Using block ciphers

Modes of operation: many time key (CBC)

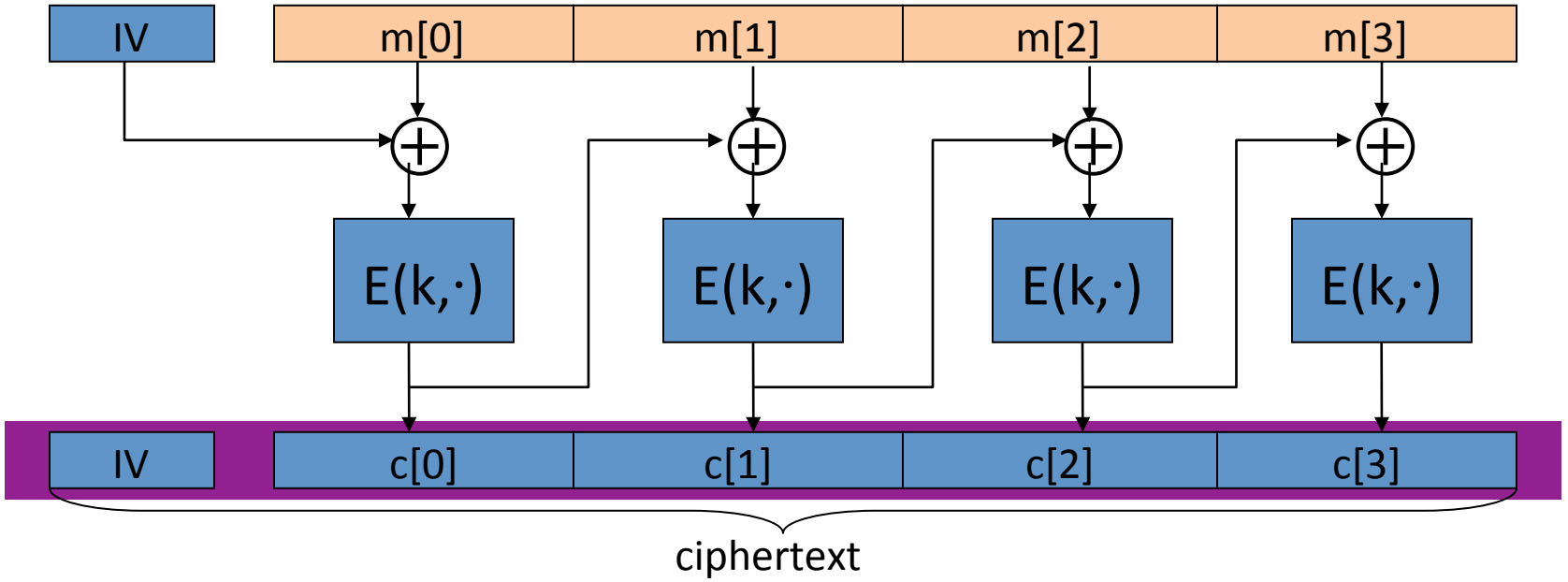
Example applications:

1. File systems: Same AES key used to encrypt many files.
2. IPsec: Same AES key used to encrypt many packets.

Construction 1: CBC with random IV

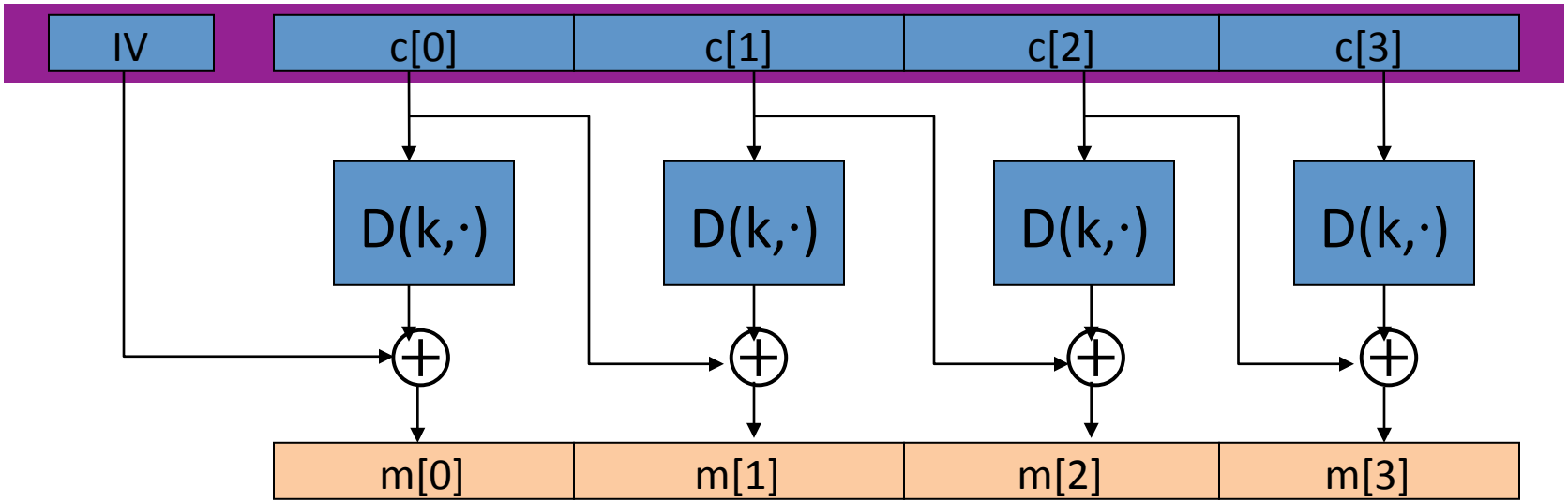
Let (E,D) be a PRP. $E_{CBC}(k,m)$: choose **random** IV $\in X$ and do:

$E : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ $\quad$ $IV \in \{0,1\}^n$



Decryption circuit

In symbols: $c[0] = E(k, IV \oplus m[0])$ $\Rightarrow$ $m[0] =$ [redacted]



CBC: CPA Analysis

CBC Theorem: For any L>0,

If E is a secure PRP over (K,X) then

E_{CBC} is a sem. sec. under CPA over (K, X^L, X^{L+1}).

In particular, for a q-query adversary A attacking E_{CBC} there exists a PRP adversary B s.t.:

$$Adv_{CPA}[A, E_{CBC}] \le 2 \cdot Adv_{PRP}[B, E] + 2\, q^2 L^2 / |X|$$

Note: CBC is only secure as long as $q^2L^2 << |X|$

An example

$$\mathrm{Adv}_{\mathrm{CPA}}[A, E_{\mathrm{CBC}}] \le 2 \cdot \mathrm{PRP\ Adv}[B, E] + \mathbf{2\, q^2\, L^2 / |X|}$$

q = # messages encrypted with k , L = length of max message

Suppose we want $\mathrm{Adv}_{\mathrm{CPA}}[A, E_{\mathrm{CBC}}] \le 1/2^{32}$ $\Leftarrow$ $q^2 L^2 / |X| < 1/ 2^{32}$

- AES: $|X| = 2^{128} \Rightarrow q L < 2^{48}$

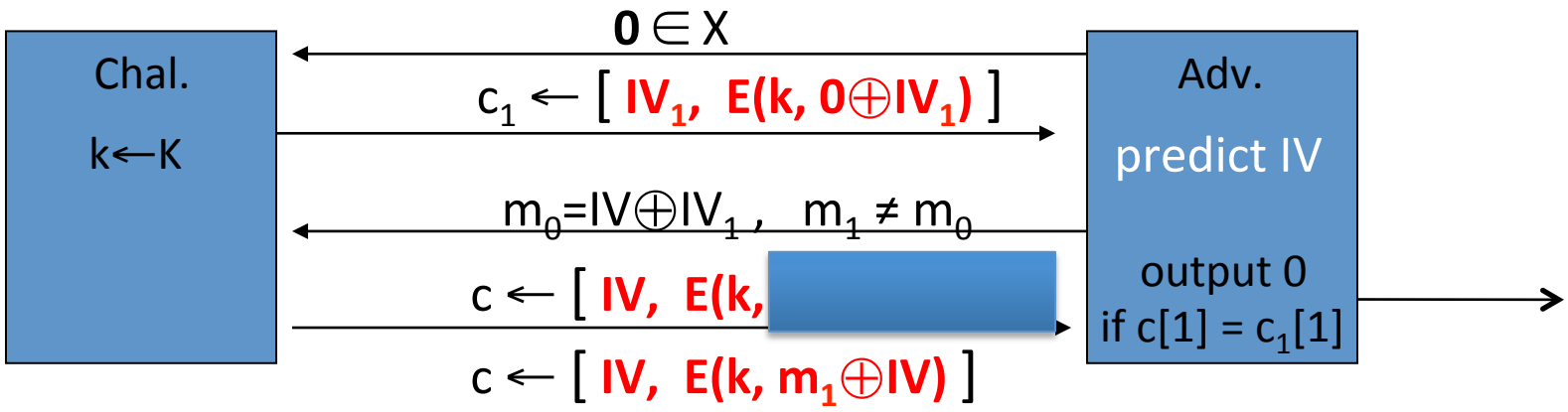
  So, after 2^{48} AES blocks, must change key

- 3DES: $|X| = 2^{64} \Rightarrow q L < 2^{16}$

Warning: an attack on CBC with rand. IV

CBC where attacker can predict the IV is not CPA-secure !!

Suppose given $c \leftarrow E_{CBC}(k,m)$ can predict IV for next message

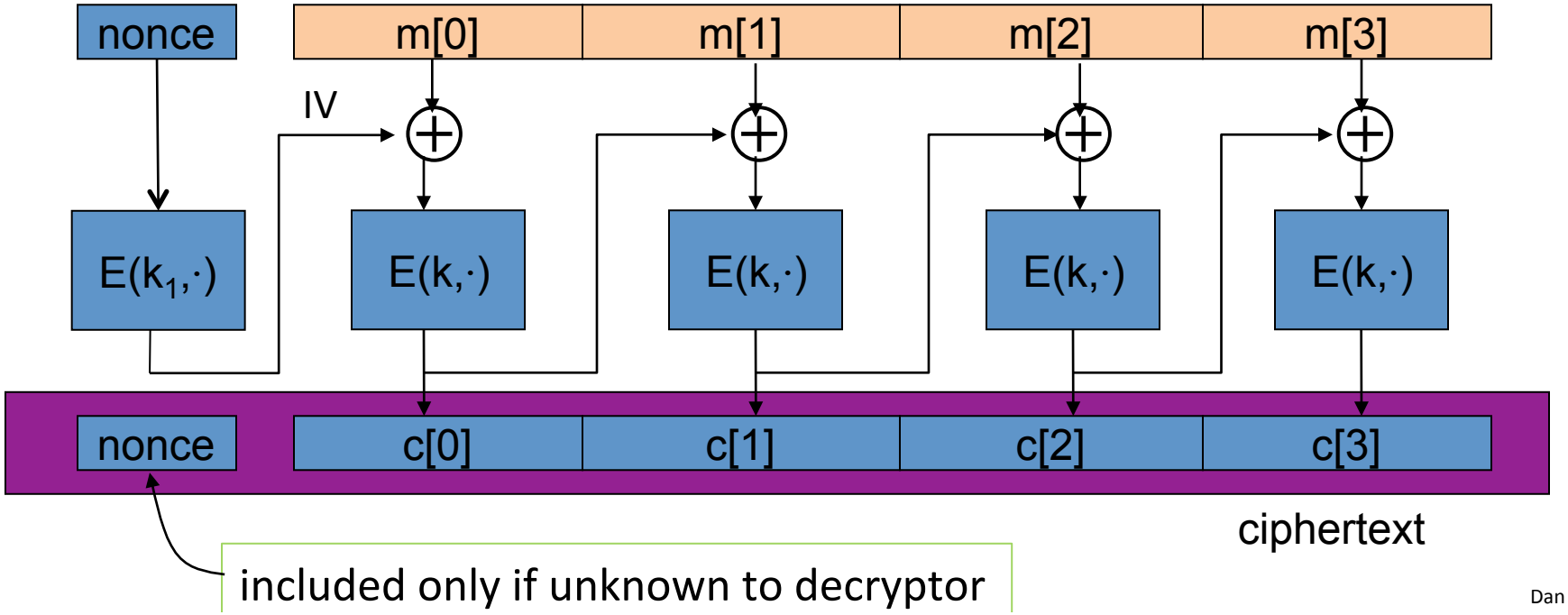


Bug in SSL/TLS 1.0: IV for record #i is last CT block of record #(i-1)

Construction 1': nonce-based CBC

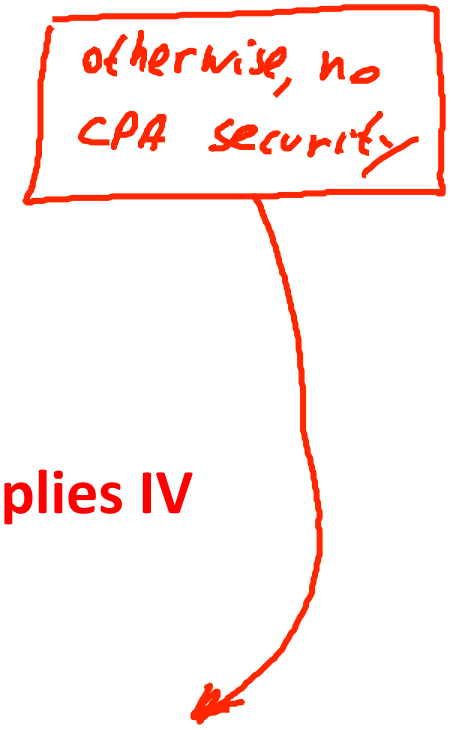
- Cipher block chaining with unique nonce: key = (k,k_1)

  unique nonce means: (key, n) pair is used for only one message



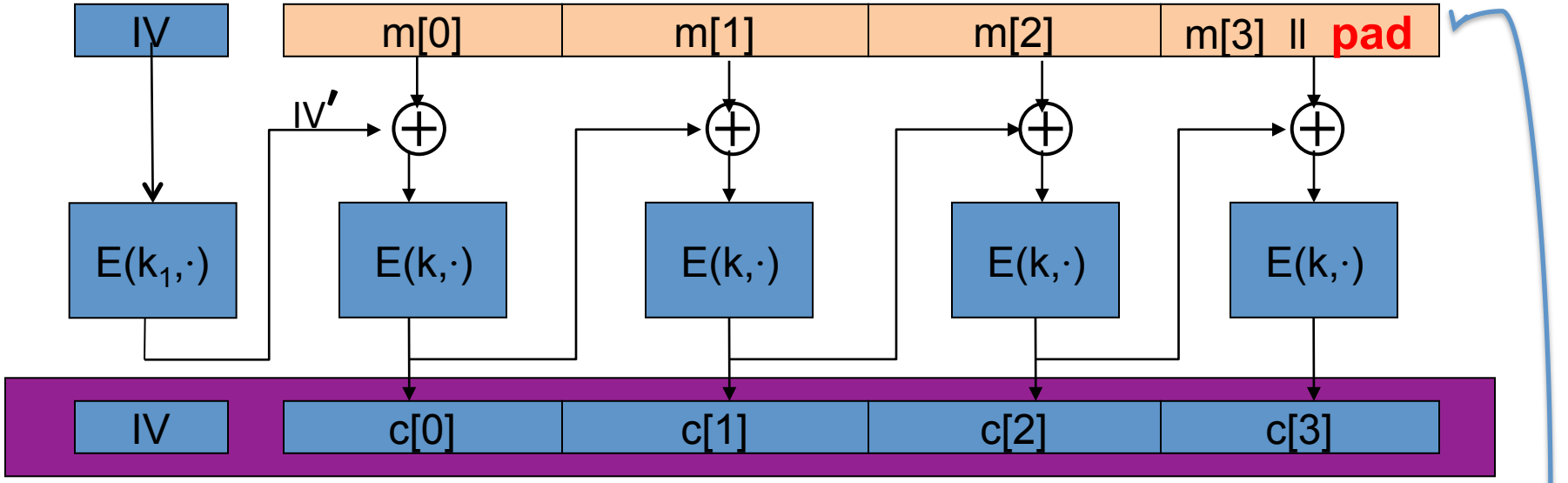
An example Crypto API (OpenSSL)

```
void AES_cbc_encrypt(
        const unsigned char *in,
        unsigned char *out,
        size_t length,
        const AES_KEY *key,
        unsigned char *ivec,          ← user supplies IV
        AES_ENCRYPT or AES_DECRYPT);
```



When nonce is non random need to encrypt it before use

A CBC technicality: padding



TLS: for n>0, n byte pad is

n	n	n	···	n

if no pad needed, add a dummy block

removed during decryption

End of Segment