

# CSE 484 / CSE M 584: Symmetric Encryption

Fall 2024

Franziska (Franzi) Roesner  
franzi@cs

UW Instruction Team: David Kohlbrenner, Yoshi Kohno, Franziska Roesner. Thanks to Dan Boneh, Dieter Gollmann, Dan Halperin, John Manferdelli, John Mitchell, Vitaly Shmatikov, Bennet Yee, and many others for sample slides and materials ...

# Announcements

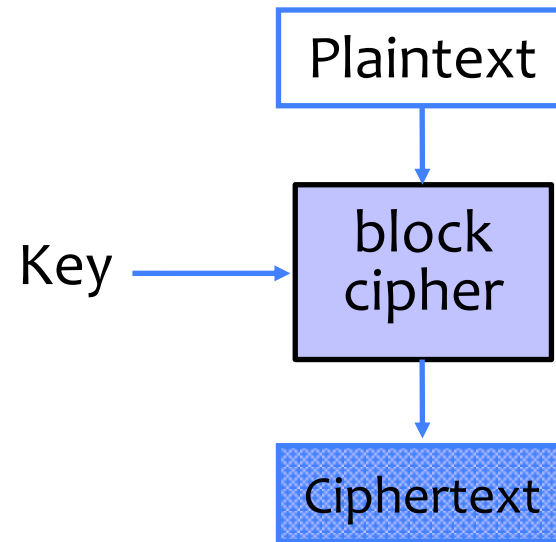
- Lab 1A due **today**, Lab 1B due next Wednesday
- In-class activities:
  - Reminder that you can submit **up to half** of them **up to a week late**
- Homework 2 (crypto) to be released soon
  - Due ~1 week after Lab 1B
  - Designed to give you hands-on experience with crypto concepts, not be tricky

# Reminder: Reducing Key Size

- What to do when it is infeasible to pre-share huge random keys?
  - When one-time pad is unrealistic...
- Use special cryptographic primitives: block ciphers, stream ciphers
  - Single key can be re-used (with some restrictions)
  - Not as theoretically secure as one-time pad

# Block Ciphers

- Operates on a single chunk (“block”) of plaintext
  - For example, 64 bits for DES, 128 bits for AES
  - Each key defines a different **permutation** of inputs to possible outputs
  - Same key is reused for each block (can use short keys)



# Keyed Permutation

Key = 00   Key = 01

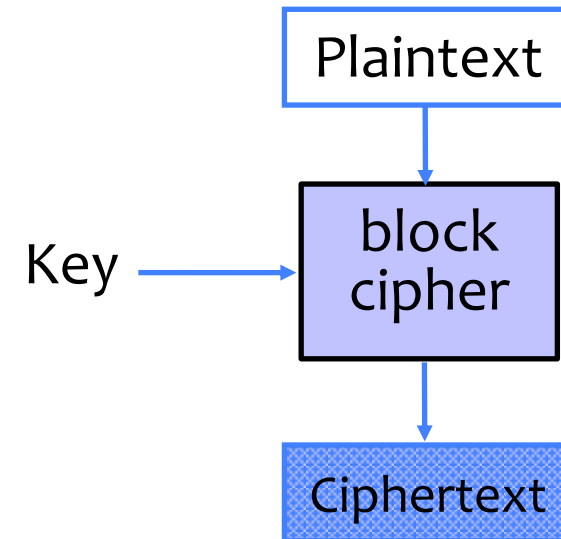
input	possible output	possible output	etc.
000	010	111	...
001	111	110	...
010	101	000	...
011	110	101	...
...	...	...	...
111	000	110	...

For N-bit input:  
 **$2^N!$  possible permutations**

For K-bit key:  
 **$2^K$  possible keys**

# Keyed Permutation

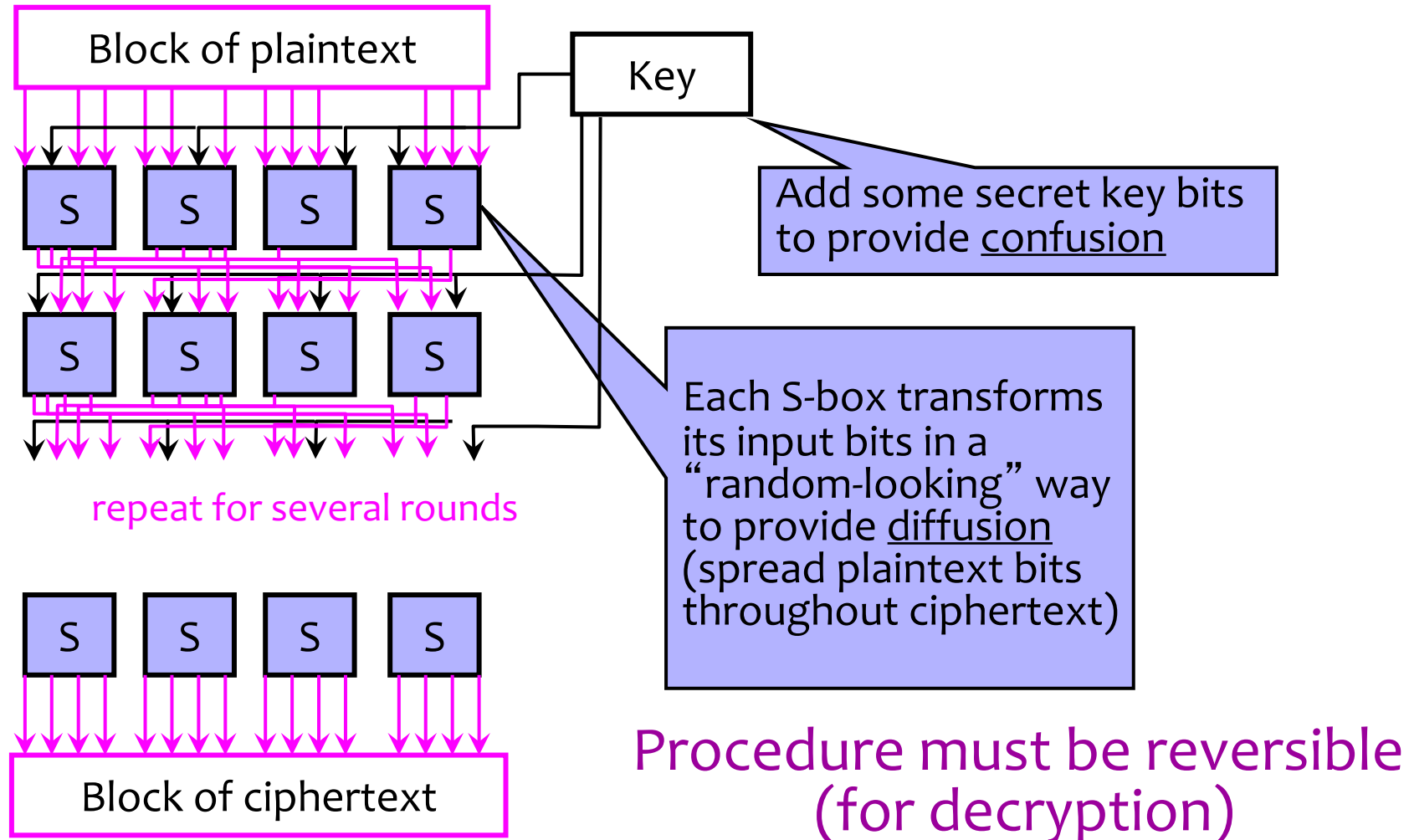
- Not just shuffling of input bits!
  - Suppose plaintext = “111”.
  - Then “111” is not the only possible ciphertext!
- Instead:
  - **Permutation of possible outputs**
  - Use secret key to pick a permutation



# Block Cipher Security

- Result should “look like” a random permutation on the inputs
  - N-bit block cipher permutes over  $2^N$  inputs.
- Only computational guarantee of secrecy
  - Not impossible to break, just very expensive
    - If there is no efficient algorithm (unproven assumption!), then can only break by brute-force, try-every-possible-key search
  - Time and cost of breaking the cipher exceed the value and/or useful lifetime of protected information

# Block Cipher Operation (Simplified)





# Standard Block Ciphers

- **DES: Data Encryption Standard**
  - Feistel structure: builds invertible function using non-invertible ones
  - Invented by IBM, issued as federal standard in 1977
  - 64-bit blocks, 56-bit key + 8 bits for parity

# DES and 56 bit keys

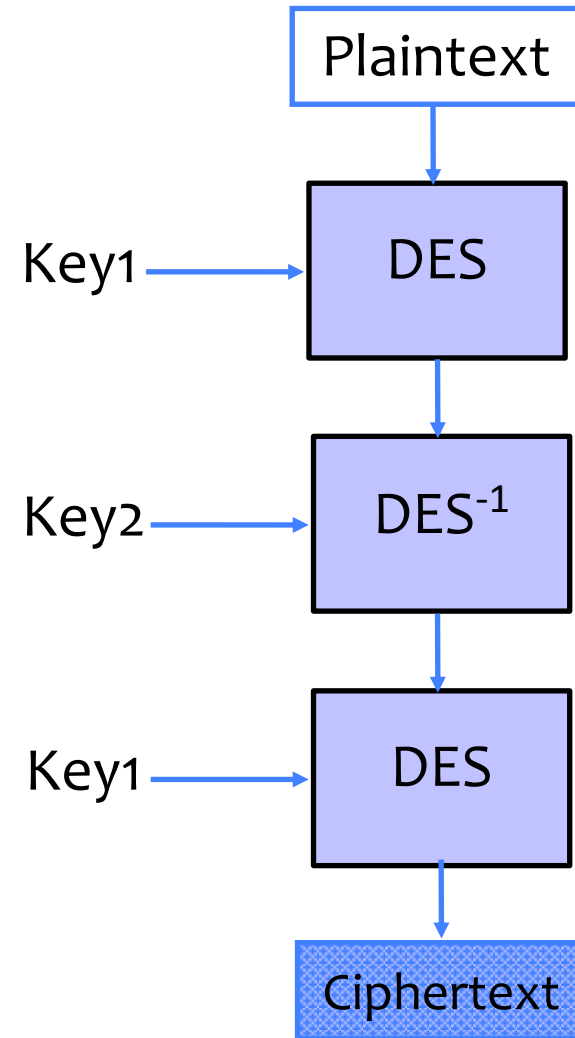
- 56 bit keys are quite short

Key Size (bits)	Number of Alternative Keys	Time required at 1 encryption/ $\mu$ s	Time required at $10^6$ encryptions/ $\mu$ s
32	$2^{32} = 4.3 \times 10^9$	$2^{31} \mu\text{s} = 35.8$ minutes	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	$2^{55} \mu\text{s} = 1142$ years	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	$2^{127} \mu\text{s} = 5.4 \times 10^{24}$ years	$5.4 \times 10^{18}$ years
168	$2^{168} = 3.7 \times 10^{50}$	$2^{167} \mu\text{s} = 5.9 \times 10^{36}$ years	$5.9 \times 10^{30}$ years
26 characters (permutation)	$26! = 4 \times 10^{26}$	$2 \times 10^{26} \mu\text{s} = 6.4 \times 10^{12}$ years	$6.4 \times 10^6$ years

- 1999: EFF DES Crack + distributed machines
  - < 24 hours to find DES key
- DES  $\rightarrow$  3DES
  - 3DES: DES + inverse DES + DES (with 2 or 3 diff keys)

# 3DES

- Two-key 3DES increases security of DES by doubling the key length
- Why not 2DES?
  - Minimal gain in security compared to DES due to **meet-in-the-middle attack** (*more in section on this*)

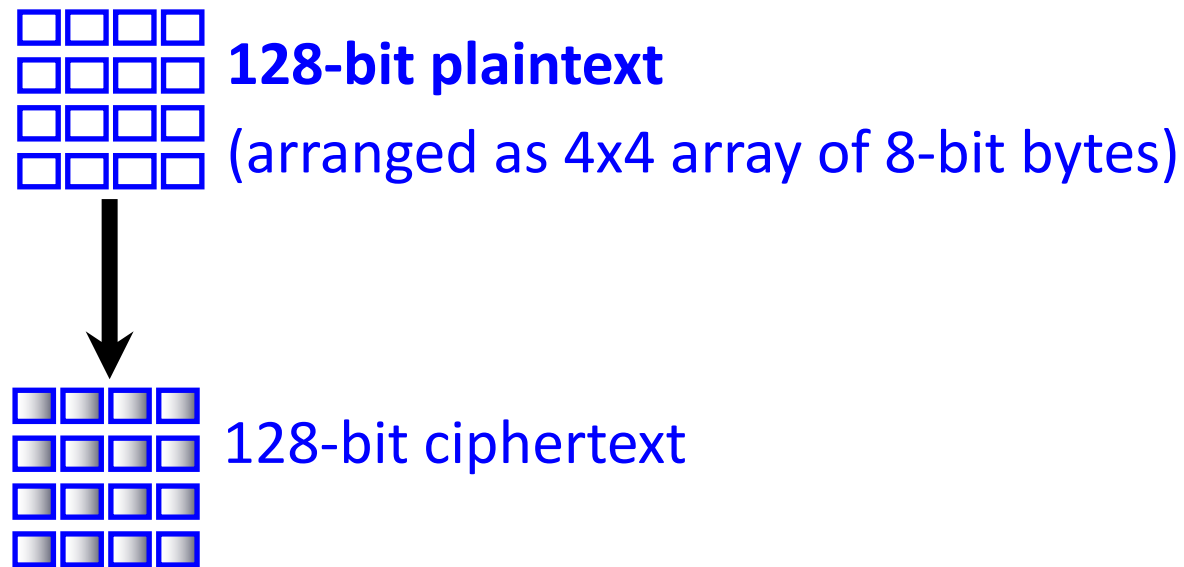


# Standard Block Ciphers

- **DES: Data Encryption Standard**
  - Feistel structure: builds invertible function using non-invertible ones
  - Invented by IBM, issued as federal standard in 1977
  - 64-bit blocks, 56-bit key + 8 bits for parity
- **AES: Advanced Encryption Standard**
  - New federal standard as of 2001
    - NIST: National Institute of Standards & Technology
  - Based on the Rijndael algorithm
    - Selected via an open process
  - 128-bit blocks, keys can be 128, 192 or 256 bits

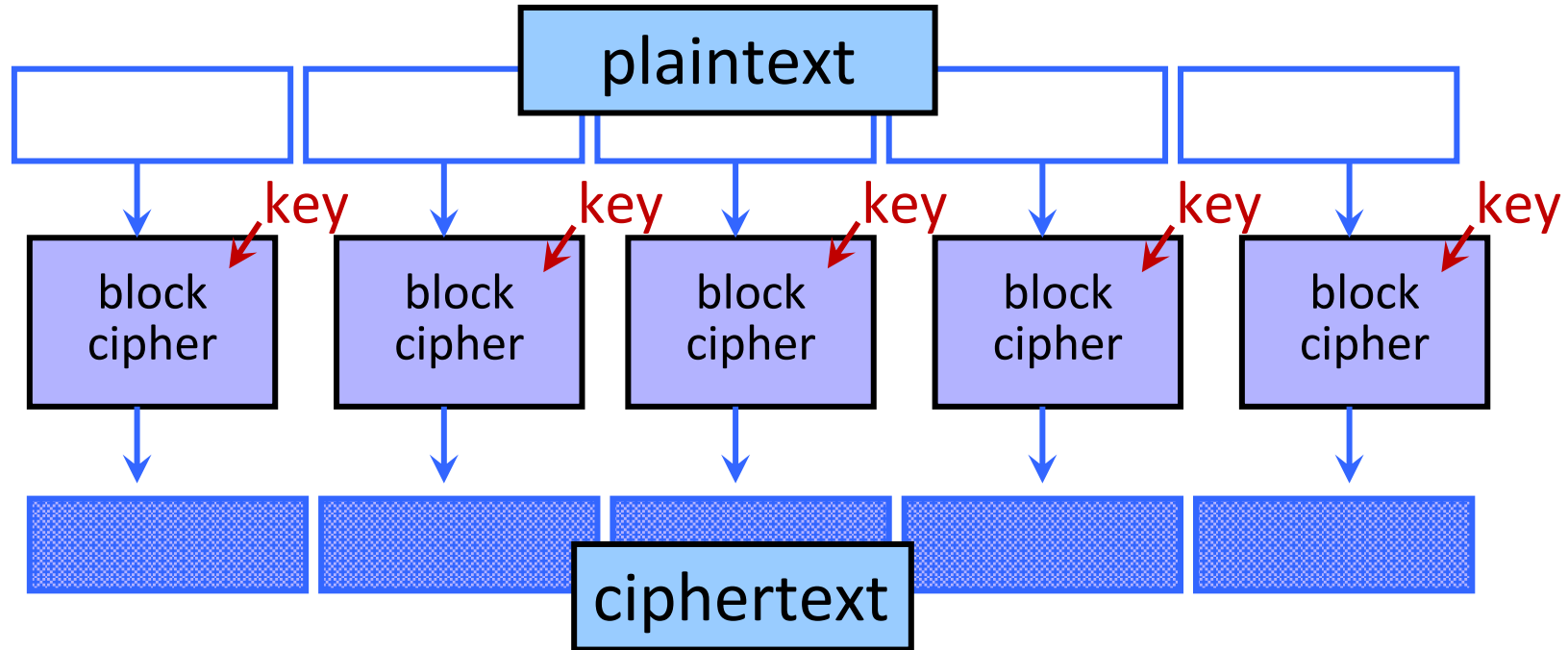
# Encrypting a Large Message

- So, we've got a good block cipher, but our plaintext is larger than 128-bit block size



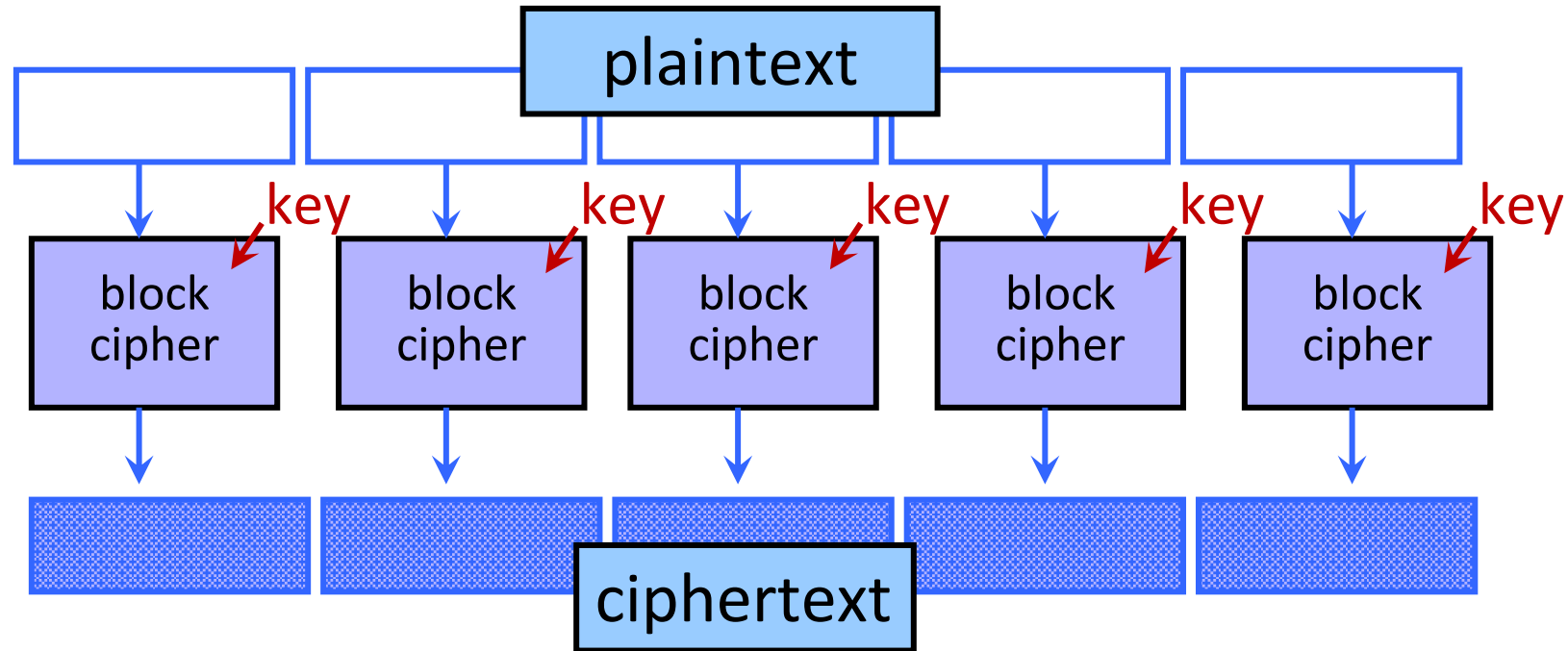
- What should we do?

# Electronic Code Book (ECB) Mode



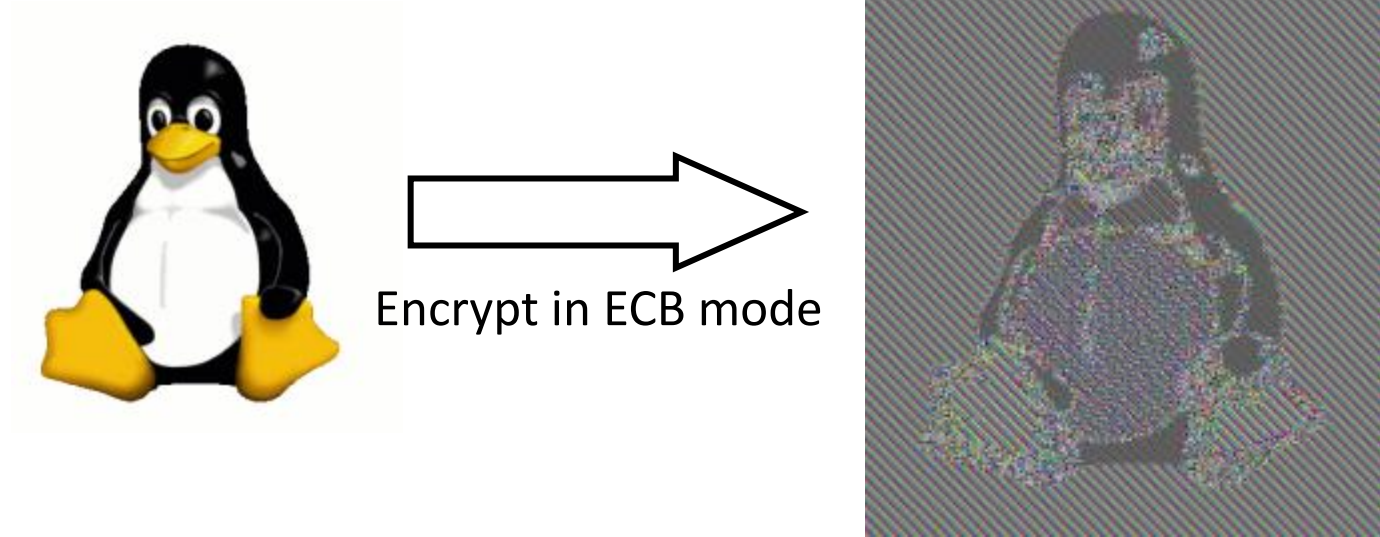
Canvas “quiz” time!

# Electronic Code Book (ECB) Mode



- Identical blocks of plaintext produce identical blocks of ciphertext
- No integrity checks: can mix and match blocks

# Information Leakage in ECB Mode



[Wikipedia]



# Oops

## Move Fast and Roll Your Own Crypto

### A Quick Look at the Confidentiality of Zoom Meetings

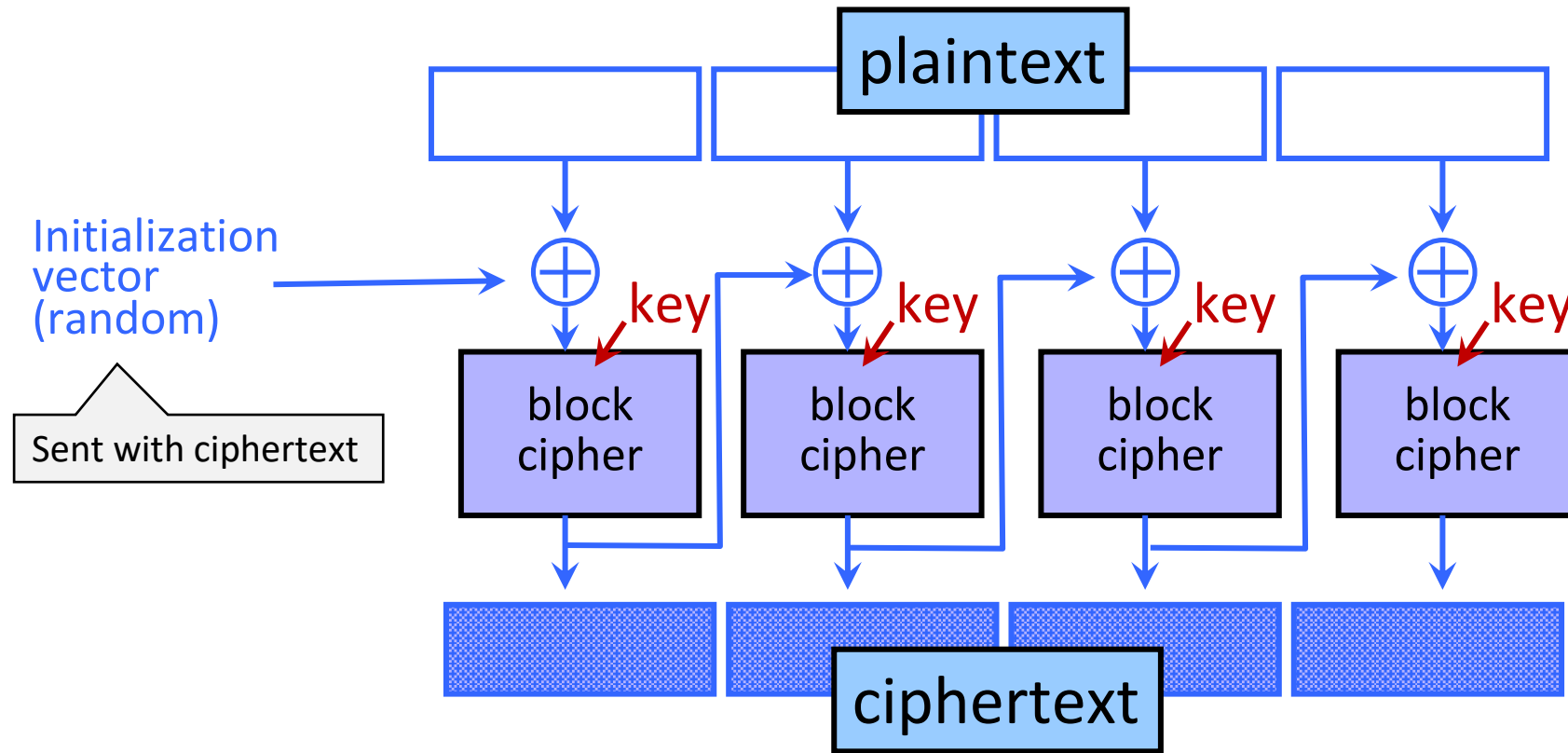
By Bill Marczak and John Scott-Railton

April 3, 2020

- 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.

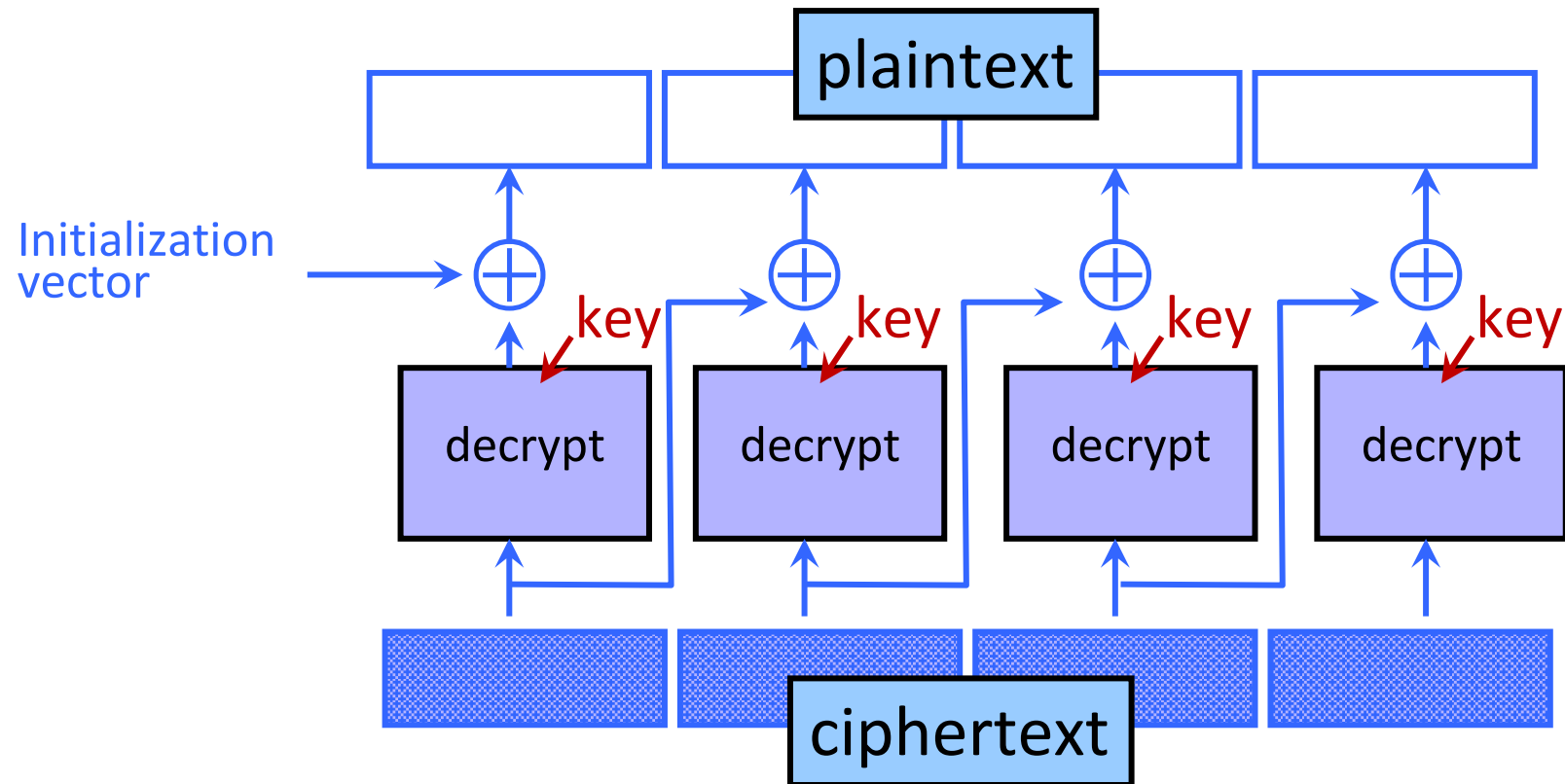
<https://citizenlab.ca/2020/04/move-fast-roll-your-own-crypto-a-quick-look-at-the-confidentiality-of-zoom-meetings/>

# Cipher Block Chaining (CBC) Mode: Encryption

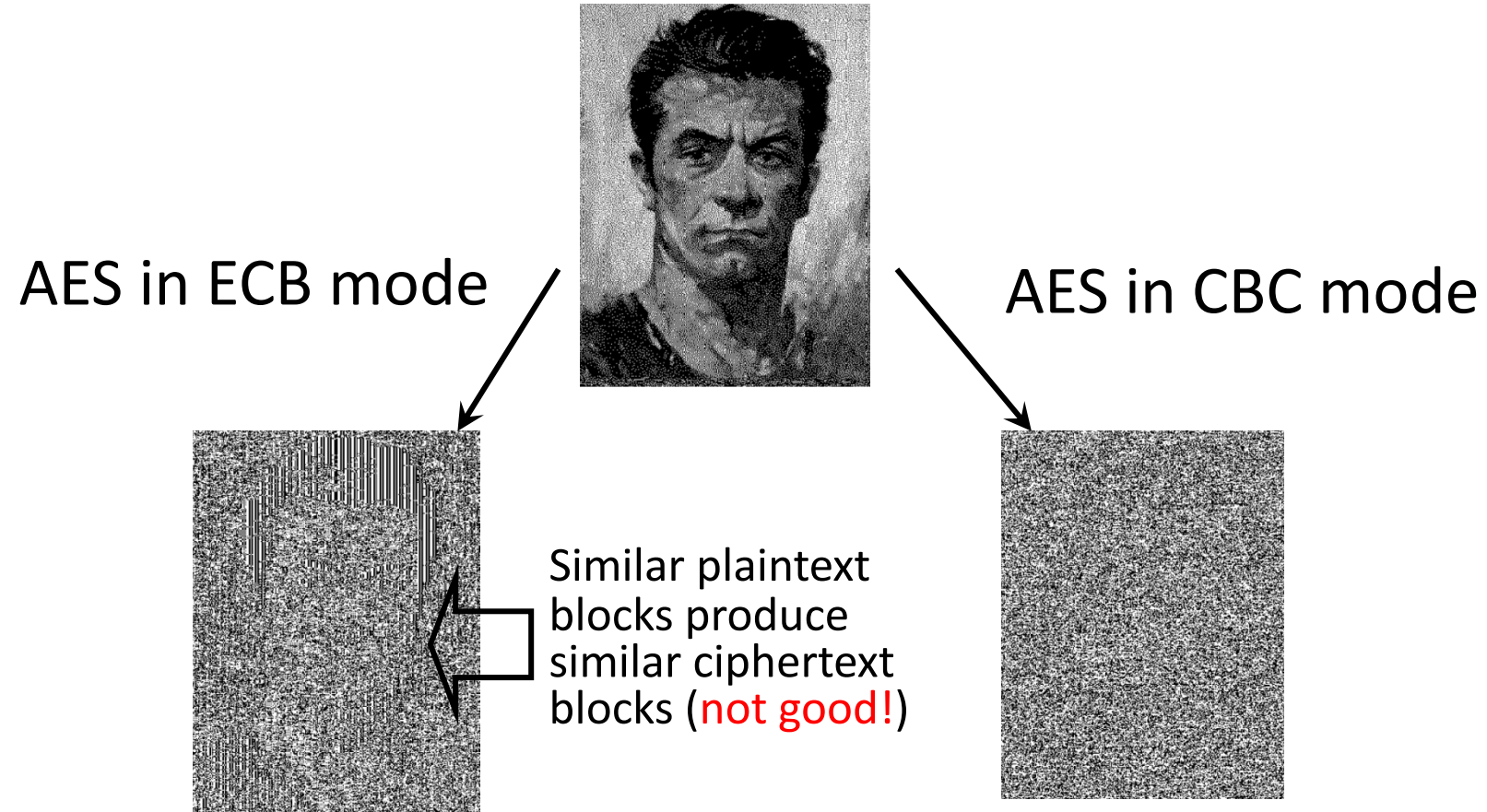


- Identical blocks of plaintext encrypted differently
- Last cipherblock depends on entire plaintext
  - Still does not guarantee integrity

# CBC Mode: Decryption

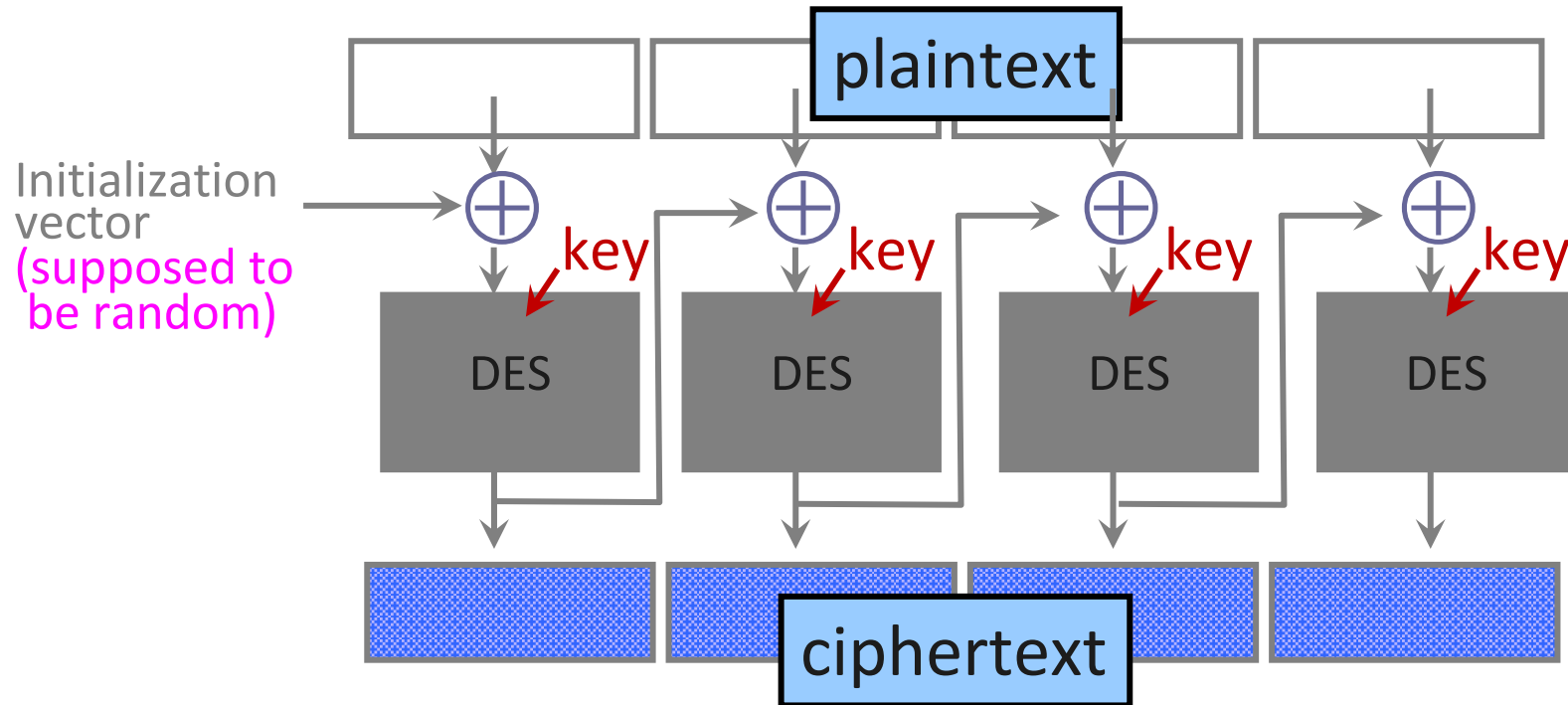


# ECB vs. CBC



[Picture due to Bart Preneel]

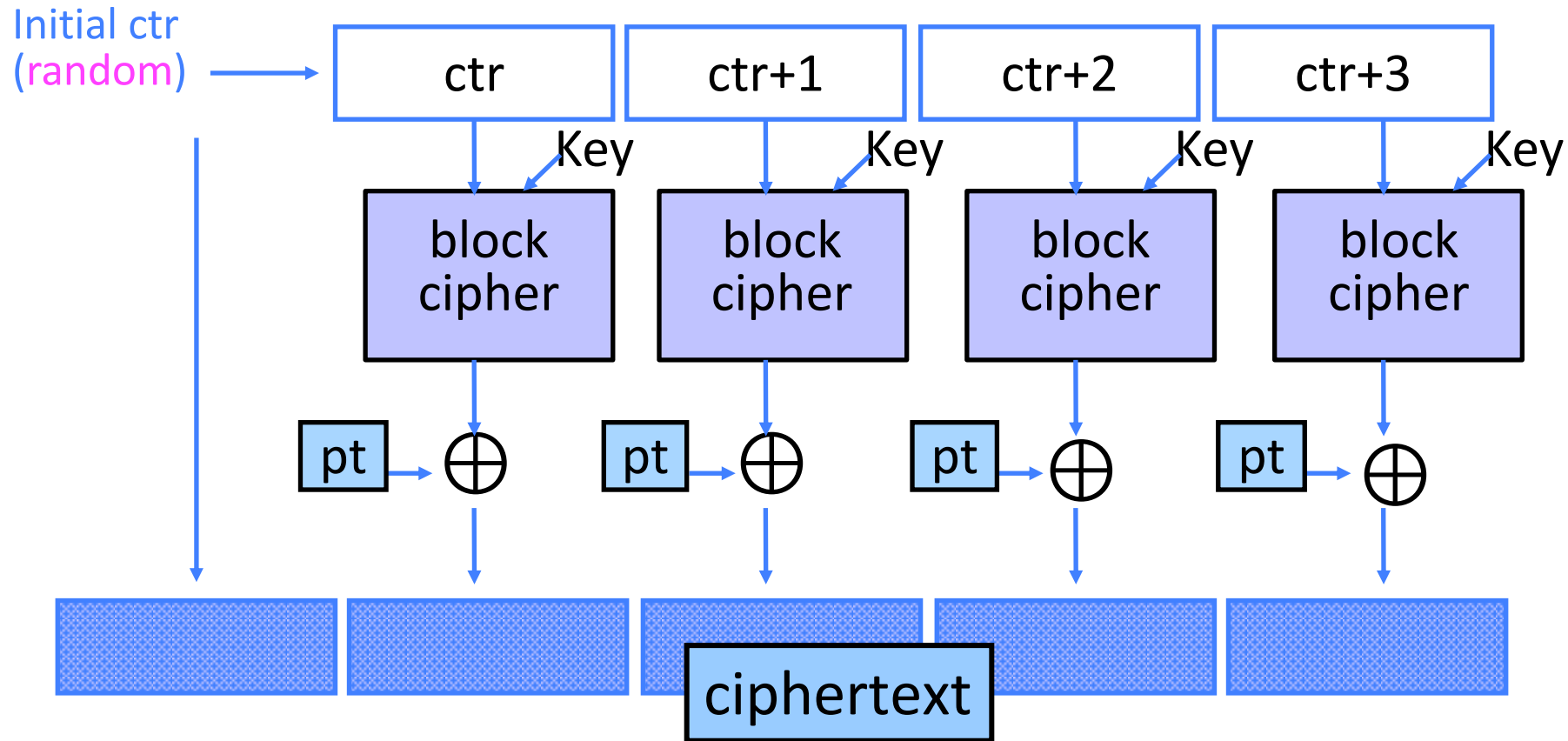
# Initialization Vector Dangers



Found in the source code for Diebold voting machines:

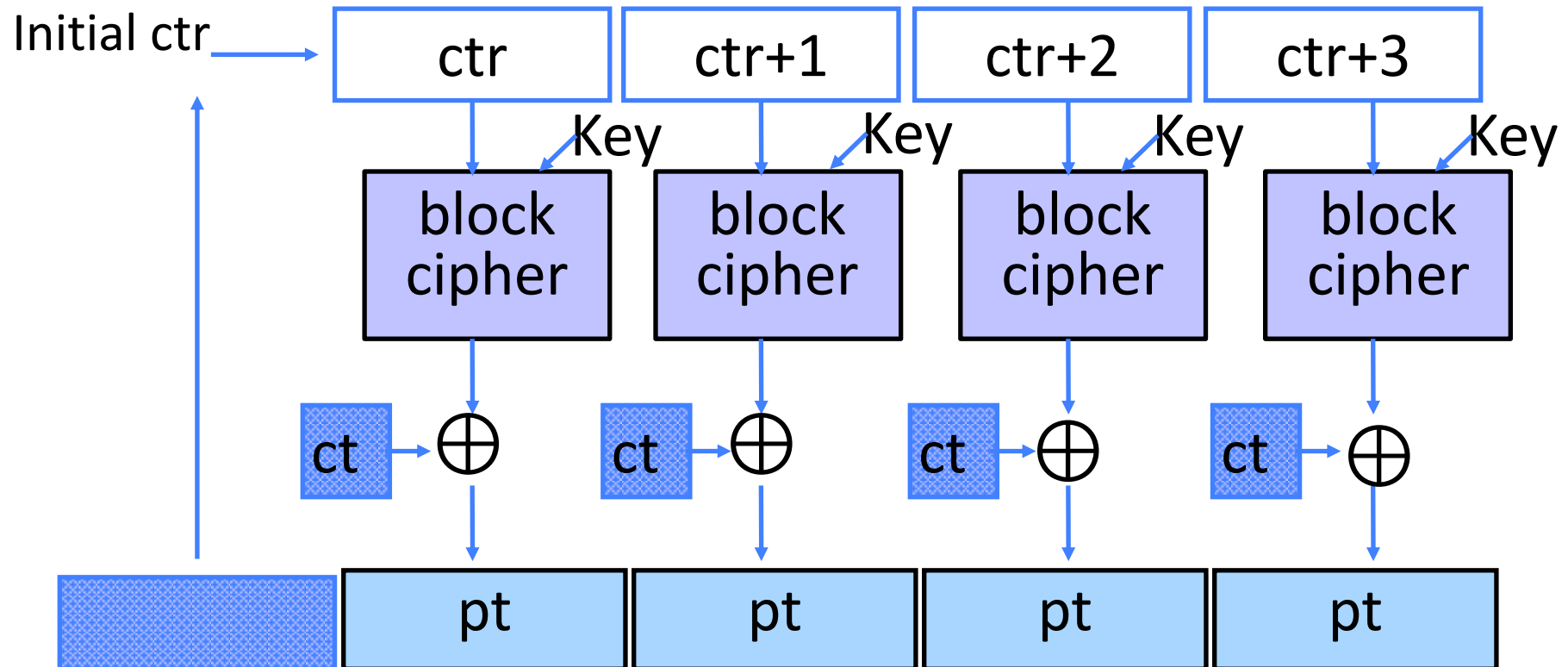
```
DesCBCEncrypt((des_c_block*)tmp, (des_c_block*)record.m_Data,  
totalSize, DESKEY, NULL, DES_ENCRYPT)
```

# Counter Mode (CTR): Encryption



- Identical blocks of plaintext encrypted differently
- Still does not guarantee integrity; Fragile if ctr repeats

# Counter Mode (CTR): Decryption



# When is an Encryption Scheme “Secure”?

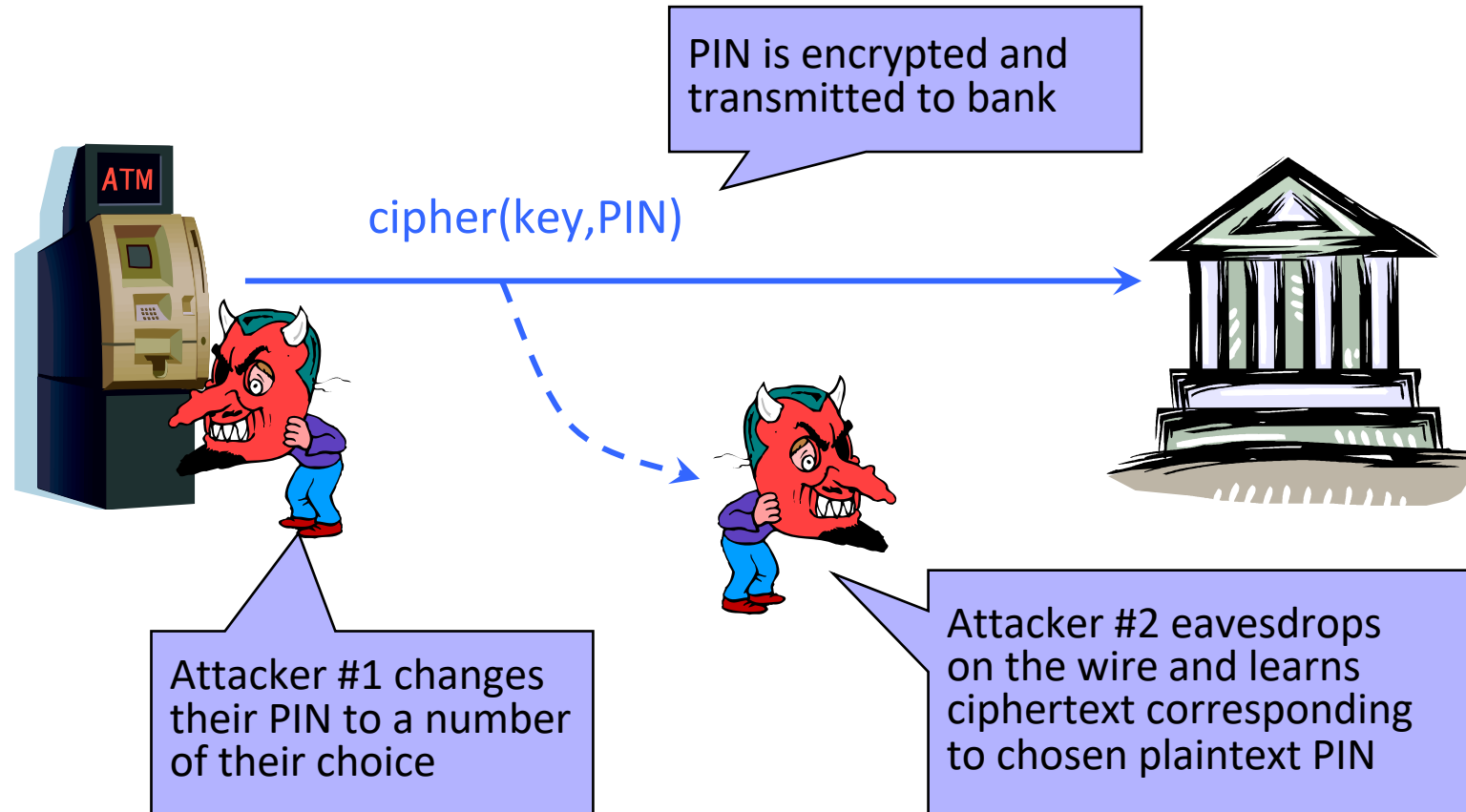
- Hard to recover the key?
  - What if attacker can learn plaintext without learning the key?
- Hard to recover plaintext from ciphertext?
  - What if attacker learns some bits or some function of bits?



# How Can a Cipher Be Attacked?

- Attackers knows ciphertext and encryption algorithm
  - **What else does the attacker know?** Depends on the application in which the cipher is used!
- **Ciphertext-only attack**
- **KPA: Known-plaintext attack** (stronger)
  - Knows some plaintext-ciphertext pairs
- **CPA: Chosen-plaintext attack** (even stronger)
  - Can obtain ciphertext for any plaintext of his choice

# Chosen Plaintext Attack



... repeat for any PIN value

# How Can a Cipher Be Attacked?

- Attackers knows ciphertext and encryption algorithm
  - **What else does the attacker know?** Depends on the application in which the cipher is used!
- Ciphertext-only attack
- **KPA: Known-plaintext attack** (stronger)
  - Knows some plaintext-ciphertext pairs
- **CPA: Chosen-plaintext attack** (even stronger)
  - Can obtain ciphertext for any plaintext of his choice
- **CCA: Chosen-ciphertext attack** (very strong)
  - Can decrypt any ciphertext except the target

# Very Informal Intuition

Minimum security requirement for a modern encryption scheme

- Security against chosen-plaintext attack (CPA)
  - Ciphertext leaks no information about the plaintext
  - Even if the attacker correctly guesses the plaintext, they cannot verify their guess
  - Every ciphertext is unique, encrypting same message twice produces completely different ciphertexts
    - Implication: encryption must be randomized or stateful

# The Shape of the Formal Approach

- INDistinguishability under Chosen Plaintext Attack
  - “IND-CPA”
- Formalized cryptographic game
  - Adversary submits pairs of plaintexts ( $M_a, M_b$ )
  - Gets back ONE of the ciphertexts ( $C_x$ )
  - Adversary must **guess** which ciphertext this is ( $C_a$  or  $C_b$ )
  - If they can do better than 50/50, they win

# Very Informal Intuition

Minimum security requirement for a modern encryption scheme

- Security against chosen-plaintext attack (CPA)
  - Ciphertext leaks no information about the plaintext
  - Even if the attacker correctly guesses the plaintext, they cannot verify their guess
  - Every ciphertext is unique, encrypting same message twice produces completely different ciphertexts
    - Implication: encryption must be randomized or stateful
- Security against chosen-ciphertext attack (CCA)
  - Integrity protection – it is not possible to change the plaintext by modifying the ciphertext