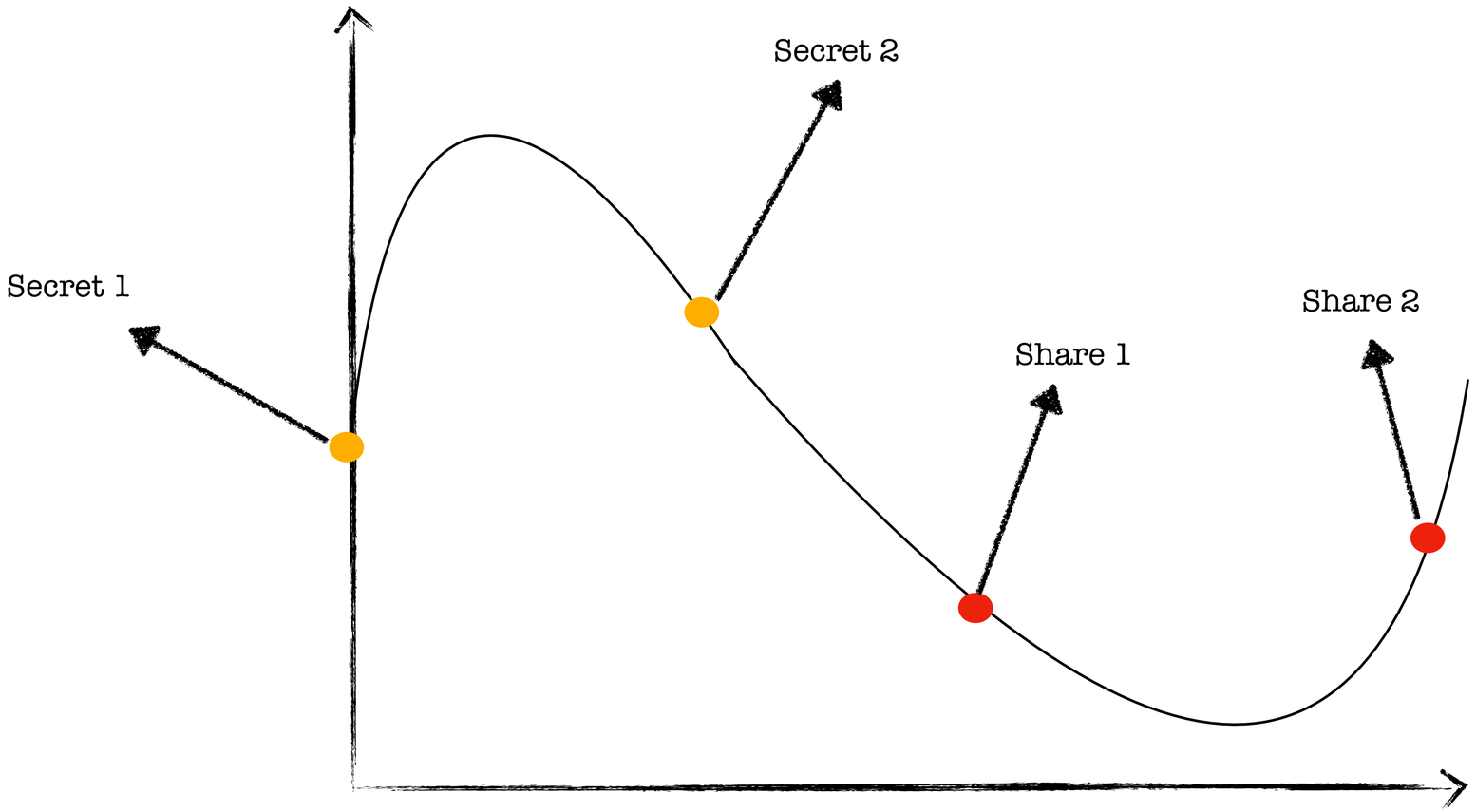
How do we achieve Rampified Shamir Secret Sharing?

2-degree polynomial encodes two secrets





Rampified Shamir Secret Sharing



q secrets $\Longrightarrow$ degree: t + q - 1

Blakley's Secret Sharing

Geometric way to threshold secret sharing! Any suggestions?

In t-dimensional space, a point (secret) can be defined as an intersection of t hyperplanes.

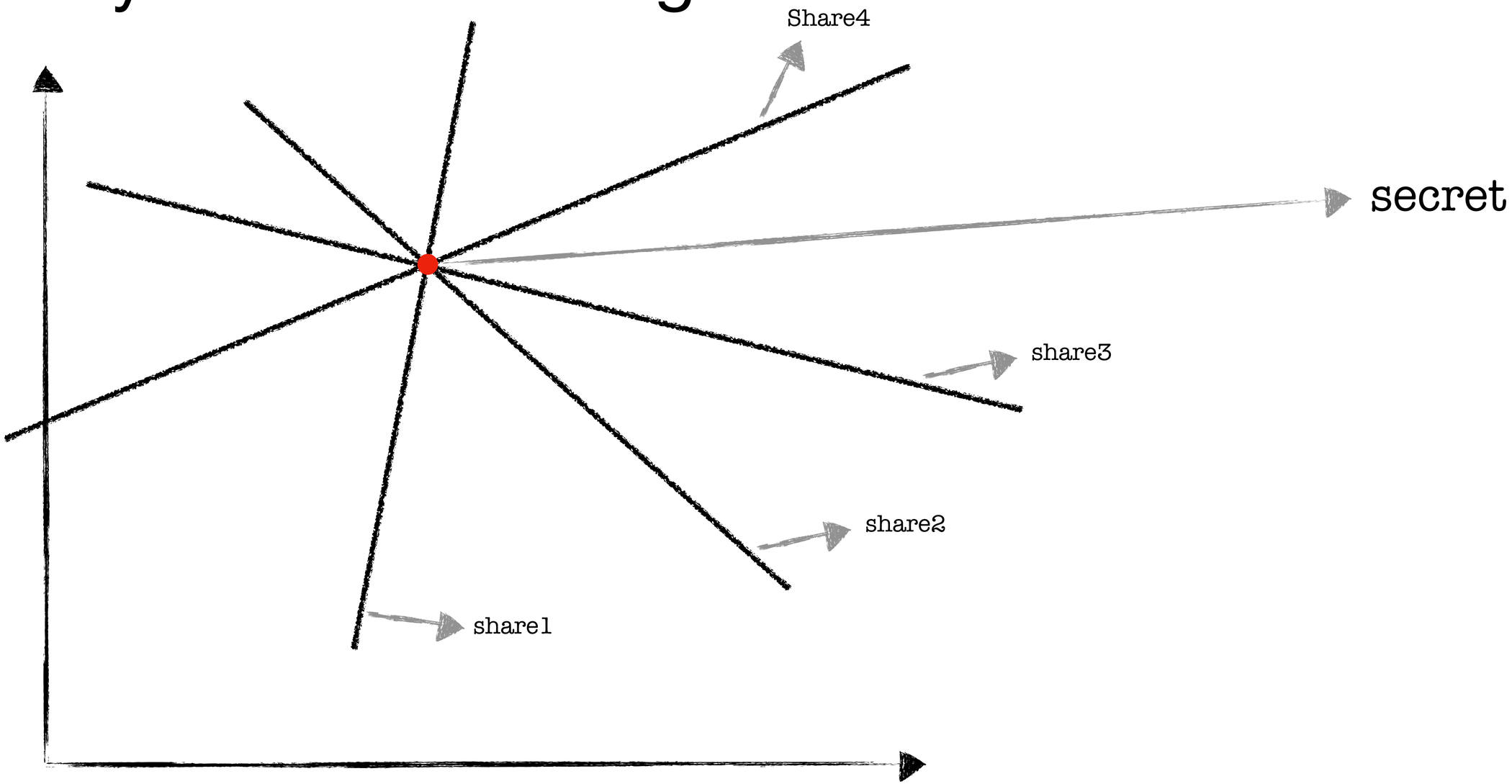
Each share is a hyperplane.

(and knowing any *t* such hyperplanes reveals the secret point).

Secret: $s = (s_1, s_2, \ldots, s_t) \in \mathbb{R}^t$

$a_1x_1 + a_2x_2 + \ldots + a_tx_t = b$ such that s lies on the hyperplane

Blakley's Secret Sharing



Observe that we can reconstruct the secret with any two shares!

Replicated Secret Shares (RSS)

Main idea of RSS

We want to divide the secret among *n* parties

We want *(t+1)* among them to recover the secret

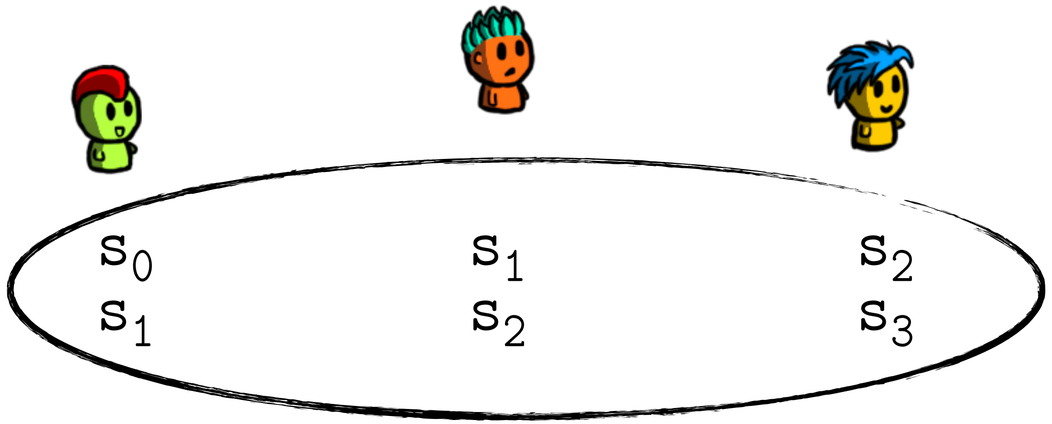
But an attacker comprising *t* of them learns nothing about the secret

How many such sets of t parties exist? $\binom{n}{t}$

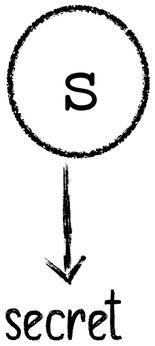
Idea: Additively split the secret into $\binom{n}{t}$ pieces

For *each set of t-1 parties* the attacker might comprise, give one of the $\binom{n}{t}$ pieces to everyone else outside the set

Replicated Secret Shares



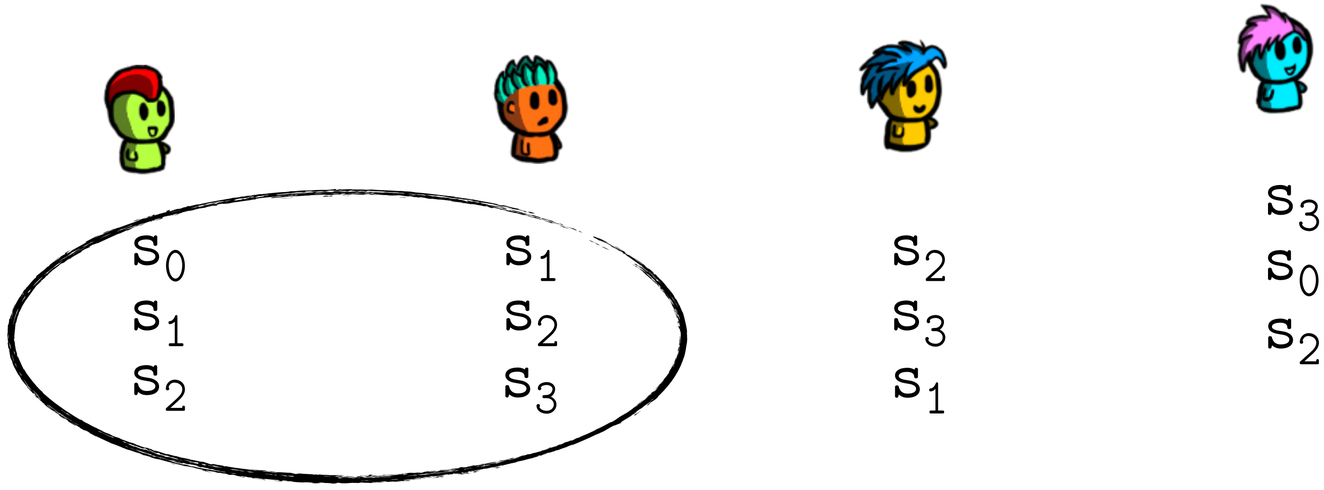
$n = 4$



Observe that any three parties can reconstruct the secret!

Can you change it so that any two parties can reconstruct the secret?

Replicated Secret Shares



$n = 4$

$s = s_0 + s_1 + s_2 + s_3$

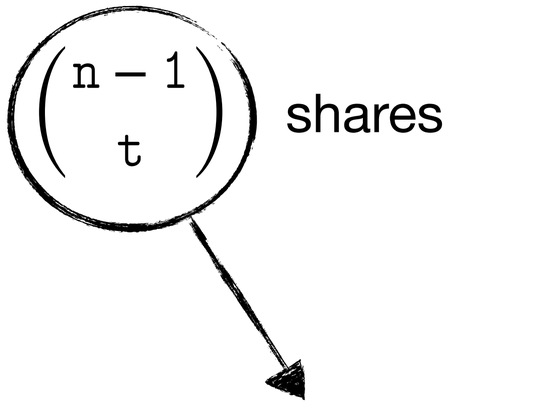
secret

Observe that any *two* parties can reconstruct the secret!

Replicated Secret Shares (RSS)

What are the problems with RSS?

Each party needs to hold $\binom{n-1}{t}$ shares



For small *n* and *t*, this works well

Threat Model

Different kinds of entities:

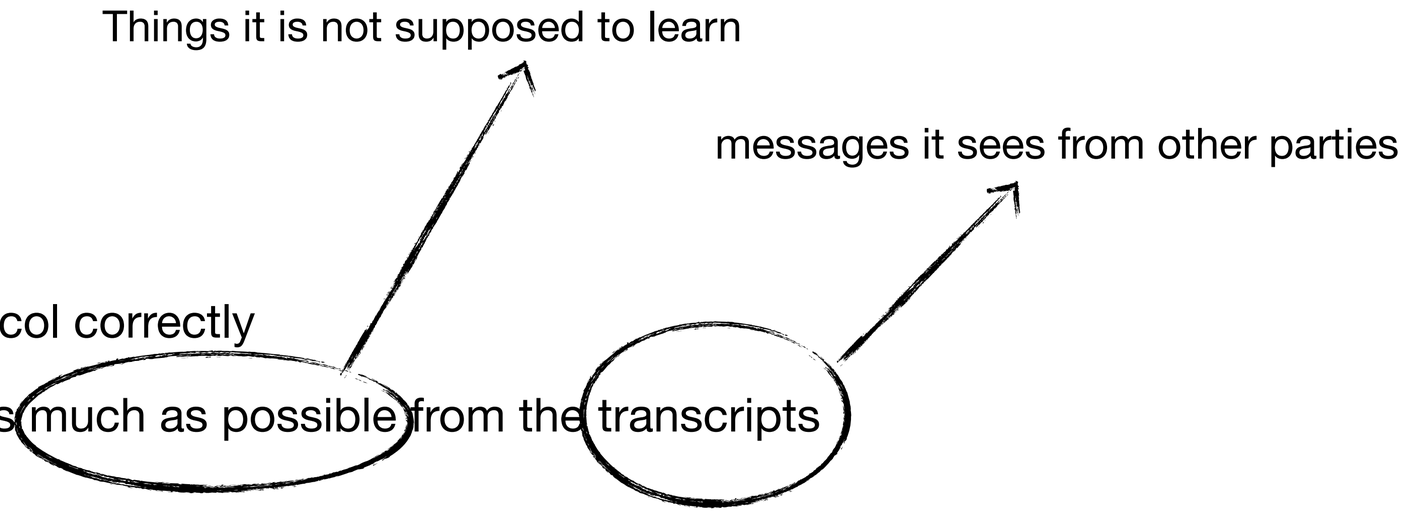
Servers Clients Third Parties

Each of these entities could be adversarial

Threat Model should describe what kind of behaviors we aim to protect against

(… and what kind of behavior we do not aim to protect against)

Threat Model



Semi-honest: Will participate in the protocol correctly

However, will try to learn as much as possible from the transcripts

Malicious: Can deviate from the protocol

Is allowed to do anything to break the protocol

Can send garbage values; stop participating in the protocol itself

Covert: Can deviate from the protocol provided they do not get caught

Threat Model

Typically we start with semi-honest model and then move to other models like malicious

Threat Model

We have so far considered parties to be honest

But, that may or may not be true.

Therefore, we must always describe what our threat model is.