

Fundamentos de Segurança Informática (FSI)

2021/2022 - LEIC

Manuel Barbosa
mbb@fc.up.pt

Aula 14

Criptografia: Parte 1

Criptografia?

- Historicamente:
 - “arte” onde o segredo e o obscurantismo eram princípios fundamentais
 - principalmente utilizada em meios militares/estatais
- Hoje:
 - ciência => área multidisciplinar, princípios rigorosos
 - normas internacionais estabelecidas e públicas => escrutínio
 - utilização generalizada => **todas** as aplicações que usamos hoje em dia

Criptografia!

- É:
 - Uma ferramenta espantosa!
 - Um componente central em muitos mecanismos de segurança
- Não é:
 - A solução para todos os problemas (apenas uma parte da solução)
 - Fiável se não for bem implementada e corretamente utilizada
 - Algo que se possa fazer em modo DIY

Segurança da Informação

- Proteção da segurança da informação:
 - em trânsito: $A \Rightarrow B$, on-line/síncrono (e.g., HTTPS)
 - em trânsito: $A \Rightarrow B$, off-line/assíncrono (e.g., email)
 - em repouso: $A \Rightarrow A$, e.g., disk encryption

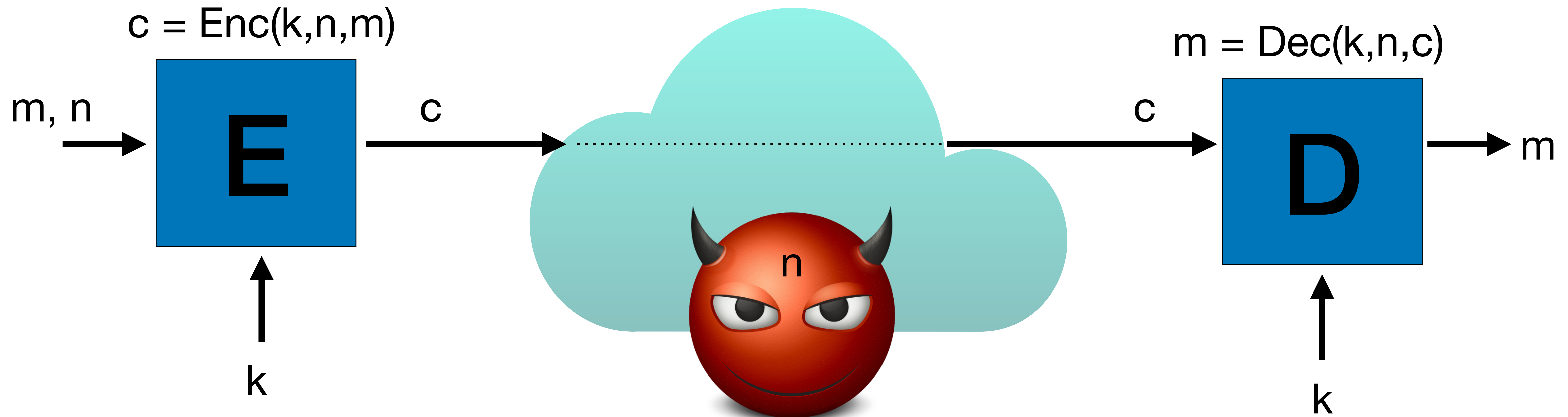
Segurança da Informação


- Proteção da segurança da informação:
 - Confidencialidade
 - informação acessível apenas a emissor A e recetor B (ou A)
 - cifras simétricas e assimétricas, acordos de chave
 - Autenticidade (e integridade)
 - recetor B (ou A) tem a certeza de que os dados vieram (sem alteração) de A
 - assinaturas, acordos de chave, MAC
 - Não-repúdio: assinaturas (emissor A não pode negar envio da mensagem)

Criptografia Moderna

1. Definições precisas e rigorosas de segurança: modelos matemáticos da realidade
2. Quando a segurança de uma construção criptográfica se baseia num pressuposto que não conseguimos provar:
 - A. esse pressuposto deve ser simples e descrito de forma precisa.
 - B. utilizam-se poucos destes pressupostos na criptografia moderna
 - C. são escrutinados por toda a comunidade
3. As construções criptográficas devem ser justificadas formalmente:
 - D. prova de que satisfaz definição de acordo com o princípio #1
 - E. possivelmente, assumindo pressupostos enunciados de acordo com o princípio #2

Cifras Simétricas

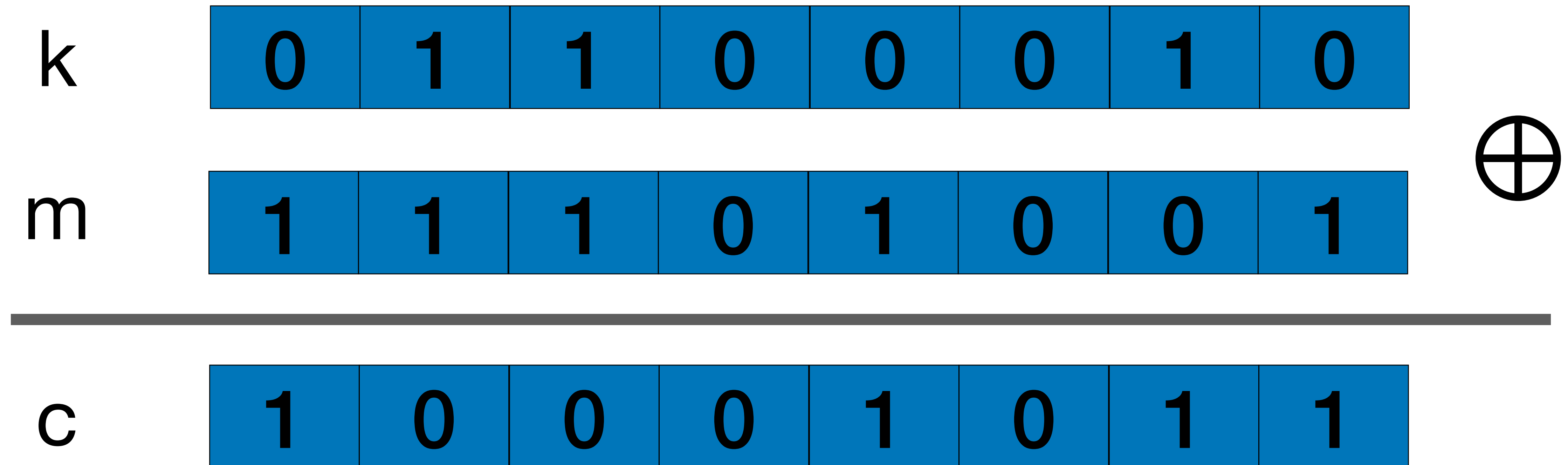


- E, D: algoritmos (encrypt, decrypt) => **públicos e standard!!!**
- k: chave secreta (hoje tipicamente 128 bits, porquê?)
- 
- n, m, c: nonce (non-repeating, **público**), texto-limpo, criptograma

Casos de Uso

- Chave usada apenas uma vez (one-time key)
 - exemplo: email cifrado usa chave fresca para cada mensagem
 - nonce não é relevante: pode ser fixado a 0 por exemplo
- Chave usada muitas vezes (many-time key)
 - Exemplo: cifrar disco ou HTTPS/TLS
 - nonce não se pode repetir: número de sequência ou valor aleatório

Exemplo: One-Time Pad (Vernam, 1917)



- Cifrar: $E(k, m) = m \oplus k$
- Decifrar: $D(k, c) = c \oplus k = (m \oplus k) \oplus k = m$

Segurança do One-Time-Pad

- Primeiro resultado formal para uma garantia de segurança (Sh49):
 - OTP garante **confidencialidade** contra “**eavesdroppers**”
 - formalmente: distribuição do criptograma totalmente aleatória
- Problema:
 - chave do tamanho do texto limpo!
 - chave usada apenas uma vez!

Cifras Sequenciais

- Solução para o comprimento da chave, mas ainda one-time

k

128-bits

$$E(k, m) = m \oplus \text{PRG}(k)$$

Gerador Pseudo-Aleatório (PRG)

0

1

1

0

0

0

1

0

m

1

1

1

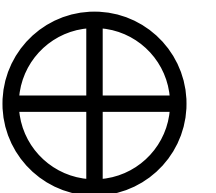
0

1

0

0

1



c

1

0

0

0

1

0

1

1

Cifras Sequenciais

- Perigos:
 - Se não houver nonce $\text{PRG}(k)$ é sempre igual
 - Em duas cifrações com a mesma chave

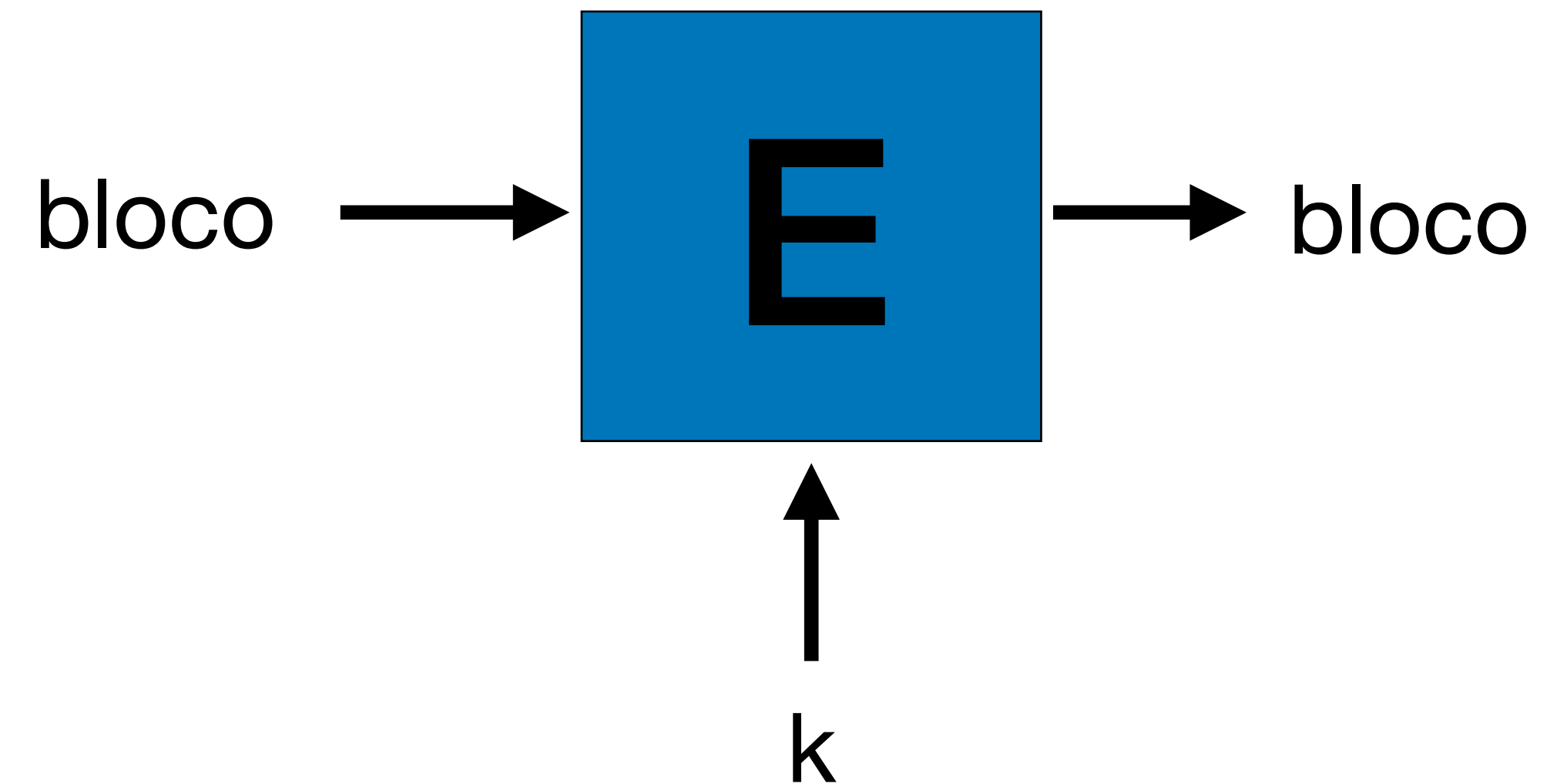
$$c_1 = m_1 \oplus \text{PRG}(k) \qquad c_2 = m_2 \oplus \text{PRG}(k)$$

- Isto é totalmente inseguro! Porquê? $\implies c_1 \oplus c_2 = m_1 \oplus m_2$
- Solução:
 - PRG moderno permite usar nonce público (E.g., ChaCha20):

$$c_1 = m_1 \oplus \text{PRG}(k, n_1) \qquad c_2 = m_2 \oplus \text{PRG}(k, n_2)$$

Cifras de Bloco

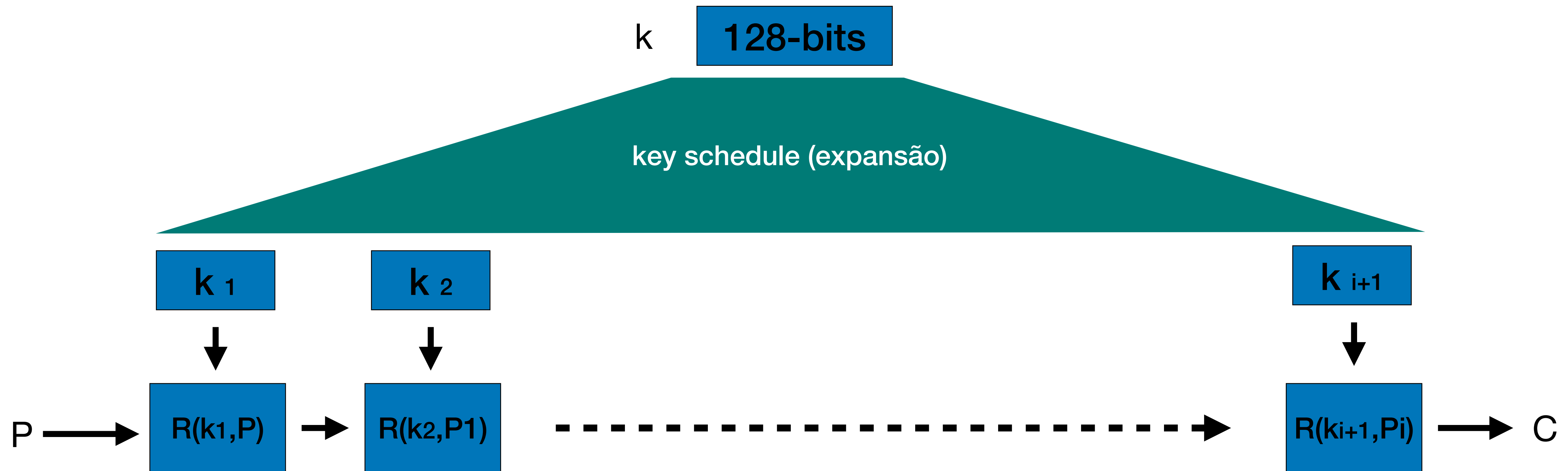
- Apesar do nome não são cifras => permitem construir cifras
- Usadas para muitas outras finalidades
- Exemplos:
 - DES (até 2000): bloco de 64 bits, chave de 56 bits
 - AES (desde 2000): bloco de 128 bits, chave de 128, 256, 512 bits



- Propriedade:
 - para k aleatório e secreto
 - $E(k, B)$ parece aleatório
 - mesmo escolhendo B

Cifras de Bloco: Como Funcionam?

- Implementações pequenas e eficientes: iteração de uma transformação



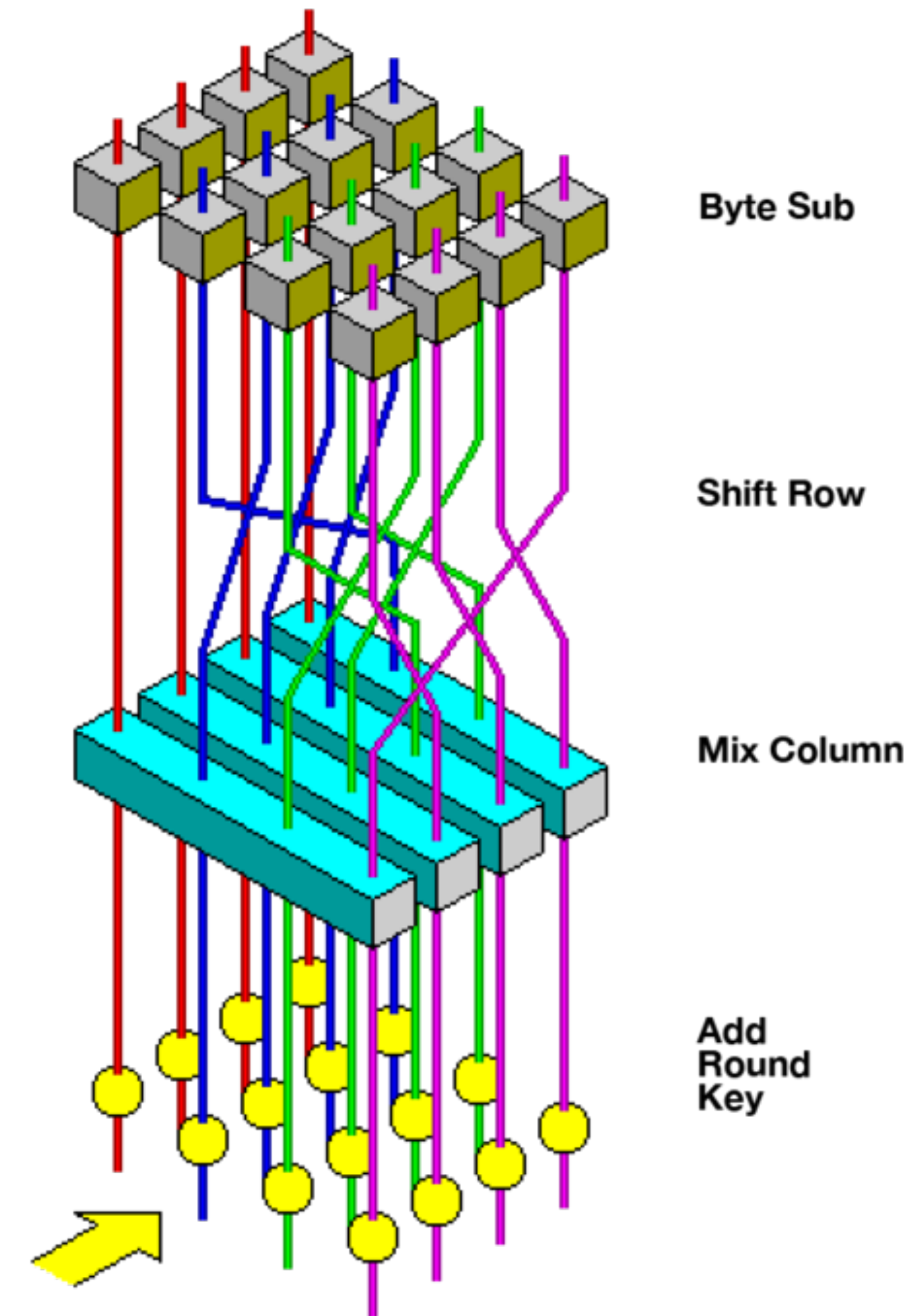
- $R(k, P_i) \Rightarrow$ transformação de round (10 rounds no AES 128, 14 no AES 256)

Cifras de Bloco: AES

- AES (Advanced Encryption Standard)
- Provavelmente o algoritmo criptográfico mais utilizado
- Implementações disponíveis em hardware
- Os processadores mais utilizados (Intel, AMD, ARM) oferecem AES-NI
 - instruções para computar um round de AES
 - instruções para preparar a sequência de chaves para cada round
 - velocidade uma ordem de magnitude melhor que software (e mais seguro, porquê? ==> *side channels*)

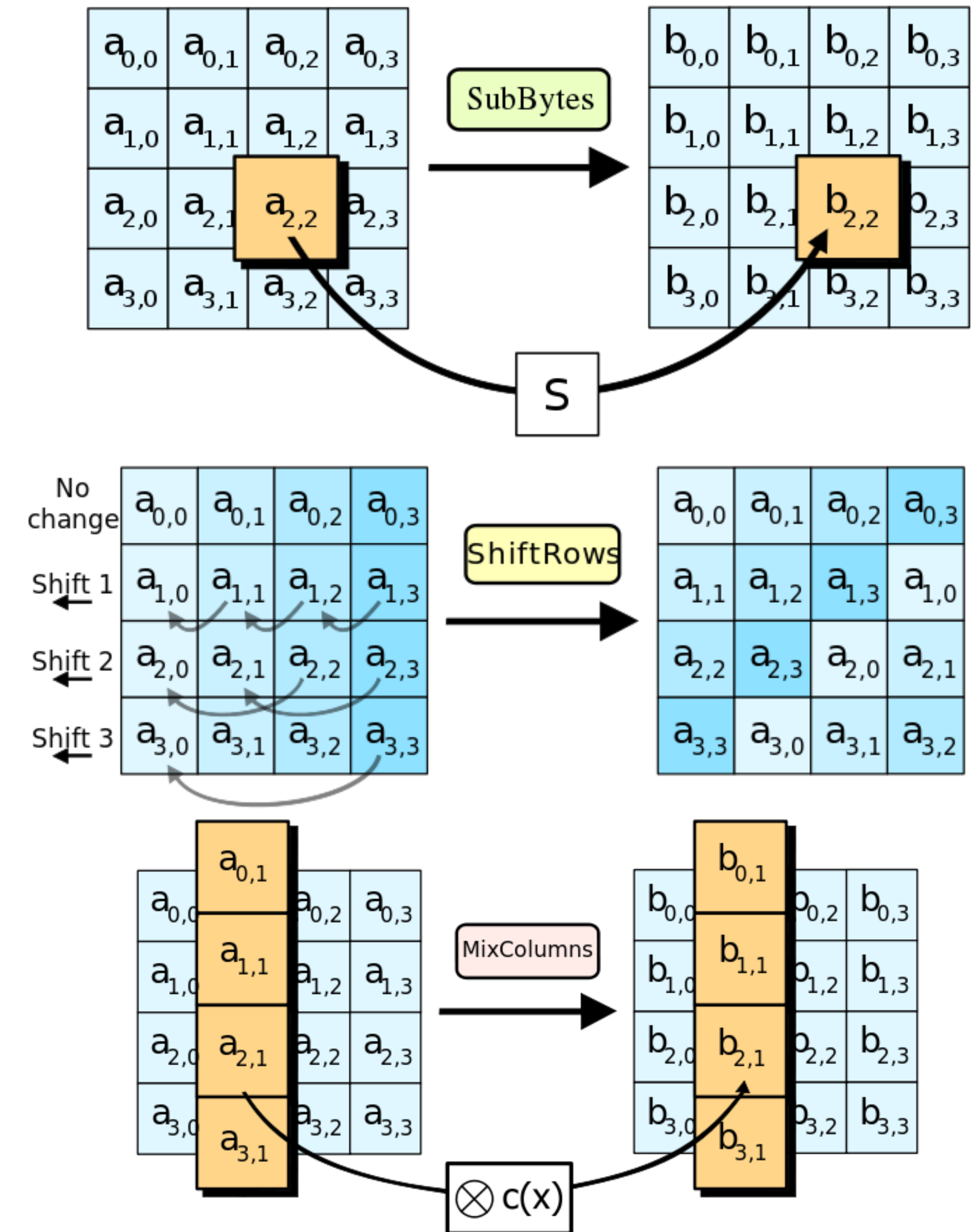
AES: Como funciona

- Estado = matriz 4x4 bytes
- Um round = 4 transformações:
 - AddRoundKey, MixColumn, ShiftRow, ByteSub
- Cada round usa uma chave derivada da chave da cifra => key schedule
- AES é seguro?
 - melhor ataque $\sim 2^{|k|}$ operações



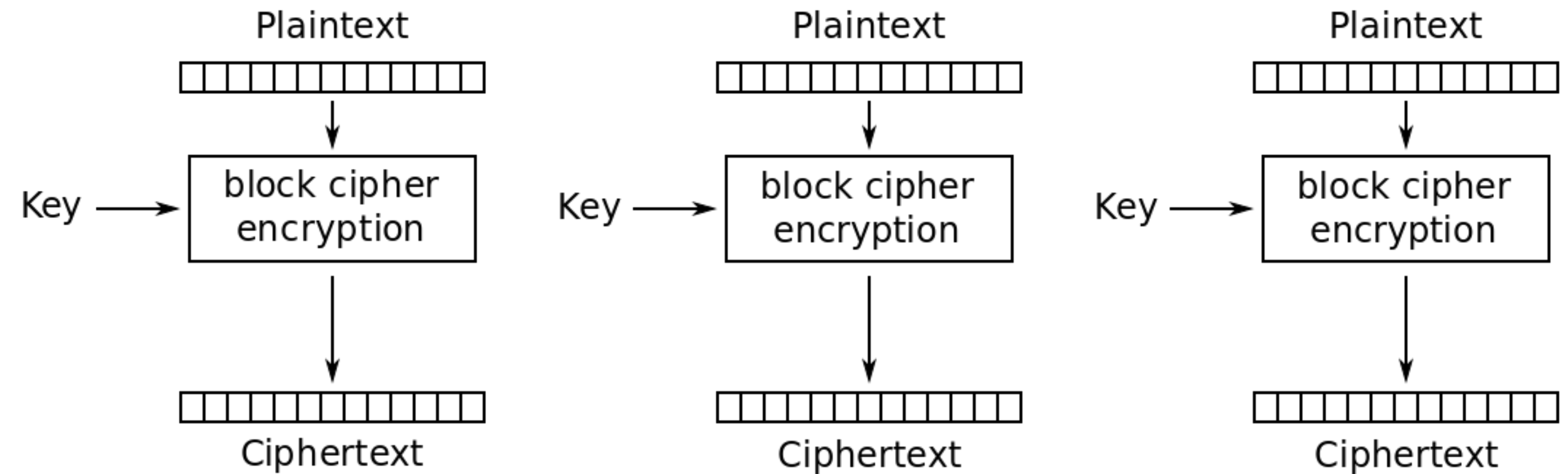
AES: Como funciona

- Estado = matriz 4x4 bytes
- Um round = 4 transformações:
 - AddRoundKey, MixColumn, ShiftRow, ByteSub
- Cada round usa uma chave derivada da chave da cifra => key schedule
- AES é seguro?
 - melhor ataque $\sim 2^{|k|}$ operações

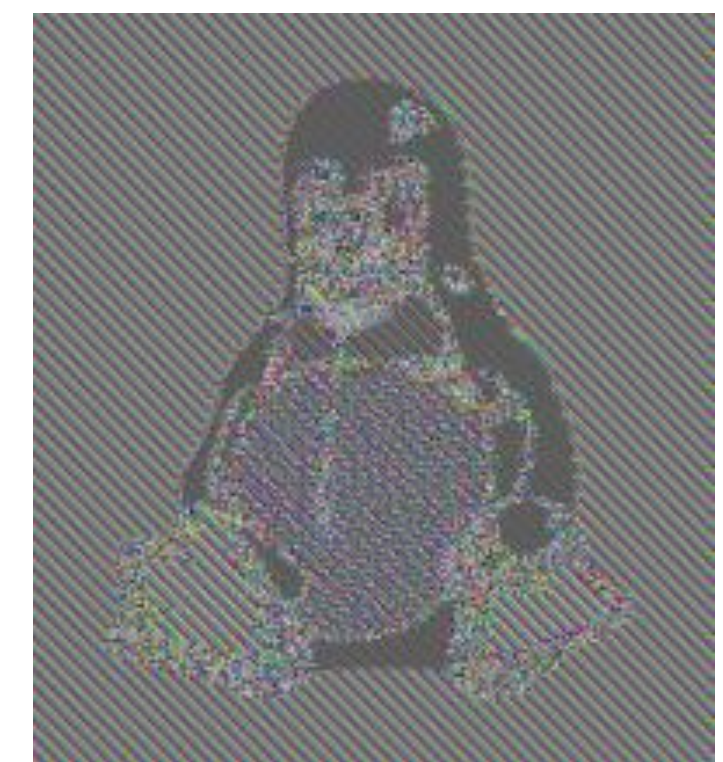
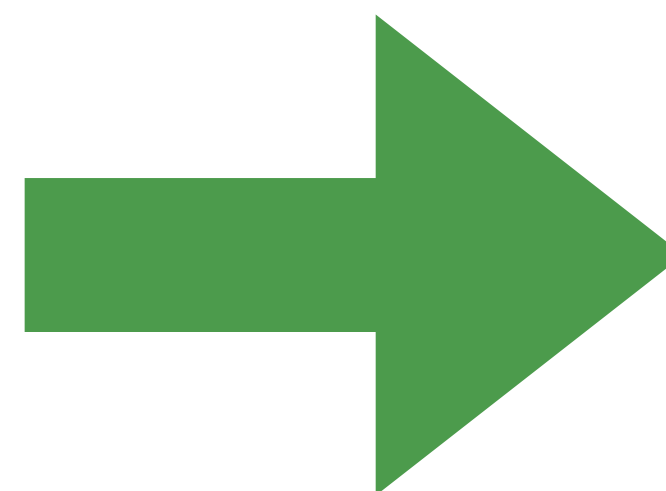


Cifra de Bloco \neq Cifra Segura

- Cifra de Bloco:
 - Bloco controlado por aplicação
=> bloco pseudo-aleatório
 - Como usar esta funcionalidade para cifrar, e.g., um ficheiro?
- Nem todas as maneiras são seguras:
 - Electronic Code Book (inseguro):
 - blocos do ficheiro iguais
 - blocos de criptograma iguais!



Electronic Codebook (ECB) mode encryption



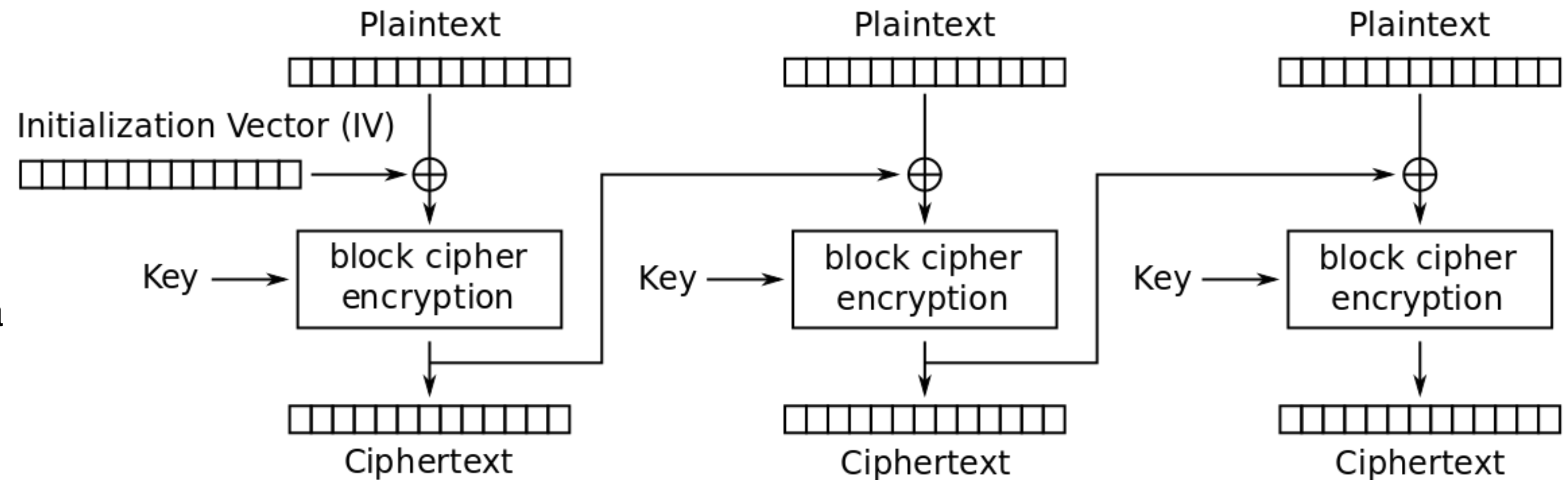
Cifra de Bloco \Rightarrow Cifra Segura

- Cifra de Bloco:

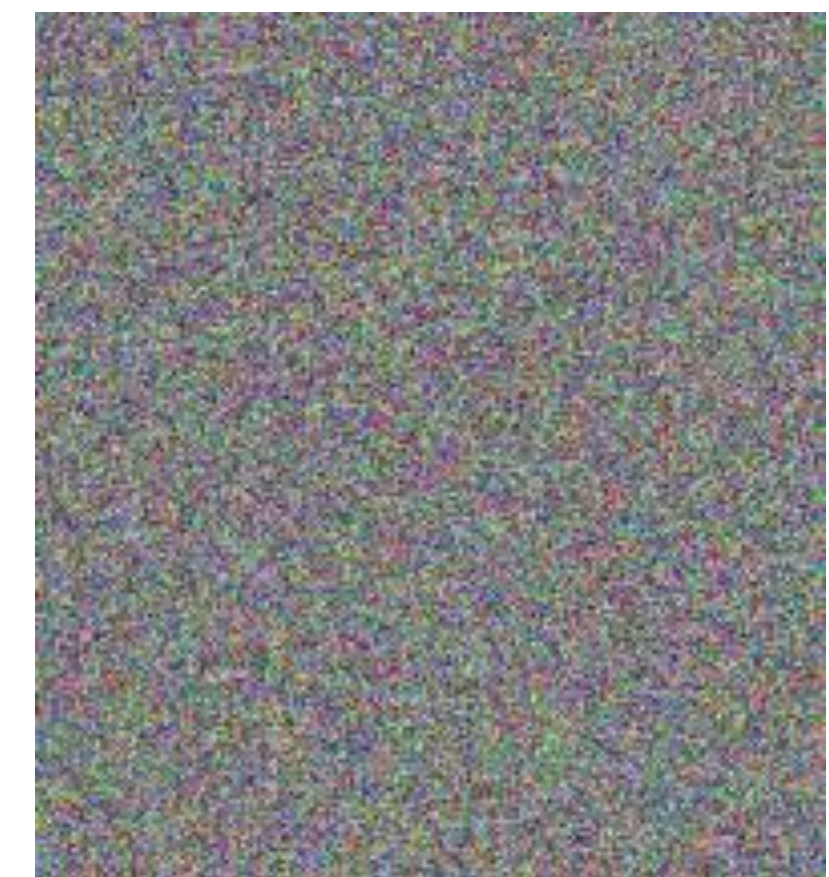
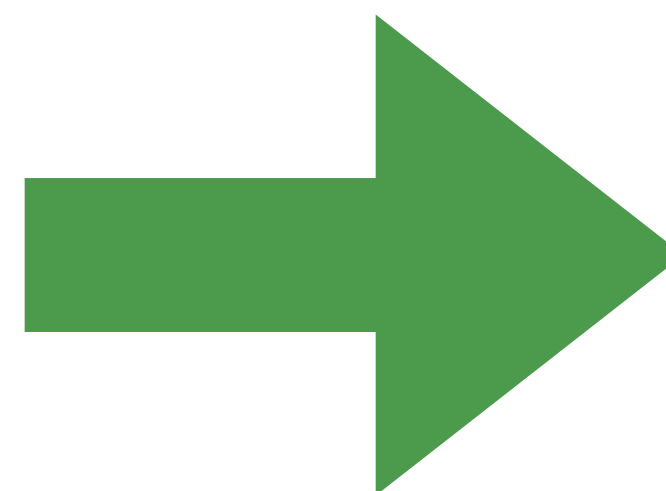
- Bloco controlado por aplicação \Rightarrow bloco pseudo-aleatório
- Como usar esta funcionalidade para cifrar, e.g., um ficheiro?

- Algumas maneiras são seguras:

- Cipher Block Chaining Mode (seguro)
- Porquê? (Estudar criptografia \Rightarrow prova)
- Intuição: plaintext não é aplicado a AES
 - aplica máscara a cada bloco: criptograma anterior
 - bloco 0: máscara = IV



Cipher Block Chaining (CBC) mode encryption



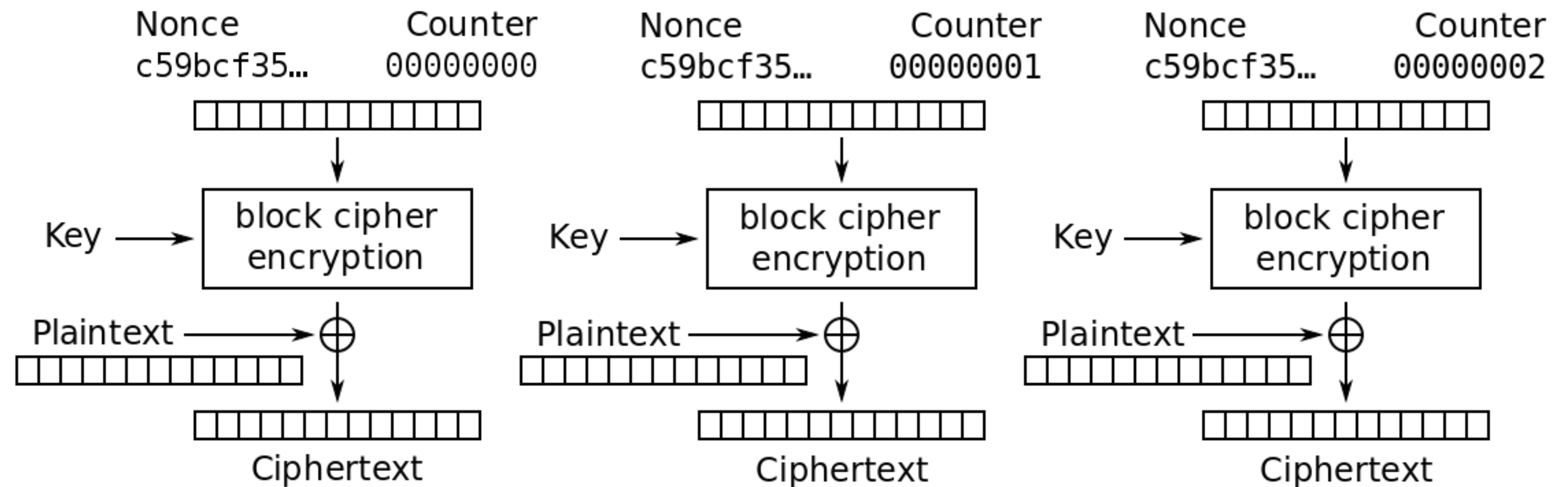
Cifra de Bloco \Rightarrow Cifra Segura

- Cifra de Bloco:

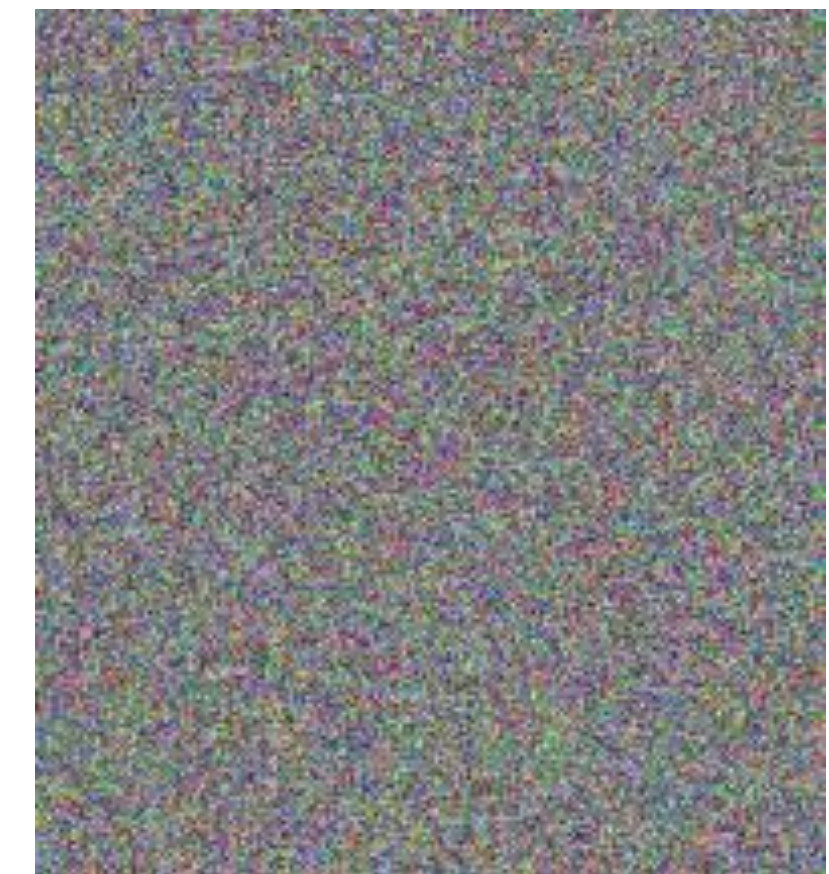
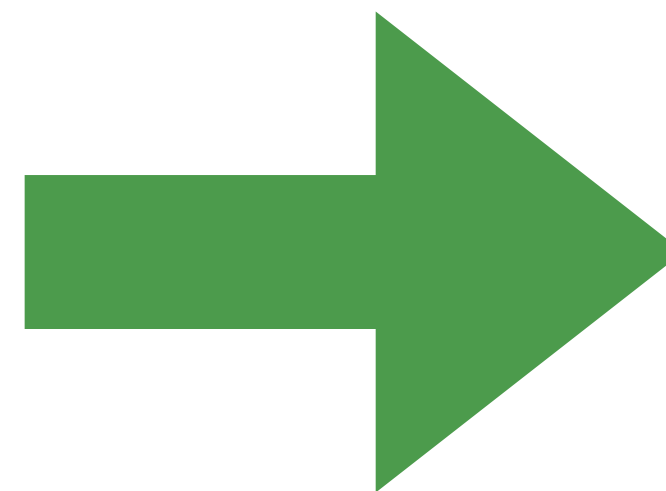
- Bloco controlado por aplicação => bloco pseudo-aleatório
- Como usar esta funcionalidade para cifrar, e.g., um ficheiro?

- Algumas maneiras são seguras:

- **Counter Mode (seguro, mais usado)**
- Porquê? (Estudar criptografia => prova)
- Intuição: cifra sequencial com nonce
 - usa AES para construir PRG
 - contador inicial = nonce



Counter (CTR) mode encryption



Outras cifras simétricas

- Nem todas as cifras simétricas começam no AES
- ChaCha20:
 - cifra sequencial com nonce
 - gerador pseudo-aleatório dedicado
 - estrutura parecida ao counter mode
 - componente central já assume counter e nonce no estado
 - popularidade crescente: eficiência em SW

