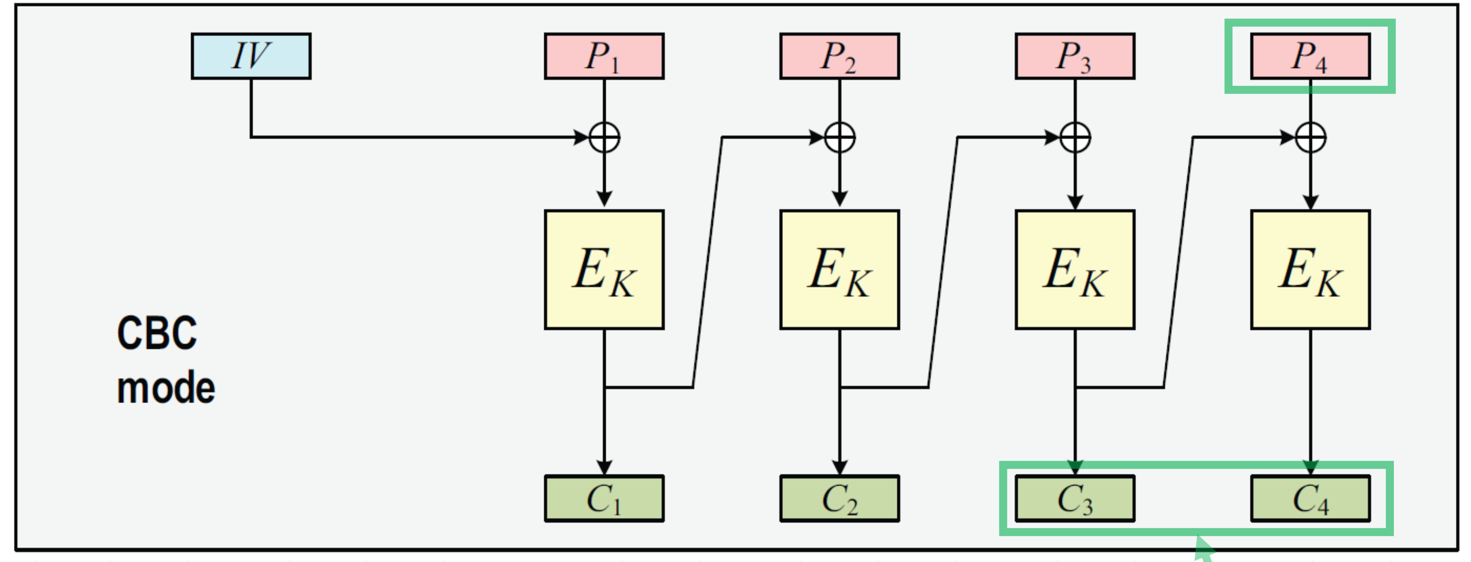
Padding Oracle Attack

Chanathip Namprempre

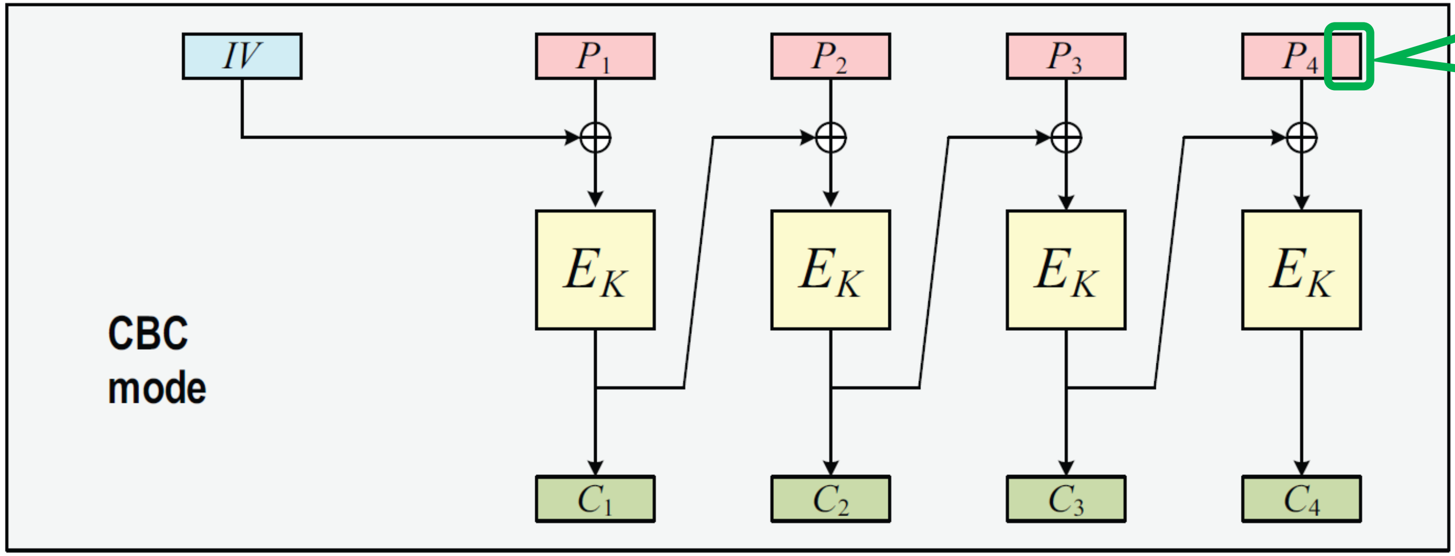
Recall CBC Mode



To decrypt one message block, we only need two ciphertext blocks.

Source of all drawings of modes: https://www.cs.ucdavis.edu/~rogaway/papers/modes.pdf

CBC Mode with Padding (simplified)



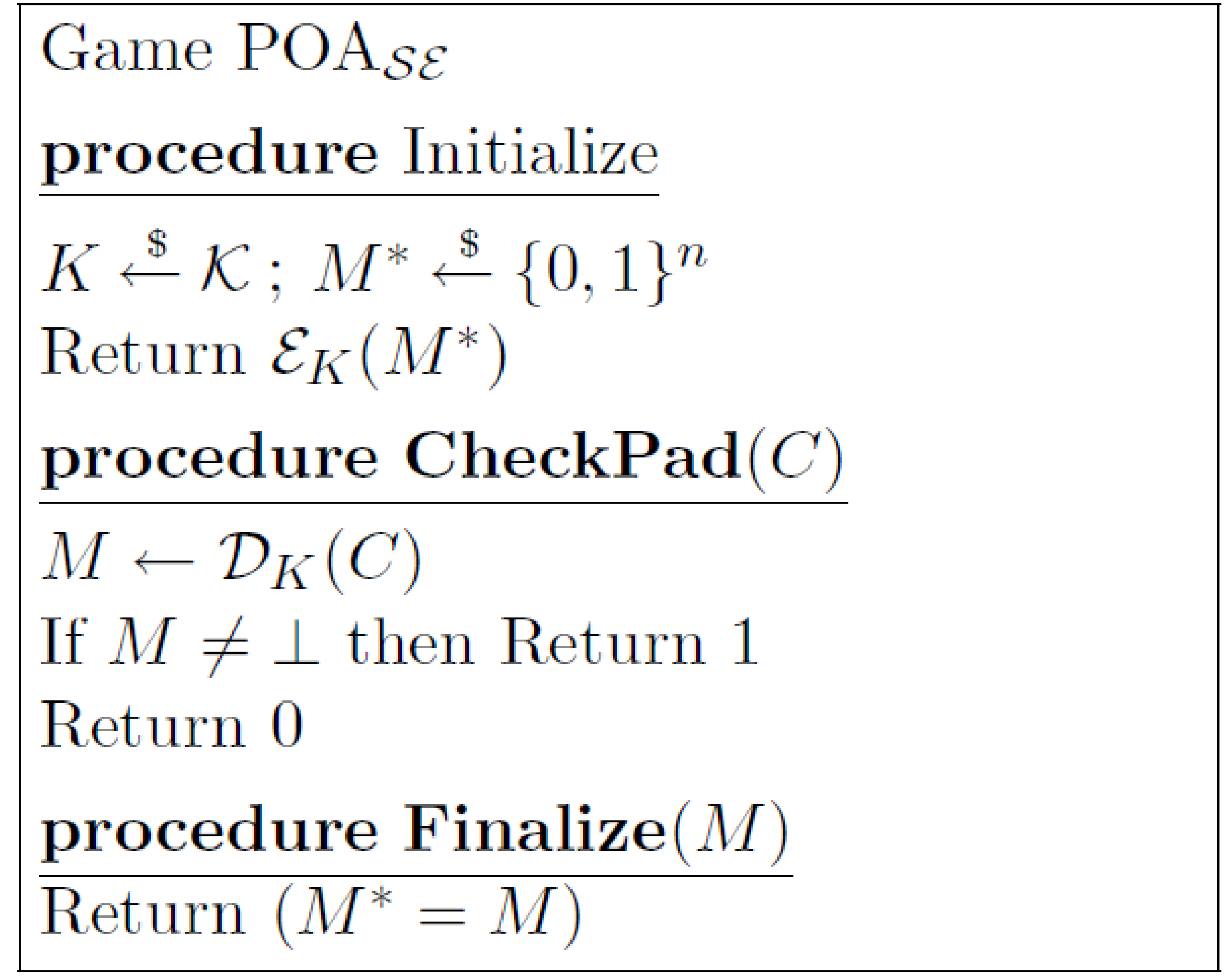
0x03 0x03 0x03
if padding with 3 bytes

Valid pads:

- 0x01
- 0x02 0x02
- 0x03 0x03 0x03
- …

Source of all drawings of modes: https://www.cs.ucdavis.edu/~rogaway/papers/modes.pdf

Padding Oracle Attack: Formal Definition



```
Game $\mathrm{POA}_{\mathcal{SE}}$

procedure Initialize
    $K \xleftarrow{\$} \mathcal{K}\,;\ M^* \xleftarrow{\$} \{0,1\}^n$
    Return $\mathcal{E}_K(M^*)$

procedure CheckPad($C$)
    $M \leftarrow \mathcal{D}_K(C)$
    If $M \neq \bot$ then Return 1
    Return 0

procedure Finalize($M$)
    Return $(M^* = M)$
```

Source:

Bellare and Rogaway, unpublished notes.

Fun Video: CBC padding oracle attack

Attacking Modern Cryptography

https://youtu.be/8Tr2aj6JETg

3:57

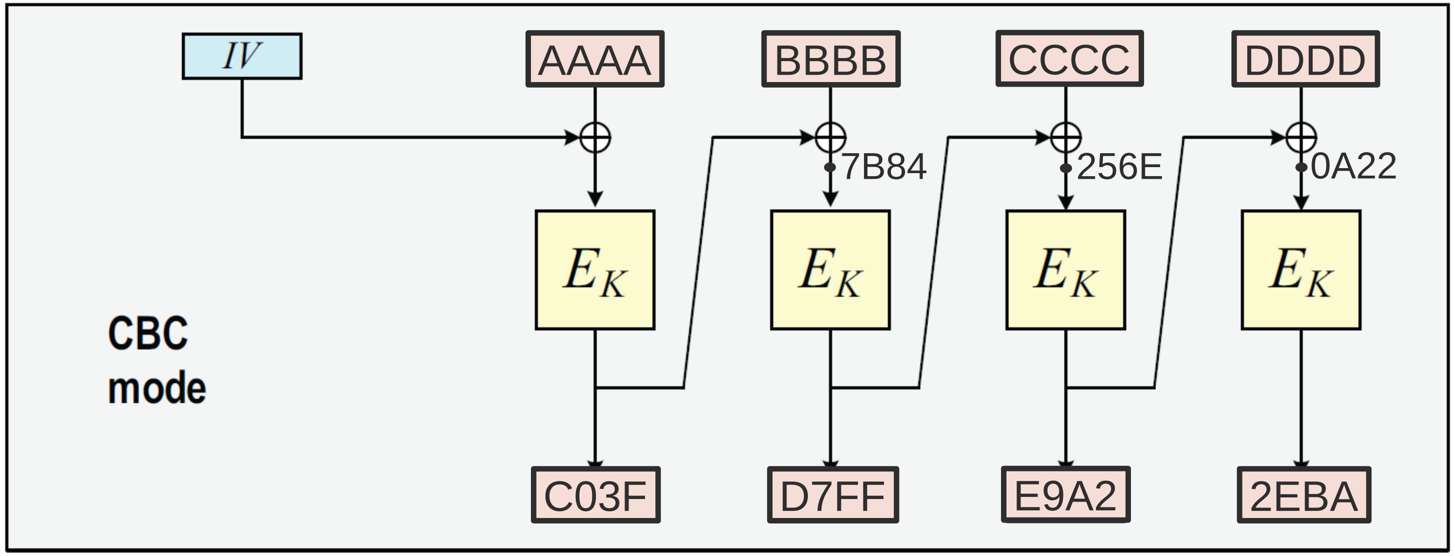
By Pastie's Bin

(Recommendation: watch it at reduced speed!)

The attack described here is applicable to SSLv3.

It was originally pointed out in a 1997 paper by David Wagner and Bruce Schneier.

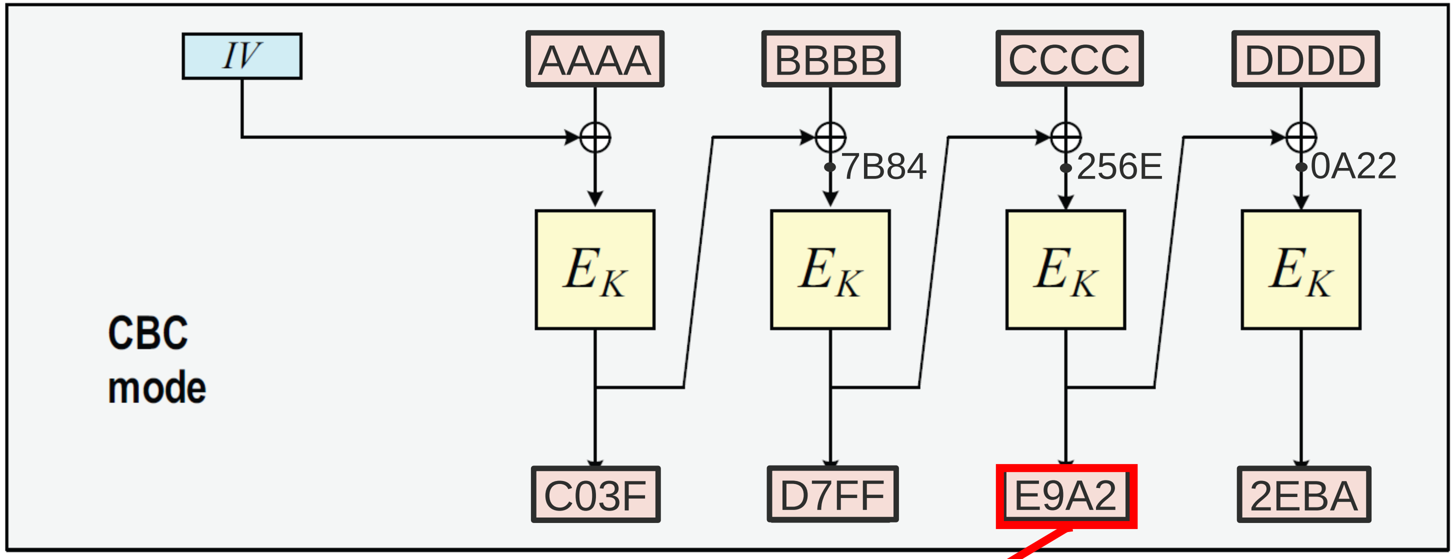
Redrawing pictures from the video



The video does not show the IV.

Source of all drawings of modes: https://www.cs.ucdavis.edu/~rogaway/papers/modes.pdf

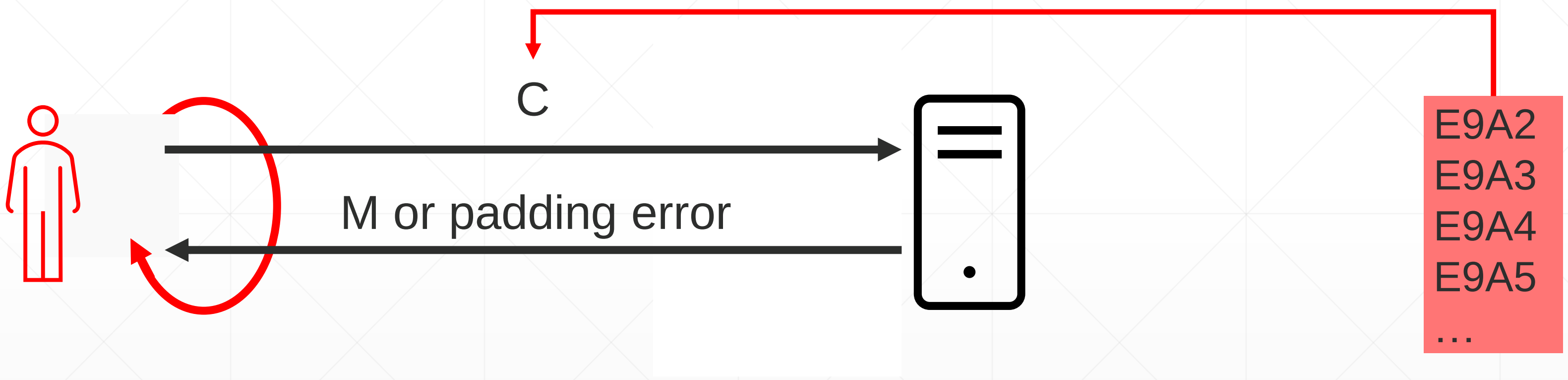
Redrawing pictures from the video



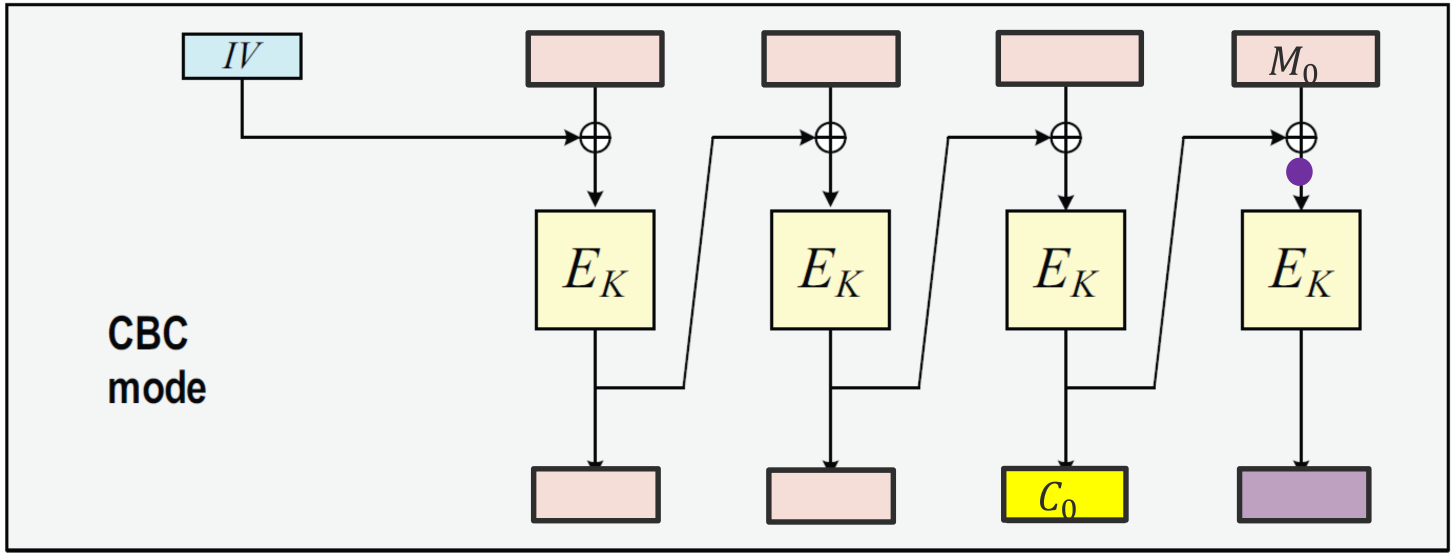
In the video, we cycle through this value one by one from right to left.

Source of all drawings of modes: https://www.cs.ucdavis.edu/~rogaway/papers/modes.pdf

What does a padding oracle attack look like?

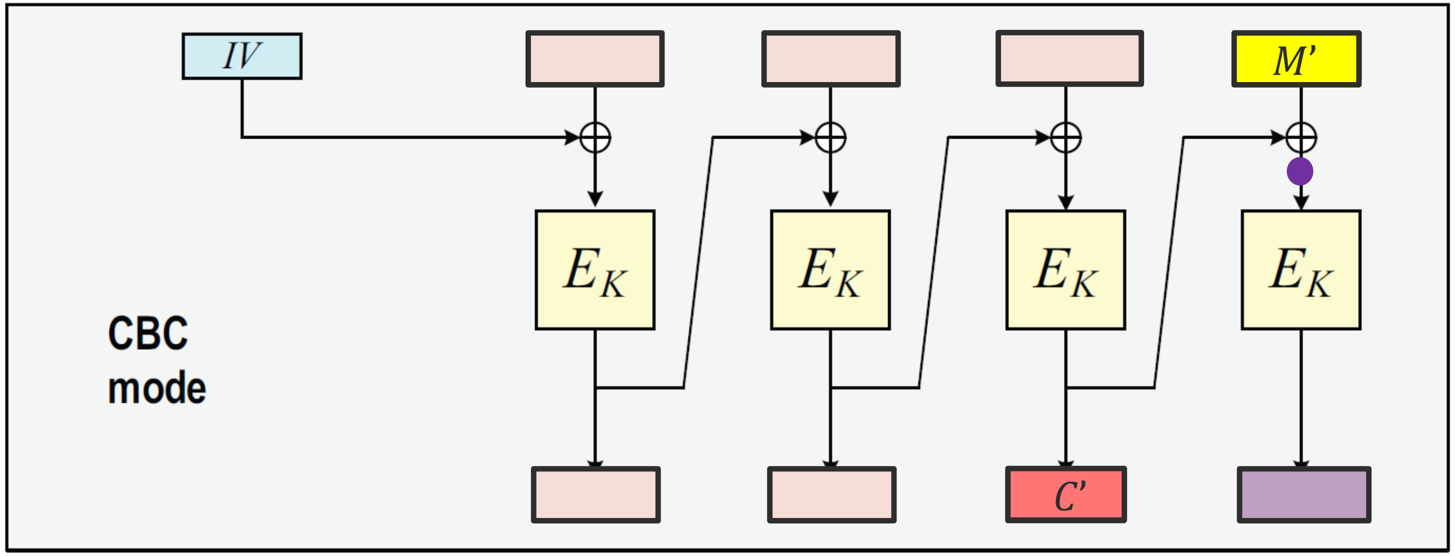


We want to find M_0



Source of all drawings of modes: https://www.cs.ucdavis.edu/~rogaway/papers/modes.pdf

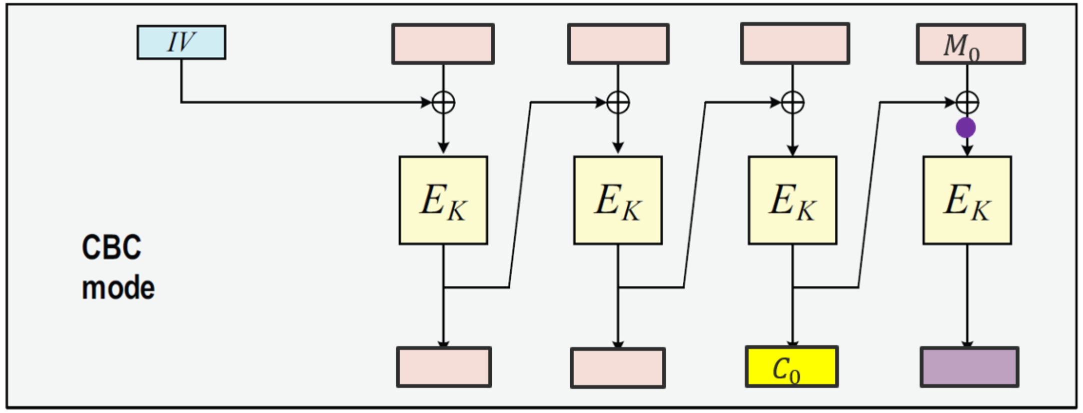
The Main Point

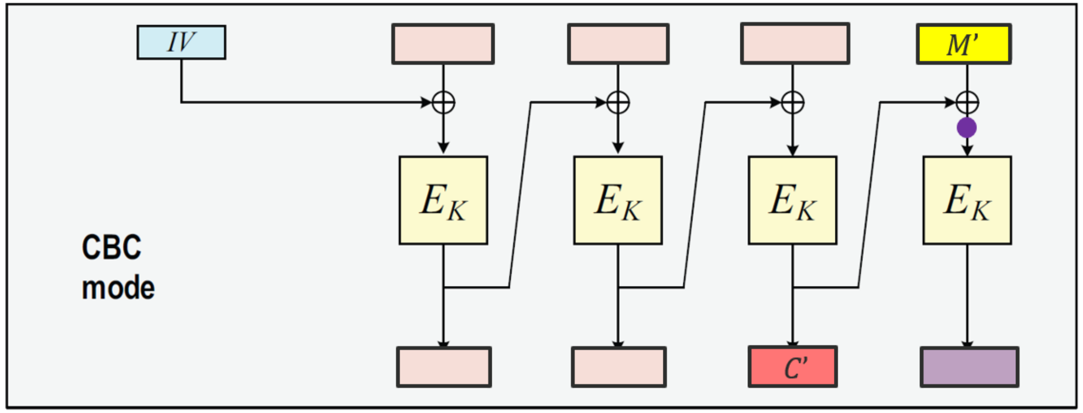


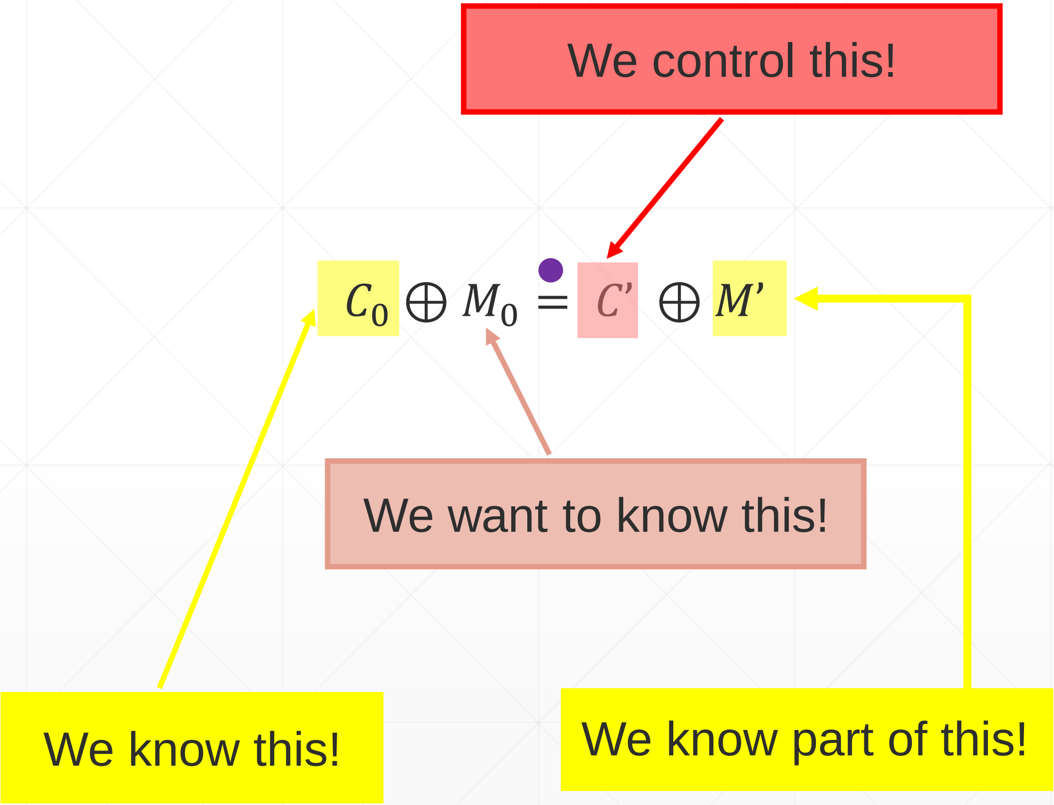
We modify C_0 to get C' and observe the server's response, which depends on M'.

Source of all drawings of modes: https://www.cs.ucdavis.edu/~rogaway/papers/modes.pdf

In summary







We control this!

$C_0 \oplus M_0 \overset{\bullet}{=} C' \oplus M'$

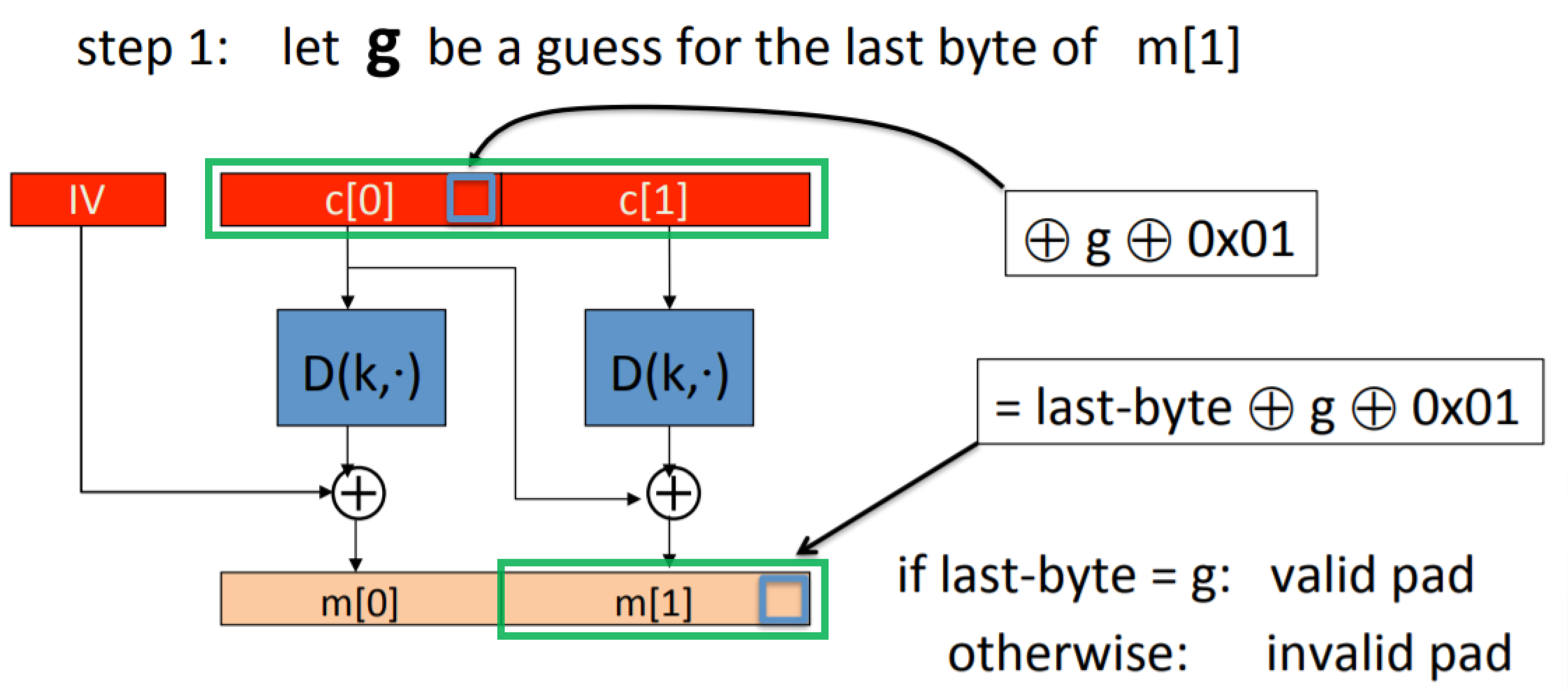
We want to know this!

We know this!

We know part of this!

Source of all drawings of modes: https://www.cs.ucdavis.edu/~rogaway/papers/modes.pdf

Strategy: Guess one byte at a time



step 1: let **g** be a guess for the last byte of m[1]

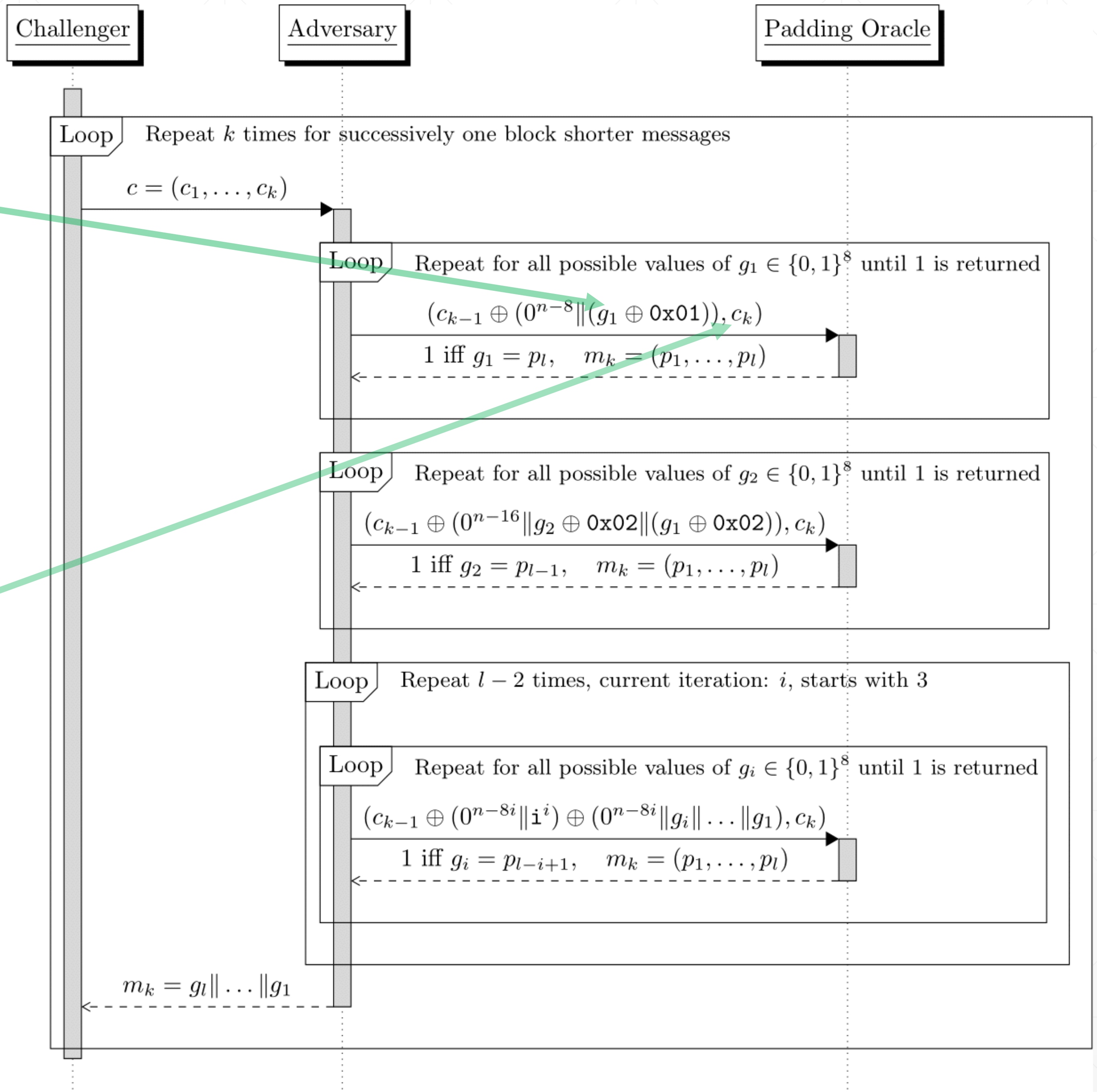
if last-byte = g: valid pad

otherwise: invalid pad

Source:

Dan Boneh’s coursera slides: https://course.ece.cmu.edu/~ece733/lectures/07-authenc-part2.pdf

UML for the Padding Oracle Attack against CBC



Challenger
Adversary
Padding Oracle
Loop
Repeat k times for successively one block shorter messages
$c = (c_1, \ldots, c_k)$
Loop
Repeat for all possible values of $g_1 \in \{0,1\}^8$ until 1 is returned
$(c_{k-1} \oplus (0^{n-8}\|(g_1 \oplus \mathtt{0x01})), c_k)$
1 iff $g_1 = p_l, \quad m_k = (p_1, \ldots, p_l)$
Loop
Repeat for all possible values of $g_2 \in \{0,1\}^8$ until 1 is returned
$(c_{k-1} \oplus (0^{n-16}\|g_2 \oplus \mathtt{0x02}\|(g_1 \oplus \mathtt{0x02})), c_k)$
1 iff $g_2 = p_{l-1}, \quad m_k = (p_1, \ldots, p_l)$
Loop
Repeat $l-2$ times, current iteration: i, starts with 3
Loop
Repeat for all possible values of $g_i \in \{0,1\}^8$ until 1 is returned
$(c_{k-1} \oplus (0^{n-8i}\|\mathtt{i}^i) \oplus (0^{n-8i}\|g_i\|\ldots\|g_1), c_k)$
1 iff $g_i = p_{l-i+1}, \quad m_k = (p_1, \ldots, p_l)$
$m_k = g_l\|\ldots\|g_1$

byte

block

Source:

Reply in https://crypto.stackexchange.com/questions/70570/how-does-the-cbc-padding-oracle-attack-work-in-general by “SEJPM”