CRYPTOGRAPHY

(Lecture 3)

Literature:

“**Handbook of Applied Cryptography**” **(ch 7.2.1, 7.4.1,** 7.4.2)

“Lecture Notes on Cryptography” by S. Goldwasser and M. Bellare (**ch 4.1, 4.5, 6.4**)

“A Graduate Course in Applied Cryptography” by D. Boneh and V. Shoup (ch 4.1.0,4.1.1, 4.1.4, 4.2.4, **4.3.4**, 4.5, 4.6)

Module 1: Agenda

Introduction

One-Way Functions / Hash Functions

Commitment Schemes

Blockchain Technology

OTP & Perfect Secrecy

PRG

Semantic Security + Proof

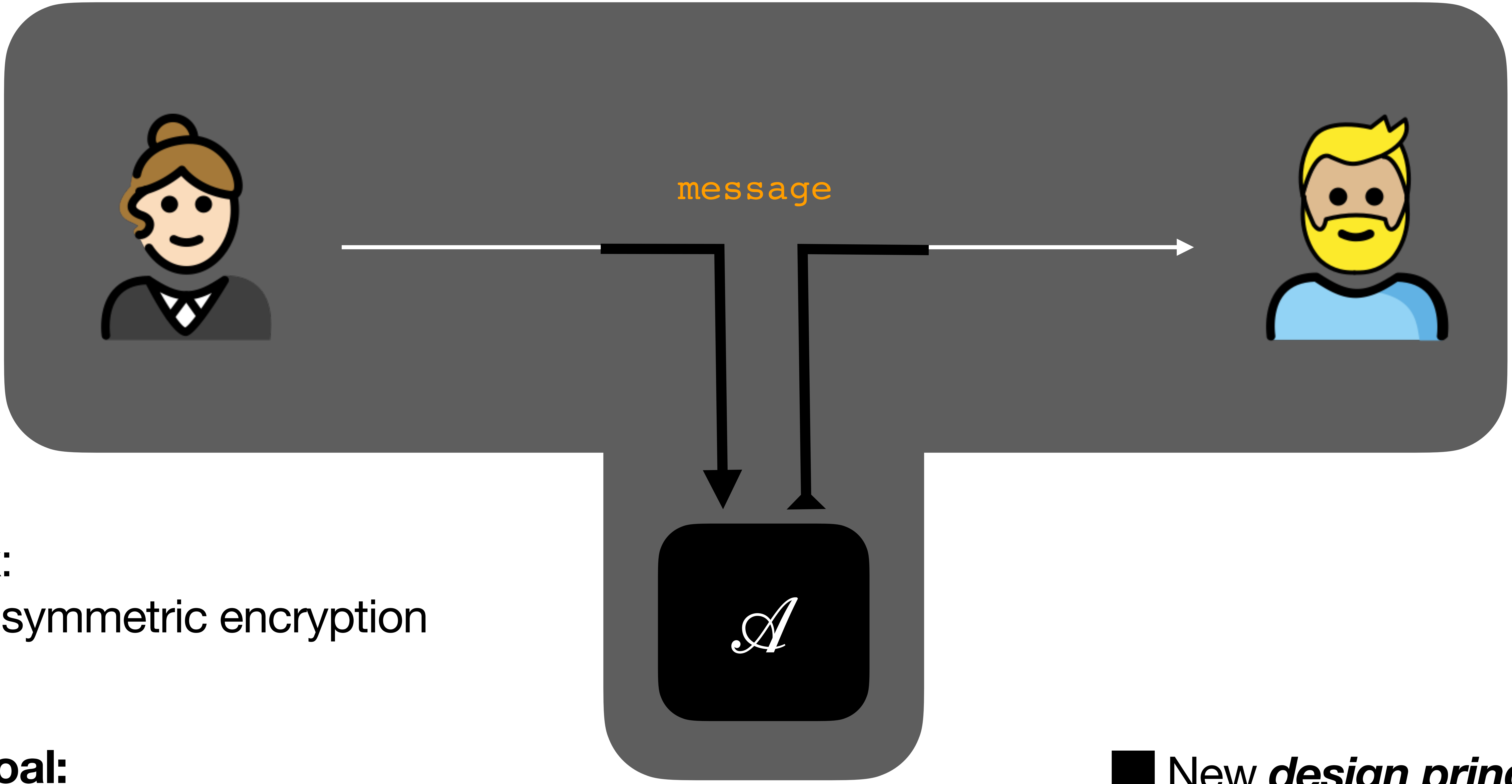
Block Ciphers

- Definition
- The Advanced Encryption Standard (AES)
- Design Principles

Modes of Operations

- ECB, CBC, CTR
- Is AES-ECB Semantically Secure?
- New Security Notion: IND-CPA
- AES Security Against Quantum Adversaries

Secure Communication Over an Insecure Channel



Last week:
one-time symmetric encryption

Today's goal:
construct secure symmetric encryption that allows for key **reuse**

- New ***design principles***
- New ***security notions***

Today’s Goal: RECYCLING

Today’s main goal:
one secret key to
encrypt **many** messages

Last week:
one-time symmetric encryption
With shorter keys (from PRG)

Mechanisms to “evolve” one key

Block Ciphers

Definition: Block Cipher

A Block Cipher is a **deterministic**, **keyed** function that is **invertible**.

Formally, $E : \mathscr{K} \times \mathscr{M} \rightarrow \mathscr{M}$, where $\mathscr{K} = \{0,1\}^K$ is a key space, $\mathscr{M} = \{0,1\}^n$ is the block space and for every $k \in \mathscr{K}$, the function $E(k, \cdot) : \{0,1\}^n \rightarrow \{0,1\}^n$ is invertible, that is to say, there exist an efficient function $D(k, \cdot) : \{0,1\}^n \rightarrow \{0,1\}^n$ such that $D(k, E(k, m)) = m$ for every $m \in \{0,1\}^n$.

Plaintexts and ciphertexts are both called blocks.

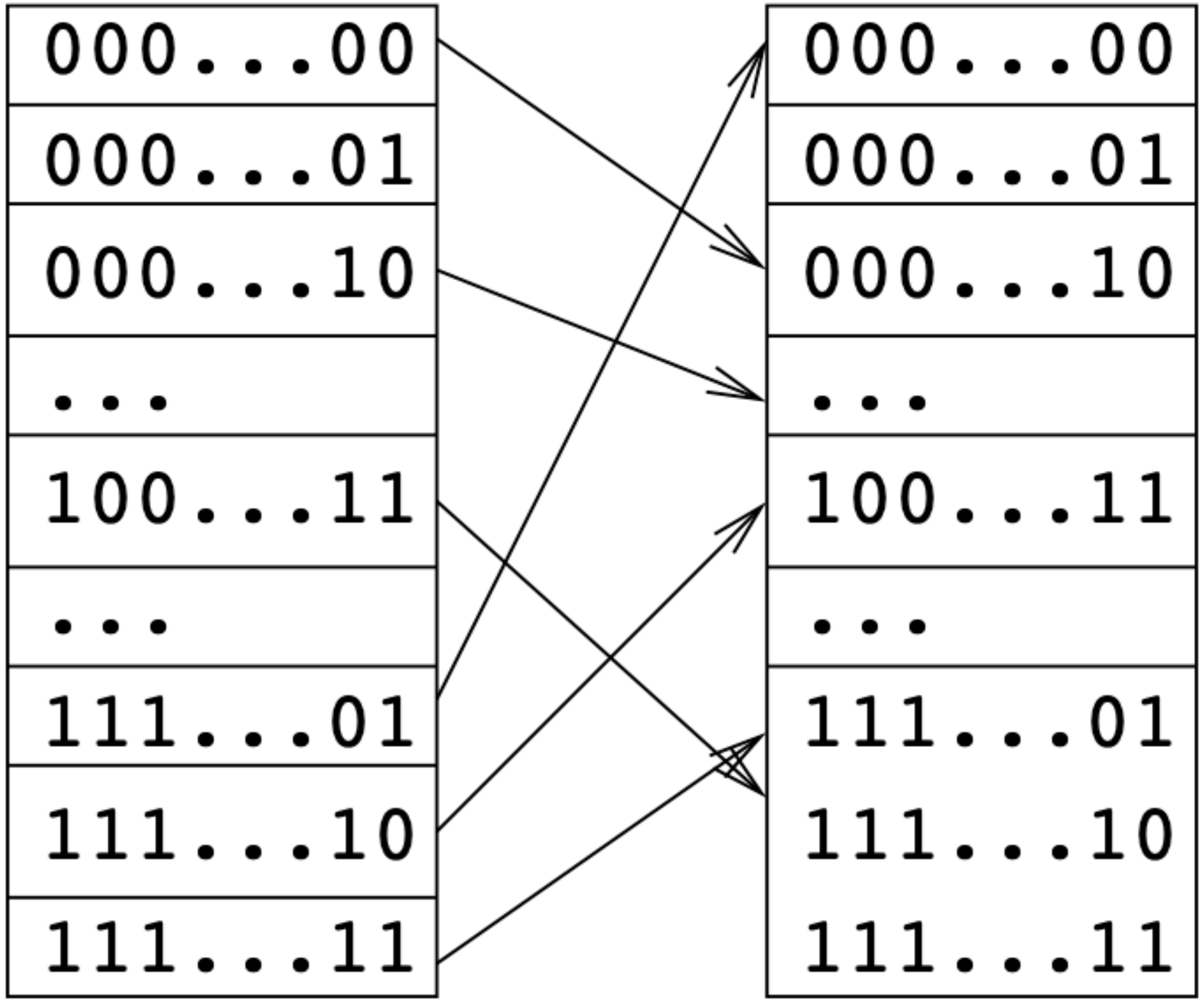
🧐 *What if the plaintext message is longer than one block (n-bits) ?*

[The Block cipher “chains” blocks according to a “mode of operation” (more on this later)]

Observation: we can now reuse the same key to encrypt/decrypt multiple messages!

Block Ciphers - Examples

For each key $k \in \mathscr{K} = \{0,1\}^K$, a block cipher is a **permutation** of bit-strings of length *n*, i.e., a bijective function from $\mathscr{M} = \{0,1\}^n$ to $\mathscr{M}$.

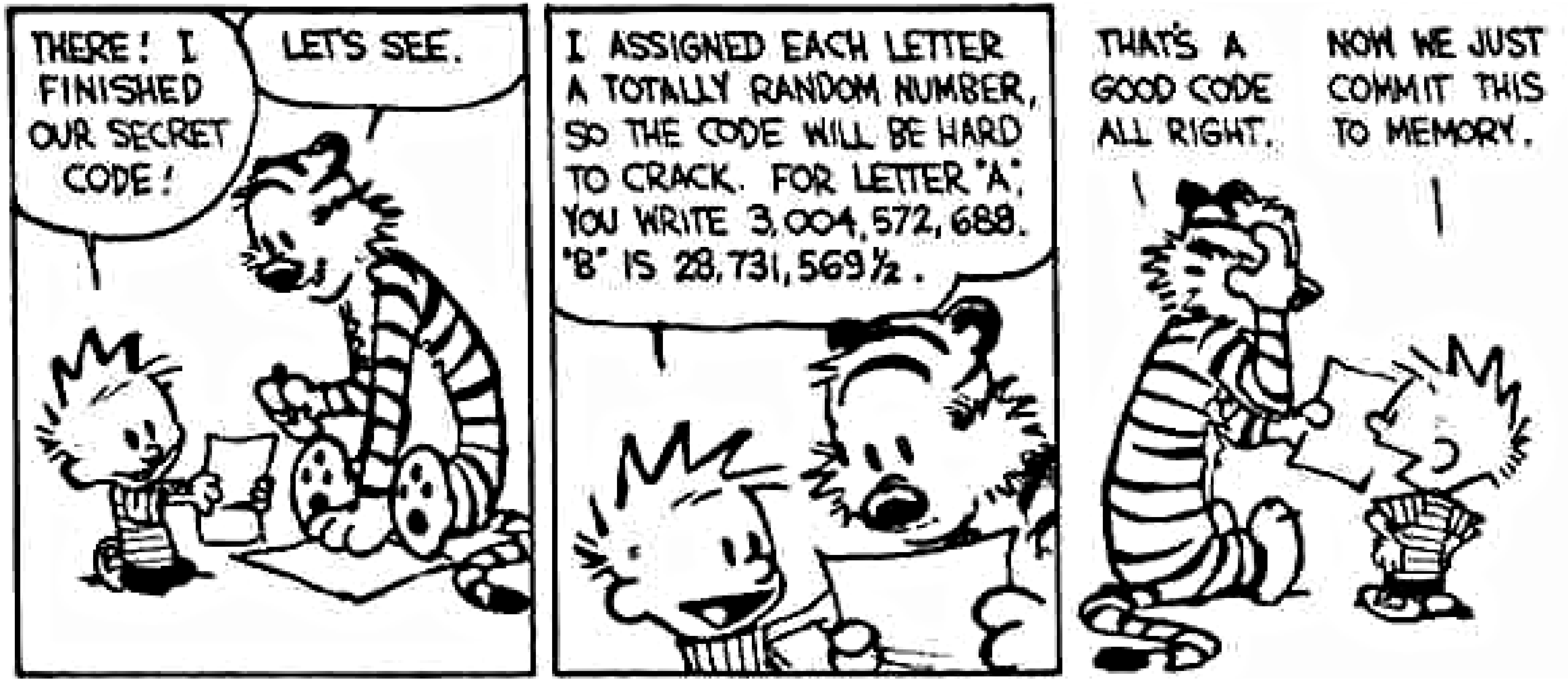


A **random permutation** is a permutation chosen uniformly at random from the set Perms($\mathscr{M}$) of all permutations on $\mathscr{M}$.

🧐 *How many possible permutations over $\mathscr{M} = \{0,1\}^n$ are there? |Perms($\mathscr{M}$)|?*

[|Perms($\mathscr{M}$)| = $2^n! \approx 2^{n2^n}$ vs |block ciphers| = |$\mathscr{K}$| = 2^K]

Security vs Efficiency - Finding the Right Balance



We cannot (efficiently) implement (all possible) random permutations for reasonable sizes of *n*.
Instead, we strive to construct ciphers that **cannot be distinguished from random permutations**.

The Block Cipher *Par Excellence:* AES

Advanced Encryption Standard

```
function AES_K(M)
(K_0,...,K_10) ← expand(K)
s ← M⊕K_0

for r = 1 to 10 do
   s ← S(s)
   s ← shift-rows(s)
   if (r≤9) then
      s ← mix-cols(s)
   fi
   s ← s ⊕ K_r
endfor
return s
```

*key of **reduced** size*

***reuse** material*

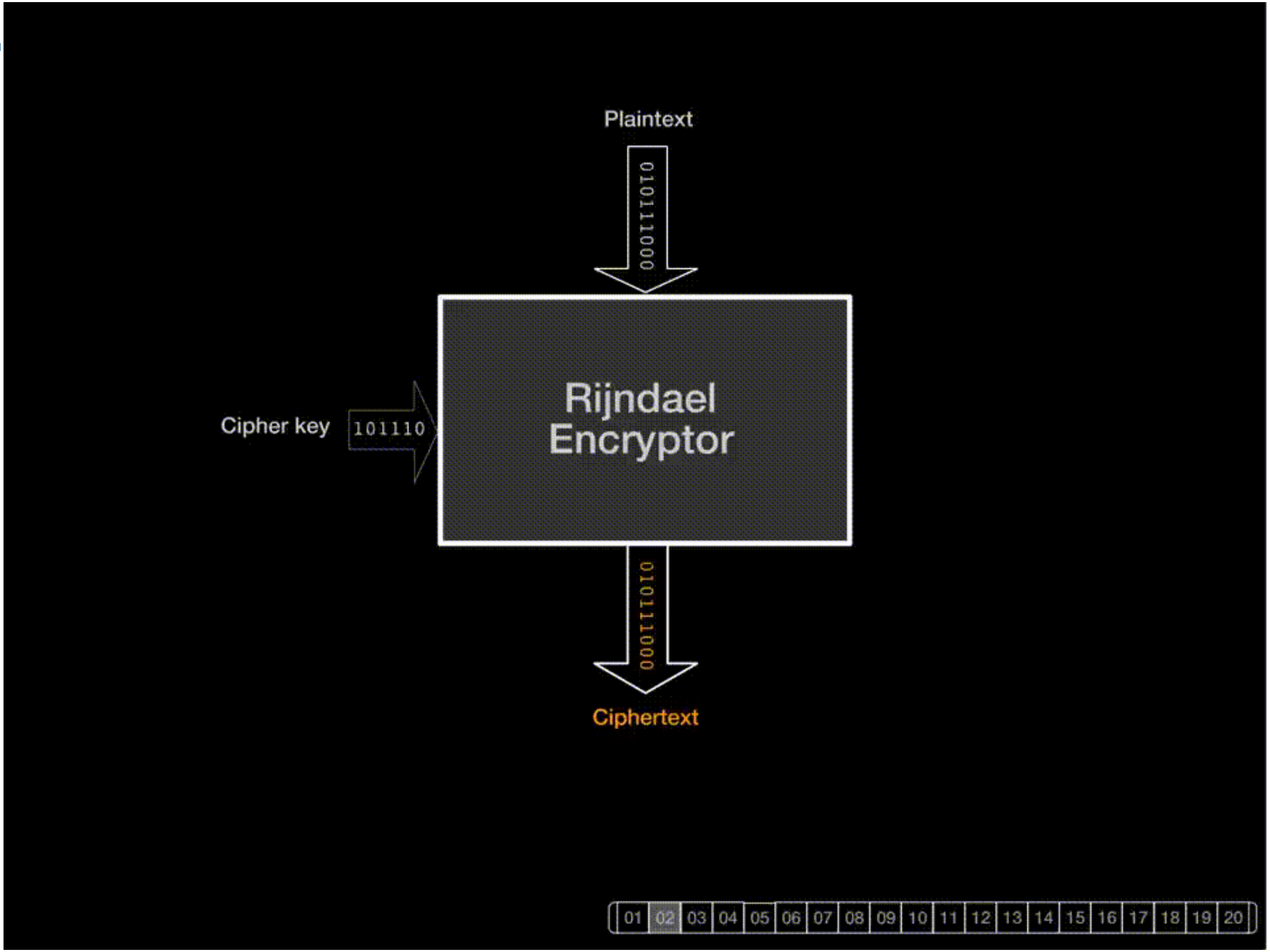
*OTP-style **recycle***

Link to a list of popular symmetric algorithms

AES Block Cipher (Rijndeal Design)

A bit of history:

- 1997 NIST opens a call for a new block cipher standard on 128-bit blocks
- 15 submissions
- 2 rounds of peer-review
- 5 finalists by 1999
- Intense cryptanalysis
- 2000 winner Rijndael
- AES official standard Nov. 2001 (FIPS197)



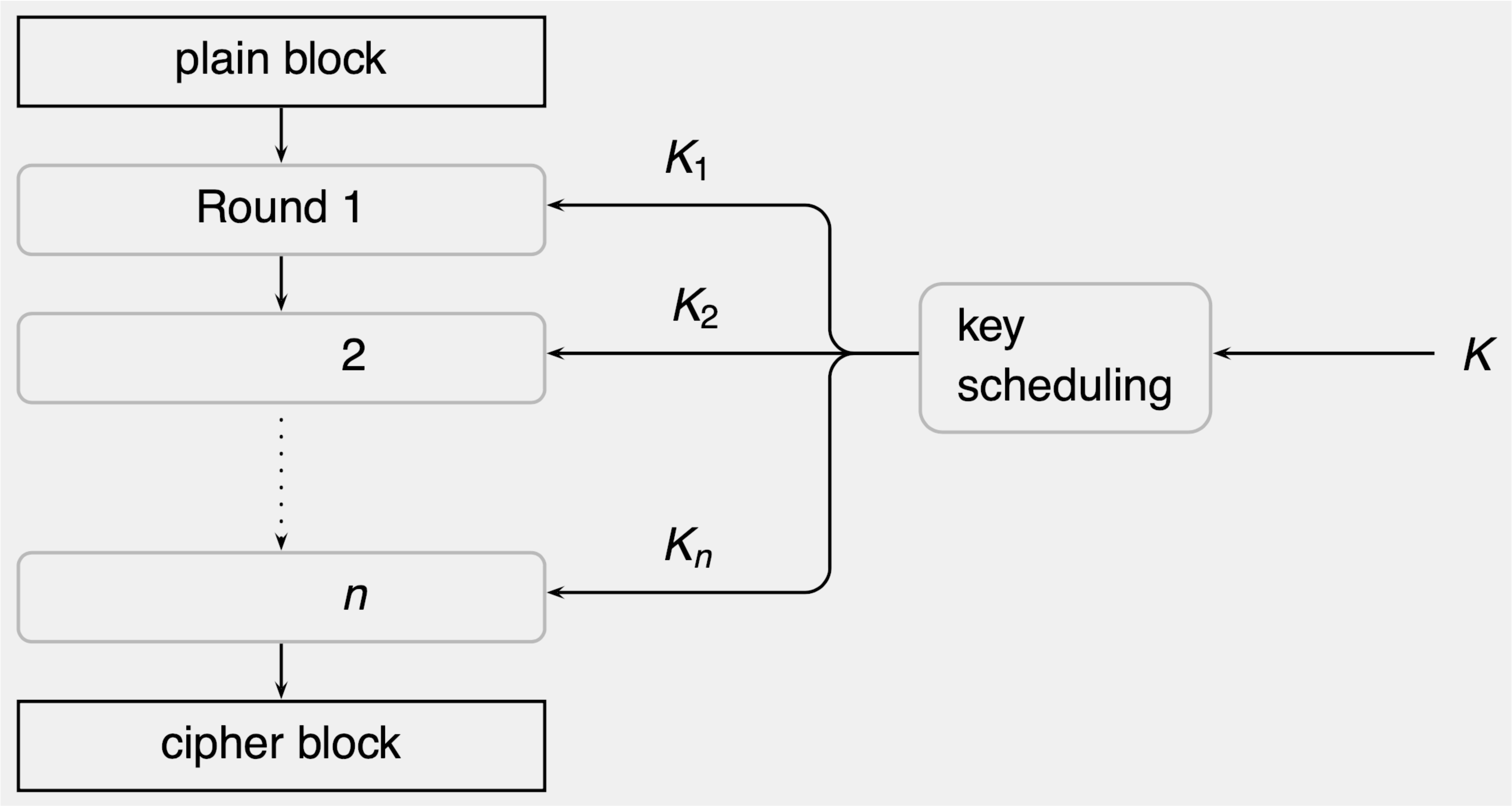
https://www.youtube.com/watch?v=gP4PqVGudtg

Why all of This?

If you cannot be a *truly* random permutation, at least strive to look like one.

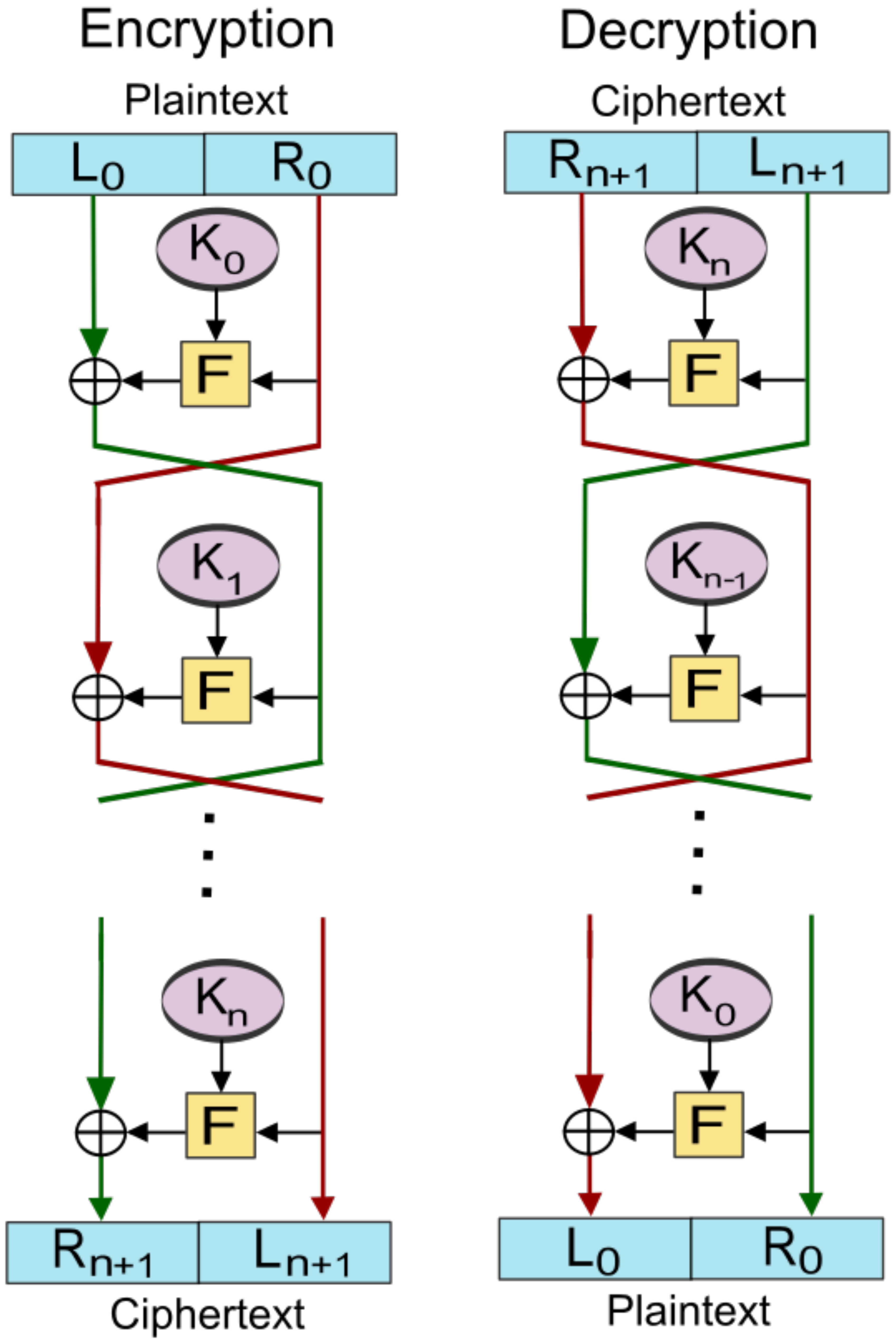
Design Principles for Block Ciphers: Iteration

<u>**Iteration:**</u> repeatedly apply a not-so-strong block cipher, each time with a different key (rounds)



This technique is not *provably secure*, but *heuristics* show that it works!

Design Principles for Block Ciphers: Feistel Networks



<u>Feistel Network:</u> The cipher function **F** is the same for every round.

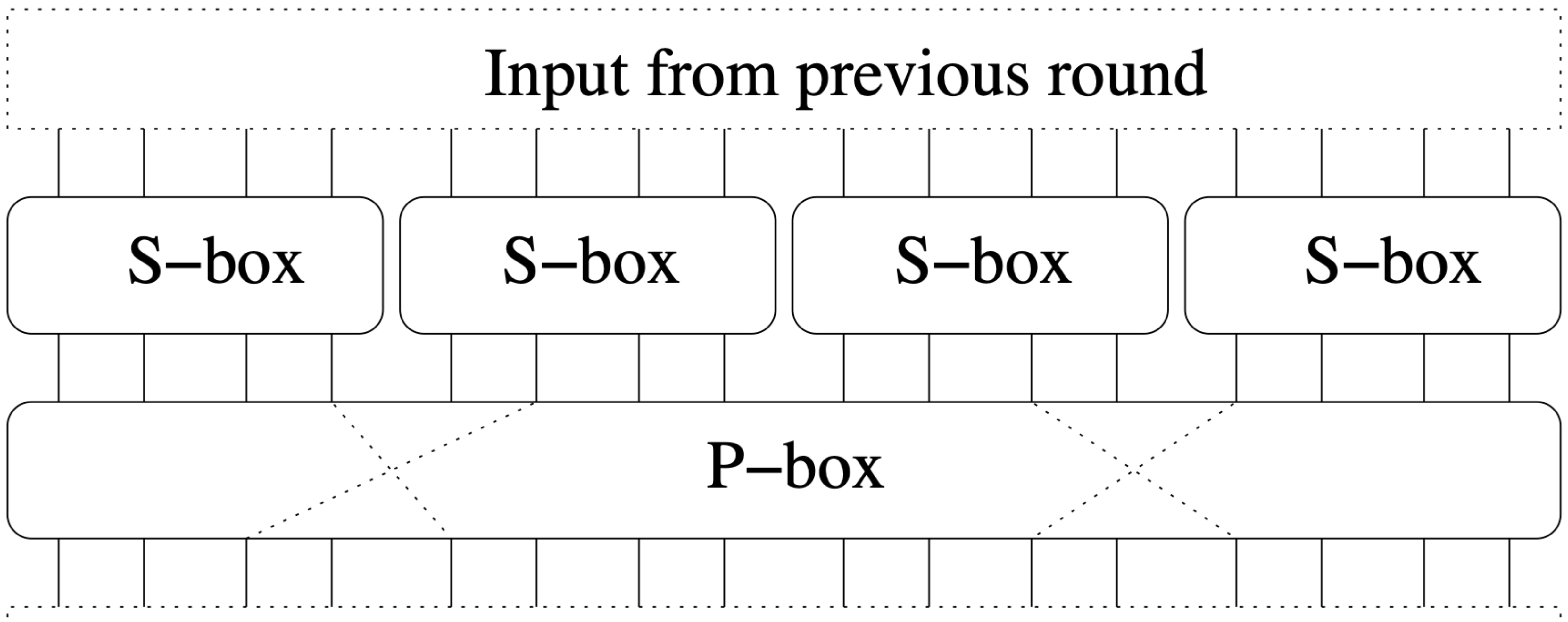
- **F** does not need to be invertible for the round to be invertible.
- **Decryption = Encryption** with round-keys in *reverse order*.

$$\begin{cases} L_{i+1} = R_i \\ R_{i+1} = L_i \oplus F(K_i, R_i) \end{cases}$$

Design Principles for Block Ciphers: Confusion & Diffusion

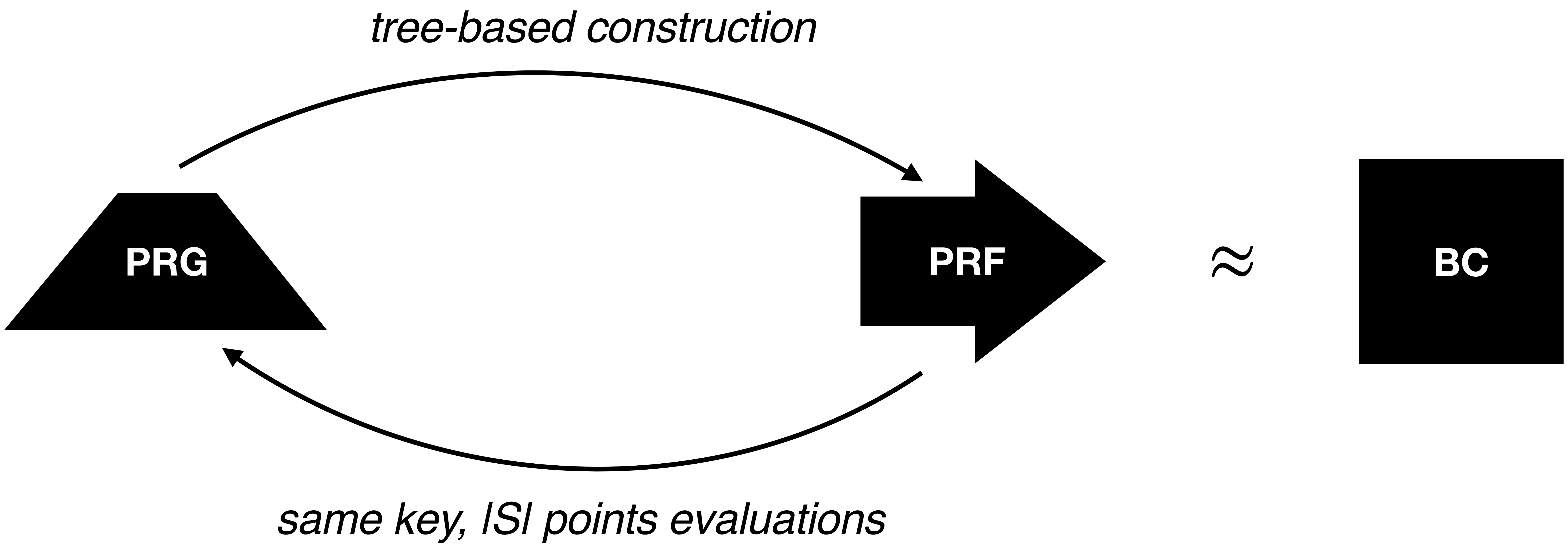
Confusion (S-boxes): Divide the *n* bits of input into *b* sub-blocks of *n*/*b* bits. Within each sub-block, apply a substitution table (**S-box**), i.e., a permutation on $\{0,1\}^{n/b}$, typically implemented as a lookup table. This introduces **confusion** to the cipher.

Diffusion (P-boxes): Confusion is local; to spread its effect apply a transposition **P-box**, permuting bits between sub-blocks, This introduces **diffusion**.



Side Note

I'm omitting a bunch of cool and fundamental results in the *theory* of cryptography connected to block ciphers…believe me, you do not want to be graded on this! But here's a glimpse



If you're interested, check out "A Graduate Course in Applied Cryptography" (ch 4.4, 4.5, 4.6,4.7)

Module 1: Agenda

Introduction

One-Way Functions / Hash Functions

Commitment Schemes

Blockchain Technology

OTP & Perfect Secrecy

PRG

Semantic Security + Proof

Block Ciphers

- Definition
- The Advanced Encryption Standard (AES)
- Design Principles

Modes of Operations

- ECB, CBC, CTR
- Is AES-ECB Semantically Secure?
- New Security Notion: IND-CPA
- AES Security Against Quantum Adversaries

Encrypting Long Messages (or Multiple Messages)

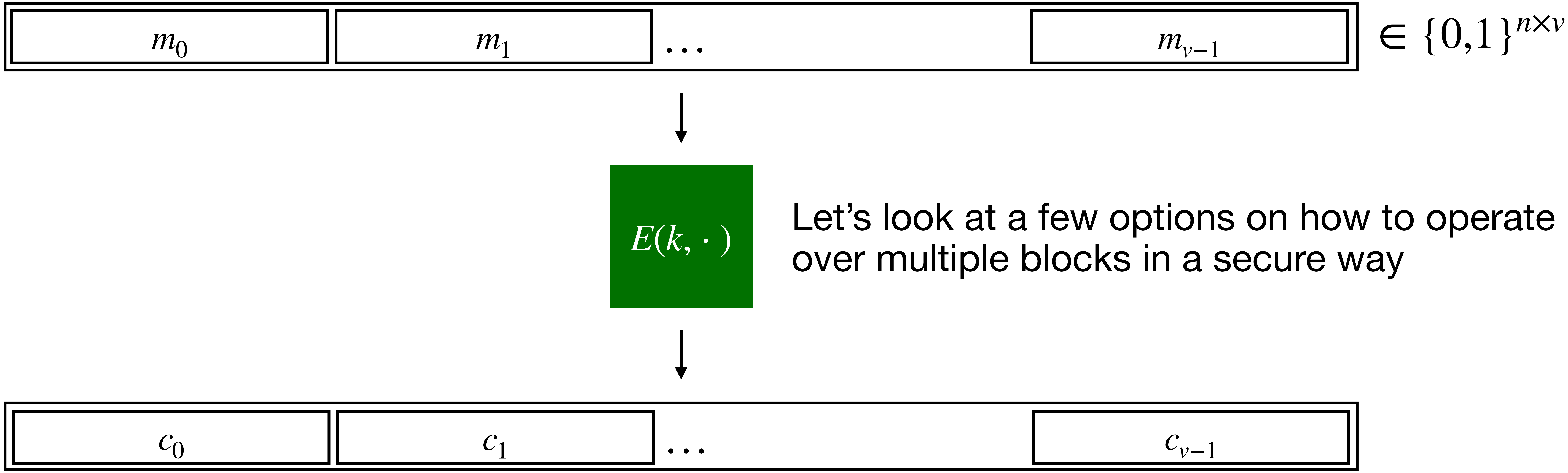
With the Same Key, *Securely*

A Block Cipher is a **deterministic**, **keyed** function that is **invertible**.

$$E : \mathscr{K} \times \{0,1\}^n \rightarrow \{0,1\}^n$$

or equivalently

$$E(k, \cdot) : \{0,1\}^n \rightarrow \{0,1\}^n$$

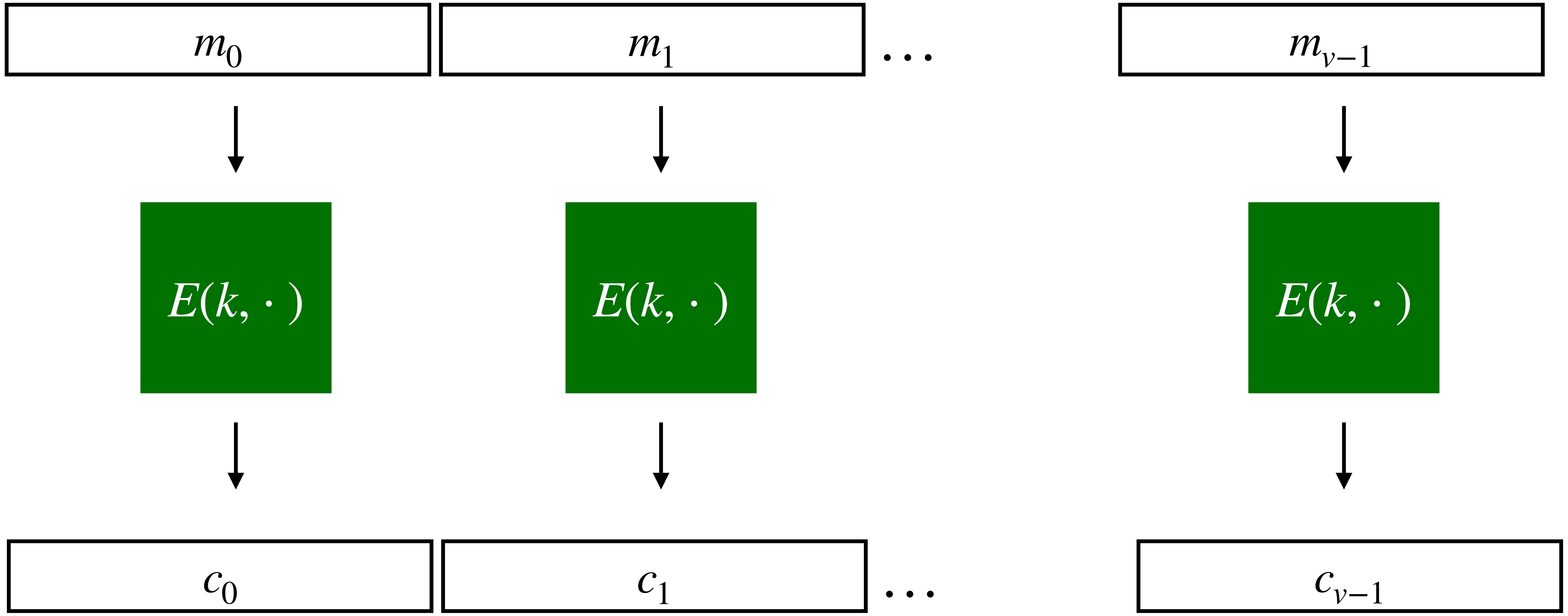


Let's look at a few options on how to operate over multiple blocks in a secure way

Electronic Code Book Mode (ECB)

+ very easy to understand and implement
+ encryption and decryption are parallelizable (important for large data)
- lacks diffusion (it encrypts identical plaintext blocks into identical ciphertext blocks)

ECB is not recommended for use in cryptographic protocols.



🧐 *How can we express this with a formula?*

$$c_i = E(k, m_i), \quad m_i = D(k, c_i), \quad for\ i = 0, \dots, v-1$$

Cipher Block Chaining Mode (CBC)

🧐 *formula?*

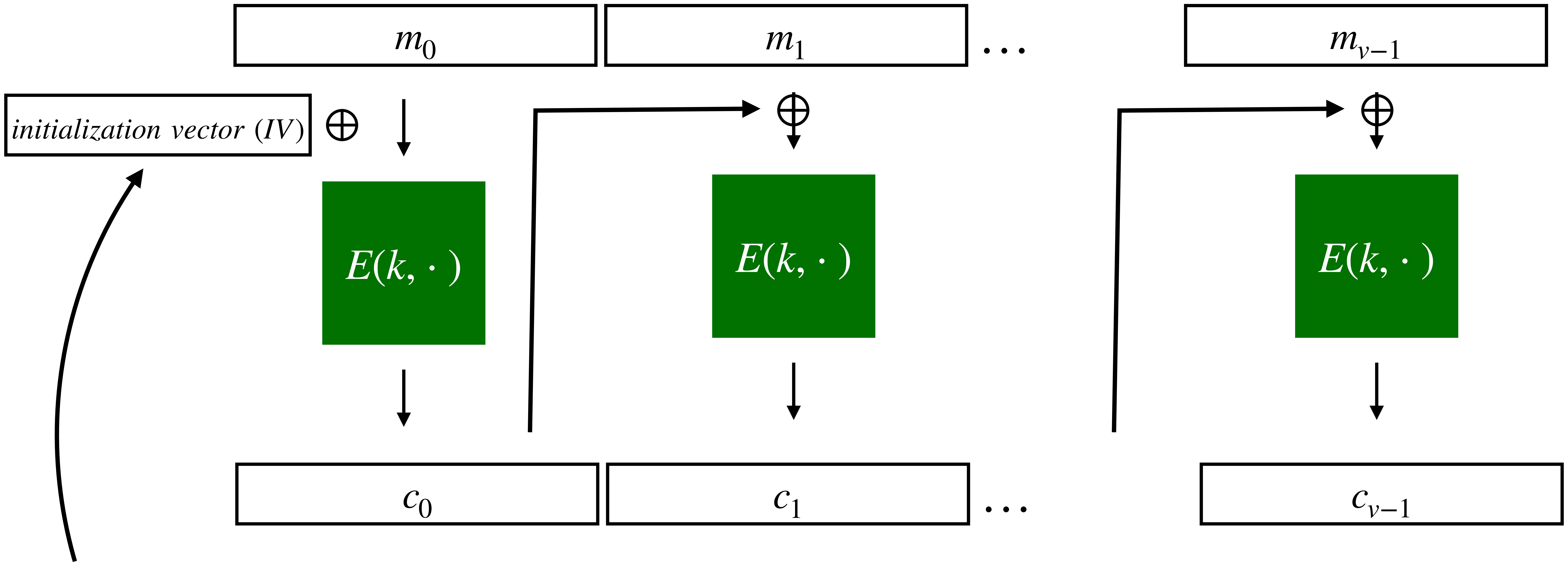
$$c_0 = E(k, m_0 \oplus IV)$$
$$c_i = E(k, m_i \oplus c_{i-1})\ for\ i > 0$$

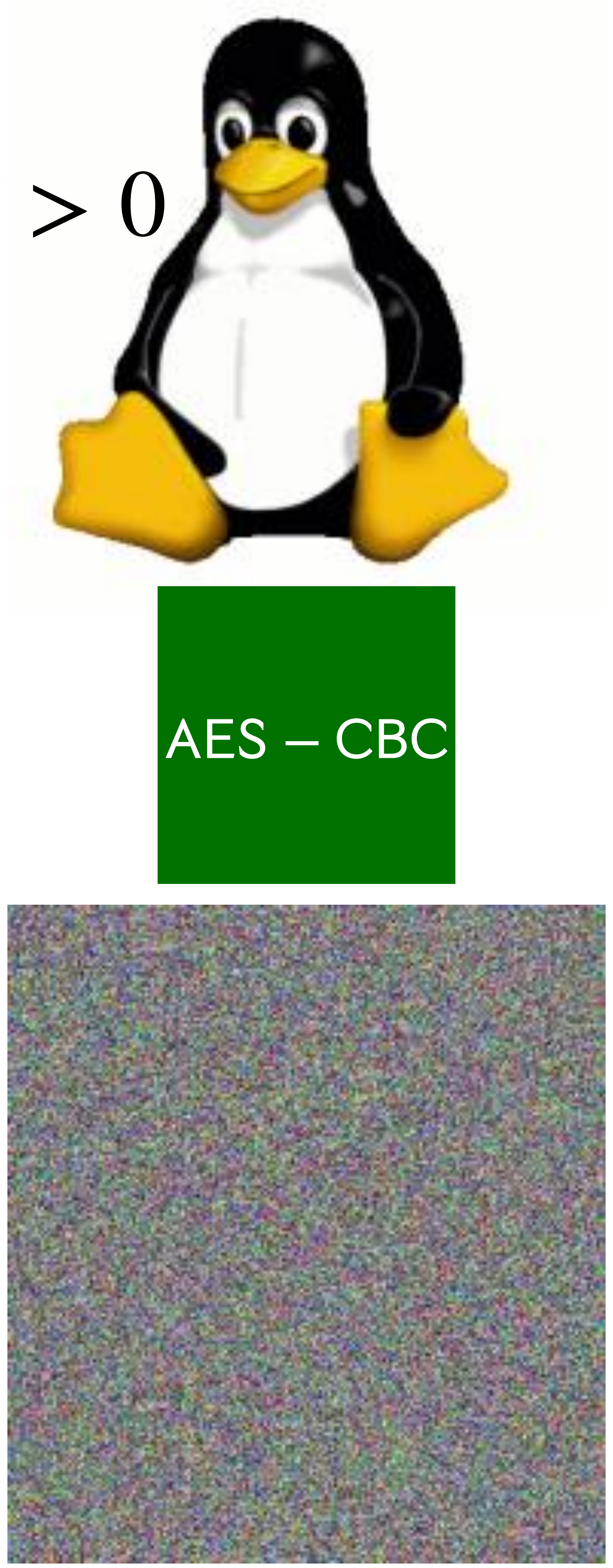
$$m_0 = D(k, c_0) \oplus IV$$
$$m_i = D(k, c_i) \oplus c_{i-1}\ for\ i > 0$$

\+ Decryption is parallelizable

\- Encryption is sequential (not parallelizable)

CBC is OK to use, but there's better.





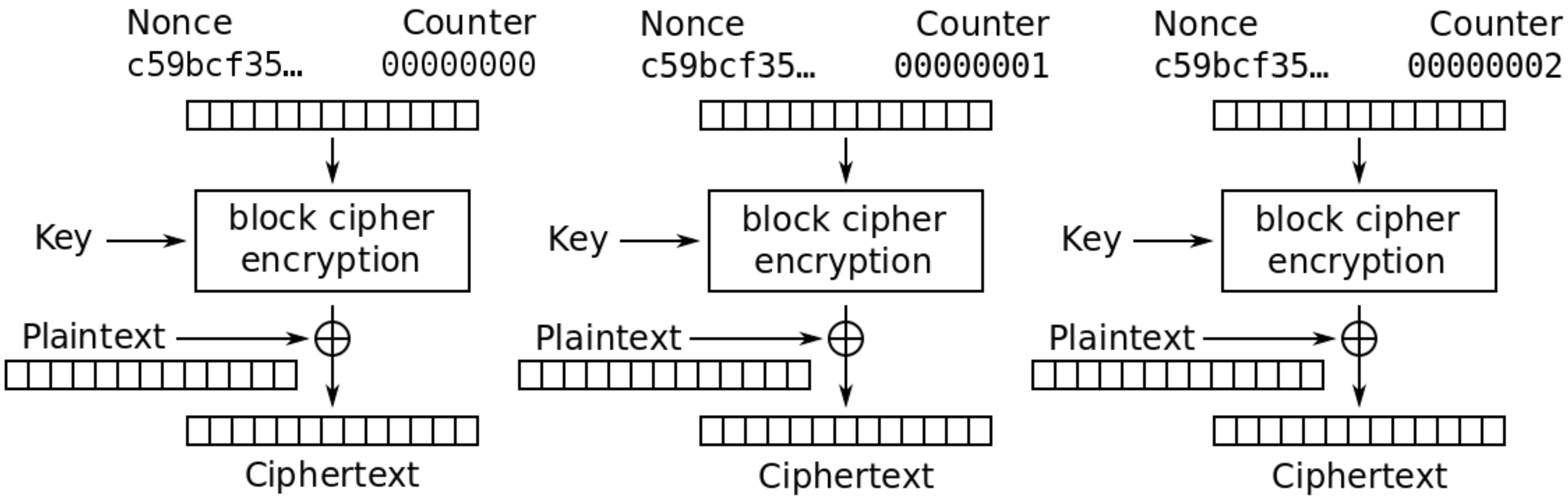
IV is the source of randomness

IV needs to travel with the ciphertext to enable the decryption of c_0

Counter Mode (CTR)

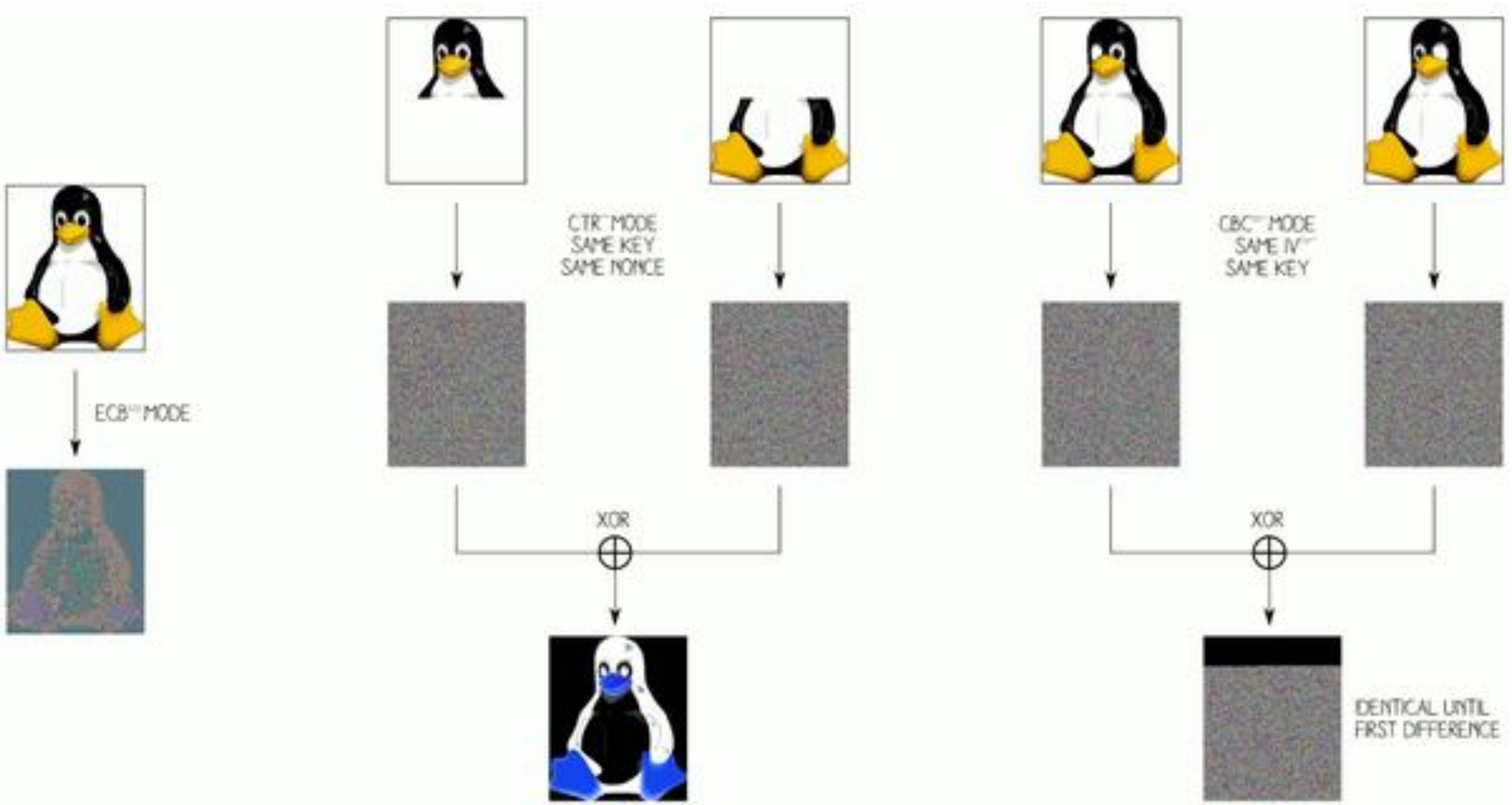
+ Encryption and Decryption are parallelizable (and basically E=D!)
- The nonce value is needed to decrypt every ciphertext block

CTR is OK to use, with care: nonce = number used **once…otherwise…**



Counter (CTR) mode encryption

Modes of Operation’s Failures - Visual Examples



Is AES-ECB Semantically Secure?

Is AES-ECB Semantically Secure?

challenger

$\mathscr{C}$

$k \leftarrow \$\{0,1\}^{128}$

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

If $b = 0$:

$c \leftarrow \text{AES} - \text{ECB}(k, m_0)$

If $b = 1$:

$c \leftarrow \text{AES} - \text{ECB}(k, m_1)$

(E, D)

(m_0, m_1)

$s.t. \ |m_0| = |m_1|$

c

$b*$

adversary

$\mathscr{A}$

Attack strategy:

choose $m_0 = (block0, block1)$
and $m_1 = (block1, block1)$

Note that $c = (c_0, c_1)$

If $c_0 = c_1$ return $b* = 1$

If $c_0 \neq c_1$ return $b* = 0$

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


Are other modes of operation “secure”?
Yes! and we need a new, stronger security notion to prove it

Lessons Learned So Far

From ECB: **deterministic encryption is not semantically secure**.
(if the same key is used to encrypt the same message, the resulting ciphertext will always be the same)

CBC and CTR use **randomness** (IV and nonce respectively), so two encryptions of the same plaintext under the same key will (generally) produce two different ciphertexts.

We want *probabilistic* encryption!

Bad news: probabilistic encryption generates ciphertexts **larger** than the plaintext
(this is an inevitable price to pay for *better* security)

Good news: in certain settings we can get *good* security **without ciphertext expansion**

Towards a New Security Notion

$\mathscr{A}$

Adversary's Goal

~~To decrypt a ciphertext~~ *Security can be damaged with much less*

~~To gain some information about the plaintext concealed in the ciphertext~~

Vague, we would need to quantify this leakage (possible but..)

To *distinguish* between the encryption of two *known* plaintext messages

In crypto jargon: **indistinguishability under chosen plaintext attack (IND-CPA)**

IND-CPA has many equivalent notions, read here if you're interested

Historical example: British military would place mines in particular locations hoping Germans would send encrypted messages about that location.

Modern example: Attacker-controlled Javascript on a web page causes victim web client to make a HTTPS connection.

Security Notions for Block Ciphers

$\mathscr{A}$

Adversary's Goal

To *distinguish* between the encryption of two *known* plaintext messages
In crypto jargon: **indistinguishability under chosen plaintext attack (IND-CPA)**

Adversary's Power

Efficient algorithm (probabilisitic, and runs in polynomial time $<2^{60}$)

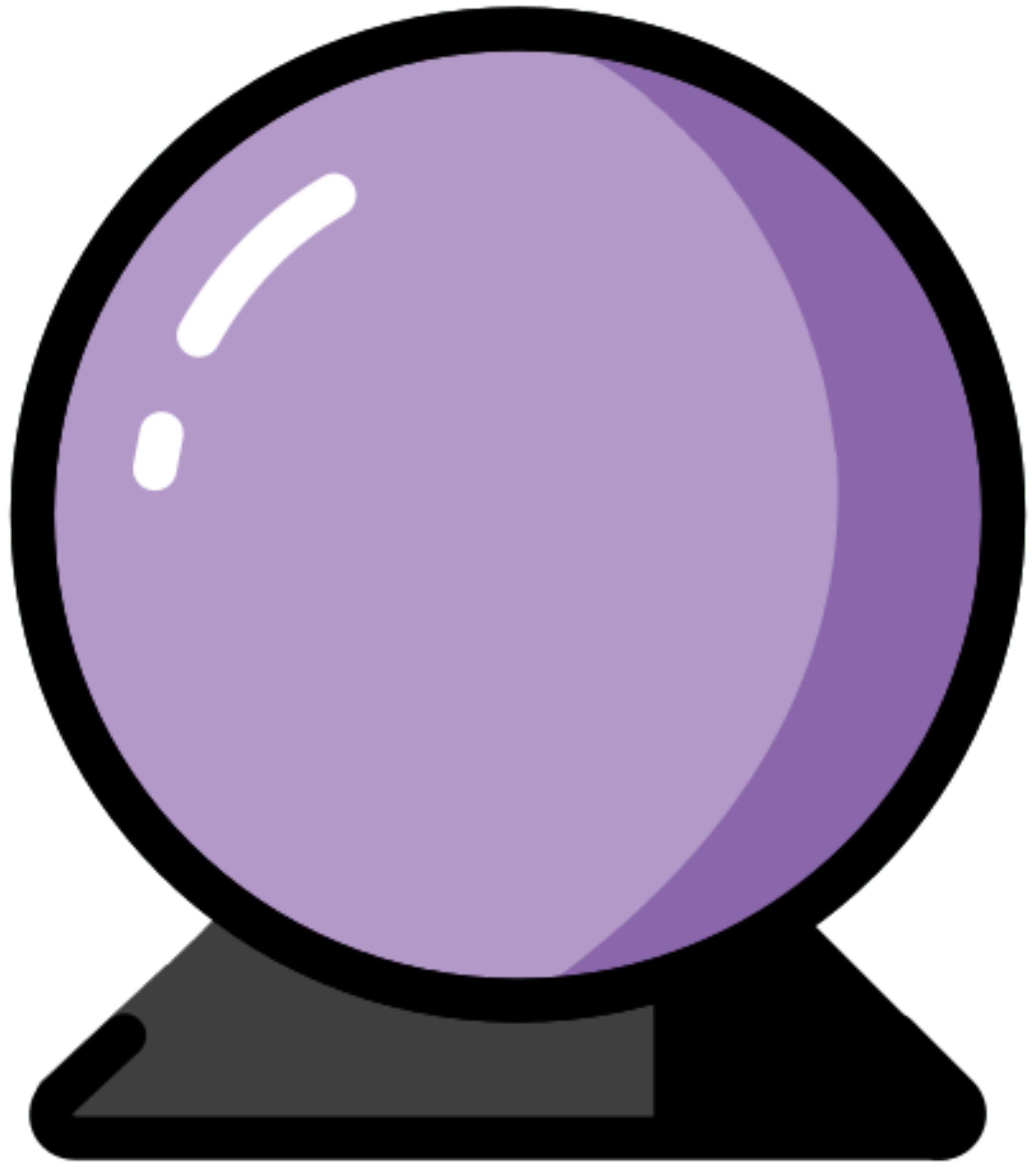
$\mathscr{A}$ can see everything transmitted over the communication channel

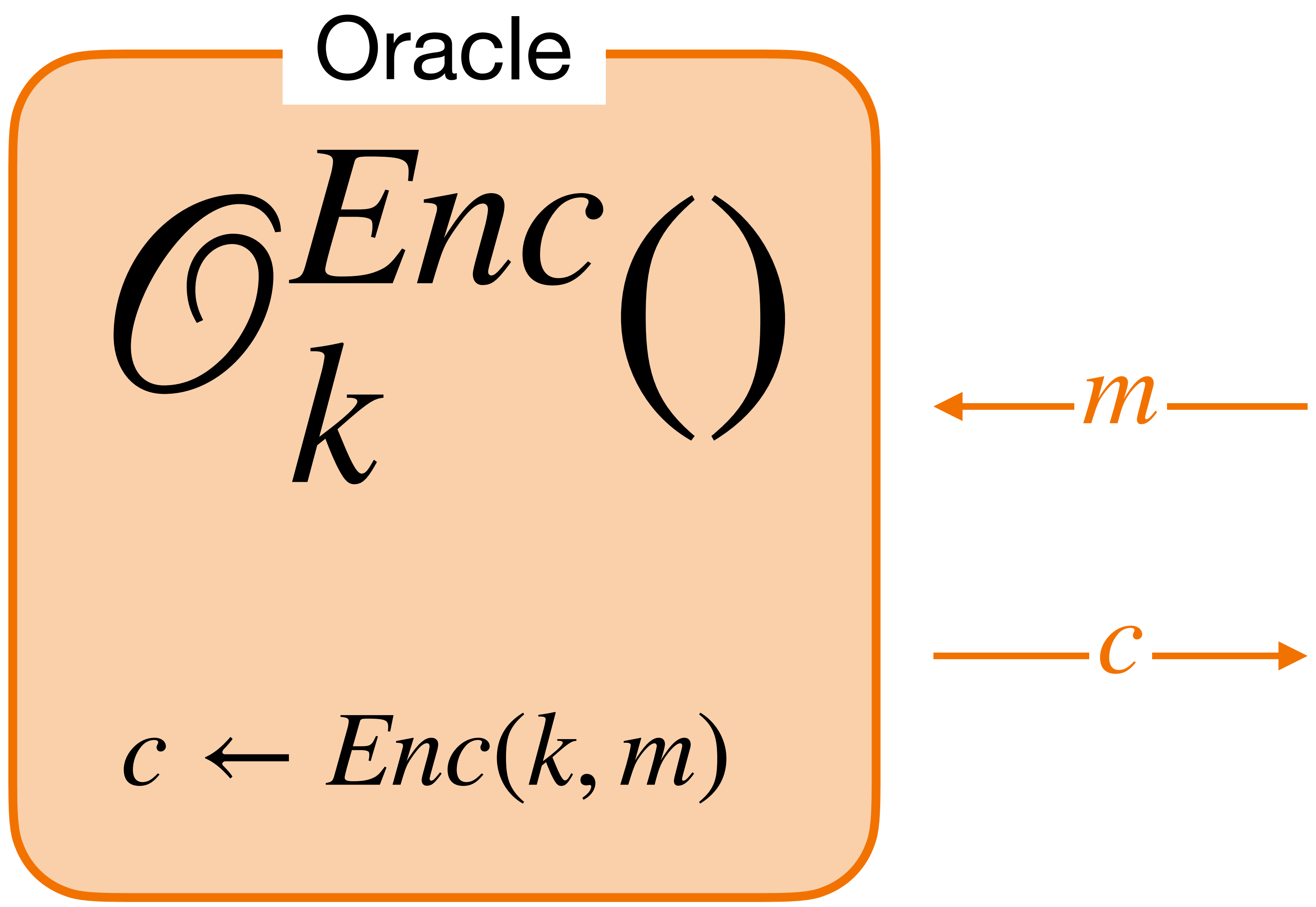
$\mathscr{A}$ knows all details of the encryption scheme except for the secret key (***Kerckhoffs' principle***)

"Security through obscurity" is an obsolete mantra
[computers are good for reverse-engineering, hackers are clever]

Indistinguishability Under Chosen Plaintext Attack (IND-CPA)

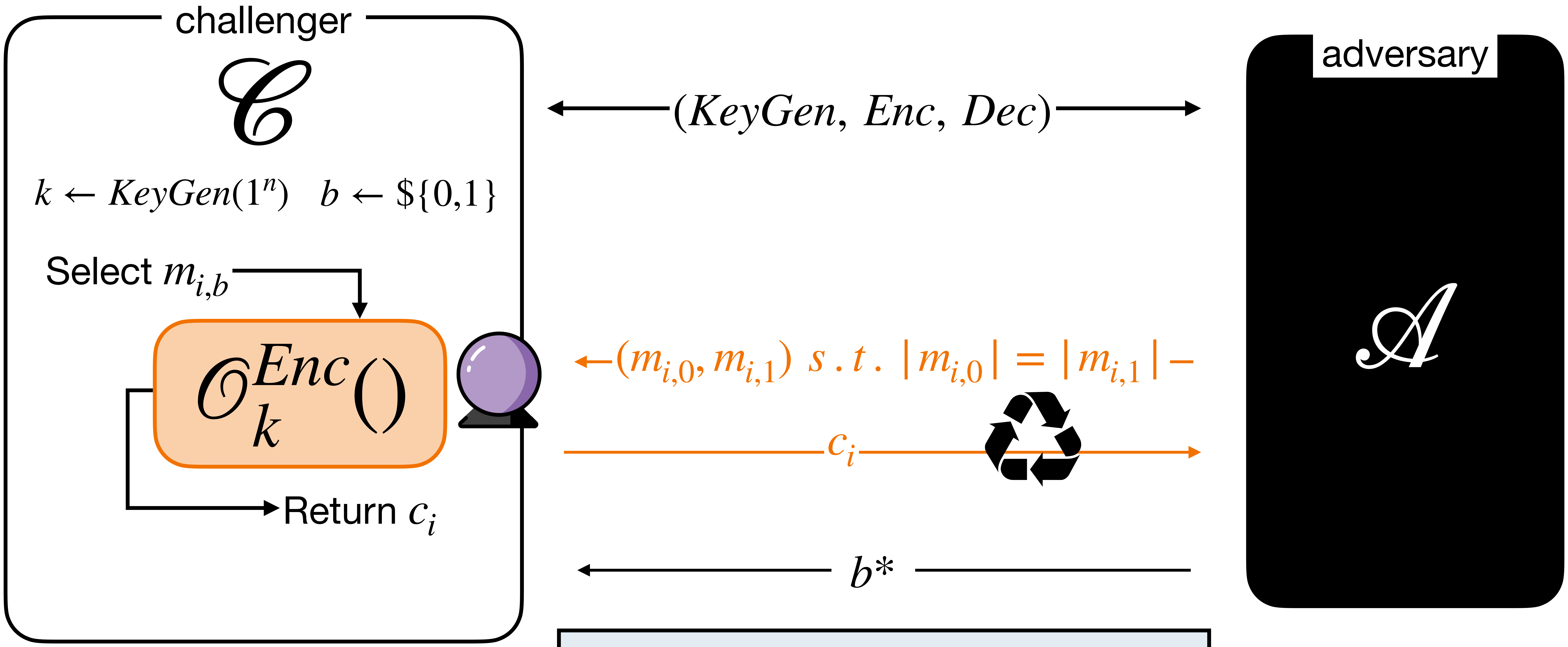
In IND-CPA $\mathscr{A}$ gets access to an encryption oracle





Indistinguishability Under Chosen Plaintext Attack (IND-CPA)

In IND-CPA $\mathscr{A}$ gets access to an encryption oracle



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

Indistinguishability Under Chosen Plaintext Attack (IND-CPA)

Definition: IND-CPA Advantage

An encryption scheme is said to be indistinguishable under chosen plaintext attack (IND-CPA secure) if for any PPT adversary $\mathscr{A}$ that engages in the IND-CPA game, $\mathscr{A}$ only has negligible advantage in winning:

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

Is CBC or CTR Mode IND-CPA Secure?

Yes, both are! But we'll skip the formal proofs (too technical for this course)

I didn't formally define what a secure block cipher is… for the purpose of this course this corresponds to (E,D) being indistinguishable from a random permutation over the block space $\{0,1\}^n$

If (E,D) is a **secure block cipher**, then:
- using (E,D) in **CBC** mode yields an **IND-CPA** secure cipher
- using (E,D) in **CTR** mode yields an **IND-CPA** secure cipher

It is not possible to mathematically prove a block cipher to be secure, instead, confidence on its security builds up over years of scrutiny

A Word on Padding for Block Ciphers

Padding of messages is often needed before encryption with a block cipher.
(A message must be as long as a integer number of full blocks)

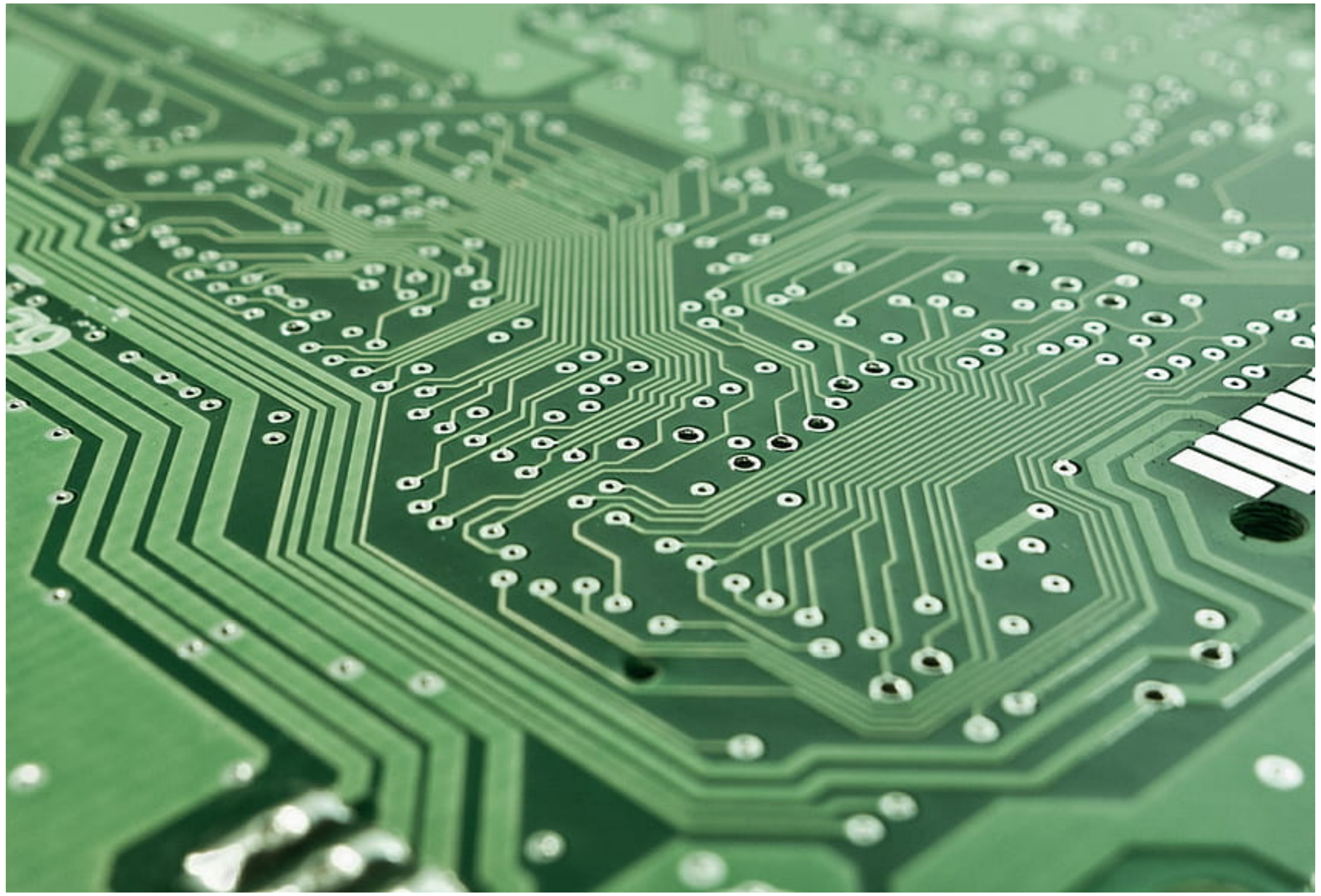
- Essential property: padding must be **reversible**, i.e., the receiver must be able to remove padding in a unique way.

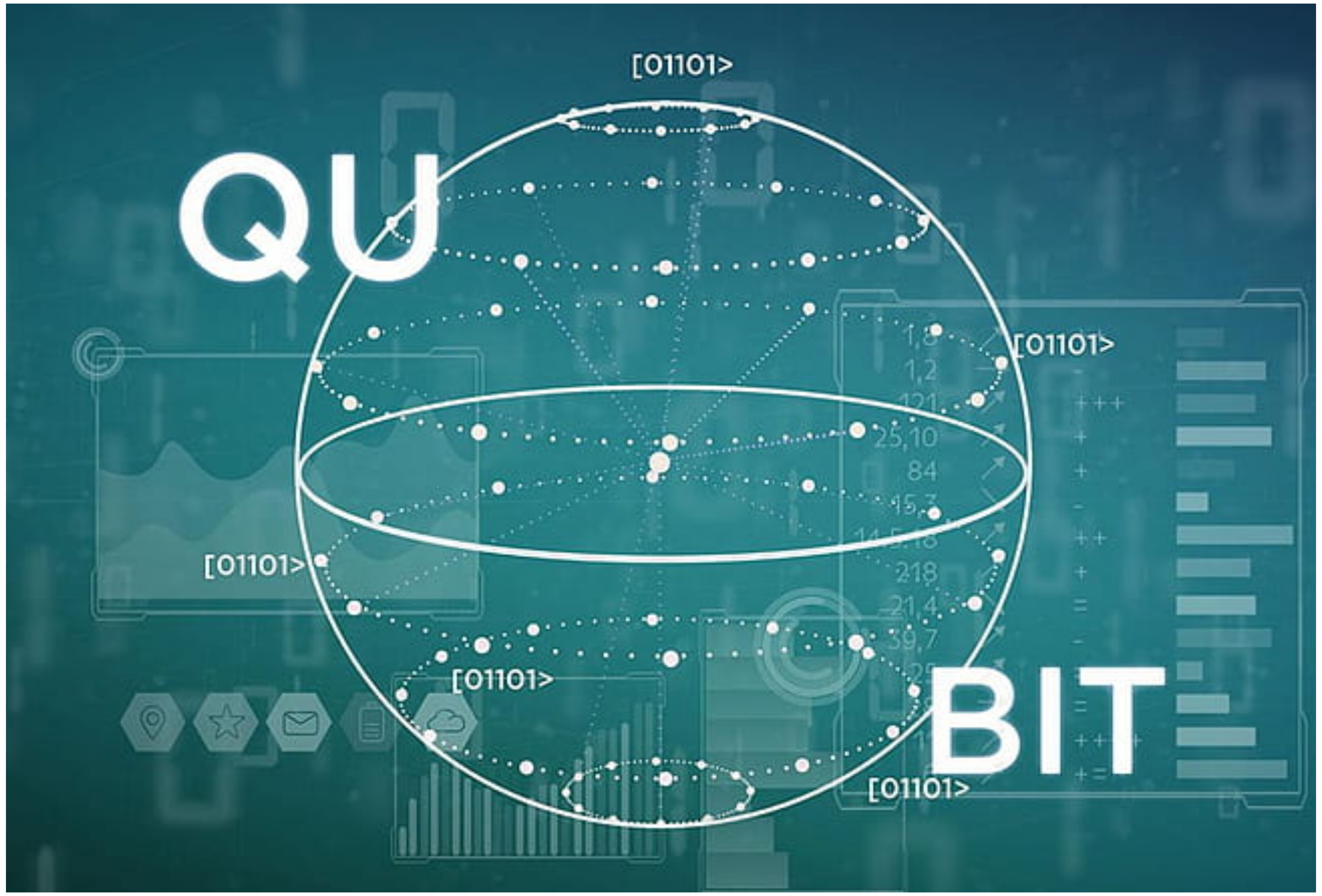
- Example of **insecure** padding: Add necessary number of **zero bytes** to fill last block.

- The receiver should always check that the padding is correctly applied when removing it. In case of mismatch, the protocol should be immediately aborted.

How Long Will AES Remain “Secure” for?

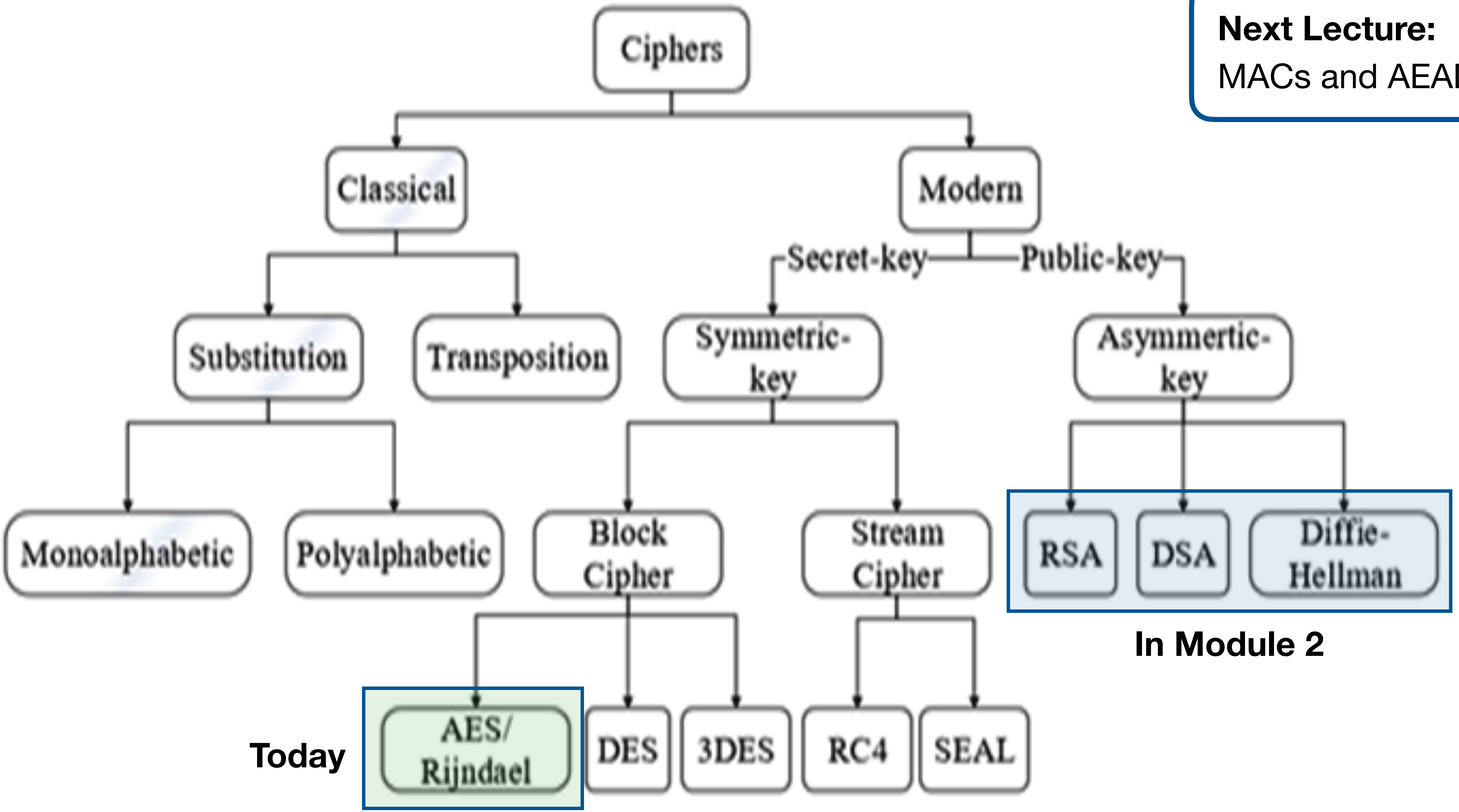
‘Classical’ vs ‘quantum’ computing





Grover’s algorithm runs a quantum brute force of AES keys in time $\sqrt{classical}$. A ‘simple’ mitigation that preserves security against quantum attackers is to double the key length.

Where in the Crypto-Universe Are We?



Next Lecture:
MACs and AEAD