Security and Privacy in Wireless Networks

Mohammad Hossein Manshaei
manshaei@gmail.com

Introduction to Cryptographic Algorithms and Protocols

Appendix A

Security and Cooperation in Wireless Networks

Contents

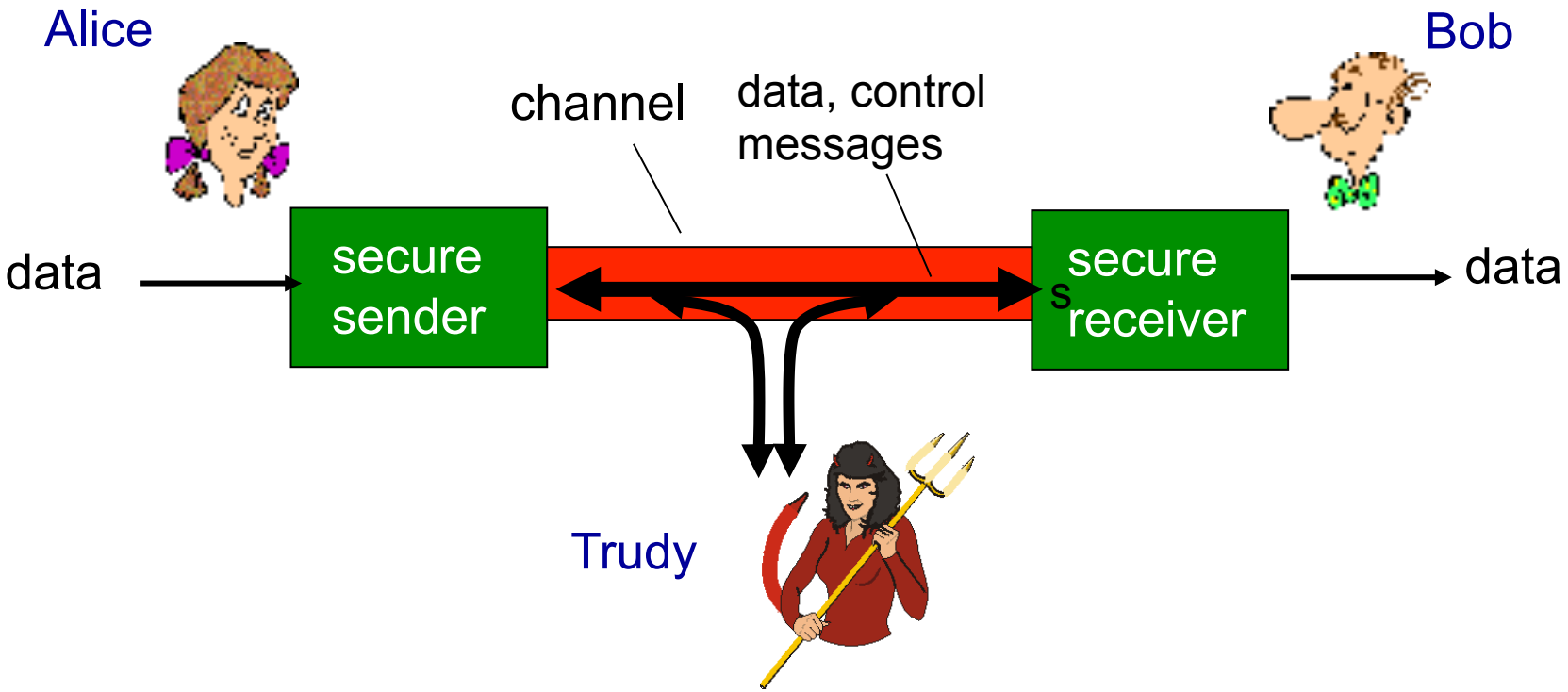
- Introduction to Cryptography
- Encryption and Adversary Model
- Symmetric Key Encryption
- Block Cipher
- Block Cipher Model of Operation

Introduction

- Security is about how to **prevent attacks**, or -- if prevention is not possible -- how to **detect attacks and recover** from them
- **An attack** is a *deliberate attempt* to compromise a system; it usually exploits weaknesses in the system's design, implementation, operation, or management
- **Attacks can be**
  - Passive
    - Attempts to learn or make use of information from the system but does not affect system resources
    - Examples: eavesdropping message contents, traffic analysis
    - Difficult to detect, should be prevented
  - Active
    - Attempts to alter system resources or affect their operation
    - Examples: masquerade (spoofing), replay, modification (substitution, insertion, destruction), denial of service
    - Difficult to prevent, should be detected

Friends and Enemies: Alice, Bob, Trudy

- Well-known in network security world
- Bob, Alice (lovers!) want to communicate “securely”
- Trudy (intruder) may intercept, delete, add messages



Who might Bob, Alice be?

- … well, *real-life* Bobs and Alices!
- Web browser/server for electronic transactions (e.g., on-line purchases)
- On-line banking client/server
- Wireless Eavesdropper
- DNS servers
- Routers exchanging routing table updates
- Other examples?

There are bad guys (and girls) out there!

Q: What can a "bad guy" do?

A: A lot!

- *eavesdrop:* intercept messages
- actively *insert* messages into connection
- *impersonation:* can fake (spoof) source address in packet (or any field in packet)
- *hijacking:* "take over" ongoing connection by removing sender or receiver, inserting himself in place
- *denial of service*: prevent service from being used by others (e.g., by overloading resources)

Main Security Services

- **Authentication**
  - aims to detect masquerade
  - provides assurance that a communicating entity is the one that it claims to be
- **Access control**
  - aims to prevent unauthorized access to resources
- **Confidentiality**
  - aims to protect data from unauthorized disclosure
  - usually based on encryption
- **Integrity**
  - aims to detect modification and replay
  - provides assurance that data received are exactly as sent by the sender
- **Non-repudiation**
  - Provides protection against denial by one entity involved in a communication of having participated in all or part of the communication
  - Two basic types: non-repudiation of origin and non-repudiation of delivery

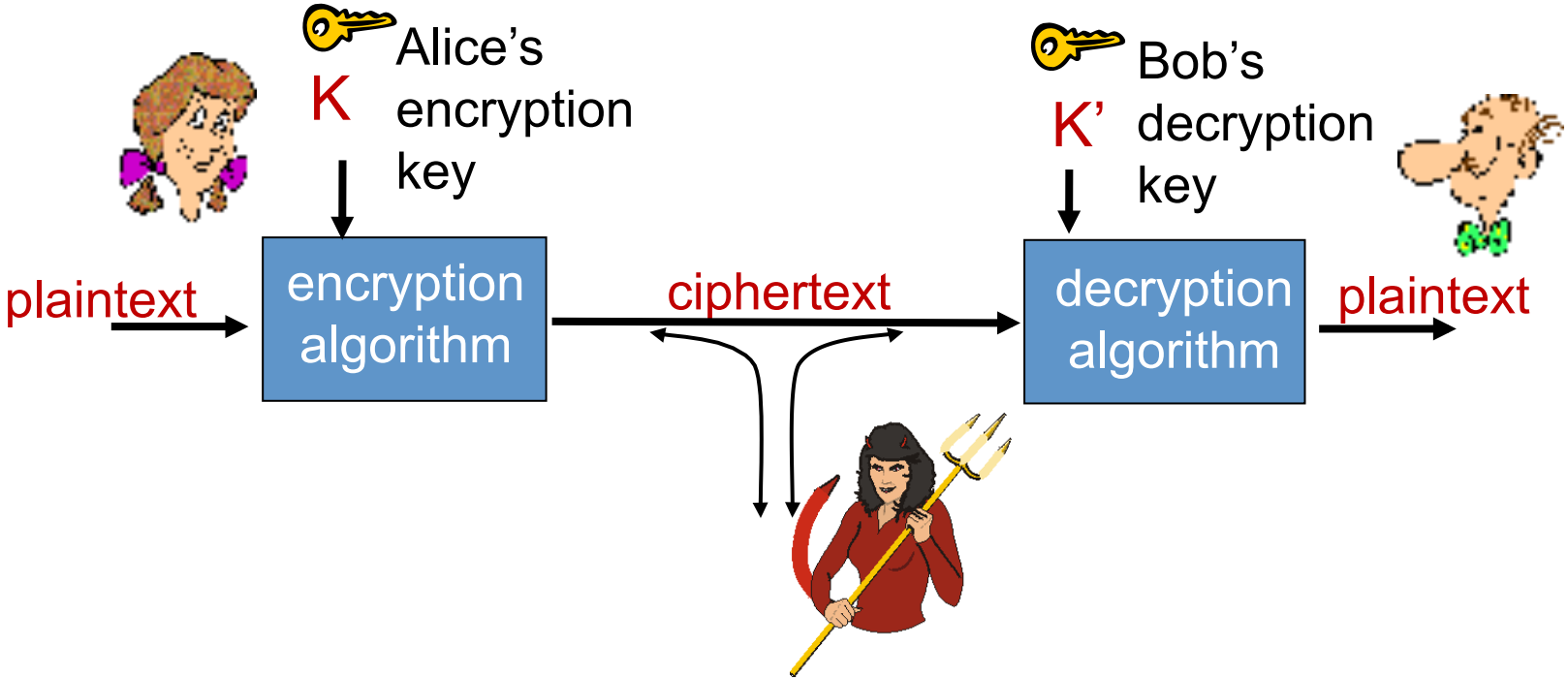
Some Security Mechanisms

- Encryption
  - symmetric key, asymmetric (public) key
- Digital signature
- Access control schemes
  - access control lists, capabilities, security labels, ...
- Data integrity mechanisms
  - message authentication codes, sequence numbering, time stamping, cryptographic chaining
- Authentication protocols
  - passwords, cryptographic challenge-response protocols, biometrics
- Traffic padding, route control, …

Contents

- Introduction to Cryptography
- Encryption and Adversary Model
- Symmetric Key Encryption
- Block Cipher
- Block Cipher Model of Operation

Classical Model of Encryption

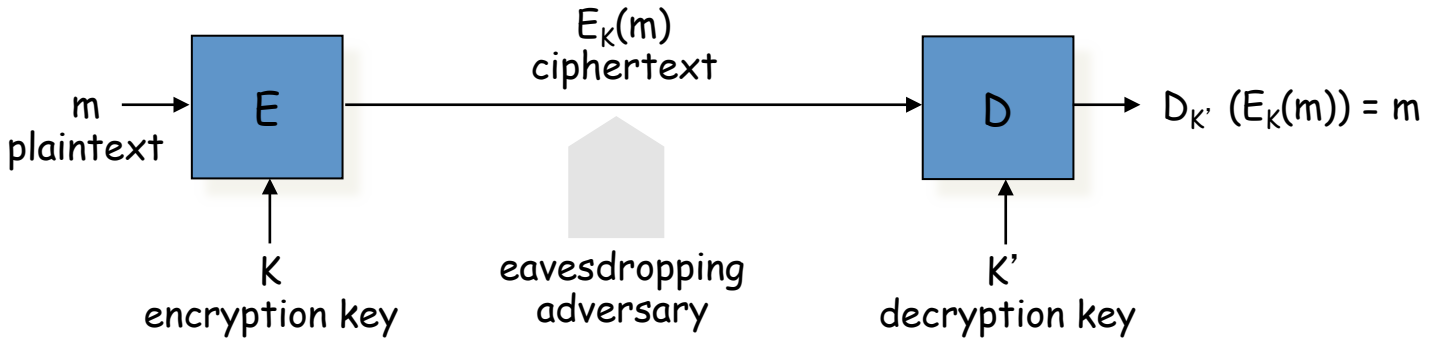


m plaintext message

$E_K(m)$ ciphertext, encrypted with key K_A

$m = D_{K'}(E_K(m))$

Classical Model of Encryption



- Goal of the adversary:
  - to systematically recover plaintexts from ciphertexts
  - to deduce the (decryption) key
- Kerckhoff's principle:
  - we must assume that the adversary knows all details of E and D
  - security of the system should be based on the protection of the decryption key

Adversary models

- Ciphertext-only attack
  - The adversary can only observe ciphertexts produced by the same encryption key
  - Two approaches: (1) **brute force**: search through all keys
    (2) **statistical analysis**
- Known-plaintext attack
  - the adversary can obtain corresponding plaintext-ciphertext pairs produced with the same encryption key
- (Adaptive) Chosen-plaintext attack
  - The adversary can choose plaintexts and obtain the corresponding ciphertexts
- (Adaptive) Chosen-ciphertext attack
  - The adversary can choose ciphertexts and obtain the corresponding plaintexts
- Related-key attack
  - the adversary can obtain ciphertexts, or plaintext-ciphertext pairs that are produced with different encryption keys that are related in a known way to a specific encryption key

Security of Encryption Schemes

- An encryption scheme is secure in a given adversary model if it is computationally infeasible for the adversary to determine the target decryption key under the assumptions of the given model

- For many encryption schemes used in practice, no proof of security exists
  - ✓ these schemes are used, nevertheless, because they are efficient and they resist all known attacks

- Some encryption schemes are provably secure, however these schemes are **often inefficient**

Basic Classification Encryption Schemes

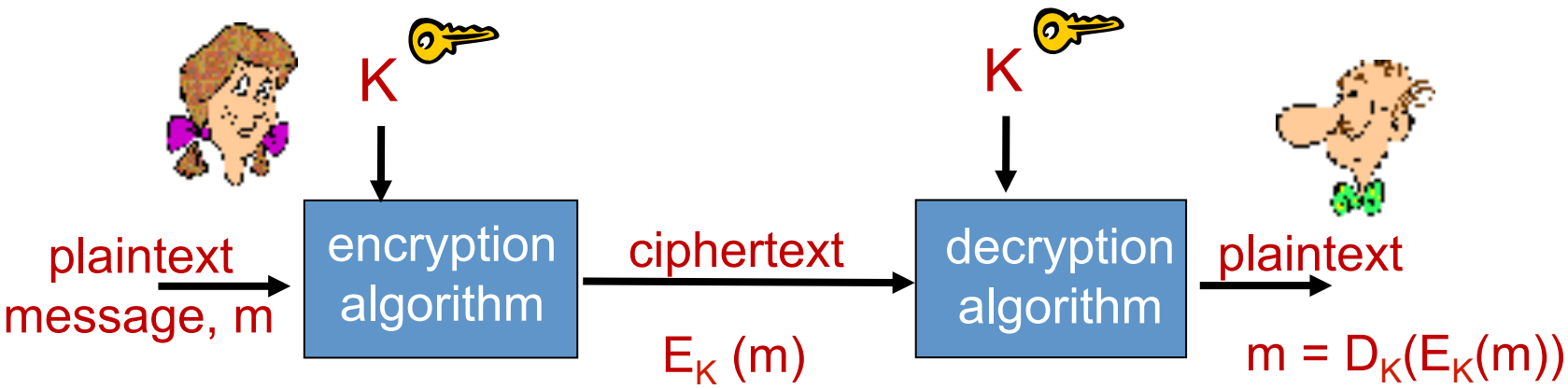
- Symmetric-key encryption
  - It is easy to compute K' from K (and vice versa)
  - Usually K' = K
  - Two main types:
    - **Stream ciphers** – operate on individual characters of the plaintext
    - **Block ciphers** – process the plaintext in larger blocks of characters

- Asymmetric-key encryption
  - it is hard (computationally infeasible) to compute K' from K
  - K can be made public (→ public-key cryptography)

Contents

- Introduction to Cryptography
- Encryption and Adversary Model
- Symmetric Key Encryption
- Block Cipher
- Block Cipher Model of Operation

Symmetric Key Cryptography



symmetric key crypto: Bob and Alice share same (symmetric) key: K

- e.g., key is knowing substitution pattern in mono alphabetic substitution cipher

Q: how do Bob and Alice agree on key value?

Simple Encryption Scheme

substitution cipher: substituting one thing for another

- monoalphabetic cipher: substitute one letter for another

```
plaintext:   abcdefghijklmnopqrstuvwxyz
                ↓                   ↓
ciphertext:  mnbvcxzasdfghjklpoiuytrewq
```

e.g.:

```
Plaintext: bob. i love you. alice
ciphertext: nkn. s gktc wky. mgsbc
```

Encryption key: mapping from set of 26 letters to set of 26 letters

26!: # of possible keys. (Note: There exists statistical analysis)

A More Sophisticated Encryption Approach

- Main Idea: Same letter appearing in different places will be encoded differently
- n substitution ciphers, $M_1, M_2, \ldots, M_n$
- Cycling pattern:
  - e.g., n=4: M_1, M_3, M_4, M_3, M_2; M_1, M_3, M_4, M_3, M_2; ..
- For each new plaintext symbol, use subsequent substitution pattern in cyclic pattern
  - dog: d from M_1, o from M_3, g from M_4

Encryption key: n substitution ciphers, and cyclic pattern

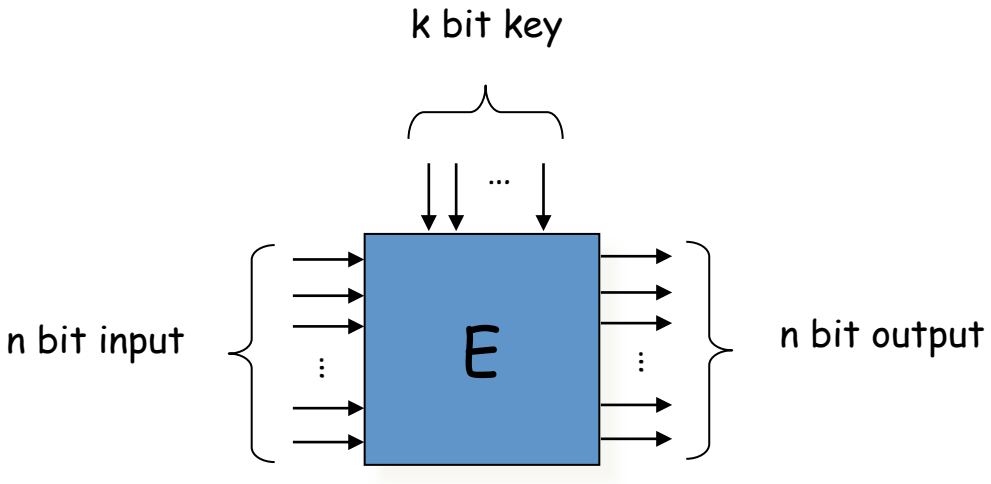
- key need not be just n-bit pattern

Contents

- Introduction to Cryptography
- Encryption and Adversary Model
- Symmetric Key Encryption
- Block Cipher
- Block Cipher Model of Operation

Block Ciphers

An *n* bit block cipher is a function E: $\{0, 1\}^n$ x $\{0, 1\}^k$ → $\{0, 1\}^n$, such that for each $K \in \{0,1\}^k$, $E(x, K) = E_K(x)$ is an invertible mapping from $\{0,1\}^n$ to $\{0,1\}^n$



Block Ciphers: Simple Example

- Message to be encrypted is processed in blocks of k bits (e.g., 3-bit blocks).

- 1-to-1 mapping is used to map k-bit block of plaintext to k-bit block of ciphertext

- What is the ciphertext for 010110001111 ?
  - 101000111001

Input	Output
000	110
001	111
010	101
011	100
100	011
101	010
110	000
111	001

Number of possible Keys = Number of possible mapping
8!= 40320

Brute-Force Attack →

It tries to decrypt ciphertext with all mapping (keys)

Exhaustive Key Search Attack

- Given a small number of plaintext-ciphertext pairs encrypted under a key K, K can be recovered by exhaustive key search with 2^{k-1} processing complexity (expected number of operations)
  - input: (X, Y), (X', Y'), …
  - progress through the entire key space, and for each candidate key K', do the following:
    - decrypt Y with K'
    - if the result is not X, then throw away K'
    - if the result is X, then check the other pairs (X', Y'), …
    - if K' does not work for at least one pair, then throw away K' and take another key
    - if K' worked for all pairs (X, Y), (X', Y'), …, then output K' as the target key
  - On average, the target key is found after searching half of the key space

- If the plaintexts are known to contain redundancy, then ciphertext-only exhaustive key search is possible with a relatively small number of ciphertexts

→ in practice, key size should be at least 128 bits

Algebraic Attacks

- Having a large key size is *only a necessary condition* for the security of a block cipher
  - A block cipher can be broken due to the weaknesses in its internal (algebraic) structure, even if it uses large keys

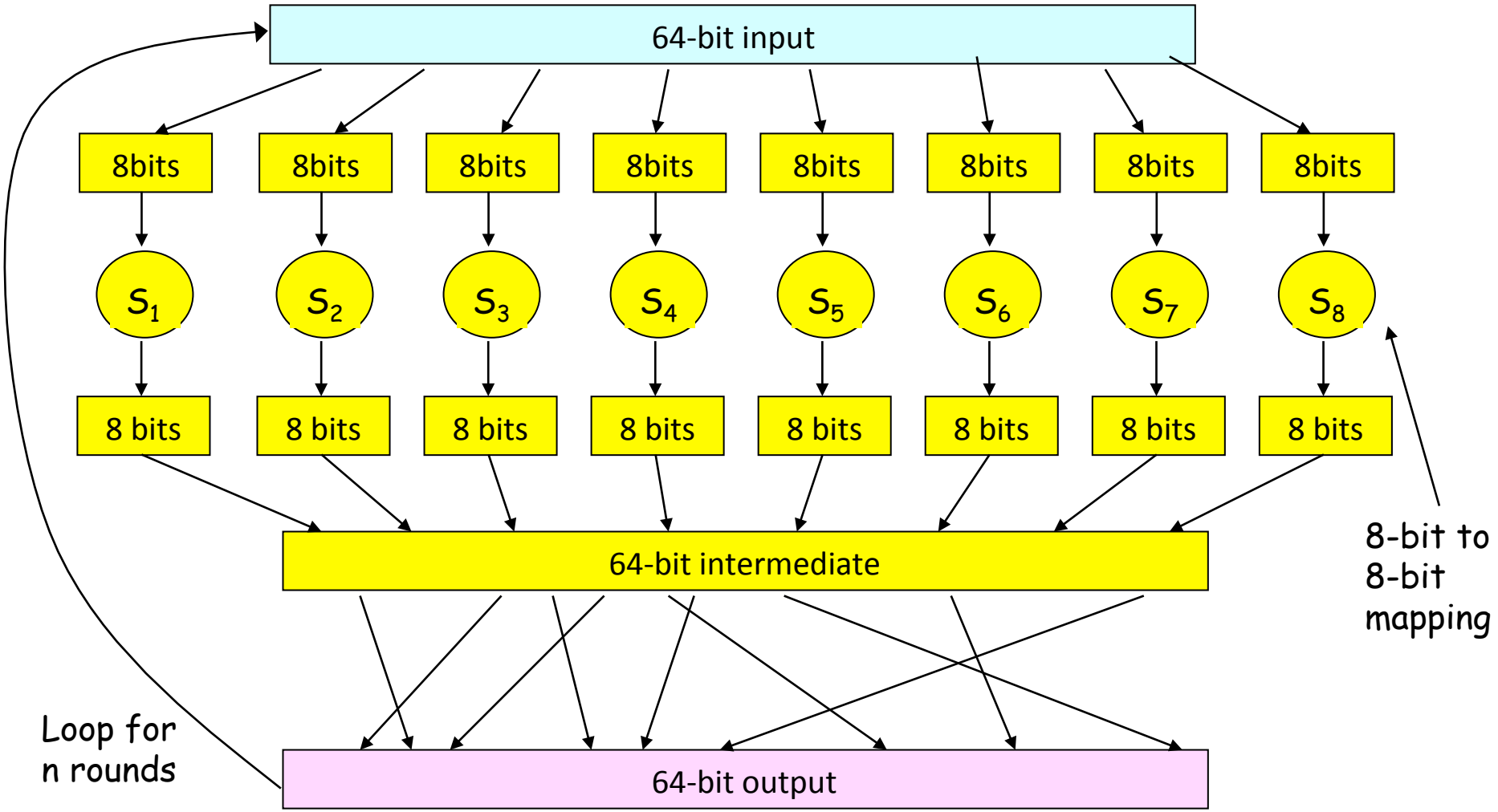
- **Example:**
  - Naïve exhaustive key search against DES: 2^{55}
  - Attack using the complementation property of DES: 2^{54}

    $Y = DES_K(X)$ implies $Y^* = DES_{K^*}(X^*)$, where X^* denotes the bitwise complement of X
  - Differential cryptanalysis of DES: 2^{47}
  - Linear cryptanalysis of DES: 2^{43}

Block Ciphers: Key and Implementation

- How many possible mappings are there for k=3?
  - How many 3-bit inputs?
  - How many permutations of the 3-bit inputs?
  - Answer: 40,320 ; not very many! (Number of Key)
- In general, $2^k!$ mappings; huge for k=64
- Problem:
  - Table approach requires table with 2^{64} entries, each entry with 64 bits
- Table too big: instead use function that simulates a randomly permuted table

Prototype Function



Why rounds in prototpe?

- If only a single round, then one bit of input affects at most 8 bits of output.
- In 2nd round, the 8 affected bits get scattered and inputted into multiple substitution boxes.
- How many rounds?
  - How many times do you need to shuffle cards
  - Becomes less efficient as n increases

Encrypting a Large Message

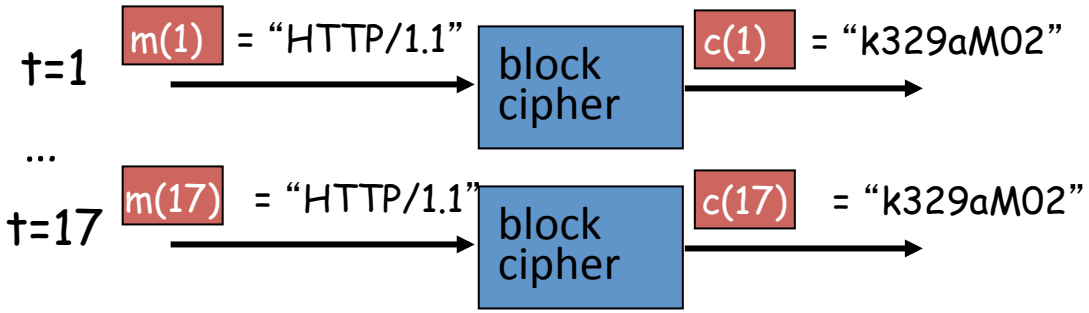
- Why not just break message in 64-bit blocks, encrypt each block separately?
  - If same block of plaintext appears twice, will give same ciphertext.
- How about:
  - Generate random 64-bit number r(i) for each plaintext block m(i)
  - Calculate $c(i) = E_K(m(i) \oplus r(i))$
  - Transmit c(i), r(i), i=1,2,…
  - At receiver: $m(i) = D_K(c(i)) \oplus r(i)$
  - Problem: inefficient, need to send c(i) and r(i)

Cipher Block Chaining (CBC)

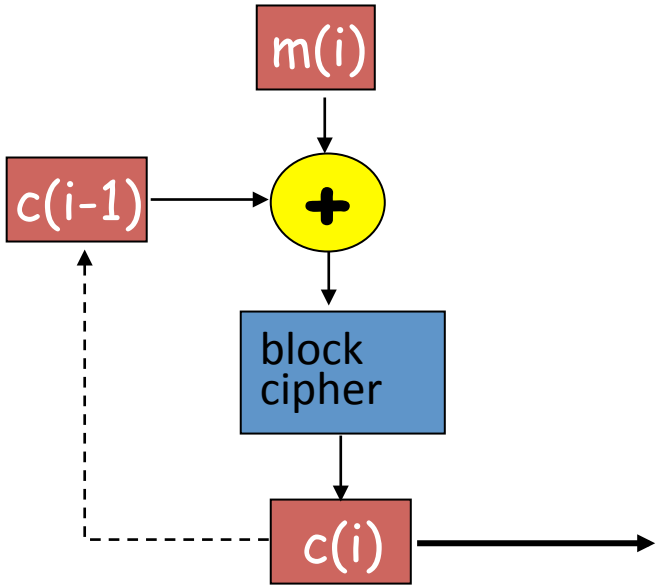
- CBC generates its own random numbers
  - Have encryption of current block depend on result of previous block
  - $c(i) = E_K(m(i) \oplus c(i-1))$
  - $m(i) = D_K(c(i)) \oplus c(i-1)$
- How do we encrypt first block?
  - **Initialization vector (IV):** random block = c(0)
  - IV does not have to be secret
- Change IV for each message (or session)
  - Guarantees that even if the same message is sent repeatedly, the ciphertext will be completely different each time

Cipher Block Chaining

- Cipher block: if input block repeated, will produce same cipher text:



- *Cipher block chaining:* XOR i^{th} input block, m(i), with previous block of cipher text, c(i-1)
  - c(0) transmitted to receiver in clear



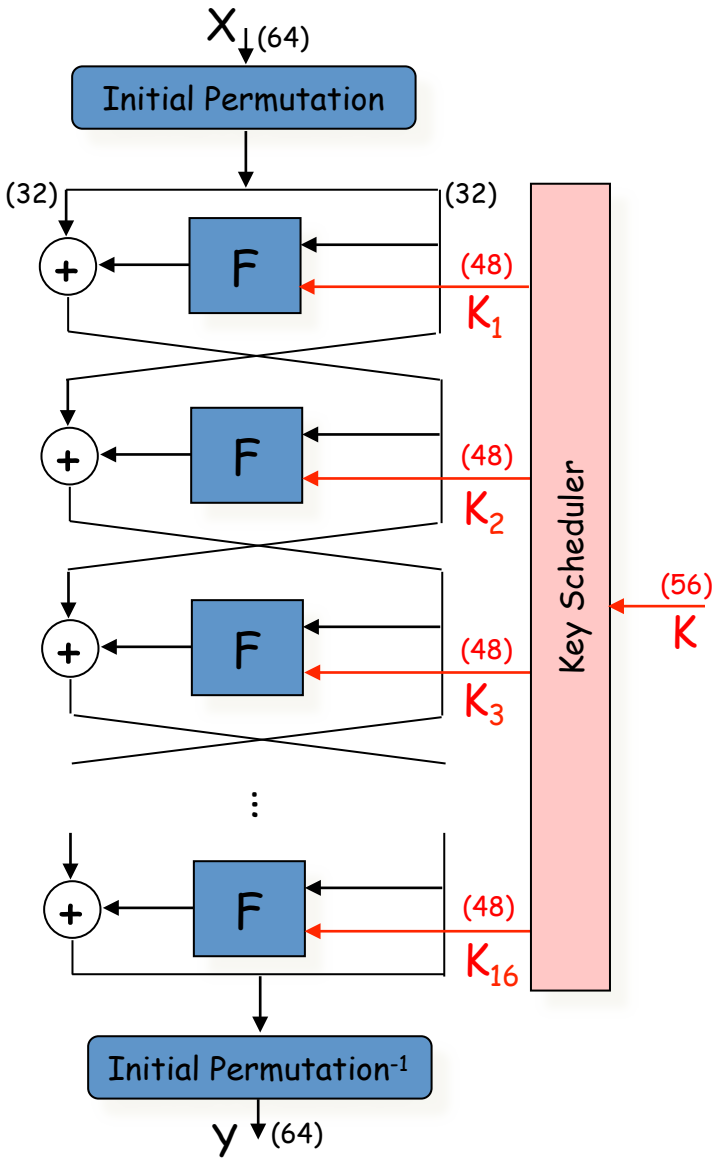
Symmetric Key Crypto: DES

DES: Data Encryption Standard

- US encryption standard [NIST 1993]
- 56-bit symmetric key, 64-bit plaintext input
- Block cipher with cipher block chaining
- How secure is DES?
  - DES Challenge: 56-bit-key-encrypted phrase decrypted (brute force) in less than a day
  - No known good analytic attack
- Making DES more secure:
  - 3DES: encrypt 3 times with 3 different keys

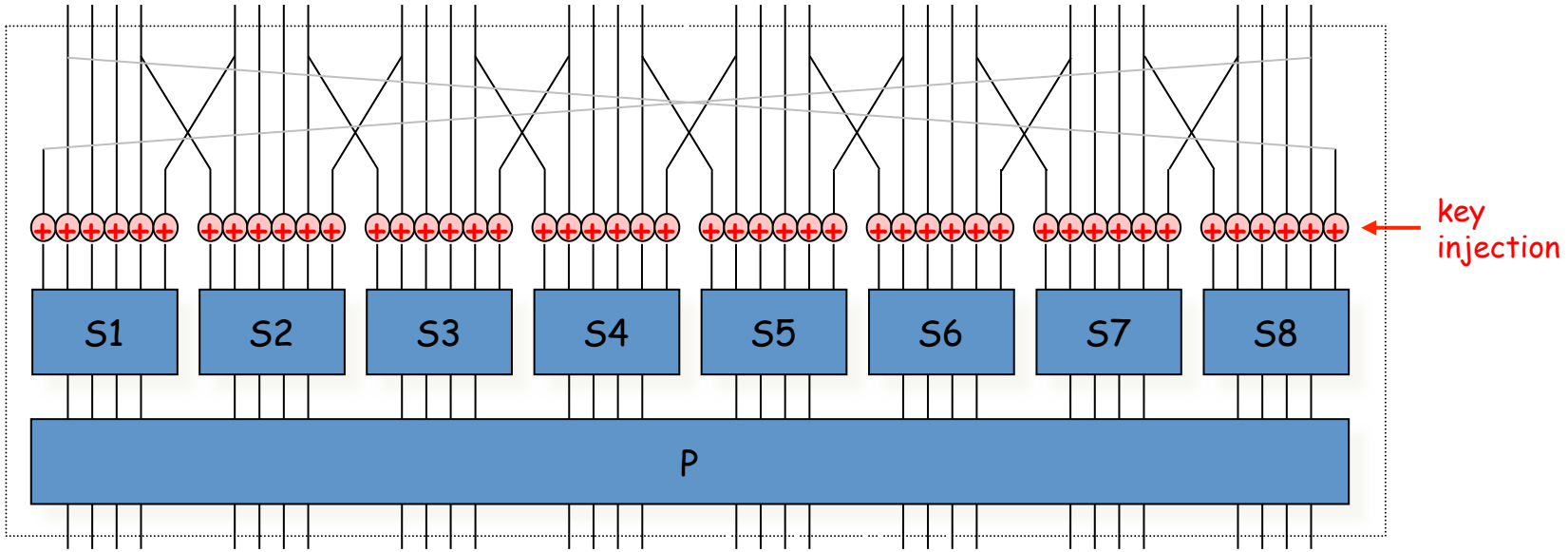
  (actually encrypt, decrypt, encrypt)

DES (Data Encryption Standard)



- input size: 64
- output size: 64
- key size: 56
- 16 rounds
- Feistel structure

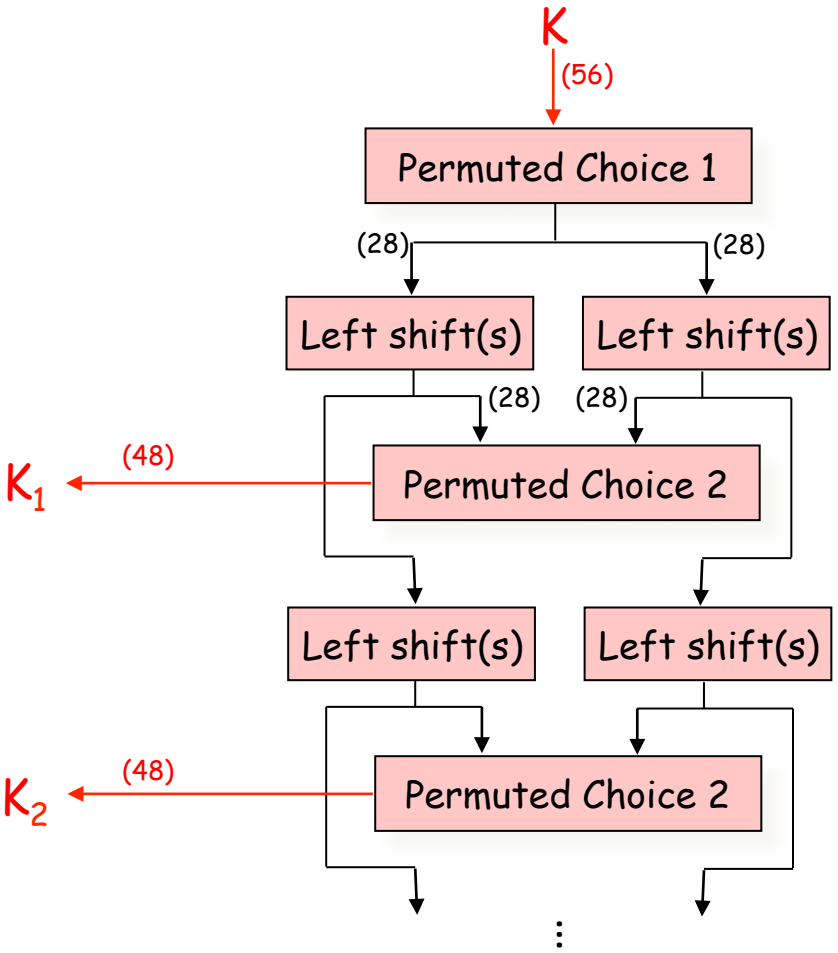
DES Round Function F



Si – Substitution box (S-box)

P – Permutation box (P-box)

DES key scheduler



each key bit is used in around 14 out of 16 rounds

Block Cipher Design Criteria

- **Completeness**
  - Each bit of the output block should depend on each bit of the input block and on each bit of the key
- **Avalanche Effect**
  - Changing one bit in the input block should change approximately half of the bits in the output block
  - Similarly, changing one key bit should result in the change of approximately half of the bits in the output block
- **Statistical Independence**
  - Input and output should appear to be statistically independent

How to satisfy the design criteria?

- Complex encryption function can be built by composing several simple operations which offer complementary – but individually insufficient – protection

- **Simple operations:**
  - elementary arithmetic operations
  - logical operations (e.g., XOR)
  - modular multiplication
  - transpositions
  - substitutions
  - ...

- Combine two or more transformations in a manner that the resulting cipher is more secure than the individual components

AES: Advanced Encryption Standard

- New (Nov. 2001) symmetric-key NIST standard, replacing DES
- processes data in 128 bit blocks
- 128, 192, or 256 bit keys
- brute force decryption (try each key) taking 1 sec on DES, takes 149 trillion years for AES

Contents

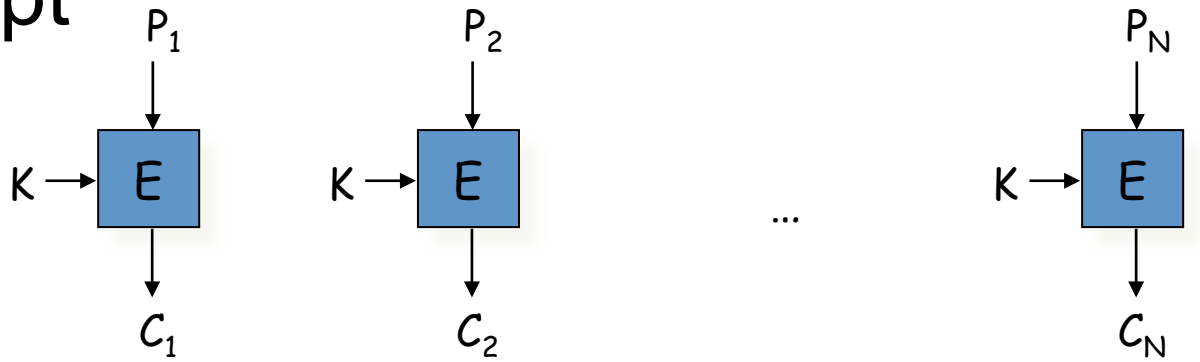
- Introduction to Cryptography
- Encryption and Adversary Model
- Symmetric Key Encryption
- Block Cipher
- Block Cipher Model of Operation

Block Cipher Modes of Operation

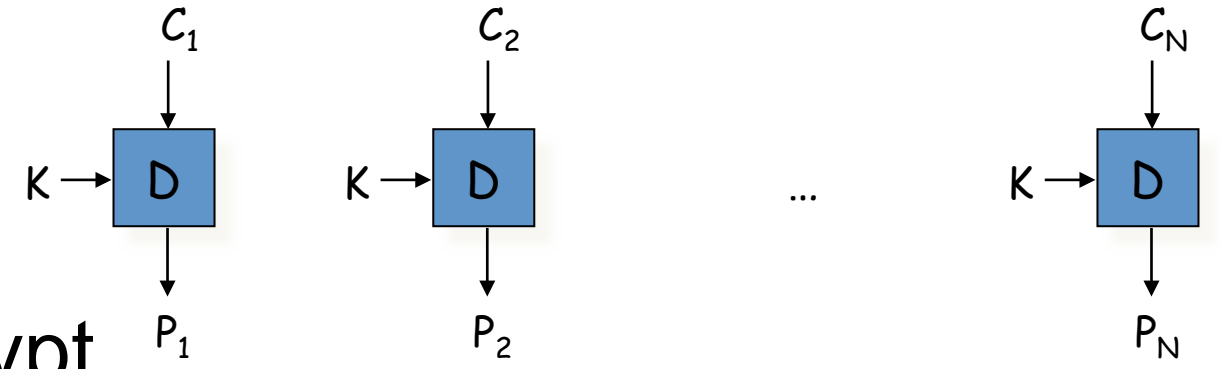
- **ECB – Electronic Codebook**
  - used to encipher a single plaintext block (e.g., a DES key)
- **CBC – Cipher Block Chaining**
  - repeated use of the encryption algorithm to encipher a message consisting of many blocks
- **CFB – Cipher Feedback**
  - used to encipher a **stream** of characters, dealing with each character as it comes
- **OFB – Output Feedback**
  - another method of **stream** encryption, used on noisy channels
- **CTR – Counter**
  - simplified OFB with certain advantages

ECB (Electronic Codebook) Mode

- Encrypt



- Decrypt



Properties of the ECB mode

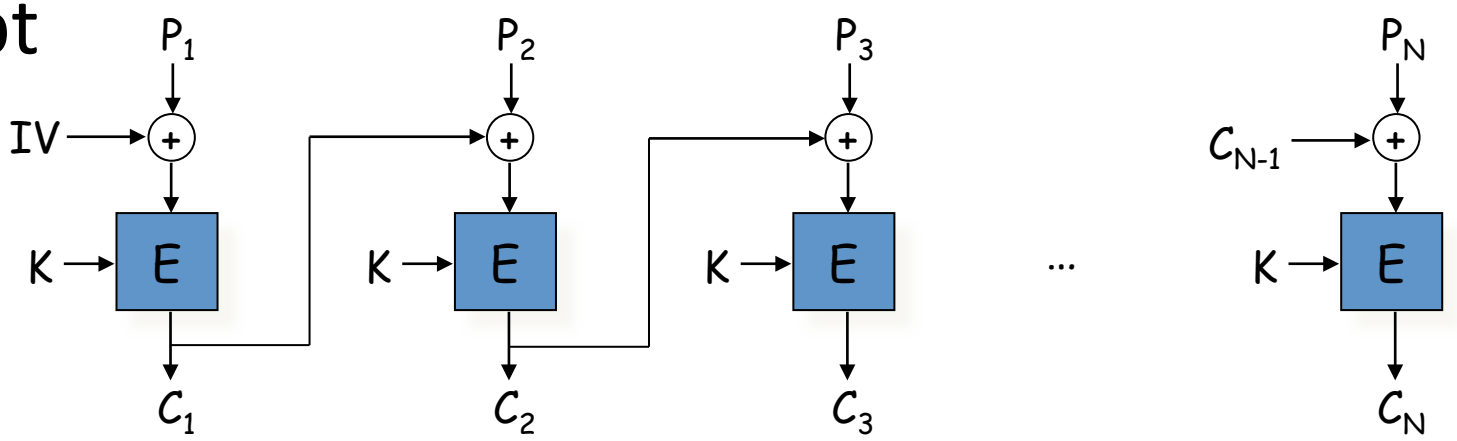
- **Identical plaintext blocks result in identical ciphertext blocks (under the same key, of course)**
  - ✓ messages to be encrypted often have very regular formats
  - ✓ repeating fragments, special headers, string of 0s, etc. are quite common
- **Blocks are encrypted independently of other blocks**
  - ✓ reordering ciphertext blocks result in correspondingly reordered plaintext blocks
  - ✓ ciphertext blocks can be cut from one message and pasted in another, possibly without detection
- **Error propagation:** one bit error in a ciphertext block affects only the corresponding plaintext block (results in garbage)
- **Overall: not recommended for messages longer than one block, or if keys are reused for more than one block**

Block Cipher Modes of Operation

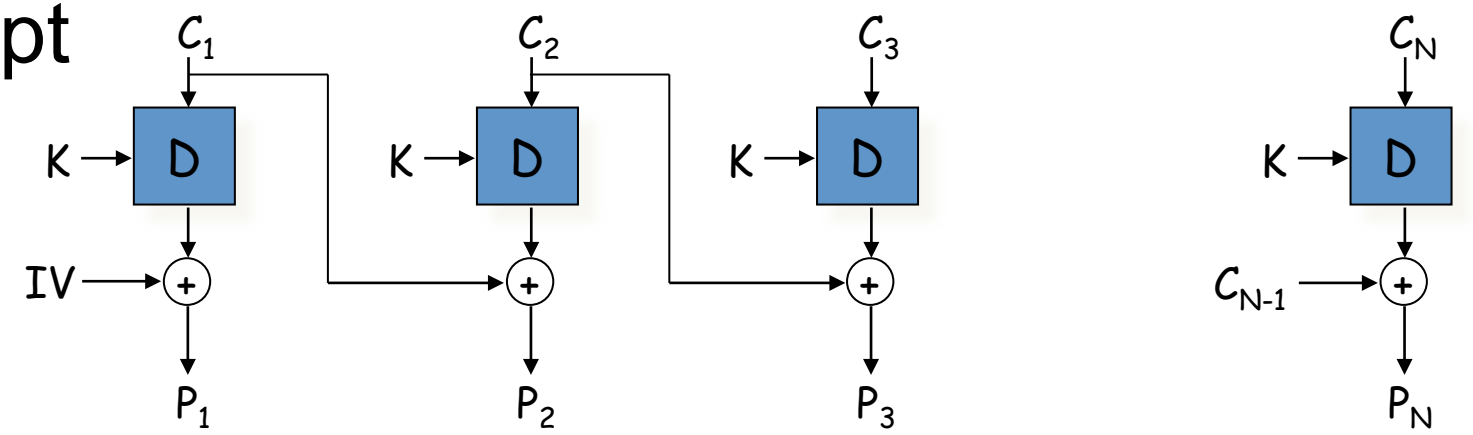
- **ECB – Electronic Codebook**
  - used to encipher a single plaintext block (e.g., a DES key)
- **CBC – Cipher Block Chaining**
  - repeated use of the encryption algorithm to encipher a message consisting of many blocks
- **CFB – Cipher Feedback**
  - used to encipher a **stream** of characters, dealing with each character as it comes
- **OFB – Output Feedback**
  - another method of **stream** encryption, used on noisy channels
- **CTR – Counter**
  - simplified OFB with certain advantages

CBC (Cipher Block Chaining) mode

encrypt



decrypt



Properties of the CBC Mode

- **Encrypting the same plaintexts** under the same key, but different IVs result in **different ciphertexts**

- Ciphertext block C_j **depends** on P_j and all preceding plaintext blocks
  - ✓ Rearranging ciphertext blocks affects decryption
  - ✓ However, dependency on the preceding plaintext blocks is only via the previous ciphertext block C_{j-1}
  - ✓ Proper decryption of a correct ciphertext block needs a correct preceding ciphertext block only

- **Error propagation:**
  - ✓ One bit error in a ciphertext block C_j has an effect on the j-th and (j+1)-st plaintext block
    - ✓ P_j' is complete garbage and P_{j+1}' has bit errors where C_j had
    - ✓ an attacker may cause predictable bit changes in the (j+1)-st plaintext block

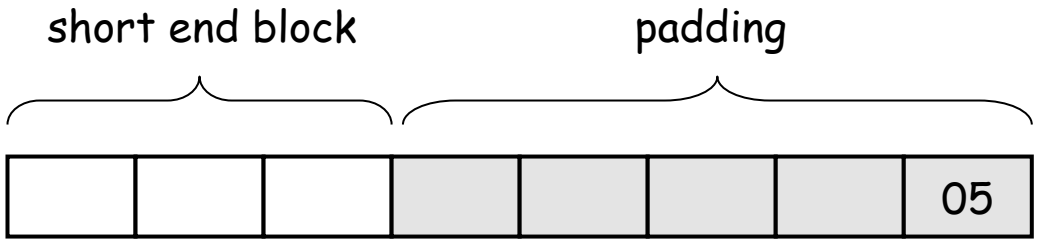
- **Error recovery:**
  - ✓ recovers from bit errors (self-synchronizing)
  - ✓ cannot, however, recover from frame errors ("lost" bits)

Integrity of the IV in the CBC mode

- The IV need not be secret, but its integrity should be protected
  - malicious modification of the IV allows an attacker to make predictable changes to the first plaintext block recovered
- One solution is to send the IV in an encrypted form at the beginning of the CBC encrypted message

Padding

- The length of the message may not be a multiple of the block size of the cipher
- One can add some extra bytes to the short end block until it reaches the correct size – this is called padding
- Usually the last byte indicates the number of padding bytes added – this allows the receiver to remove the padding



Block Cipher Modes of Operation

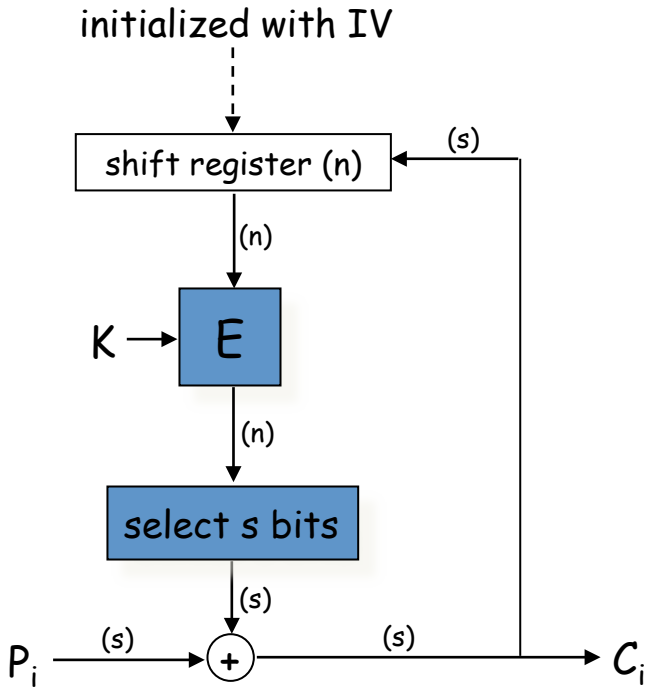
- **ECB – Electronic Codebook**
  - used to encipher a single plaintext block (e.g., a DES key)
- **CBC – Cipher Block Chaining**
  - repeated use of the encryption algorithm to encipher a message consisting of many blocks
- **CFB – Cipher Feedback**
  - used to encipher a **stream** of characters, dealing with each character as it comes
- **OFB – Output Feedback**
  - another method of **stream** encryption, used on noisy channels
- **CTR – Counter**
  - simplified OFB with certain advantages

From Block Cipher to Stream Cipher

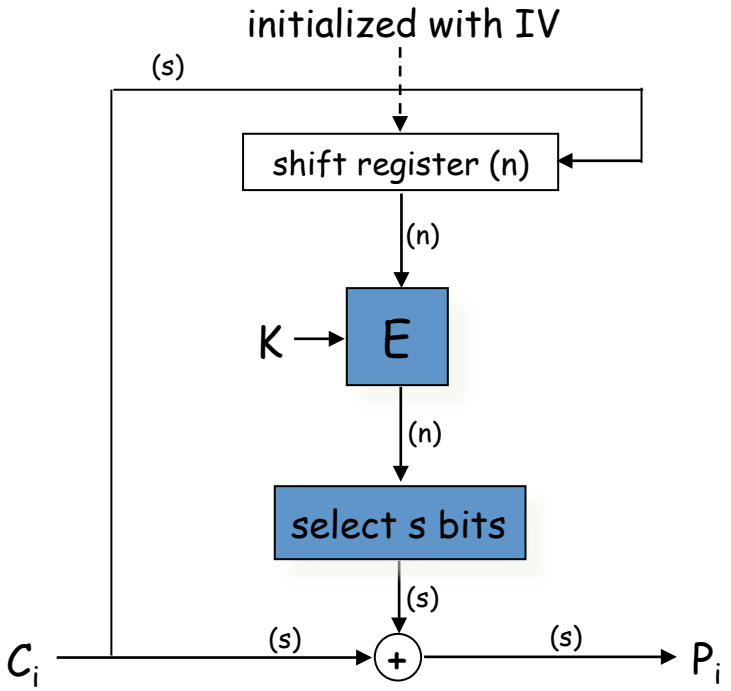
- The block cipher is used to produce a key stream, which is then XORed to the plaintext characters.
- The key stream is generated by iteratively encrypting and updating some internal state.
- The various modes differ only in the way in which this internal state is updated.

CFB (Cipher Feedback) mode

- Encrypt



- Decrypt



Properties of the CFB mode

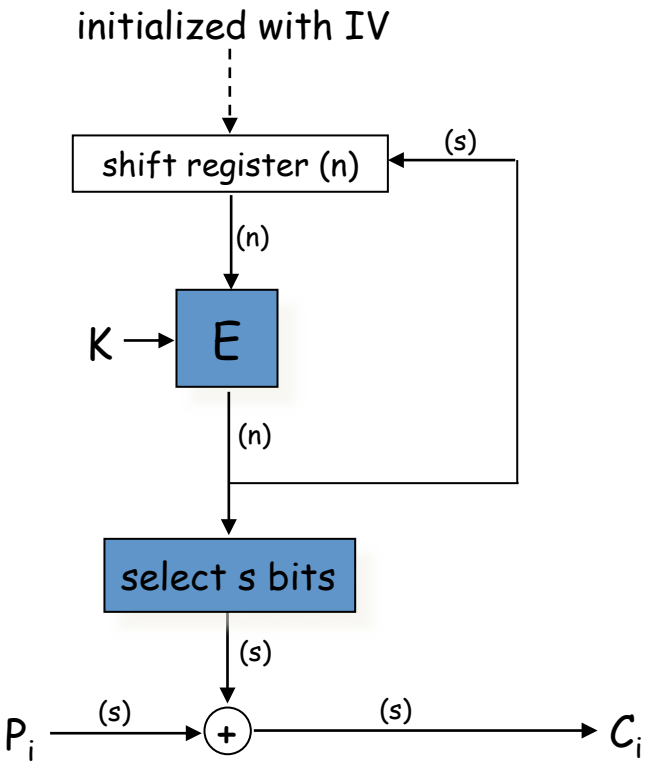
- Encrypting the same plaintexts under the same key, but **different IVs result in different ciphertexts**
- The IV can be sent in clear
- Ciphertext block C_j **depends** on P_j and all preceding plaintext blocks
  - ✓ rearranging ciphertext blocks affects decryption
  - ✓ proper decryption of a correct ciphertext block needs the preceding n/s ciphertext blocks to be correct
- **Error propagation:**
  - ✓ one bit error in a ciphertext block C_j has an effect on the decryption of that and the next n/s ciphertext blocks (the error remains in the shift register for n/s steps)
    - ✓ P_j' has bit errors where C_j had, all the other erroneous plaintext blocks are garbage
    - ✓ an attacker may cause predictable bit changes in the j-th plaintext block
- **Error recovery:**
  - ✓ self synchronizing, but requires n/s blocks to recover

Block Cipher Modes of Operation

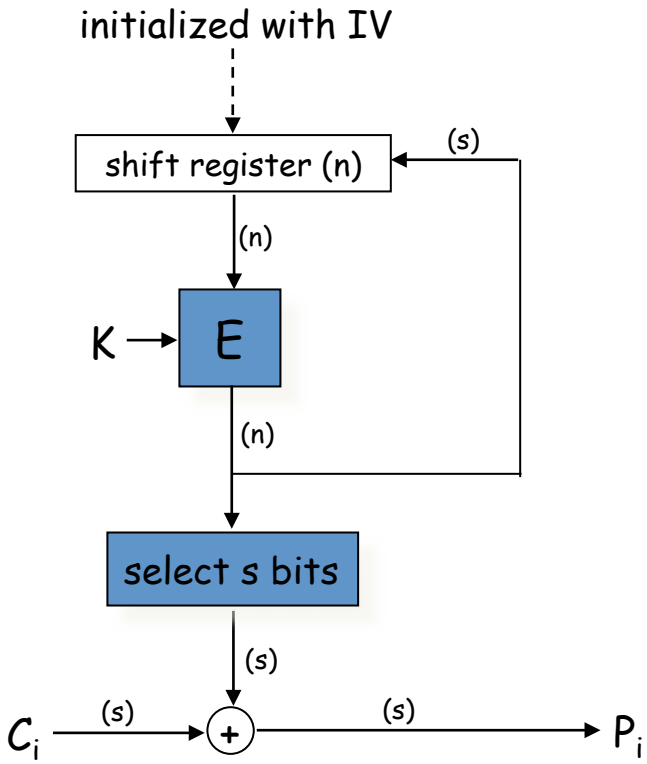
- **ECB – Electronic Codebook**
  - used to encipher a single plaintext block (e.g., a DES key)
- **CBC – Cipher Block Chaining**
  - repeated use of the encryption algorithm to encipher a message consisting of many blocks
- **CFB – Cipher Feedback**
  - used to encipher a **stream** of characters, dealing with each character as it comes
- **OFB – Output Feedback**
  - another method of **stream** encryption, used on noisy channels
- **CTR – Counter**
  - simplified OFB with certain advantages

OFB mode

- encrypt



- decrypt



Properties of the OFB mode

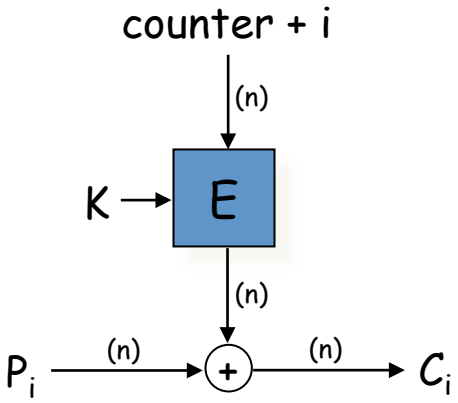
- A **different IV should be used for every new message**, otherwise messages will be encrypted with the same key stream
- **The IV can be sent in clear**
  - ✓ however, if the IV is modified by the attacker, then the cipher will never recover (unlike CFB)
- Ciphertext block C_j **depends** on P_j only (does not depend on the preceding plaintext blocks)
  - ✓ however, rearranging ciphertext blocks affects decryption
- **Error propagation:**
  - ✓ one bit error in a ciphertext block C_j has an effect on the decryption of only that ciphertext block
    - ✓ P_j' has bit errors where C_j had
    - ✓ an attacker may cause predictable bit changes in the j-th plaintext block
- **Error recovery:**
  - ✓ recovers from bit errors
  - ✓ never recovers if bits are lost or the IV is modified

Block Cipher Modes of Operation

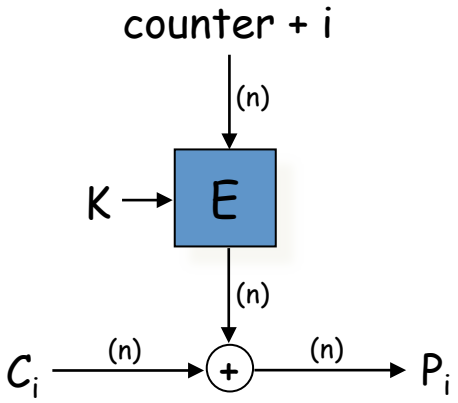
- **ECB – Electronic Codebook**
  - used to encipher a single plaintext block (e.g., a DES key)
- **CBC – Cipher Block Chaining**
  - repeated use of the encryption algorithm to encipher a message consisting of many blocks
- **CFB – Cipher Feedback**
  - used to encipher a **stream** of characters, dealing with each character as it comes
- **OFB – Output Feedback**
  - another method of **stream** encryption, used on noisy channels
- **CTR – Counter**
  - simplified OFB with certain advantages

CTR mode

- **Encrypt**



- **Decrypt**



Properties of the CTR mode

- Similar to OFB
- Cycle length depends on the size of the counter (typically 2^n)
- The i-th block can be decrypted independently of the others
  - ✓ parallelizable (unlike OFB)
  - ✓ random access
- The values to be XORed with the plaintext can be pre-computed
- At least as secure as the other modes

- **note1:** in CFB, OFB, and CTR mode only the encryption algorithm is used (decryption is not needed), that is why some ciphers (e.g., AES) is optimized for encryption
- **note2:** the OFB and CTR modes essentially make a synchronous stream cipher out of a block cipher, whereas the CFB mode converts a block cipher into a self-synchronizing stream-cipher