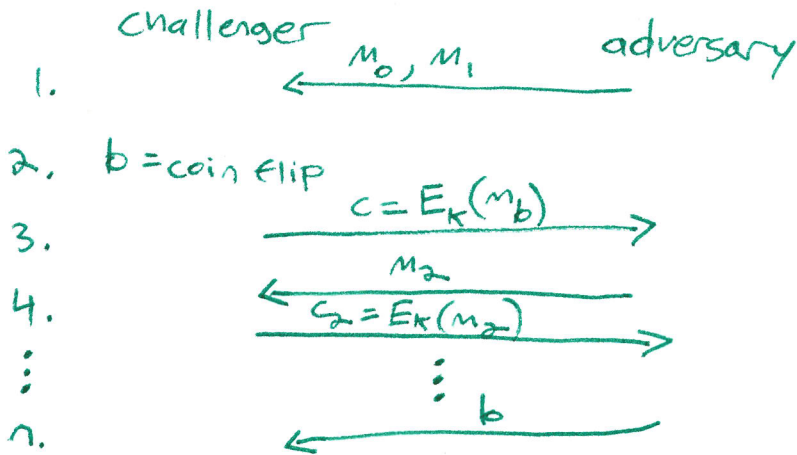


IND-CPA

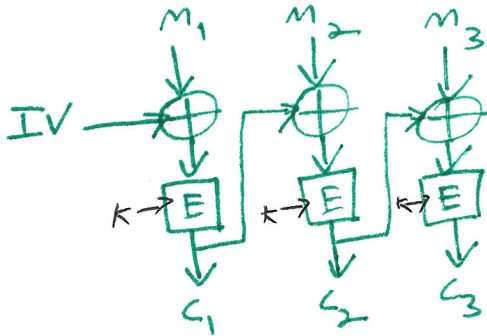
intuition: adversary can query encryption oracle

Game



CBC mode

Recall:



Theorem: CBC with incremental IV is not CPA secure.

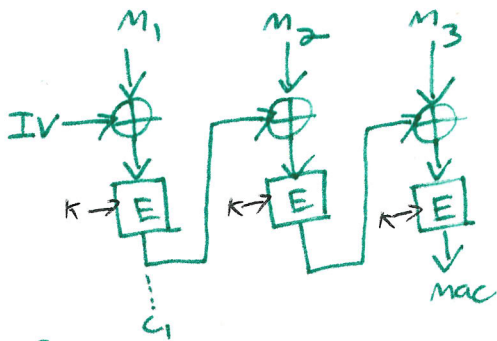
Proof sketch:

- Adv. gets $c = E_K(m_0 \oplus 1)$ or $c = E_K(m_1 \oplus 1)$
- Adv. queries $m_2 = m_0 \oplus 1 \oplus 2$
- Adv. gets $c_2 = E_K(m_2 \oplus 2) = E_K(m_0 \oplus 1 \oplus 2 \oplus 2) = E_K(m_0 \oplus 1)$
- $b = c_2 = c$

Integrity:

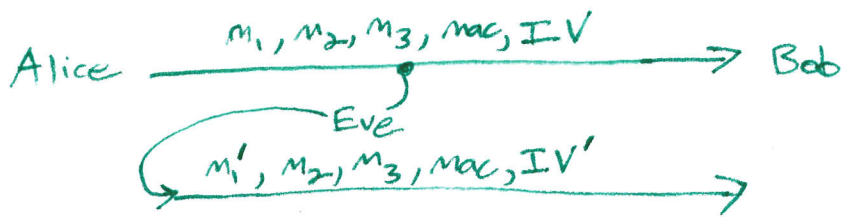
Alice $\xrightarrow{m_1, m_2, m_3, mac}$ Bob

Recall: CBC-MAC



Theorem: CBC-MAC with randomized IV is not secure.

Proof sketch:



- $c_1 = E_K(m_1 \oplus IV)$
- $c_1' = E_K(m_1' \oplus IV')$
- want $c_1 = c_1'$
- solve for IV' : $m_1' \oplus IV' = m_1 \oplus IV$
 $\Rightarrow IV' = m_1' \oplus m_1 \oplus IV$

Padding oracle attacks

Padding scheme

n bytes of padding: $\overbrace{1, 1, 1, 1, \dots, 1}^{n \text{ times}}$

✓ $0x04, 0x04, 0x04, 0x04$

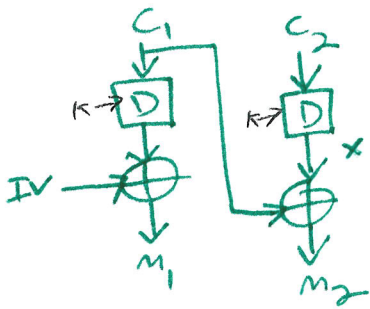
✓ $0x01$

X $0x49, 0x02$

Padding oracle

Given ciphertext, return "ok" if decrypted message has valid padding or "error" otherwise.

Recall: CBC decryption



Padding oracle attack

$$m_2[15] = x[15] \oplus c_1[15]$$

$$c'_1[15] = c_1[15] \oplus 0 \oplus 0x01$$

$$\begin{aligned} m'_2[15] &= \cancel{x[15]} \oplus c'_1[15] \\ &= x[15] \oplus c_1[15] \oplus 0 \oplus 0x01 \\ &= 0x01 \text{ if } 0 = m_2[15] \end{aligned}$$

same for 2nd to last byte:

$0x02, 0x02$

... 16×256 queries to decrypt whole block!