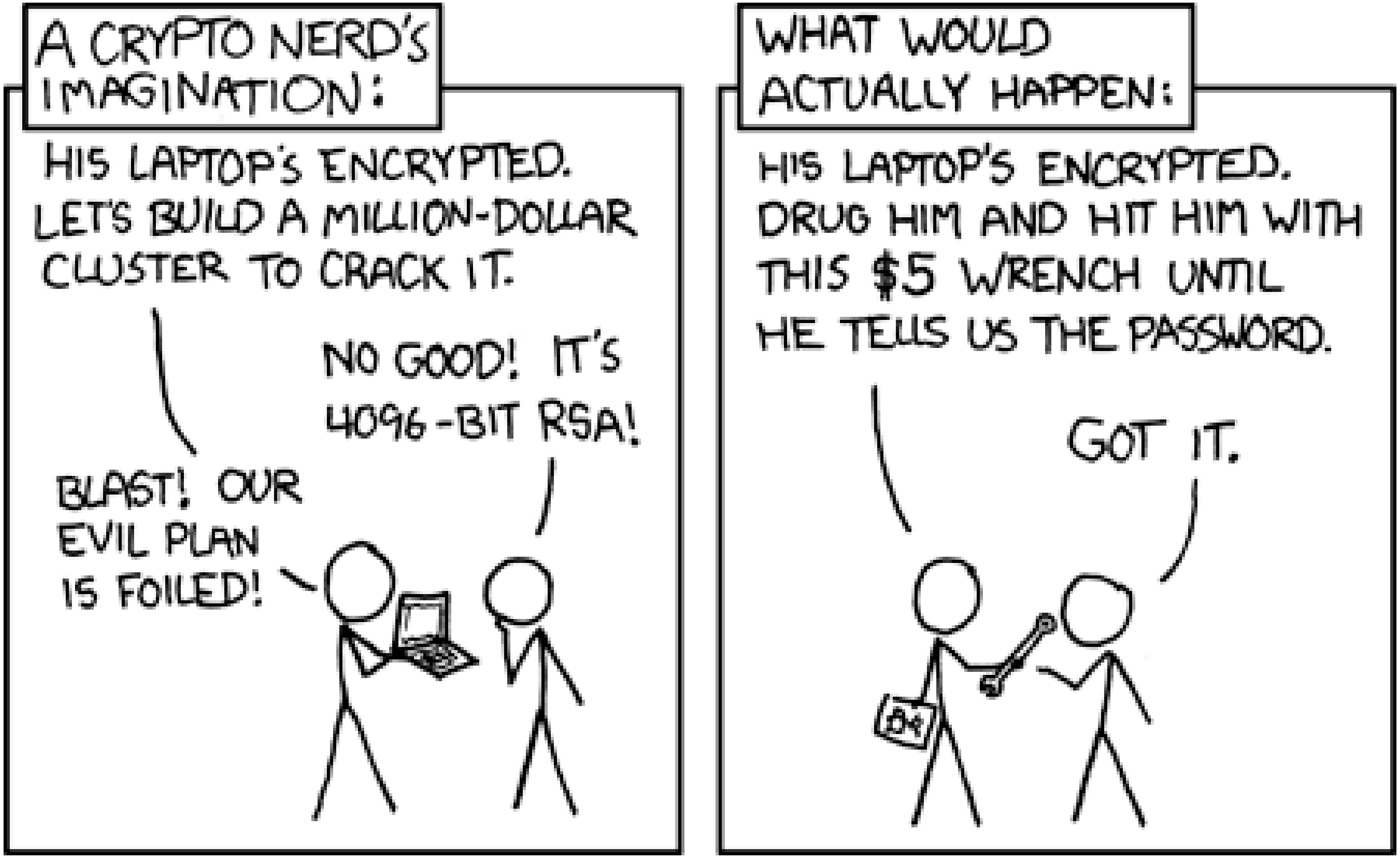
CS 161: Computer Security

Lecture 8

September 23, 2014

“How security works” (xkcd)



This lecture

- Symmetric Cryptography
- One-time pad
- DES
- Meet-in-the-middle & 3DES
- AES
- Non-repudiation and signatures
- HMAC
- Block modes of operation

“The enemy knows the system”

- (Claude) Shannon’s Maxim
  - “The enemy knows the system”

> the possible ones.
>
> To make the problem mathematically tractable we shall assume that *the enemy knows the system being used*. That is, he knows the family of transformations T_i, and the probabilities of choosing various keys. It might be

- (Auguste) Kerckhoff’s Principle
  - “[A cipher] should not require secrecy, and it should not be a problem if it falls into enemy hands”

> 2° Il faut qu’il n’exige pas le secret, et qu’il puisse sans inconvénient tomber entre les mains de l’ennemi ;

Security through obscurity

- Not required to publish the details of security
- But cannot depend on security of obscurity

- In cryptography, what is secret is keys

Attacks on ciphers

- Ciphertext only
  - Adversary has $E(m_1), E(m_2)$, ...
- Known plaintext
  - Adversary has $E(m_1)$ & $m_1, E(m_2)$& m_2, ...
- Chosen plaintext (offline)
  - Adversary picks m_1, m_2, ...
  - Adversary sees $E(m_1), E(m_2)$, ...
- Chosen plaintext (adaptive)
  - Adversary picks m_1 and sees $E(m_1)$
  - Then adversary picks m_2 and sees $E(m_2)$
- Chosen ciphertext (offline & adaptive)
  - Like chosen-plaintext, but adversary picks $E(m)$

Attacks on ciphers

- For general purpose ciphers, we want resistance against all attacks
    - Ciphertext only
    - Known plaintext
    - Chosen plaintext (offline & adaptive)
    - Chosen ciphertext (offline & adaptive)

Brute force attacks

- We can try all possible keys
- We can usually recognize valid plaintext

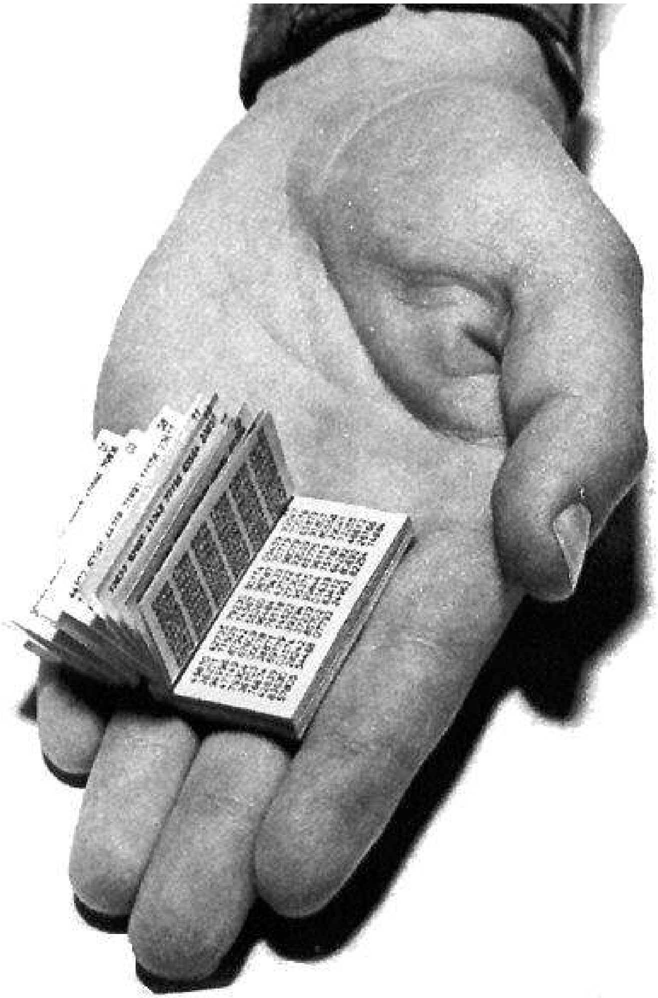
- NGGNPXNGQNJA vs ATTACKATDAWN
- Unicity distance
  - Minimum number of characters of ciphertext needed for a single intelligible plaintext

One time pad

- Key: a list of truly random bits
- Both Alice & Bob have this key
- $E(m) = m \ xor \ k \qquad D(c) = c \ xor \ k$
- Can only use key once
- Perfectly secure, because unicity distance ∞

$$m = 1010101010101010101010$$
$$k = 0011000000000101100100$$
$$c = 1001101010101111001110$$

One time pad



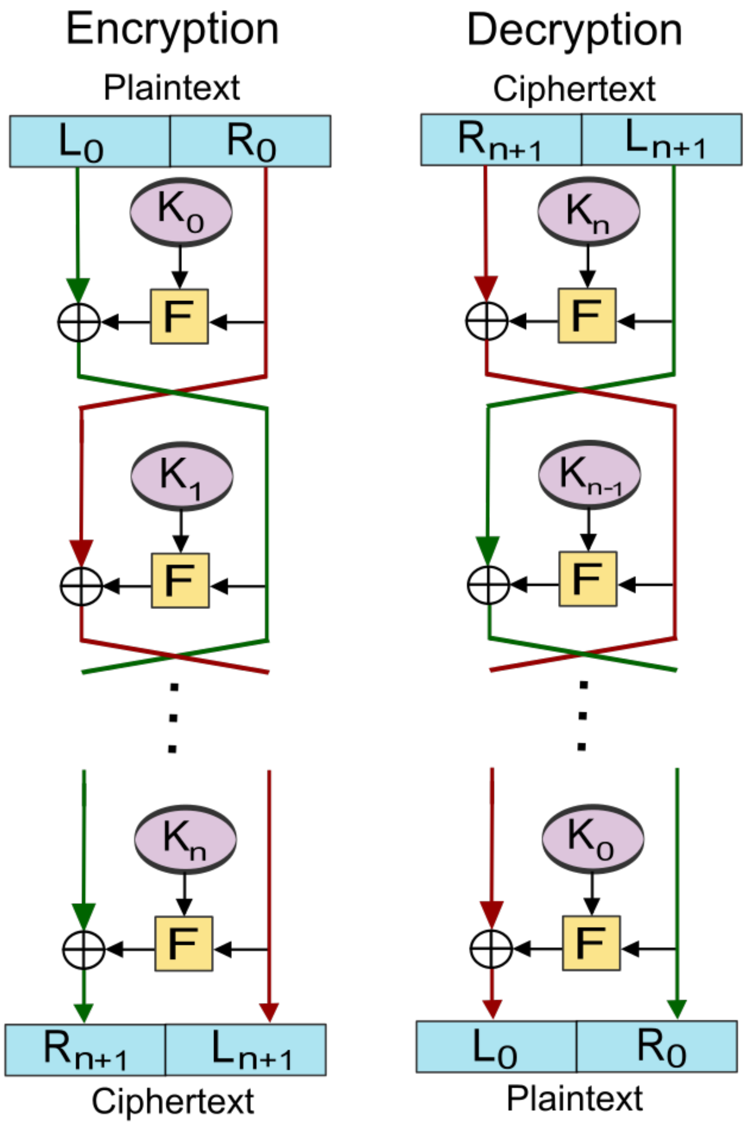
One time pad

- One time pad is perfectly secure
- But key size makes it impractical for use

Building blocks of ciphers

- Substitution cipher
  - ETW EWMME EWXETWMME
  - E→S M →L T →H W → E X →A
  - SHE SELLS SEASHELLS
  - monoalphabetic vs polyalphabetic
- Transposition cipher
  - Permutation of bytes (or bits) in a message
  - Gnitirw sdrawkcab

Feistel cipher



Encryption

Start with (L_0, R_0)

$L_{i+1} \leftarrow R_i$

$R_{i+1} \leftarrow L_i \; xor \; F(R_i, K_i)$

Decryption

Start with (L_{n+1}, R_{n+1})

$R_i \leftarrow L_{i+1}$

$L_i \leftarrow R_{i+1} \; xor \; F(L_{i+1}, K_i)$

DES - Data Encryption Standard (1977)

- Feistel cipher
- Works on 64 bit block with 56 bit keys
- Developed by IBM (Lucifer) improved by NSA
- Brute force attack feasible in 1997

Patches to DES

- Run DES three times

$$DES_{k_1}(DES_{k_2}^{-1}(DES_{k_1}(m)))$$

- Reverts to DES when $k_1 = k_2$
- But why not just twice?

$$DES_{k_2}(DES_{k_1}(m))$$

Meet in the middle attack

- Rewrite $c = DES_{k_2}(DES_{k_1}(m))$ as
  - (known plaintext)

$$x = DES_{k_2}^{-1}(c)$$

$$x = DES_{k_1}(m)$$

x	k_2

x	k_1

Find equal x values and corresponding k_1, k_2

Complexity of meet in the middle

- Each table has 2^{56} entries, so both tables take 2^{57}operations to generate
- Finding collision is easy

- Note: do not confuse
  - Meet in the middle
  - Man in the middle

AES – Advanced Encryption Standard (1997)

- Rijndael cipher
  - Joan Daemen & Vincent Rijmen
- Block size 128 bits
- Key can be 128, 192, or 256 bits

Non-repudiation

- Encrypting with AES cipher ensures other side knows key
- But encrypted messages not “proof in court”

- Digital signatures give us non-repudiation
- Proof that key-holder signed document

Review: issues w/ RSA signatures

- How does verifier check true value for d, n ?
  - Digital certificates Solved!
- What about large documents ($m > n$)?
  - Cryptographic hashes Solved!
- What if we want to **both** encrypt & sign?

- Signature (non-repudiation) first, then encrypt
  - Use two sets $\langle e, d, n \rangle$ and $\langle e', d', n' \rangle$

HMAC – keyed hash message authentication code

- Use a key in a crytpo-hash function
- Let H be a crypto hash function (SHA1, SHA2)
- K is a key padded with extra zeros;
- m is the message to be authenticated

$$HMAC(K, m) =$$
$$H((K\ xor\ OPAD)|H((K\ xor\ IPAD)|m))$$

- $OPAD = 0x5c5c \ldots 5c5c$
- $IPAD\ = 0x3636 \ldots 3636$

Modes of Operation

- Block ciphers encrypt fixed size blocks
  - eg. DES encrypts 64-bit blocks with 56-bit key
- Need to en/decrypt arbitrary amounts of data
- NIST SP 800-38A defines 5 modes
- **Block** and **stream** modes
- Cover a wide variety of applications
- Can be used with any block cipher

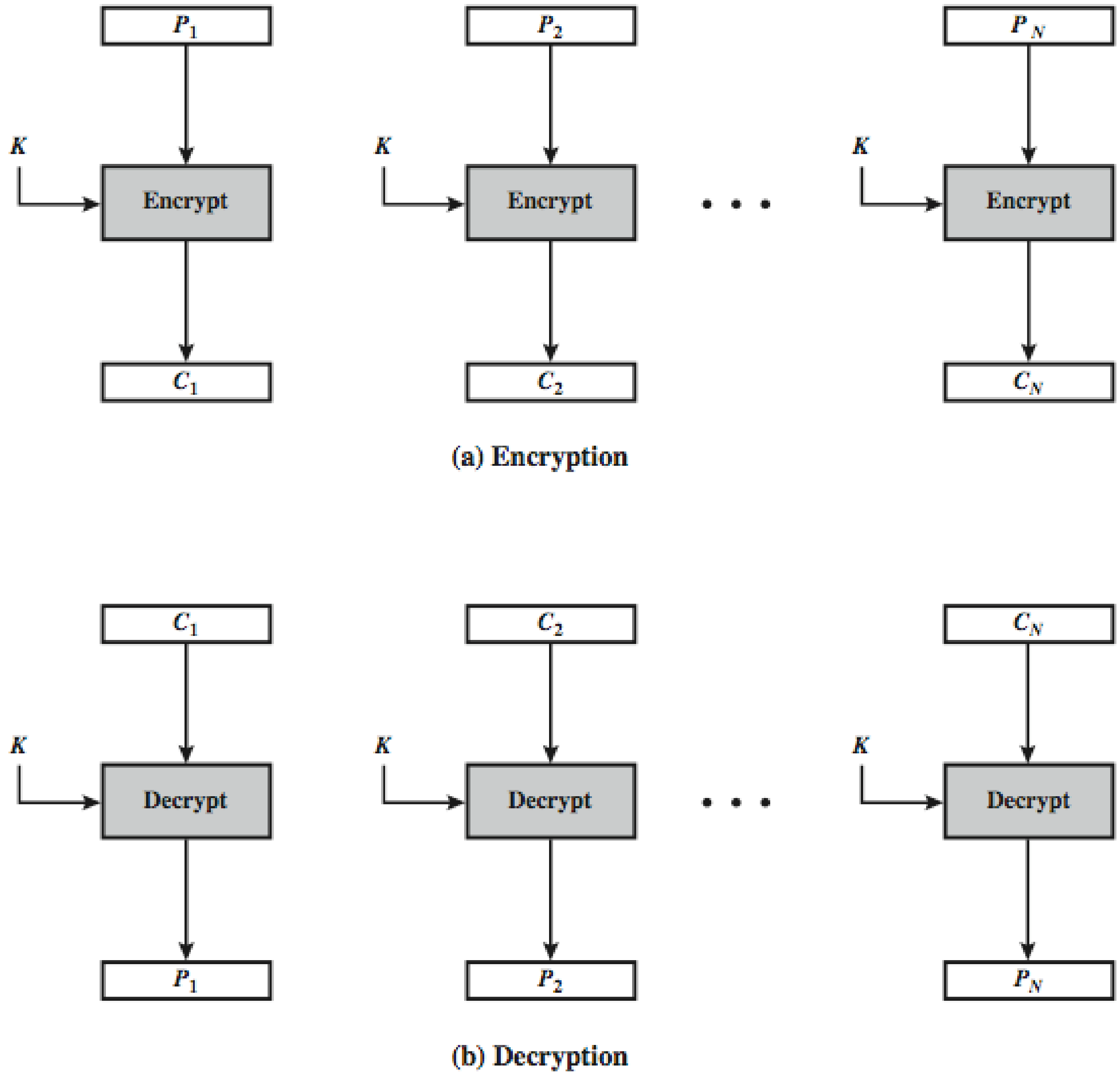
Electronic Codebook Book (ECB)

- Message is broken into independent blocks which are encrypted
- Each block is a value which is substituted, like a codebook
- Each block is encoded independently of the other blocks

$$C_i = E_K(P_i)$$

- Uses: secure transmission of single values

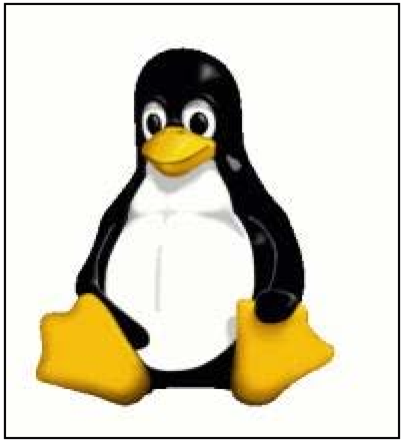
Electronic Codebook Book (ECB)



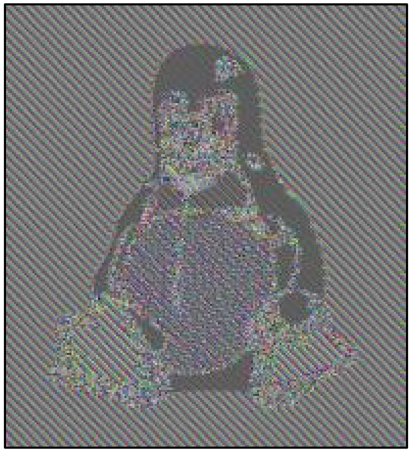
Advantages and Limitations of ECB

- Message repetitions may show in ciphertext
  - If aligned with message block
  - Particularly with data such graphics
  - Or with messages that change very little
- Encrypted message blocks independent
- Not recommended

Penguin ECB



original image



ECB

Cipher Block Chaining (CBC)

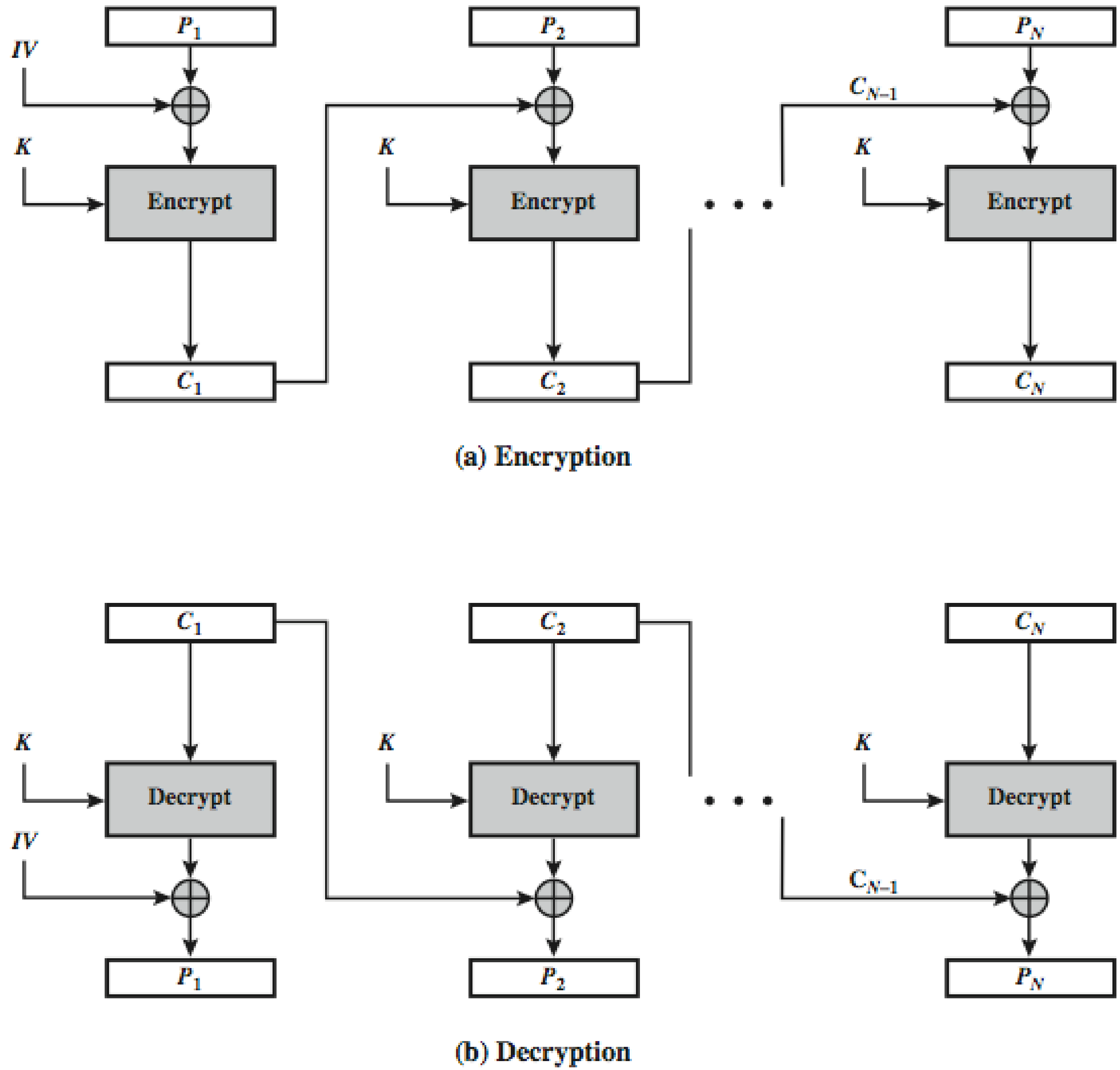
- Message broken into blocks
- Blocks “chained” in encryption
- Initial Vector (IV) to start process

  $$C_i = E_K(P_i \; xor \; C_{i-1})$$

  $$C_{-1} = IV$$

- uses: bulk data encryption, authentication

Cipher Block Chaining (CBC)



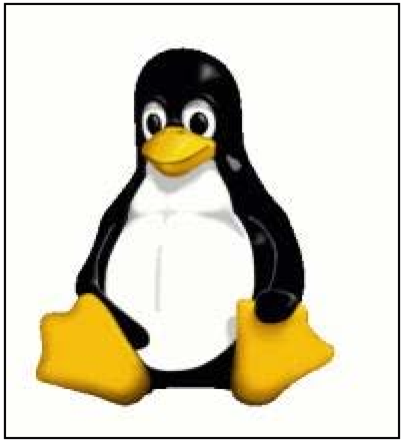
Message Padding

- End of message may be a short block
  - Not as large as cipher blocksize
  - Pad with known non-data value (eg nulls)
  - Or pad last block along with count of pad size
    - eg. [b1 b2 b3 0 0 0 0 5]
    - means have 3 data bytes, then 5 bytes pad+count
  - This may require an extra entire block over those in message
- There are other, more esoteric modes, which avoid the need for an extra block

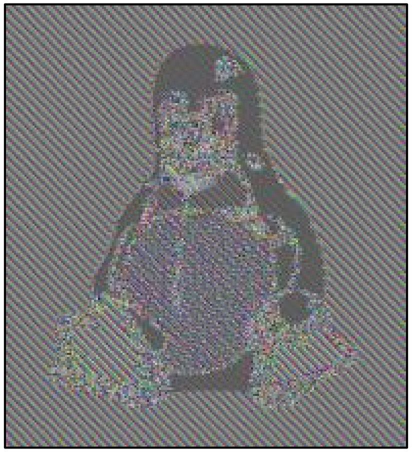
Advantages and Limitations of CBC

- Ciphertext block depends on **all** blocks before it
- Change to a block affects all following blocks
- Need **Initialization Vector** (IV)
  - Which must be known to sender & receiver
  - If sent in clear, attacker can change bits of first block, and change IV to compensate
  - So IV must either be a fixed value
  - Or must be sent encrypted in ECB mode before rest of message

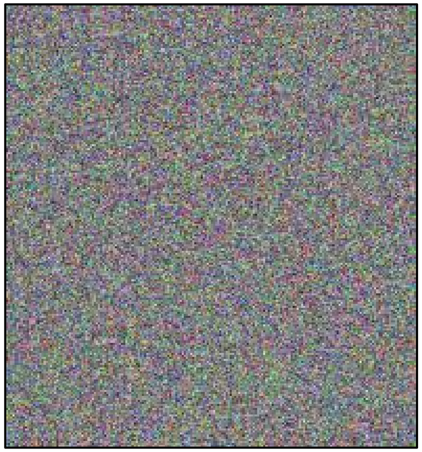
Penguin CBC



original image



ECB



CBC

Stream Modes of Operation

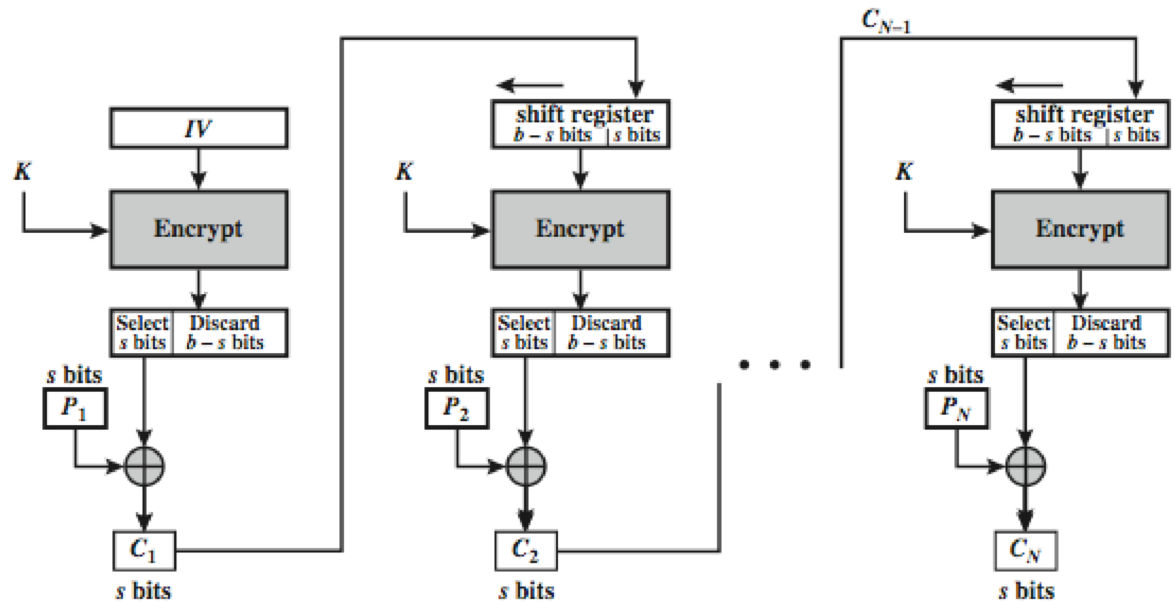
- Block modes encrypt entire block
- May need to operate on smaller units
  - Real time data
- Convert block cipher into stream cipher
  - Cipher feedback (CFB) mode
  - Output feedback (OFB) mode
  - Counter (CTR) mode
- Use block cipher as PRNG

Cipher FeedBack (CFB)

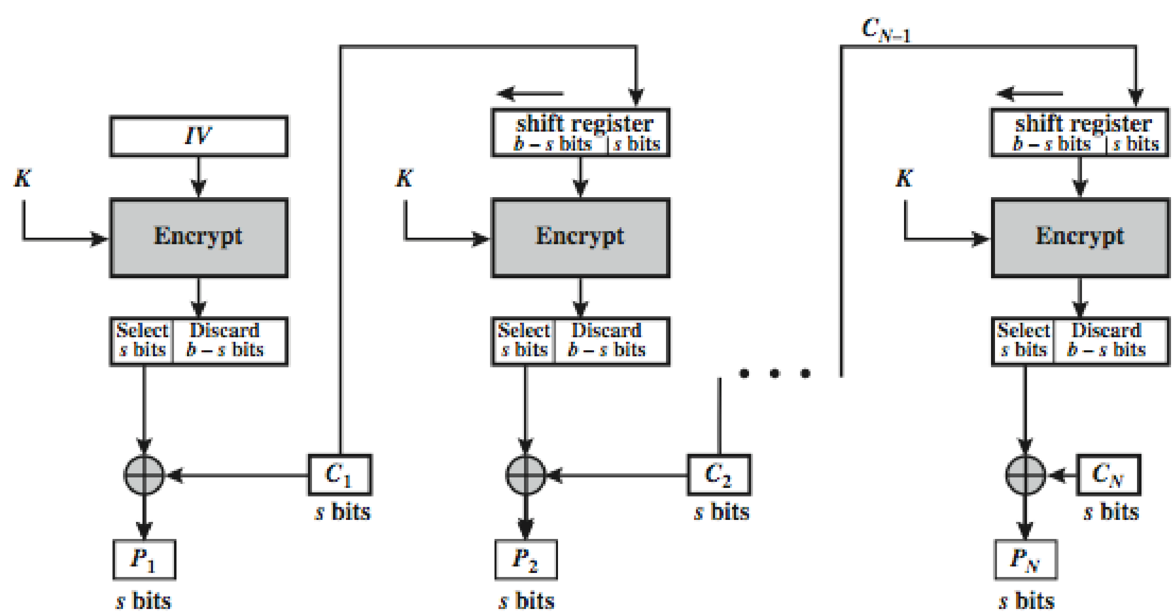
- Message is treated as a stream of bits
- Added to the output of the block cipher
- Result is feedback for next stage
- Standard allows any number of bit (1,8, 64 or 128 etc.) to be feedback
  - Denoted CFB-1, CFB-8, CFB-64, CFB-128 etc
- Most efficient to use all bits in block (64 or 128)

$$C_i = P_i \; xor \; E_K(C_{i-1})$$
$$C_{-1} = IV$$

s-bit Cipher FeedBack (CFB-s)



(a) Encryption



(b) Decryption

Advantages and Limitations of CFB

- Appropriate when data arrives in bits/bytes
- Most common stream mode
- Limitation is need to stall while do block encryption after every n-bits
- Note that the block cipher is used in **encryption** mode at **both** ends
- Errors propagate for several blocks after the error

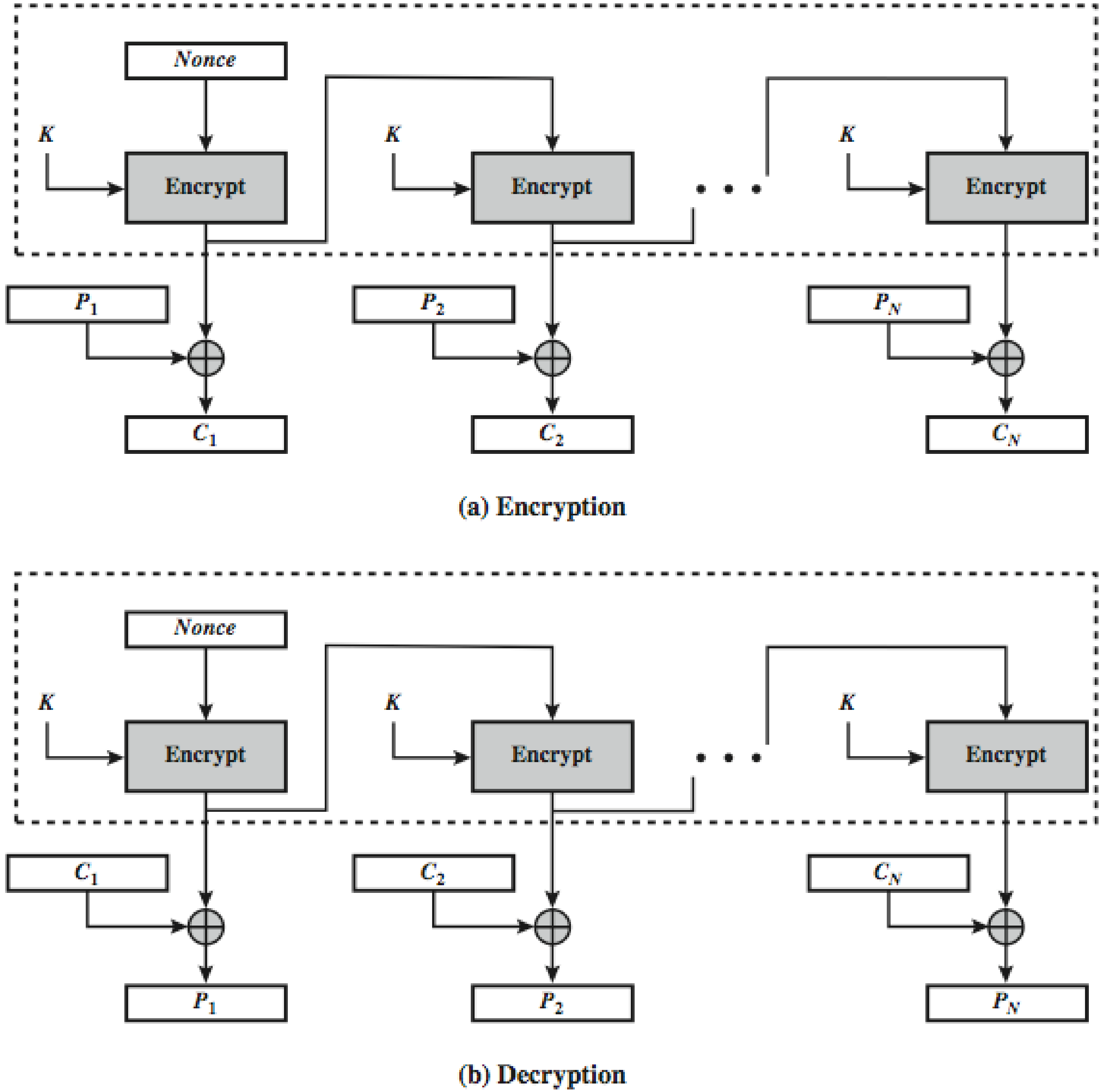
Output FeedBack (OFB)

- Message is treated as a stream of bits
- Output of cipher is added to message
- Output is then feedback
- Feedback is independent of message
- Can be computed in advance

$$O_i = E_K(O_{i-1})$$
$$C_i = P_i \; xor \; O_i$$
$$O_{-1} = IV$$

- Uses: stream encryption on noisy channels

Output FeedBack (OFB)



Advantages and Limitations of OFB

- Needs an IV which is unique for each use
  - If ever re-used attacker can recover outputs
- Bit errors do not propagate
- More vulnerable to message stream modification
- Sender & receiver must remain in sync
- Only use with full block feedback

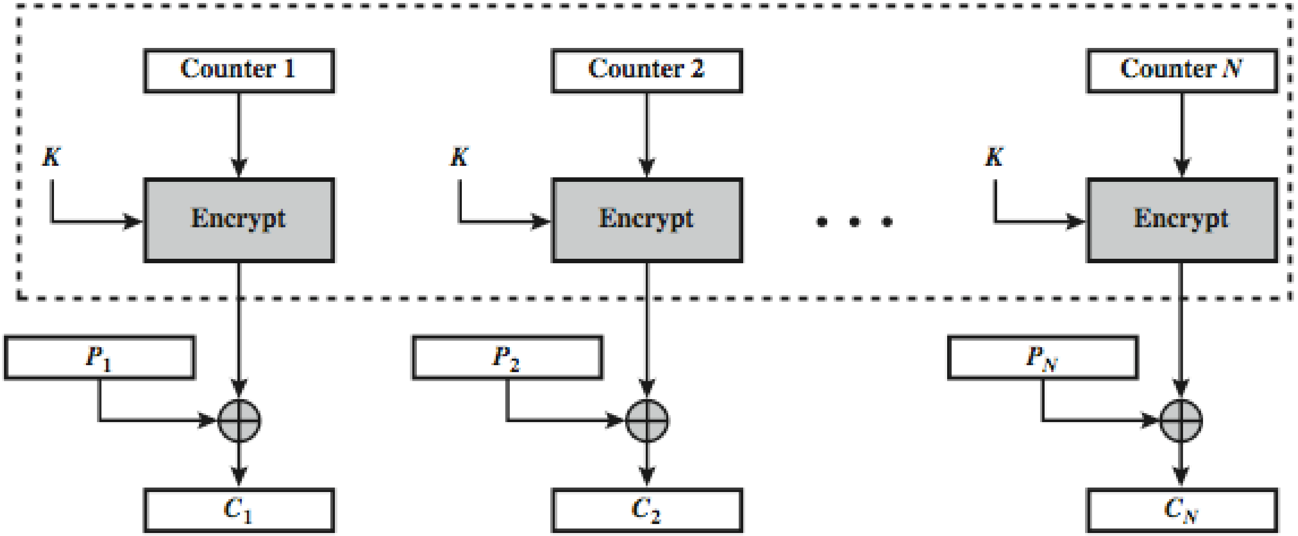
Counter (CTR)

- Similar to OFB but encrypts counter value rather than feedback value
- Must have a different key-counter value combination for every plaintext block (never reused)

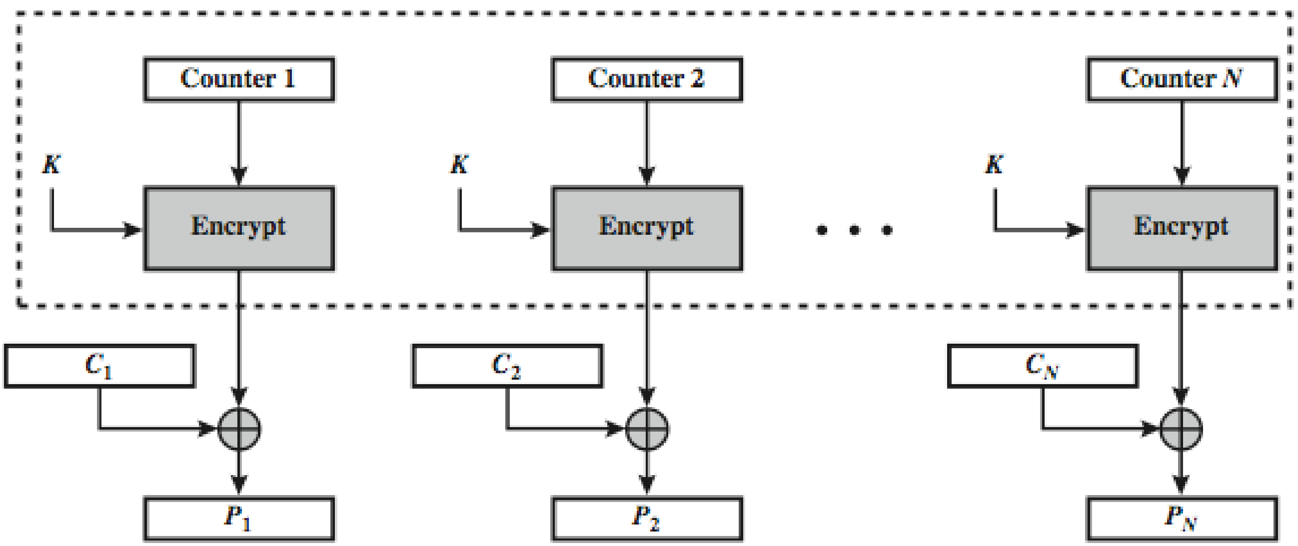
$$O_i = E_K(i)$$
$$C_i = P_i \text{ } xor \text{ } O_i$$

- Uses: high-speed network encryptions

Counter (CTR)



(a) Encryption



(b) Decryption

Advantages and Limitations of CTR

- Efficiency
  - Can do parallel encryptions in h/w or s/w
  - Can preprocess in advance of need
  - Good for bursty high speed links
- Random access to encrypted data blocks
- Must ensure never reuse key/counter values, otherwise could break