WEEK3

ENCRYPTING WITH BLOCK CIPHERS

SE 4472 - Information Security

Western Engineering

WHISPER LAB

BLOCK CIPHERS

- Encryption of fixed length chunks of bits (called “blocks”)
- Each block is encrypted separately

BLOCK CIPHERS

KEYGEN

Accepts a security parameter k, outputs a random k-bit key

Gen : $k \rightarrow \{0,1\}^k$

ENCRYPT

Accepts a b-bit plaintext and k-bit key and outputs a b-bit ciphertext

Enc : $\{0,1\}^b \times \{0,1\}^k \rightarrow \{0,1\}^b$

DECRYPT

Accepts a b-bit ciphertext and k-bit key and outputs a b-bit plaintext

Dec : $\{0,1\}^b \times \{0,1\}^k \rightarrow \{0,1\}^b$

AN IDEAL BLOCK CIPHER

AN IDEAL BLOCK CIPHER

Encryption is injective

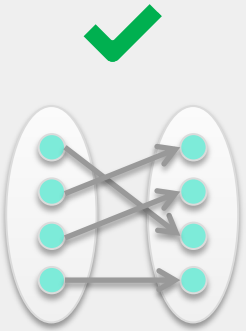
Every plaintext maps to a unique ciphertext

AN IDEAL BLOCK CIPHER

Encryption is surjective

Every ciphertext is associated with some plaintext

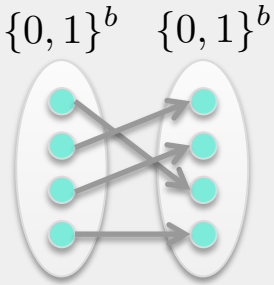
AN IDEAL BLOCK CIPHER



Injective + Surjective = Bijective

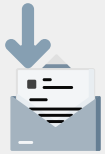
Every plaintext is associated with a unique ciphertext, and vice versa

AN IDEAL BLOCK CIPHER



Encryption is a permutation (shuffle) of b-bit strings

AN IDEAL BLOCK CIPHER



PLAINTEXT SPACE

There are 2^b plaintexts

CIPHERTEXT SPACE

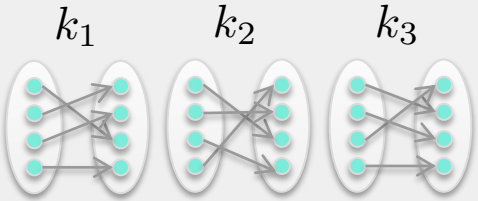
There are 2^b ciphertexts

PERMUTATION SPACE

There are $2^b!$ permutations of 2^b elements

KEY SPACE

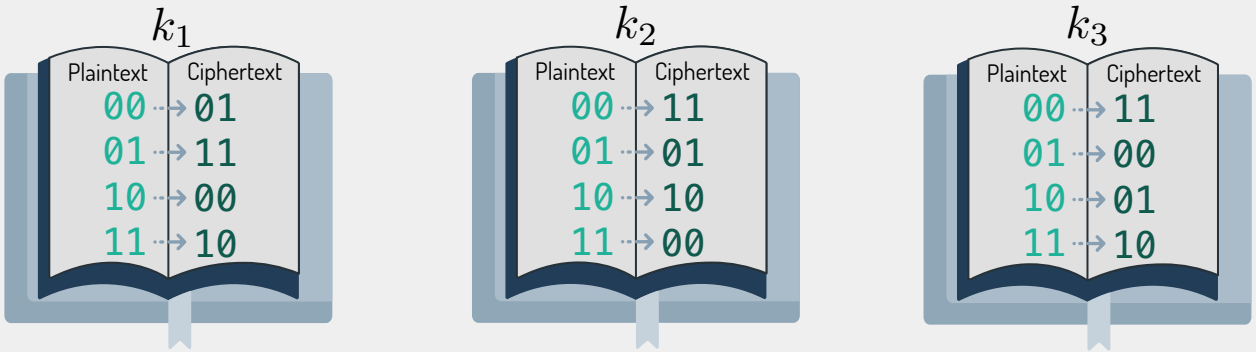
There are 2^k keys



PERMUTATION SELECTION

Each key “chooses” some permutation from the permutation space

AN ELECTRONIC CODEBOOK



So a block cipher is like a collection of codebooks

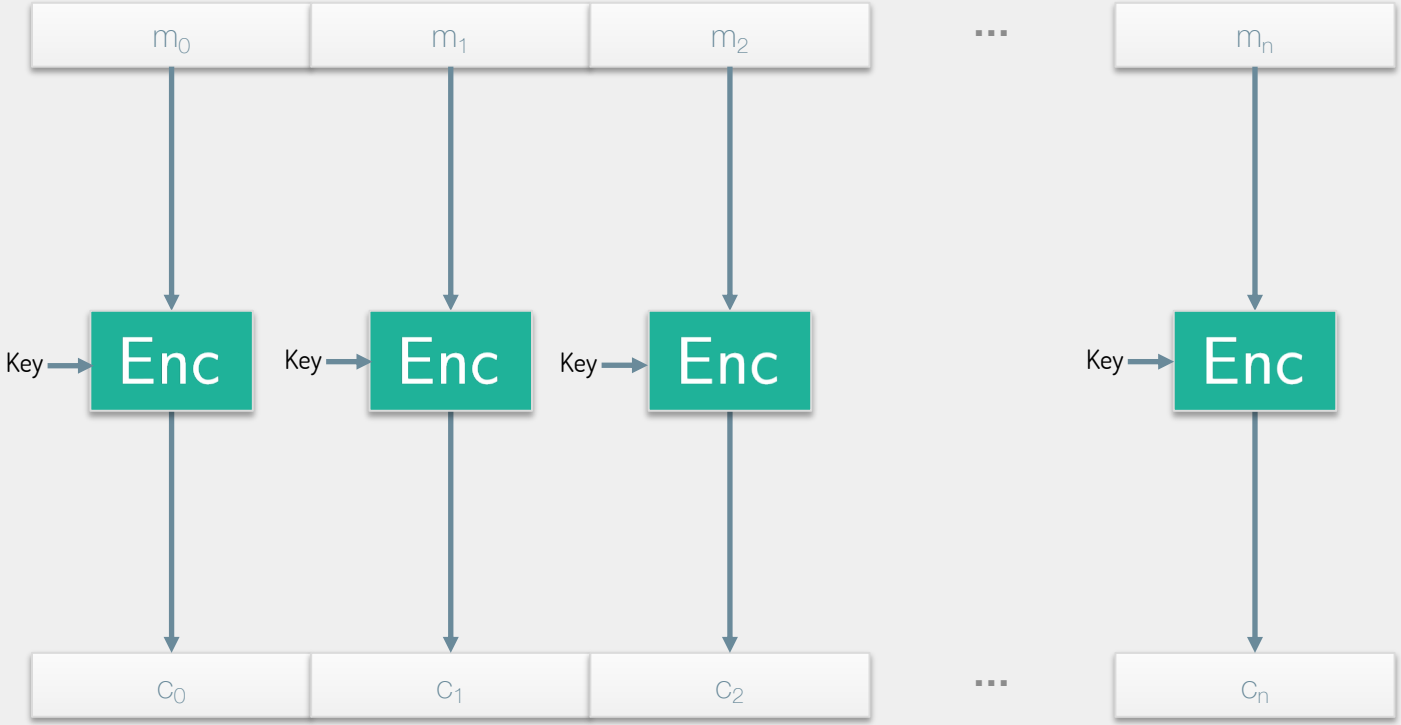
Which codebook did you pick? Shh, it's a secret

ENCRYPTING LARGE PLAINTEXTS

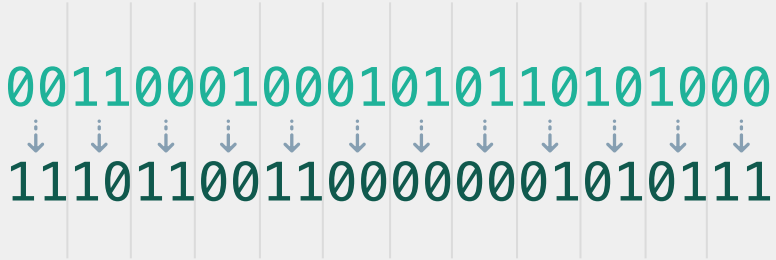
IF BLOCK CIPHERS ACT ON SHORT BLOCKS, HOW DO WE ENCRYPT A LONG MESSAGE?

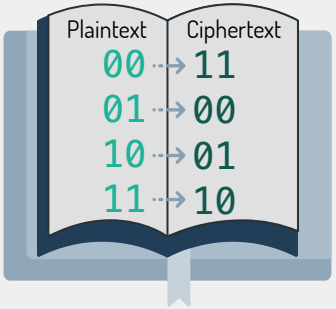
CIPHER MODES OF OPERATION

ELECTRONIC CODEBOOK MODE (ECB)



ELECTRONIC CODEBOOK MODE (ECB)





APPROACH

ENCRYPT EACH BLOCK SEPARATELY

ELECTRONIC CODEBOOK MODE (ECB)

ECB with a 128-bit block size (for a given key)

The codebook has 2^{128} entries

PLAINTEXT	CIPHERTEXT
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	67 8E EE 31 4C FC 24 2E F7 62 9E 39 D3 08 A1 4E
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01	51 03 35 71 62 68 87 E3 CB 16 86 BD EF C6 13 C3
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02	3B 1D F9 59 A8 31 C6 70 AF A8 F2 4A B9 B8 27 4B
⋮	⋮
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FD	F1 34 B9 FD 4B 01 43 33 C7 20 C2 17 29 A5 CB 11
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FE	D1 2E 8D B2 4F F4 F3 77 E8 9F 92 31 A3 38 48 B0
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	B0 5B 46 3E 58 2A AE FC 5A 40 58 E9 B4 5F 34 B3

ELECTRONIC CODEBOOK MODE (ECB)

Wait a second

Doesn't that mean this:

```
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
```

always encrypts to this?:

```
B0 5B 46 3E 58 2A AE FC 5A 40 58 E9 B4 5F 34 B3
```

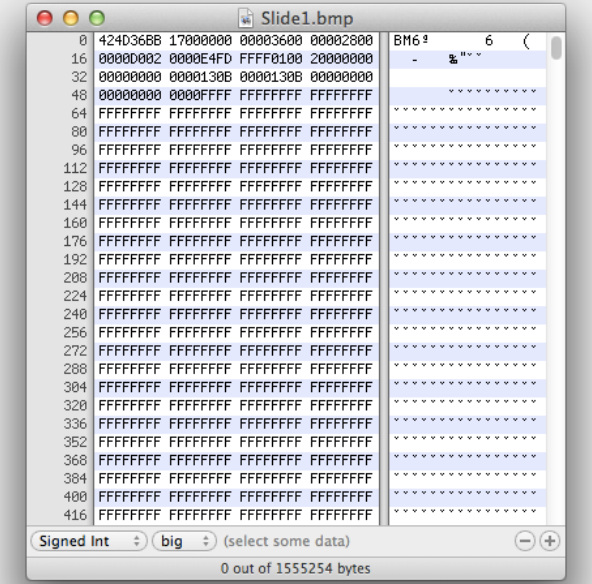
ELECTRONIC CODEBOOK MODE (ECB)

Slide1.bmp

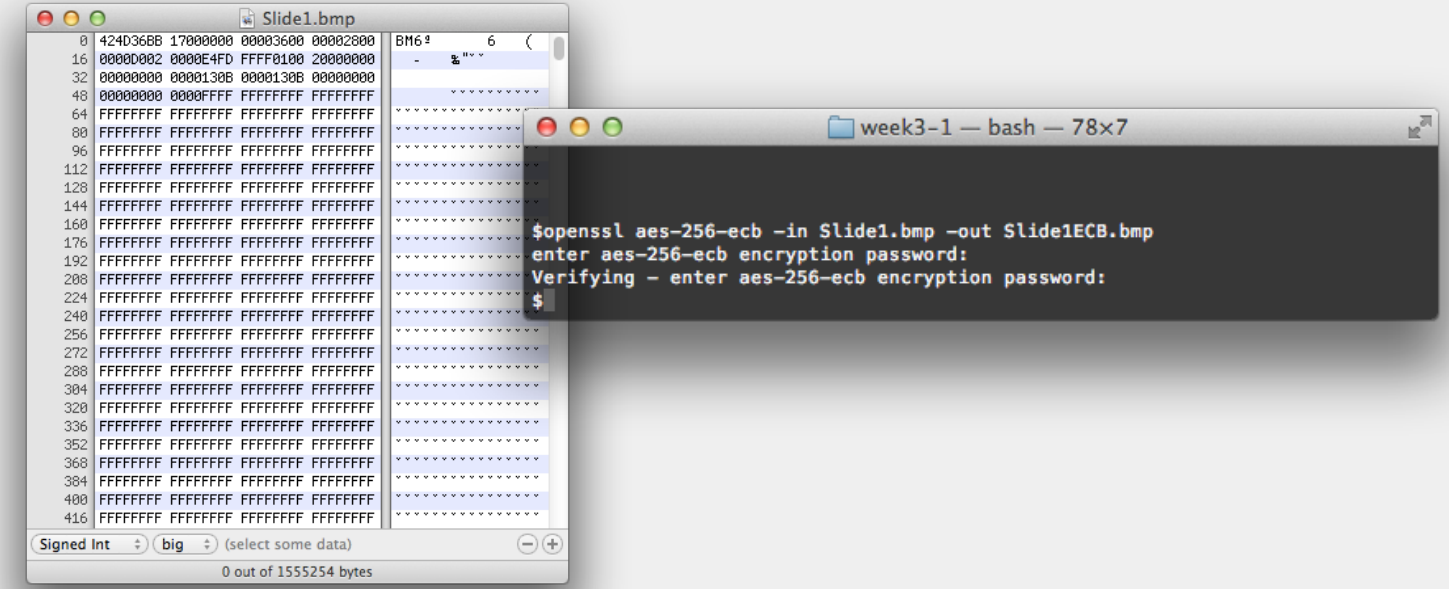
SE 4472: Electronic Codebook Mode

What could possibly go wrong?

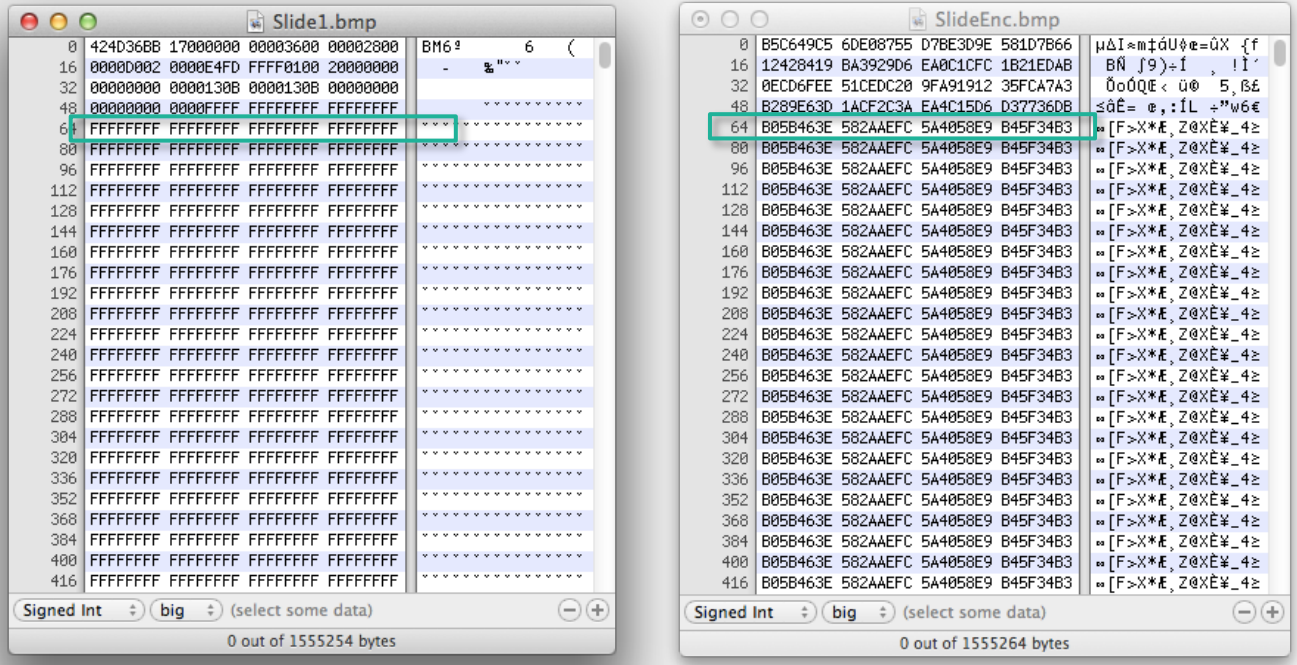
hexdump



ELECTRONIC CODEBOOK MODE (ECB)

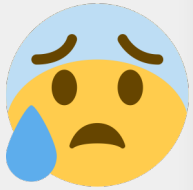


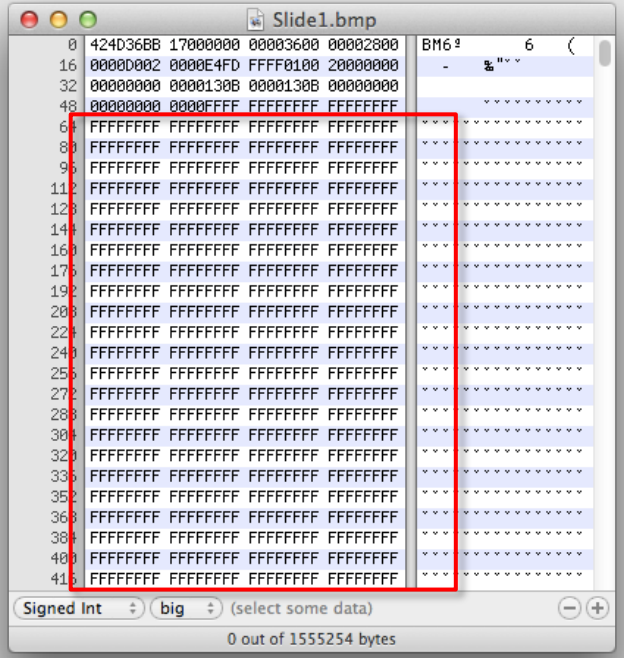
ELECTRONIC CODEBOOK MODE (ECB)

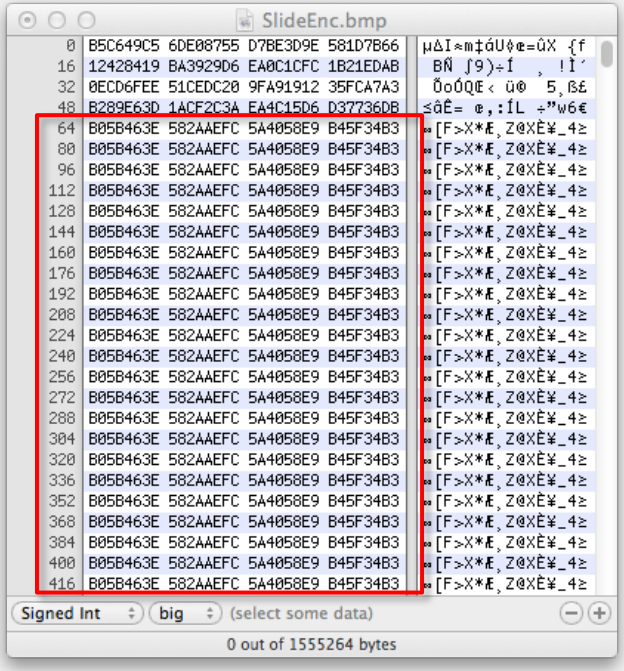


BLOCKS OF WHITE PIXELS ENCRYPT TO RANDOM-LOOKING COLOURS

ELECTRONIC CODEBOOK MODE (ECB)

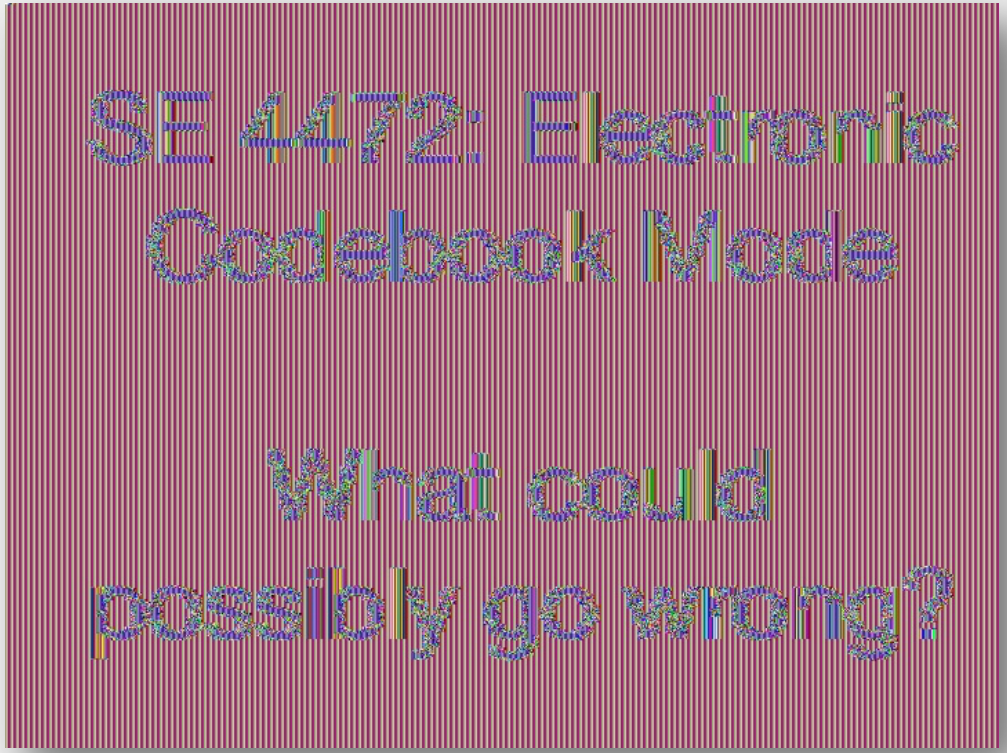






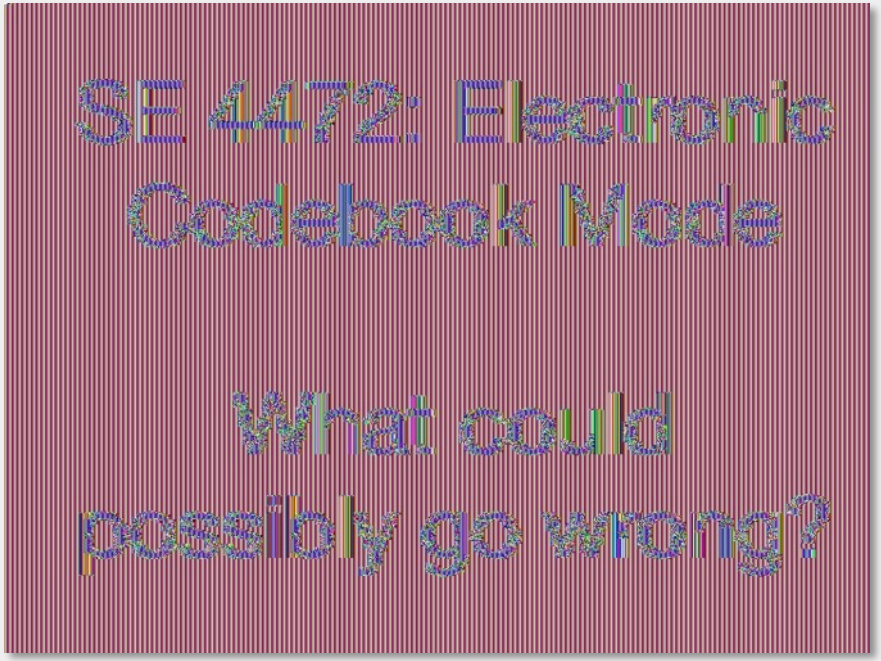
PROBLEM: THEY ALWAYS ENCRYPT TO THE SAME COLOURS

ELECTRONIC CODEBOOK MODE (ECB)



ELECTRONIC CODEBOOK MODE (ECB)

Is this cipher mode indistinguishable under eavesdropping? (IND-EAV)



ELECTRONIC CODEBOOK MODE (ECB)

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

> Read our description of Zoom's waiting room vulnerability, as well as frequently asked question about Zoom and encryption issues.

This report examines the encryption that protects meetings in the popular Zoom teleconference app. We find that Zoom has "rolled their own" encryption scheme, which has significant weaknesses. In addition, we identify potential areas of concern in Zoom's infrastructure, including observing the transmission of meeting encryption keys through China.

Key Findings

- 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.

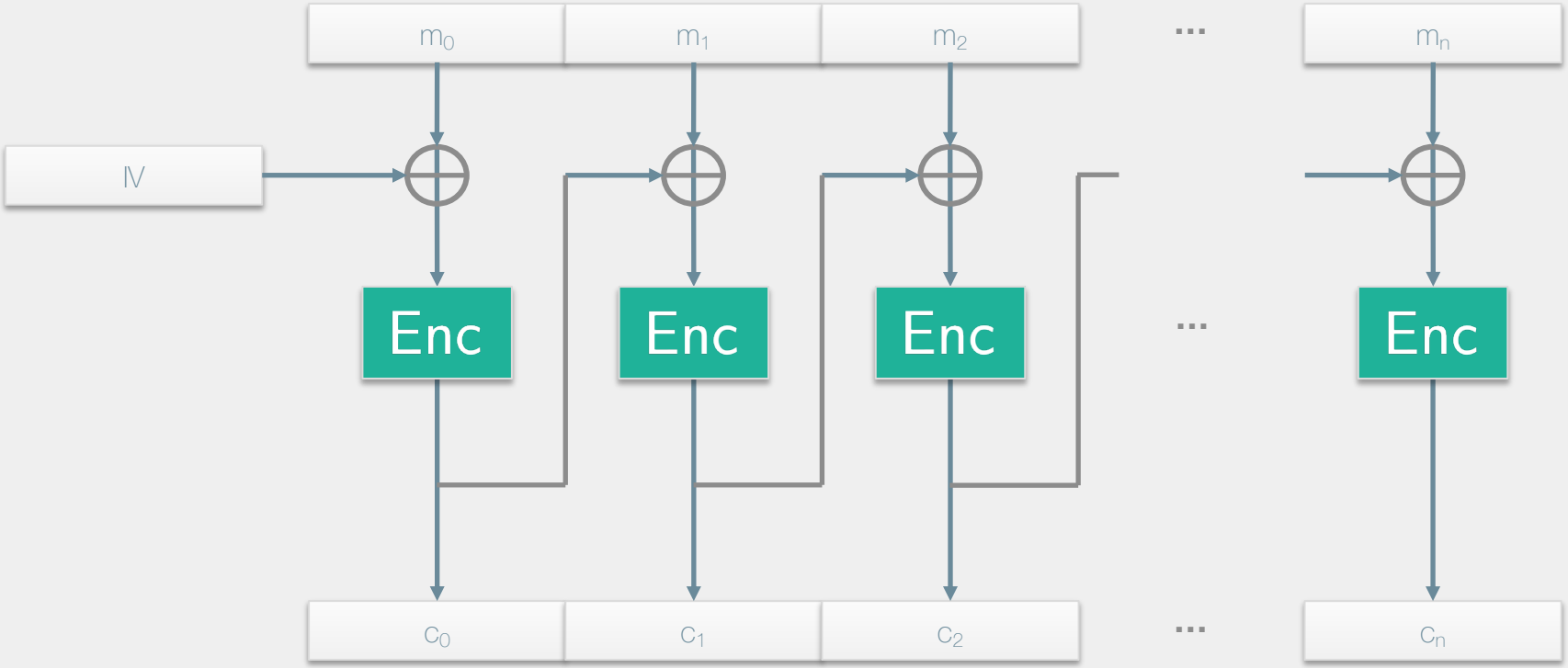
ELECTRONIC CODEBOOK MODE (ECB)

WHAT IF WE COMBINED BLOCKS TOGETHER SOMEHOW TO BREAK UP LOCAL STRUCTURE?

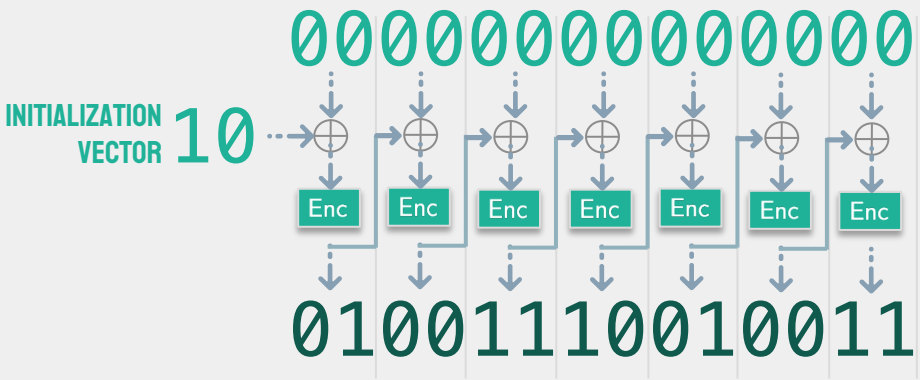
ELECTRONIC CODEBOOK MODE (ECB)

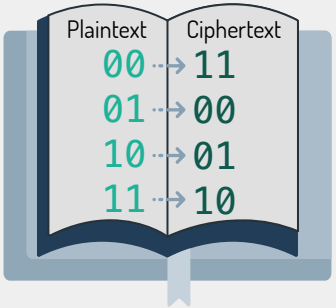
WHAT IF YOU COULD ENCRYPT A PLAINTEXT TWICE BUT GET TWO DIFFERENT CIPHERTEXTS?

CIPHER BLOCK CHAINING (CBC)



CIPHER BLOCK CHAINING (CBC)

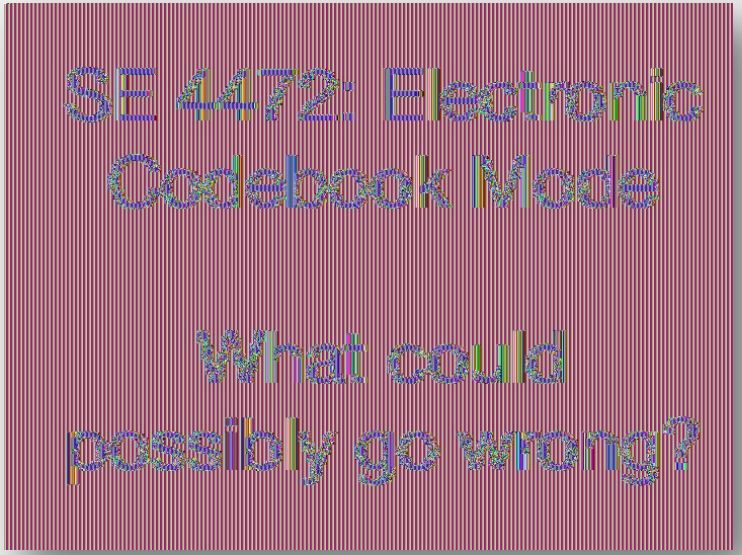




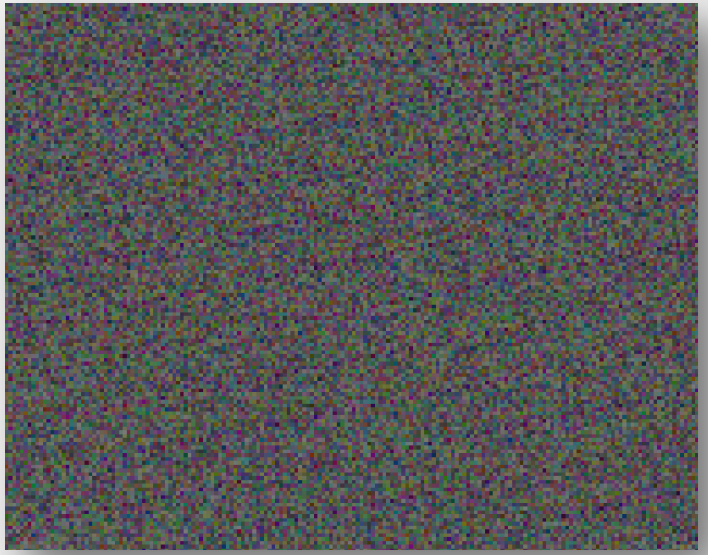
APPROACH

XOR EACH PLAINTEXT BLOCK WITH PREVIOUS CIPHERTEXT BLOCK

CIPHER BLOCK CHAINING (CBC)

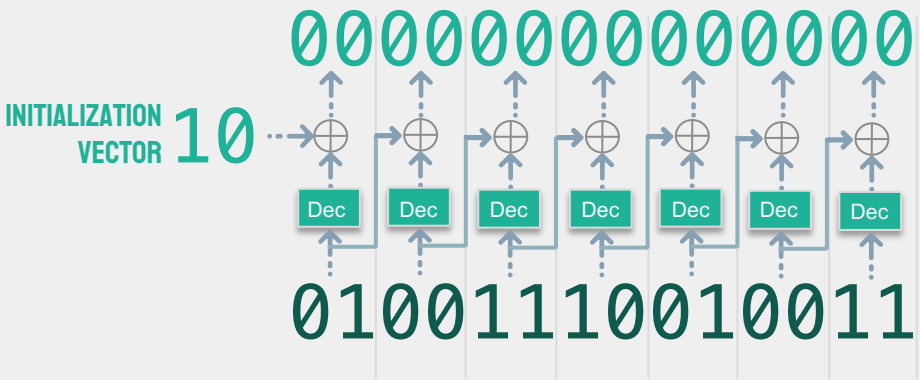


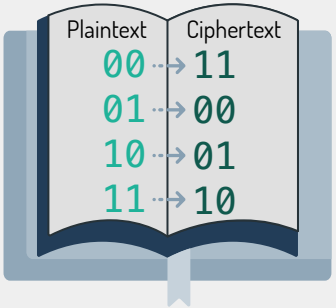
ECB MODE



CBC MODE

CIPHER BLOCK CHAINING (CBC)

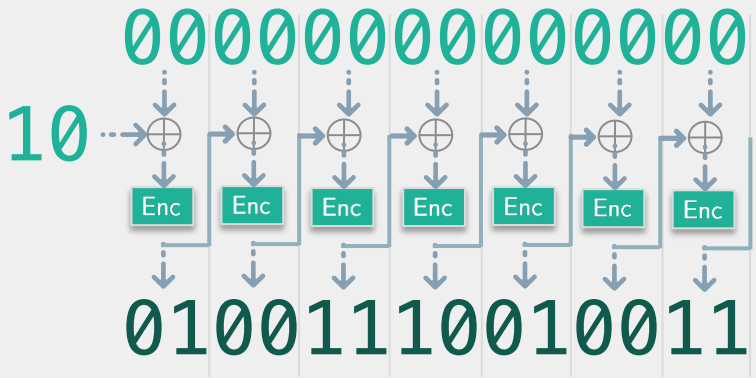


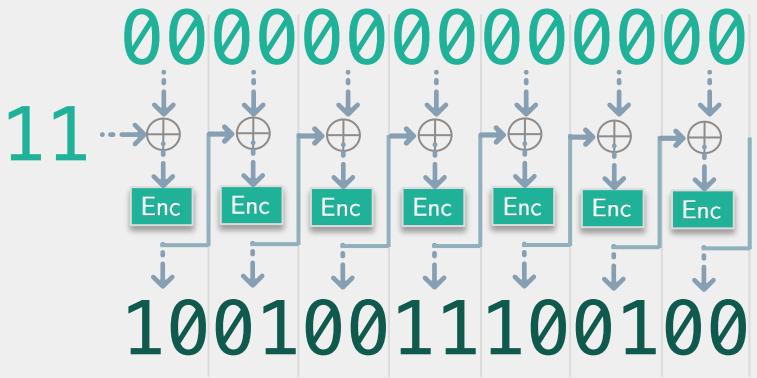


DECRYPTION STILL FEEDS THE PREVIOUS CIPHERTEXT FORWARD

INITIALIZATION VECTOR IS NEEDED TO DECRYPT

CIPHER BLOCK CHAINING (CBC)





RANDOMIZED ENCRYPTION

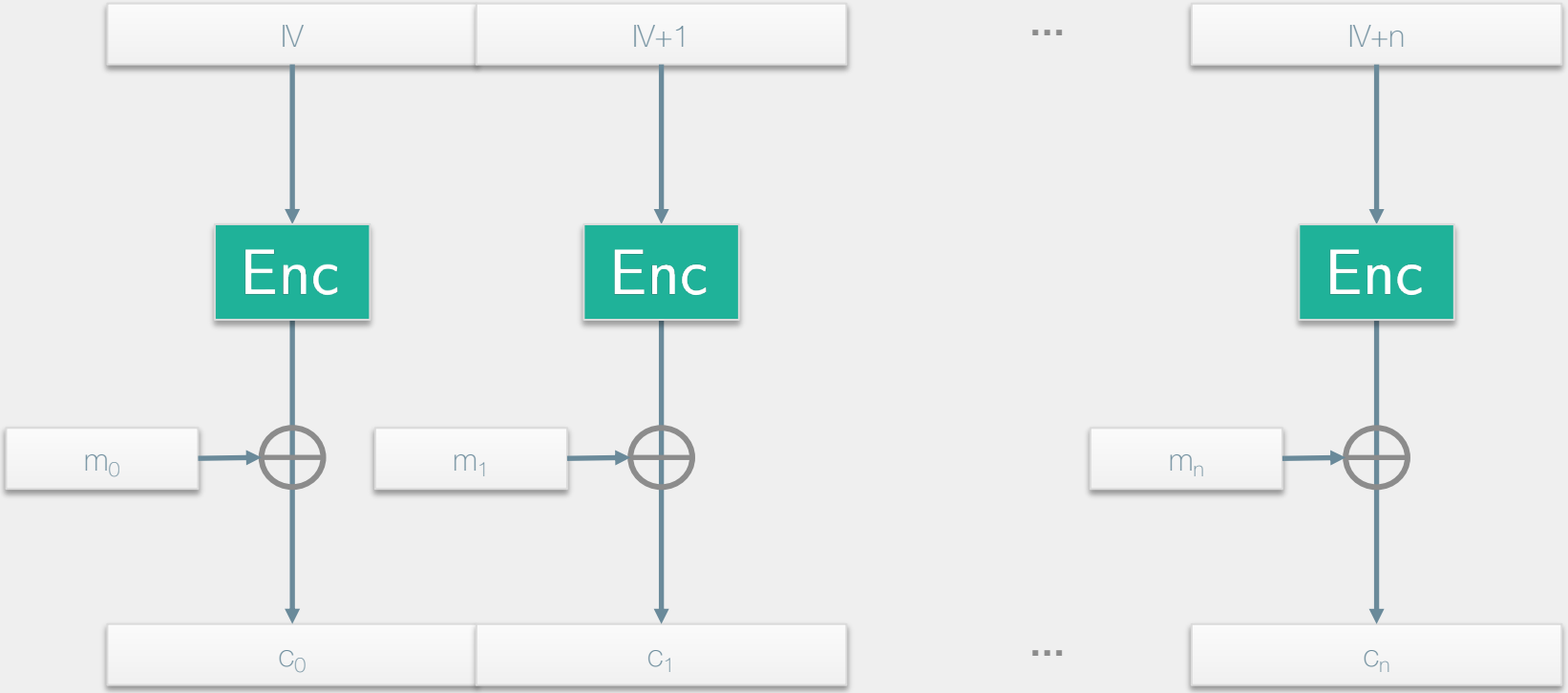
DIFFERENT INITIALIZATION VECTORS PRODUCE
DIFFERENT CIPHERTEXTS ON THE SAME PLAINTEXT

CIPHER BLOCK CHAINING (CBC)

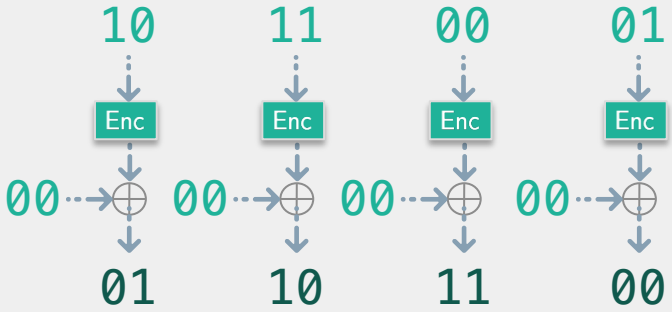
The initialization vector is:

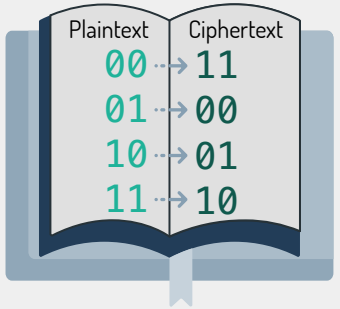
- Not secret
- Sent along with the ciphertext
- Must only ever be used once
- Must not be predictable

COUNTER MODE (CTR)



CIPHER BLOCK CHAINING (CBC)





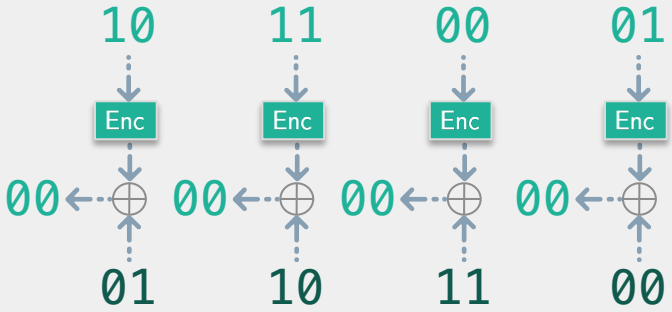
PLAINTEXT: 00000000

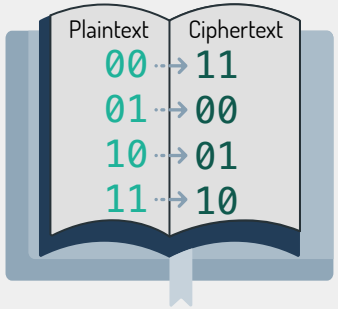
IV: 10

APPROACH

ENCRYPT A COUNTER. XOR RESULT WITH PLAINTEXT

CIPHER BLOCK CHAINING (CBC)





CIPHERTEXT: 01101100

IV: 10

DECRYPTION

NOTICE IT USES THE ENCRYPTION FUNCTION!

CIPHER BLOCK CHAINING (CBC)

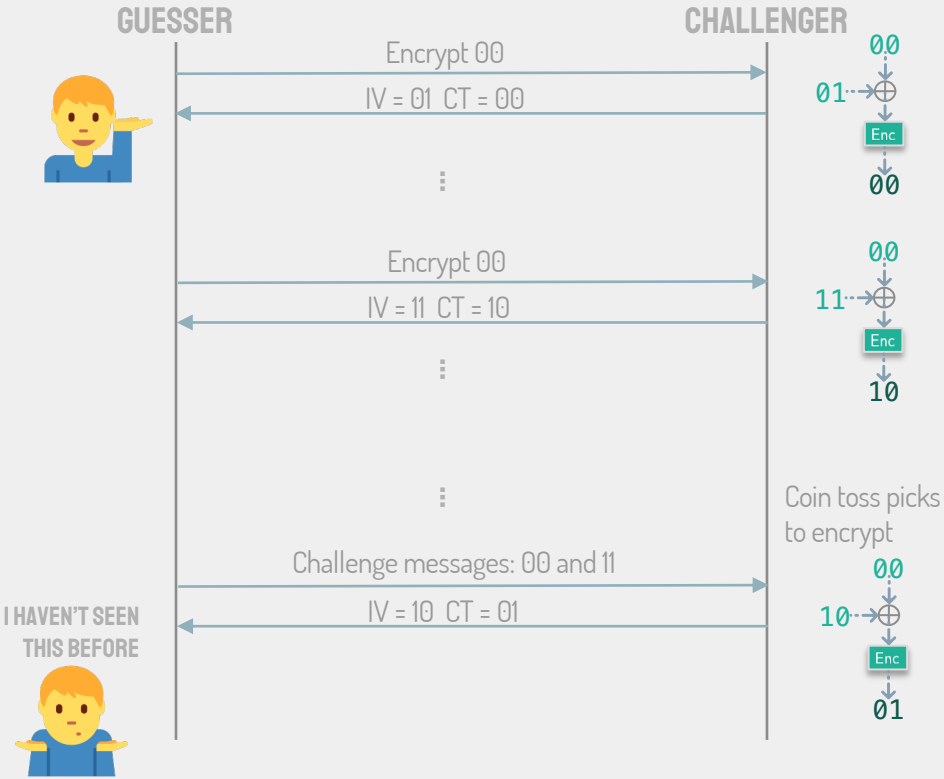
Pros:

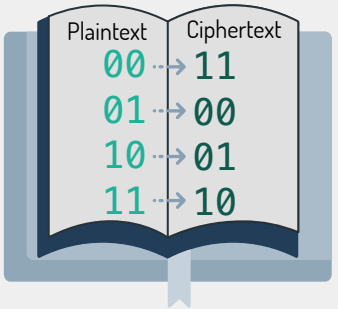
- Not vulnerable to padding oracle attacks
- Simpler implementation
- Parallelizable, pre-processable, random access

Cons:

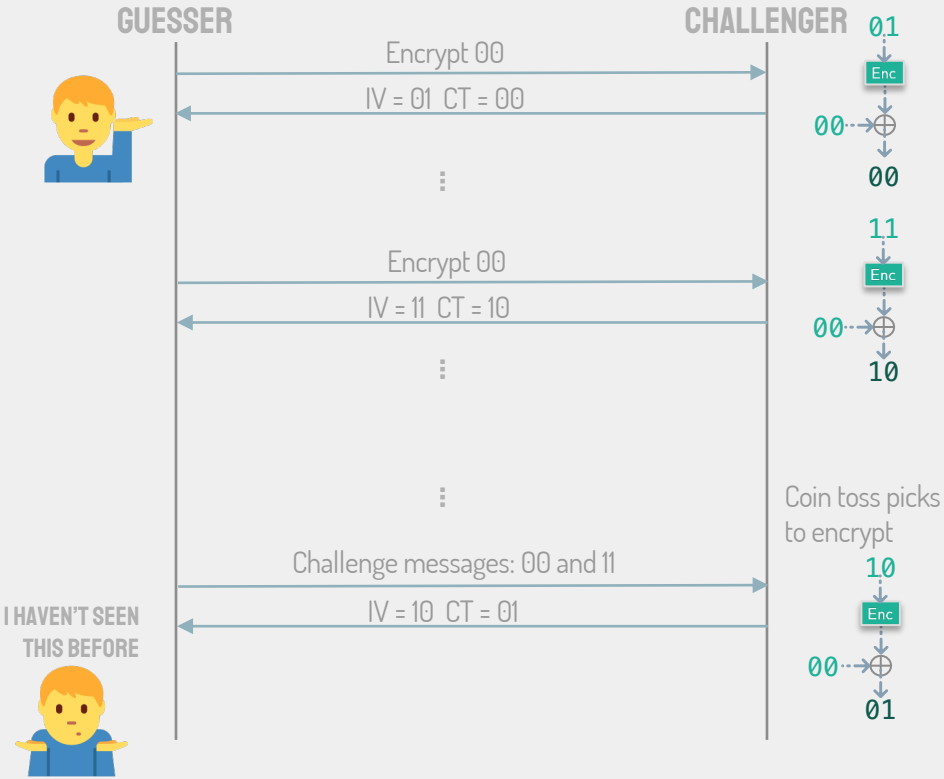
- Not safe for small block lengths (<128-bits)

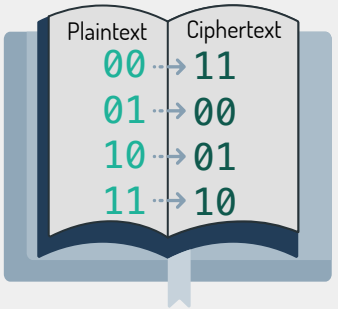
CBC MODE IN THE CPA GAME





CTR MODE IN THE CPA GAME





DEFEATING CHOSEN PLAINTEXT ATTACKS

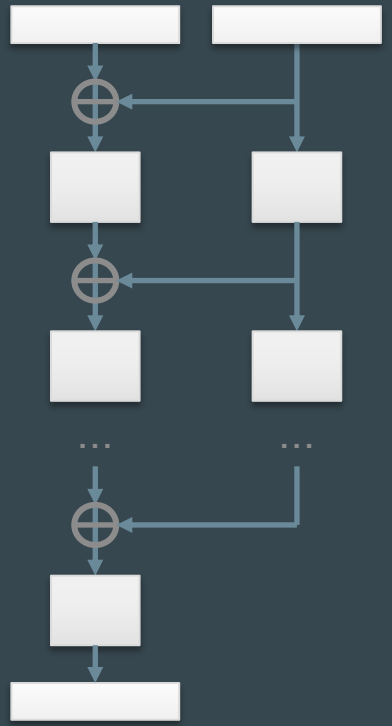
RANDOMIZED ENCRYPTION IS HOW WE CAN DEFEAT THE CHOSEN PLAINTEXT ATTACK

EXERCISE

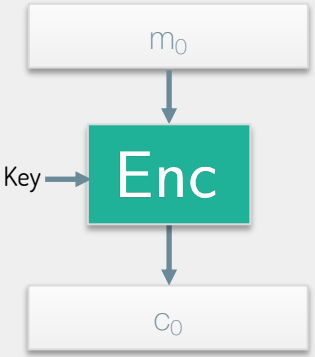
PROVE CBC MODE IS NOT IND-CPA SECURE IF YOU CAN PREDICT THE IV/COUNTER

Hint: Imagine the challenger incremented the IV by 1 with each new encryption. How could you exploit this fact to counteract the contribution of the IV?

ADVANCED ENCRYPTION STANDARD (AES)



AES ENCRYPTION

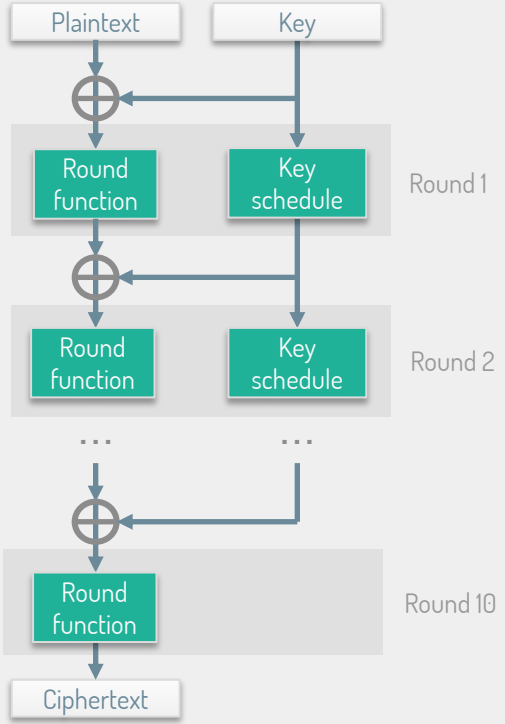


WHAT'S INSIDE?

AES ENCRYPTION

TOP-LEVEL STRUCTURE

- 128-bit plaintext
- 128-bit ciphertext
- 128-, 192, or 256-bit key
- 10, 12, or 14 identical “rounds”
- Key schedule creates different “sub-keys” for each round



AES ROUND FUNCTION

Input (16 bytes)

b_0	b_4	b_8	b_{12}
b_1	b_5	b_9	b_{13}
b_2	b_6	b_{10}	b_{14}
b_3	b_7	b_{11}	b_{15}

SubBytes

c_0	c_4	c_8	c_{12}
c_1	c_5	c_9	c_{13}
c_2	c_6	c_{10}	c_{14}
c_3	c_7	c_{11}	c_{15}

where:

$$c_i = b_i^{-1} + 0\text{x}63$$

Byte-wise inverse in GF(256) plus affine transform

ShiftRows

c_0	c_4	c_8	c_{12}
c_5	c_9	c_{13}	c_1
c_{10}	c_{14}	c_2	c_6
c_{15}	c_3	c_7	c_{11}

Bytes are permuted

MixColumns

where:

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 03 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} c_0 \\ c_5 \\ c_{10} \\ c_{15} \end{pmatrix} = \begin{pmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{pmatrix}$$

Byte-wise matrix multiplication in GF(256) of columns

d_0	d_4	d_8	d_{12}
d_1	d_5	d_9	d_{13}
d_2	d_6	d_{10}	d_{14}
d_3	d_7	d_{11}	d_{15}

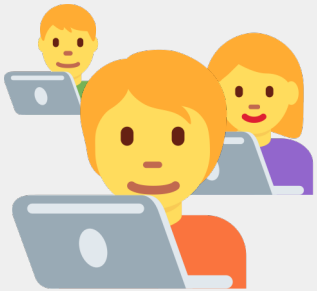
AddRoundKey

e_0	e_4	e_8	e_{12}
e_1	e_5	e_9	e_{13}
e_2	e_6	e_{10}	e_{14}
e_3	e_7	e_{11}	e_{15}

$$e_i = d_i + k_i$$

Bitwise XOR with roundkey

to next round



TREAT AES LIKE A BLACK BOX

What matters to you is what lives outside:

- A random, secret key
- A safe, randomized mode of operation
- An unpredictable initialization vector

QUESTIONS?

Contact Prof. Essex:
aessex@uwo.ca
@aleksessex

See course website for slides and videos:
https://whisperlab.org/security