Block Cipher Modes

Recap: CPA security (chosen plaintext attack)

"bare" block cipher (PRP) is <u>not</u> CPA-secure

better: $m \longrightarrow (r, F(k,r) \oplus m)$

λ bits $\longrightarrow$ 2λ bits

Ex: AES $\lambda = 128$ (16 bytes)

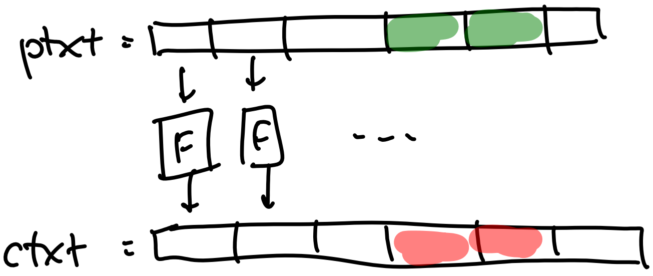
What about longer ptxts?

split into blocks of length λ, encrypt 1-by-1

If ptxt is N bits
ctxt is $2N$ bits

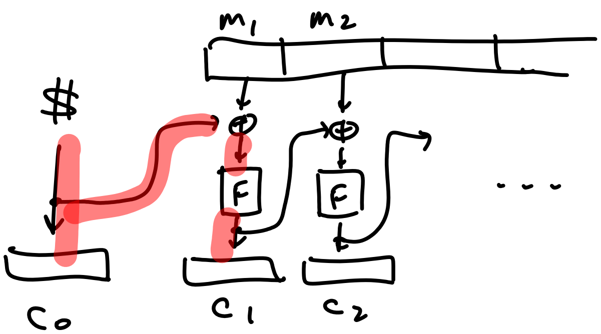
Block cipher mode = method of using block cipher on bigger data

ECB mode = electronic codebook



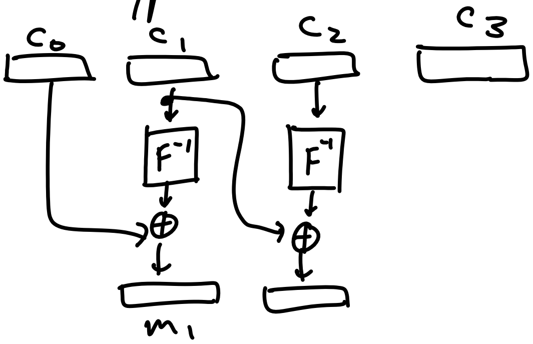
Same ptxt block
↓
same ctxt block

CBC mode = cipher-block chaining

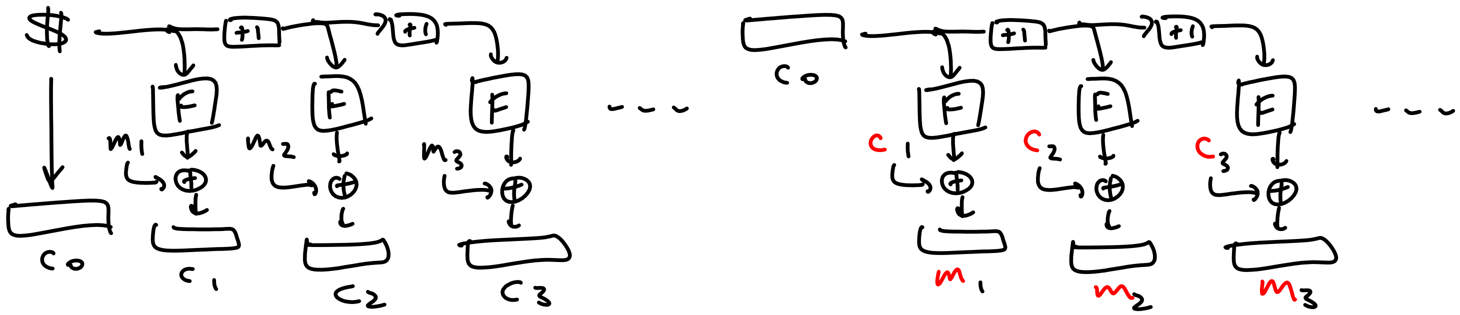


Initialization Vector (IV)

Decryption



CTR mode = counter



Security

CBC, CTR are CPA-secure for encryption*...

* except, no they aren't

length of ctxt depends on (leaks) length of ptxt

```
k ← KeyGen
CHALLENGE(m_L, m_R):
    c = Enc(k, m_L)
    ret c
```

$\approx$

```
k ← KeyGen
CHALLENGE(m_L, m_R):
    c = Enc(k, m_R)
    ret c
```

Attack by passing m_L long
m_R short

fixed definition ☆

```
k ← KeyGen
CHALLENGE(m_L, m_R):
    if |m_L| ≠ |m_R|: ret null
    c = Enc(k, m_L)
    ret c
```

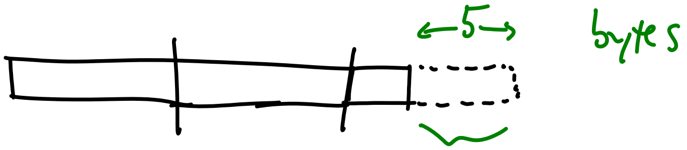
$\approx$

```
k ← KeyGen
CHALLENGE(m_L, m_R):
    if |m_L| ≠ |m_R|: ret null
    c = Enc(k, m_R)
    ret c
```

What if ptxt length not exact multiple of λ ?

padding: extend data to next multiple of λ
(must be reversible !)

ex:



pad w/ 0x 00 00 00 00 05

clever truncation (ciphertext stealing)