CRYPTOGRAPHY

(Lecture 2)

Literature:

"Handbook of Applied Cryptography" (ch 6.1 Note on OTP)
"Lecture Notes on Introduction to Cryptography" by V. Goyal (**ch1.2,1.3**, 3.5,3.7,4.0,4.1,4.2)
"A Graduate Course in Applied Cryptography" by D. Boneh and V. Shoup (ch2-2.2.2, **3.1**)

Announcements

- Typo on HA1: $c \in \{0,1\}^X$ (pdf updated yesterday evening)
- For questions on HA1 contact **Victor** or **Oscar**
- “Discussions” are now available on Canvas (pairing up)
- How do I prepare for the final exam? For now: Solve the weekly exercises

Lecture Agenda

Recap From Last Lecture

Blockchain Technology

- Digital Bulletin Boards
- Cryptographic Puzzles & Proof of Work

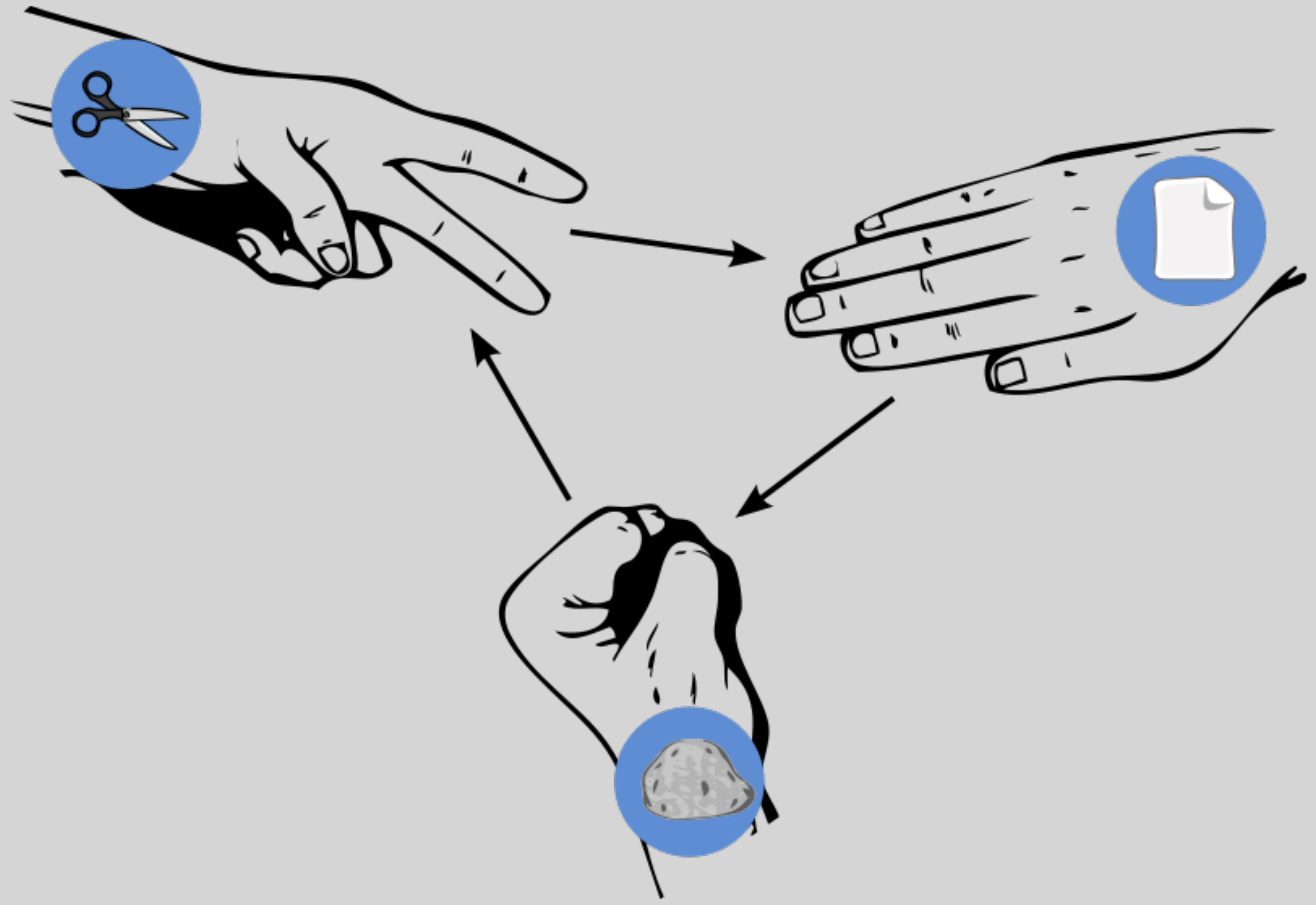
Perfect Secrecy

- Symmetric Encryption
- The One Time Pad (OTP) [Proof]
- Perfect Secrecy
- Shannon's Theorem [Proof]

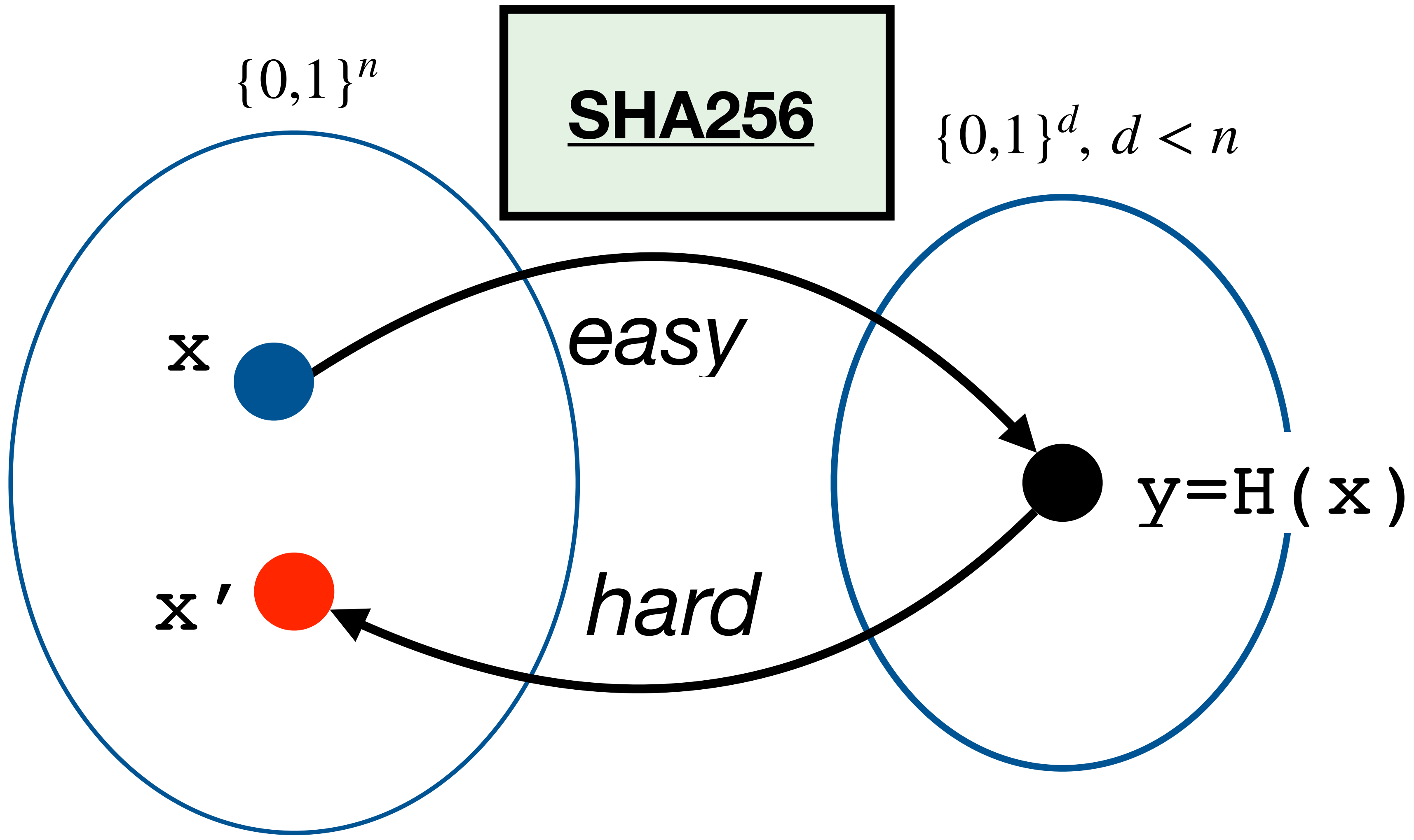
Pseudorandom Generators (PRG)

- Definition
- Security
- Secure Encryption From PRG
- Semantic Security [Proof]

Home Assignment 1



Deadline: Nov 15th (1st Submission)



Lecture Agenda

Recap From Last Lecture

Blockchain Technology

- Digital Bulletin Boards
- Cryptographic Puzzles & Proof of Work

Perfect Secrecy

- Symmetric Encryption
- The One Time Pad (OTP) [Proof]
- Perfect Secrecy
- Shannon's Theorem [Proof]

Pseudorandom Generators (PRG)

- Definition
- Security
- Secure Encryption From PRG
- Semantic Security [Proof]

Hash Functions Quick Recap

<u>Definition: Collision Resistant HASH FUNCTION</u>

A function $H : \{0,1\}^n \to \{0,1\}^d$ is a collision resistant hash function if:

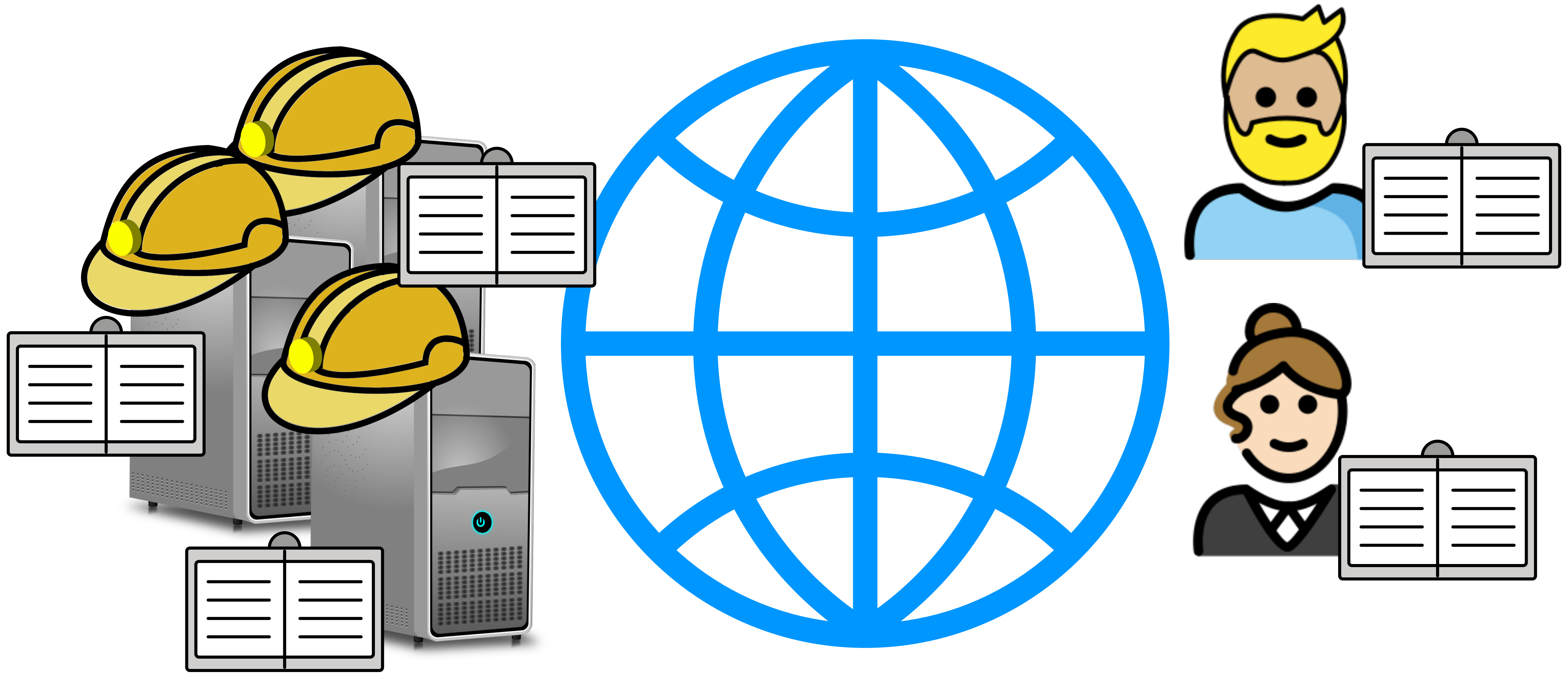
It is compressing (i.e., $n > d$), it is one-way (efficient to compute, hard to invert), and

$$Pr[f(x) = f(x') \,|\, x, x' \leftarrow \mathcal{A}(f), x \neq x'] \leq negl(n)$$

Collision Resistant Hash Functions are at the Core of how BitCoin works

Basics of Cryptocurrencies

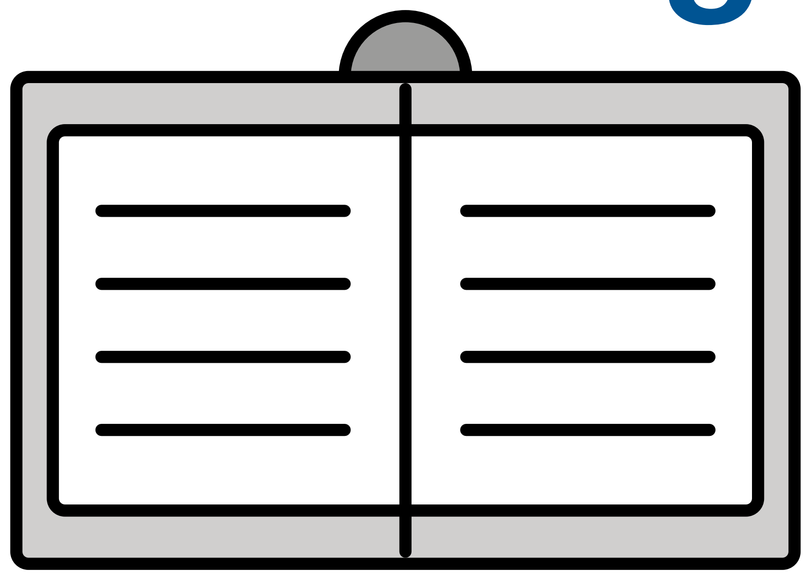
replace the bank with
miners and
a **bulletin board**



Initial challenges

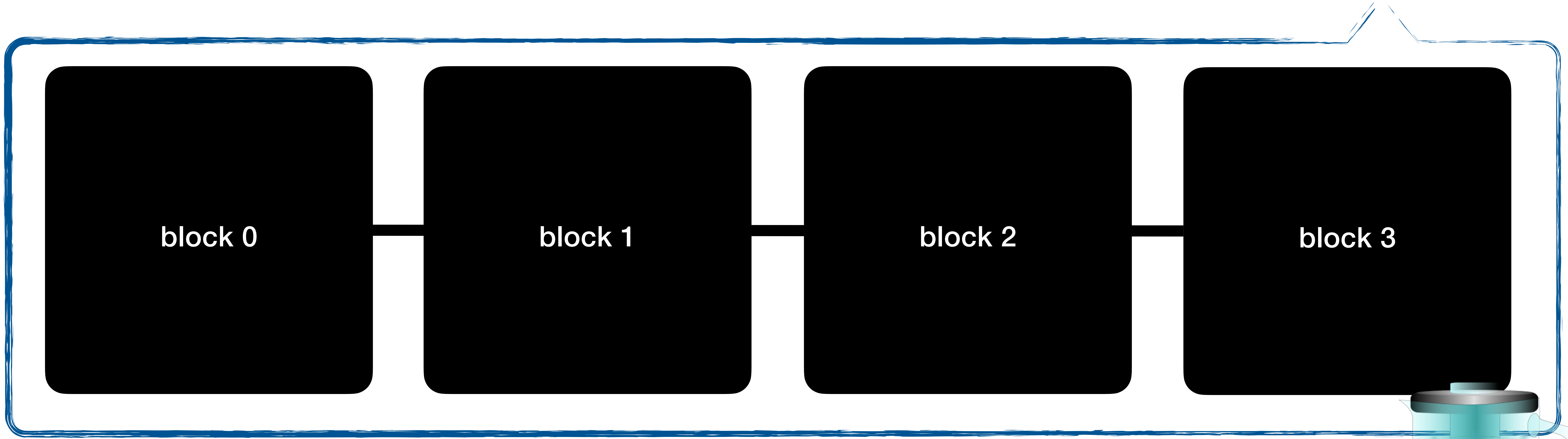
1) How to **create** a digital bulletin board (distributed **ledger**)?
2) How to **agree** on **one** ledger view?

How To Create a Digital Bulletin Board (Distributed Ledger)?



The main property to implement is record keeping

1) the past is immutable
2) everyone agrees on the history



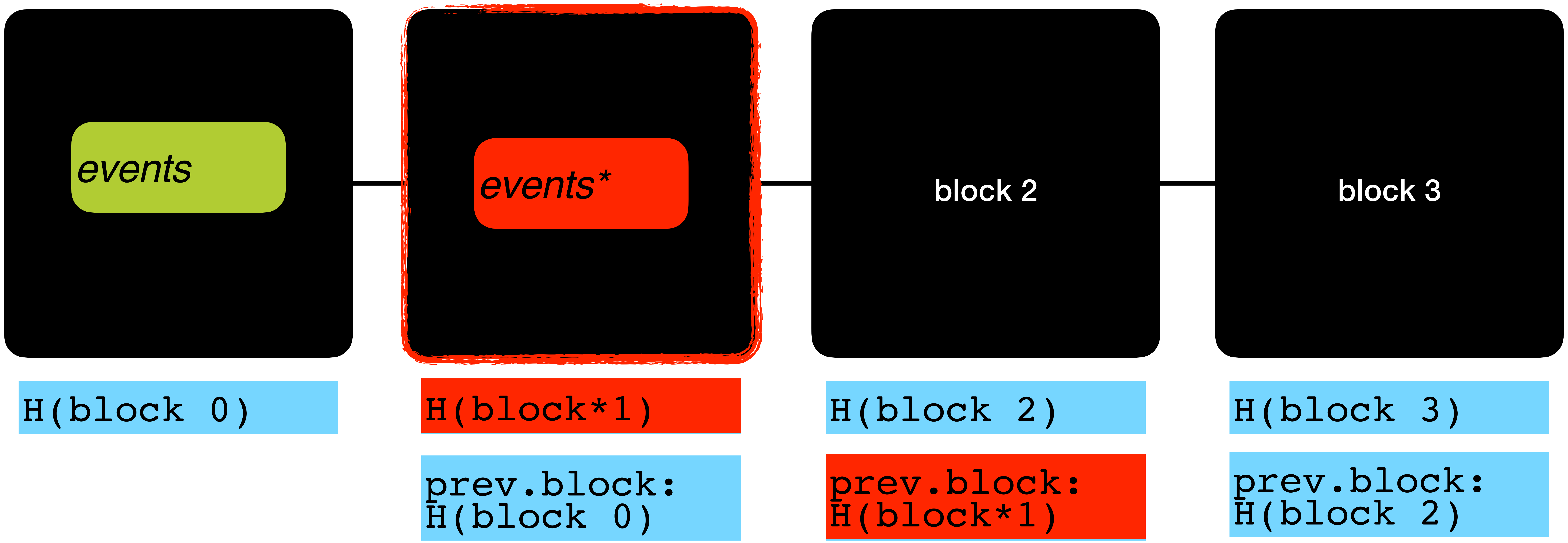
- Partition time into époques / periods / time windows
- Anything that happens in one time period is recorded into a block
- Any change to an ‘old’ block affects all following blocks

how can we implement this property using a cryptographic object?

How To Set Up a Bulletin Board

Main property we want to implement = record keeping

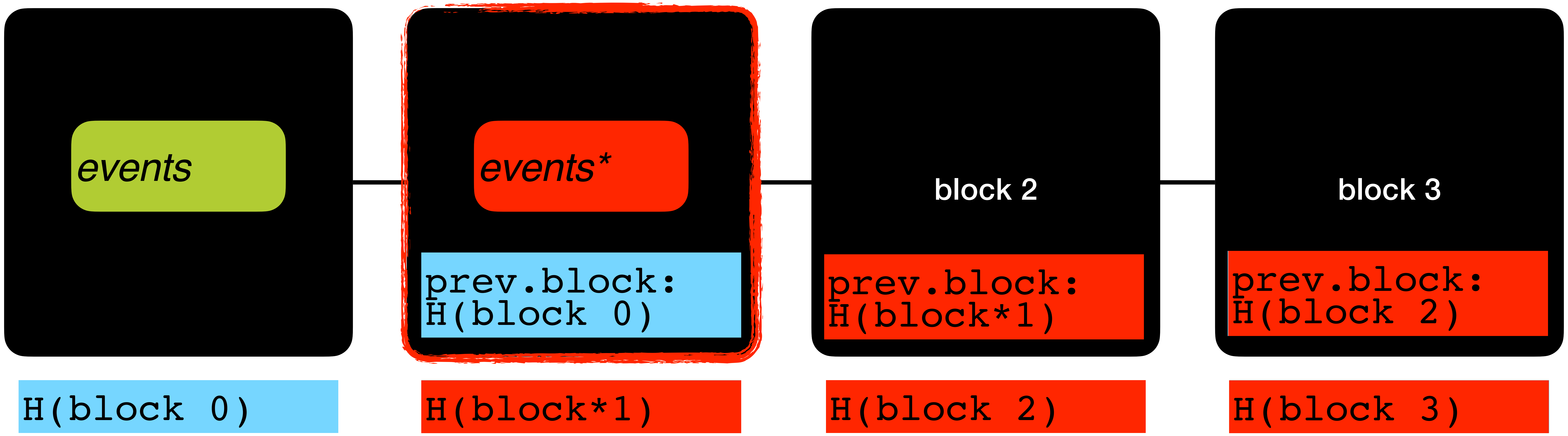
1) the past is immutable
2) everyone agrees on the history



How To Set Up a Bulletin Board

Main property we want to implement = record keeping

1) the past is immutable *use the hash function to **chain** blocks*
2) everyone agrees on the history

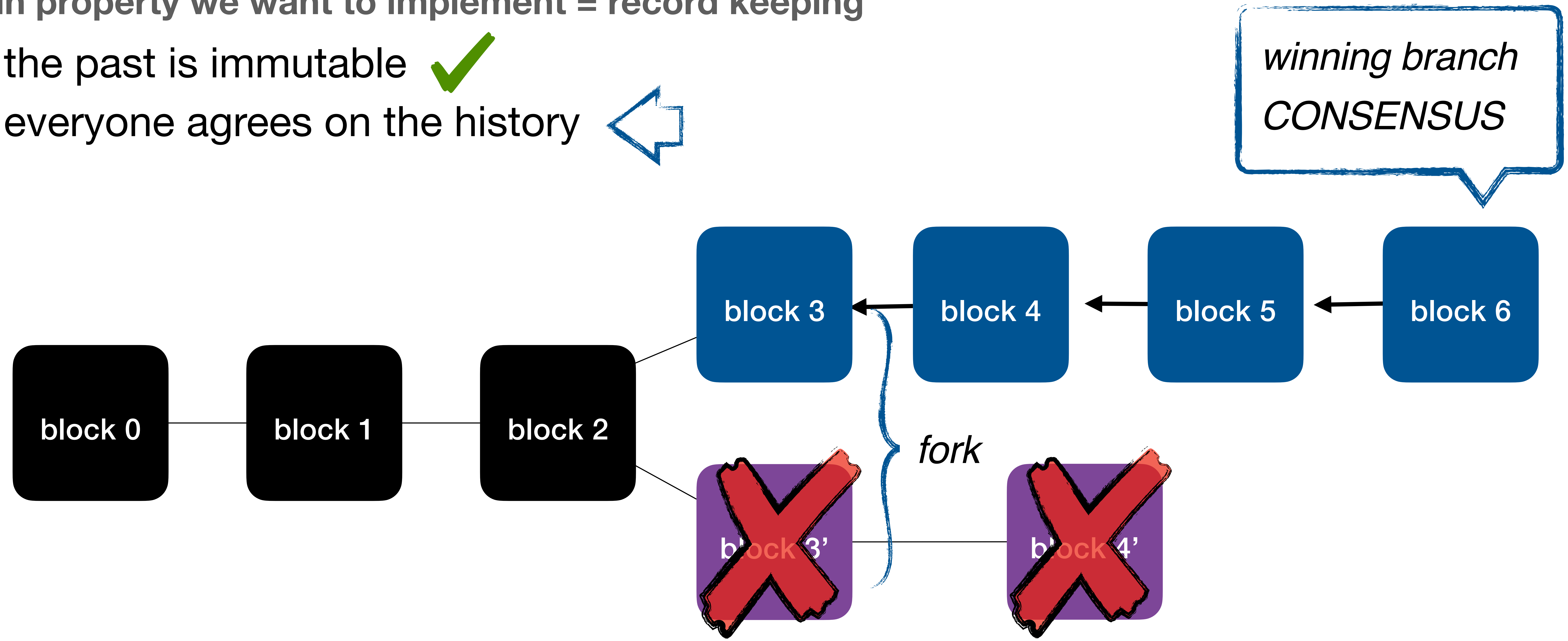


Any change to an ‘old’ block affects all following blocks

How To Set Up a Bulletin Board

Main property we want to implement = record keeping

1) the past is immutable
2) everyone agrees on the history



- Always build on the longest branch (longest chain rule)
- How to lower the chance that blocks appear at the same time?

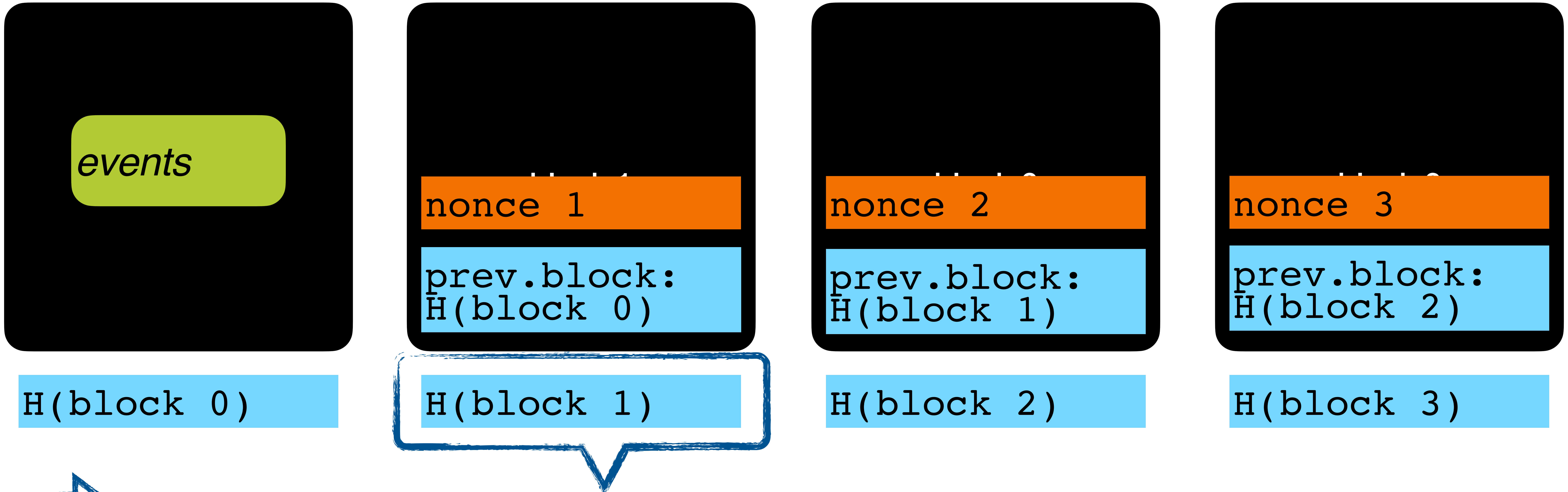
Proof of Work

Proof of Work (Cryptographic Hash Puzzles)

Main property we want to implement = record keeping

1) the past is immutable ✓
2) everyone agrees on the history

put a rule that makes it "hard" to compute a "good" hash digest



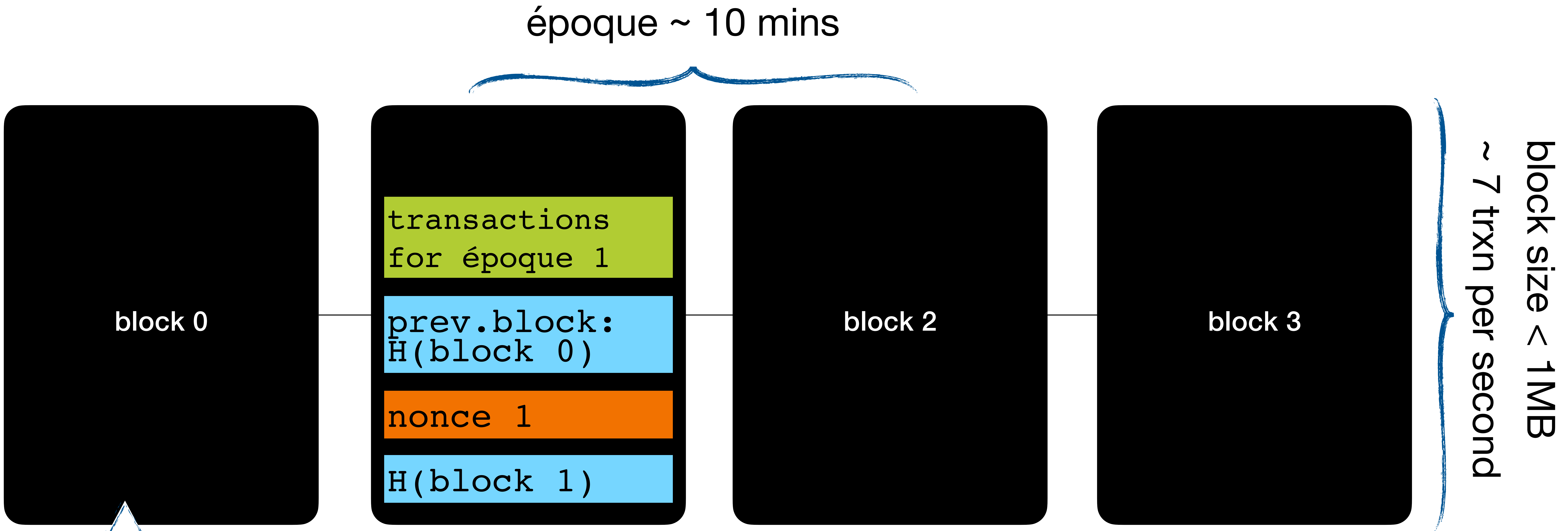
*RULE : EACH BLOCK HASH NEEDS TO START WITH A GIVEN **PREFIX***

find a value `nonce` such that `H(block||nonce) = 00000****`

```
sha256(I love Crypto!) = e8f6178df67ea4ec791b9fd72a2d710a3d832c113ee933a0654ae0e423d49ac9
sha256(I love Crypto!-251509386766) = 0000092273023b5bc71c29852a01d0121336c16e700535cca2a8c5ef1459becd
```

Example: Bitcoin

set by the puzzle difficulty
(currently 19 leading zeroes)



genesis block
created by Nakamoto
on 03.01.2009

🧐 *How <u>large</u> is the BitCoin Ledger?*

Lecture Agenda

Recap From Last Lecture

Blockchain Technology

- Digital Bulletin Boards
- Cryptographic Puzzles & Proof of Work

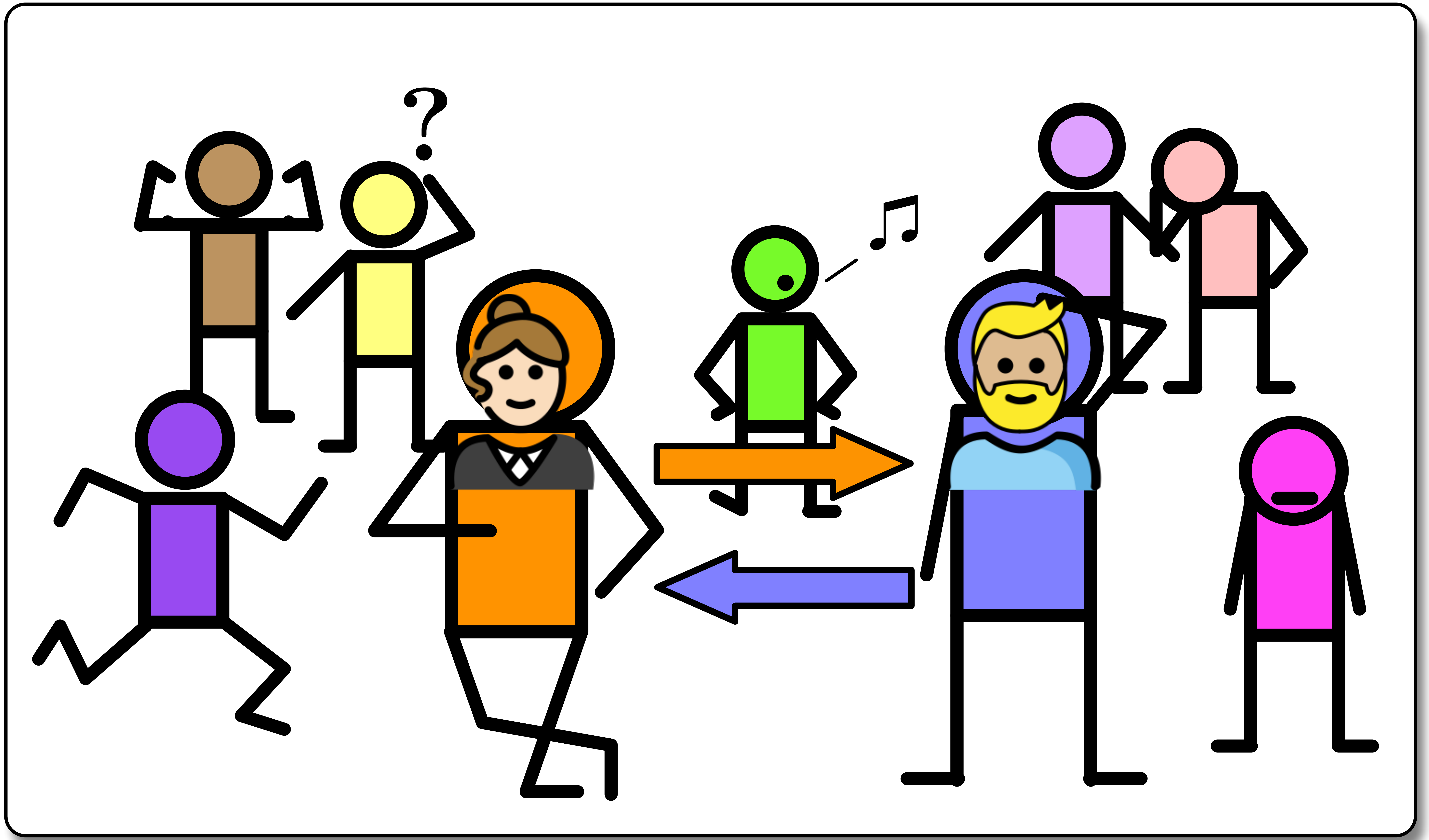
Perfect Secrecy

- Symmetric Encryption
- The One Time Pad (OTP) [Proof]
- Perfect Secrecy
- Shannon's Theorem [Proof]

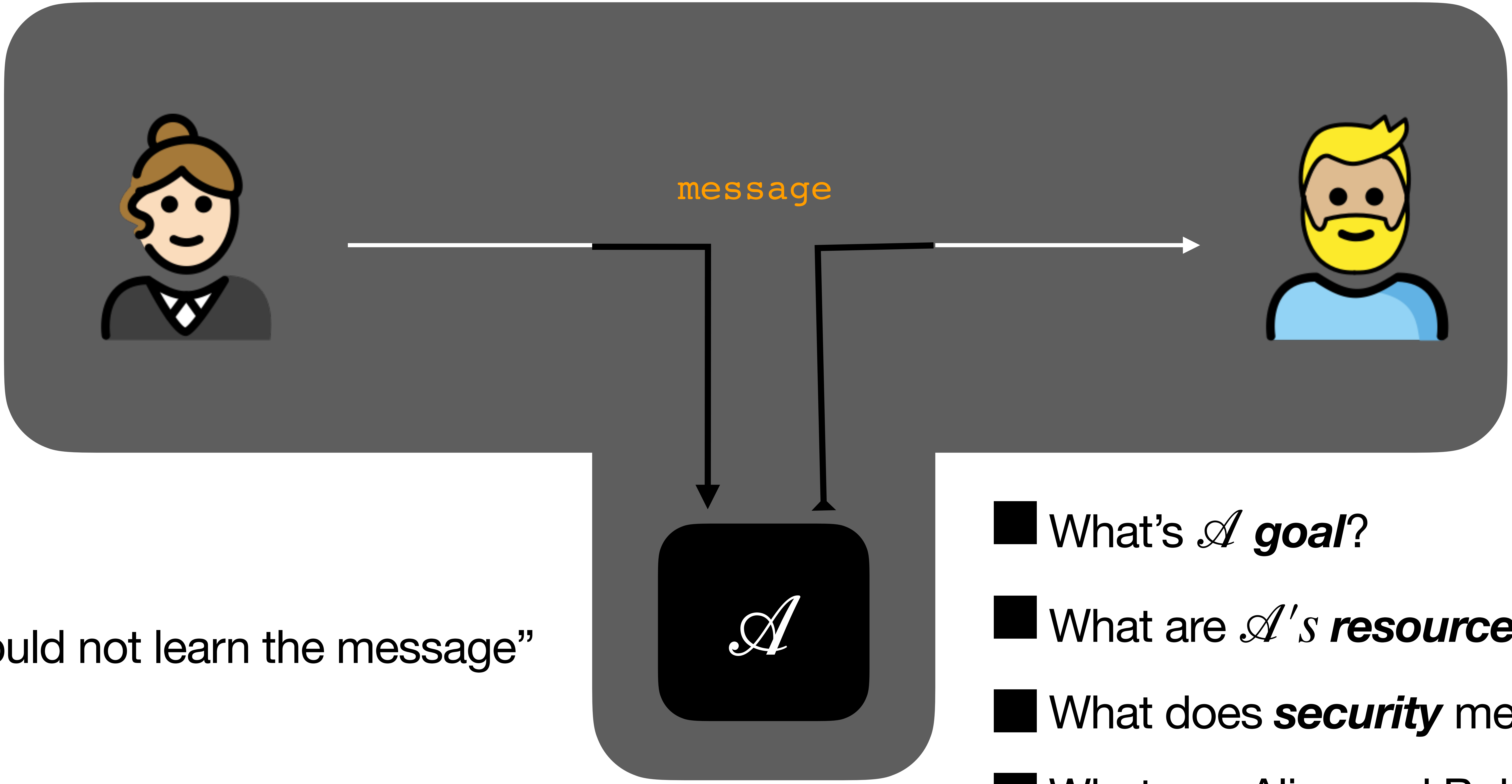
Pseudorandom Generators (PRG)

- Definition
- Security
- Secure Encryption From PRG
- Semantic Security [Proof]

Secure Communication Over an Insecure Channel



Secure Communication Over an Insecure Channel



"$\mathscr{A}$ should not learn the message"

- What's $\mathscr{A}$ ***goal***?
- What are $\mathscr{A}$'s ***resources***?
- What does ***security*** mean?
- What can Alice and Bob use?

Let's start with: a **symmetric encryption** scheme

Symmetric Encryption - Syntax

Definition: Symmetric Encryption

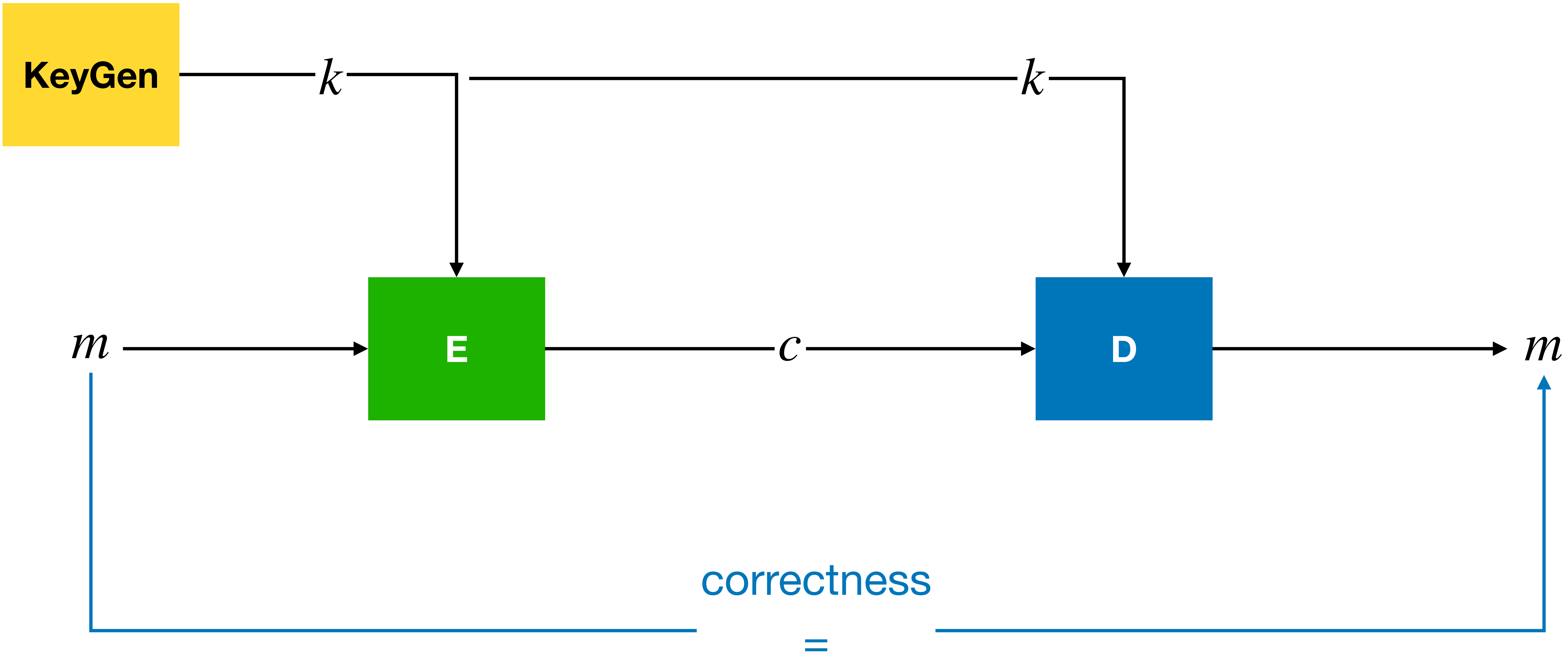
A tuple (KeyGen, E, D) is a symmetric encryption scheme over the sets $\mathscr{K}$ (key space), $\mathscr{M}$ (message space), and $\mathscr{C}$ (cihpertext space) if all algorithms are efficient and satisfy the following:

KeyGen(1^n) → k : the key generation is a randomised algorithm that returns a key k. (This algorithm is often implicit when $k \leftarrow \$\mathscr{K}$)
E(k,m) → c : the encryption is a possibly randomised algorithm that on input a key k and a (plaintext) message m, outputs a ciphertext c.
D(k,c) → m : the decryption is a deterministic algorithm that on input a key k and ciphertext c, outputs a plaintext message m.

CORRECTNESS:

$Pr[\ D(k, E(k, m)) = m \ \ | k \leftarrow KeyGen(1^n)] = 1$ …. for all messages $m \in \mathscr{M}$

Symmetric Encryption - Visualisation



Symmetric Encryption - the One Time Pad (OTP)

Definition: Symmetric Encryption

A tuple (KeyGen, Enc, Dec) is a sym
sets $\mathscr{K}$ (key space), $\mathscr{M}$ (message sp
algorithms are efficient and satisfy the

KeyGen(1^n) → k : the key generat
returns a key k. (This algorithm is often
E(k,m) → c : the encryption is a pos
input a key k and a (plaintext) message
D(k,c) → m : the decryption is a deter
k and ciphertext c, outputs a plaintext message m.

Example: the One Time Pad (OTP)

$\mathscr{K} = \mathscr{M} = \mathscr{C} = \{0,1\}^n$

$KeyGen(1^n) \rightarrow k \ (where\ k \leftarrow \$\{0,1\}^n)$

$E(k, m) = k \oplus m$

$D(k, c) = k \oplus c$

CORRECTNESS:

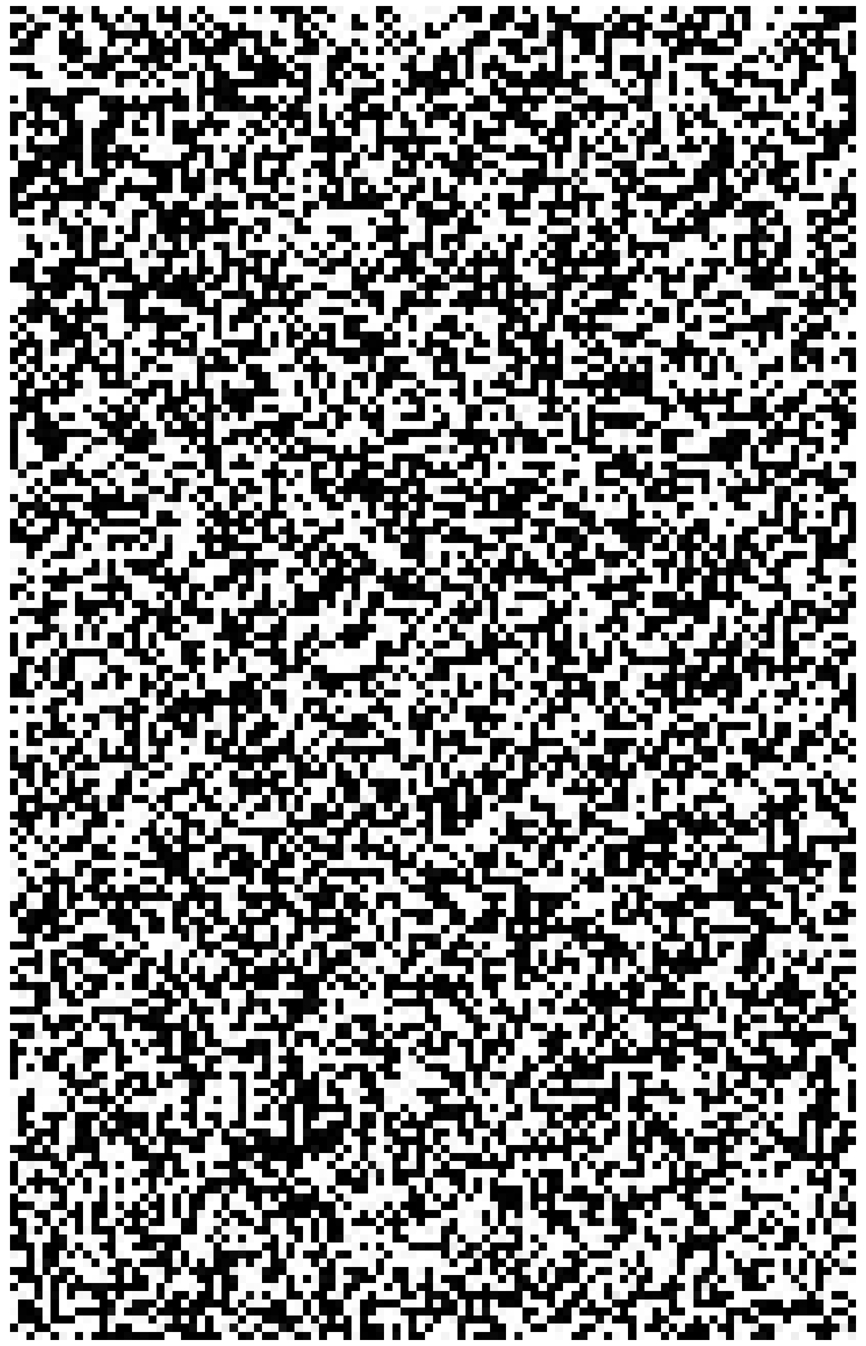
$Pr[\ D(k, E(k, m)) = m \ \ |k \leftarrow KeyGen(1^n)] = 1$ …. for all messages $m \in \mathscr{M}$

OTP From the Attacker’s Point of View

Uniformly random key

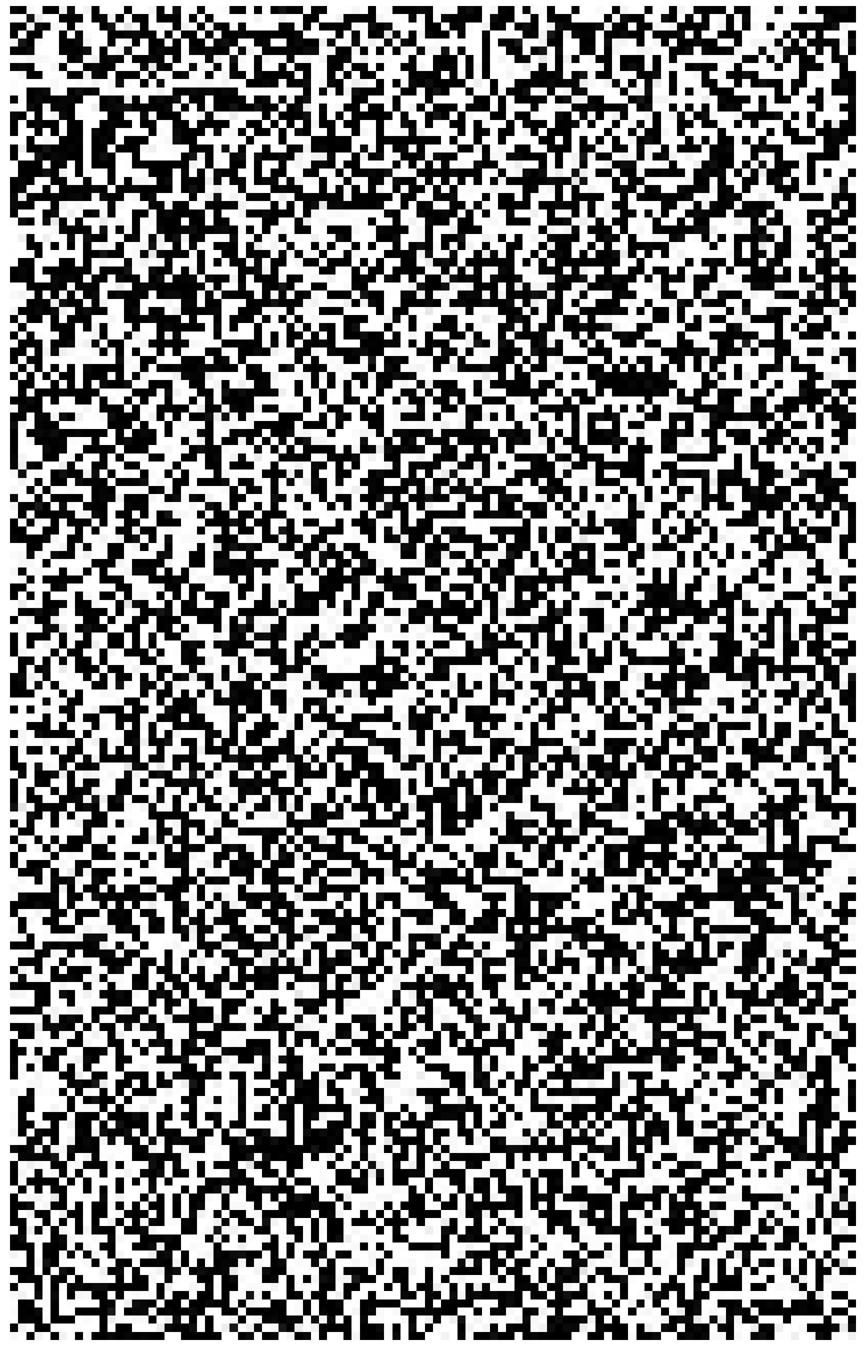
$$k \quad \oplus \quad m \quad = \quad c$$

Uniformly random ciphertext



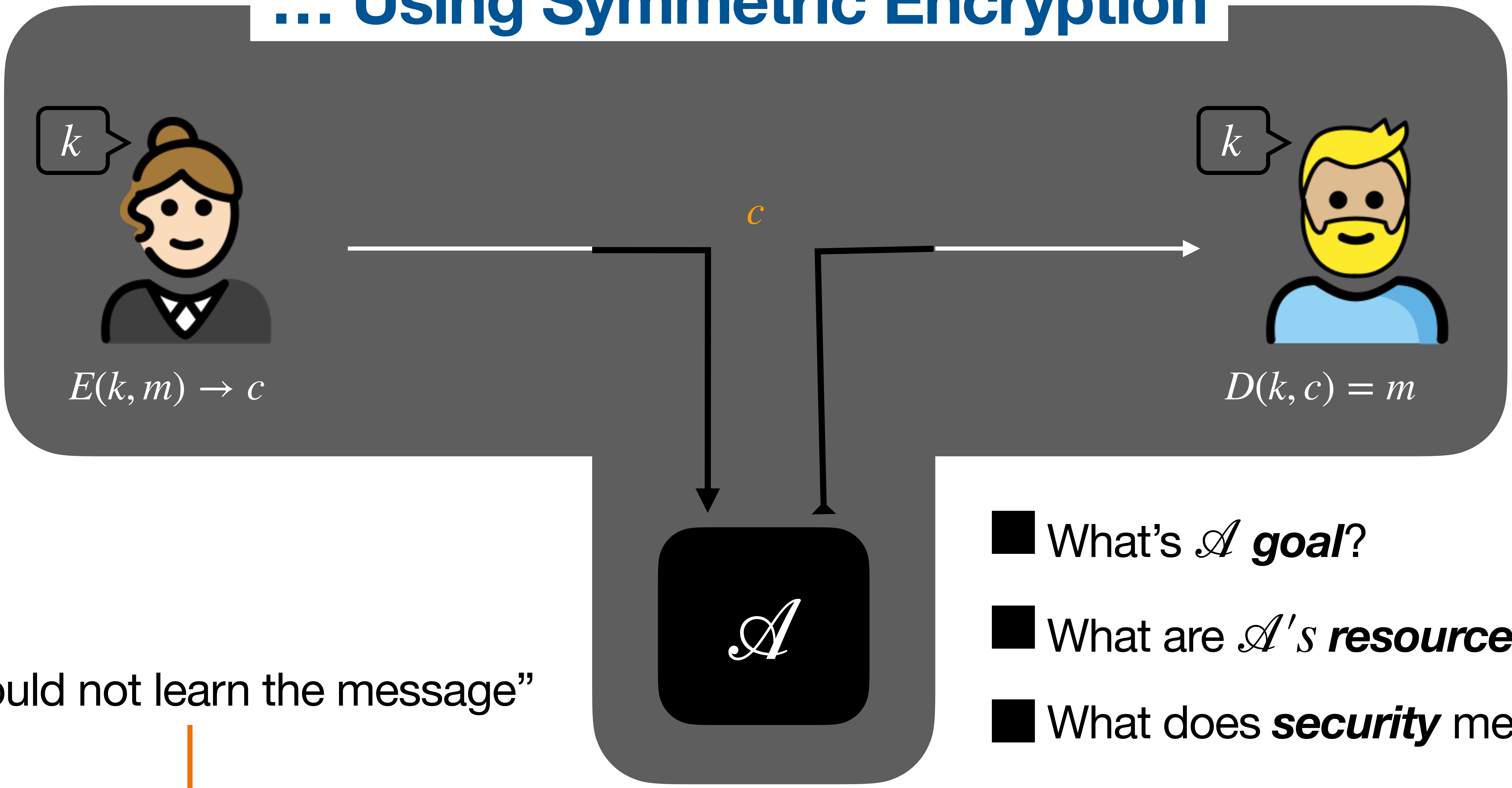
$\oplus$

$=$



Secure Communication Over an Unsecured Channel

… Using Symmetric Encryption



- What's $\mathscr{A}$ ***goal***?
- What are $\mathscr{A}$'s ***resources***?
- What does ***security*** mean?

"$\mathscr{A}$ should not learn the message"

"The ciphertext c should not leak any information about the message m"

Perfect Secrecy

Definition: Perfect Secrecy (Perfect Security)

A symmetric encryption scheme $(KeyGen, E, D)$ is perfectly secret if for all pair of messages $m_0, m_1 \in \mathscr{M}$ and for all ciphertexts c it holds that:

$$Pr[E(k, m_0) \rightarrow c \,|\, k \leftarrow KeyGen(1^n)] = Pr[E(k, m_1) \rightarrow c \,|\, k \leftarrow KeyGen(1^n)]$$

This is an example of unconditional security

This security notions essentially states that:

An attacker who does not know *k* learns nothing *new* about the plaintext *m* from seeing *c*.

The OTP Is Perfectly Secret

Proof: In the OTP, for every m and c there is exactly one key $k(= m \oplus c)$ such that $c = E(k, m)$.

Thus $\Pr[c = E(k, m)] = 1/|\mathscr{K}|$.

Hence: $Pr[E(k, m_0) \to c \mid k \leftarrow KeyGen(1^n)] = \dfrac{1}{|\mathscr{K}|} = Pr[E(k, m_1) \to c \mid k \leftarrow KeyGen(1^n)]$

$\boxed{\text{Case } n = 1}$

m	k	$c = m \oplus k$	$\Pr[(m, k)]$
0	0	0	$p(0) \cdot 1/2$
0	1	1	$p(0) \cdot 1/2$
1	0	1	$p(1) \cdot 1/2$
1	1	0	$p(1) \cdot 1/2$

So, $\Pr[c = 0] = p(0)/2 + p(1)/2 = (p(0) + p(1))/2 = 1/2.$

c is uniformly distributed and independent of m!

One Time Pad: Problems

1- The key is as long as the message

2- The key should only be used to encrypt ONE message

Assume that the same key is used twice, i.e. $c_0 = k \oplus m_0$ and $c_1 = k \oplus m_1$ and an adversary gets hold of the two ciphertexts. He can then compute

$$c_0 \oplus c_1 = (k \oplus m_0) \oplus (k \oplus m_1) = m_0 \oplus m_1.$$

$m_0 \oplus m_1$ conveys a lot of information about m_0 and m_1, so this is unacceptable.

3- The ciphertext is (intentionally!) malleable

$$E(k,m) \to c \quad \xrightarrow{\;c\;} \quad \boxed{\mathscr{A}} \quad \xrightarrow{\;c^* = c \oplus e\;} \quad D(k, c^*) = m \oplus e$$

Shannon's Theorem

Theorem (Shannon 1940s)
A symmetric encryption scheme $(KeyGen, E, D)$ define over $(\mathscr{K}, \mathscr{M}, \mathscr{C})$has perfect security if and only if $|\mathscr{K}| \geq |\mathscr{M}|$.

Proof: Fix an arbitrary $m_0 \in M$ and $k_0 \in K$, and let $c_0 = E(k_0, m_0)$. Since the cipher has perfect secrecy, for any $m \in M$ we have when $k \leftarrow \$\mathscr{K}$ that
$Pr[c_0 = E(k, m)] = Pr[c_0 = E(k, m_0)] > 0$.
Thus for each $m \in \mathscr{M}$ there is a key $k \in \mathscr{K}$ such that $E(k, m) = c_0$.
But these keys must all be different; if there was a key $k$and plaintexts $m_1$and m_2 such that $E(k, m_1) = E(k, m_2) = c_0$, then we lose correctness (the decryption of c_0 for that key becomes ambiguous). Thus $|\mathscr{K}| \geq |\mathscr{M}|$.

Take away: ***perfect security is impractical***

How close to perfect security can we go, while being practical?

A Little Secret: the Core of Crypto Is Randomness

This is the goal of Pseudo Random Generators (PRG)

The perfect secrecy of OTP comes from using one random key to mask/hide one message
We cannot reuse the key (otherwise we lose security) but can we '*expand*' it?

Lecture Agenda

Recap From Last Lecture

Blockchain Technology
- Digital Bulletin Boards
- Cryptographic Puzzles & Proof of Work

Perfect Secrecy
- Symmetric Encryption
- The One Time Pad (OTP) [Proof]
- Perfect Secrecy
- Shannon's Theorem [Proof]

Pseudorandom Generators (PRG)
- Definition
- Security
- Secure Encryption From PRG
- Semantic Security [Proof]

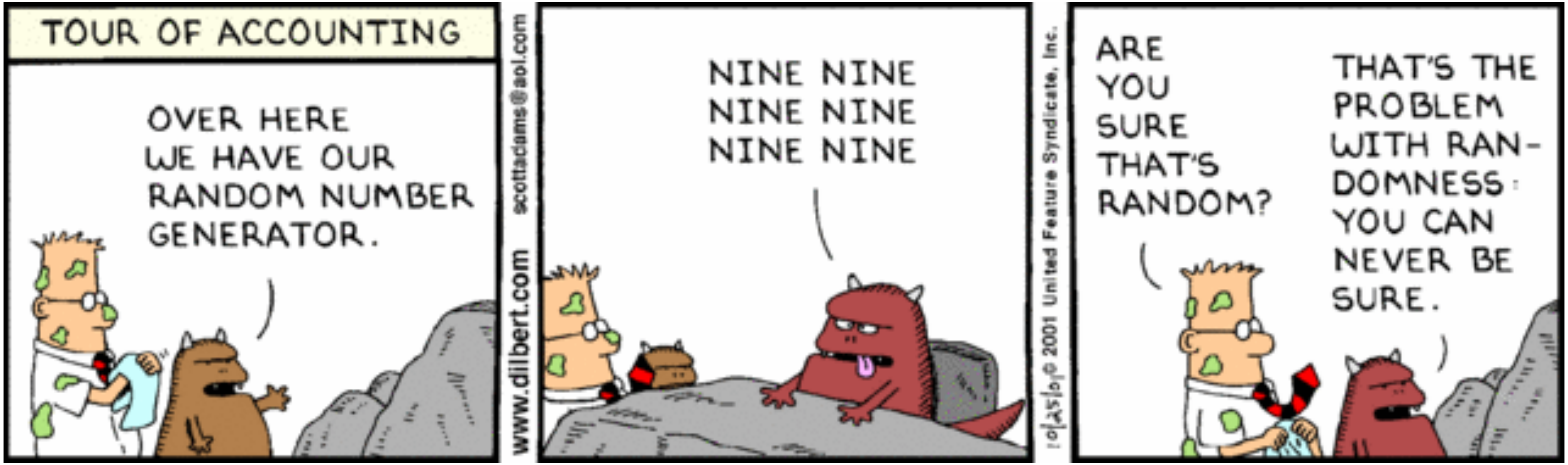
Pseudo Random Generators (PRG)

Definition: PRG

A Pseudo Random Generator is a *deterministic,* efficiently computable function PRG : $\{0,1\}^S \rightarrow \{0,1\}^L$ that on input a seed s of S bits, outputs a sequence of $L > S$. *Moreover, for* $s \leftarrow \$\{0,1\}^S$ *no efficient adversary can tell apart* PRG(s) *from a random string* $l \leftarrow \$\{0,1\}^L$*.*

The best way to check if a candidate algorithm is a PRG is by running a series of tests, there is no mathematical proof! But we can reason about the *security* of a PRG using a formal (mathematical) security game.

Pseudo Random Generators (PRG)



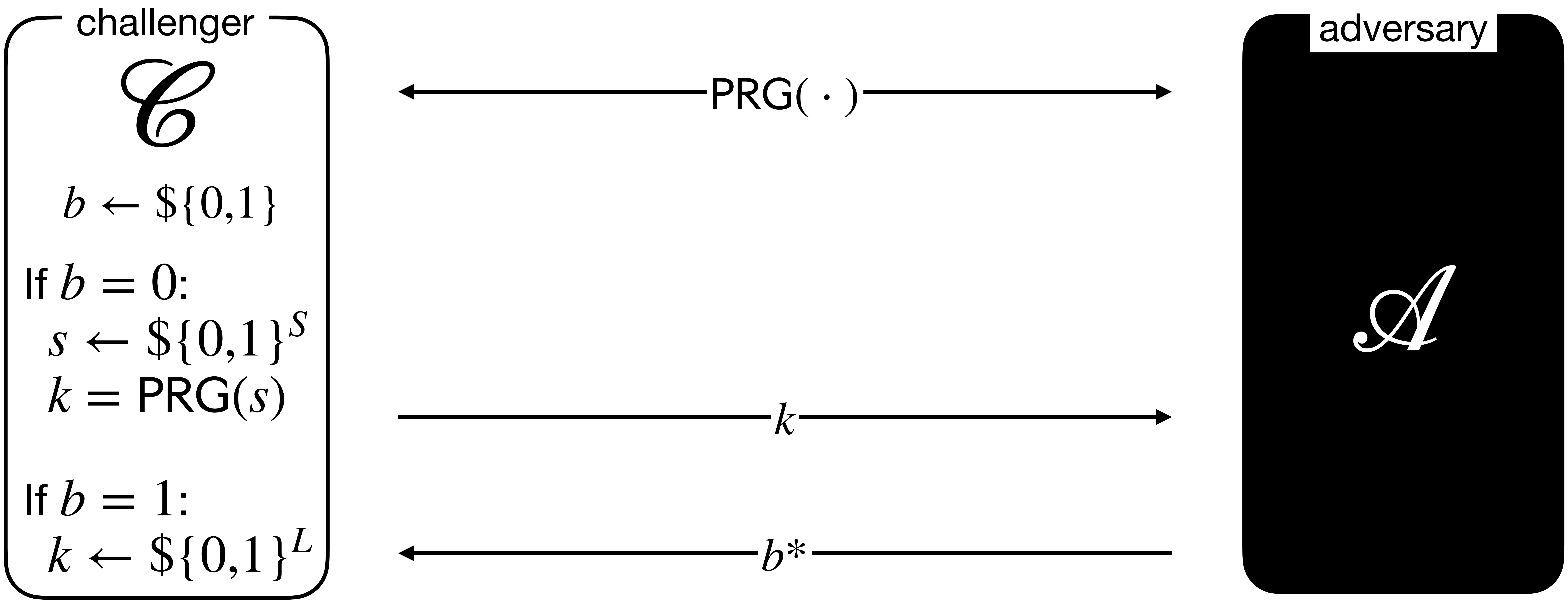
The best way to check if a candidate algorithm is a PRG is by running a series of tests, there is no mathematical proof! **But we can reason about the *security* of a PRG using a formal (mathematical) security game.**

“Real OR Random” Security (Intuition)

https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.youtube.com%2Fwatch%3Fv%3DzGLx_mLLu0c&psig=AOvVaw2apYsTgLReXypLHAZ3jP-3&ust=1650024734701000&source=images&cd=vfe&ved=2ahUKEwixrN2iw5P3AhXYZfEDHcXCDJcQjhx6BAgAEAw

Security Game For PRG

Aim: quantify the attacker's likelihood in distinguishing PRG from a source of uniform randomness over $\{0,1\}^L$



win or **lose**

$\mathscr{A}$ wins the security game if $b^* = b$.
If $b^* \neq b$, $\mathscr{A}$ loses the game.

Security Game For PRG

Definition: Secure PRG

A pseudo random function PRG : $\{0,1\}^S \rightarrow \{0,1\}^L$ is a secure PRG if any PPT attacker $\mathscr{A}$ has only negligible advantage in winning the secure PRG game. Formally,

$$Adv(\mathscr{A}) = |Pr[\mathscr{A}\ wins] - \frac{1}{2}| < negl(S)$$

Verbose description of the PRG security game

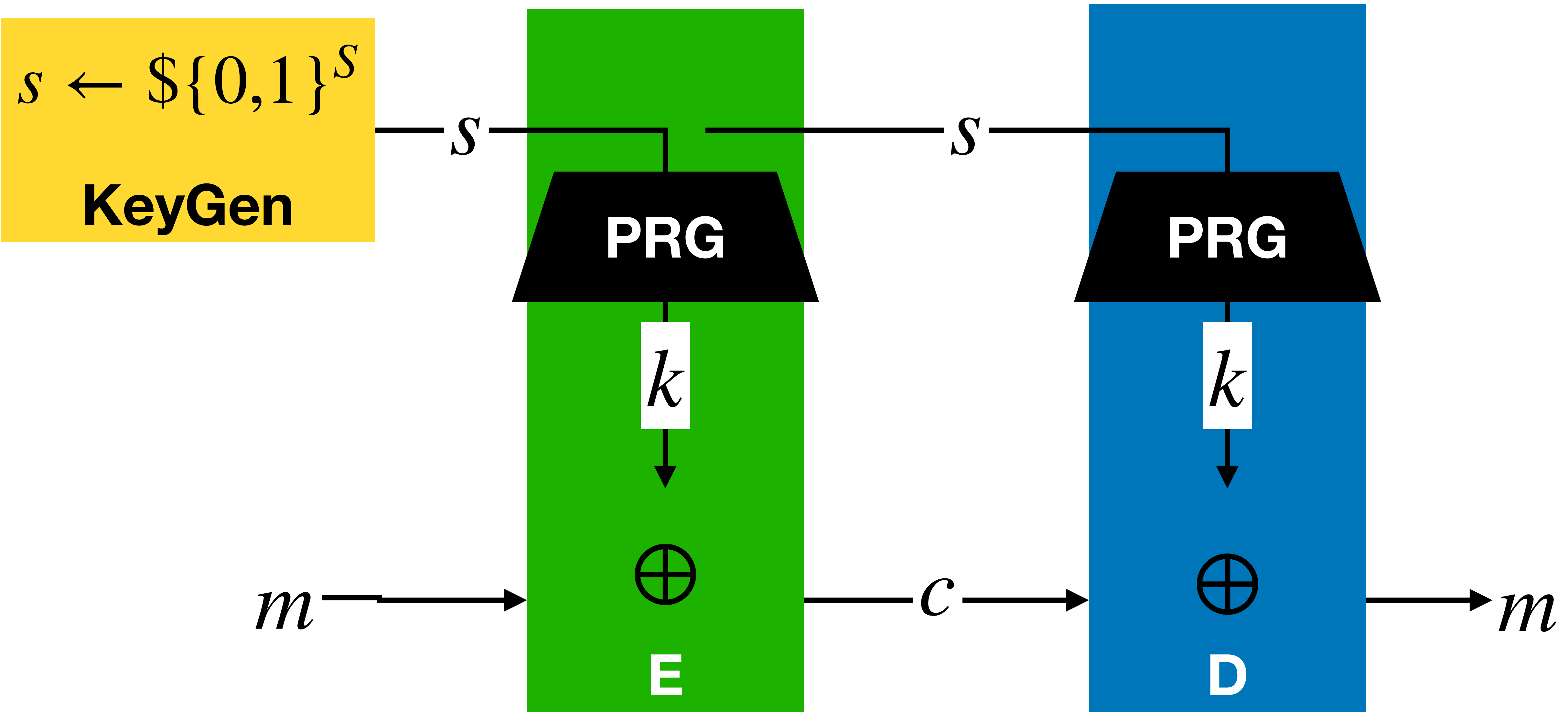
1. The challenger $\mathscr{C}$ draws a uniformly random bit $b \leftarrow \$\{0,1\}$.
2. If $b = 0$, the challenger draws a random seed $s \leftarrow \$\{0,1\}^S$ and computes $k =$ PRG(s).
   If $b = 1$, the challenger draws a uniformly random string $k \leftarrow \$\{0,1\}^L$.
3. $\mathscr{C}$ sends k to $\mathscr{A}$.
4. $\mathscr{A}$ tries to determine b from k, and eventually (within polynomial time) returns its guess b^*.
5. $\mathscr{A}$ sends b^* to the $\mathscr{C}$. The adversary wins if $b^* = b$.

Construct a Secure Encryption Scheme From a PRG

A generic PRG

$\text{PRG} : \{0,1\}^S \rightarrow \{0,1\}^L$

$\text{PRG}(s) = k$



A One-time PRG cipher

$\mathcal{M} = C = \{0,1\}^L$, $\mathcal{K} = \{0,1\}^S$, $S < L$

$KeyGen(1^S) \rightarrow s$ $(where\ s \leftarrow \$\{0,1\}^S)$

$Enc(s, m) = \text{PRG}(s) \oplus m$

$Dec(s, c) = \text{PRG}(s) \oplus c$

🧐 *Does this cipher have perfect security?*

We need a new security definition that works when $|\mathcal{K}| < |\mathcal{M}|$

“Left OR Right” Security (Intuition)

Semantic Security

Aim: quantify the attacker's likelihood in distinguishing an encryption of a (chosen) message m_0 from an encryption of another (chosen) message m_1

challenger

$\mathscr{C}$

$k \leftarrow KeyGen(1^n)$

$b \leftarrow \$\{0,1\}$

If $b = 0$:
  $c \leftarrow E(k, m_0)$
If $b = 1$:
  $c \leftarrow E(k, m_1)$

$\longleftarrow (KeyGen,\ E,\ D) \longrightarrow$

$\longleftarrow (m_0, m_1)$
$s.t.\ |m_0| = |m_1|$

$\xrightarrow{\quad c \quad}$

$\xleftarrow{\quad b^* \quad}$

adversary

$\mathscr{A}$

`win or lose`

$\mathscr{A}$ wins the security game if $b^* = b$.
If $b^* \neq b$, $\mathscr{A}$ loses the game

Semantic Security for Symmetric Encryption

Definition: Semantic security

A symmetric encryption scheme is semantically secure if any PPT attacker $\mathscr{A}$ has only negligible advantage in winning the semantic security game. Formally,

$$Adv(\mathscr{A}) = |Pr[\mathscr{A}\ wins] - \frac{1}{2}| < negl(n)$$

Verbose description of the semantic security game

1. The challenger $\mathscr{C}$ generates a key $k \leftarrow KeyGen(1^n)$ and draws a random bit $b \leftarrow \$\{0,1\}$.
2. The adversary $\mathscr{A}$ chooses two messages m_0, m_1 of the same length and sends them to $\mathscr{C}$.
3. $\mathscr{C}$ encrypts m_b according to the bit drawn in step 1, and returns $c = Enc(k, m_b)$ to $\mathscr{A}$.
4. $\mathscr{A}$ tries to determine b from $c, m_0,$ and m_1.
5. $\mathscr{A}$ sends b^* to the $\mathscr{C}$. The adversary wins if $b^* = b$.

Remarks on the Definition

Definition: Semantic security

A symmetric encryption scheme is semantically secure if any PPT attacker $\mathscr{A}$ has only negligible advantage in winning the semantic security game. Formally,

$$Adv(\mathscr{A}) = |Pr[\mathscr{A}\ wins] - \frac{1}{2}| < negl(n)$$

- We don't expect the encryption scheme to hide the length of the plaintext; (hence m_0 and m_1 must have the same length).
- An attacker who just guesses, choosing a random $b^* \leftarrow \$\{0,1\}$, has advantage 0.
- An attacker who always answers $b^* = 1$ (or $b^* = 0$) also has advantage 0.
- If the encryption scheme is the one time pad, any attacker has advantage 0.

Proving our Construction Is Semantically Secure

If PRG : $\{0,1\}^S \rightarrow \{0,1\}^N$ is a secure PRG, then the cipher defined by $Enc(s, m) = \text{PRG}(s) \oplus m$; $\quad Dec(s, c) = \text{PRG}(s) \oplus c$ is semantically secure.

Formally, for any efficient $\mathscr{A}$: $Adv_{sem.sec}(\mathscr{A}) = |\,Pr[\mathscr{A}\ wins] - \frac{1}{2}\,| < negl(S)$

Proof Plan:

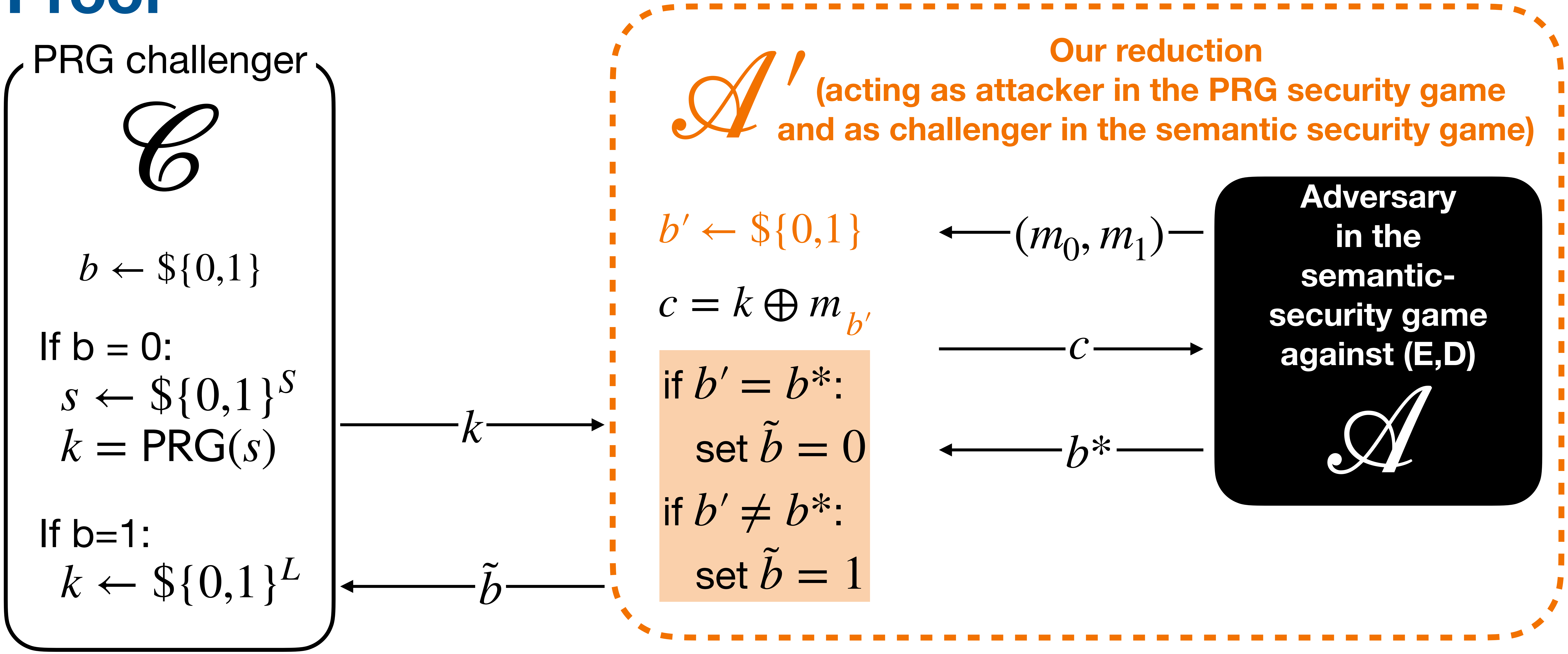
We must prove that any efficient adversary against the encryption's semantical security has negligible advantage, without knowing anything about the adversary's strategy.

🧐 *HOW ?*

…or… ***proof by reduction to absurd***

Assume that there exists an adversary $\mathscr{A}$ that can break the semantic security of the encryption. Then we build a new adversary $\mathscr{A}'$ that uses $\mathscr{A}$ to break the security of the PRG. Since PRG is assumed to be secure, such $\mathscr{A}'$ cannot exist. Thus it was absurd to assume $\mathscr{A}$ exists in the first place.

The Proof



Important observations

If $b = 0$, the ciphertext c is the encryption using the PRG cipher. Because we assumed that $\mathscr{A}$ wins this game with non negligible probability this means $b' = b^*$. So $\mathscr{A}'$ wins when $\mathscr{A}$ does. If $b = 1$, $\mathscr{A}'$ encryption is the **OTP** (perfectly secure), thus $\mathscr{A}$ has no advantage. So $\mathscr{A}'$ only guesses correctly with 1/2 probability (0 advantage).

The Proof

complementary events

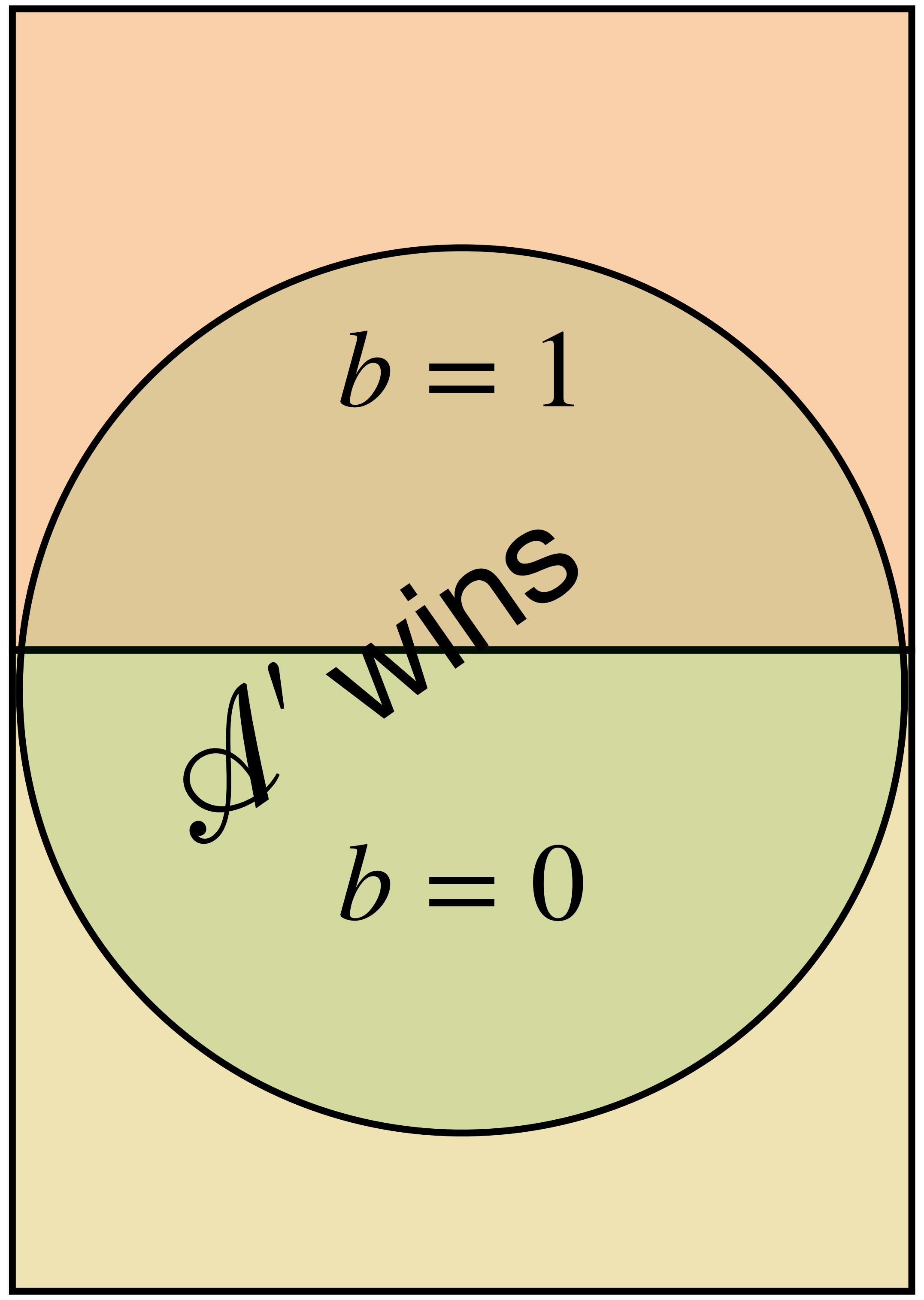
Pr[$\mathscr{A}'$ wins] = Pr[$\mathscr{A}'$ wins AND $b = 0$] + Pr[$\mathscr{A}'$ wins AND $b = 1$]

conditional probability $Pr[A \mid B] = \dfrac{Pr[A \cap B]}{Pr[B]}$

= Pr[$\mathscr{A}'$ wins | $b = 0$] Pr[b=0] + Pr[$\mathscr{A}'$ wins | $b = 1$] Pr[b=1]

PRG-cipher **1/2** *OTP-cipher (perfect security)* **1/2**

=
Pr[$\mathscr{A}$ wins sem.sec game against (E,D)]



Thus Pr[$\mathscr{A}'$ wins PRG] = Pr[$\mathscr{A}$ wins sem.sec] · (1/2) + 1/2 · (1/2)

Or, reorganising the terms: Pr[$\mathscr{A}$ wins sem.sec] = 2 Pr[$\mathscr{A}'$ wins PRG] - 1/2

$$Adv_{sem.sec}(\mathscr{A}) = \left| Pr[\mathscr{A} \ wins] - \frac{1}{2} \right| \quad = \left| (2Pr[\mathscr{A}' \ wins \ PRG] - 1/2) - \frac{1}{2} \right| \quad = 2 \cdot Adv_{PRG}(\mathscr{A}')$$

The Proof

$$Adv_{sem.sec}(\mathscr{A}) = |Pr[\mathscr{A}\ wins] - \frac{1}{2}| \ = |(2Pr[\mathscr{A}'\ wins\ PRG] - 1/2) - \frac{1}{2}| \ = 2 \cdot Adv_{PRG}(\mathscr{A}')$$

This concludes the proof of the theorem ⬆️

🧐 *WHY ?*

If our PRG-based encryption is not secure then $\mathscr{A}$ has a non-negligible advantage in winning the semantic security game. If that was the case, we have constructed an efficient (PPT) reduction/adversary $\mathscr{A}'$ that uses $\mathscr{A}$ to win the PRG security game and has twice the advantage of $\mathscr{A}$. Since we assumed the PRG to be secure, it is impossible for any efficient adversary to break the PRG. So such an $\mathscr{A}'$ cannot exist. Which in turn implies that $\mathscr{A}$ cannot exist. So it was absurd to assume such an $\mathscr{A}$ exists. This reasoning implies that our PRG-based encryption is *provably* secure.