Block Cipher Modes, Var-length ptxts

Definitions changed: var-length ptxts

Ch 8 def:

$m_L, m_R \in \Sigma.M$

Ch 9 def:

$m_L, m_R \in \Sigma.\mathcal{M}$

Ex: $Enc(k,m) = (r, F(k,r) \oplus m)$

plaintexts $\mathcal{M} = \{0,1\}^{out}$

ALL ptxts have same length!

⇒ length is public

⇒ nothing to hide

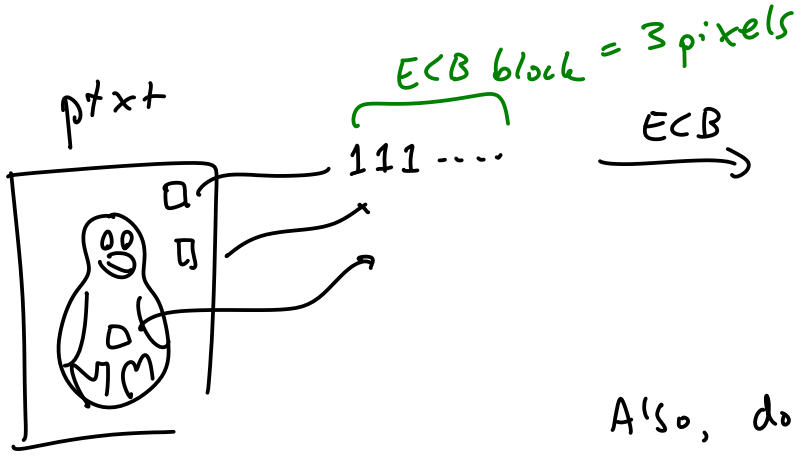
Ex: CBC mode,

$\mathcal{M} = (\{0,1\}^{\lambda})^*$

= all strings whose length is mult. of λ

Library must enforce $|m_L| = |m_R|$ explicitly

ECB sucks



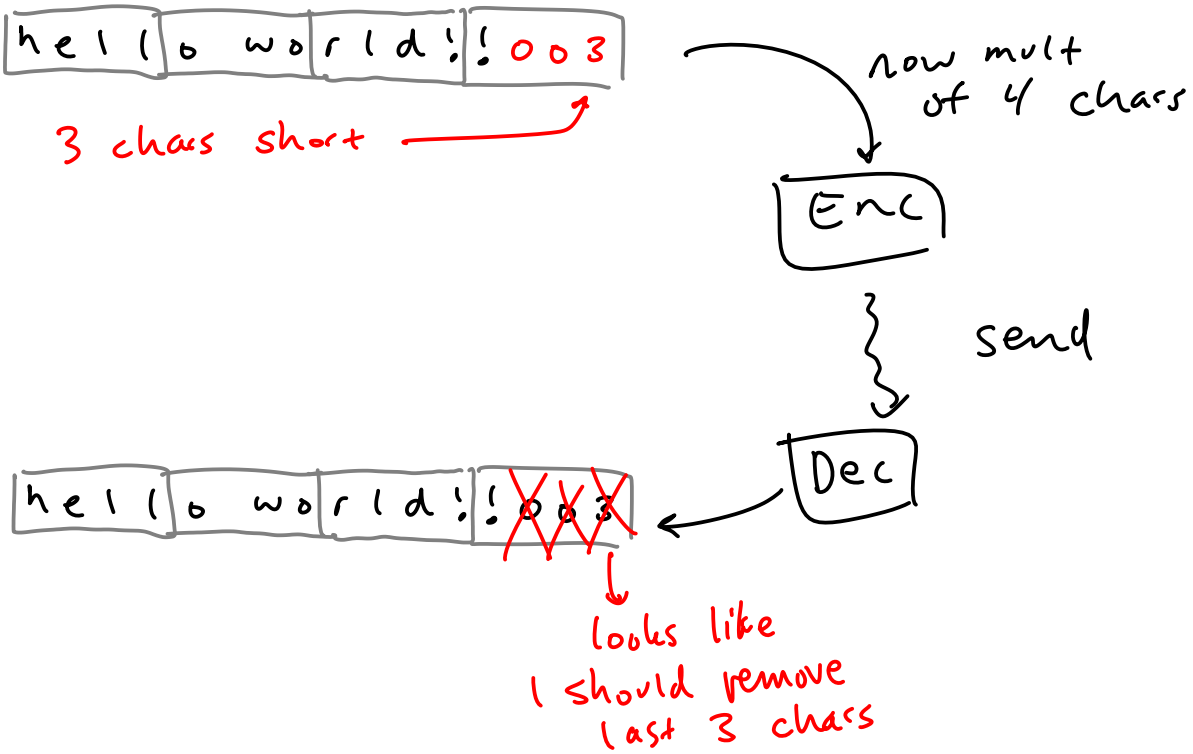
Also, don't encrypt passwords

padding/stealing examples

padding: have Enc scheme, only accepts input multiple of blocklength

Ex: block length = 4 chars

want to encrypt: (13 chars)

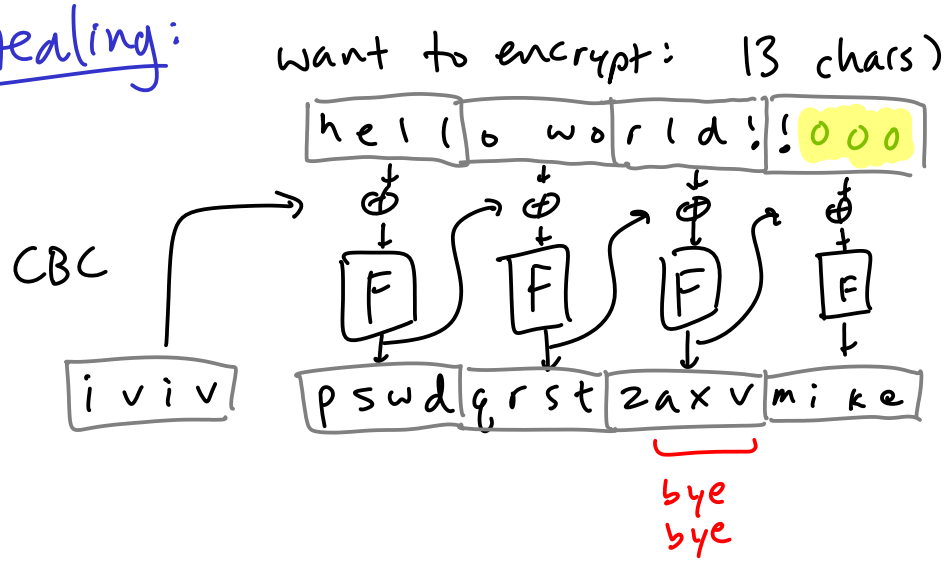


Ex:

want to encrypt hell o!!3 0004

(If length = mult of block length ⇒ add extra block)

Stealing:

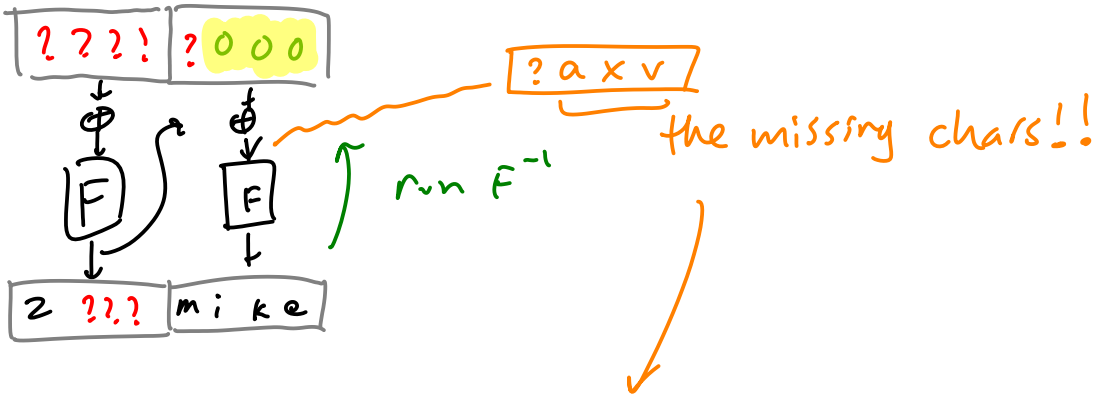


3 chars short, so I should find 3 chars of ctxt to delete

final ctxt: ivIvpswdqrstzmike

4 13

looks 3 chars short, so try to find 3 missing ctxt chars



ivIvpswdqrstzaxvmike

4

↓ decrypt w/ regular CBC