“Hollywood Style Decryption” on Block Cipher-CBC

Rizki Wicaksono / ilmuHacking.com

Rizki Wicaksono

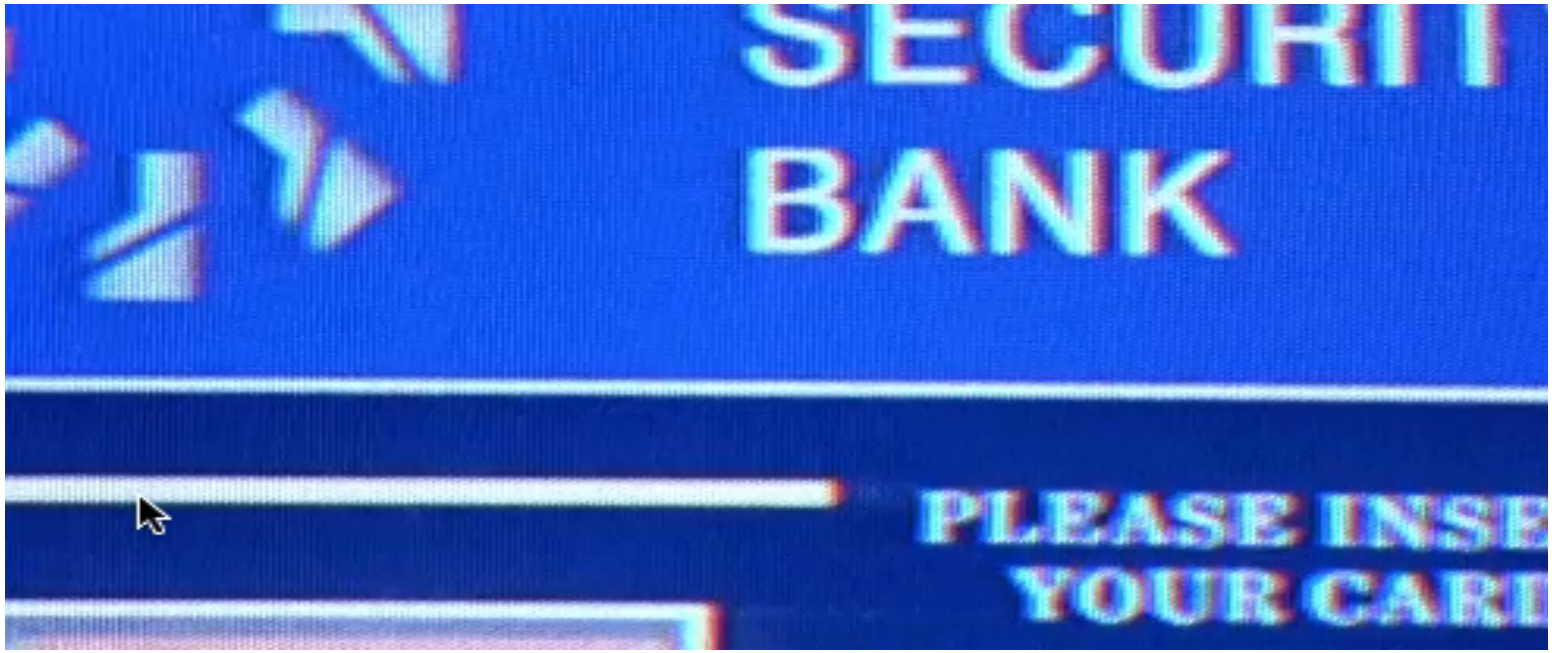
- Penetration tester
- Programming, application security, cryptography
- S1 Teknik Informatika ITB, ECSP, OSWP, ITIL-F
- ilmuHacking.com , facebook.com/ilmuHacking

Hollywood Style Password Cracking

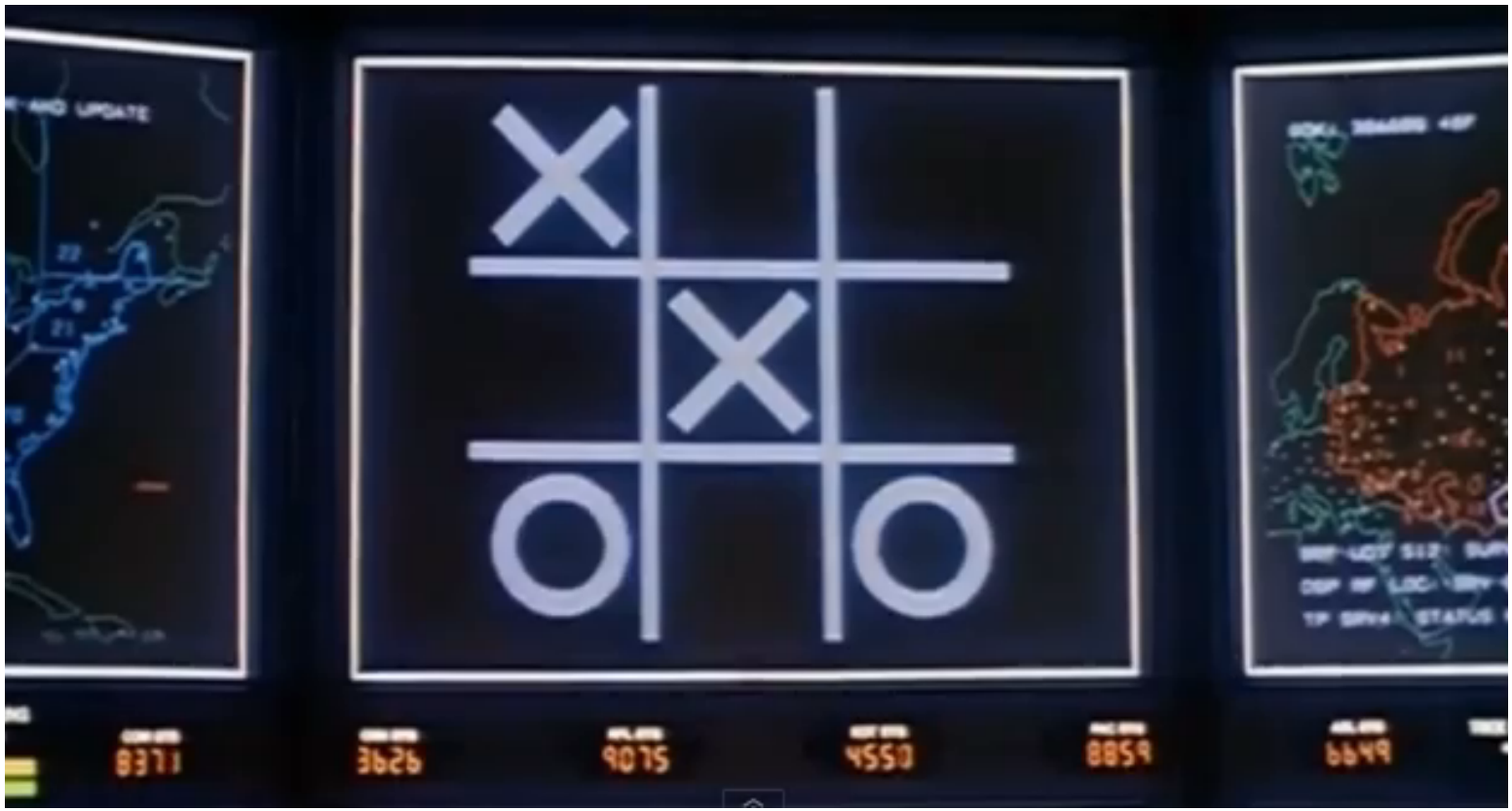
Lets Watch Some Movies

Resident Evil Breaking Door Key Scene

Terminator 2 ATM PIN Cracking Scene



Wargame Launch Code Hacking Scene

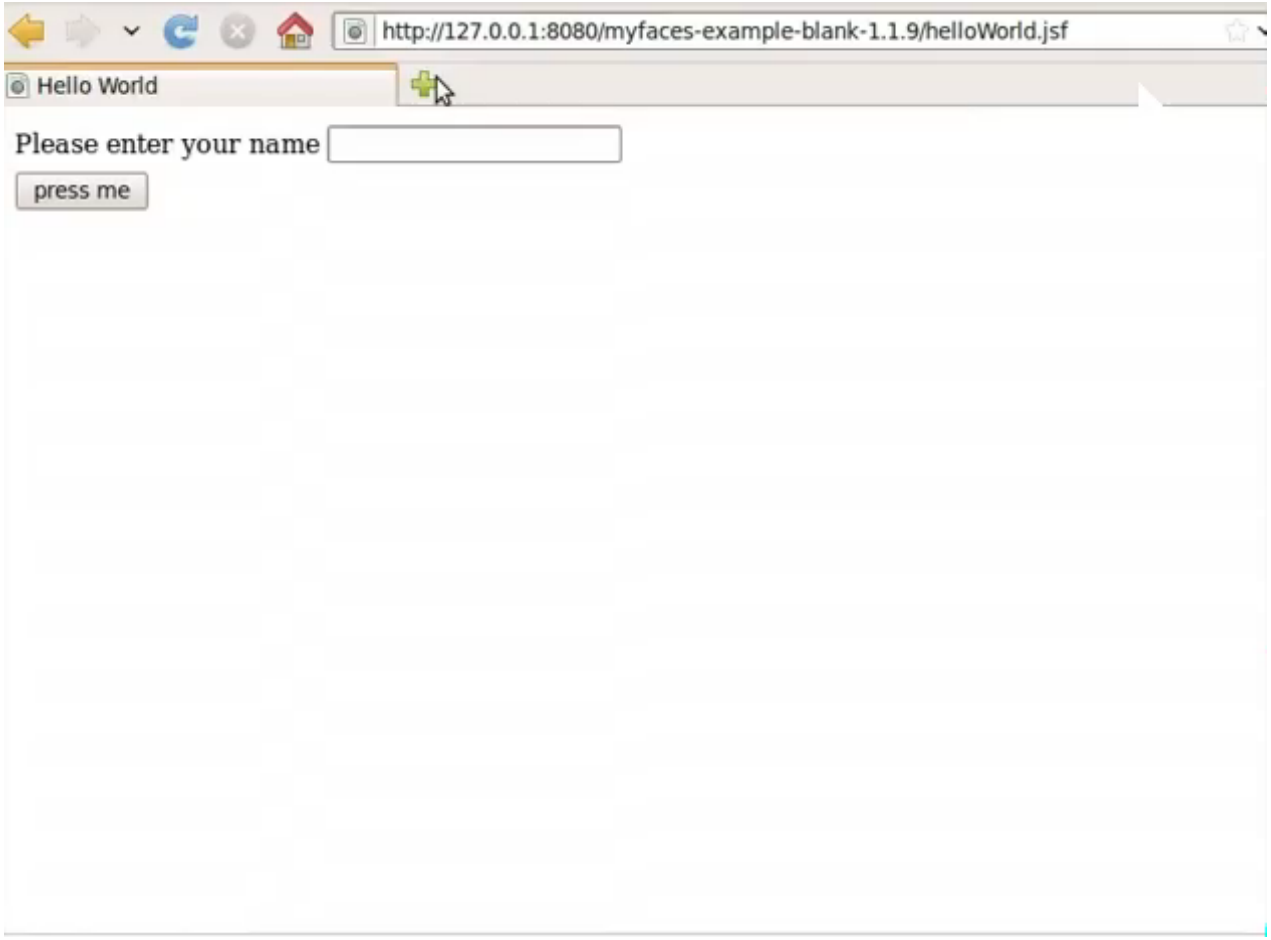


The Matrix Beginning Scene

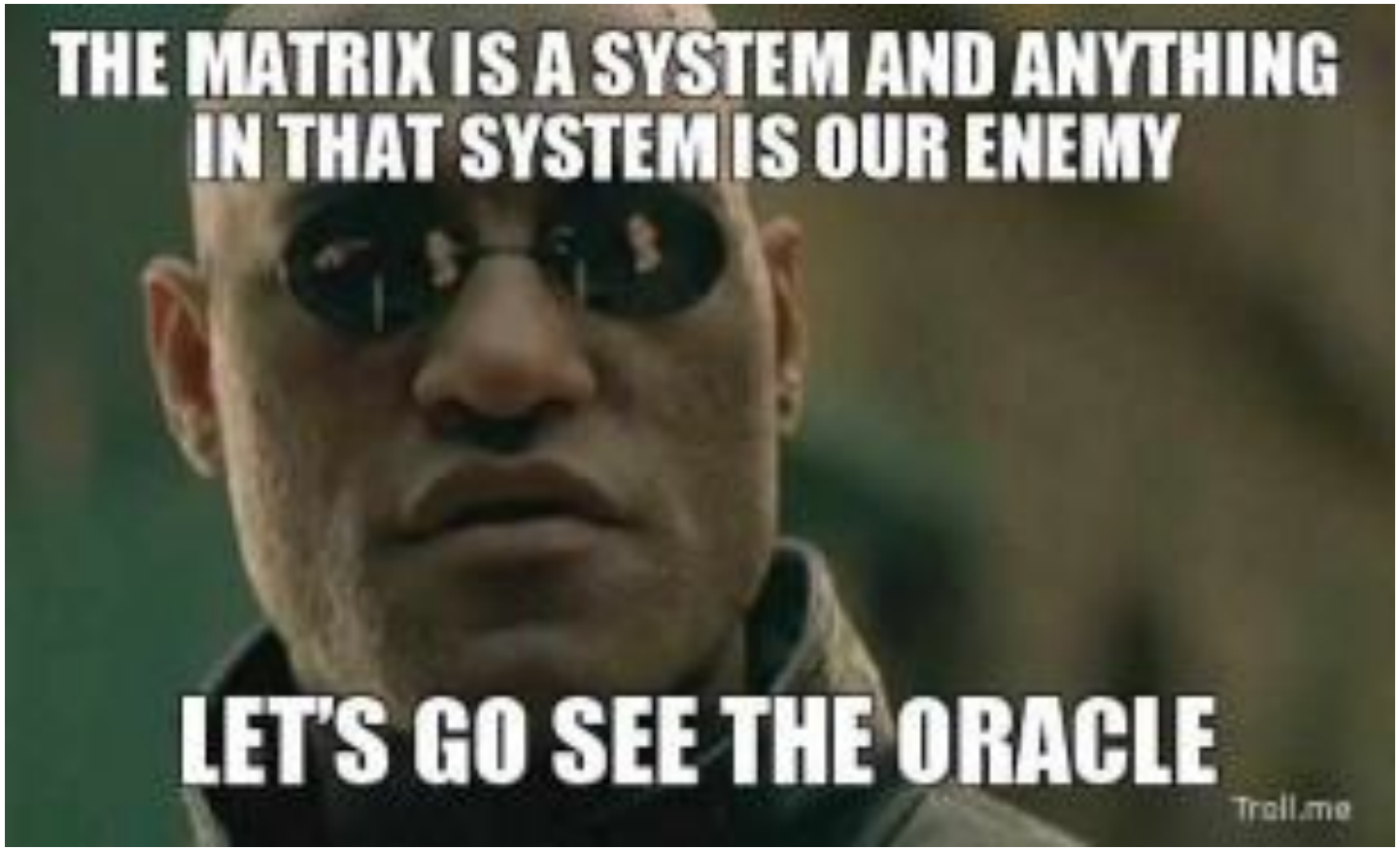
Bloodfist IV Passcode Breaking Scene

Hollywood Style Decryption with Padding Oracle Attack

Sample Real Attack

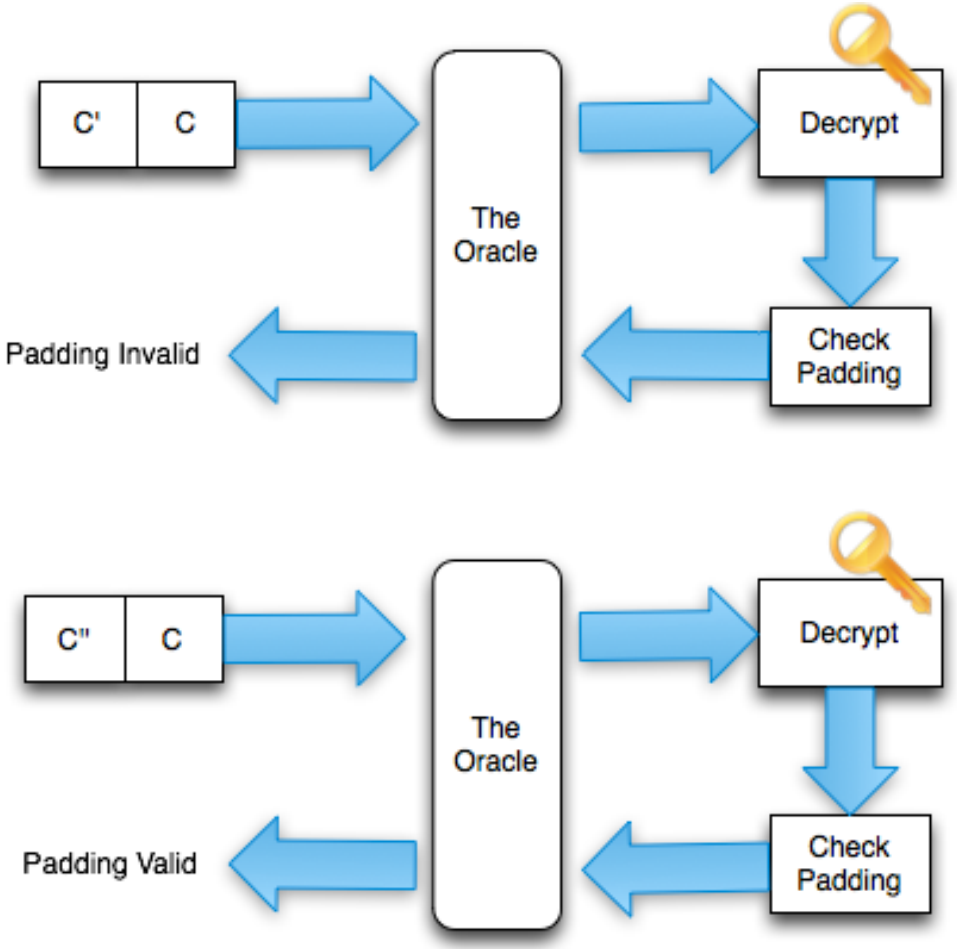


Morpheus: Let's Go See the Oracle



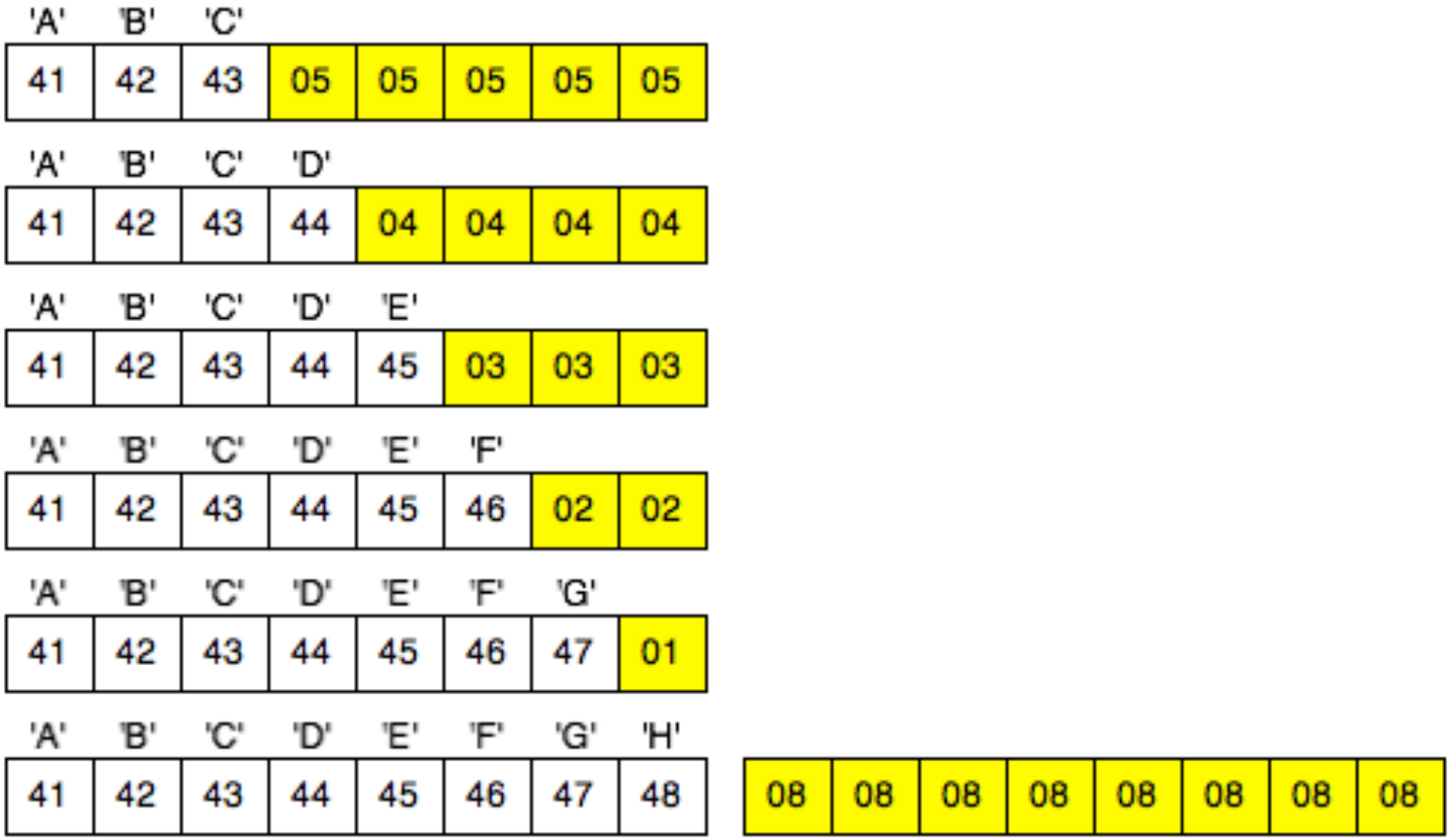
The Oracle

Padding Oracle: Valid/Invalid Pad

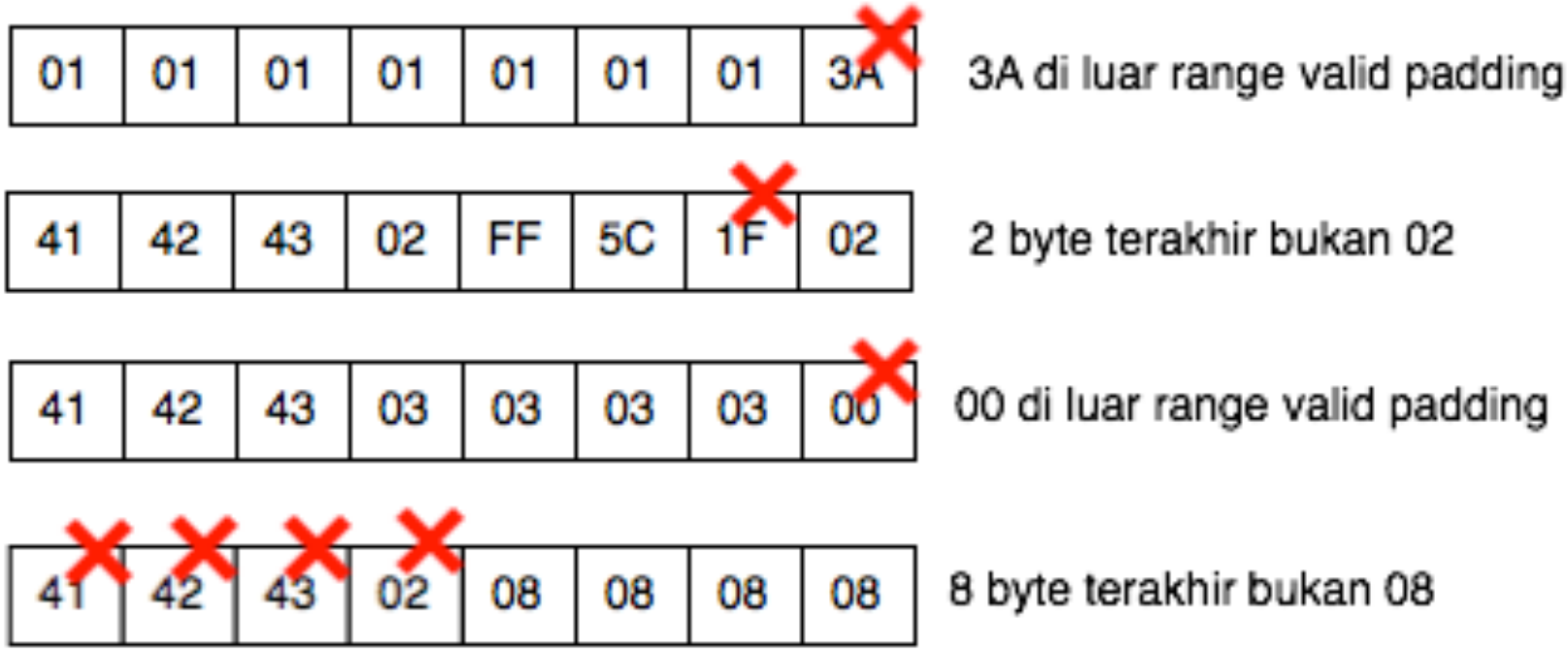


1 Bit Information Leakage

PKCS#7 Valid Padding



PKCS#7 Invalid Padding

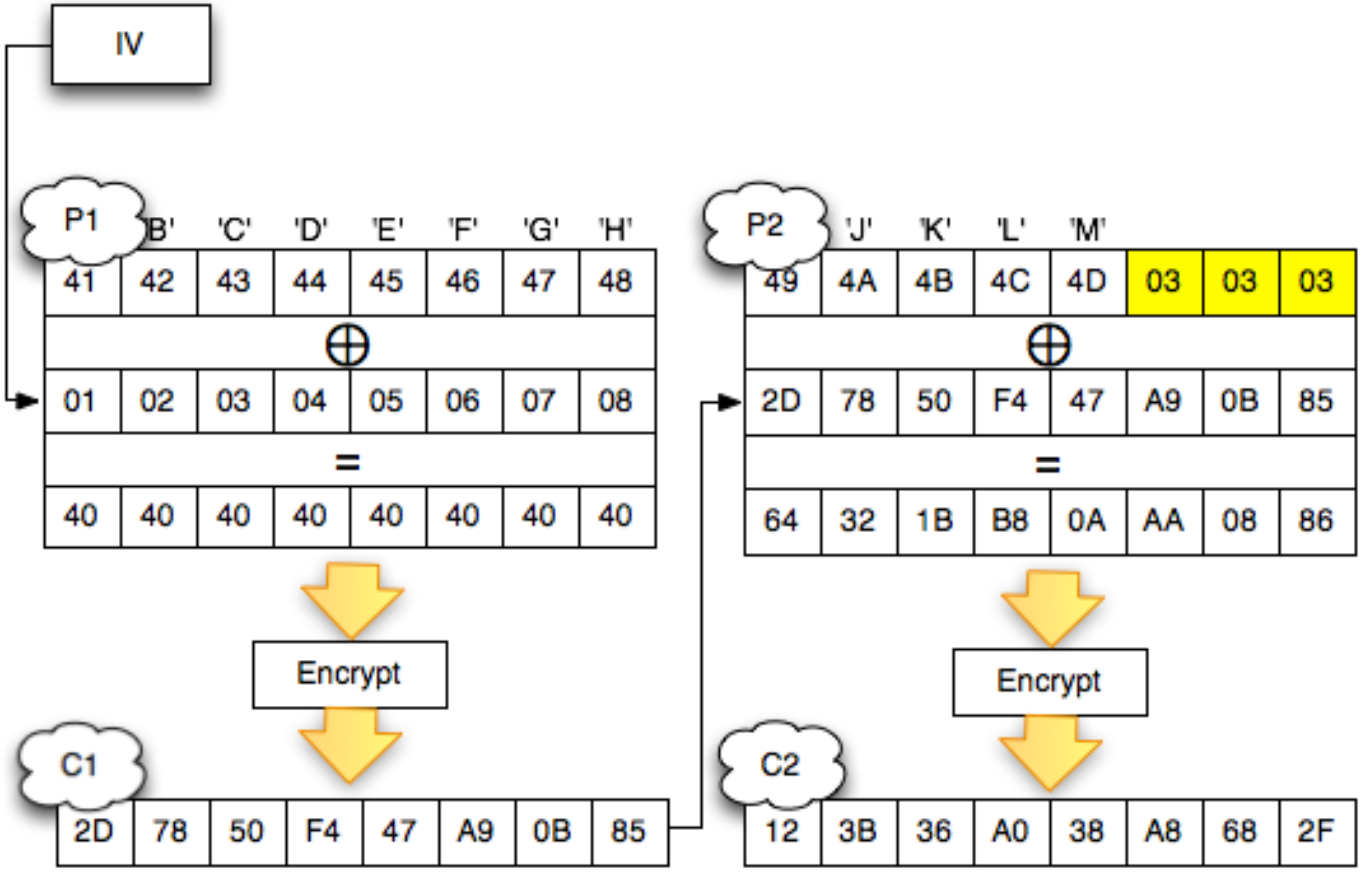


Cipher Block Chaining

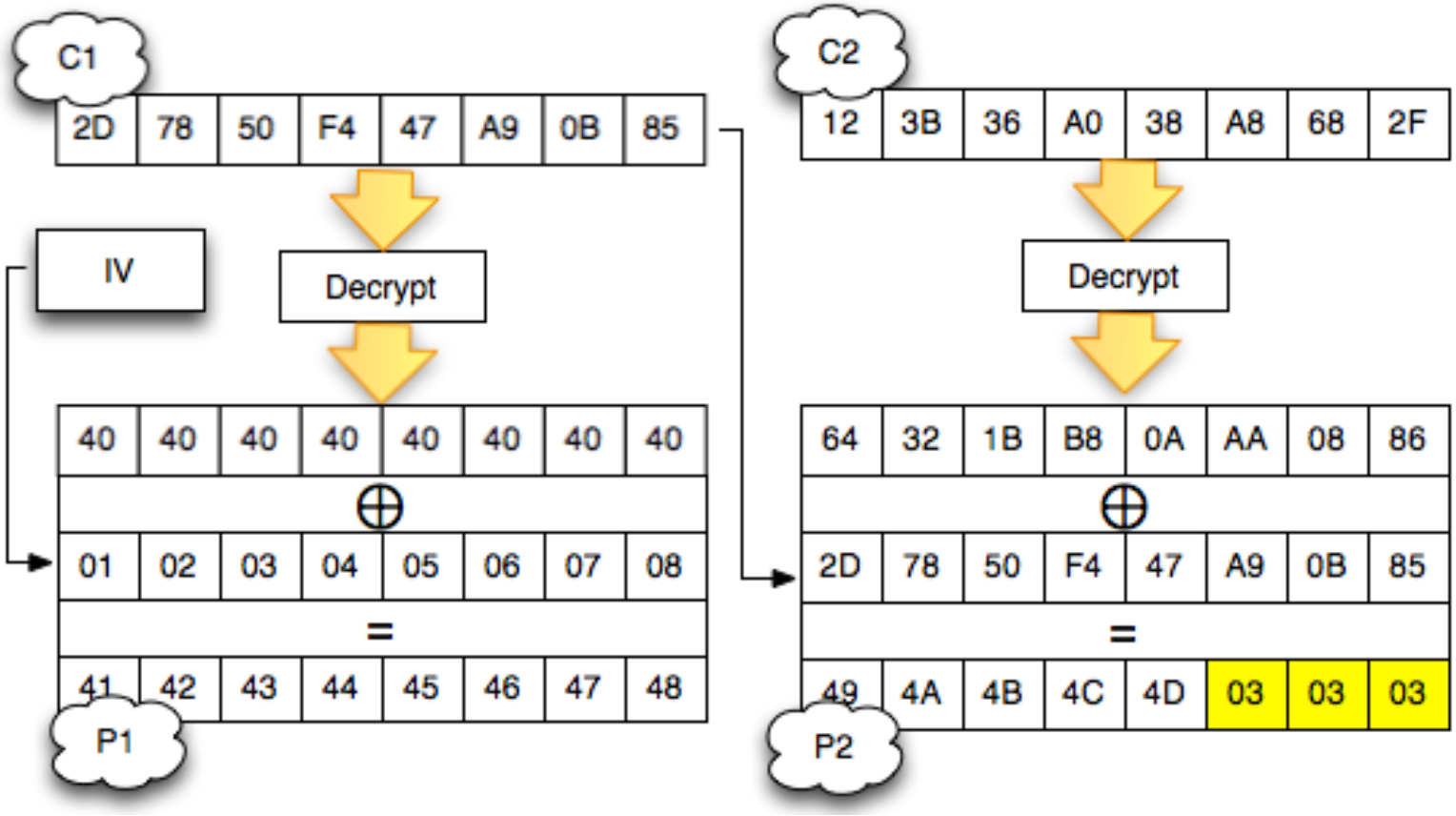
$$C_1 = E_k(P_1 \oplus IV) \qquad P_1 = D_k(C_1) \oplus IV$$
$$C_2 = E_k(P_2 \oplus C_1) \qquad P_2 = D_k(C_2) \oplus C_1$$
$$C_3 = E_k(P_3 \oplus C_2) \qquad P_3 = D_k(C_3) \oplus C_2$$
$$C_4 = E_k(P_4 \oplus C_3) \qquad P_4 = D_k(C_4) \oplus C_3$$

$$C_n = E_k(P_n \oplus C_{n-1}) \qquad P_n = D_k(C_n) \oplus C_{n-1}$$

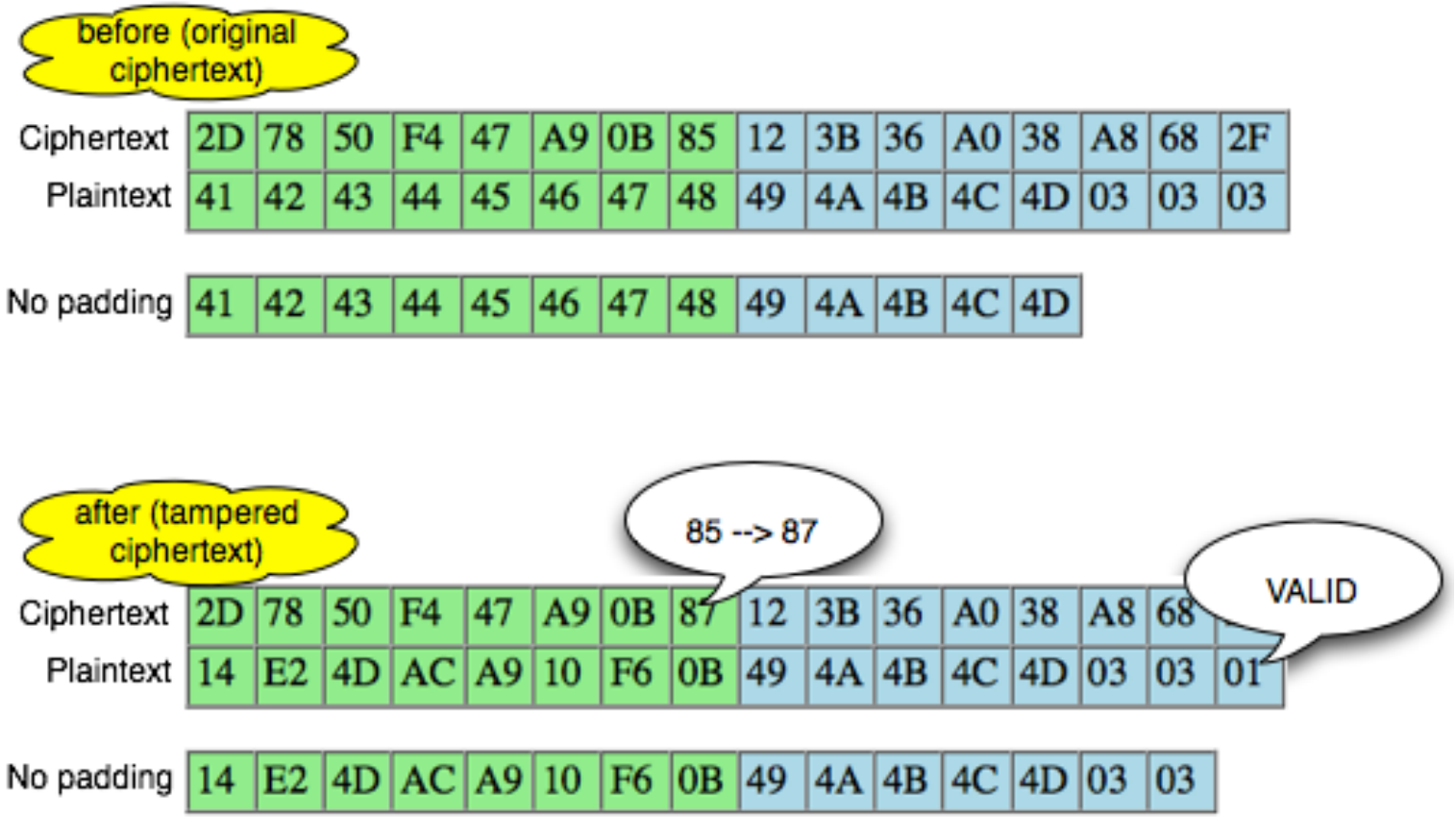
CBC Mode Encryption



CBC Mode Decryption



Malleability

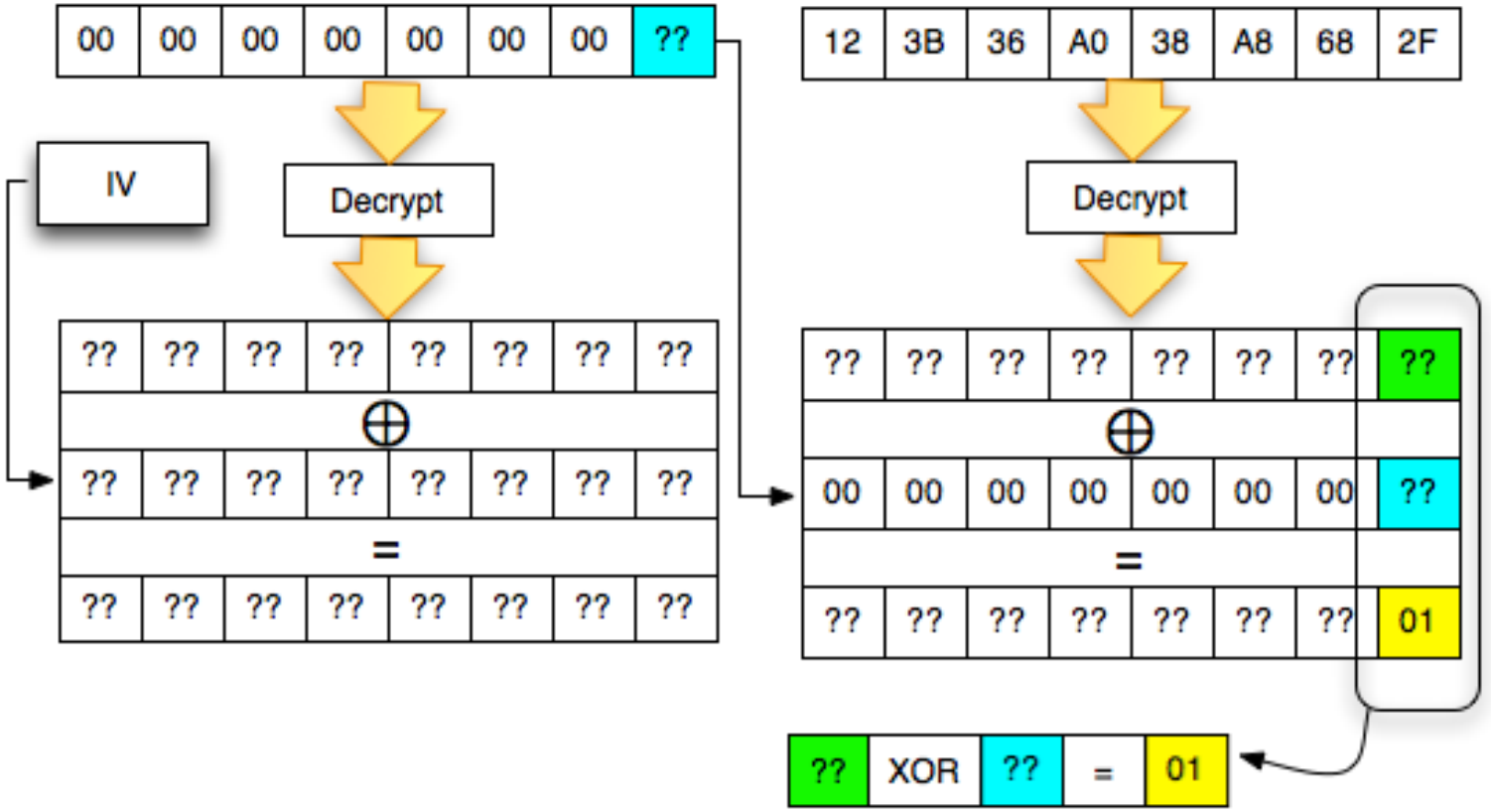


Enough Talking, Start Cracking!

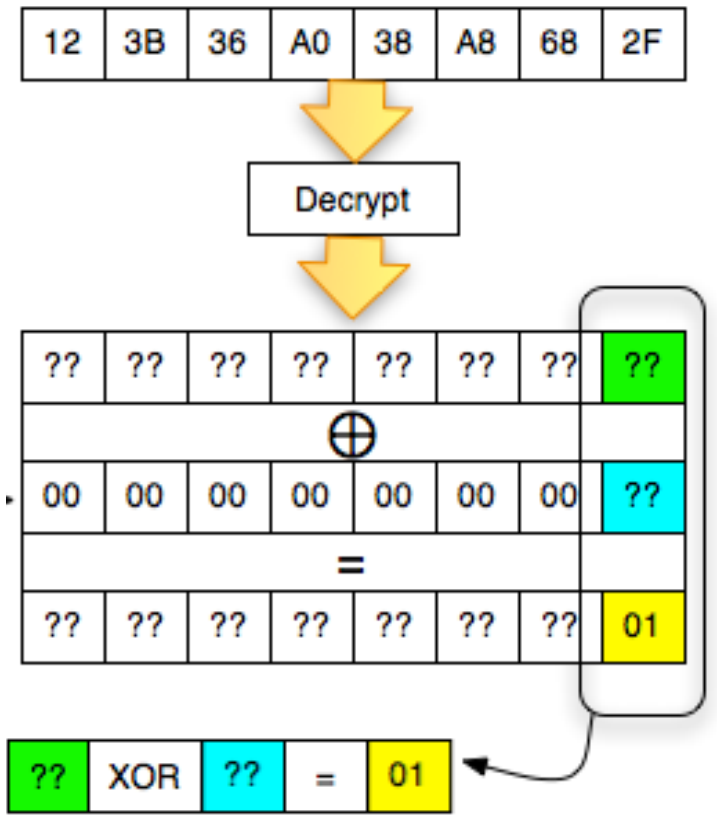
Sample Case

- Decrypt this: 2D7850F447A90B87123B36A038A8682F
- Split into two 8 byte blocks:
  - C_1 = 2D7850F447A90B87
  - C_2 = 123B36A038A8682F
- Decrypt C_2 first, send two block to oracle:
  - One block + 123B36A038A8682F
- Decrypt one byte at a time ("hollywood style") starting from the last byte

Decrypt Last Byte

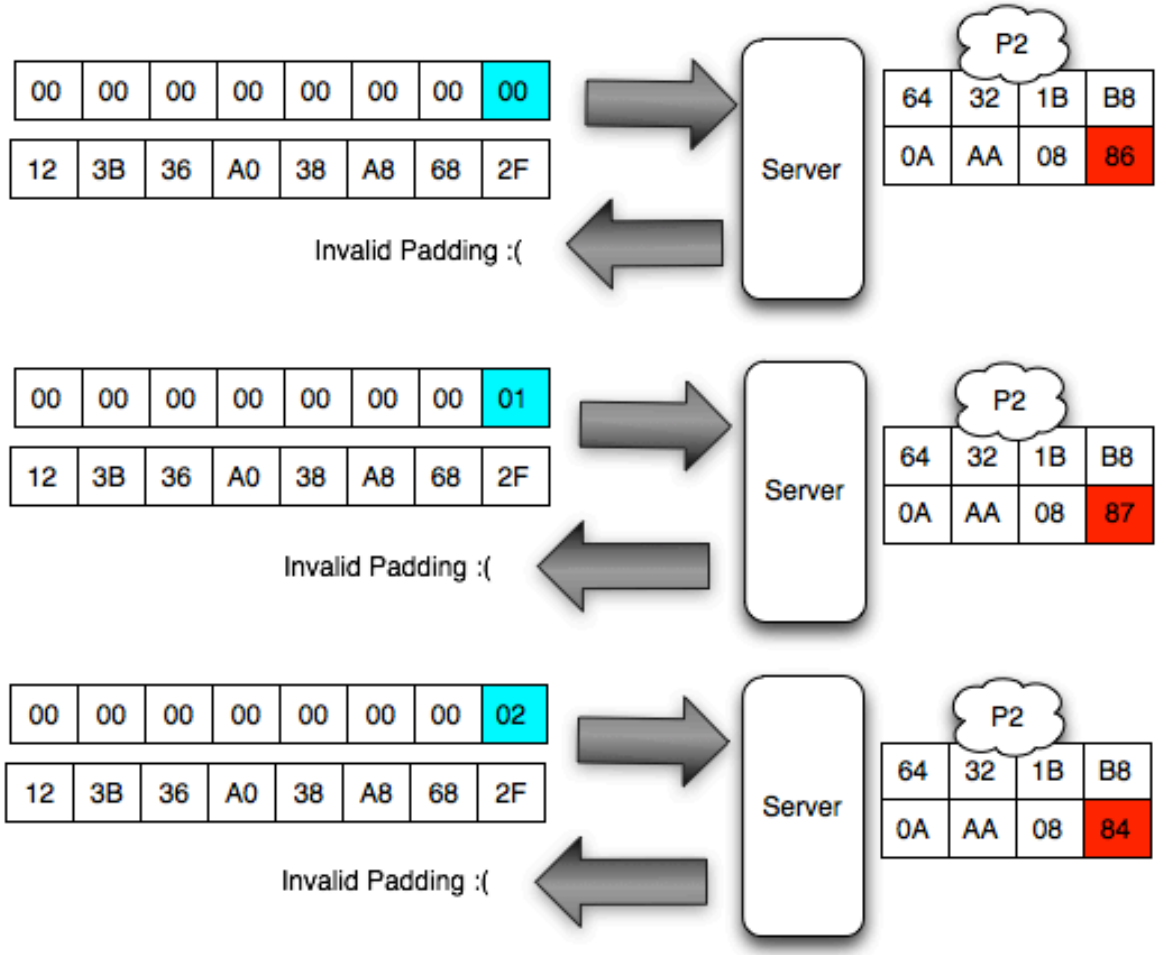


Ask the Oracle

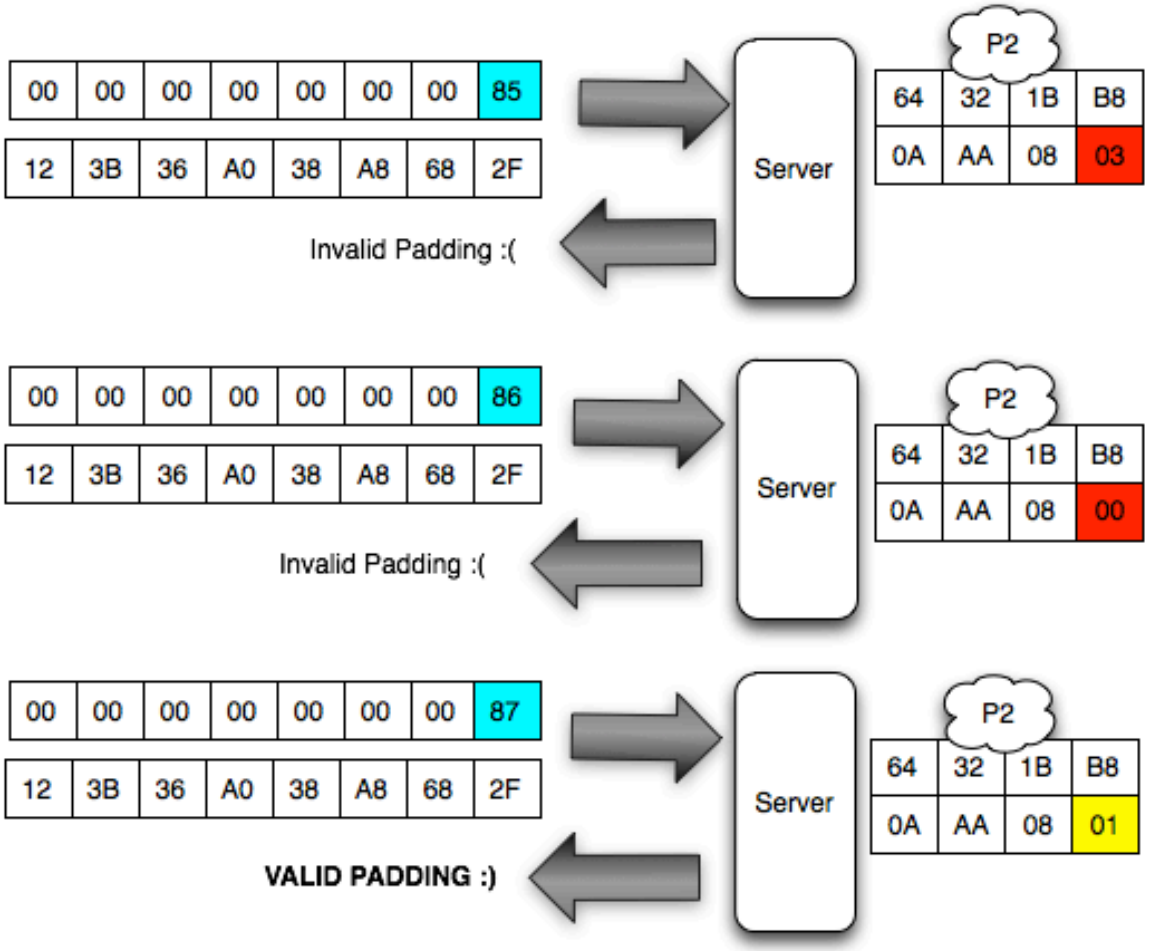


- A xor B = 01. Find A and B!
- Ask the Oracle:
  - A xor 0 = 01 ?
  - A xor 1 = 01 ?
  - ….
  - A xor 255 = 01 ?
- Oracle answer:
  - Valid pad = Yes
  - Invalid pad = No

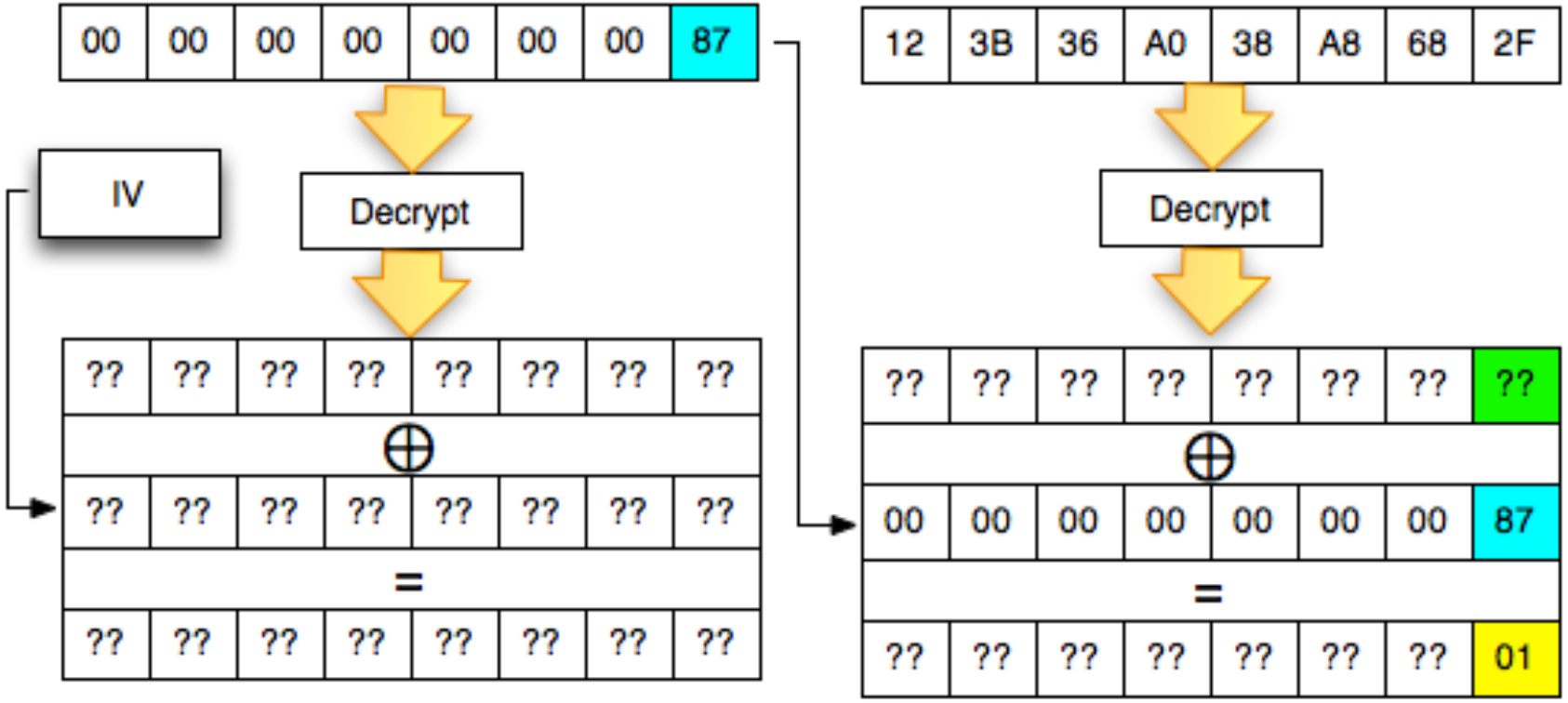
Look for Valid Single Byte Pad



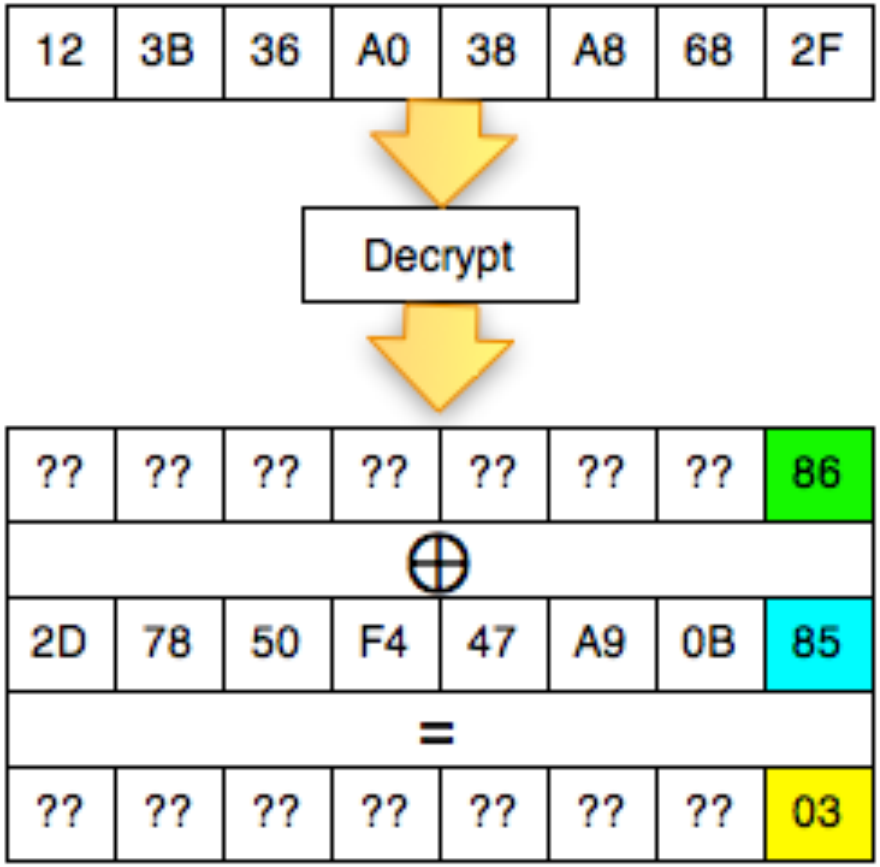
Valid Single Byte Pad Found!



Last Byte Decrypted



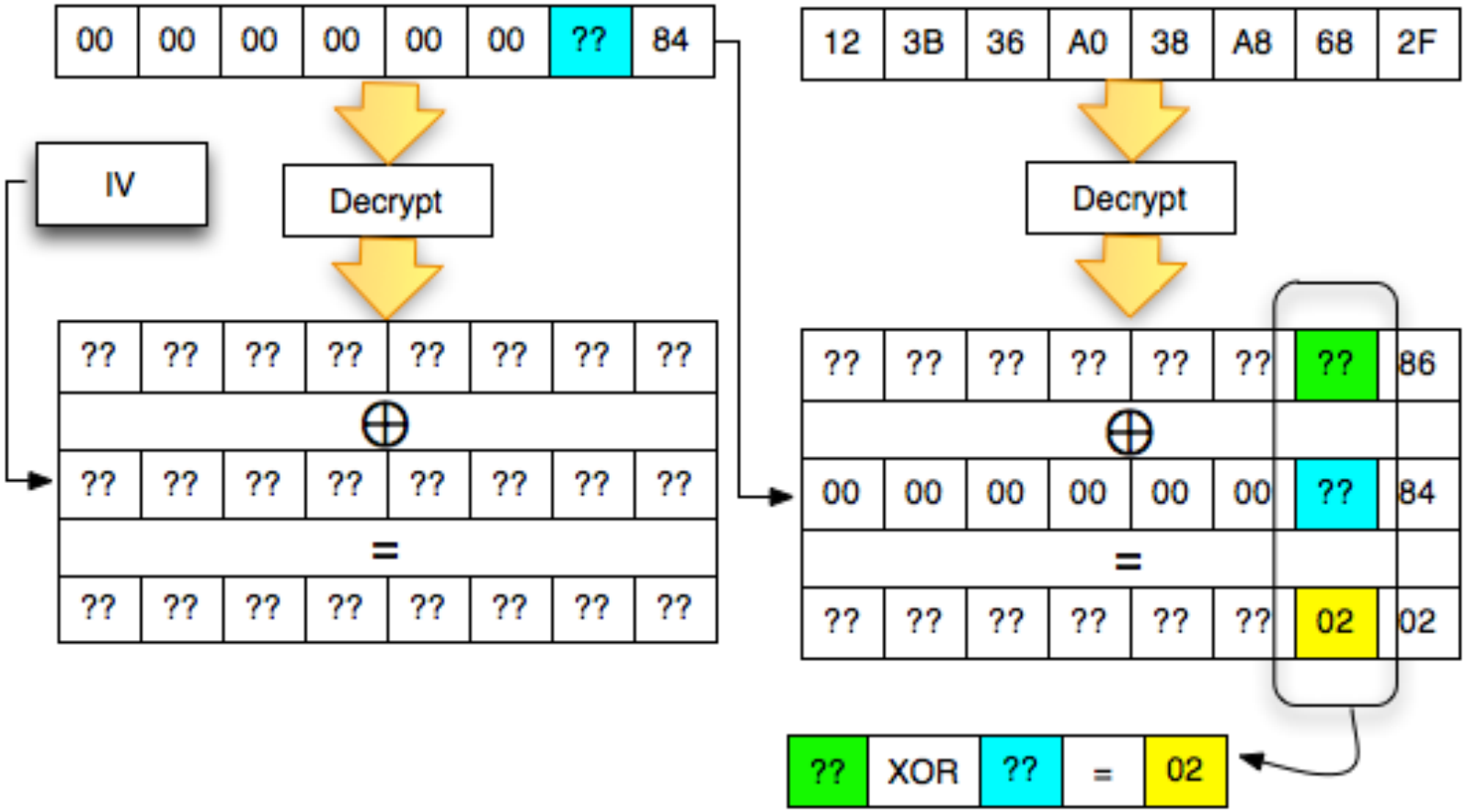
Last Byte = 0x86



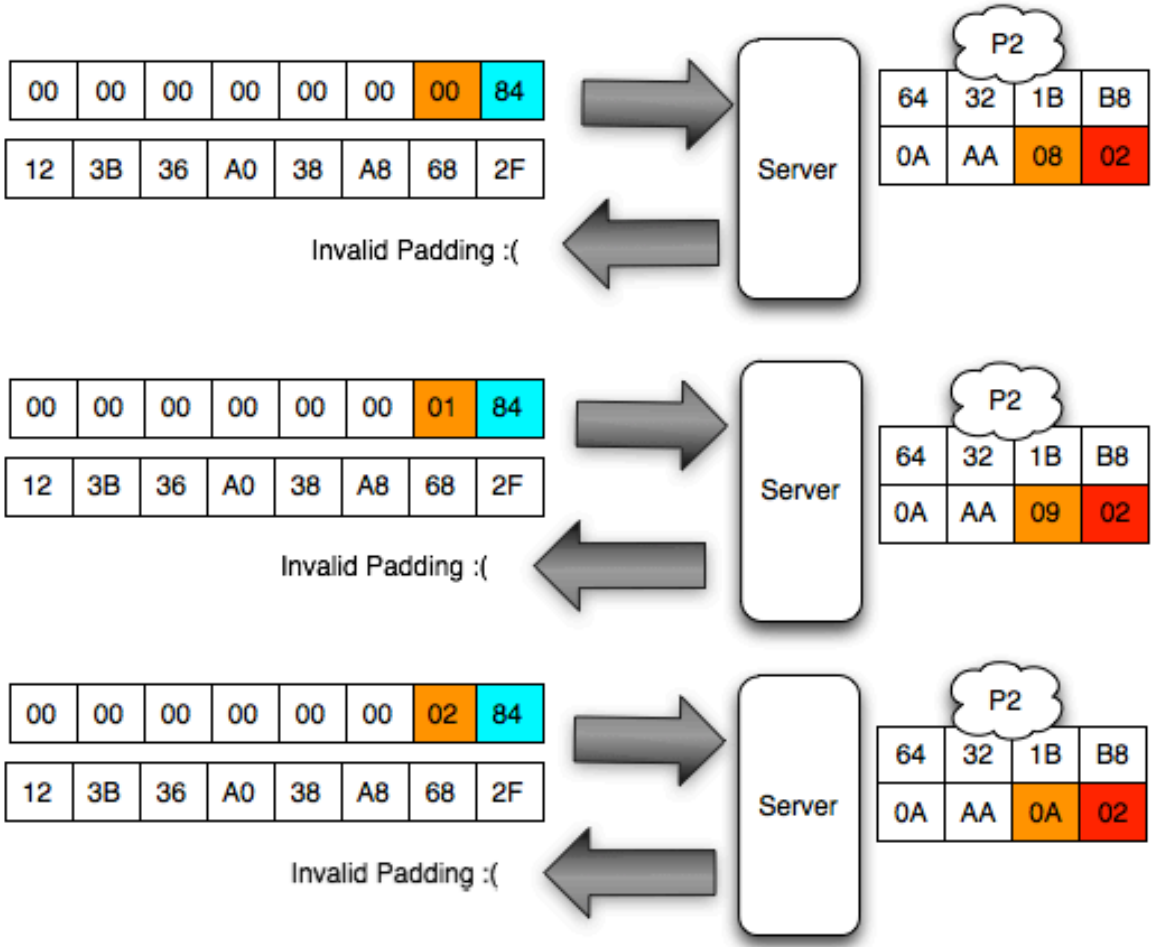
- A xor B = 01. Find A and B!
- Ask the Oracle:
  - A xor 0x85 = 01 ?
- Oracle answer:
  - Valid pad = Yes
- A must be 0x86

Decrypt 7th Byte

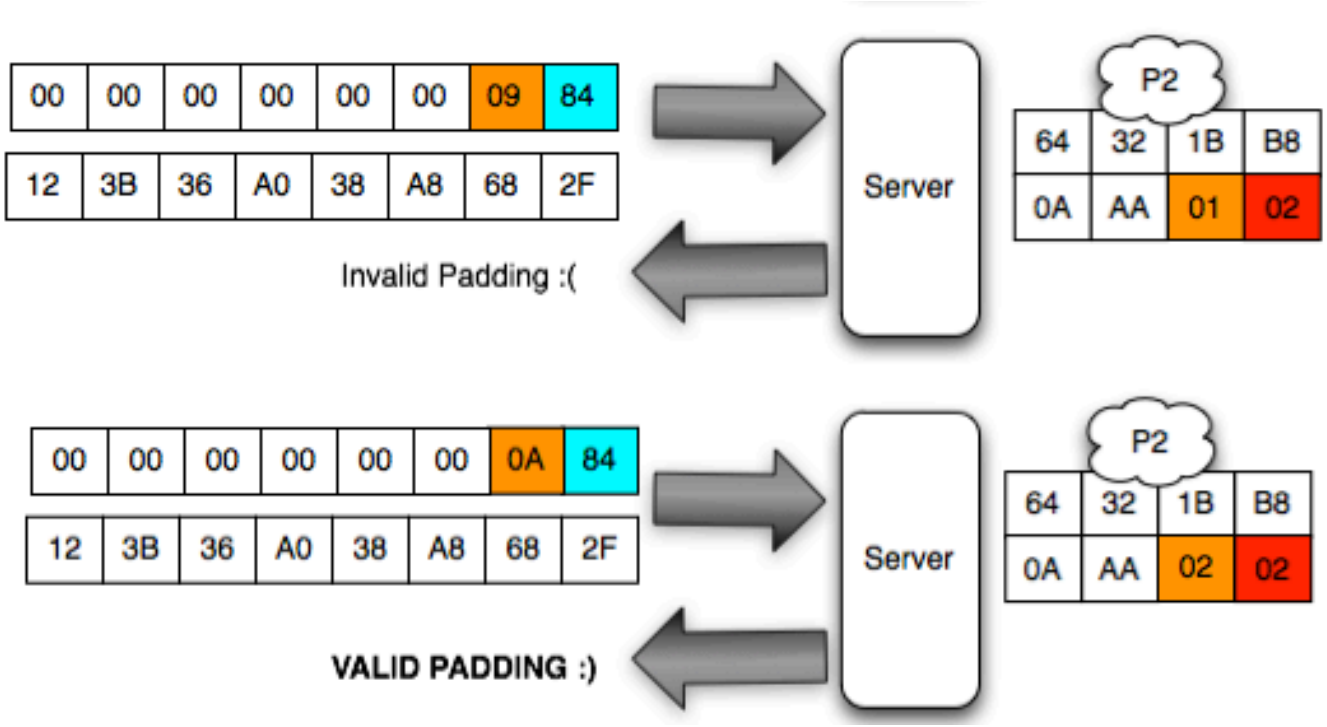
Decrypt 7th Byte



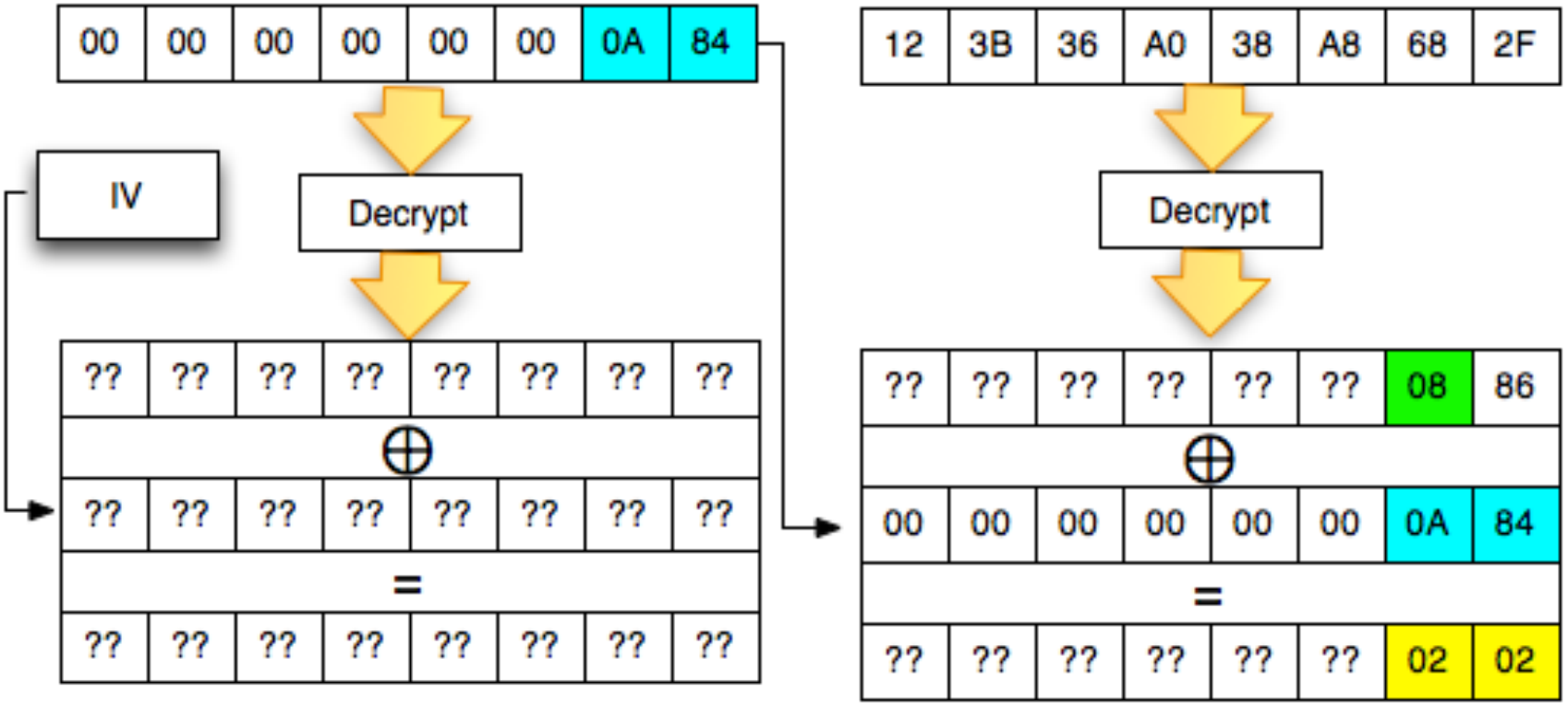
Look for Valid 2 Byte Pad



Valid 2 Byte Pad Found!

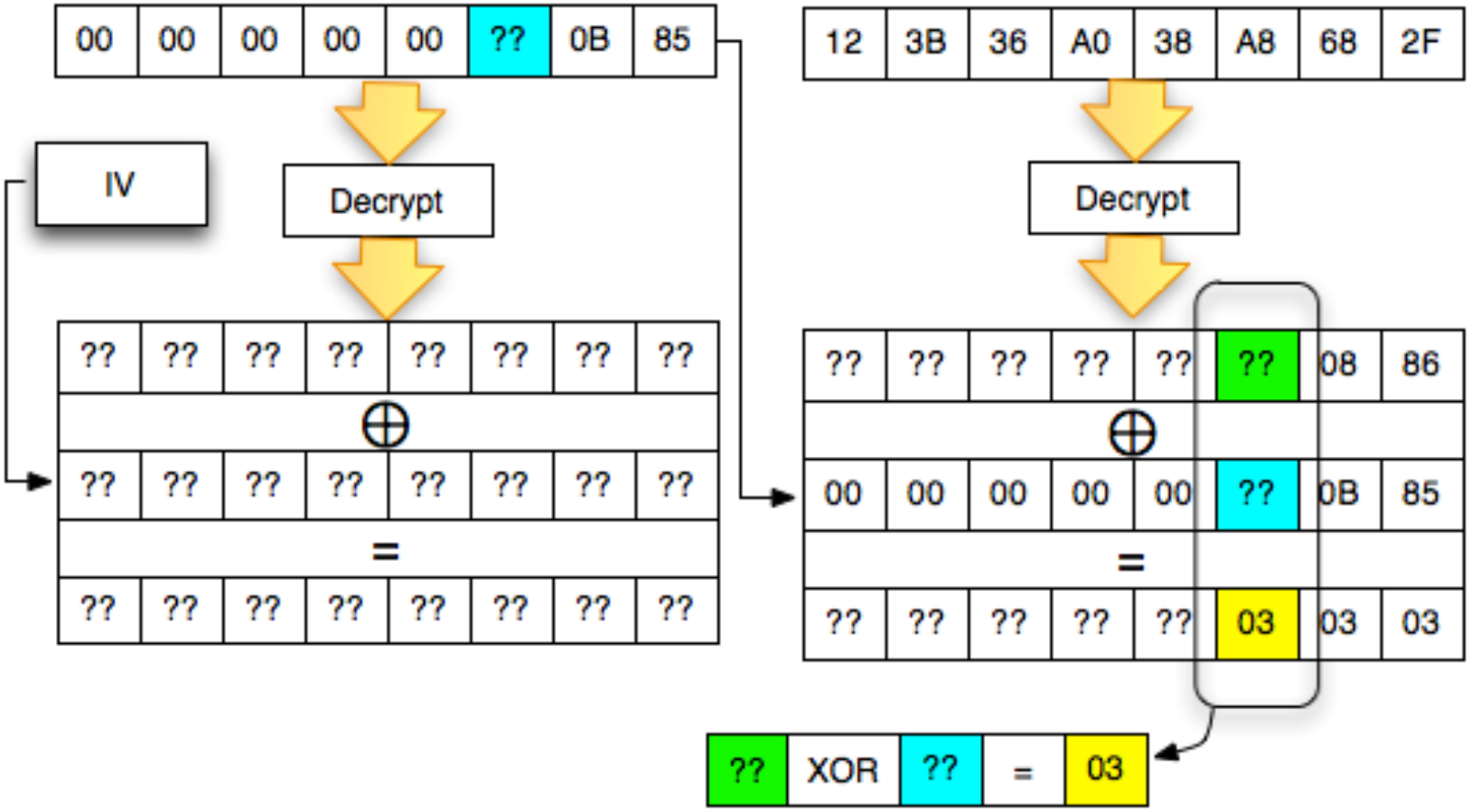


7th Byte Decrypted

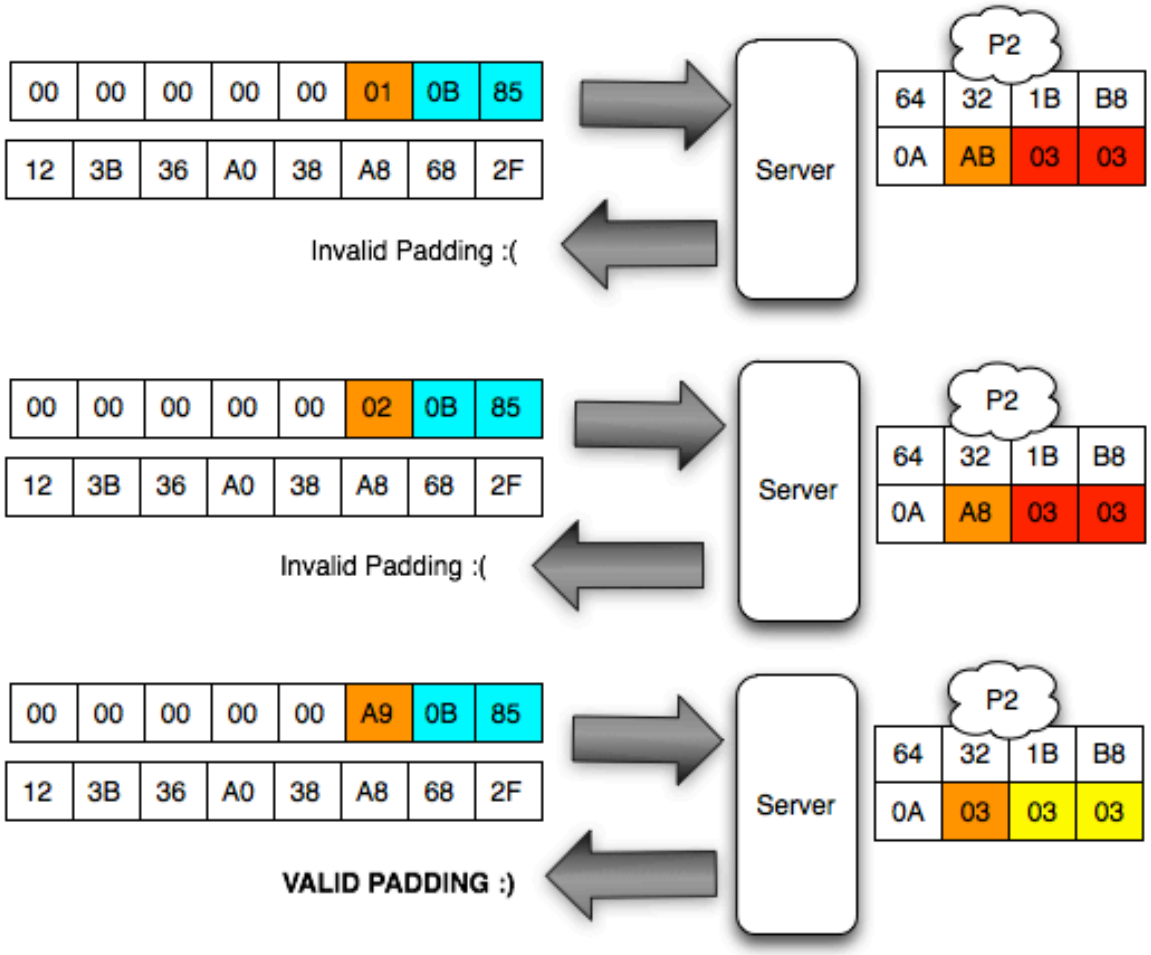


Decrypt 6th Byte

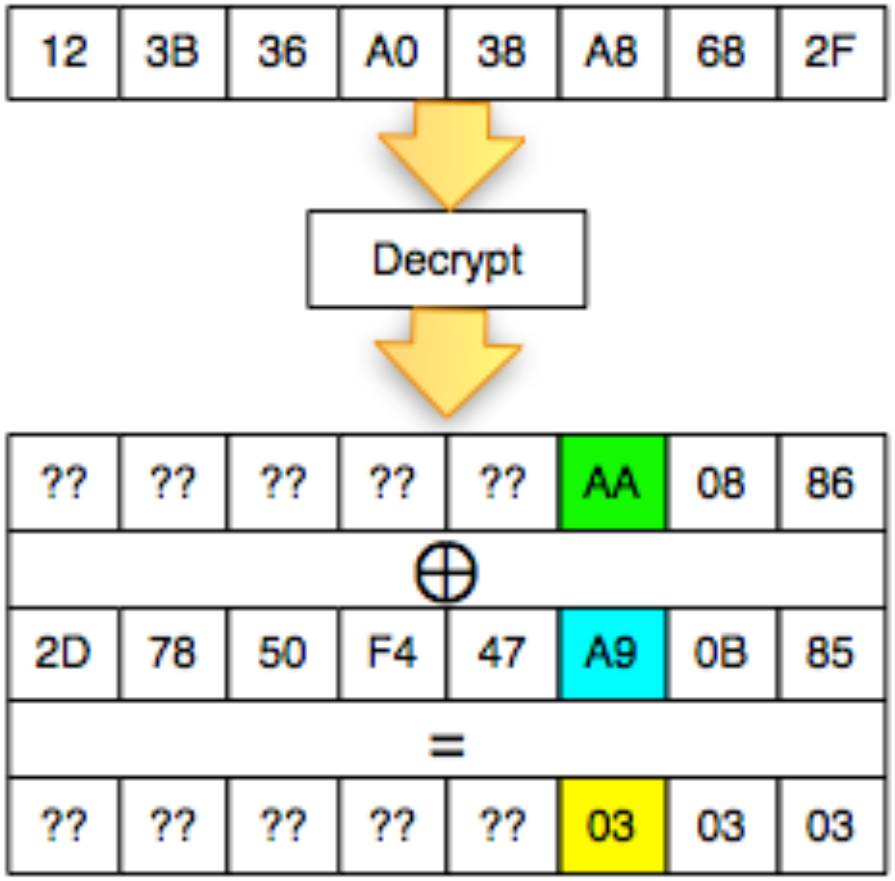
Decrypt 6th Byte



Valid 3 Byte Pad Found

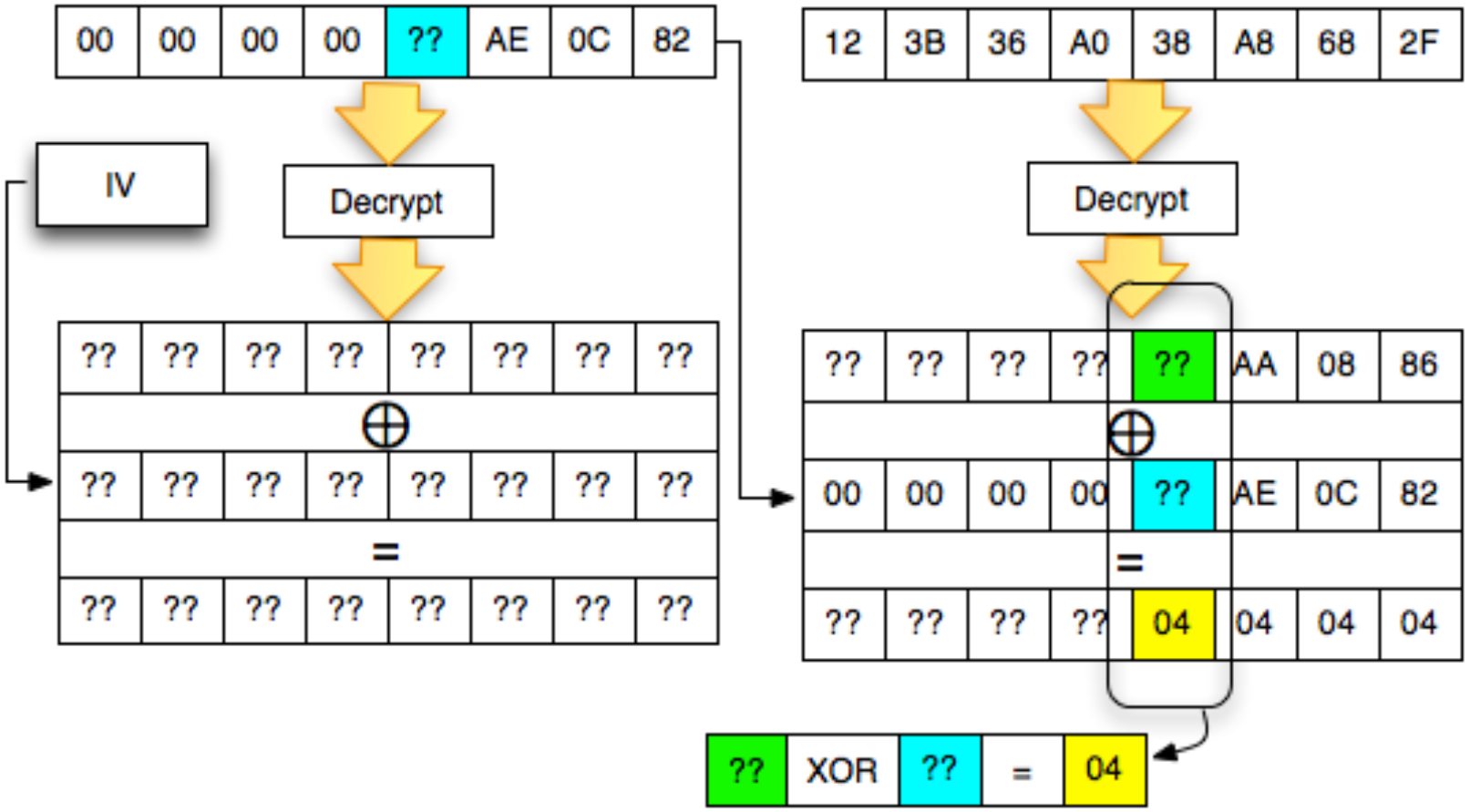


6th Byte Decrypted

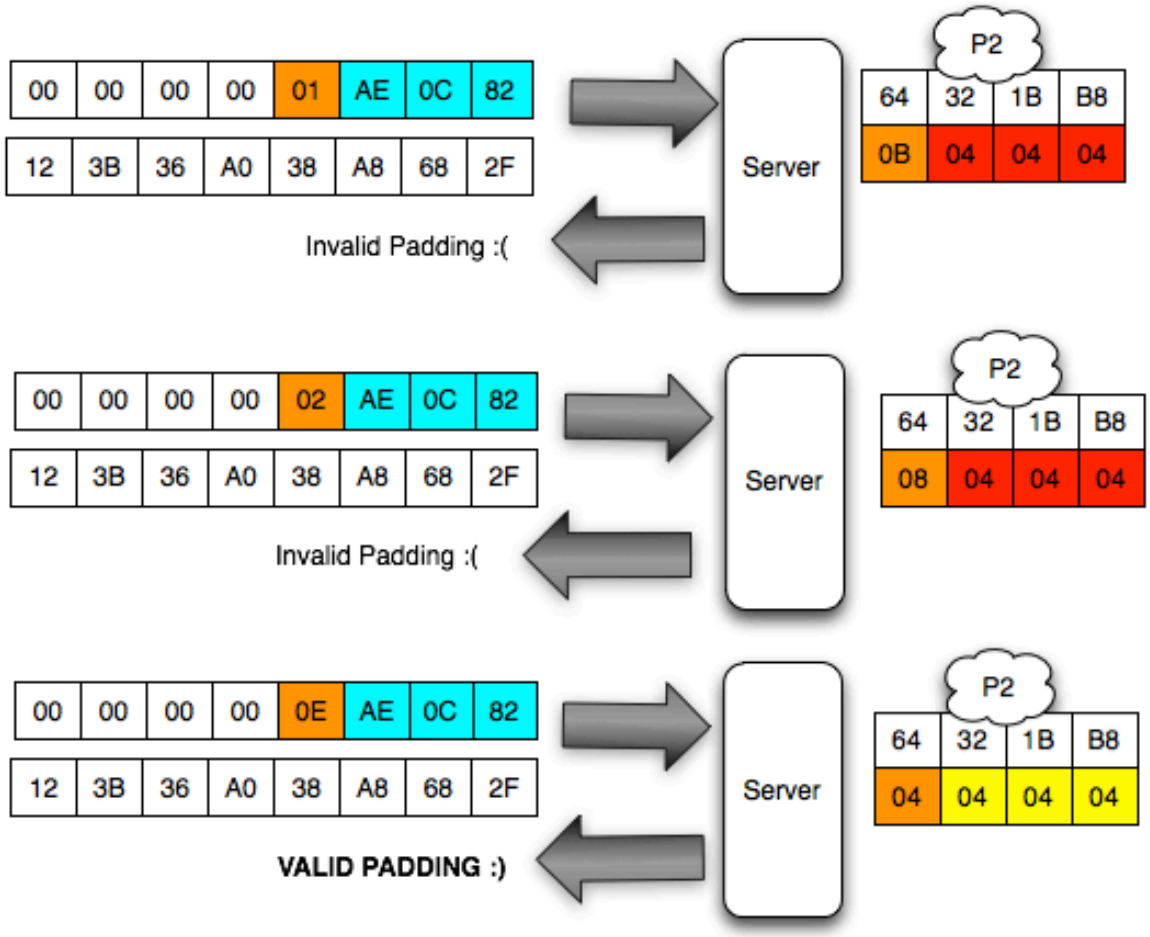


Decrypt 5th Byte

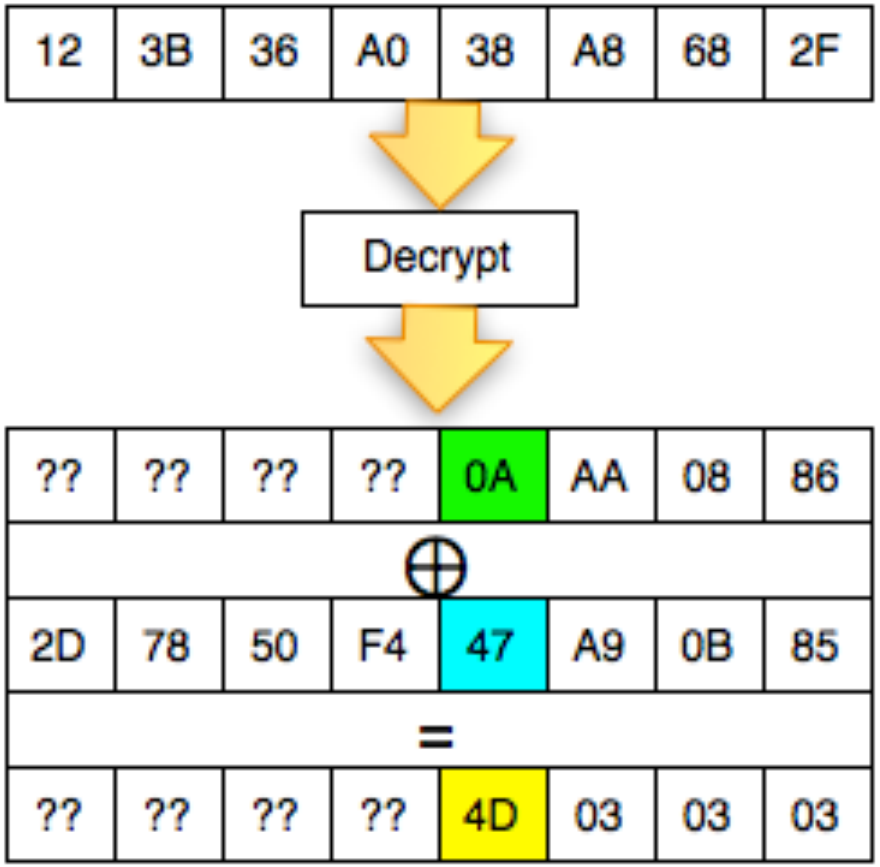
Decrypt 5th Byte



Valid 4 Byte Pad Found

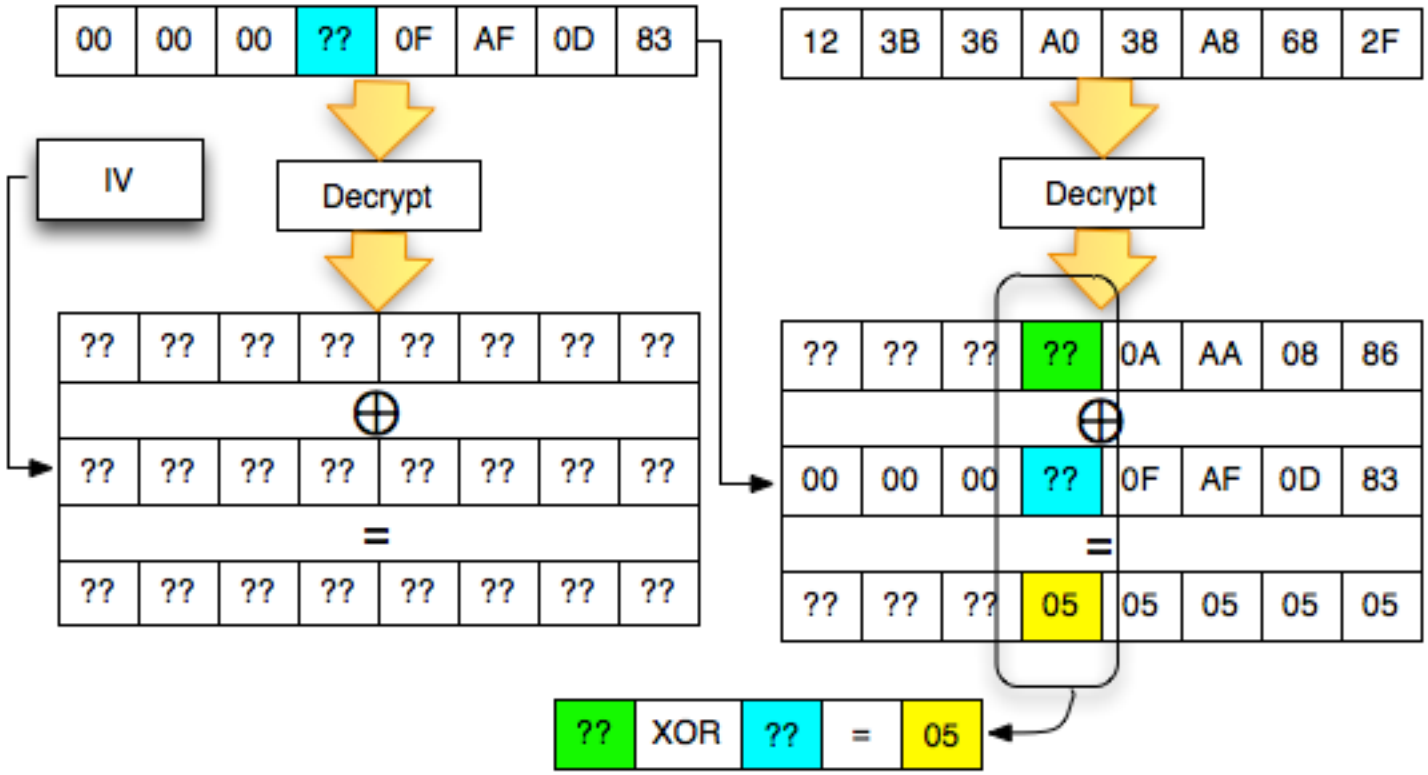


5th Byte Decrypted

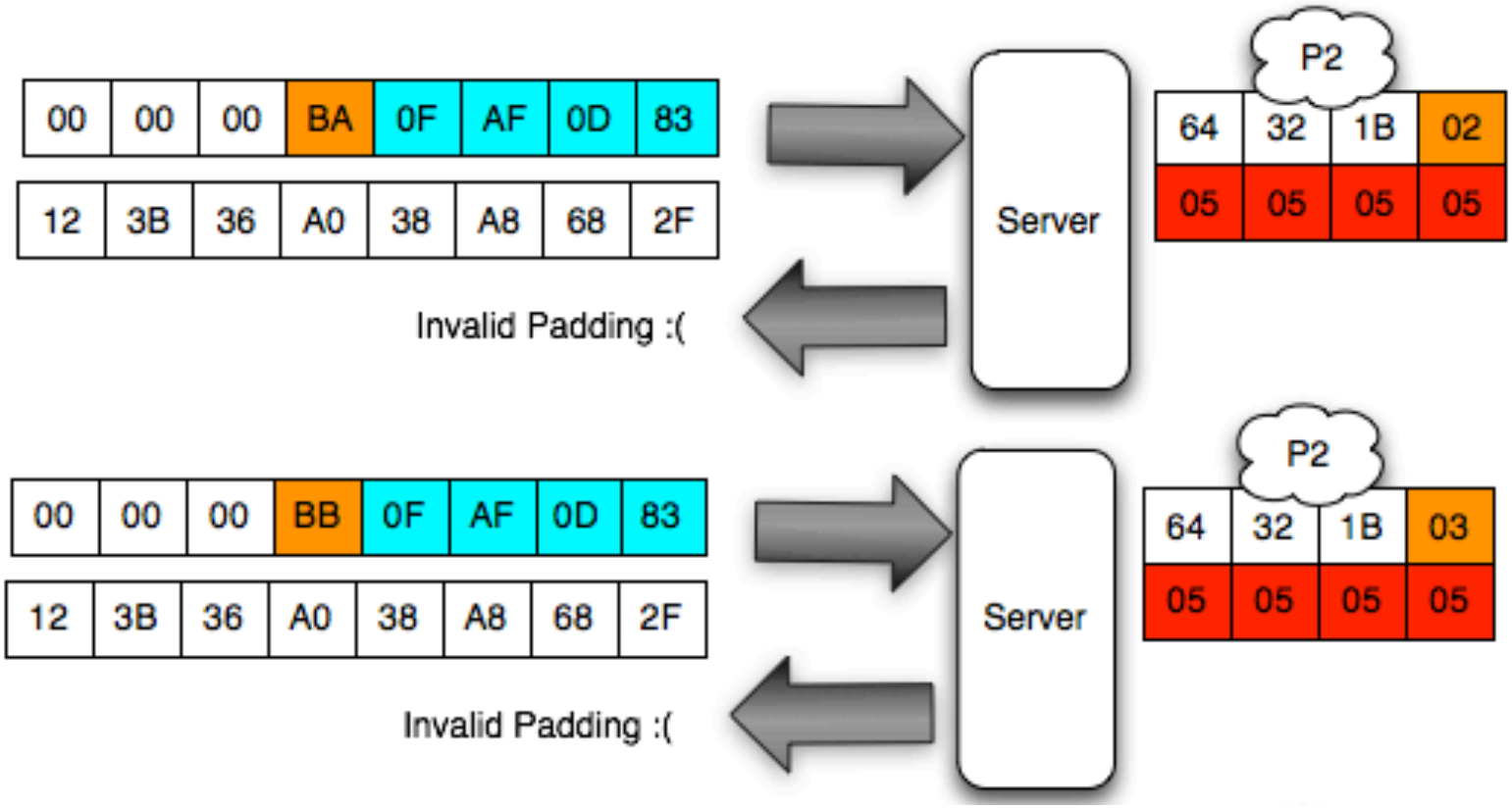


Decrypt 4th Byte

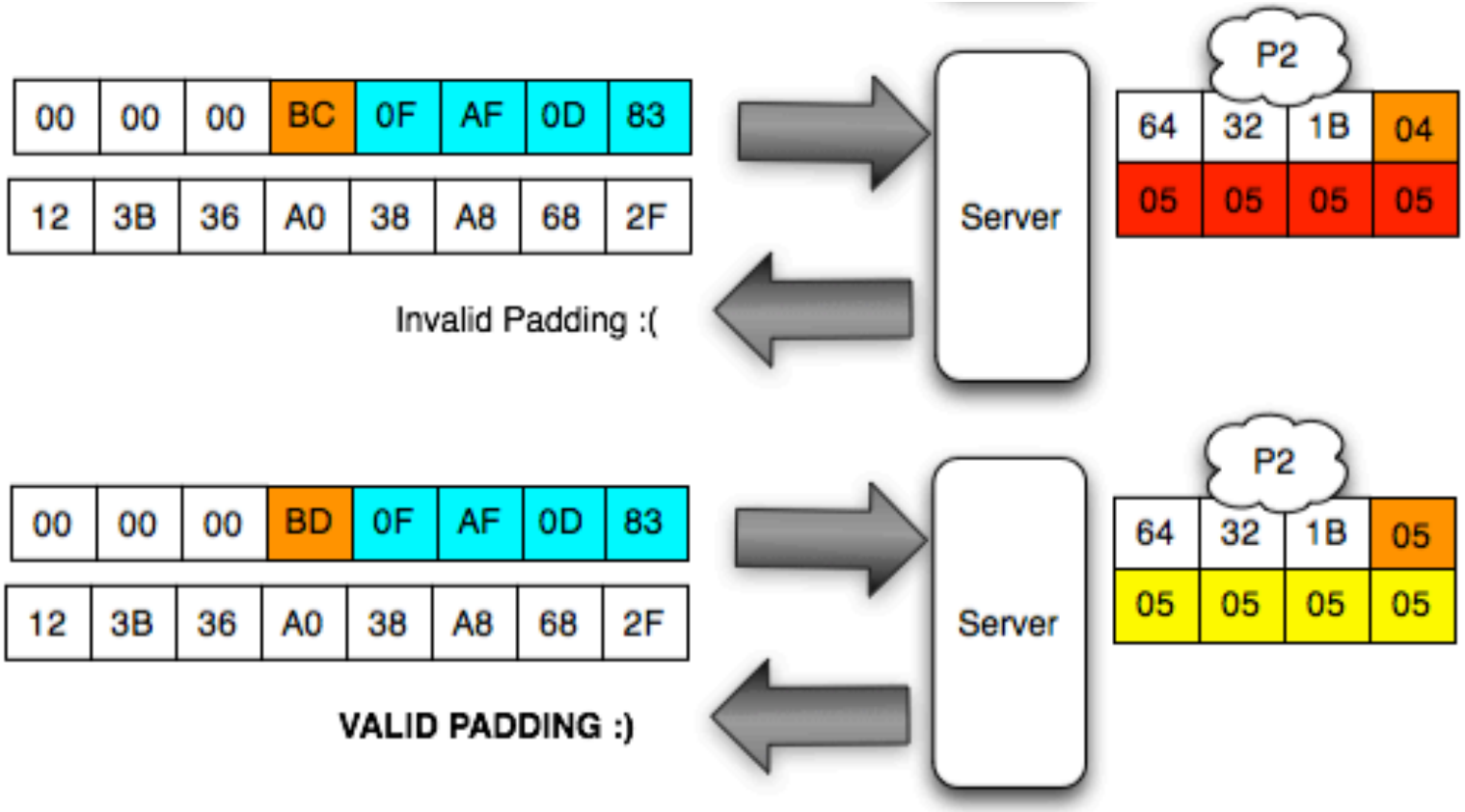
Decrypt 4th Byte



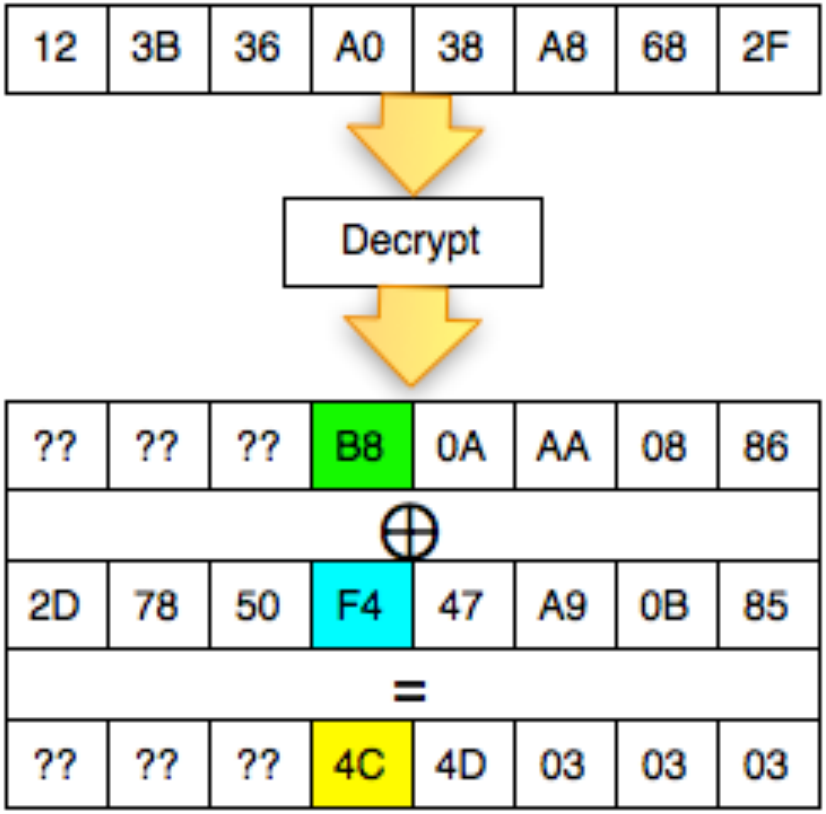
Look for Valid 5 Byte Pad



Valid 5 Byte Pad Found

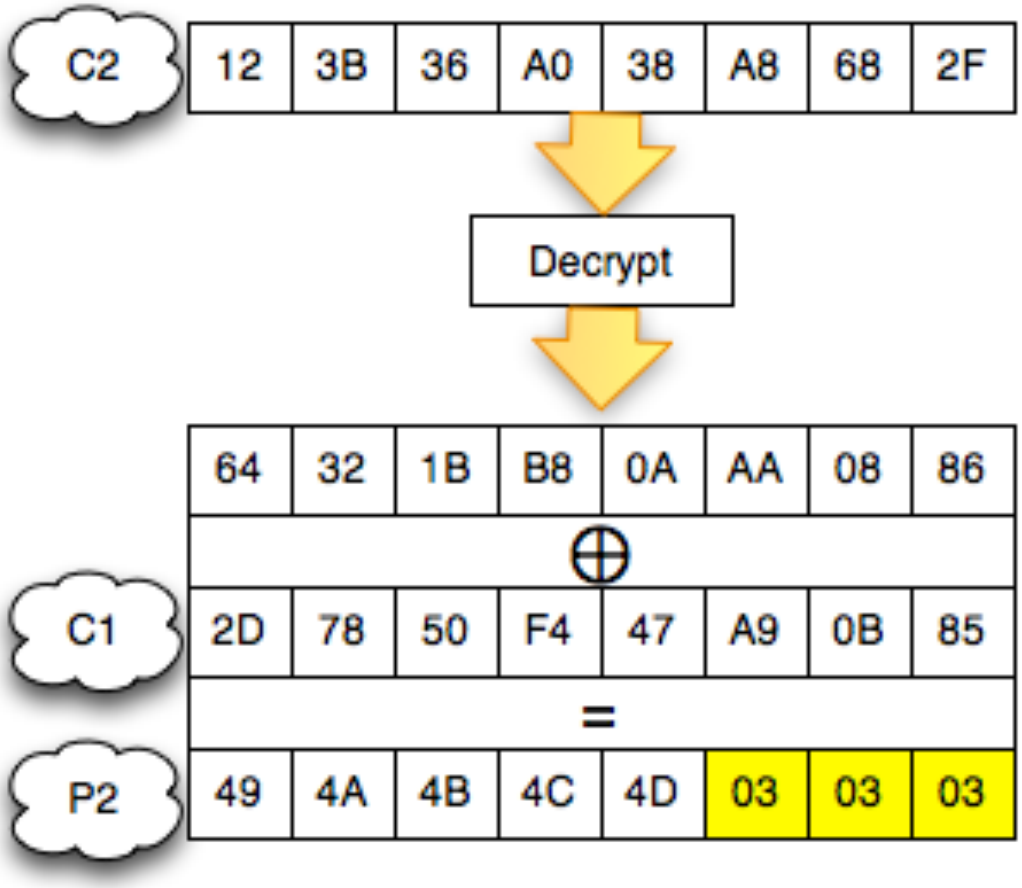


4th Byte Decrypted

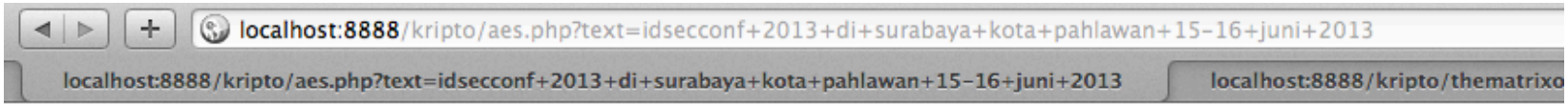


Full Block Decrypted

C_2 Block Decrypted



Case



IV

4A	32	B2	9B	21	32	17	B9	3A	F2	7A	FB	3B	BD	72	7E

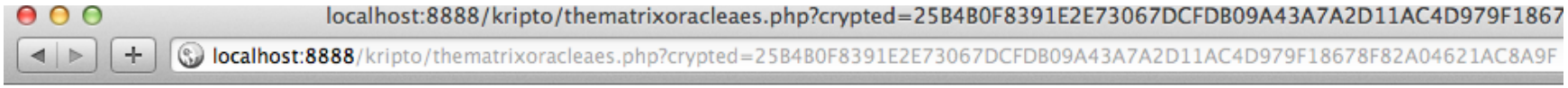
Plaintext dan Ciphertext

69 **i**	64 **d**	73 **s**	65 **e**	63 **c**	63 **c**	6F **o**	6E **n**	66 **f**	20	32 **2**	30 **0**	31 **1**	33 **3**	20	64 **d**	69 **i**	20	73 **s**	75 **u**	72 **r**	61 **a**	62 **b**	61 **a**	79 **y**	61 **a**	20	6B **k**	6F **o**	74 **t**	61 **a**	20	7 **p**
CA	26	07	76	F0	9F	52	5A	19	3C	EE	4F	48	A0	06	23	47	1C	33	5F	00	18	FD	CD	D2	75	64	DF	94	65	12	74	2

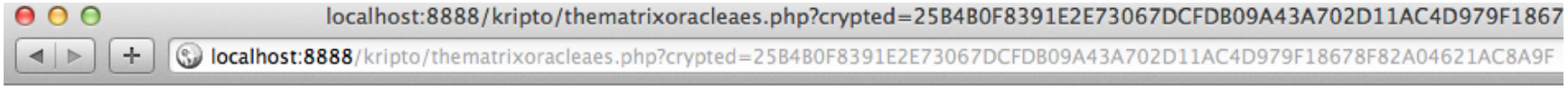
IV + Ciphertext

4A32B29B213217B93AF27AFB3BBD727ECA260776F09F525A193CEE4F48A00623471C335F0018FDCDD27564DF94651

The Oracle



PADDING OK



INVALID

Decryptor

```
Pucuk:idsecconf rizki$ ./padfinder-aes.py 4A32B29B213217B93AF27AFB3BBD727ECA2607
76F09F525A193CEE4F48A00623471C335F0018FDCDD27564DF9465127425B4B0F8391E2E73067DCF
DB09A43A7A2D11AC4D979F18678F82A04621AC8A9F

64203331303220666e6f636365736469   Plaintext Blok ke-1:idsecconf 2013 d
2061746f6b2061796162617275732069   Plaintext Blok ke-2:i surabaya kota
6a2036312d3531206e6177616c686170   Plaintext Blok ke-3:pahlawan 15-16 j
0808080808080808333131303220696e75   Plaintext Blok ke-4:uni 2013

Plaintext: 'idsecconf 2013 di surabaya kota pahlawan 15-16 juni 2013'
Pucuk:idsecconf rizki$
```

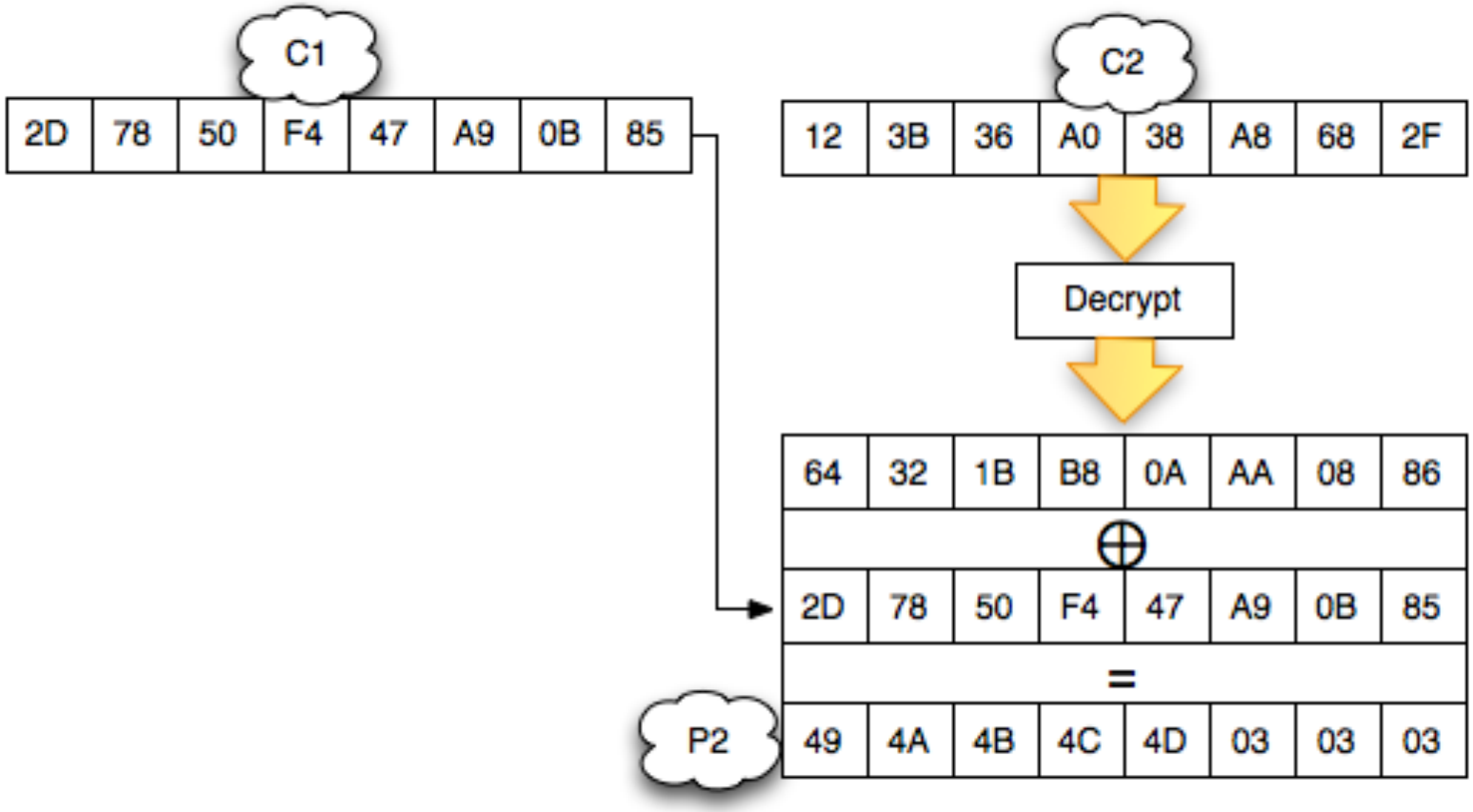
Decryption Demo

Encrypt Fake Message

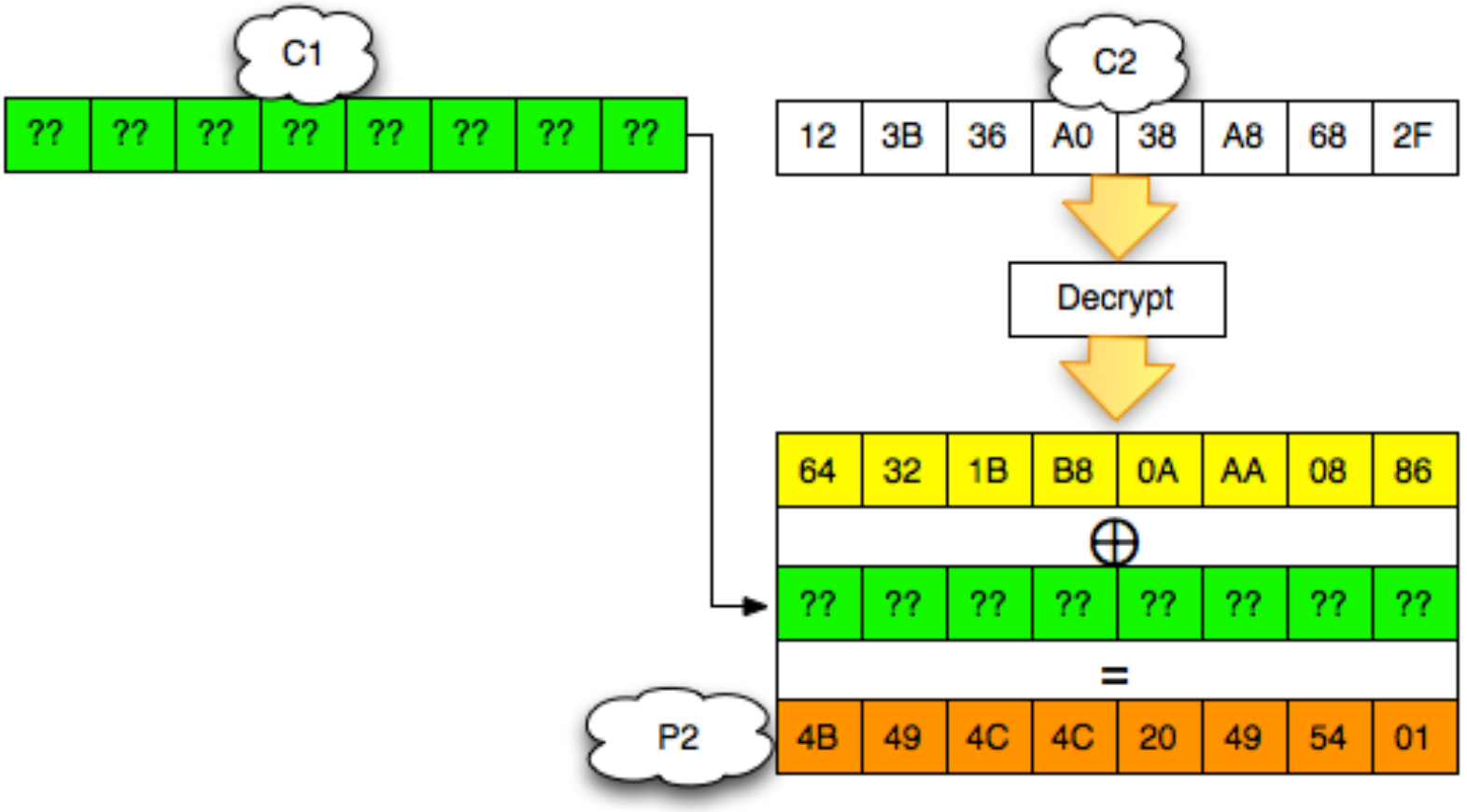
Encrypt without Knowing the Key

- You can make cipher text say whatever you want when decrypted
- Property of CBC mode

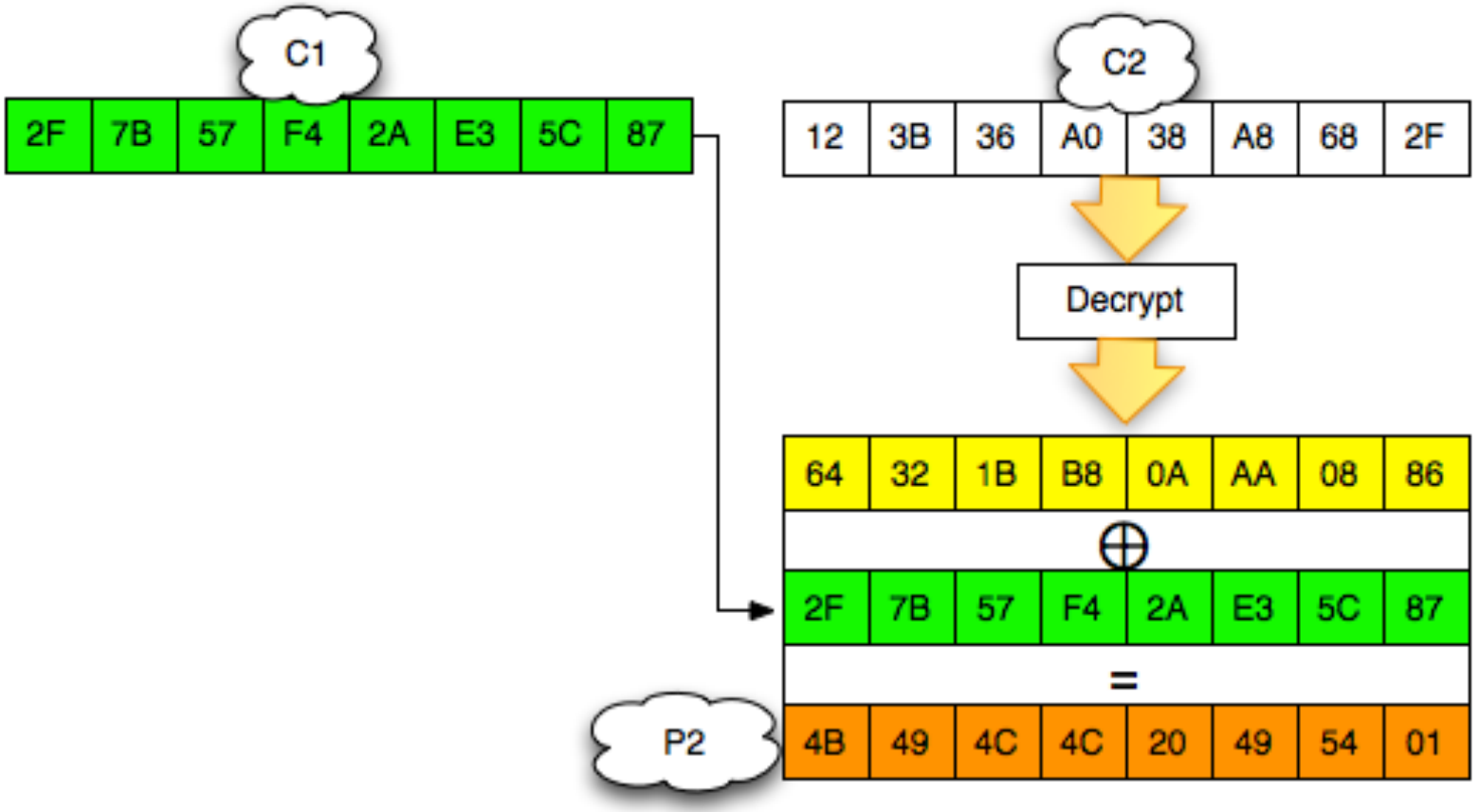
P_2 depends on C_1



“KILL IT”


C1
?? ?? ?? ?? ?? ?? ?? ??
C2
12 3B 36 A0 38 A8 68 2F
Decrypt
64 32 1B B8 0A AA 08 86
⊕
?? ?? ?? ?? ?? ?? ?? ??
=
P2
4B 49 4C 4C 20 49 54 01

“KILL IT”


C1
2F 7B 57 F4 2A E3 5C 87
C2
12 3B 36 A0 38 A8 68 2F
Decrypt
64 32 1B B8 0A AA 08 86
⊕
2F 7B 57 F4 2A E3 5C 87
=
P2
4B 49 4C 4C 20 49 54 01

Encryption Procedure

- Encrypt: “BESOK PAGI SERANGAN UMUM IWO JIMA”
- Split plaintext into blocks:
  - P_1 = ‘BESOK PA’
  - P_2 = ‘GI SERAN’
  - P_3 = ‘GAN UMUM’
  - P_4 = ‘ IWO JIM’
  - P_5 = ‘A’+07+07+07+07+07+07+07

Encryption Procedure

- Choose C_5 all-zeros
- Use padding oracle attack to find Decrypt(C_i)
- C_4 = Decrypt(C_5) XOR P_5
- C_3 = Decrypt(C_4) XOR P_4
- C_2 = Decrypt(C_3) XOR P_3
- C_1 = Decrypt(C_2) XOR P_2
- IV = Decrypt(C_1) XOR P_1

Encryption Demo

Authenticated Encryption

Authenticate before Decrypt

- Why we need to authenticate/verify encrypted message before decrypting it ? It's already encrypted with shared secret key, after all.
- Imagine that only Alice and Bob know the key. If Bob could decrypt a cipher text with the secret key and get a clean and understandable plain text, then Bob know it only could be encrypted by Alice
- Many people have thought that, but they were wrong
- Without message authentication, active attacker could use padding oracle attack to decrypt and also encrypt without knowing the key

Encryption and MAC

- Encryption provides confidentiality, it doesn't provide integrity and authenticity
- Don't use encryption without message authentication
- Encrypt your message then calculate MAC
- Never decrypt message without checking MAC
- Decrypt only when ciphertext is MAC-authenticated