Blockchain 101

2# Secure Distributed Systems

In the last episode…

- We learn the fundamental security properties:
  - **Integrity, Confidentiality and Authenticity**

- These can be guaranteed by cryptographic techniques:
  - **ciphers, hash functions, digital signatures**

Blockchain101: contents

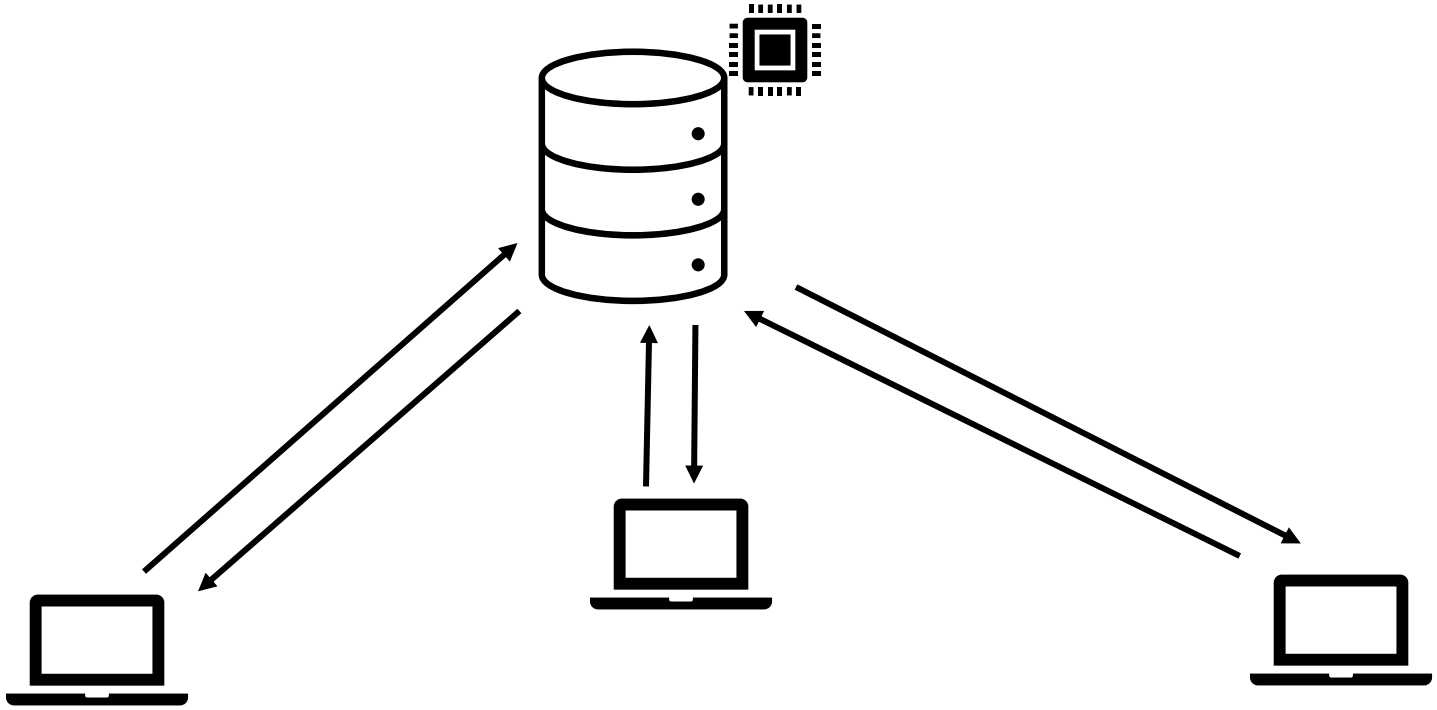
1. ~~Security Fundamental Concepts~~
2. Secure Distributed Systems
3. Blockchain in a Nutshell
4. Assembling the pieces: Blockchain prototype

Distributed Systems

A distributed system can be defined as a (variable sized) group of computers that communicate/work together in a coherent way. Typically, these machines have a shared state and operate concurrently.

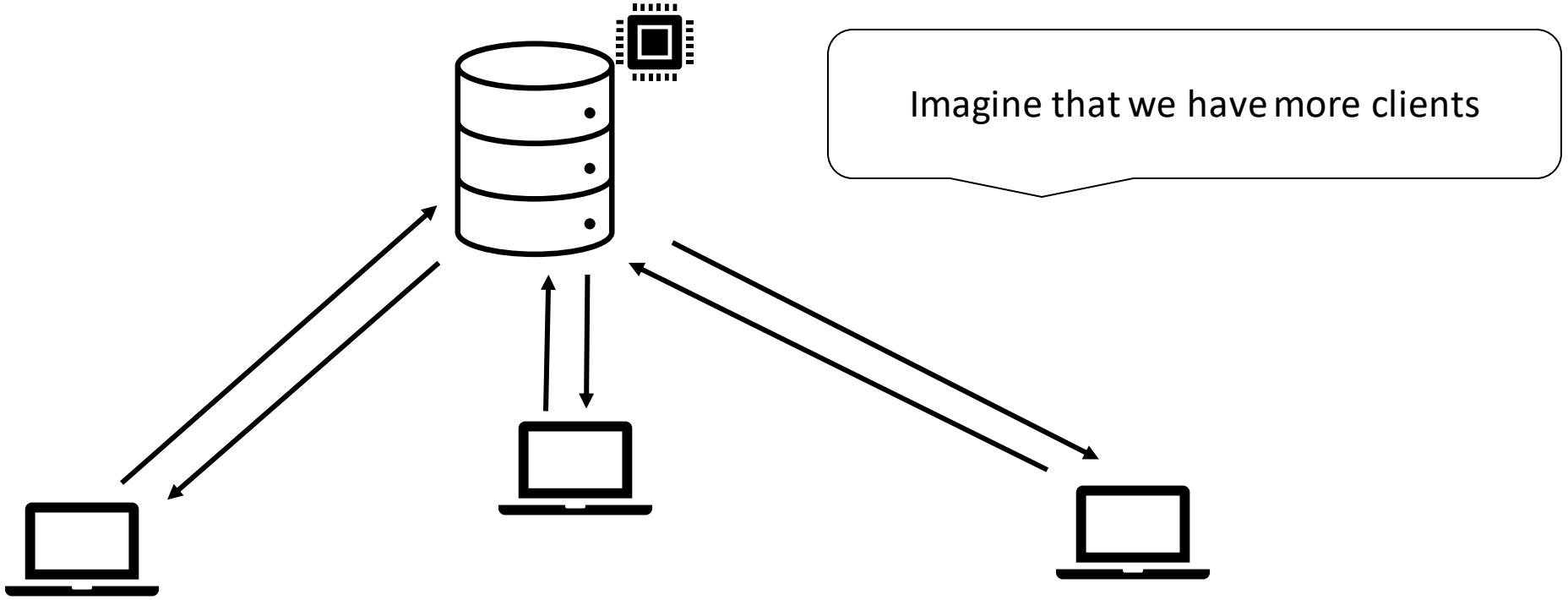
Distributed Systems: Scalling

- Consider a database server that responds to a few client requests

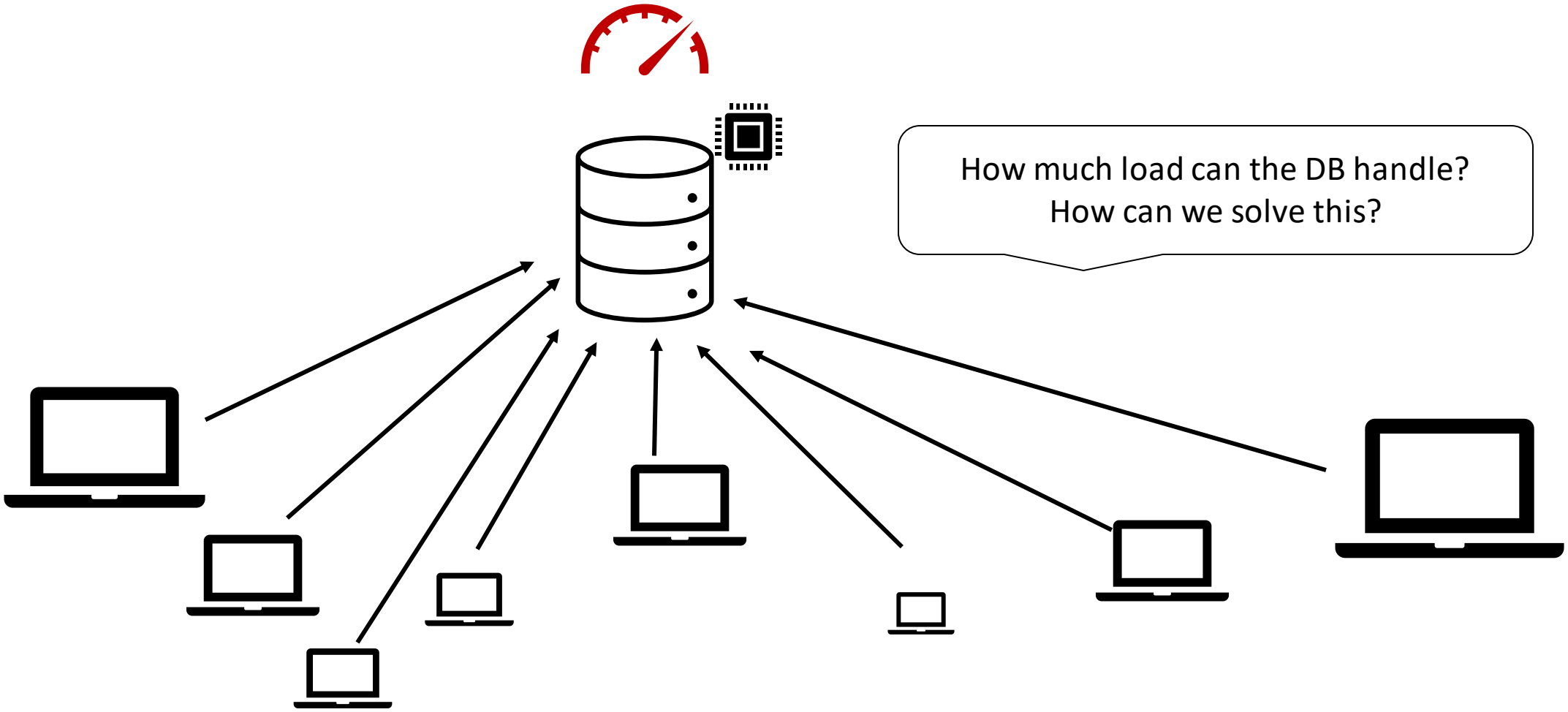


Distributed Systems: Scalling

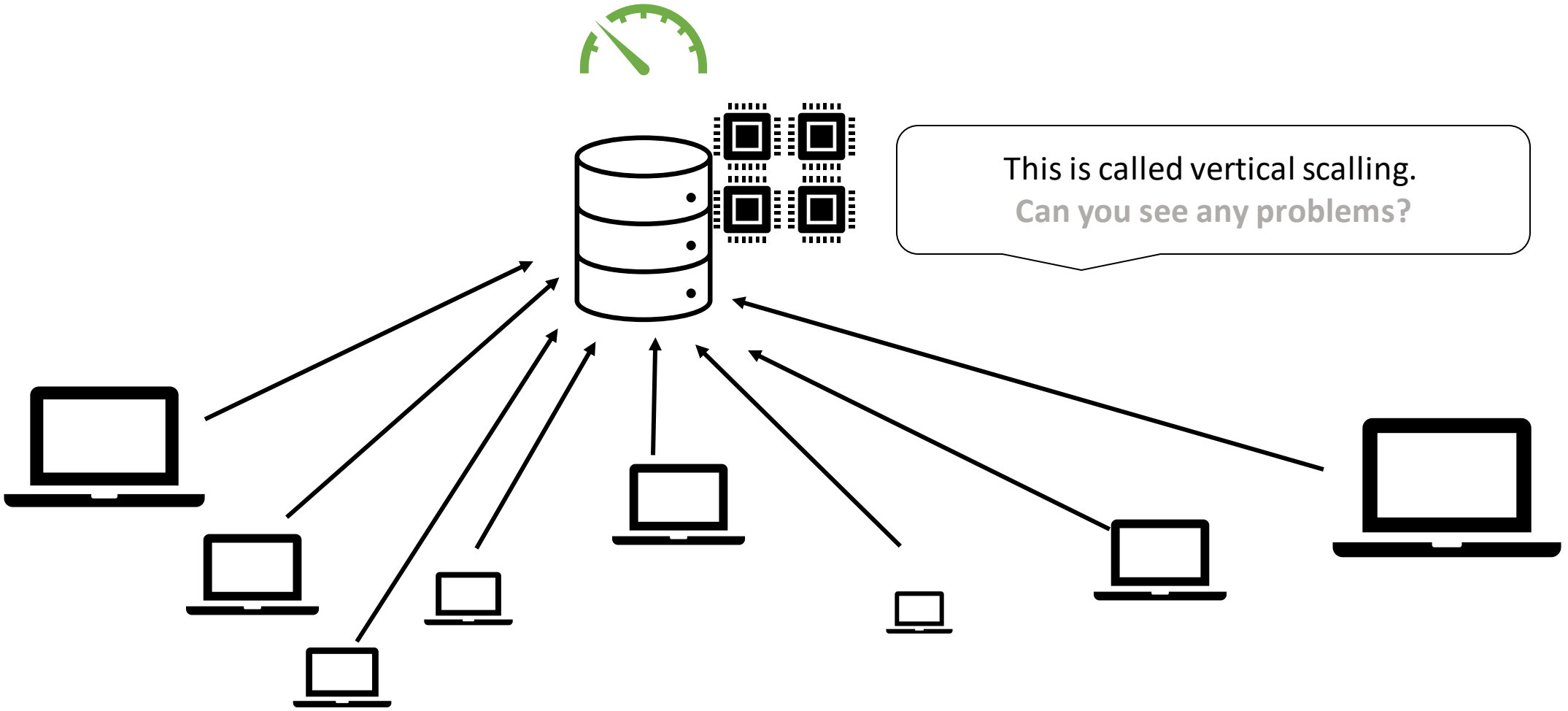
- Consider a database server that responds to a few client requests



Distributed Systems: Scalling

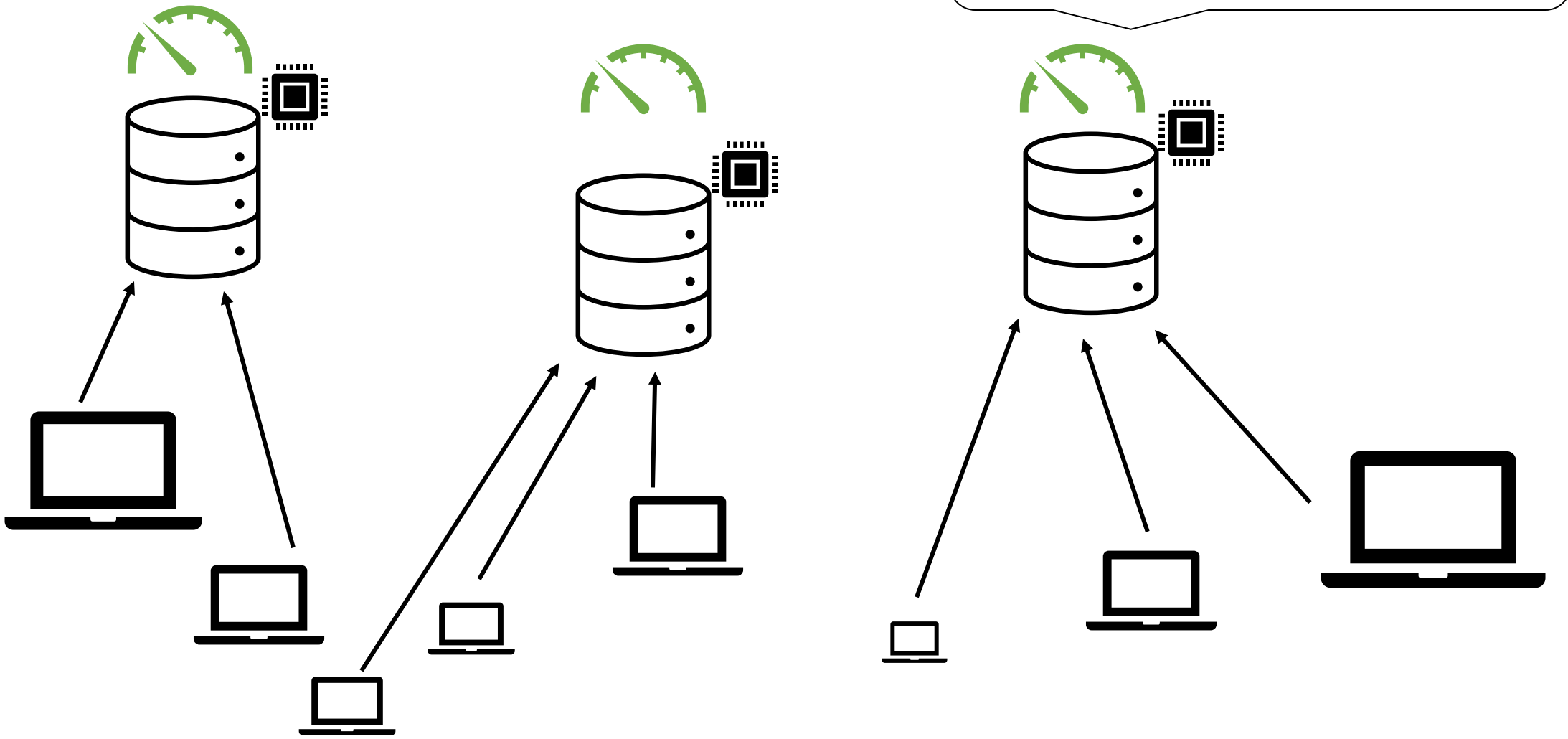


Distributed Systems: Scalling



Distributed Systems: Scalling

This is called horizontal scalling.
virtually infinite
Can you see any problems?



Distributed Systems

- A distributed system is a way to improve performance (increase throughput and decrease latency) and add fault-tolerance to the distributed service
- But there are no free lunches:
  - Harder to manage
  - Harder to secure
  - Inter-node latencies
  - Guarantee properties like ACID (**A**tomicity, **C**onsistency, **I**solation and **D**urability)
  - Harder to synchronize
  - … different types of distributed system offer different challenges

Types of distributed systems

- A distributed system is a way to improve performance (increase throughput and drecrease latency) and add fault-tolerance to the distributed service
- But there are no free lunches:
  - Harder to manage
  - Harder to secure
  - Inter-node latencies
  - Guarantee properties like ACID (**A**tomicity, **C**onsistency, **I**solation and **D**urability)
  - Harder to syncronize
  - … different typs of distributed system offer different challenges

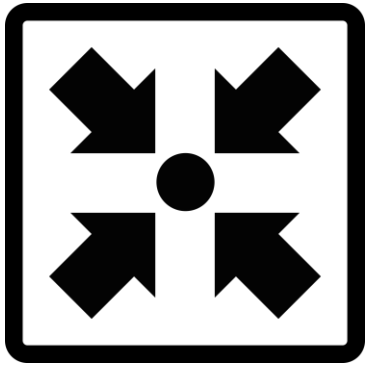
P2P networks

- The most basic definition of a Peer-to-peer network is a network where each node acts as a client and as a server. Thus, there is no need for central server.
- To join a p2p network, a node just needs a network connection (intra/internet) and the software (middleware) to communicate with the other nodes.
  - Some p2p need invites, others allow any node to join
- The p2p network is an overlay over the physical network
- They can be used to store files, process information in a distributed way
- Advantages: resilient and scalable
- There are different algorithms to join and organize p2p networks

Signtures guarantee integrity and authentication.

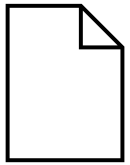
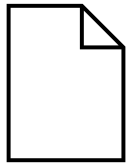
Can we guarantee integrity with a simple solution?

Meeting point slide



Cryptographic primitive Security Goal	Hash	MAC	Digital Signature
Integrity	Yes	Yes	Yes
Authentication	No	Yes	Yes
Non-repudiation	No	No	Yes
Type of key	none	Symmetric	Asymmetric

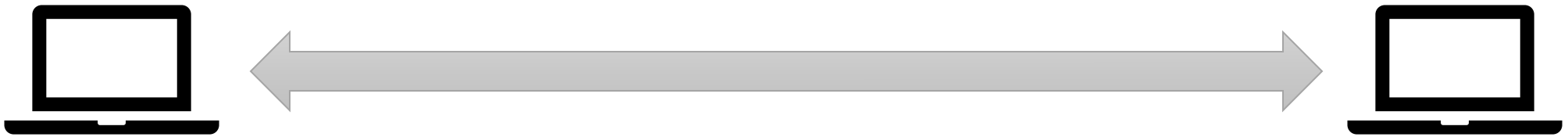
Game time: Rock paper scissors

		winner
		paper
		rock
		scissors

Game time: Rock paper scissors

How can we implement this game in a distributed system?

Game time: Rock paper scissors



Alice

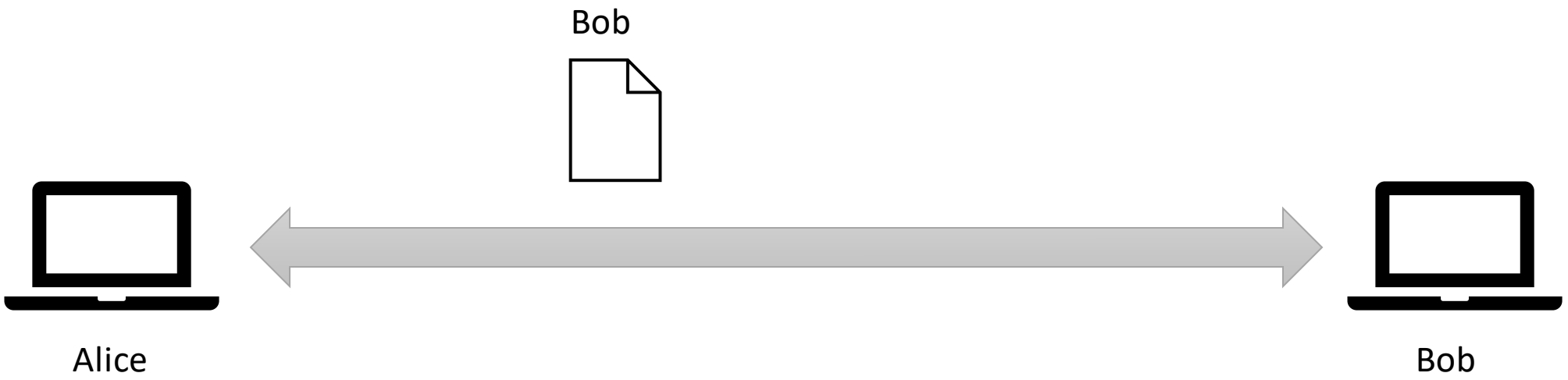
They communicate with a reliable and insecure channel
They do not have syncronyzed clocks

Bob

Game time: Rock paper scissors

Game time: Rock paper scissors

Game time: Rock paper scissors



Game time: Rock paper scissors

Game time: Rock paper scissors

Game time: Rock paper scissors

Alice

$%DLAS

Alice

Bob

Hash( ) = $%DLAS

Can you see the problem?

Game time: Rock paper scissors

Alice

$%DLAS

Alice

Hash( ) = $%DLAS

Bob

Hash( ) = $%DLAS

Hash( ) = ?L$LAS

Hash( ) = #AdD3S

Game time: Rock paper scissors

Alice

$%DLAS

Alice

Hash( ) = $%DLAS

Suggestions?

Bob

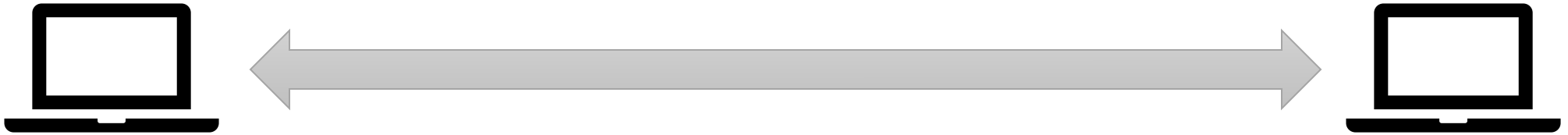
Hash( ) = $%DLAS

Hash( ) = ?L$LAS

Hash( ) = #AdD3S

Game time: Rock paper scissors

Here we need to define a protocol:
1) Each play sends the Hash
2) when they receive the other's Hash, he/she can send the random secret



Alice

Bob

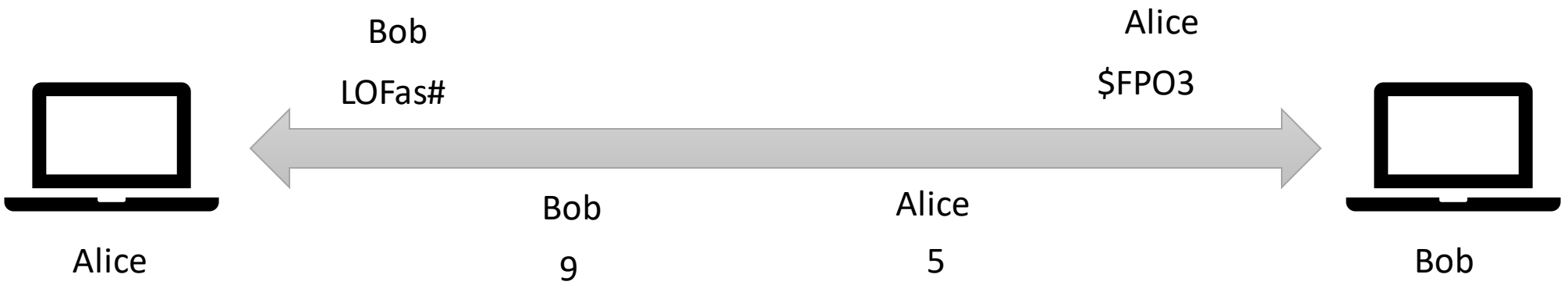
Random() = 5

Hash(+5) = $%DLAS

Random() = 9

Hash(+9) = ?L$LAS

Game time: Rock paper scissors



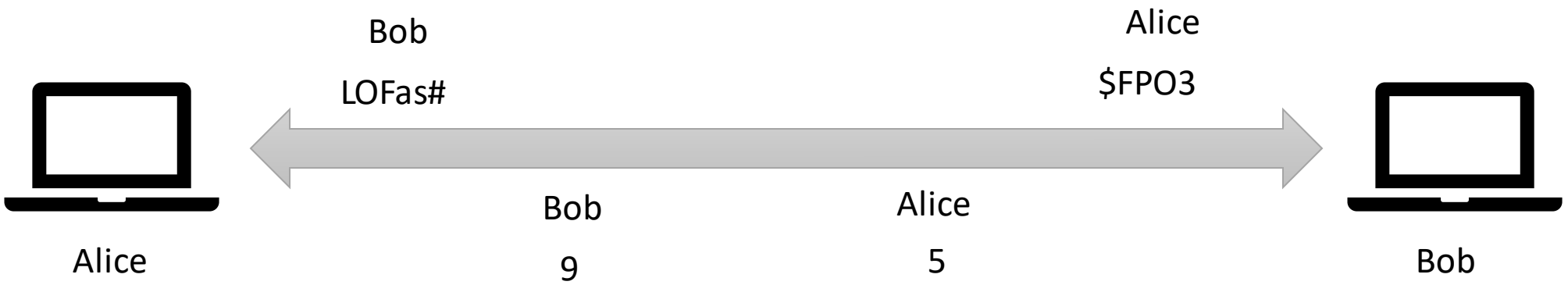
Random() = 5

Hash(+5) = $FPO3

Random() = 9

Hash(+9) = LOFas#

Game time: Rock paper scissors



Random() = 5

Hash(+5) = $FPO3

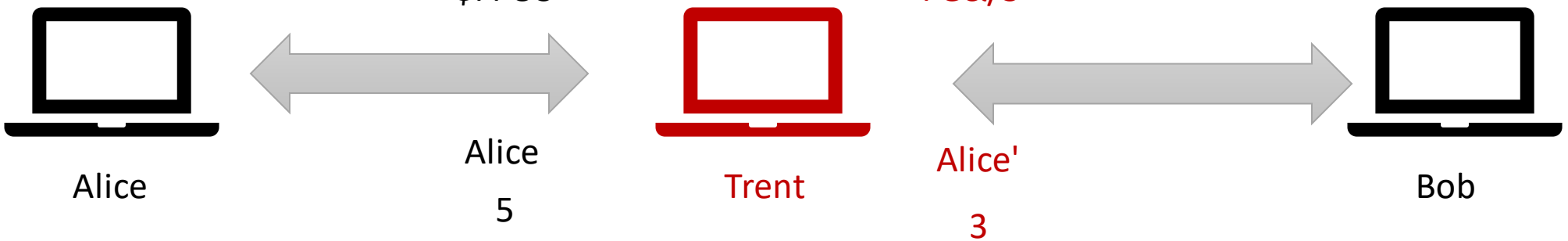
It supports weak* confidentiality...
but does it guarantee authenticity?
*It is hard to make secure randoms

Random() = 9

Hash(+9) = LOFas#

Game time: Rock paper scissors

Man-in-the-middle attack



Game time: Rock pape

Man-in

Long story short:
We could HMAC, they guarantee authenticity, but they do not provide non-repudiation, moreover HMAC don't allow more than 2 player 'cause it uses symmetric keys.
So how can we guaratee authentication?

Random() = 5

Hash(+5) = $FPO3

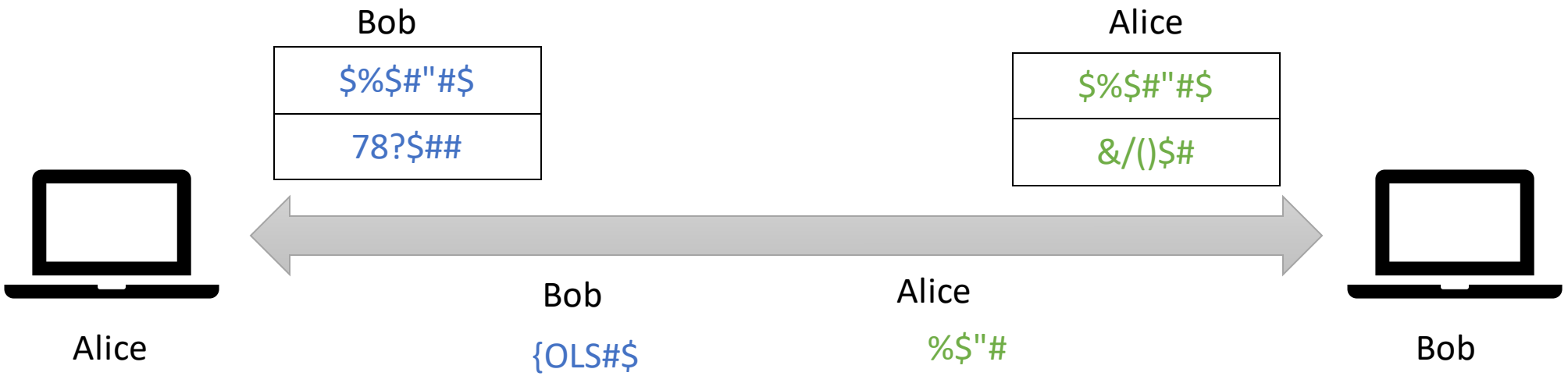
Random() = 3

Hash(+3) = FG&/3

Random() = 9

Hash(+9) = LOFas#

Game time: Rock paper scissors



Random() = 5

Hash(+5) = $FPO3

E(K, $FPO3) = $%$#"#$

S($%$#"#$, KAp) = &/()$#

S(5, KAp) = %$"#

Random() = 9

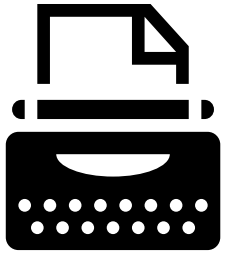
Hash(+9) = LOFas#

E(K, LOFas#) = $%$#"#$

S($%$#"#$, KBp) = 78?$##

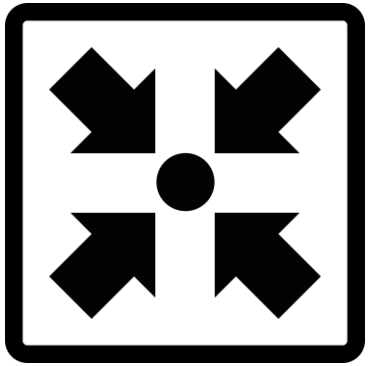
S(9, KBp) = {OLS#$

Coding time

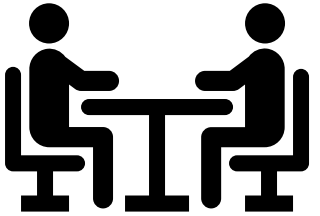


- Implement Rock, paper, scissors game
  - Socket communication
  - Encryption
  - Digital Signatures

Meeting point slide



- Cryptographic hash functions are one-way functions
  - This means it (should be) is impossible to find the input based on the output
- They are also deterministic
  - This means that for the same input the output is always the same
- The hash function output has a fixed size output, the input can have any size
- It guarantees integrity, it can be used – with caution – to encrypt data



Discussion slide

Can you think of a use for hash functions in digital signatures?