CSE 484 / CSE M 584: Cryptography

Winter 2022

Tadayoshi (Yoshi) Kohno
yoshi@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 ...

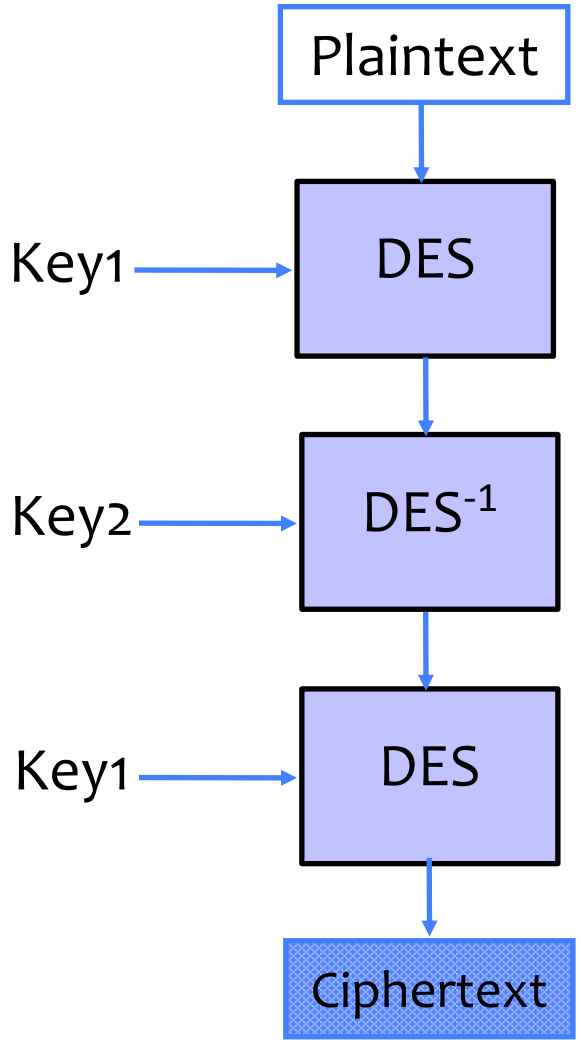
Announcements

- Office Hours this week: Virtual
- HW 2 out soon
- Feb 14: Alex Gantman (Qualcomm) – entirely virtual lecture
- Feb 16: Lucy Simko (UW) – entirely virtual lecture
- Feb 17: Final Project Part #1 deadline
  - https://courses.cs.washington.edu/courses/cse484/22wi/assignments/final_project.html
  - Groups strongly encouraged; groups of up to 3 people
  - 12–15-minute video, at non-accelerated or non-slowed timing (this is a security class, after all :P)
  - Use guest lectures, homework 1, reading of news, personal interests, and any other resource you wish for inspiration

Begin Review

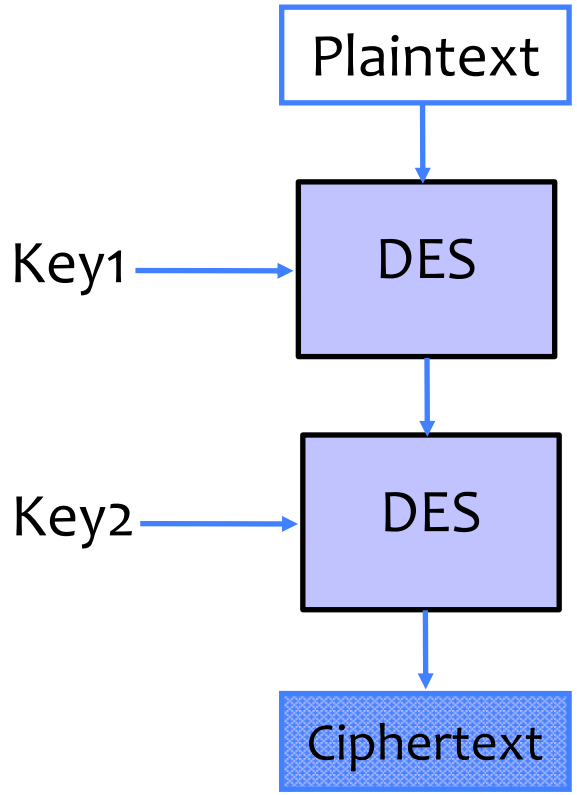
3DES

- Two-key 3DES increases security of DES by doubling the key length



But wait… what about *2DES*?

- Suppose you are given plaintext-ciphertext pairs (P1,C1), (P2,C2), (P3,C3)
- Suppose Key1 and Key2 are each 56-bits long
- Can you figure out Key1 and Key2 if you try all possible values for both (2^{112} possibilities) → Yes
- Can you figure out Key1 and Key2 more than that? → Breakout

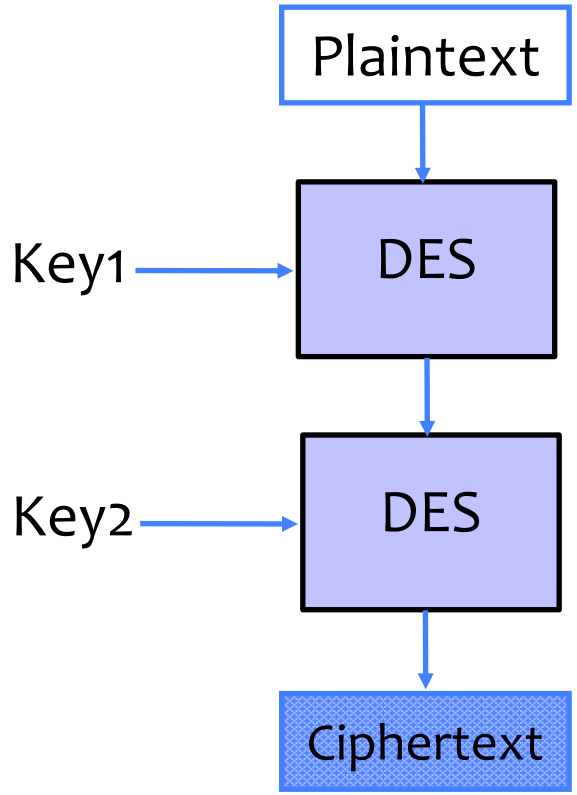


But wait… what about *2DES*?

- Meet-in-the-middle attack

Meet-in-the-Middle Attack

- Guess 2^{56} values for Key1, and create a table from P1 to a middle value M1 for each key guess ($M1^{G1}$, $M1^{G2}$, $M1^{G3}$, …)
- Guess 2^{56} values for Key2, and create a table from C1 to a middle value M'1 for each key guess ($M'1^{G1}$, $M'1^{G2}$, $M'1^{G3}$, …)
- Look for collision in the middle values → if only one collision, found Key1 and Key2; otherwise repeat for (P2,C2), …



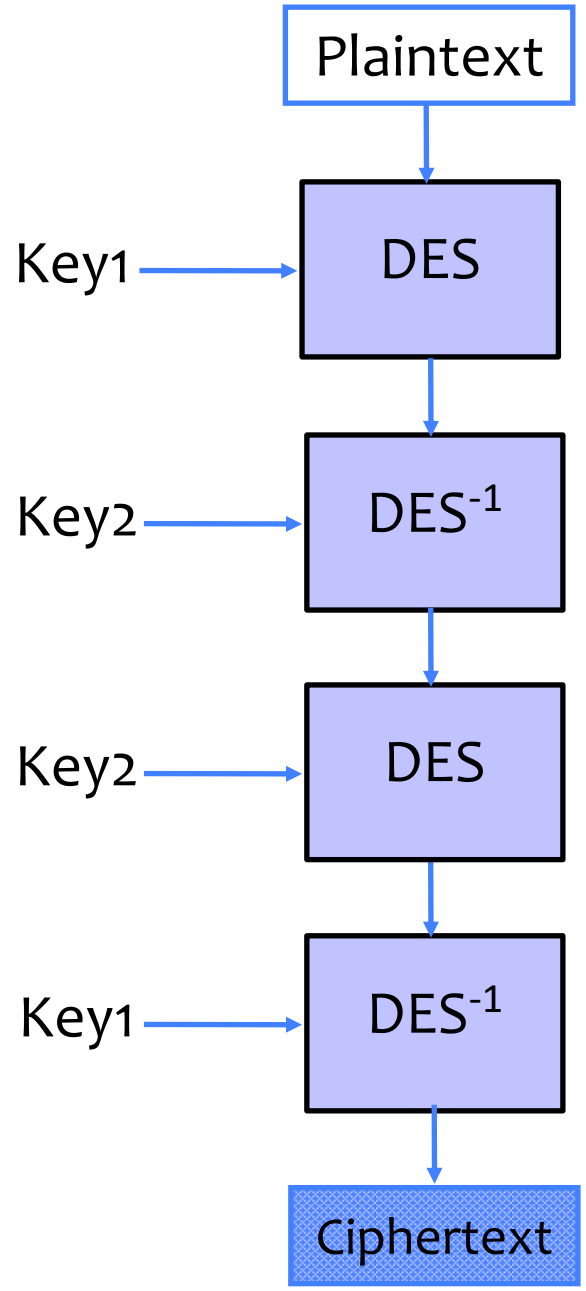
Defining the strength of a scheme

- *Effective Key Strength*
  - Amount of 'work' the adversary needs to do
- DES: 56-bits
  - 2^56 encryptions to try 'all keys'
- 2DES: 57-bits
  - 2*(2^56) encryptions = 2^57
- 3DES: 112-bits (or sometimes 80-bits)
  - Meet-in-the-middle + more work = 2^112 (for 3 keys, e.g. K1, K2, K3)
  - Various attacks = 2^80 (for 2 keys, e.g. K1, K2, K1)

End Review

4DES

- Paul Kocher's *JOKE* proposal
- If two-key 3DES is good, would two-key 4DES be even better?



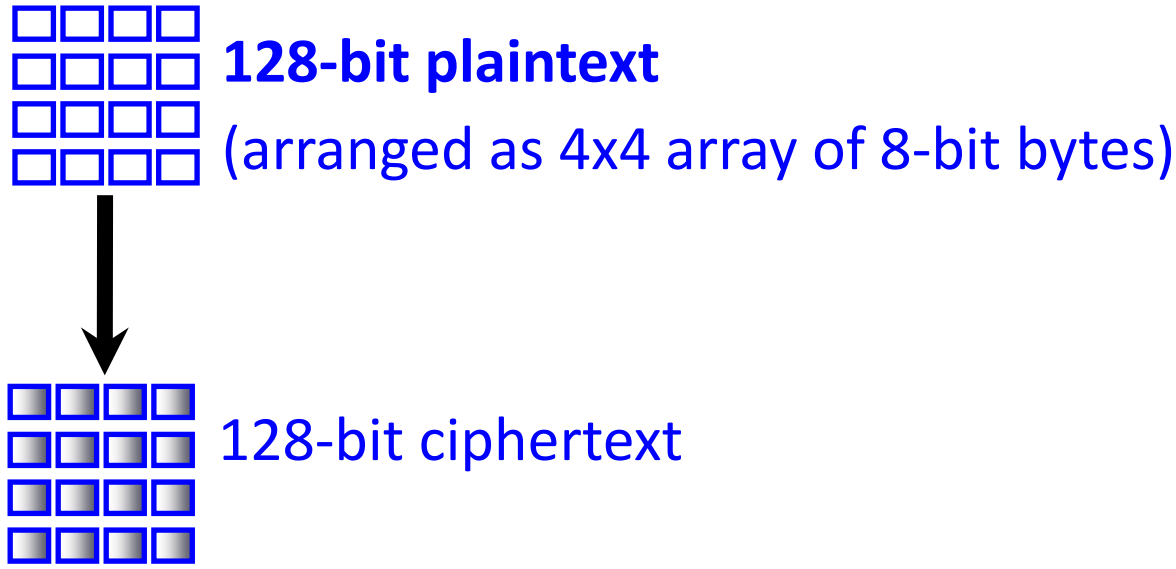
Standard Block Ciphers

- **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**
  - 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

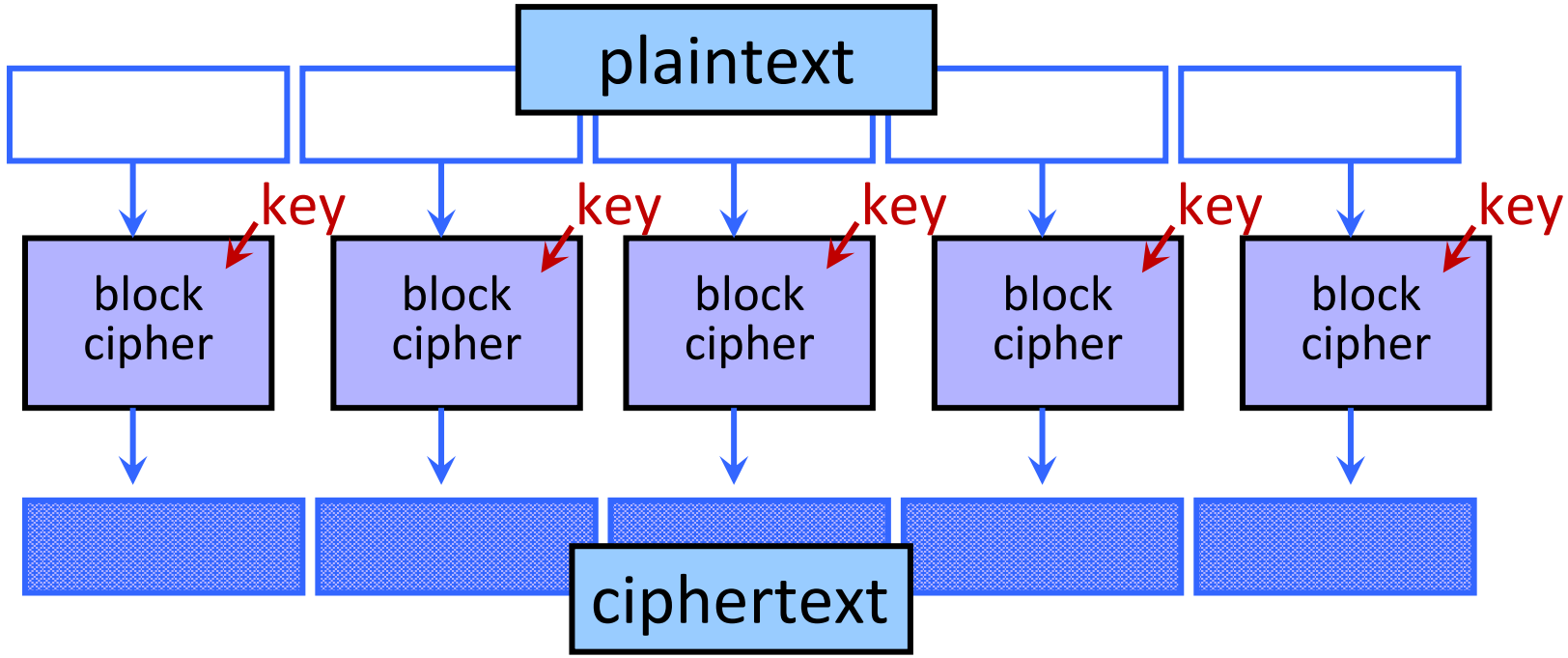
Encrypting a Large Message

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



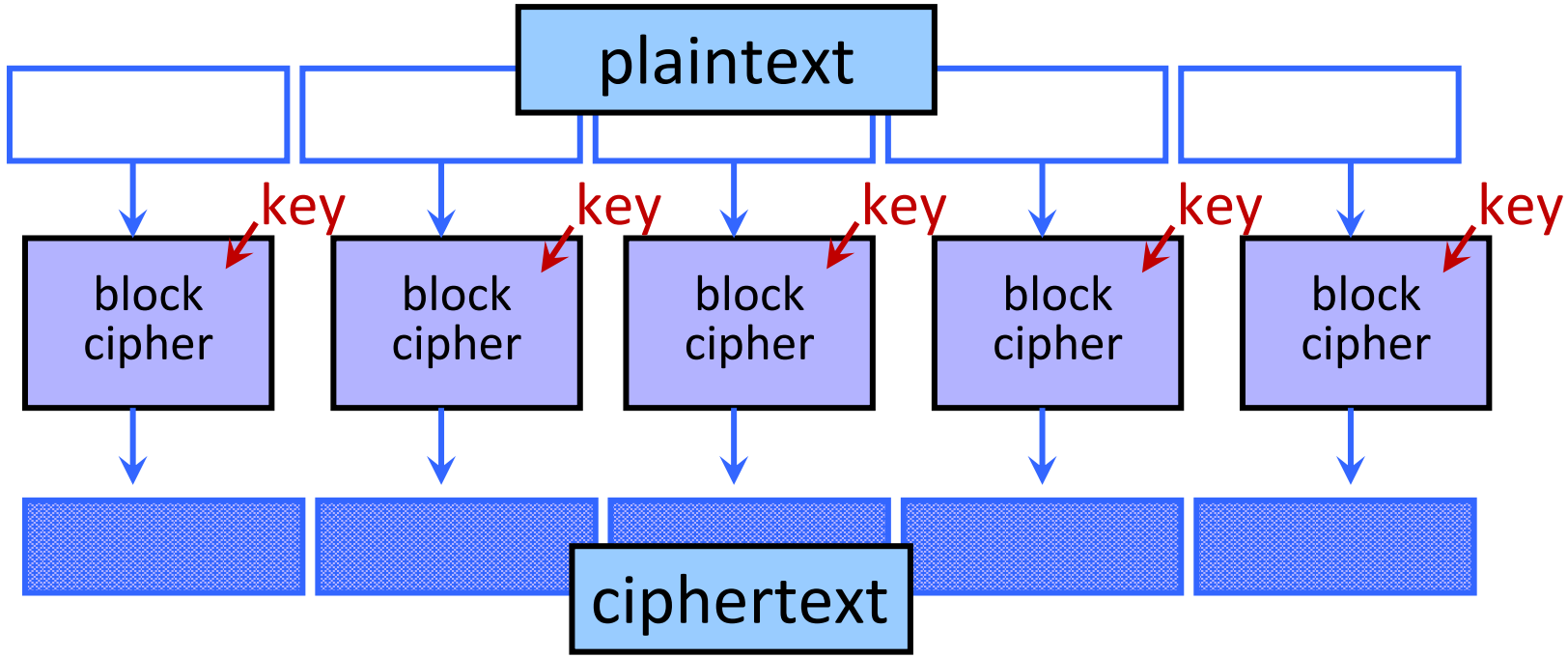
- What should we do?

Electronic Code Book (ECB) Mode



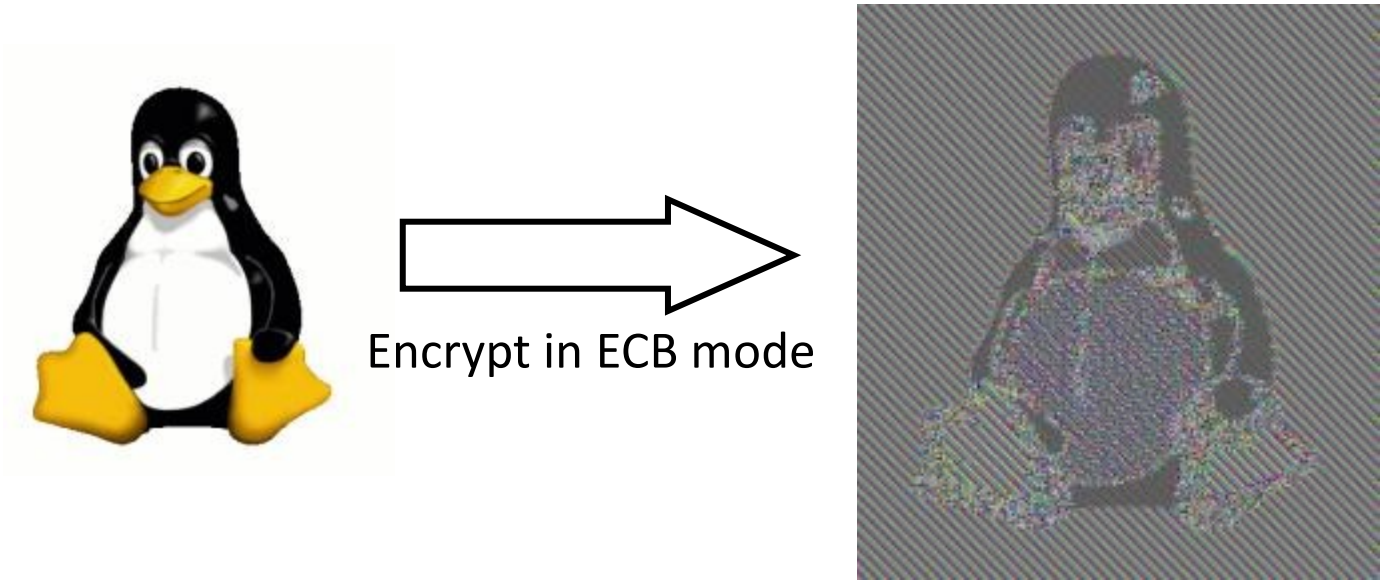
Canvas time!

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]

Zoom…

Move Fast and Roll Your Own Crypto

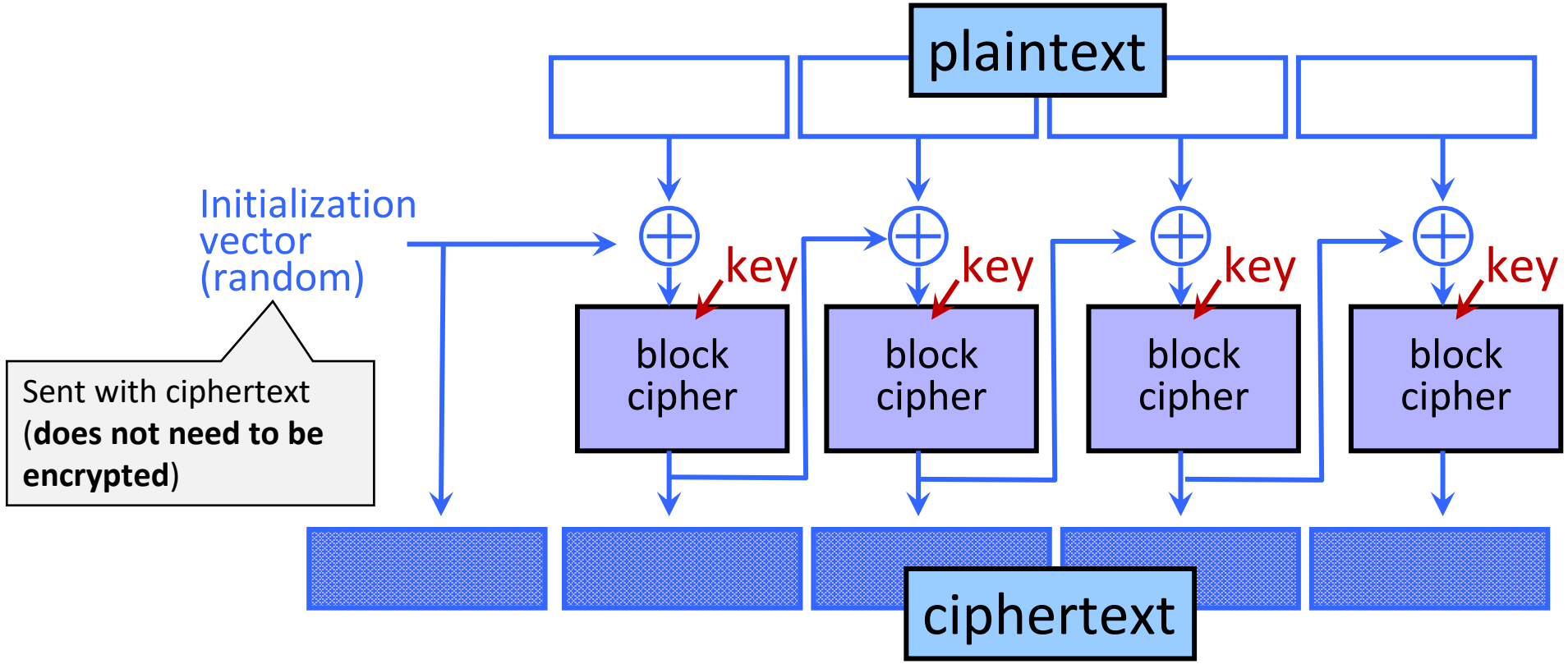
A Quick Look at the Confidentiality of Zoom Meetings

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.

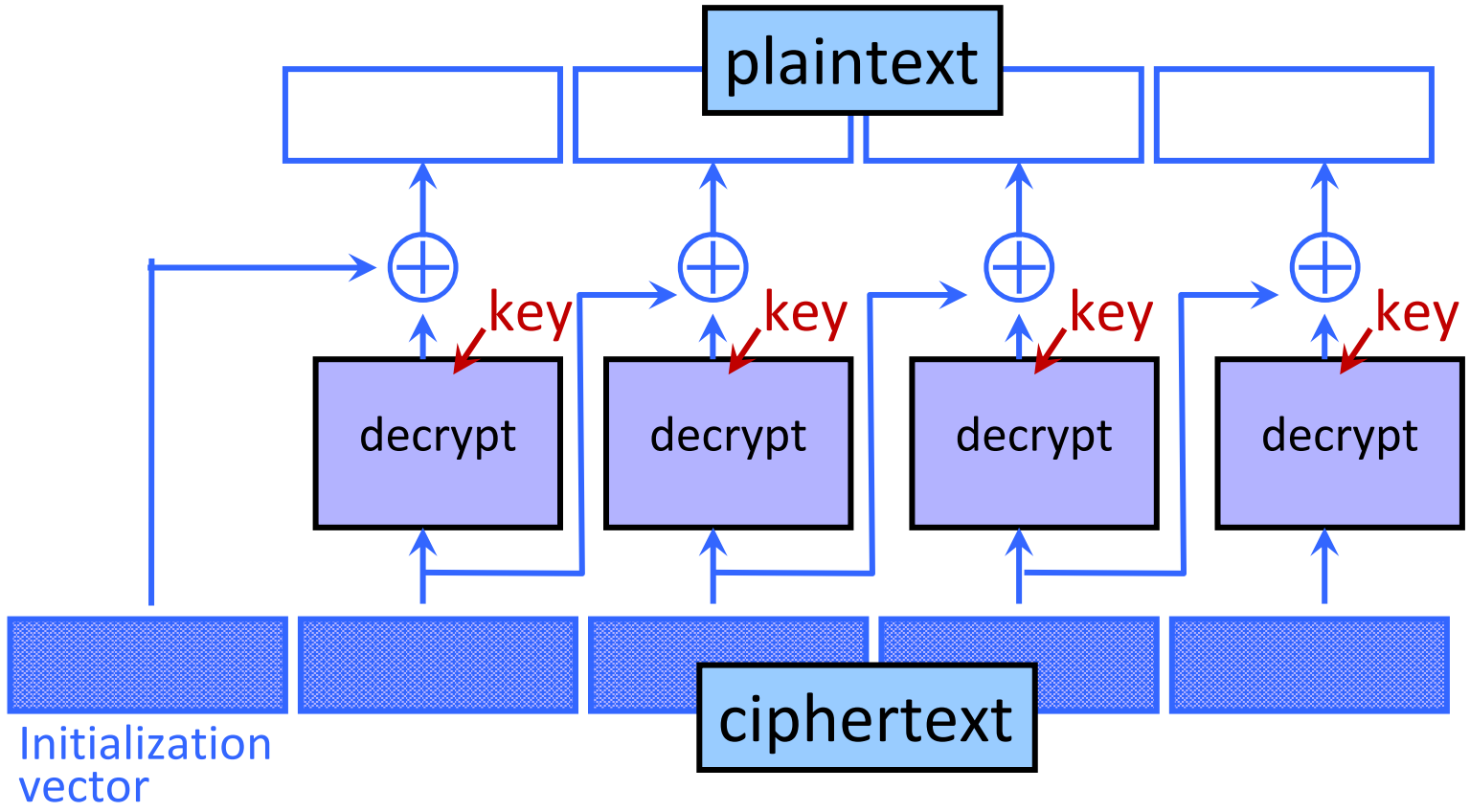
https://citizenlab.ca/2020/04/move-fast-roll-your-own-crypto-a-quick-look-at-the-confidentiality-of-zoom-meetings/

Cipher Block Chaining (CBC) Mode: Encryption

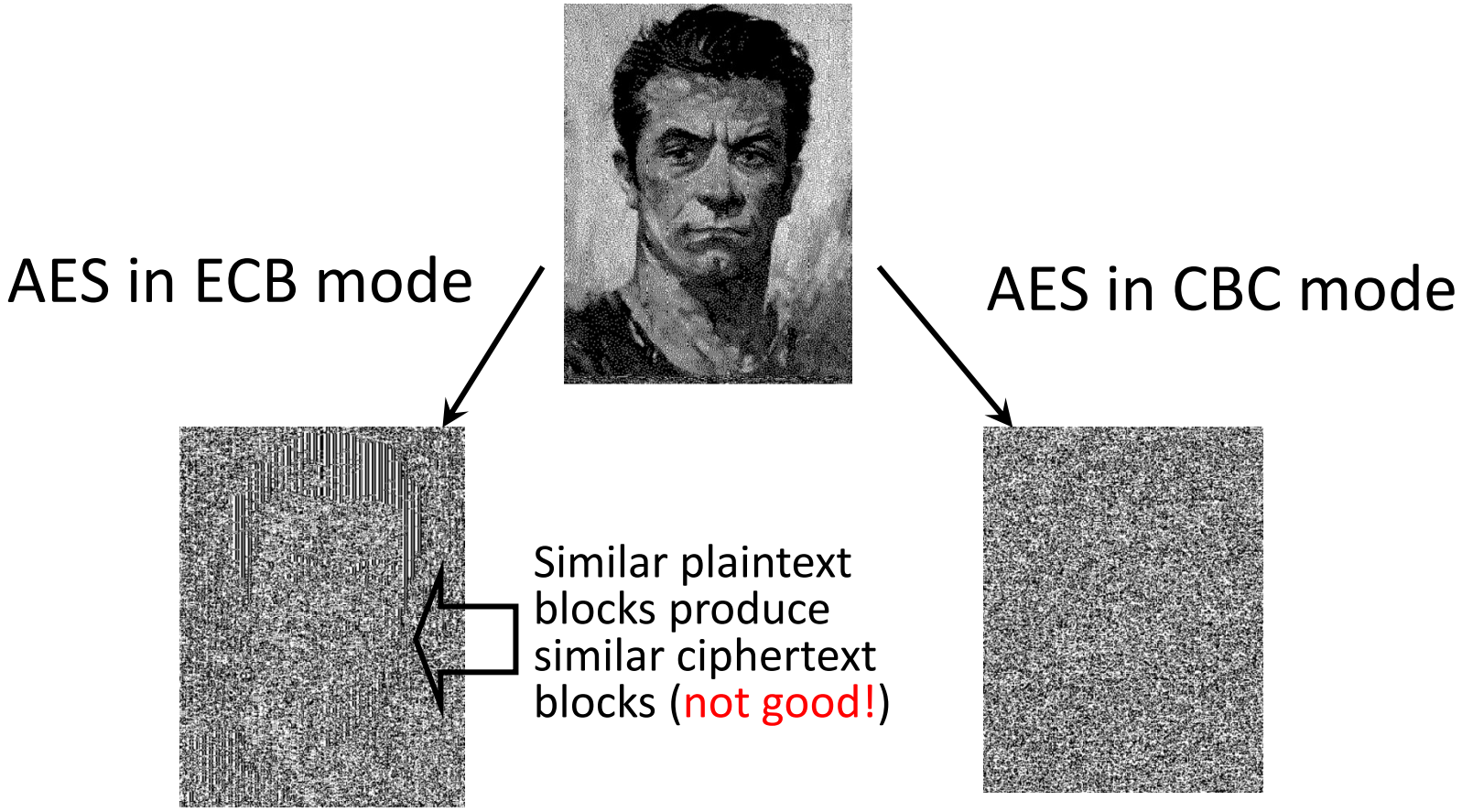


- Identical blocks of plaintext encrypted differently
- Last cipherblock depends on entire plaintext
  - Still does not guarantee integrity

CBC Mode: Decryption

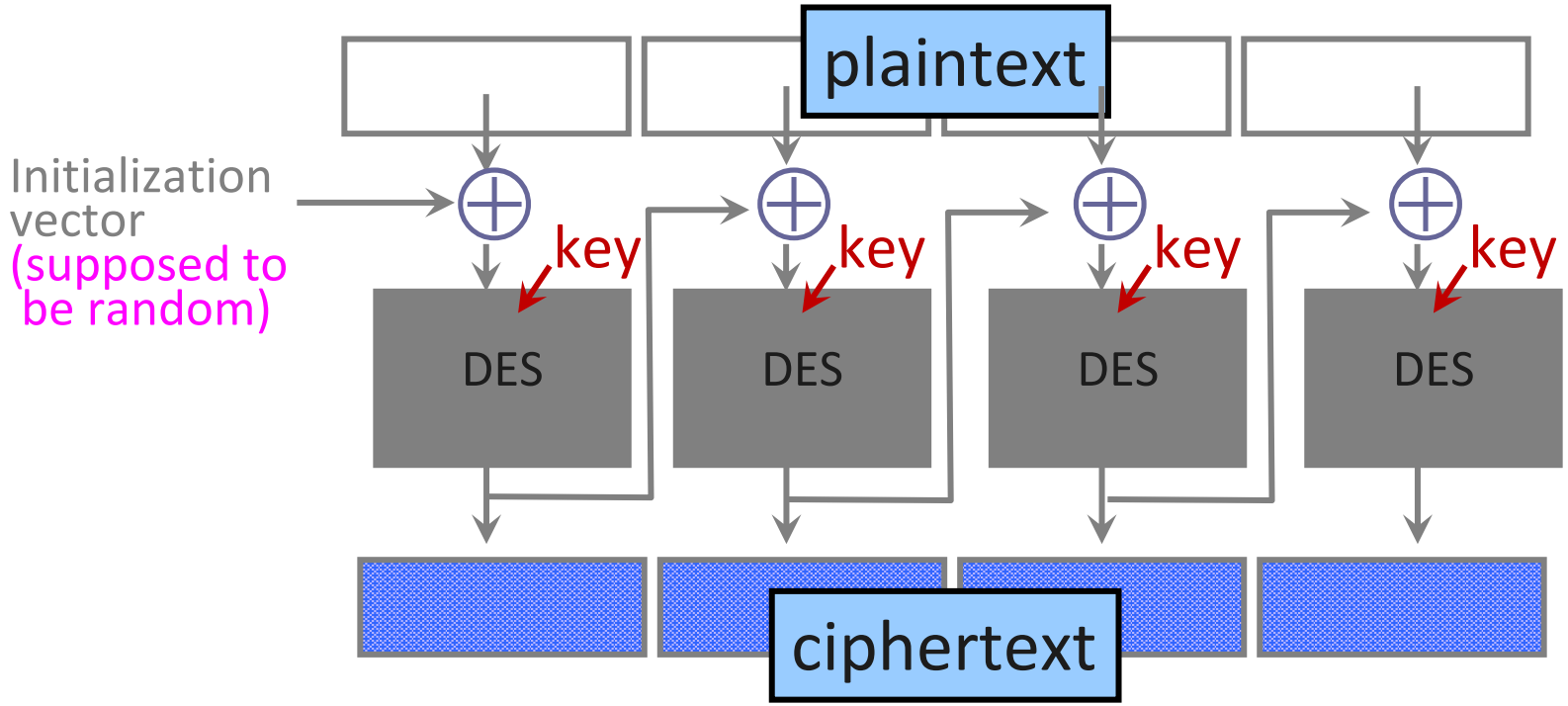


ECB vs. CBC



[Picture due to Bart Preneel]

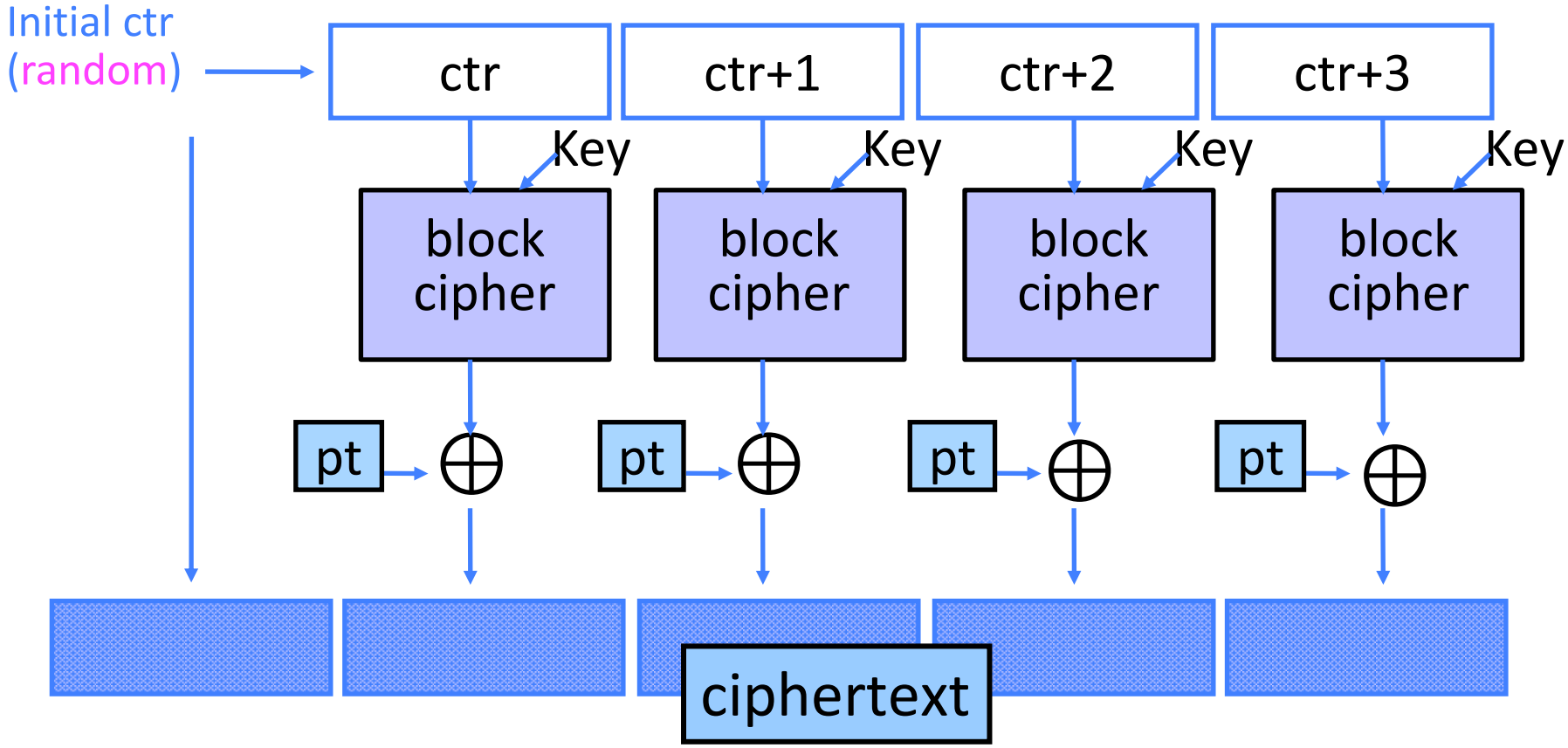
Initialization Vector Dangers



Found in the source code for Diebold voting machines:

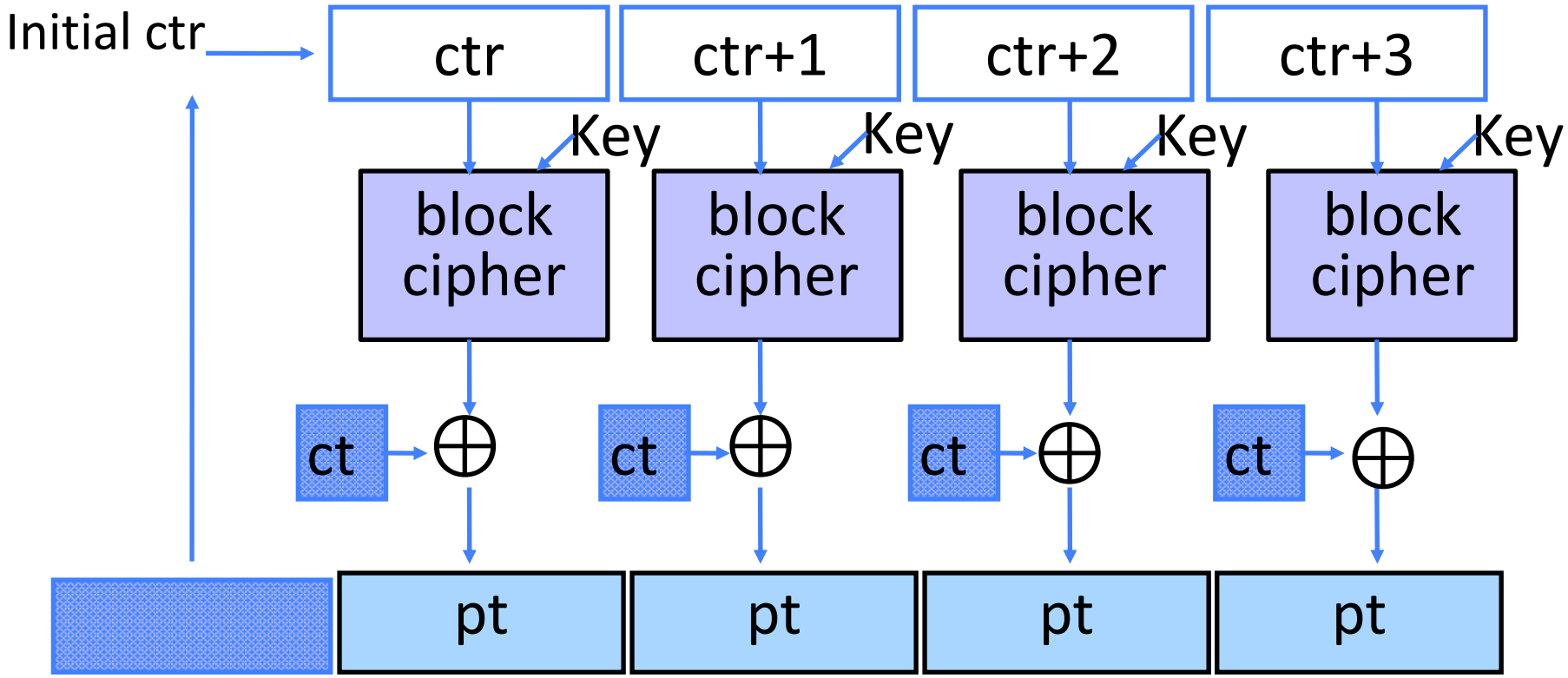
```
DesCBCEncrypt((des_c_block*)tmp, (des_c_block*)record.m_Data,
        totalSize, DESKEY, NULL, DES_ENCRYPT)
```

Counter Mode (CTR): Encryption



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

Counter Mode (CTR): Decryption



Ok, so what mode do I use?

- Don’t choose a mode, use established libraries ☺

- Good modes:
  - GCM - Galois/Counter Mode
  - CTR (sometimes)
  - Even ECB is fine in “the right circumstance” ← so much packed into “the right circumstances” however, so best to avoid

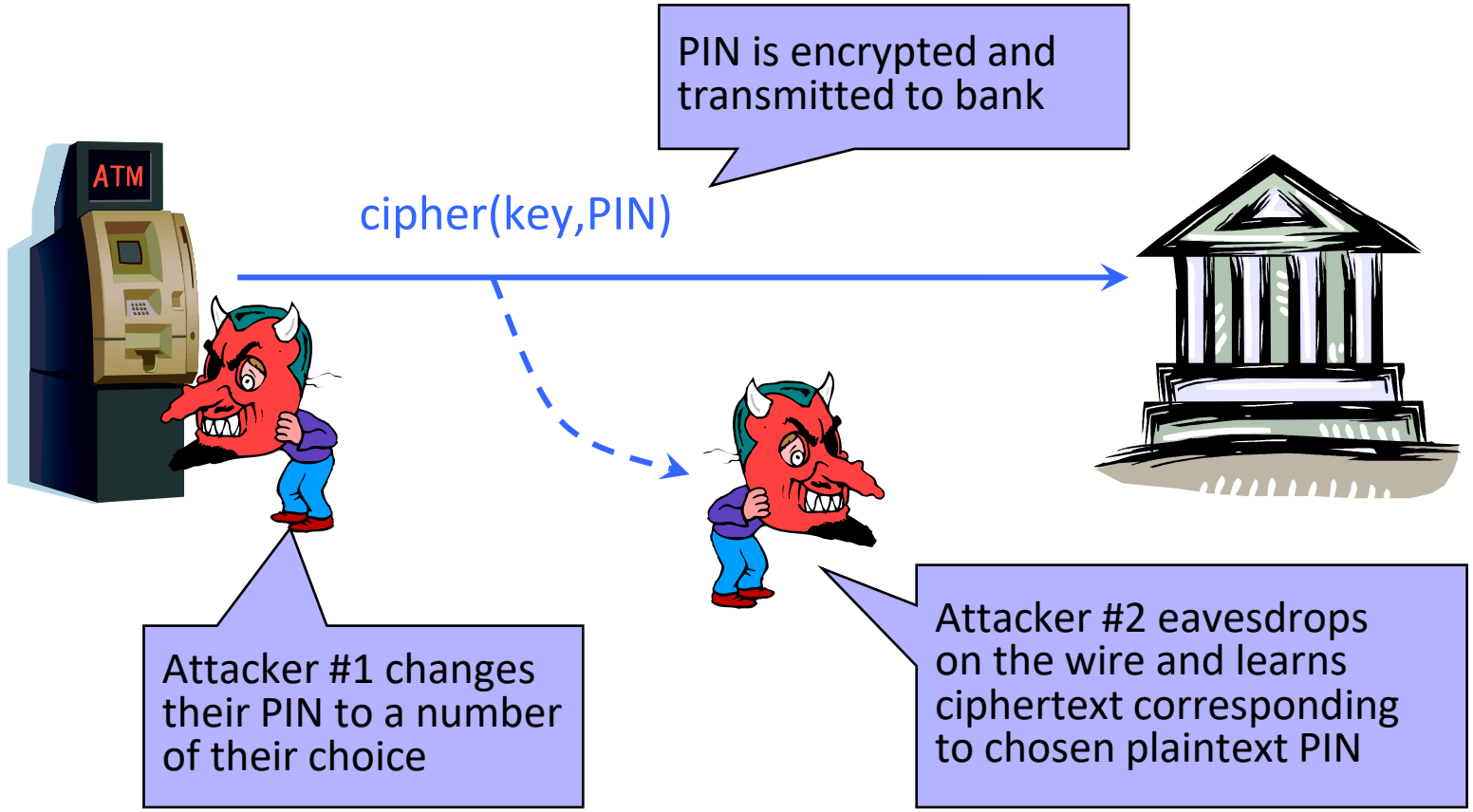
When is an Encryption Scheme “Secure”?

- 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 choice
- CCA: Chosen-ciphertext attack (very strong)
  - Can decrypt any ciphertext except the target

Chosen Plaintext Attack



… repeat for any PIN value

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