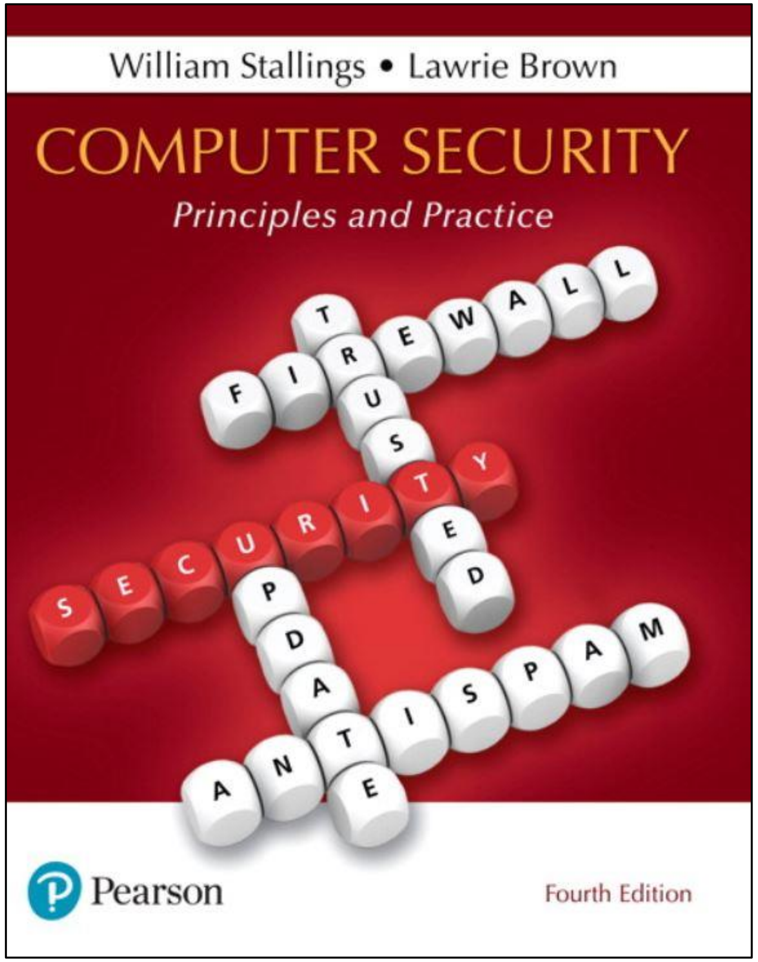
ECE 4309

Basics of cryptography – part 1

Dr. Valerio Formicola

Computer Security: Principles and Practice

Fourth Edition



Chapter 2 and 20

Cryptographic Tools

Cryptography

It's a technic that can be used in various ways to provide all CIA properties, Confidentiality, Integrity and Availability.

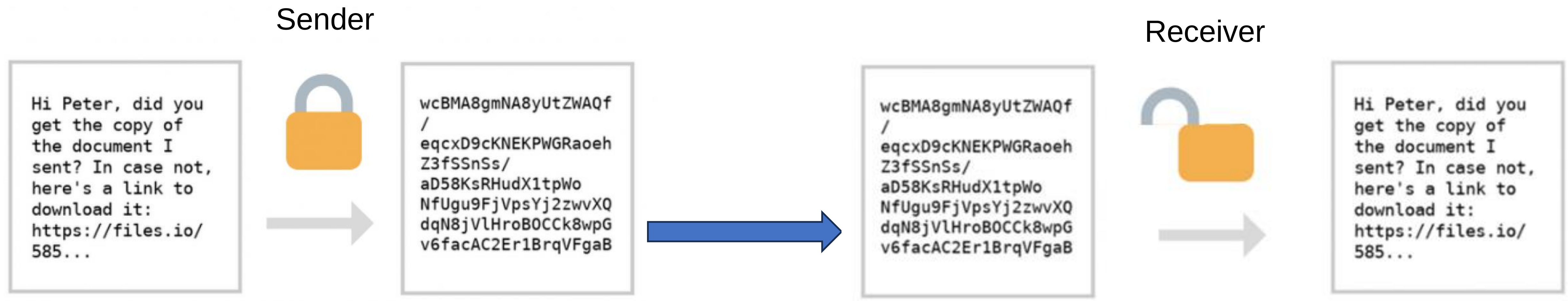
Techniques used in cryptography with different purposes:

- *Symmetric encryption*
- *Secure message hashing (hash functions)*
- *Asymmetric encryption*

Confidentiality with symmetric encryption

Confidentiality

- It can be achieved if we avoid anybody able to observe a message, is still not able to understand the content of the message
  - e.g., the attacker might captur it on a transmission medium, like cable or air (type of attack called as *eavesdropping* attack, a type of passive attack)



Hi Peter, did you get the copy of the document I sent? In case not, here's a link to download it: https://files.io/585...

```
wcBMA8gmNA8yUtZWAQf
/
eqcxD9cKNEKPWGRaoeh
Z3fSSnSs/
aD58KsRHudX1tpWo
NfUgu9FjVpsYj2zwvXQ
dqN8jVlHroBOCCk8wpG
v6facAC2Er1BrqVFgaB
```

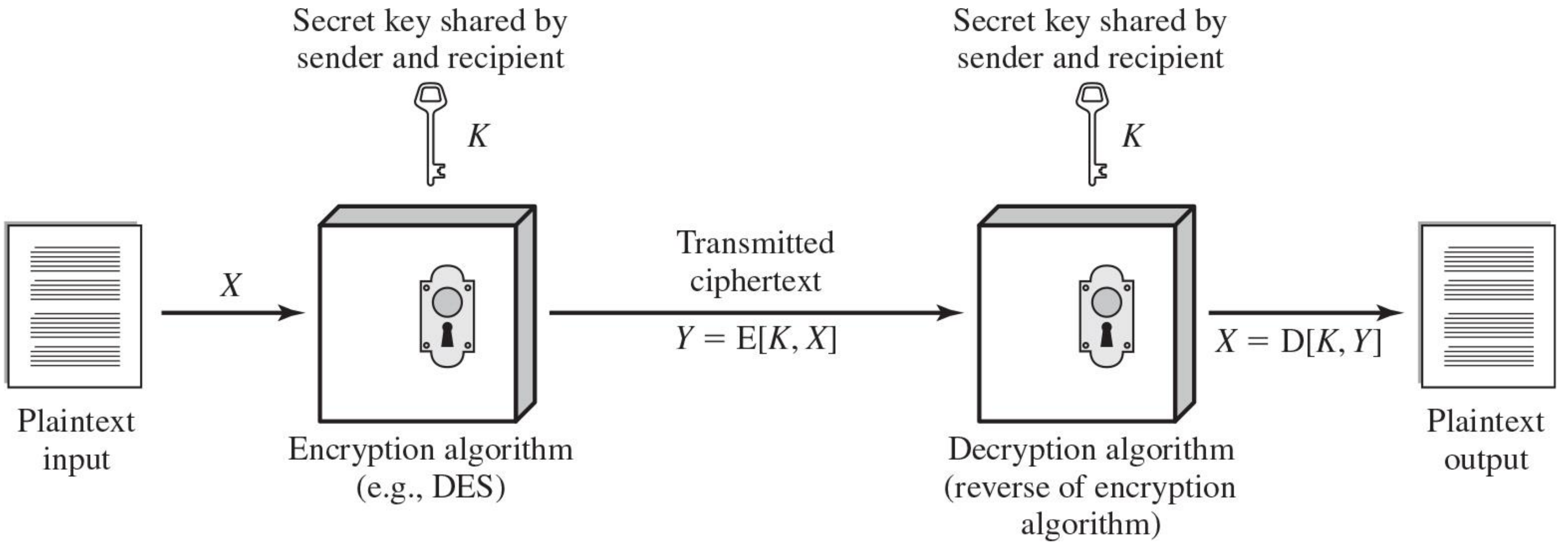
```
wcBMA8gmNA8yUtZWAQf
/
eqcxD9cKNEKPWGRaoeh
Z3fSSnSs/
aD58KsRHudX1tpWo
NfUgu9FjVpsYj2zwvXQ
dqN8jVlHroBOCCk8wpG
v6facAC2Er1BrqVFgaB
```

Hi Peter, did you get the copy of the document I sent? In case not, here's a link to download it: https://files.io/585...

Symmetric Encryption

- Also referred to as:
  - Conventional encryption
  - Secret-key or single-key encryption
- Only alternative before public-key encryption in 1970’s
  - Still most widely used alternative
- Has five ingredients:
  - Plaintext
  - Encryption algorithm
  - Secret key
  - Ciphertext
  - Decryption algorithm

Figure 2.1 Simplified Model of Symmetric Encryption


Secret key shared by sender and recipient
K
Secret key shared by sender and recipient
K
X
Transmitted ciphertext
$Y = E[K, X]$
$X = D[K, Y]$
Plaintext input
Encryption algorithm (e.g., DES)
Decryption algorithm (reverse of encryption algorithm)
Plaintext output

Note: a plaintext message is not necessarily text. It is anything that you can use with proper decoding; for example, a text with letters or a multimedia forma, or simply commands or instructions

Symmetric Encryption

- The universal technique for providing confidentiality for transmitted or stored data
- Also referred to as conventional encryption or *single-key encryption*
- Two requirements for secure use:
  - Need a strong encryption algorithm
  - Sender and receiver must have obtained copies of the secret key in a secure fashion and must keep the key secure

What can be attacked in such a schema?

- The attacker and anybody else usually knows how the encryption/decryption algorithm works, so that’s not a secret
  - Algorithms are mostly standard and public

- The attacker might very well intercept and study the encrypted messages
  - that’s why we need a mechanism that it doesn’t allow to reverse the process

Attacking Symmetric Encryption

Cryptanalytic Attacks

- Rely on:
  - Mathematical “nature” of the algorithm
  - Some knowledge of the general characteristics of the plaintext
    - e.g., all emails start with Hello X …
  - Ideal for the crypto-analyst: Some sample plaintext-ciphertext pairs (aka, *cleartext* analysis)
- Exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or the key being used
  - If successful, all future and past messages encrypted with that key are compromised

Brute-Force Attacks

- Try all possible keys on some ciphertext until an intelligible translation into plaintext is obtained
  - On average half of all possible keys must be tried to achieve success
  - An attacking tool shows results of decryption with a wrong key

A cryptographic algorithm is said to be **breakable** if a crypto-analyst can systematically recover the original message without knowing the key

- *Example of attack in cryptoanalysis*: **frequency analysis** (also known as **counting letters**) is the study of the frequency of letters or groups of letters in a ciphertext.
- *Example of attack in brute force attacks*: **Dictionary attack** to use names and common words with small substitutions; **credential stuffing** to use databases of users and passwords on different accounts.

Table 2.1 Comparison of Three Popular Symmetric Encryption Algorithms

Block ciphers: The most commonly used symmetric encryption algorithms.
A block cipher processes the plaintext input in fixed-size blocks and produces a block of ciphertext of equal size for each plaintext block

	DES	Triple DES	AES
Plaintext block size (bits)	64	64	128
Ciphertext block size (bits)	64	64	128
Key size (bits)	56	112 or 168	128, 192, or 256

DES = Data Encryption Standard

AES = Advanced Encryption Standard

Data Encryption Standard (DES)

- Until recently was the most widely used encryption scheme
  - FIPS PUB 46
  - Referred to as the Data Encryption Algorithm (DEA)
  - Uses 64 bit plaintext block and 56 bit key to produce a 64 bit ciphertext block

- Strength concerns:
  - Concerns about the algorithm itself
    - DES is the most studied encryption algorithm in existence.
      - Quite stable in practice but very old (i.e., very studied)
  - Concerns about the use of a 56-bit key
    - The speed of commercial off-the-shelf processors makes this key length woefully inadequate

Table 2.2 Average Time Required for Exhaustive Key Search

Key Size (bits)	Cipher	Number of Alternative Keys	Time Required at 10 to the ninth power decryptions per microseconds 10^9 decryptions/s	Time Required at 10 to the 13 power decryptions per microseconds 10^{13} decryptions/s
56	DES	$2^{56} \approx 7.2 \times 10^{16}$	$2^{55}\ \mu s = 1.125$ years	1 hour
128	AES	$2^{128} \approx 3.4 \times 10^{38}$	$2^{127}\ \mu s = 5.3 \times 10^{21}$ years	5.3×10^{17} years
168	Triple DES	$2^{168} \approx 3.7 \times 10^{50}$	$2^{167}\ \mu s = 5.8 \times 10^{33}$ years	5.8×10^{29} years
192	AES	$2^{192} \approx 6.3 \times 10^{57}$	$2^{191}\ \mu s = 9.8 \times 10^{40}$ years	9.8×10^{36} years
256	AES	$2^{256} \approx 1.2 \times 10^{77}$	$2^{255}\ \mu s = 1.8 \times 10^{60}$ years	1.8×10^{56} years

- Note: book as some error. The number of decryptions are per seconds (not per micro-seconds)

Triple DES (3DES)

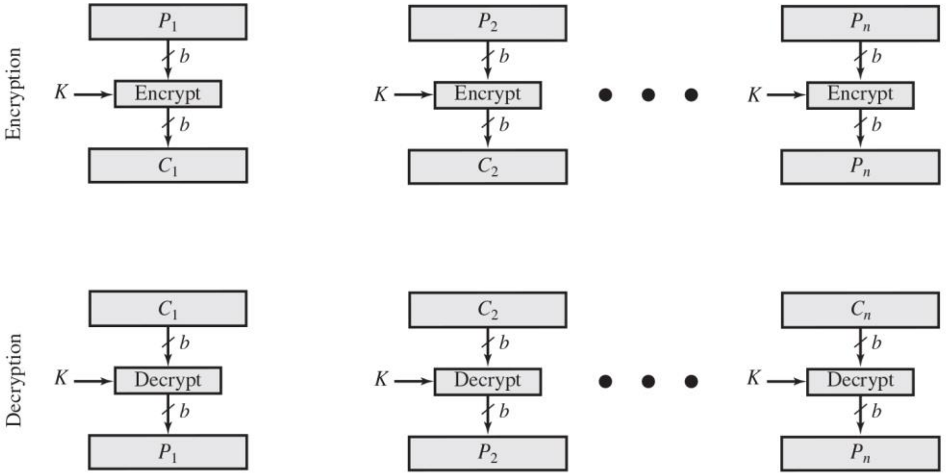
- Repeats basic DES algorithm three times using either two or three unique keys
- First standardized for use in financial applications in ANSI standard X9.17 in 1985
- Attractions:
  - 168-bit key length overcomes the vulnerability to brute-force attack of DES
  - Underlying encryption algorithm is the same as in DES
- Drawbacks:
  - Algorithm is sluggish in software (slow when implemented)
  - Uses a 64-bit block size
    - Too many fragmentations for larger data

Advanced Encryption Standard (AES)

- Needed a replacement for 3DES
  - 3DES was not reasonable for long term use
- NIST called for proposals for a new AES in 1997
  - Should have a security strength equal to or better than 3DES
  - Significantly improved efficiency
  - Symmetric block cipher
  - 128 bit data and 128/192/256 bit keys
- Selected Rijndael in November 2001
  - Published as FIPS 197
- AES is now widely available in commercial products.

Figure 2.2 Types of Symmetric Encryption

Block Cipher
(in ECB mode)

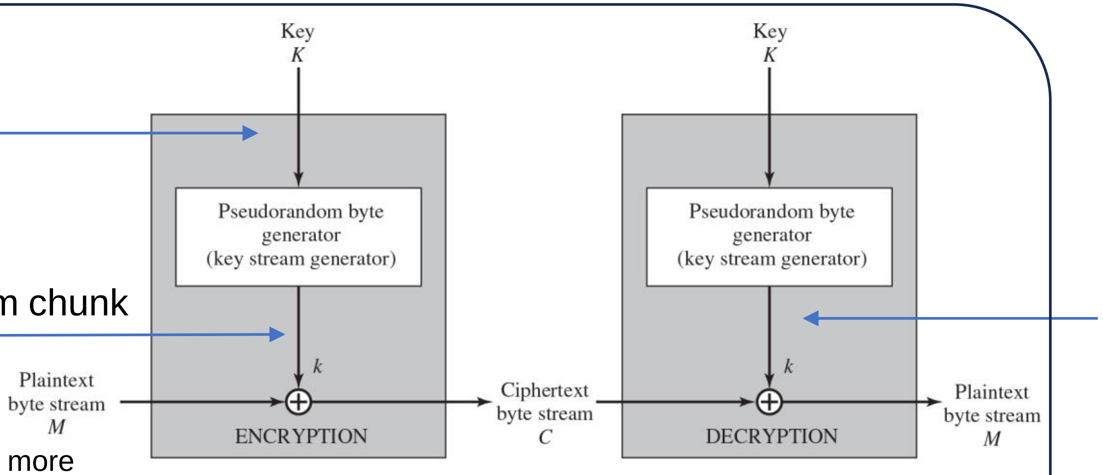


(a) Block cipher encryption (electronic codebook mode)

Stream Cipher

Always the same
key k

Always different
keystream generated for each stream chunk


Key
K
Pseudorandom byte generator (key stream generator)
k
Plaintext byte stream M
ENCRYPTION
Ciphertext byte stream C
DECRYPTION
Plaintext byte stream M

M can be 1 byte or 1 bit or more

(b) Stream encryption

In Stream cipher
sender and receiver have
the same key.
How do they have
the same keystream?

The principle of stream ciphers is that the sender and the receiver agree on an ***algorithm***,
a **secret key** and ***some parameters***, and both calculate the *keystream* from those parameters.

The ***parameters*** can be something like a way to derive a session key from a shared master key or from a key exchange, an IV (a unique value sent at the beginning of each message that allows using the same key for multiple messages).

Practical Security Issues of Block ciphers

- Typically, symmetric encryption is applied to a unit of data larger than a single 64-bit or 128-bit block
- **Electronic codebook (ECB)** mode is the simplest approach to multiple-block encryption
  - Each block (of same size) of plaintext is encrypted using the same key
  - Cryptanalysts may be able to exploit regularities in the plaintext
    - E.g., the beginning of a message might follow some patterns like a headline in an email, etc.
- Modes of operation
  - Alternative techniques developed to increase the security of symmetric block encryption for large sequences:
    - Cipher Block Chaining (CBC)
    - Cipher Feedback Mode (CFB)
    - Output Feedback Mode (OFB)
    - Counter Mode (CM)
  - Overcomes the weaknesses of ECB

Advantages of using ECB –

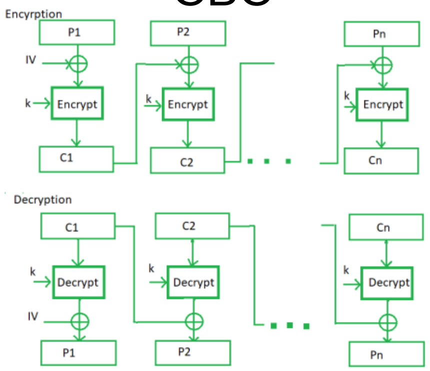
•Faster way of encryption in parallel mode.

•Simple way of the block cipher.

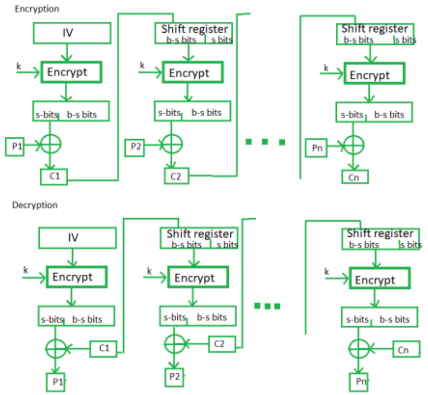
Disadvantages of using ECB –

•Prone to cryptanalysis since there is a direct relationship between plaintext and ciphertext.

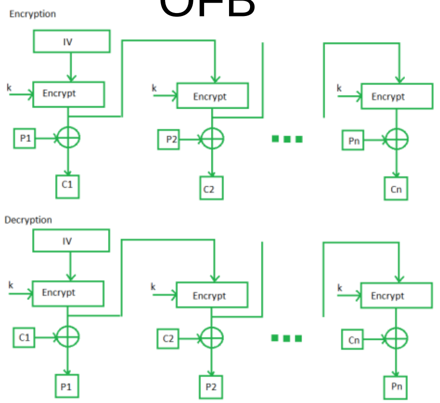
CBC



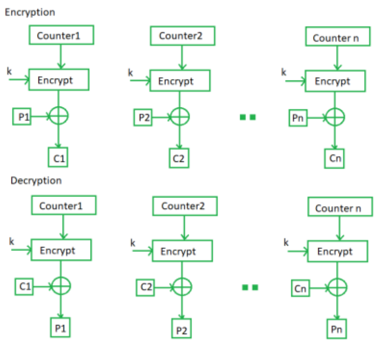
CFB



OFB



CM



Applications for Block ciphers

1. **Data Encryption (independently of support media):** Block Ciphers are widely used for the encryption of private and sensitive data such as passwords, credit card details and other information that is transmitted or stored for a communication. This encryption process converts a plain data into non-readable and complex form. Encrypted data can be decrypted only by the authorised person with the private keys.
2. **File and Disk Encryption (encryption of support media, as hard drives):** Block Ciphers are used for encryption of entire files and disks in order to protect their contents and restrict from unauthorised users. The disk encryption softwares such as BitLocker, TrueCrypt aslo uses block cipher to encrypt data and make it secure.
3. **Virtual Private Networks (VPN):** Virtual Private Networks (VPN) use block cipher for the encryption of data that is being transmitted between the two communicating devices over the internet. This process makes sure that data is not accessed by unauthorised person when it is being transmitted to another user.
4. **Secure Sockets Layer (SSL) and Transport Layer Security (TLS):** SSL and TLS protocols use block ciphers for encryption of data that is transmitted between web browsers and servers over the internet. This encryption process provides security to confidential data such as login credentials, card information etc. (initially, stream ciphers were used, but now they have been replaced)
5. **Digital Signatures:** Block ciphers are used in the digital signature algorithms, to provide authenticity and integrity to the digital documents. This encryption process generates the unique signature for each document that is used for verifying the authenticity and detecting if any malicious activity is detected.

Block & Stream Ciphers

- Block Cipher
  - Processes the input one block of elements at a time
  - Produces an output block for each input block
  - Can reuse keys
  - More common

- Stream Cipher
  - Processes the input elements continuously
  - Produces output one element at a time
  - Primary advantage is that they are almost always faster and use far less code
  - Encrypts plaintext one byte at a time
  - Pseudorandom stream is one that is unpredictable without knowledge of the input key and/or initialization parameters

Applications for Stream ciphers

- Stream ciphers are often used for their speed and simplicity of implementation in hardware, and in applications where plaintext comes in quantities of unknowable length like a secure wireless connection (WEP, WPA) or wired (TLS/SSL, deprecated). If a block cipher (not operating in a stream cipher mode) were to be used in this type of application, the designer would need to choose either transmission efficiency or implementation complexity, since block ciphers cannot directly work on blocks shorter than their block size.
- Mostly used for Real Time applications, e.g., media stream cryptography in DVD/BD players, etc.

More details about algorithms in symmetric key

Cryptography in symmetric keys algorithms

- Classified along three independent dimensions:
  - The type of operations used for transforming plaintext to ciphertext
    - ***Substitution*** – each element in the plaintext is mapped into another element
    - ***Transposition*** – elements in plaintext are rearranged
  - The number of keys used
    - Sender and receiver use same key – symmetric
    - Sender and receiver each use a different key - asymmetric
  - The way in which the plaintext is processed
    - Block cipher – processes input one block of elements at a time
    - Stream cipher – processes the input elements continuously

Table 20.1 Types of Attacks on Encrypted Messages

Difficulty level

Type of Attack	**Known to Cryptanalyst**
Ciphertext only (unrealistic under attack)	• Encryption algorithm • Ciphertext to be decoded
Known plaintext (most common target of defense)	• Encryption algorithm • Ciphertext to be decoded • One or more plaintext–ciphertext pairs formed with the secret key (includes when the plaintext is a partial message that follows a pattern, e.g., a common text written in English during communications in headers or banners)
Chosen plaintext (less common, but possible)	• Encryption algorithm • Ciphertext to be decoded • Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key • (the analyst may deliberately pick patterns that can be expected to reveal the structure of the key, e.g., a hacker writes in the commented source code header some information in a software company)
Chosen ciphertext (rare)	• Encryption algorithm • Ciphertext to be decoded • Purported ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key (E.g., analyst submit encrypted queries (not by him/her but with the unkown key) and retrieves the answer in plaintext, aka “lunchtime attacks”.)
Chosen text	• Encryption algorithm • Ciphertext to be decoded • Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key • Purported ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key

Computationally Secure Encryption Schemes

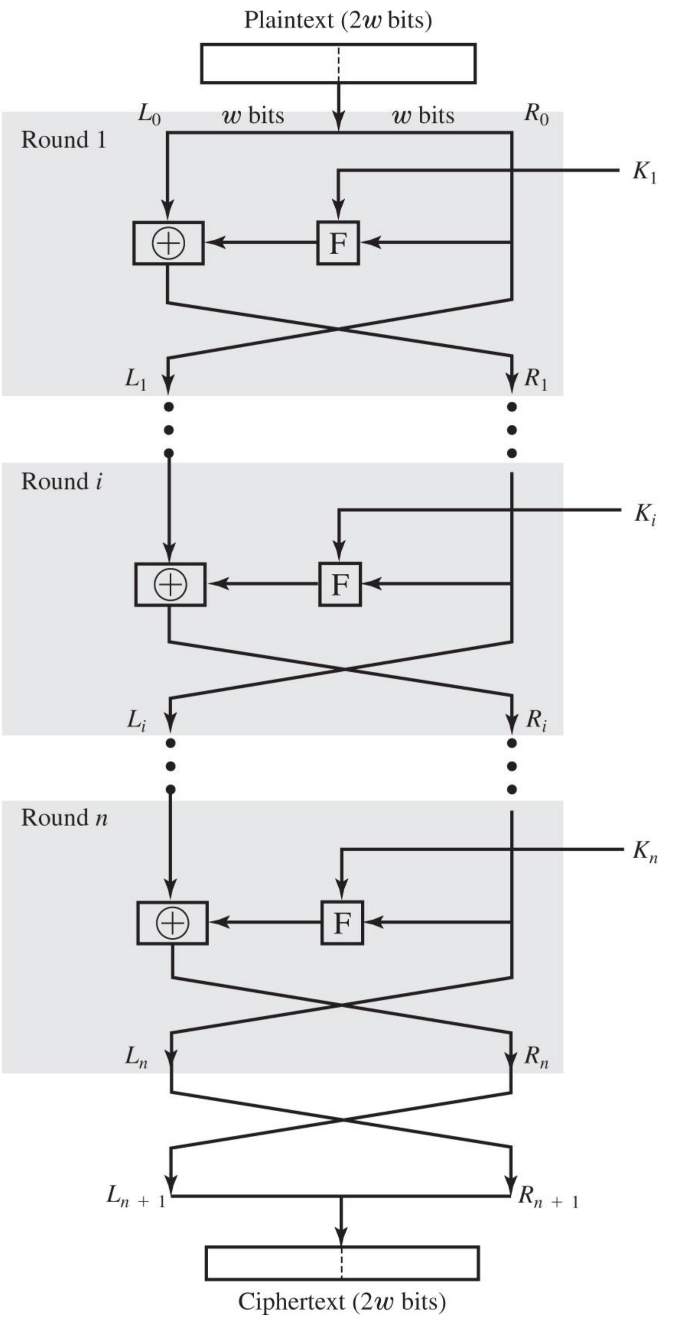
- Encryption is **computationally secure** if:
  - Cost of breaking cipher exceeds value of information
  - Time required to break cipher exceeds the useful lifetime of the information
    - E.g., time to break a key is 1 hour. Time to change the key is 10 seconds
    - E.g., even worst, the information encrypted is not useful after 1 hour
- Usually very difficult to estimate the amount of effort required to break
  - **Can estimate time/cost in a brute-force attack**
    - **Time to try half of the keys possible**

Figure 20.1 Classical Feistel Network

Common schema for many algorithms, most notably DES

Encryption in DES:
2w = 64 bits (1 word w is 4 bytes)
Primary K size = 56 bit
# Rounds = 16
# Subkeys = 16
# F (round functions) = 1 (same for each round)

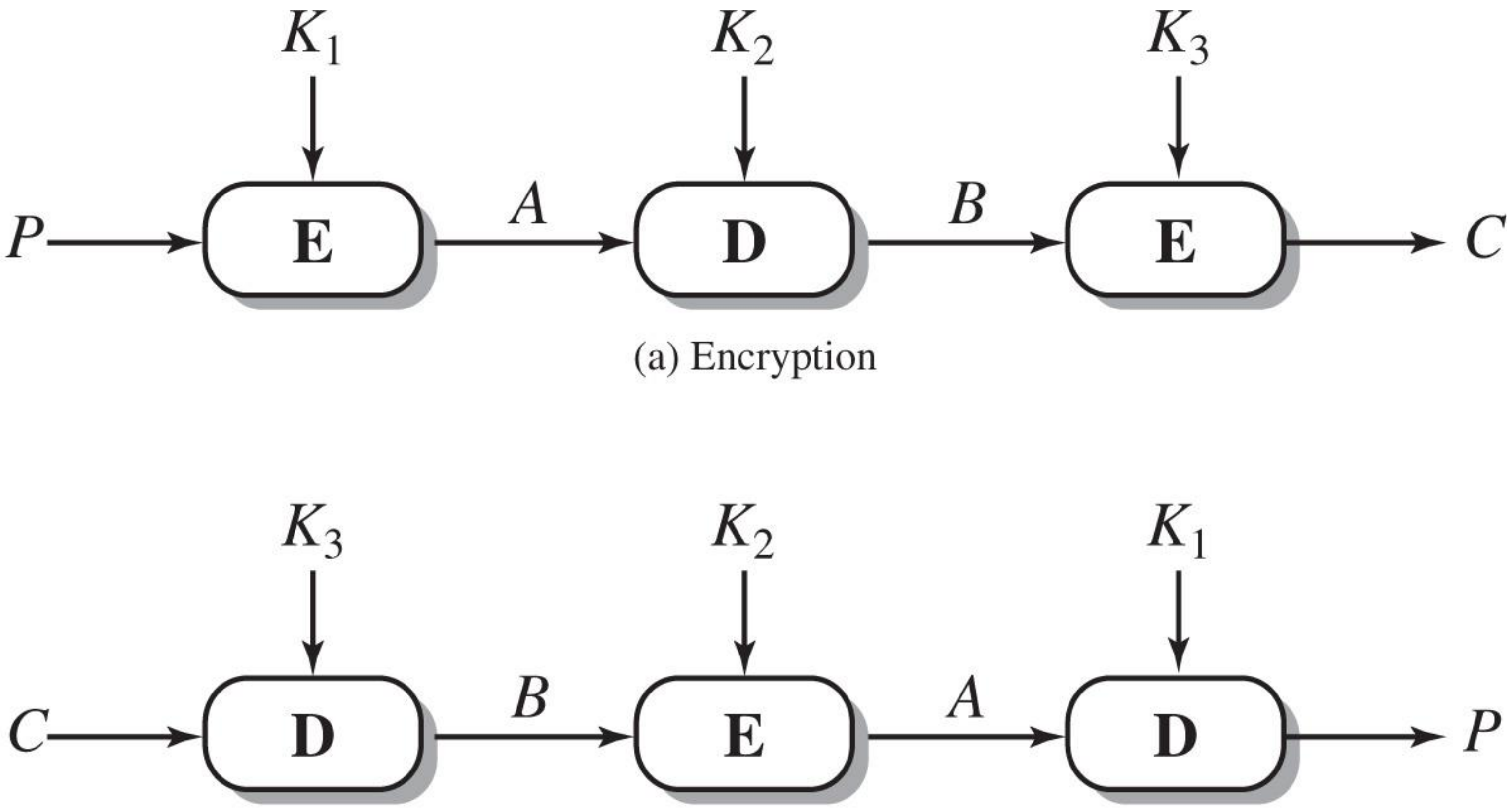
Decryption in DES:
Subkeys as before but in reverse order
Round as before but in reverse order



Block Cipher Structure

- Symmetric block cipher consists of:
  - A sequence of rounds
  - With substitutions and permutations controlled by key
- Parameters and design features:
  - Block size: larger is better, 128 bits usually
  - Key size: larger better but slower
  - Number of rounds: more is better, usually 16 rounds
  - Subkey generation algorithm: more complex is better
  - Round function: more complex is better
  - Fast software encryption/decryption: software versions are cheaper but slower
  - Ease of analysis: tradeoff more analysis possible for clearness vs more difficult for strength

Figure 20.2 Triple DES



(a) Encryption

(b) Decryption

P = Plaintext
C = Ciphertext
A, B = intermediate ciphertexts

Total key size is 56 * 3 = 168 bit

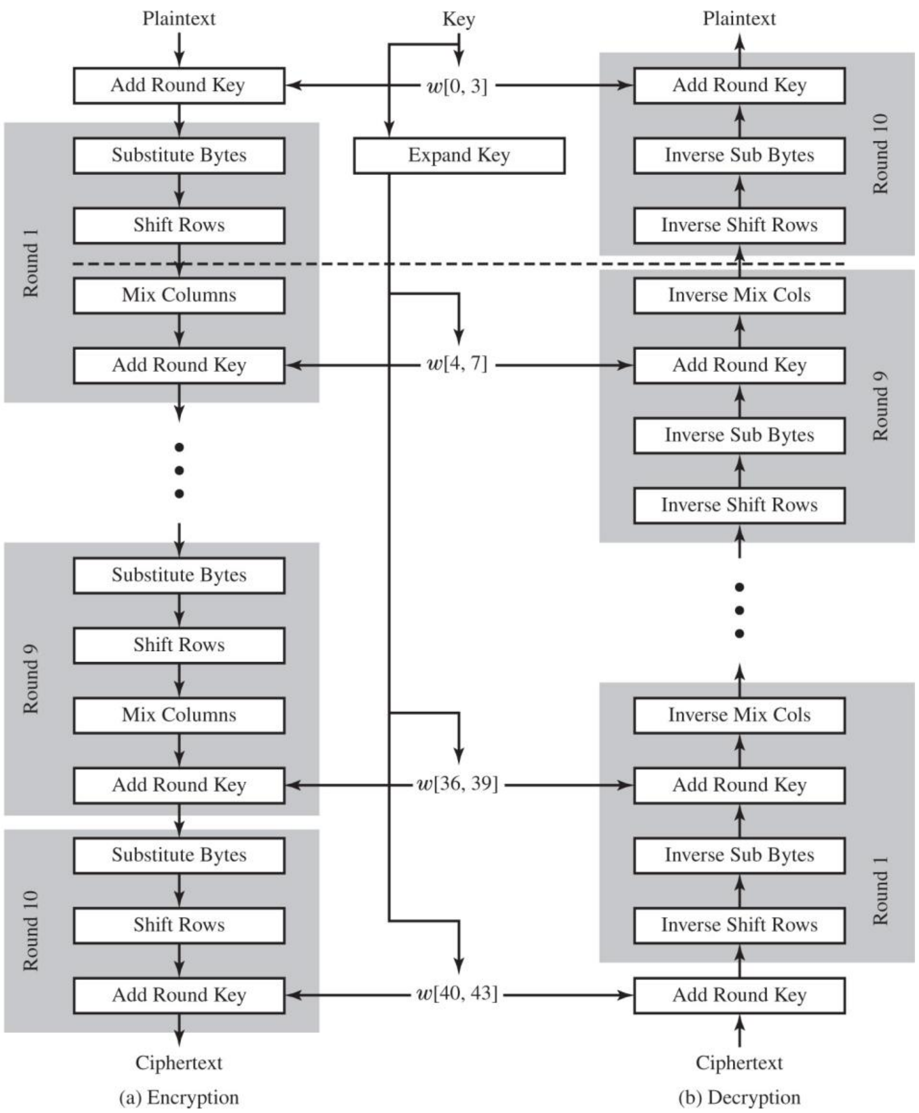
Figure 20.3 AES Encryption and Decryption

w = word (4 bytes)

Block length = 128 bits

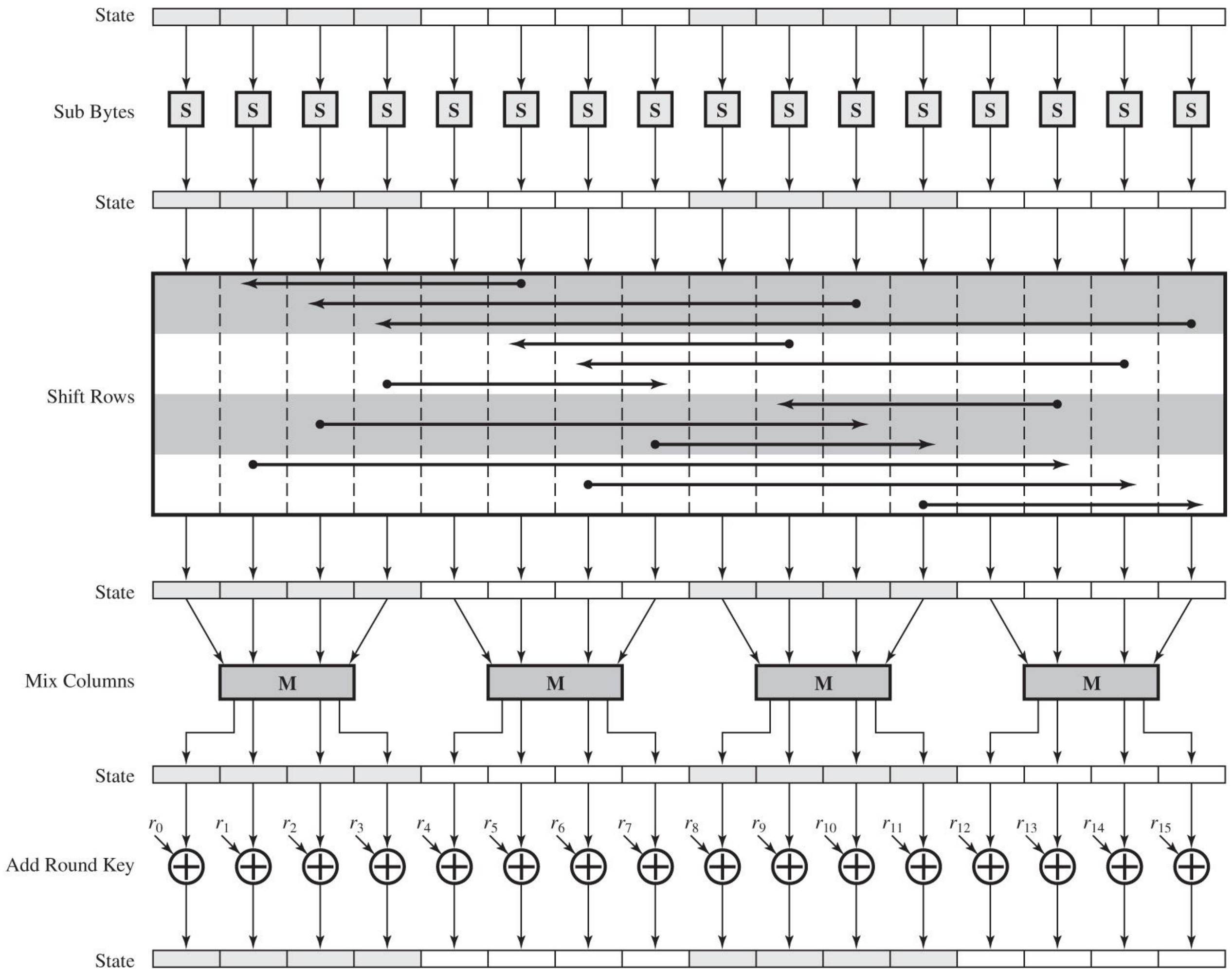
key length = 128, 192, 256

Initial Key becomes 44 expanded keys of
4 bytes (32 bits)



Block of bytes is arranged as a square matrix ordered by columns,
128 bits are 1 column copied into a State array

Figure 20.4 AES Encryption Round


State
Sub Bytes
S
State
Shift Rows
State
Mix Columns
M
State
r_0 r_1 r_2 r_3 r_4 r_5 r_6 r_7 r_8 r_9 r_{10} r_{11} r_{12} r_{13} r_{14} r_{15}
Add Round Key
State

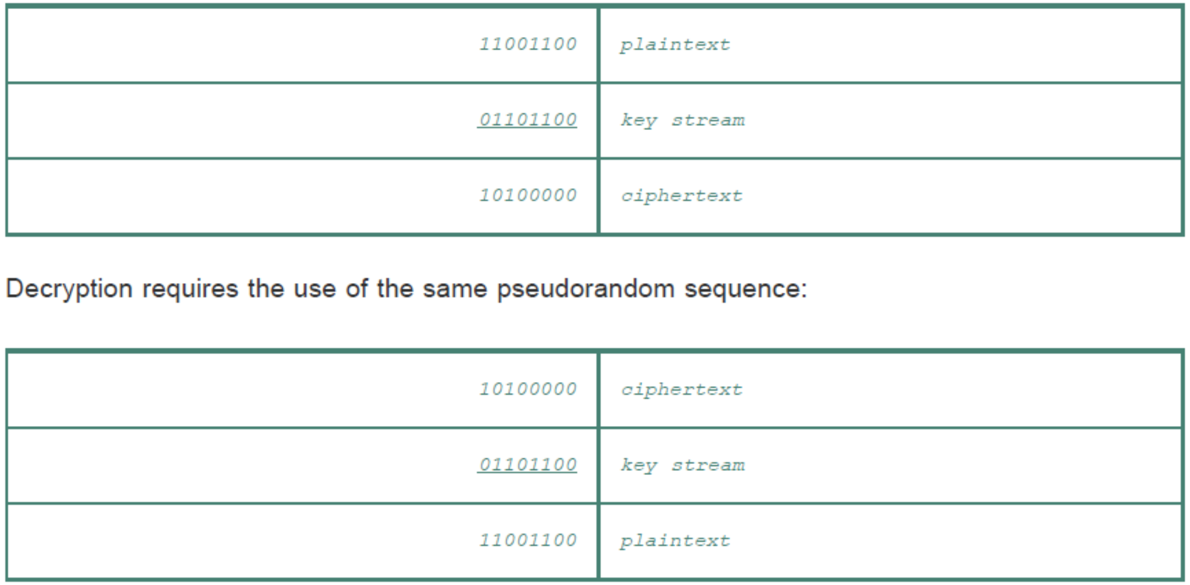
Stream Ciphers

- Processes input elements continuously
- Key input to a pseudorandom bit generator
  - Produces stream of random like numbers
  - Unpredictable without knowing input key
  - XOR keystream output with plaintext bytes

Stream Cipher operations

A typical stream cipher encrypts plaintext 1 byte at a time, although a stream cipher may be designed to operate on 1 bit at a time or on units larger than a byte at a time. In this structure, a key is input to a pseudorandom bit generator that produces a stream of 8-bit numbers that are apparently random. A pseudorandom stream is one that is unpredictable without knowledge of the input key and that has an apparently random character. The output of the generator, called a **keystream**, is combined 1 byte at a time with the plaintext stream using the bitwise exclusive-OR (XOR) operation.

For example, if the next byte generated by the generator is 01101100 and the next plaintext byte is 11001100, then the resulting ciphertext byte is:



11001100	plaintext
01101100	key stream
10100000	ciphertext

Decryption requires the use of the same pseudorandom sequence:

10100000	ciphertext
01101100	key stream
11001100	plaintext

Security of stream ciphers

- The advantage of a block cipher is that you can reuse keys.
- However, if two plaintexts are encrypted with the same key using a stream cipher, then cryptanalysis is often quite simple [DAWS96]. If the two ciphertext streams are XORed together, the result is the XOR of the original plaintexts. If the plaintexts are text strings, credit card numbers, or other byte streams with known properties, then cryptanalysis may be successful.

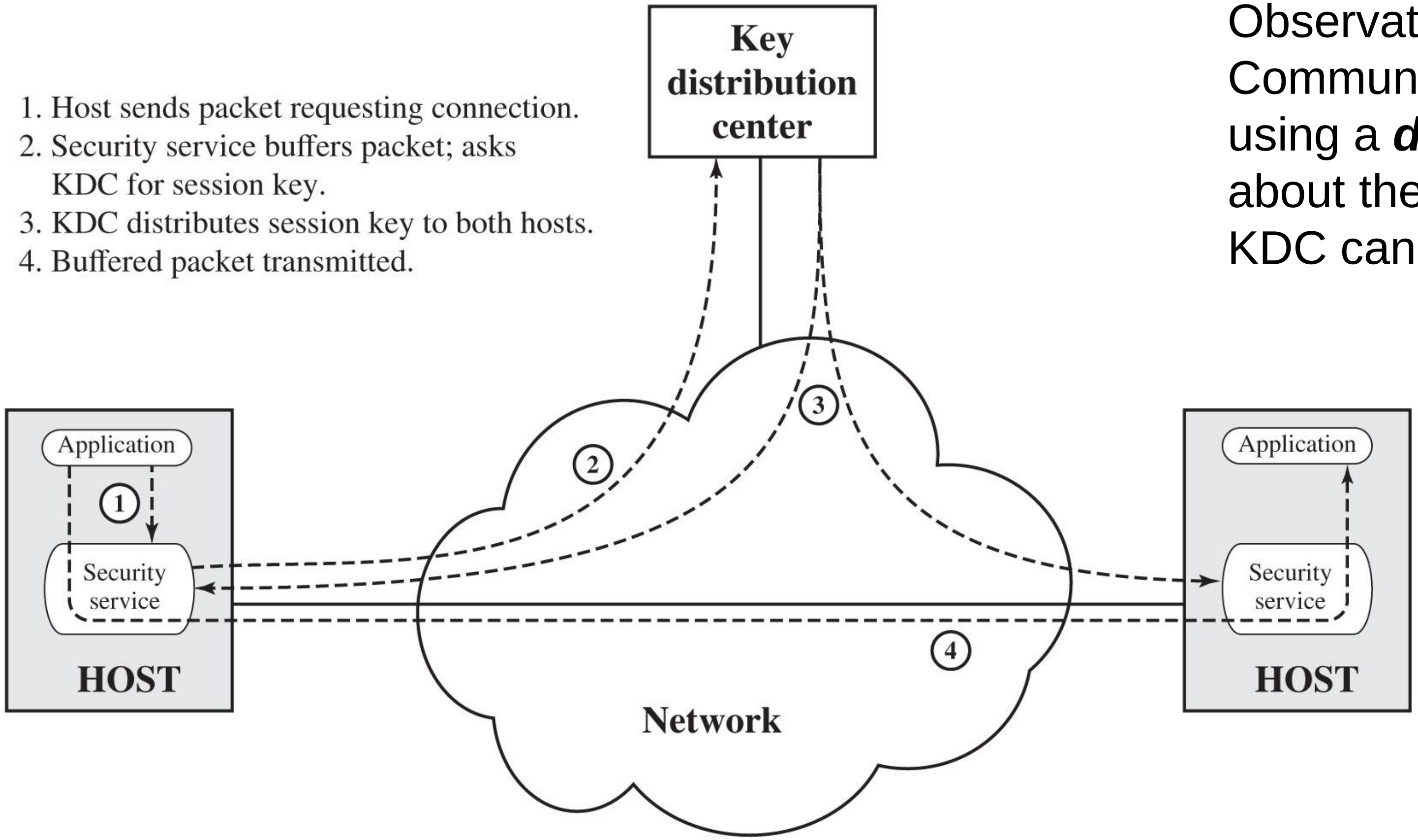
Most common standards

- RC4 (proprietary but leaked):
  - Variable key size
  - 1 byte stream
  - Ver simple software implementation, so very fast (8 to 16 instructions)
  - Used in SSL/TLS and in WEP, WPA protocols of Wi-Fi
- ChaCha20:
  - Future replacement for RC4

Schema 1 for Automatic Key Distribution for Connection-Oriented Protocol

Objective: obtain a ***session key*** for both hosts from a **third party** (KDC)

Observation:
Communication between Hosts and KDC is encrypted using a ***derivation key*** known to the two HOSTs about the third party KDC.
KDC can accept requests to generate ***session keys***



1. Host sends packet requesting connection.
2. Security service buffers packet; asks KDC for session key.
3. KDC distributes session key to both hosts.
4. Buffered packet transmitted.

Schema 2: Diffie-Hellman

- It is an asymmetric algorithm
- We’ll see soon…