CSE 484 / CSE M 584: Computer Security and Privacy

Cryptography
[Symmetric Encryption]

Spring 2020

Franziska (Franzi) Roesner

franzi@cs.washington.edu

Admin

- Final Lab 1 deadline now Friday 5/1 (not Wed)
- More help
  - We'll be adding one more OH on Fridays
  - More on sploit 5 in section this week
- Homework 2 (crypto) will be out soon
  - Due on 5/8 (designed to give you hands-on experience with crypto concepts, not be tricky -- should not take you a full 2 weeks)

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

Handwritten annotation (in green): smallseed ↓
plaintext ⊕ PRNG(seed)

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
  - Same key is reused for each block (can use short keys)
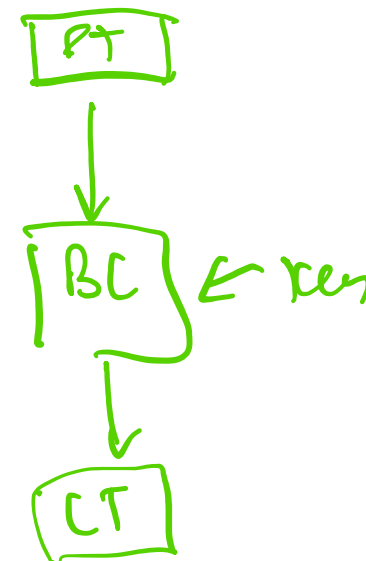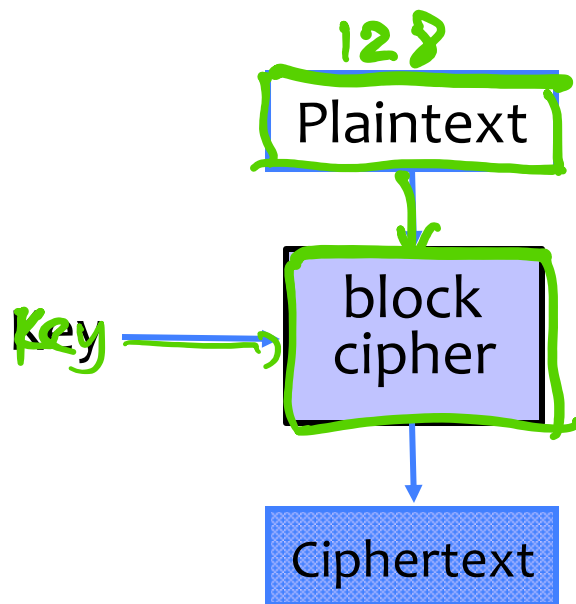
Keyed Permutation

- 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

0101

Plaintext

Key → block cipher

Ciphertext
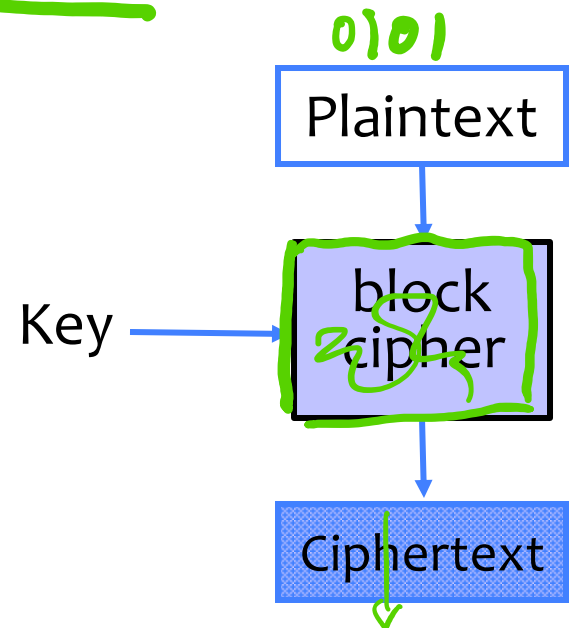
Keyed Permutation

how many cols? (permutations)

$2^n!$ $2^{128}!$

input	possible output	possible output	etc.
3 bit block cipher			
000	010	111	…
001	111	110	…
010	101	000	…
011	110	101	…
…	…		…
111	000	110	…

Key = 00
Key = 01   *2 bit key*

$2^2 = 4$
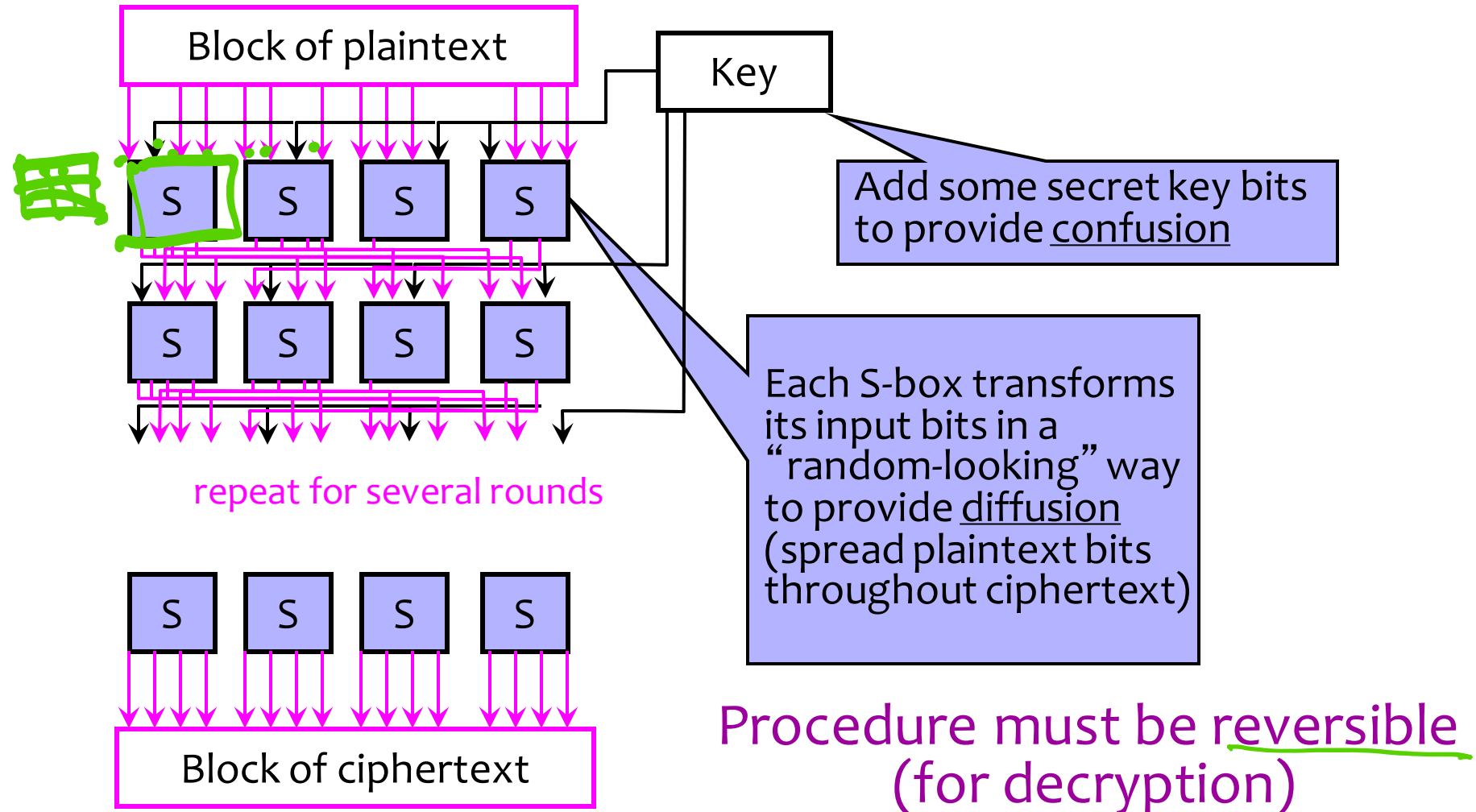
2^{256}

input → 2(B_C) → output

For N-bit input, $2^N!$ possible permutations
For K-bit key, 2^K possible keys

Block Cipher Security

- Result should look like a random permutation on the inputs
  - Recall: not just shuffling bits. 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)

Block of plaintext

Key

S S S S

Add some secret key bits to provide <u>confusion</u>

S S S S

Each S-box transforms its input bits in a "random-looking" way to provide <u>diffusion</u> (spread plaintext bits throughout ciphertext)

repeat for several rounds

S S S S

Block of ciphertext

Procedure must be <u>reversible</u> (for decryption)
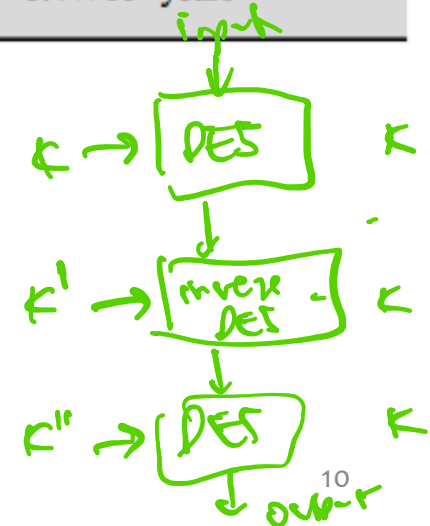
Standard Block Ciphers

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

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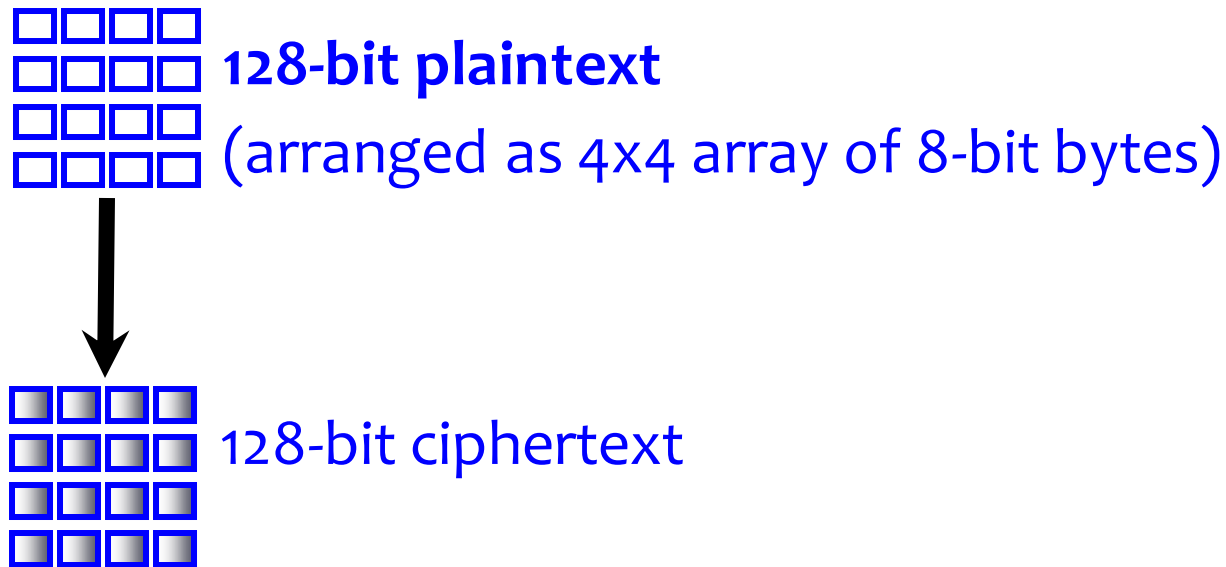 - 3DES: DES + inverse DES + DES (with 2 or 3 diff keys)

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   ← algorithm is public
    - 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

128-bit plaintext

(arranged as 4x4 array of 8-bit bytes)

128-bit ciphertext

- What should we do?

Electronic Code Book (ECB) Mode

128 bit   128 bit   plaintext   ...

Hello

key   key   key   key   key

| block cipher | block cipher | block cipher | block cipher | block cipher |
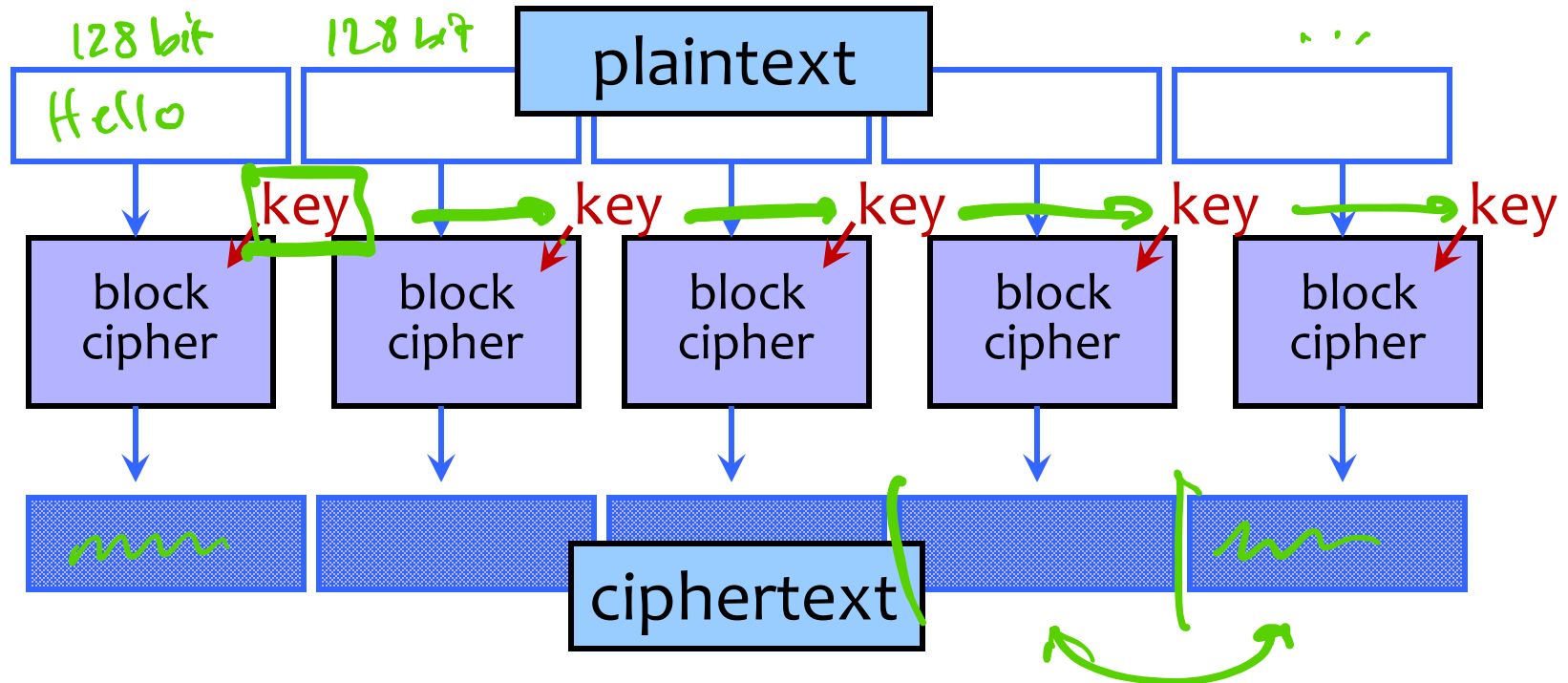
ciphertext

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

Information Leakage in ECB Mode
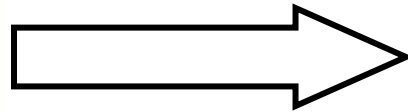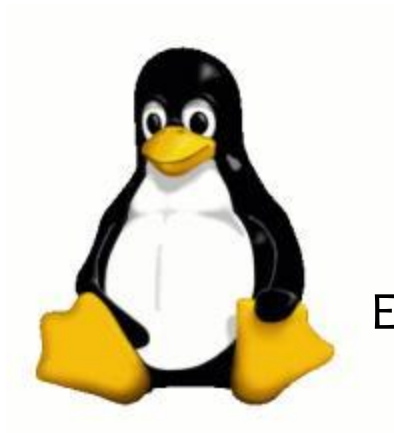
Encrypt in ECB mode
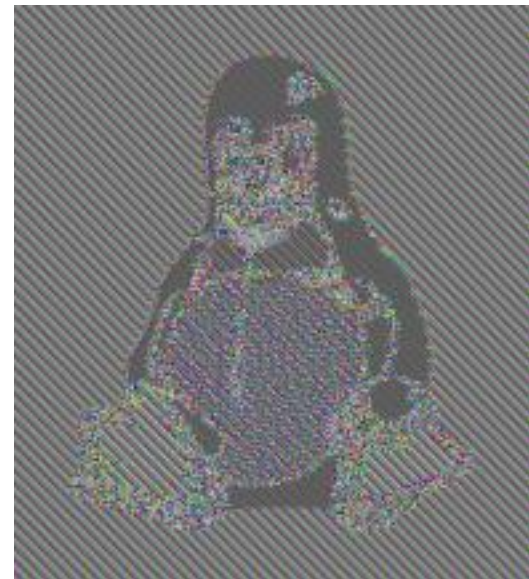
[Wikipedia]

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.
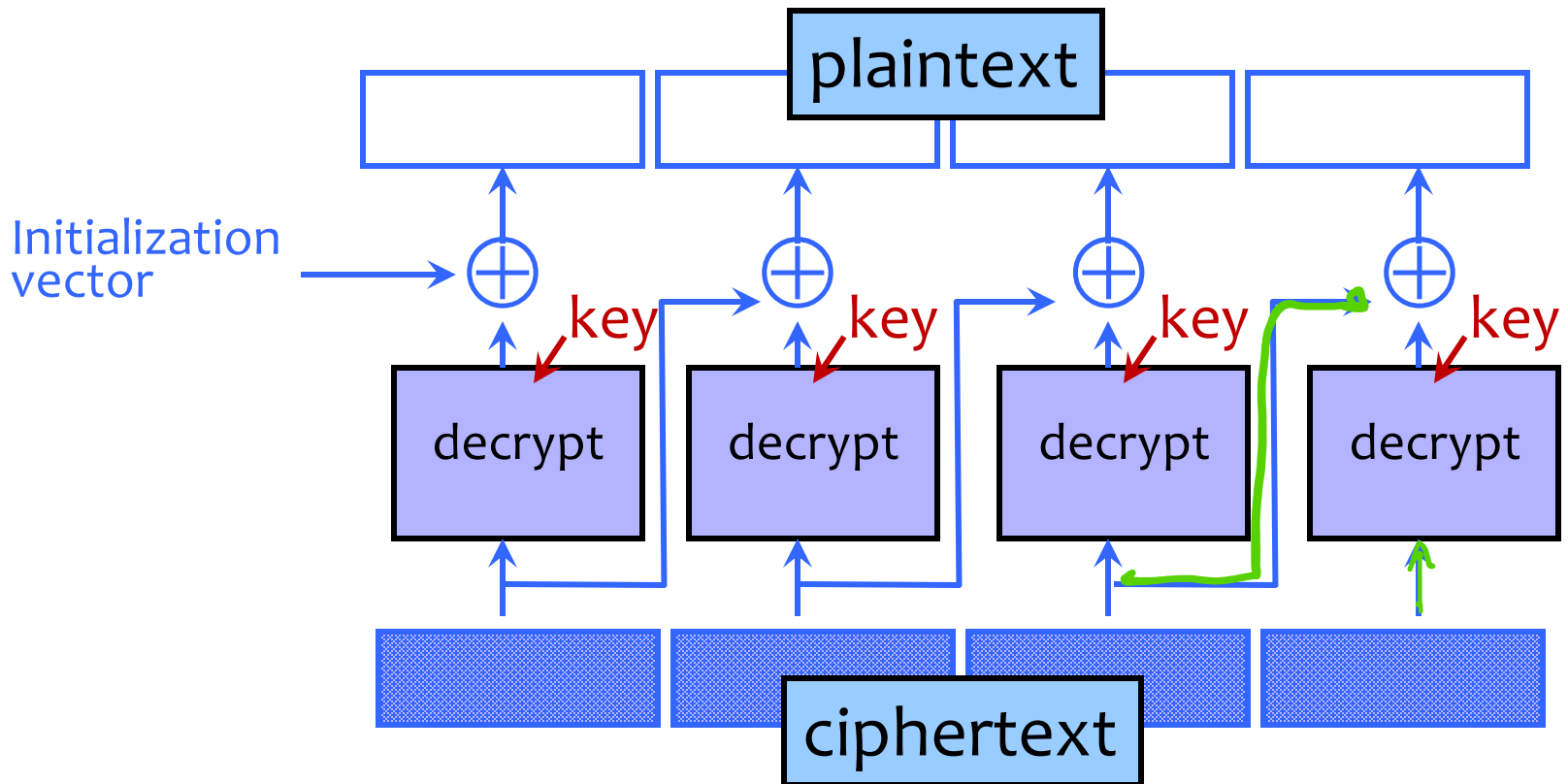
https://citizenlab.ca/2020/04/move-fast-roll-your-own-crypto-a-quick-look-at-the-confidentiality-of-zoom-meetings/
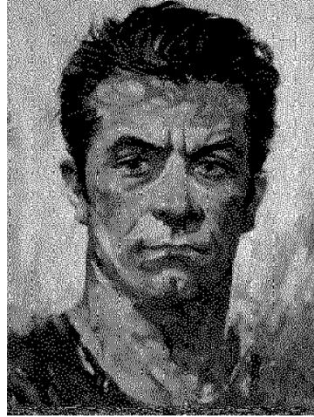
Cipher Block Chaining (CBC) Mode: Encryption
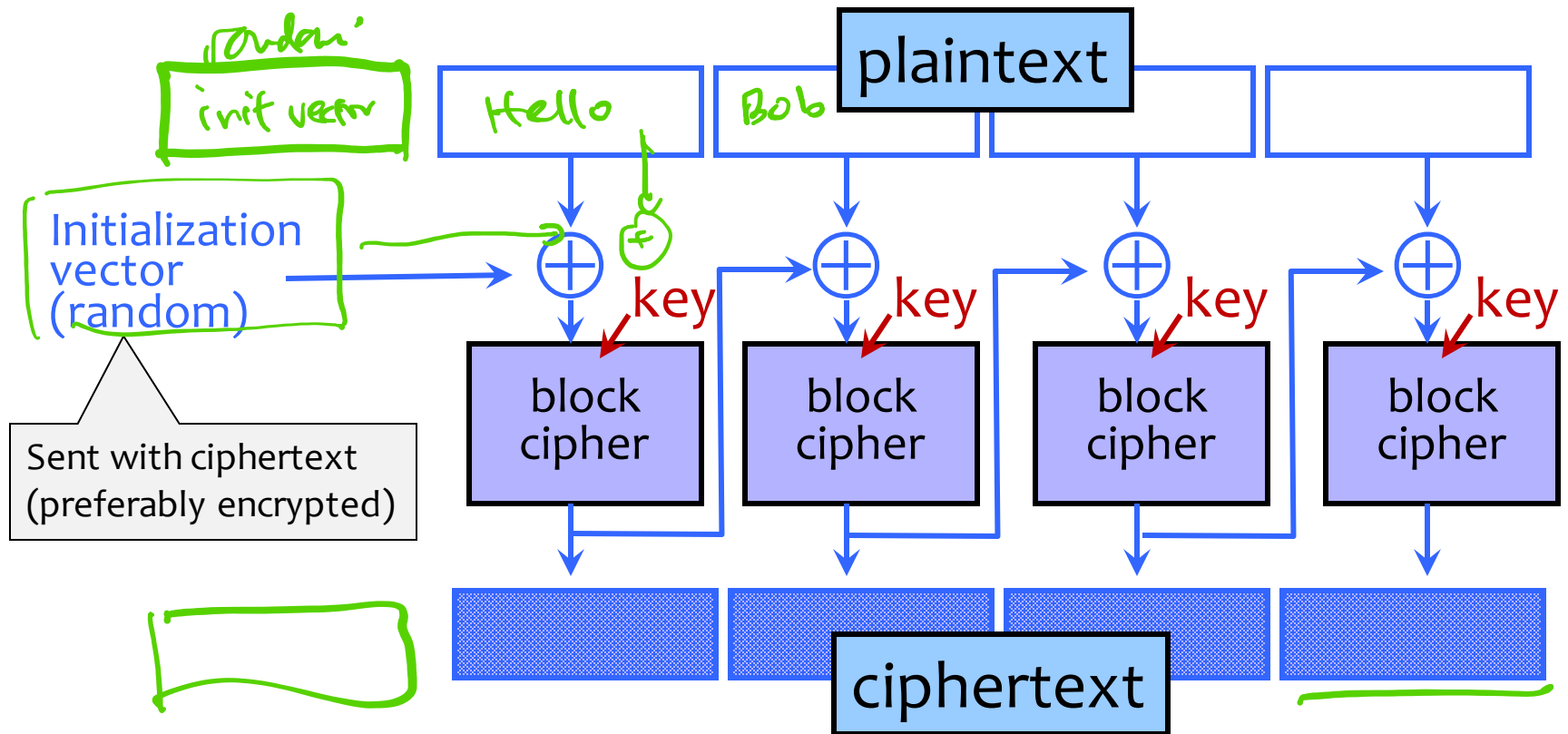
- Identical blocks of plaintext encrypted differently
- Last cipherblock depends on entire plaintext
  - Still does not guarantee integrity
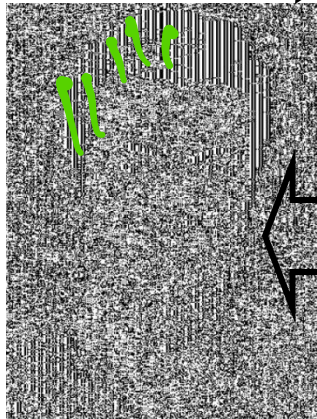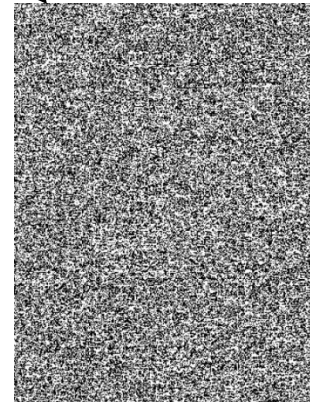
CBC Mode: Decryption
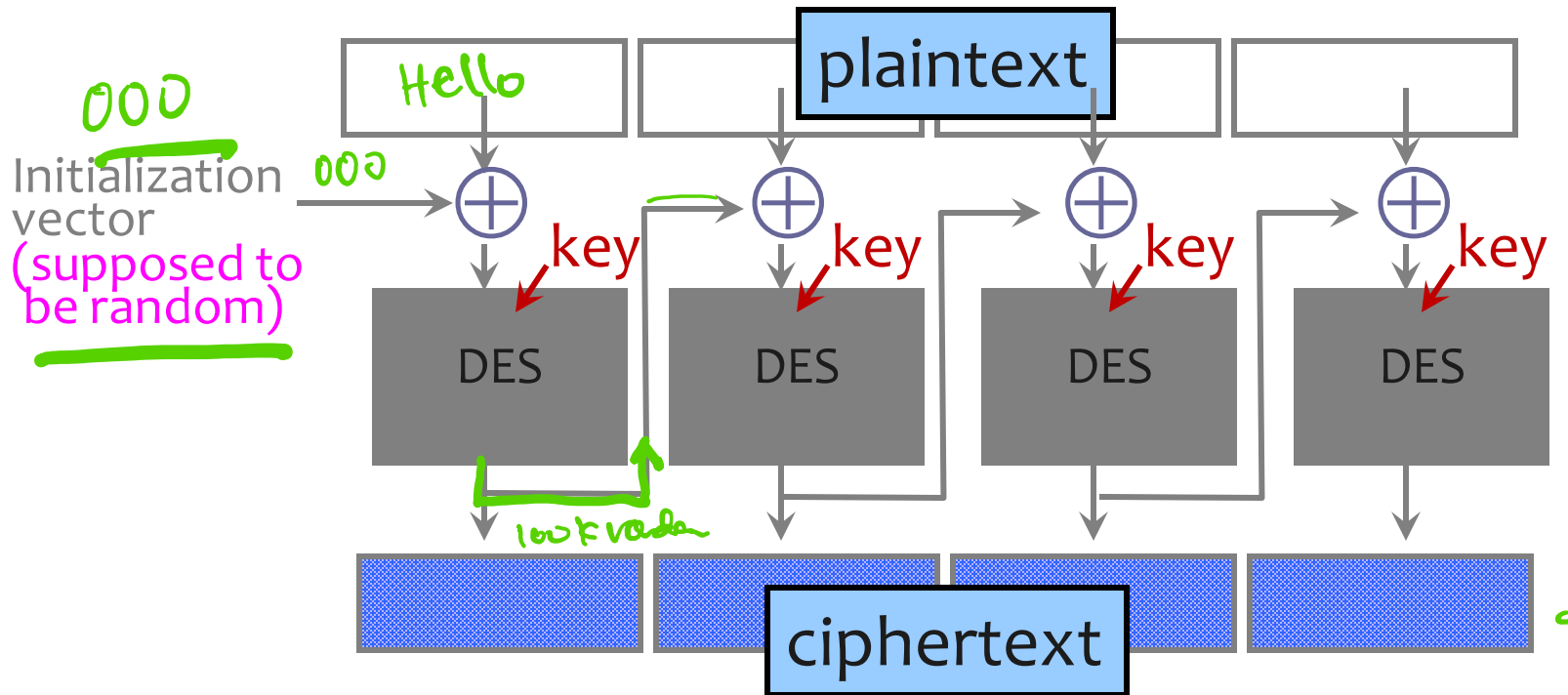
ECB vs. CBC

AES in ECB mode

AES in CBC mode

Similar plaintext blocks produce similar ciphertext blocks (not good!)

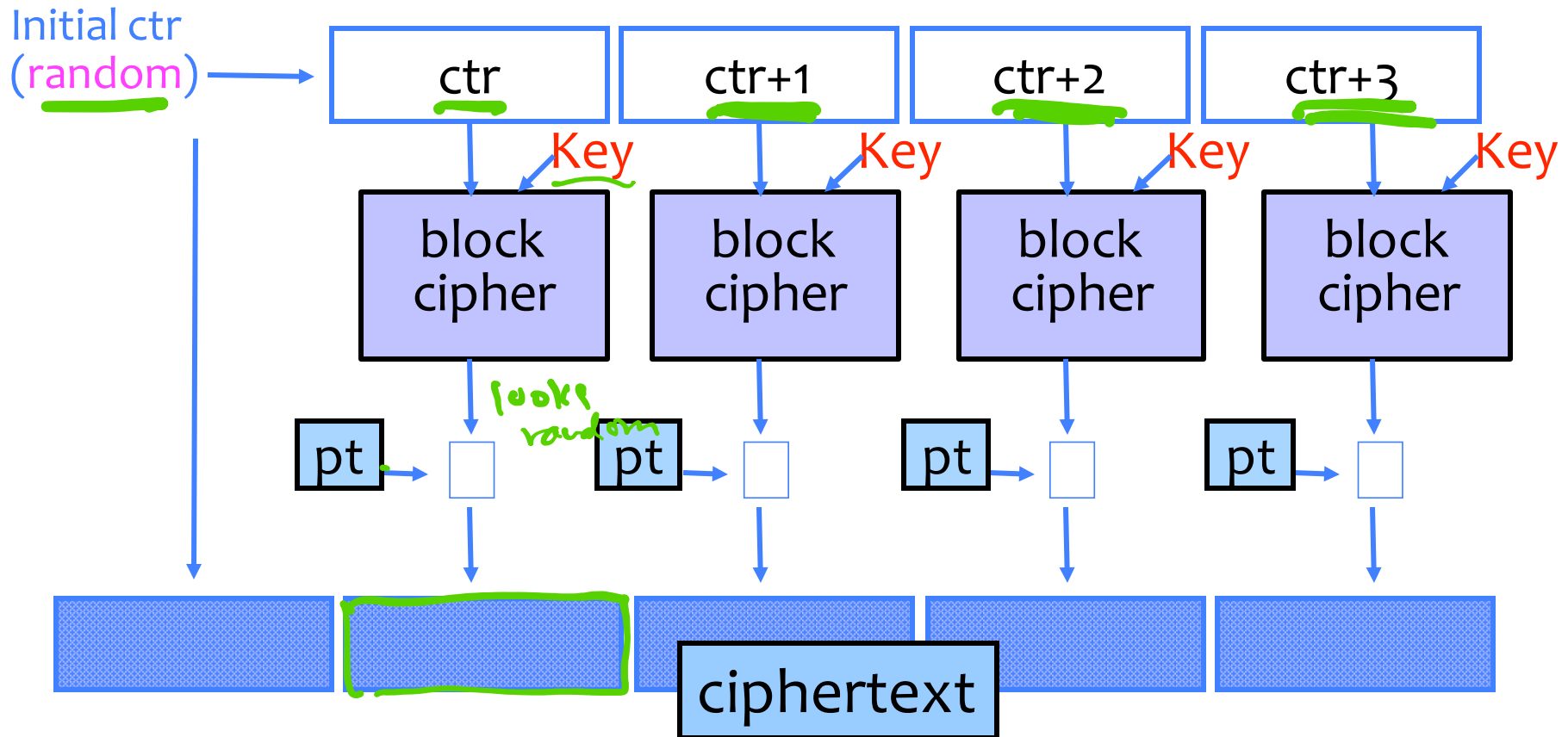[Picture due to Bart Preneel]

slide 18

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

Initial ctr (random)

| ctr | ctr+1 | ctr+2 | ctr+3 |

Key   Key   Key   Key

block cipher   block cipher   block cipher   block cipher

looks random

pt   pt   pt   pt

ciphertext

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

Counter Mode (CTR): Decryption