pre_pt ⊕ block → ct_block

↓ AES-ECB key↓

ct_block

c
:
c

ct_block = ct_block ⊕ pre_ct

ct += ct_block

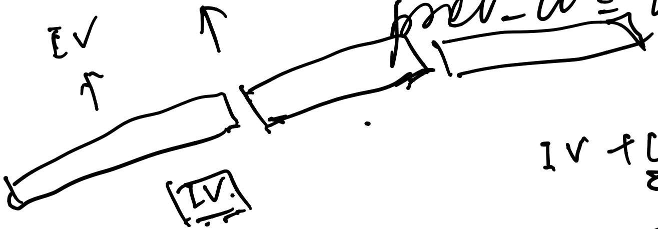
pre_pt = block

prev_ct = ctblock

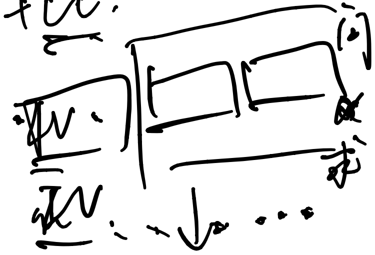
a → 输入.

b. dec_tsb

a==b.



IV + ct.



问询

求控制 / a

得 b.

再控制 b.

求 a.

g g

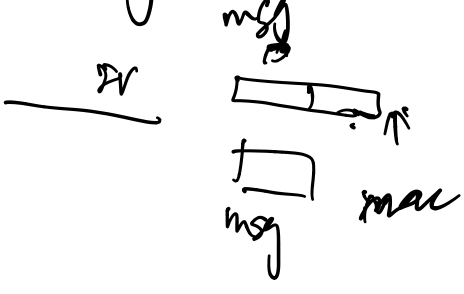
IV.

Iv

~~give me_flag~~

不变 mac

msg



msg forge

$b_1 \oplus m_1$
$b_2 \oplus m_2$
$b_3 \oplus m_3$ ⇒ ptblock_

enc_msg

msg → dec

g.

IV ⊕

a mesage.

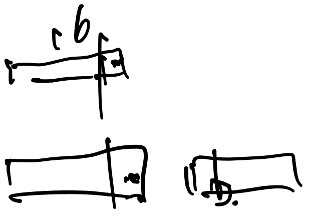
U IV ⊕ mes g

↓

AES ⊕

↓

⊕ IV.



prive_pt = pt_blok.

prive_ct = blok

give_

16

use givene_flag.

givene_flag.

pre_bt bock =

jg

pre. =

86

........

IV + enctrymessage.

a IV.

mess

prev_pt = blank

givenm_flag

prev_ct = ct_block

m [0:16]

ppt . m[0:16] → m[16:32] → IV

ppt = IV
pct

pct = c[0:16]

ppt[16:32]

pct [16:32

= ~~ppt[16:32]~~

ppt [16:32 32:48

pct [32:48]

enc

IV

dec.

ppt =

pct = IV.

c[0:16]

[0 ]

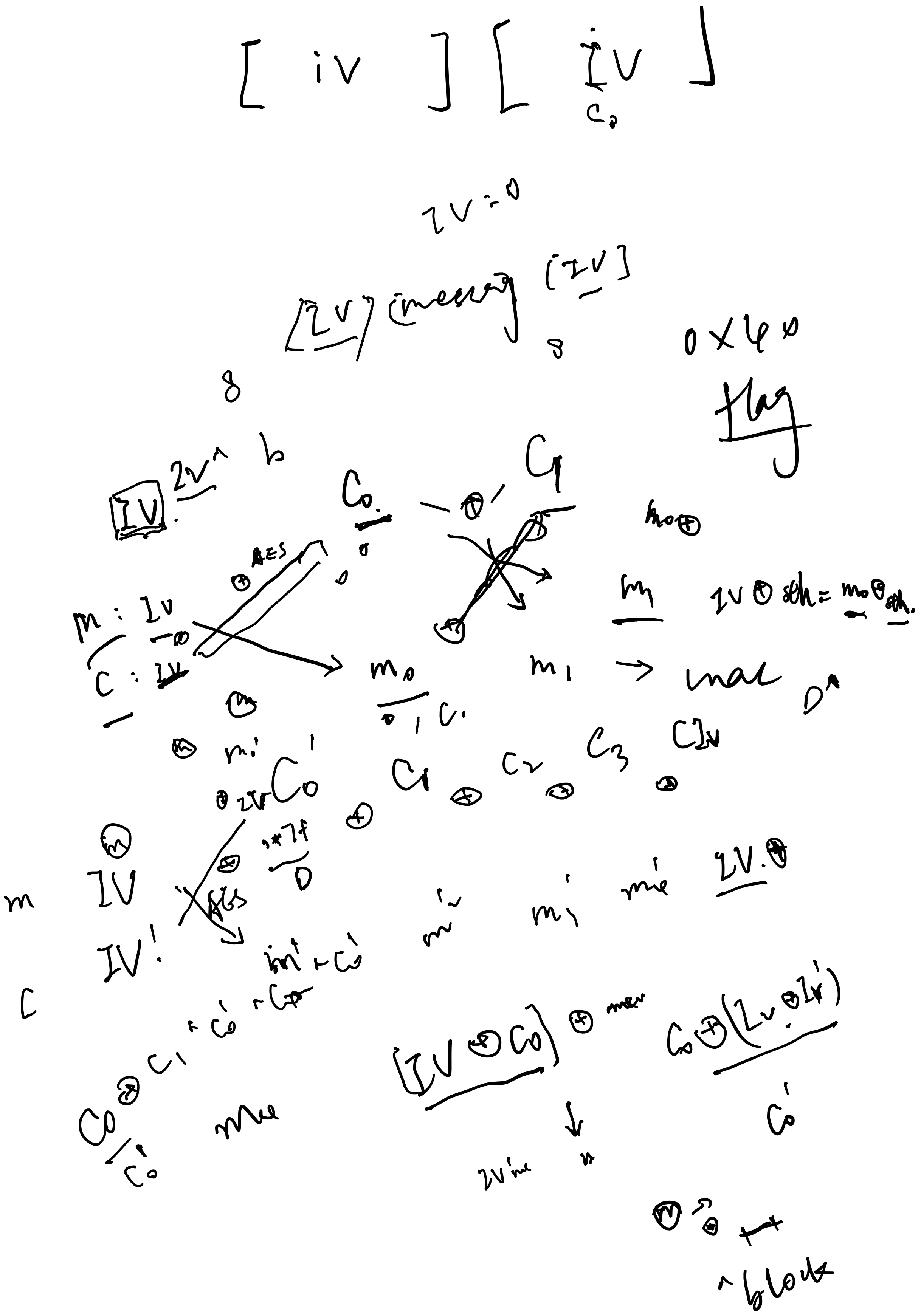
c[16:32]

m[0:16]

pct [0:16]

ppt [16:32] → IV

[ ] pct [16:32]


[ iv ] [ IV ]
C₀
IV = 0
[IV] (message) [IV]
8
0x40
8
flag
IV 2v^ b
C₀ ⊕ C₁
m₀⊕
⊕ AES
m : IV
C : IV
m₁
IV ⊕ sth = m₀⊕ sth.
m₀
m₁ → mac
⊕, C₁
D^
m₀'
C₀' C₁ ⊕ C₂ ⊕ C₃ ⊕ CIV
⊕ 2IV
0x7f
D
⊕
m IV
AES
m' m₁ m₂' m₂' IV.⊕
C IV!
m' + C₀' + C₀ ⊕
C₀ ⊕ C₁ + C₀' + C₀
C₀ ⊕ C₁
[IV ⊕ C₀] ⊕ mev
C₀ ⊕ (IV ⊕ IV')
C₀' me
C₀
IV'me
C₀'
⊕ ⊕
^ block

m ^ byte

→ IV

m

IV    IV    p    p    p

IV.    IV    flag    ^ ^ D ^ byte

Dr

p  IV    m_1    m_2    p    m_i

AES(k⊕C) ⊕ p    (C_1 ⊕ C_2)    ^

c  IV    c_1    c_2    ↓    D^ D^ byte

byte

b ^ 48^    b    b ~ b    63 b

15

63

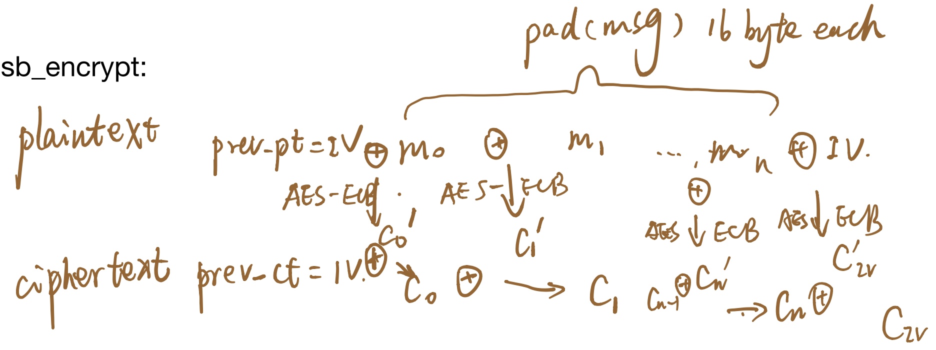
48^ 48^ (63 − len p)

47    47

15 ^ (63 − 15) m

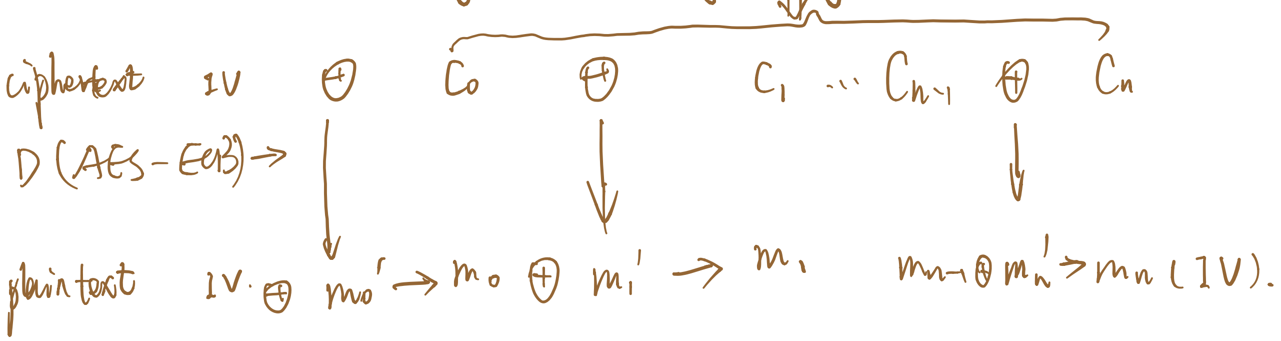
15^ (63 − len p) x m.

tsb_encrypt:



return iv + $C_0$ + ⋯ + $C_n$ + $C_{IV}$.

tsb_decrypt: iv = msg[:16] msg = msg[16:].


ciphertext IV ⊕ C₀ ⊕ C₁ ⋯ C_{n-1} ⊕ C_n
D(AES-ECB) →
plaintext IV ⊕ m₀' → m₀ ⊕ m₁' → m₁
m_{n-1} ⊕ m_n' → m_n (IV).

then check $IV == m_n$ $pt = m_0 + \dots + m_{n-1}$

return unpad(pt). here is the problem with the crypto system.

we can change the last byte of $IV, C_0, \dots, C_n$, to keep $IV == m_n$, the same time, the last byte of pt changes. so len( unpad(pt)) = 1, and makes a == b much easier by brutefore first byte of IV ( remember to change every block first byte to make IV =