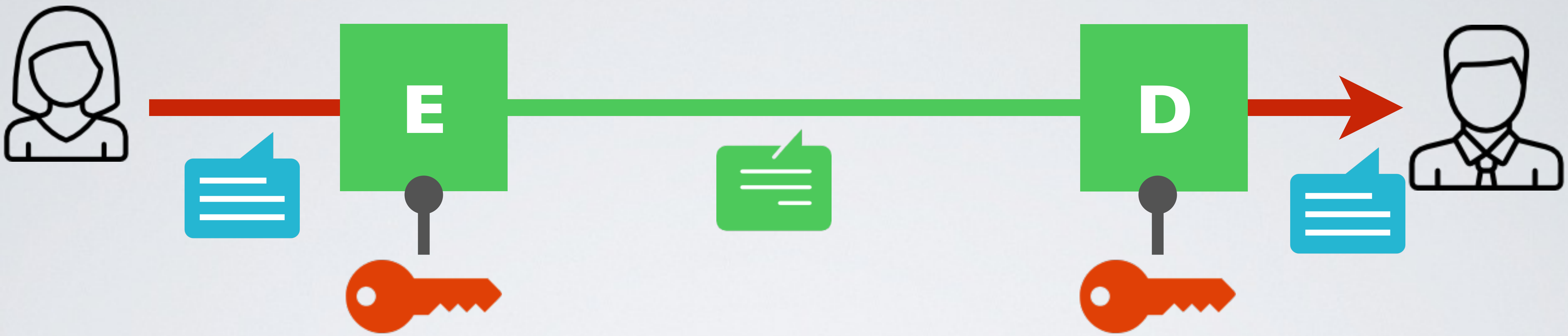
Introductory Cryptography
Symmetric Encryption

Kc Udonsi

Symmetric Key Encryption



➡ The same key $\mathbf{k}$ is used for encryption $\mathbf{E}$ and decryption $\mathbf{D}$

1. $D_k(E_k(m))=m$ for every k, E_k is an injection with inverse D_k
2. $E_k(m)$ is easy to compute (either polynomial or linear)
3. $D_k(c)$ is easy to compute (either polynomial or linear)
4. $c = E_k(m)$ finding m is hard without k (exponential)

Types of Symmetric Key Algorithms/Ciphers

Stream cipher

➡ Each bit is encrypted independently in a "stream"

RC4 - Rivest Cipher 4 (now deprecated)

Salsa20

Block cipher

➡ Blocks of data are encrypted in rounds

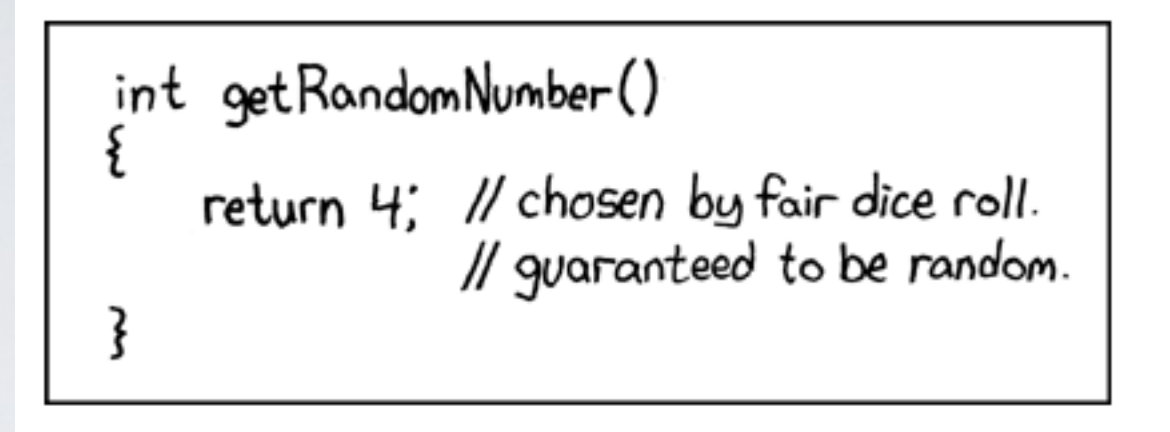
- Encryption standards

  DES (and 3DES) - Data Encryption Standard (now deprecated)

  AES - Advanced Encryption Standard

- Block cipher modes of operation

Random Number Generator

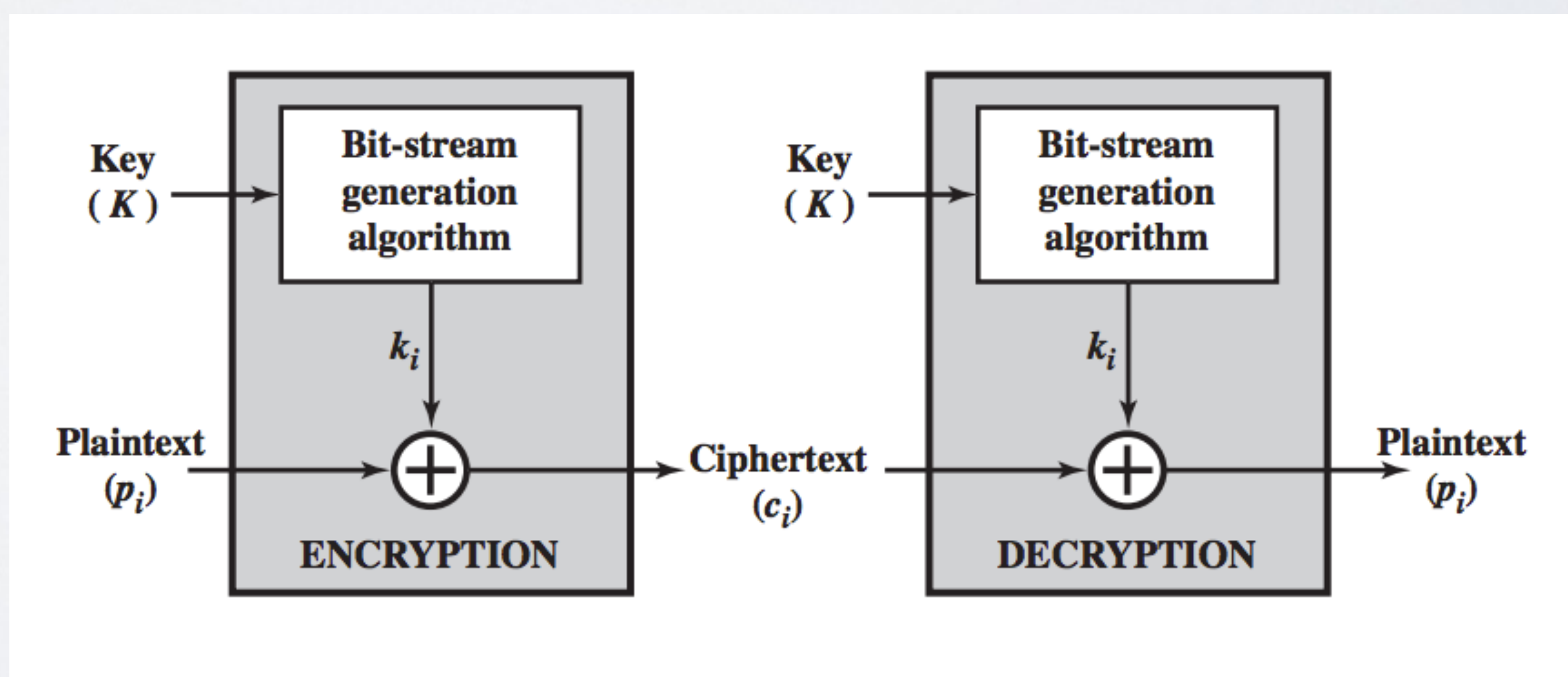


True Random Number Generator

➡ <u>No</u>, because we want to be able to encrypt and decrypt

Pseudo-Random Generator

➡ Stretch a a fixed-size <u>seed</u> to obtain an unbounded random sequence



Stream cipher

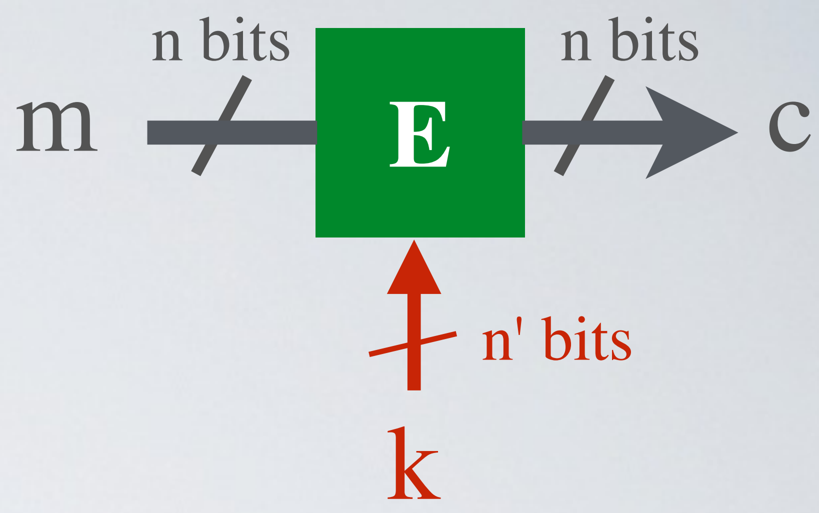
Can we use **k** as a seed?

$$E_k(m) = m \oplus RNG(k)$$

➡ Be careful of key reused attack !

Block Cipher

Ideal block cipher



- Combines confusion (substitution) and diffusion (permutation)
- Changing single bit in plaintext block or key results in changes to approximately half the ciphertext bits

➡ Completely obscure statistical properties of the original message

➡ A known-plaintext attack does not reveal the key

DES - Data Encryption Standard

Block size	64 bits
Key Size	56 bits
Speed	~ 50 cycles per byte
Algorithm	Feistel Network

Timeline

- **1972** NBS call for proposals
- **1974** IBM Lucifer proposal
  analyzed by DOD and enhanced by NSA
- **1976** adopted as standard
- **2004** NIST withdraws the standard

Security of DES - DES Challenges (brute force contests)

1998 *Deep Crack*, the EFF's DES cracking machine used 1,856 custom chips

- Speed : matter of days
- Cost : $250,000

2006 *COPACOBANA, the* COst-optimized Parallel COdeBreaker used 120 FPGAs

- Speed : less than 24h
- Cost : $10,000

3DES (Triple DES)

$$3DES_{k1,k2,k3}(m) = E_{k3}(D_{k2}(E_{k1}(m)))$$

- ➡ Uses three keys; some same or all distinct
- ➡ Effective key length (entropy) : 112 bits or 168 bits
- ✓ Very popular, used in PGP, TLS (SSL) …
- ◉ But terribly slow

AES - Advanced Encryption Standard

Timeline

- **1996** NIST issues public call for proposal
- **1998** 15 algorithms selected
- **2001** winners were announced

Rijndael by *J. Daemen and V. Rijmen*

Block size	128 bits
Key Size	128, 192, 256 bits
Speed	~18-20 cycles / byte
Mathematical Foundation	Galois Fields
Implementation	• Substitution-permutation network • Basic operations : $\oplus$, + , shift • Small code : 98k

Adopted by the NIST in December 2001

(pure) Encryption Modes
a.k.a. how to encrypt long messages

ECB - Electronic Code Book

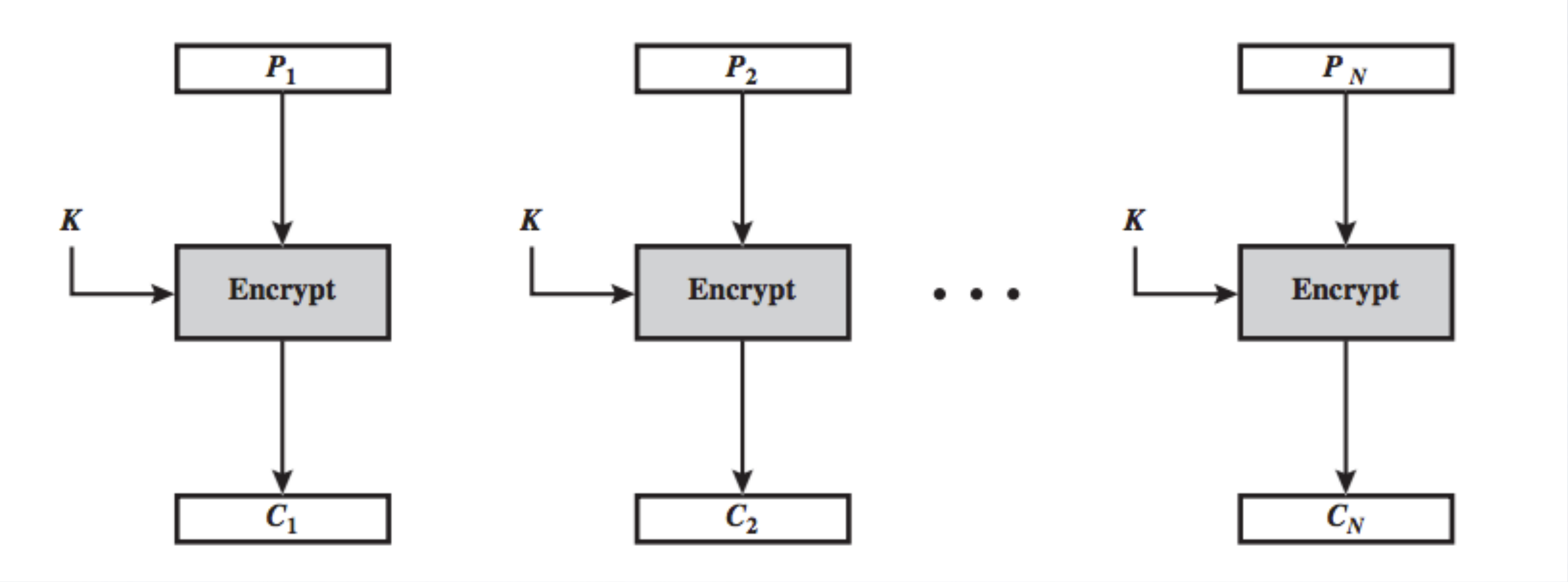
CBC - Cipher Block Chaining

CFB - Cipher Feedback

OFB - Output Feedback

CTR - Counter

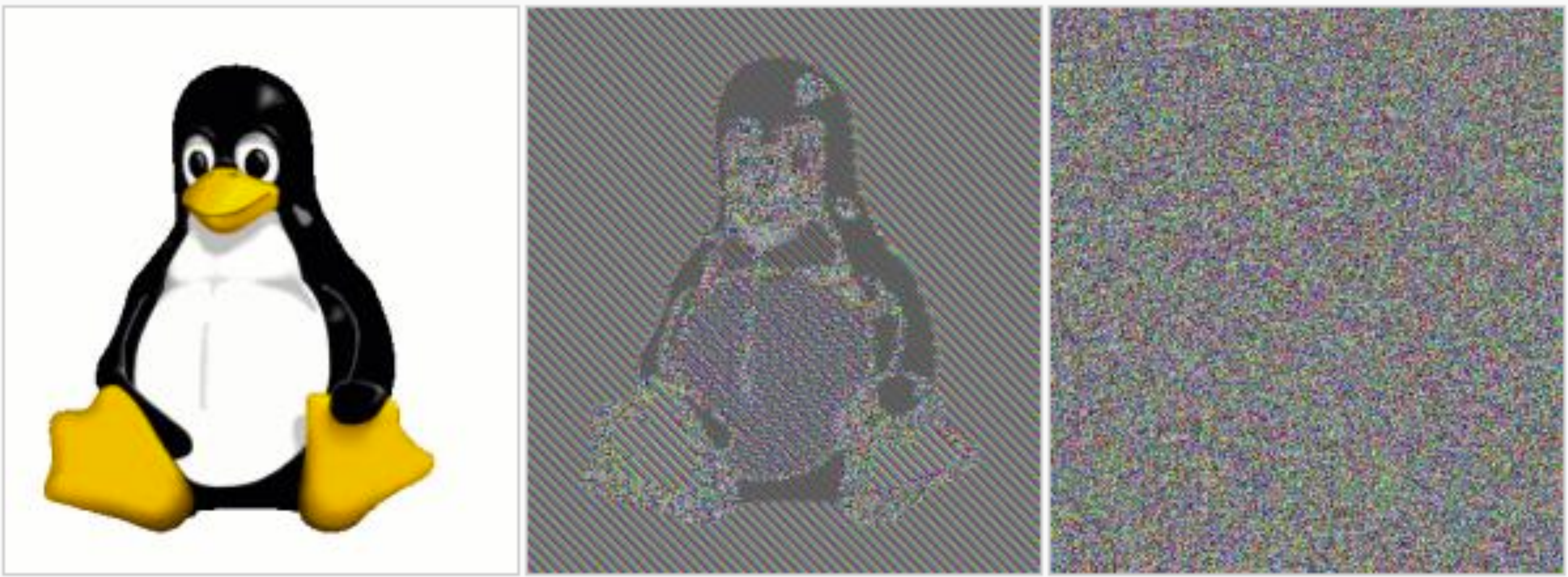
ECB - Electronic Code Book



Each plaintext block is encrypted <u>independently with the key</u>

- ✓ Block can be encrypted in parallel
- ◉ The same block is encrypted to the same ciphertext

How bad is ECB mode with a large data?

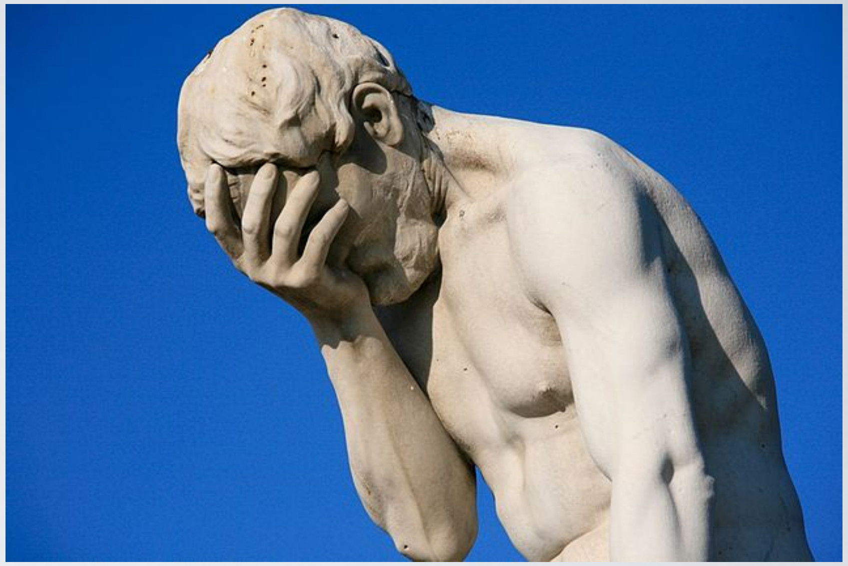


Original image

Encrypted using ECB mode

Modes other than ECB result in pseudo-randomness

source: *Wikimedia*



Simple Illustration of Zoom Encryption Failure

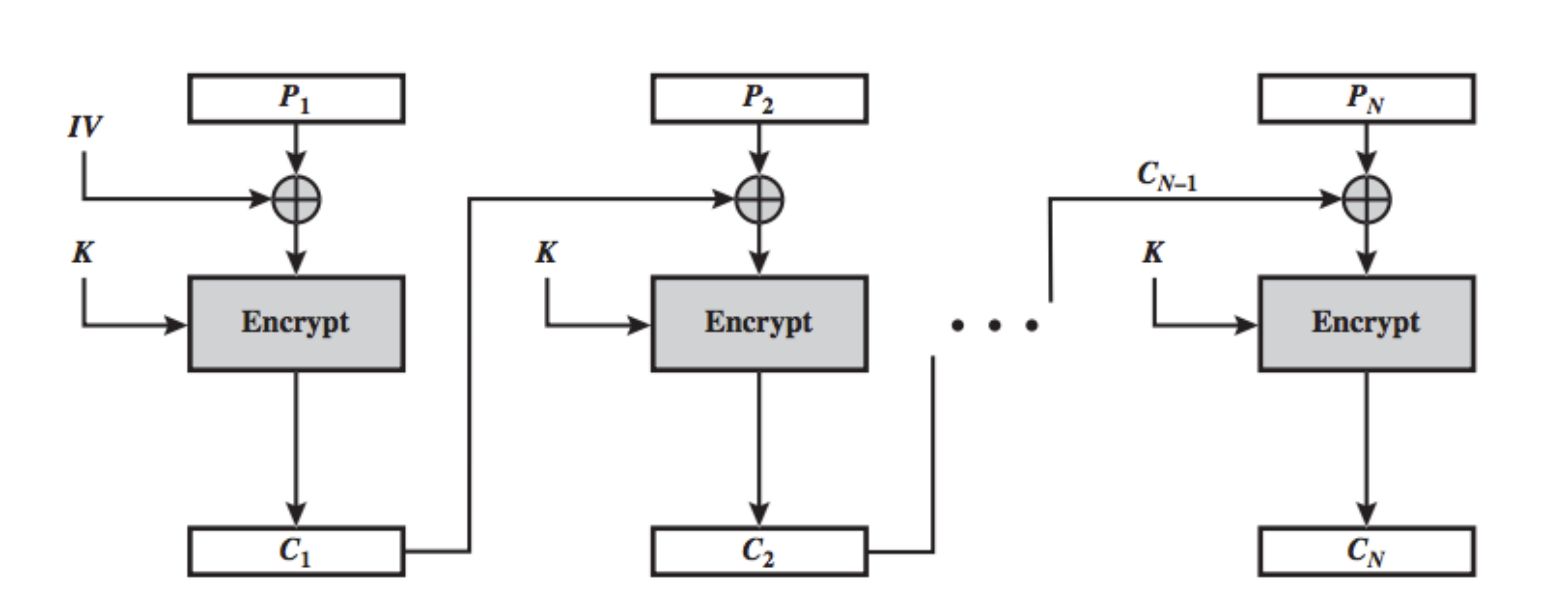
by Davi Ottenheimer on April 10, 2020

The Citizen Lab April 3rd, 2020 report broke the news on Zoom using weak encryption and gave this top-level finding:

> Zoom documentation claims that the app uses "AES-256" encryption for meetings where possible. However, we find that in each Zoom meeting, a single AES-128 key is used in ECB mode by all participants to encrypt and decrypt audio and video. The use of ECB mode is not recommended because patterns present in the plaintext are preserved during encryption.

source: *Security Boulevard*

CBC - Cipher Block Chaining



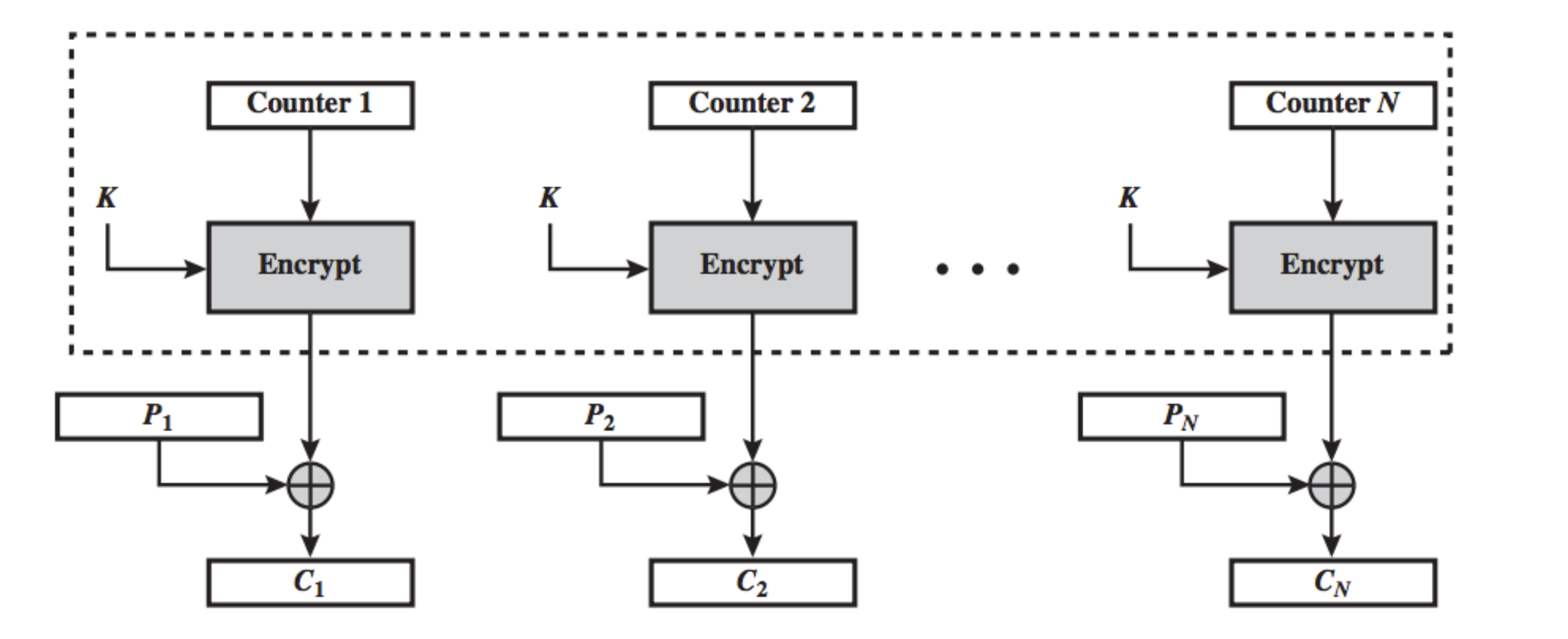
Introduce some <u>randomness</u> using the previous ciphertext block

✓ Repeating plaintext blocks are not exposed in the ciphertext

◉ No parallelism

➡ The Initialization Vector should be known by the recipient

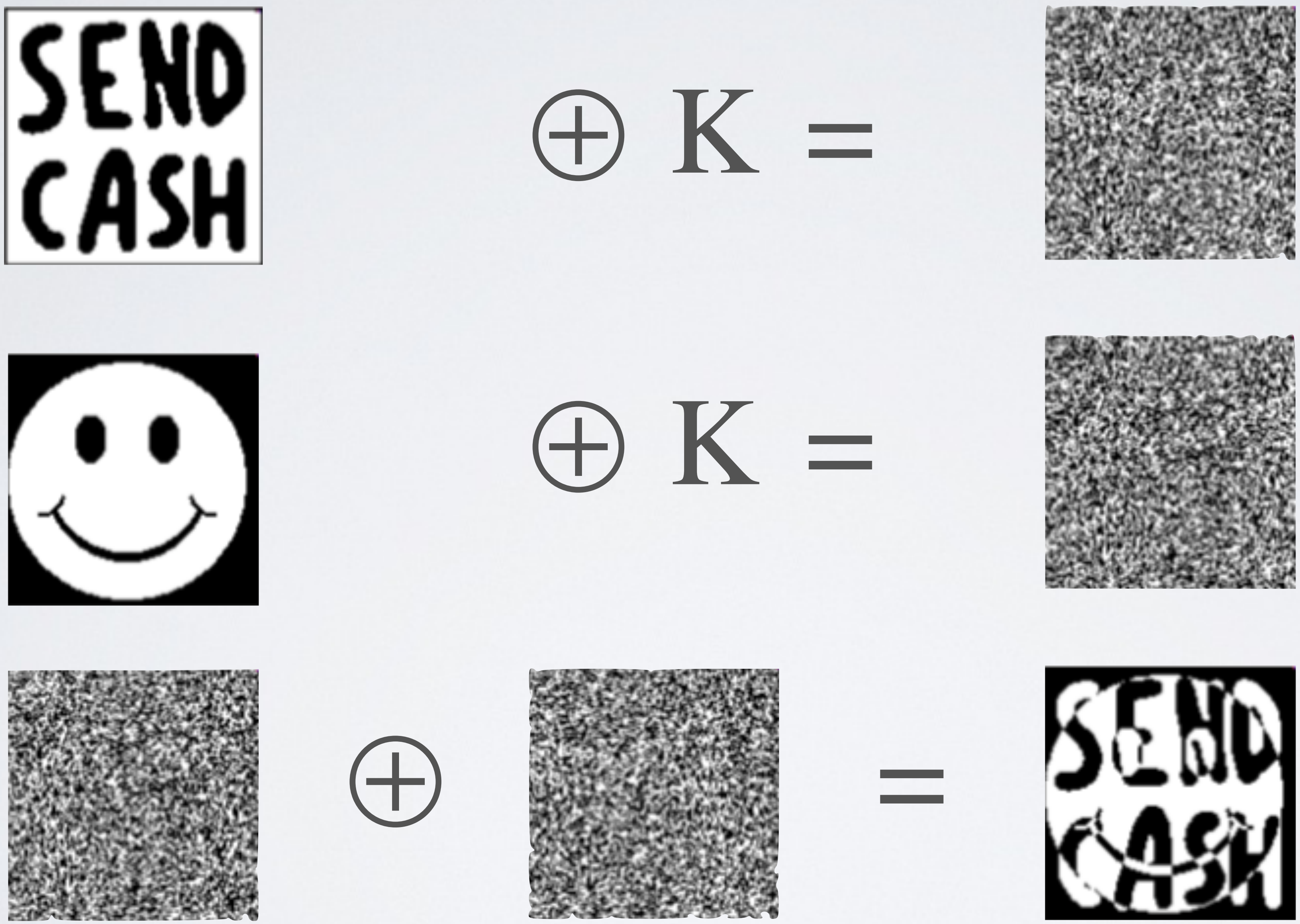
CTR - Counter



Introduce some <u>randomness</u> using a counter

- ✓ High entropy and parallelism
- ◉ Sensitive to key-reused attack

Key-reused attack on CTR



Stream Cipher vs Block Cipher

	Stream Cipher	Block Cipher
Approach	Encrypt one symbol of plaintext directly into a symbol of ciphertext	Encrypt a group of plaintext symbols as one block
Pro	Fast	High diffusion
Cons	Low diffusion Key reused attack	Slow

Stream cipher and block cipher are often used together

- Stream cipher for encrypting large volume of data
- Block cipher for encrypting fresh pseudo-random seeds

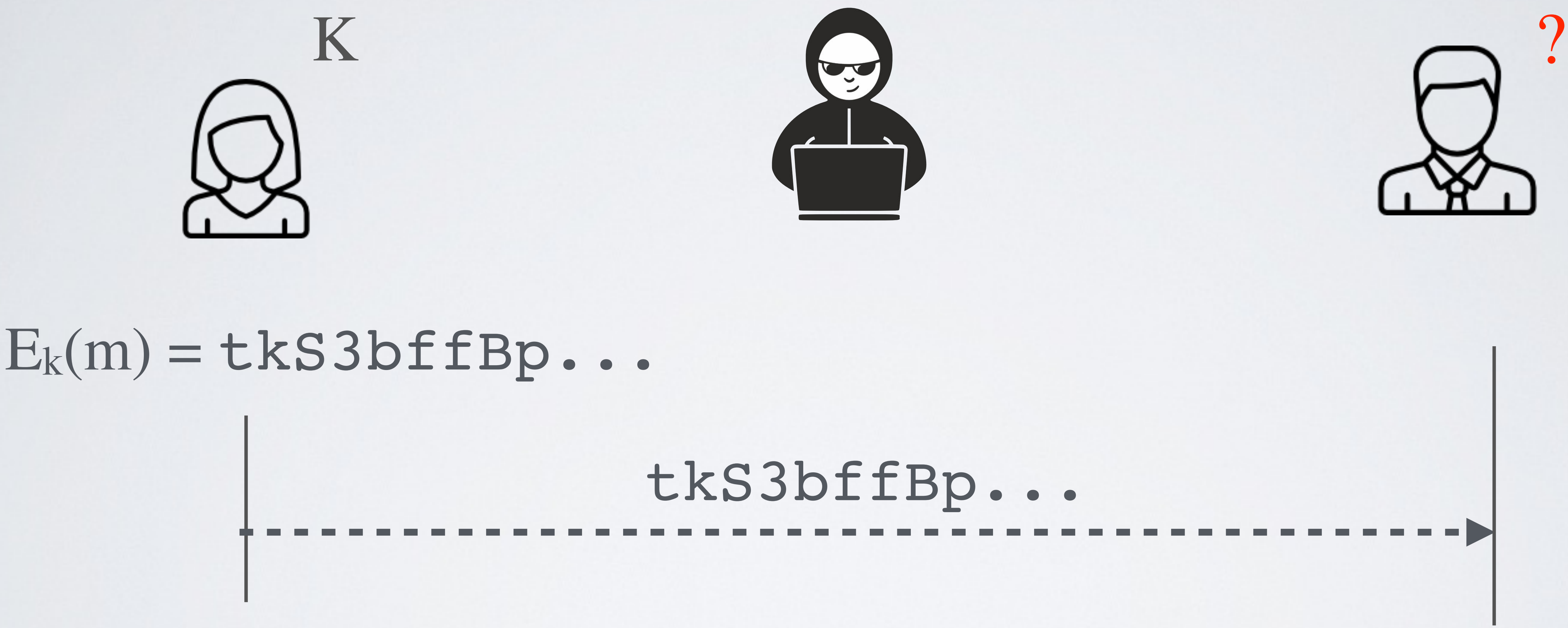
Latest trends

AES is now hardware accelerated (AES-NI native instruction)

➡ AES is fast enough (~1.3 cycles per byte)
to be used as the go-to cipher for any application

https://security.stackexchange.com/questions/22905/how-long-would-it-take-a-single-processor-with-the-aes-ni-instruction-set-to-bru

An issue ...



- How does Alice and Bob agree on a symmetric key?