Set2 Challenge 11

I think this was a very experimental exercise.

Task (cryptopals set2 challenge 11):
Write an encryption oracle with encrypts a plaintext with AES in ECB mode half the time and CBC mode other half.

Write a detection oractle, which given a ciphertext, detects if it's been encrypted with ECB mode with CBC mode.

Easy. Both tasks are pretty simple. Detection of ECB works on the principle that ECB is stateless and deterministic; the same 16 byte plaintext block will always produce the same 16 byte ciphertext.

Extras:
I also wrote a function that will keep track of contents and indices of any repetetive blocks when given a string of bytes, to know to see what's happening under the hood.

Let's test it it now.
First, I need to feed my encryption oracle with something that is somewhat random but also has a lot of repetetive bits. Pop Songs! Pop music is mostly made of 4 chords(often 1st, 5th, 6th, 4th of a scale) playing in a repetetive manner, and the same chorus lyrics coming after every short while. I decide to feed it bits of "Elle Ma dit"- MIKA(i still speak questionale french) to see what the results are:

Results:
I wrote a dumb oracle :(

This is the plaintext I fed the encryption oracle:
link

Repeated blocks in plantext:

Plaintext Block content	Plaintext Block ID
b'tu g\xc3\xa2ches ta vi'	51
b'e?\nPourquoi tu g'	52
b'tu g\xc3\xa2ches ta vi'	97
b'e?\nPourquoi tu g'	98
b'Danse, danse, da'	103

Plaintext Block content	Plaintext Block ID
b'Danse, danse, da'	112
b'e me dit danse, '	147
b'e me dit danse, '	147

Encryption under: CBC mode

Cipher Block Chianing mode basically XORs the last block's cipher text with the current block's plaintext before running it through an encryption function. For the first block it XOR's the plaintext with the Initialisation Vector(IV)(My IV is random 16 bytes)

Formula for CBC encryption:

$$\begin{gathered} C_i = E_k(P_i \oplus C_{i-1}) \\ C_0 = IV \end{gathered}$$

Well it looks liek there is diffusion. My detection oracle detects ECB if it find any identical ciphertext blocks. Shold it detect ECB (we encrypted in CBC, and AES-CBC seems to provide enough diffusion) ?

Turns out it detects ECB! and not CBC!
Is my oracle dumb? Why is CBC mode churnign out identicle ciphertexts? After all you are not supposed to see penguins through it.

Let's investigate.

Repeated blocks in Ciphertext:

Plaintext Block content	Plaintext Block ID
b'\xaf\x90\xee\xe5\x84&i\xad\x0e\x81&\xc6\xd2\xb5\x11\xc0'	52
b'\xaf\x90\xee\xe5\x84&i\xad\x0e\x81&\xc6\xd2\xb5\x11\xc0'	98

This means,

$$\begin{gathered} C_{52} = E_k(P_{52} \oplus C_{51} \\ C_{98} = E_k(P_{98} \oplus C_{97}) \end{gathered}$$

Since,

$$\begin{gathered} C_{52} = C_{98} \\ \Rightarrow E_k(P_{52} \oplus C_{51}) = E_k(P_{98} \oplus C_{97}) \end{gathered}$$

This would be possible only if,

$$P_{52} \oplus C_{51} = P_{98} \oplus C_{97}$$

We can see,

$$P_{52} = P_{98}$$

We also see a patterns, identical plaintexts produce identical ciphertexts only of the preceding plaintext blocks were also identical.

XOR MAGIC??
XOR is associative. That means:

$$A \oplus (B \oplus C) = (A \oplus B) \oplus C$$

It's also commutative:

$$\begin{gathered} A = B \oplus C \\ B = A \oplus C \\ C = B \oplus A \end{gathered}$$

My intuition declares it's because of this property of XOR, but i'm just a peasant CS major afraid of math, send help.

> takes math courses next sem :)