Security in Distributed Systems

Overview

Slides Prepared by: Hadachi&Lind

Some of These slides are inspired from the books:

Hwang, Fox, and Dongarra, Distributed and Cloud Computing : from Parallel Processing to the Internet of Things, Morgan Kaufmann, October 2011.

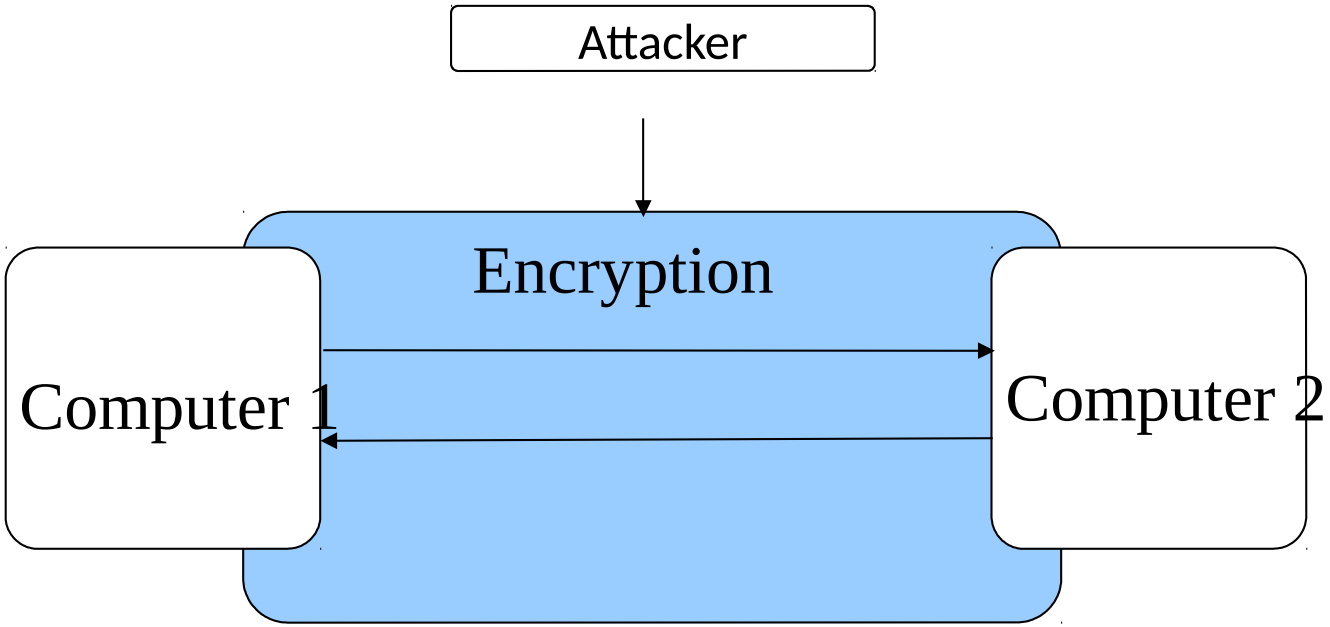
George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair, Distributed Systems: Concepts and Design (5th Edition), Addison-Wesley 2012

Outline

- Security Threats
- Security Mechanisms
- Cryptography
  - Symmetric cryptography
    - Example: DES
  - Asymmetric cryptography
    - Example: RSA
- SSL protocol

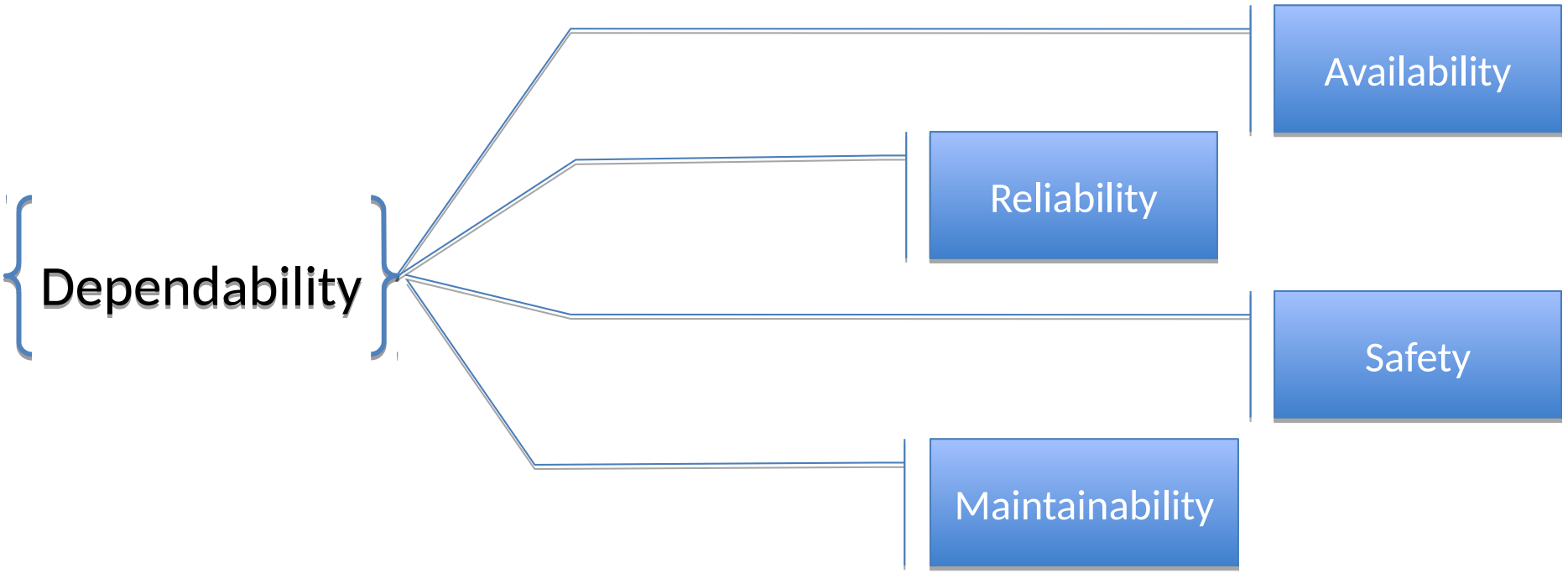
Distributed System Security: Goals

What was the definition of a DS ?
What was the definition of a protocol ?
What will happen if protocol is known to a third party ?
How to protect DS against it ?



Security Threats

- Dependability
  - "A dependable computer system is one that we justifiably trust to deliver its services" [Laprie, 1995]



Security Threats

- Confidentiality
  - Refers to the property of a computer system whereby its information is disclosed only to authorized parties.
- Integrity
  - Refers to the characteristic that alterations to a system's assets can be made only in an authorized manner.

Security Threats

- Interception
  - Access by unauthorized users
- Interruption
  - Service or data becomes unavailable
- Modification
  - Unauthorized tampering of data or service
- Fabrication
  - Additional data or info is fabricated

Type of attack

Passive attacks

- Browsing
- Inferencing
- Masquerading

Active attacks

- Virus
- Worm
- Logic bomb
- Integrity attack
- …

Security policy

- Definition:
  - SP defines precisely which actions the entities in a systems are allowed to take and which ones are prohibited.

Security Mechanisms

- Encryption
  - Transform the data into something an attacker cannot understand
- Authentication
  - Used to verify the claimed identity of a user.
- Authorization
  - Checking if the authenticated user has the right to perform the action requested.
- Auditing
  - Are set of tools used to trace which clients accessed what, and which way.

Security Mechanisms

- Digital signatures
  - Encrypting a file or message using a key known only to signer
- Certificates
  - A digital certificate is a document containing a statement signed by a principal.
- Access control
  - Refer to controlling access to resource via: protection domain, access control list, etc.
- Implementation
  - Creating you own security protocol using APIs.

Security Mechanisms

- Credentials
  - Encrypting a file or message using a key known only to signer Credentials are a set of evidence provided by a principal when requesting access to a resource.
- Firewalls
  - Produce a local communication environment in which all external communication is intercepted.

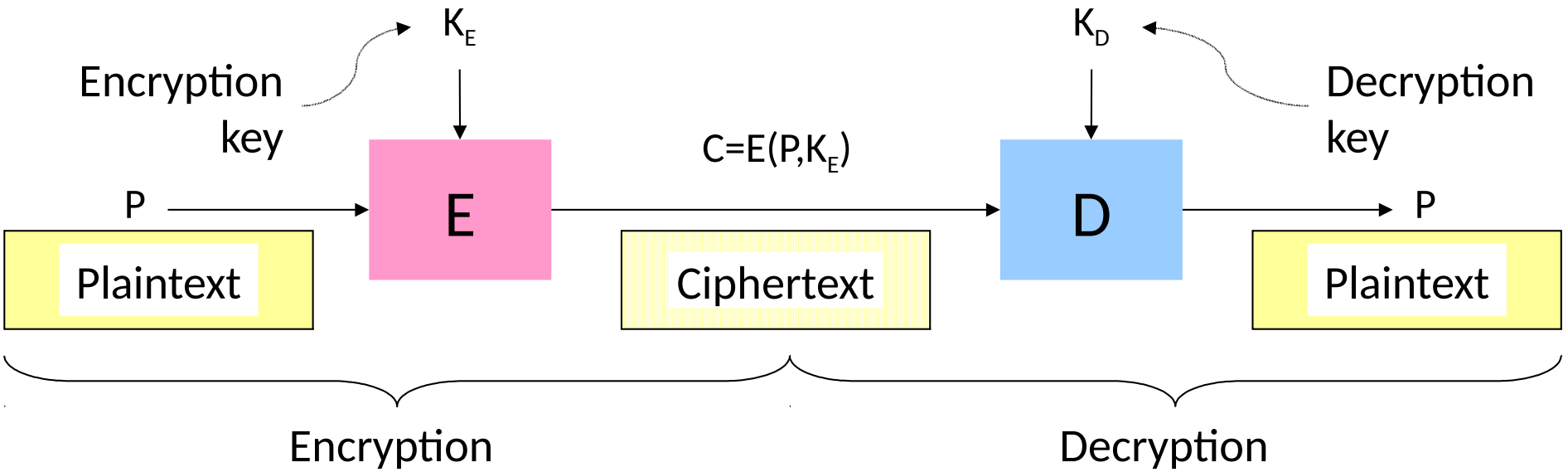
Example For reading

- The Globus Security policy [Foster et al., 1998]
  - Link: http://toolkit.globus.org/ftppub/globus/papers/security.pdf

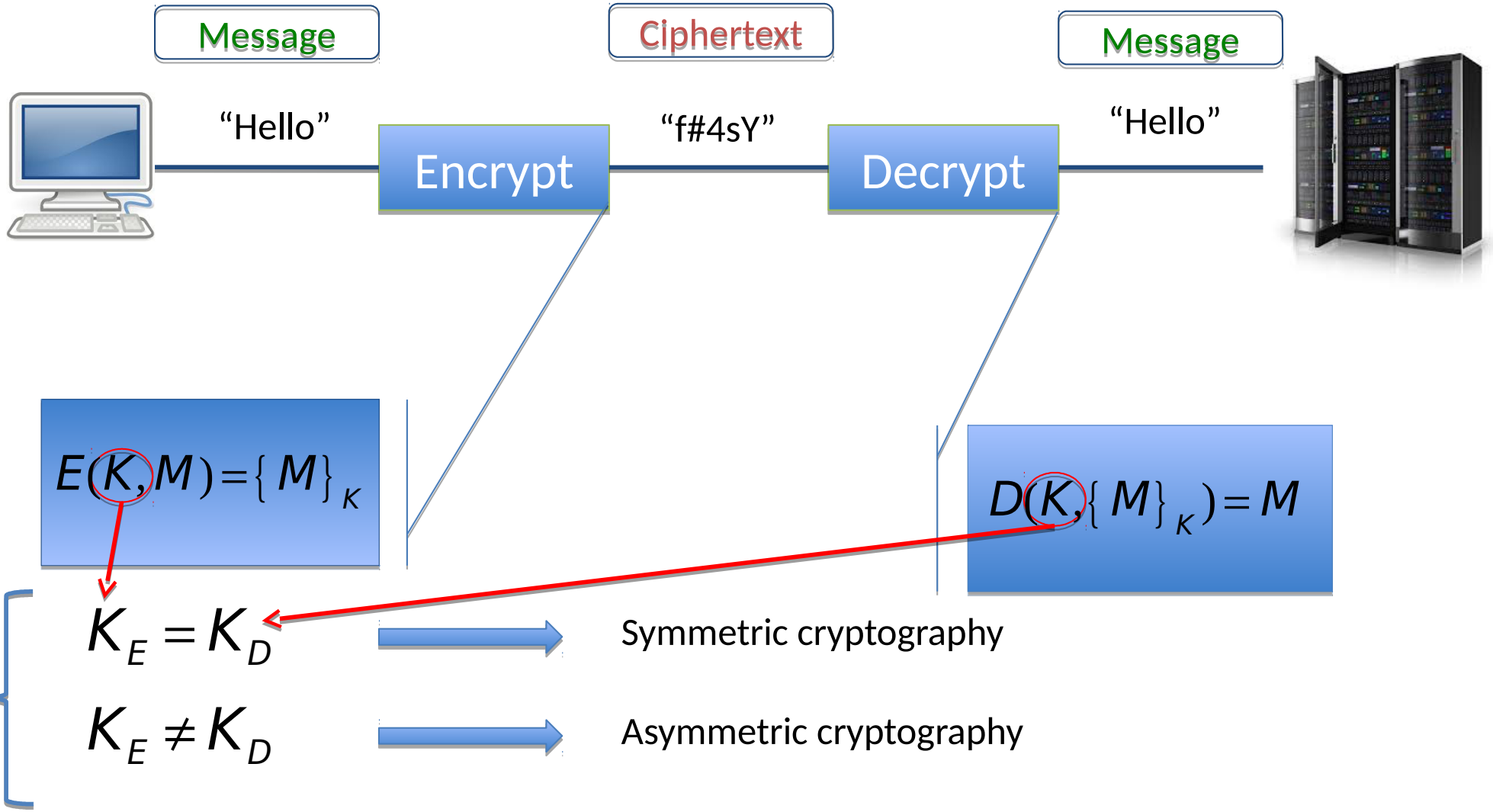
- The Globus Security Architecture [Chervenak et al., 2000].
  - Link: http://www.sciencedirect.com/science/article/pii/S1084804500901103

Cryptography basics

- Algorithms (E, D) are widely known
- Keys (K_E, K_D) may be less widely distributed
- For this to be effective, the ciphertext should be the only information that's available to the world
- Plaintext is known only to the people with the keys (in an ideal world…)

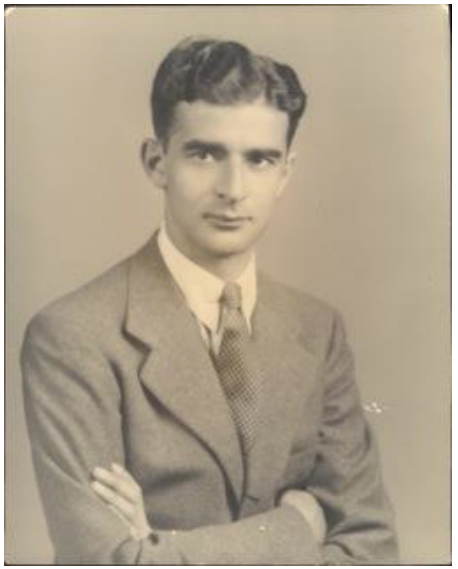


Cryptography



Cryptography algorithms

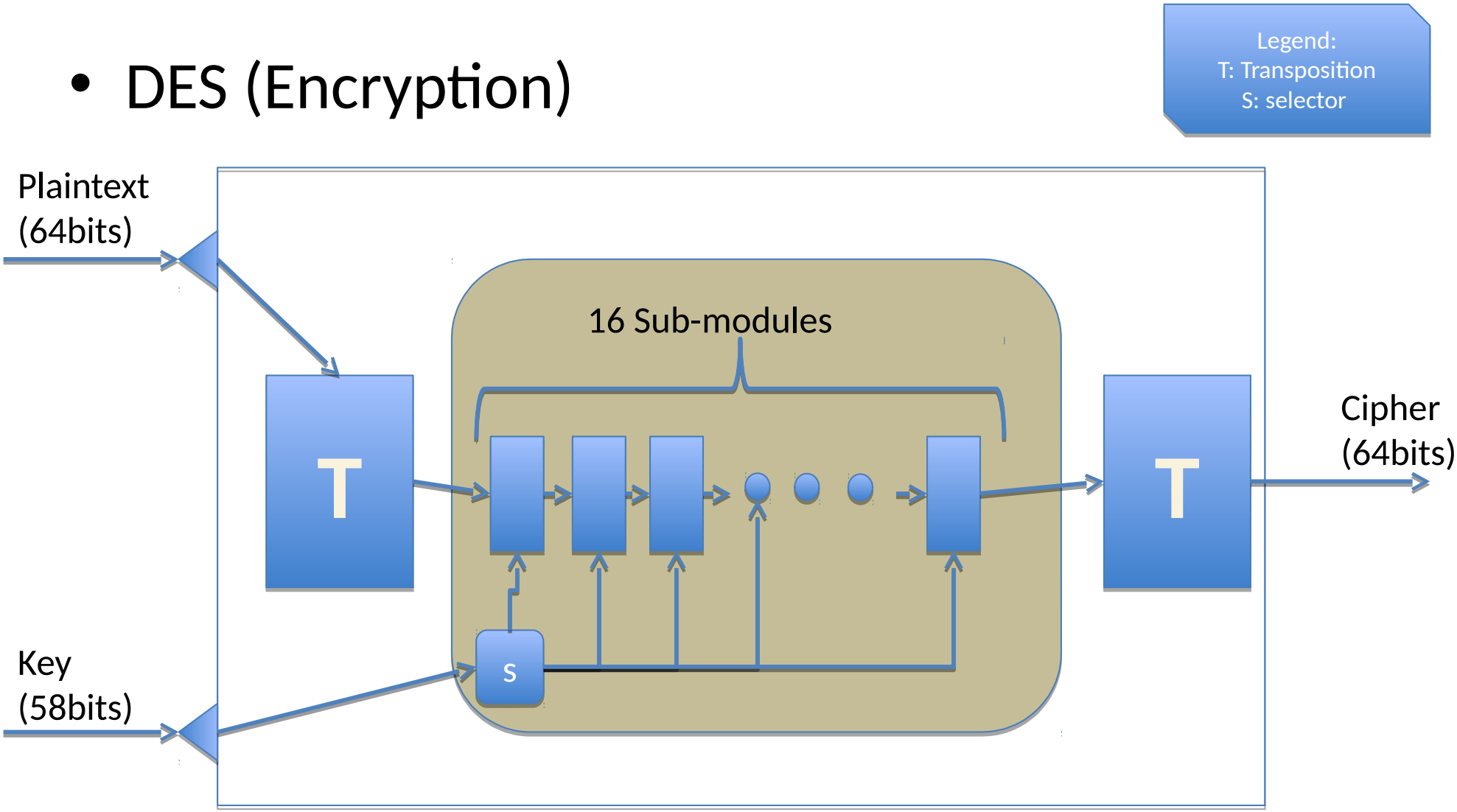
- Symmetric algorithms
  - Data Encryption Standard (DES)
    - Developed at IBM (1970)
    - Based on the design of Horst Feistel, who
      - joins the Computer Science Department (IBM Research Center) in 1968.
      - switches to Mathematical Sciences Department (IBM Research Center) in 1971.
      - retires from IBM to Cape Cod and passes away on November 14, 1990.



Horst Feistel

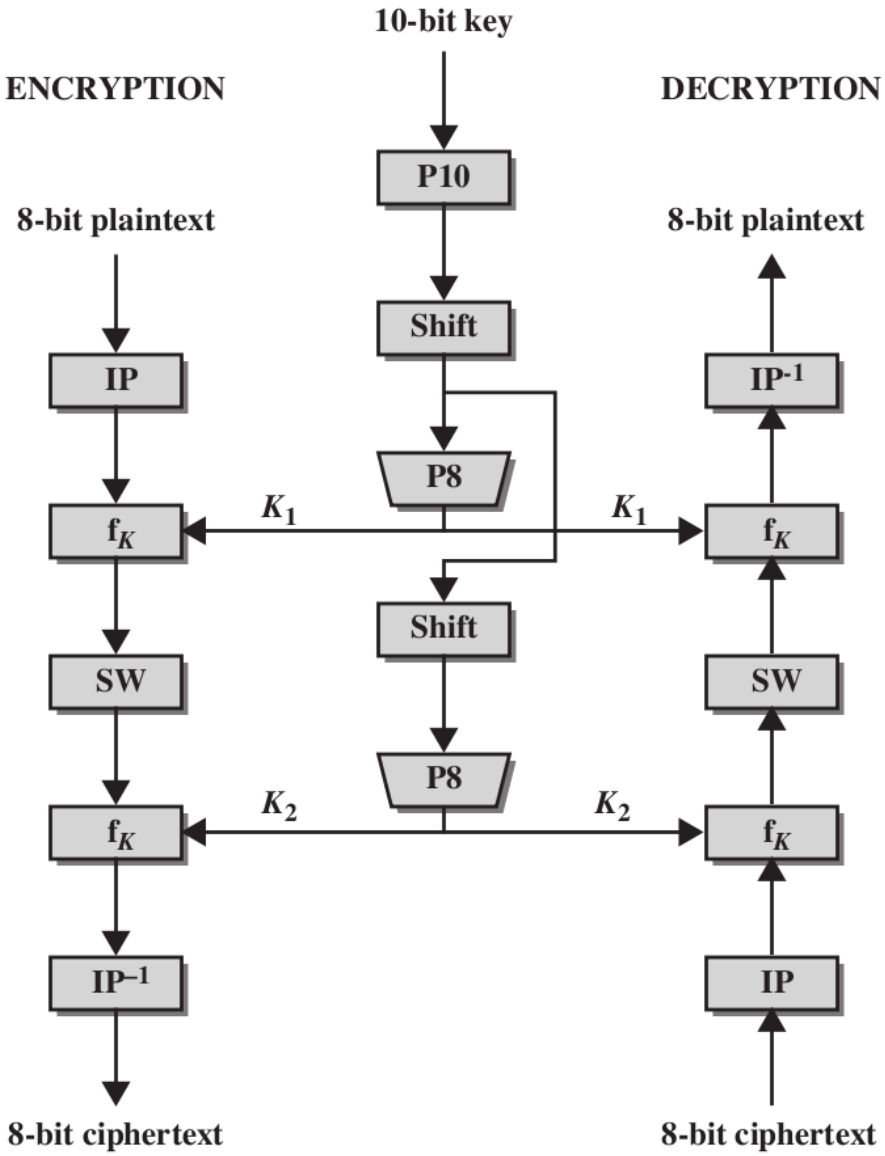
Cryptography algorithms

- DES (Encryption)



Cryptography algorithms

- Simplified DES
  - Input (plaintext) block: 8-bits
  - I Output (ciphertext) block: 8-bits
  - I Key: 10-bits
  - I Rounds: 2
  - I Round keys generated using permutations and left shifts
  - I Encryption: initial permutation, round function, switch halves
  - I Decryption: Same as encryption, except round keys used in opposite order



Cryptography algorithms

- Simplified DES operations
  - P10 (permutate)
    - Input : 1 2 3 4 5 6 7 8 9 10
    - Output: 3 5 2 7 4 10 1 9 8 6
  - P8 (select and permutate)
    - Input : 1 2 3 4 5 6 7 8 9 10
    - Output: 6 3 7 4 8 5 10 9
  - P4 (permutate)
    - Input : 1 2 3 4
    - Output: 2 4 3 1

Cryptography algorithms

- Simplified DES operations
  - EP (expand and permutate)
    - Input : 1 2 3 4
    - Output: 4 1 2 3 2 3 4 1
  - IP (initial permutation)
    - Input : 1 2 3 4 5 6 7 8
    - Output: 2 6 3 1 4 8 5 7
  - IP^{-1} (inverse of IP)
  - LS-1 (left shift 1 position)
  - LS-2 (left shift 2 positions)

Cryptography algorithms

- Simplified DES operations
  - S-DES (and DES) perform substitutions using S-Boxes
  - S-Box considered as a matrix: input used to select row/column; selected element is output
  - 4-bit input: bit1; bit2; bit3; bit4
  - bit1bit4 species row (0, 1, 2 or 3 in decimal)
  - bit2bit3 species column
  - 2-bit output

$$S0 = \begin{bmatrix} 01 & 00 & 11 & 10 \\ 11 & 10 & 01 & 00 \\ 00 & 10 & 01 & 11 \\ 11 & 01 & 11 & 10 \end{bmatrix} \qquad S1 = \begin{bmatrix} 00 & 01 & 10 & 11 \\ 10 & 00 & 01 & 11 \\ 11 & 00 & 01 & 00 \\ 10 & 01 & 00 & 11 \end{bmatrix}$$

Cryptography algorithms

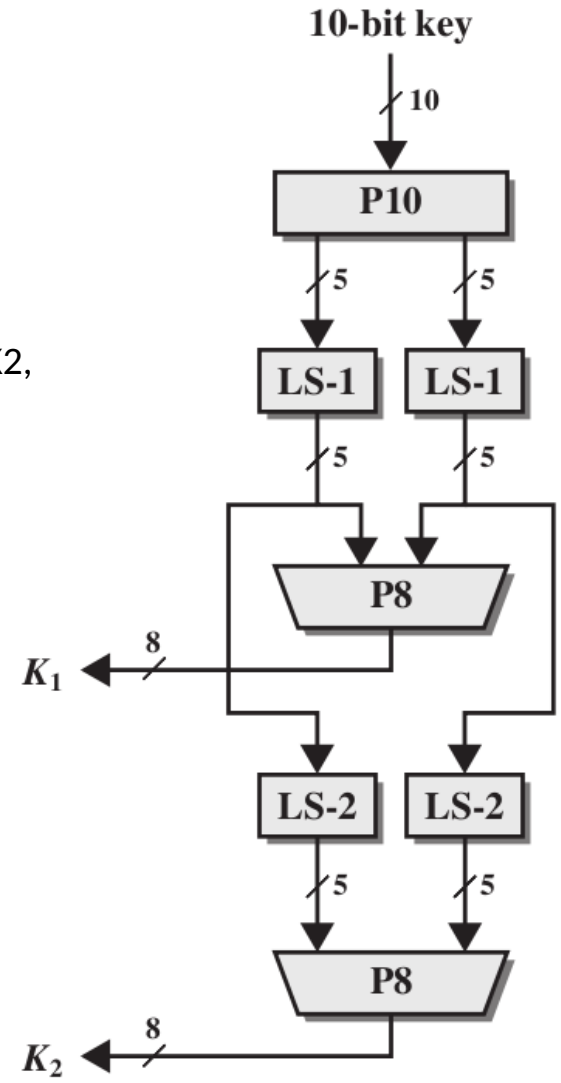
- Key Generator

Assume input 10-bit key, K, is: 1010000010

Then the steps for generating the two 8-bit round keys, K1 and K2, are:

1. Rearrange K using P10: 1000001100
2. Left shift by 1 position both the left and right halves: 00001 11000
3. Rearrange the halves with P8 to produce K1: 10100100
4. Left shift by 2 positions the left and right halves: 00100 00011
5. Rearrange the halves with P8 to produce K2: 01000011

K1 and K2 are used as inputs in the encryption and decryption stages.



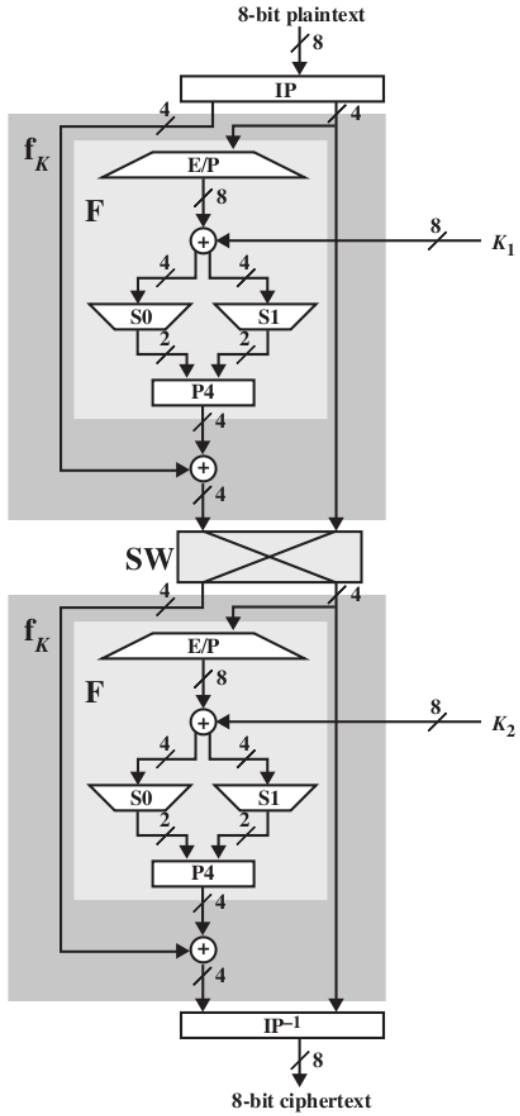
Cryptography algorithms

- Encryption

Assume a 8-bit plaintext, P: 01110010

Then the steps for encryption are:

1. Apply the initial permutation, IP, on P: 10101001
2. Assume the input from step 1 is in two halves, L and R: L=1010, R=1001
3. Expand and permutate R using E/P: 11000011
4. XOR input from step 3 with K1: 10100100 XOR 11000011 = 01100111
5. Input left halve of step 4 into S-Box S0 and right halve into S-Box S1:
    a. For S0: 0110 as input: b1,b4 for row, b2,b3 for column
    b. Row 00, column 11 -> output is 10
    c. For S1: 0111 as input:
    d. Row 01, column 11 -> output is 11
6. Rearrange outputs from step 5 (1011) using P4: 0111
7. XOR output from step 6 with L from step 2: 0111 XOR 1010 = 1101
8. Now we have the output of step 7 as the left half and the original R as the right half.
Switch the halves and move to round 2: 1001 1101
9. E/P with right half: E/P(1101) = 11101011
10. XOR output of step 9 with K2: 11101011 XOR 01000011 = 10101000

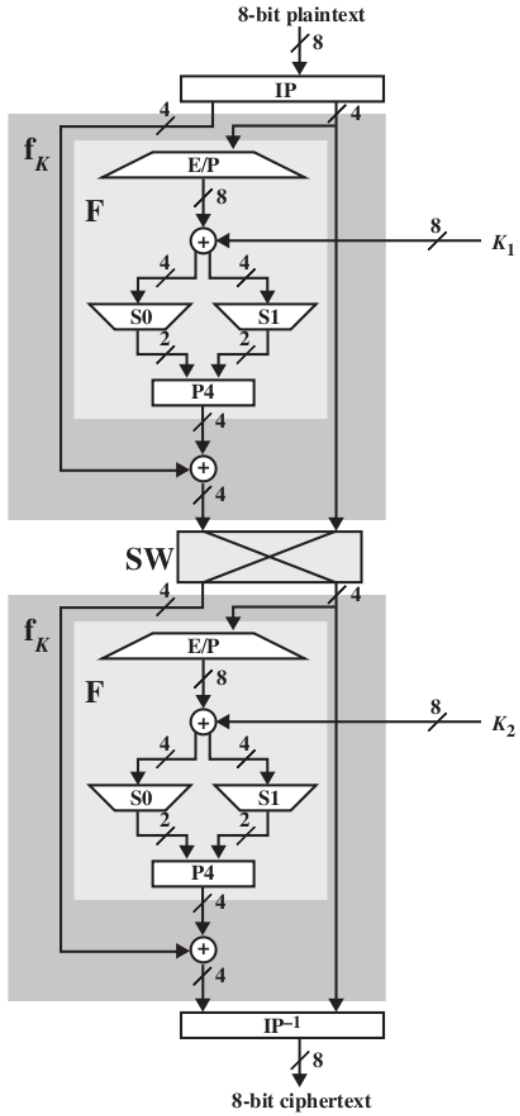


Cryptography algorithms

- Encryption

11. Input to s-boxes:
    a. For S0, 1010
    b. Row 10, column 01 -> output is 10
    c. For S1, 1000
    d. Row 10, column 00 -> output is 11
12. Rearrange output from step 11 (1011) using P4: 0111
13. XOR output of step 12 with left halve from step 8: 0111 XOR 1001 = 1110
14. Input output from step 13 and right halve from step 8 into inverse IP
    a. Input us 1110 1101
    b. Output is: 01110111

So our encrypted result of plaintext 01110010 with key 1010000010 is: 01110111



Cryptography algorithms

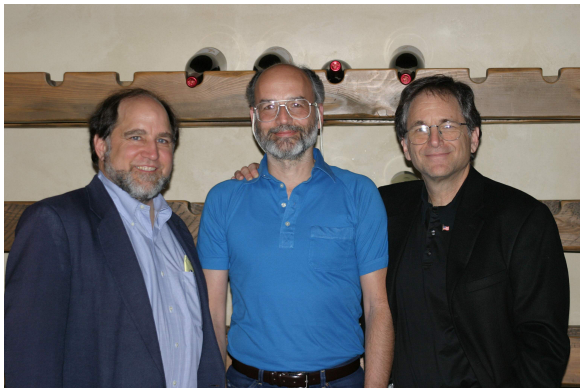
- S-DES Summary:
  - S-DES expressed as functions:
    - ciphertext = IP^{-1} (fK2 (SW (fK1 (IP (plaintext)))))
    - plaintext = IP^{-1} (fK1 (SW (fK2 (IP (ciphertext)))))

Cryptography algorithms

- Asymmetric algorithms
  - Ron Rivest, Adi Shamir, and Len Adelman (RSA)

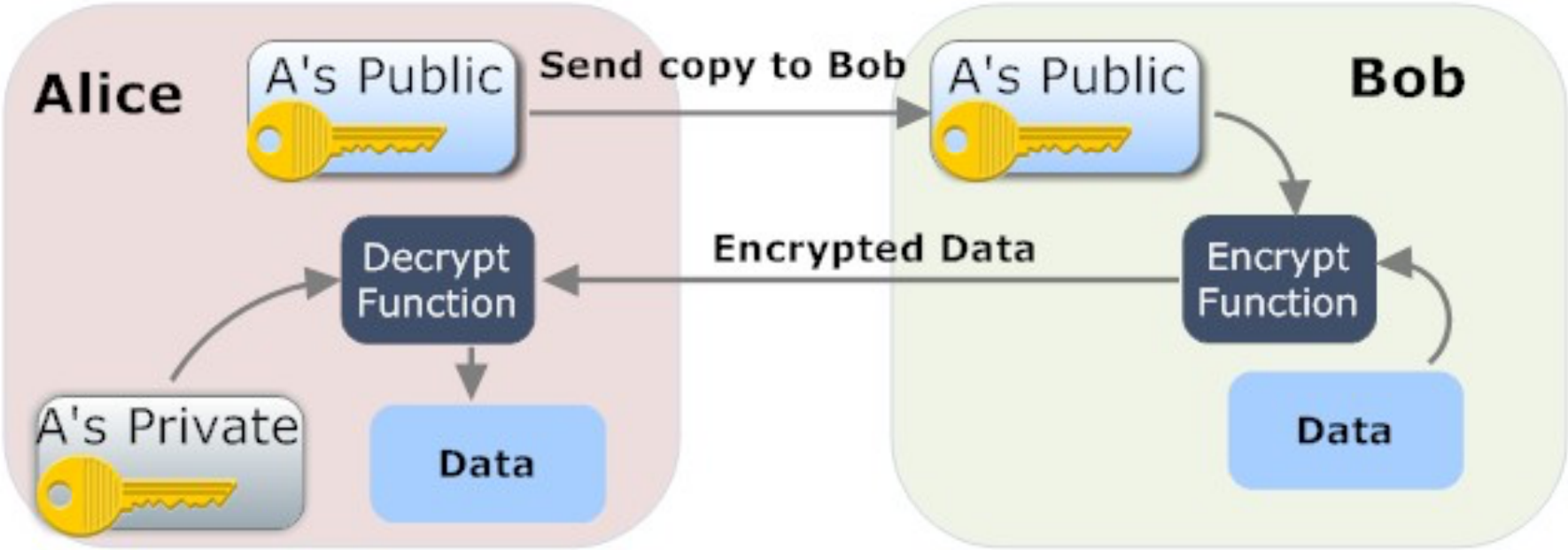
Diffie and Hellman
Concept of public-key cryptography
(1976)

The First algorithm that met the requirement of the public-key systems was the RSA algorithm developed by Rivest, Shamir and Adelman (1978)



Cryptography algorithms

- RSA algorithm



Cryptography algorithms

RSA Algorithm:

Key Generation

Select p, q	p and q both prime, $p \quad q$
Calculate $n = p \times q$	
Calculate $\phi(n) = (p-1)(q-1)$	
Select integer e	$\gcd(\phi(n), e) = 1; \quad 1 < e < \phi(n)$
Calculate d	$d \equiv e^{-1} \bmod \phi(n)$
Public key	$KU = \{e, n\}$
Private key	$KR = \{d, n\}$

Encryption

Plaintext:	$M < n$
Ciphertext:	$C = M^e \pmod{n}$

Decryption

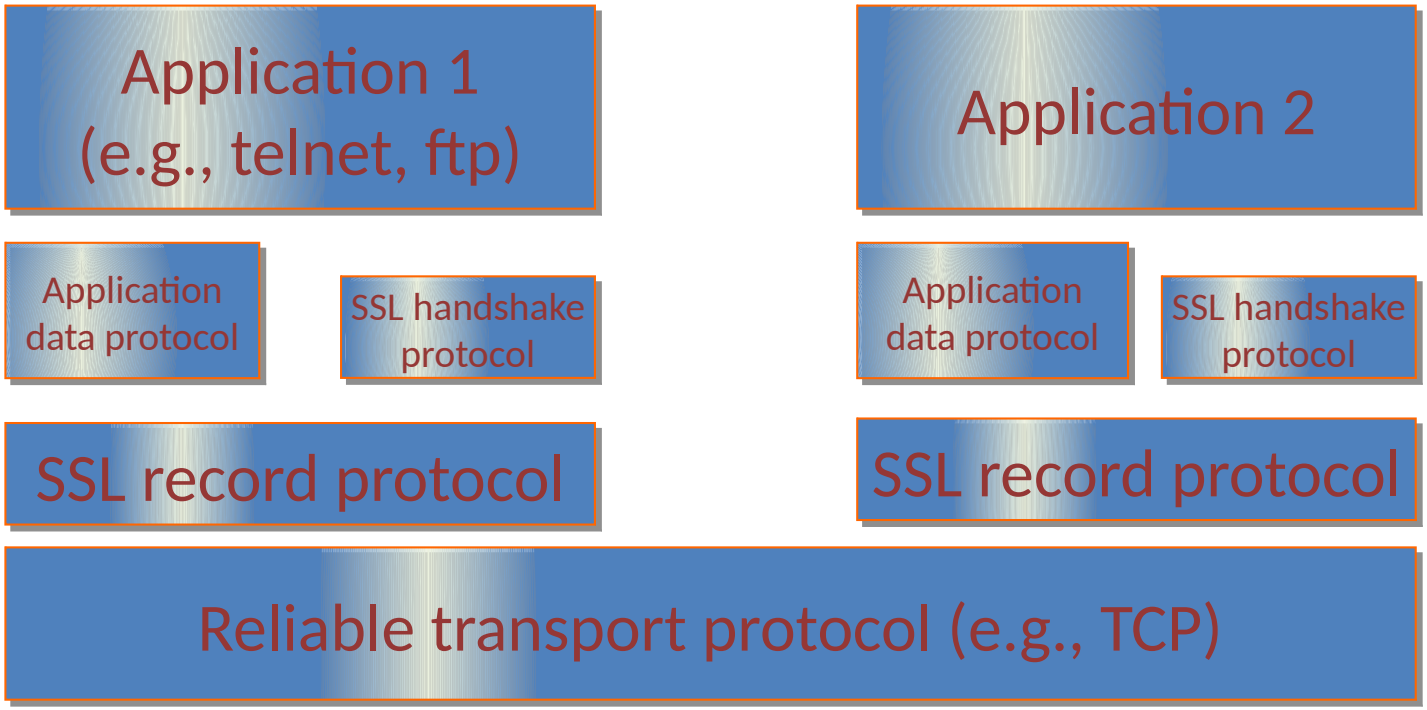
Ciphertext:	C
Plaintext:	$M = C^d \pmod{n}$

Example Code: RSA KeyGen

```
def generate_RSA(bits=2048):
    '''
    Generate an RSA keypair with an exponent of 65537 in PEM format
    param: bits The key length in bits
    Return private key and public key
    '''
    from Crypto.PublicKey import RSA
    new_key = RSA.generate(bits, e=65537)
    public_key = new_key.publickey().exportKey("PEM")
    private_key = new_key.exportKey("PEM")
    return private_key, public_key
```

SSL Protocol

- Secure Socket Layer
  - Provide security for application over the insecure internet

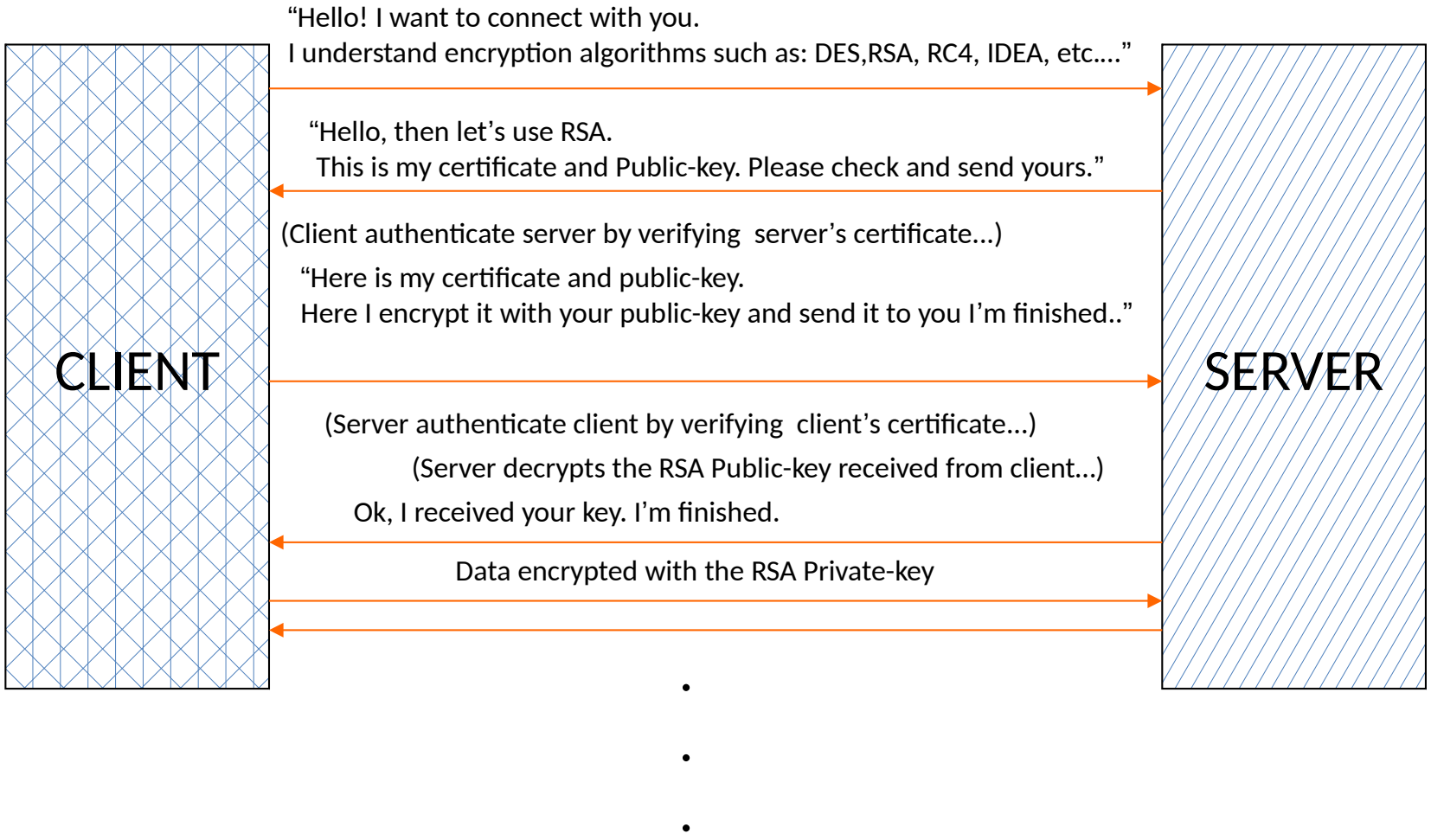


SSL Protocol - Components

- Record protocol
  - Encapsulate
  - Message splitting into blocks
  - Compresses message blocks
  - Applies MAC to message blocks
  - Encrypts and transmits message blocks
- SSL handshake protocol
  - Allow mutual authentication (b/w client & server)
  - Ensure an encryption algorithm and key for application data transmission
- Application data protocol
  - Transmits data: applications record layer (data sent securely)

SSL Protocol - Handshake

- Illustration:



Case study - Kerberos

- Kerberos?
  - Network authentication protocol
  - Developed by MIT, 1980
  - Open source / some support commercial software.

Case study - Kerberos

- Why?
  - Because sending user names & password in the clear jeopardizes the security of the network.
  - There is always a chance that the password can be intercepted.

Case study - Kerberos

- Firewall vs Kerberos?
  - Firewall thinks always that the weak link is coming from outside. (attackers are coming from outside)
  - Kerberos assume that the network connection are the weak link.

Case study - Kerberos

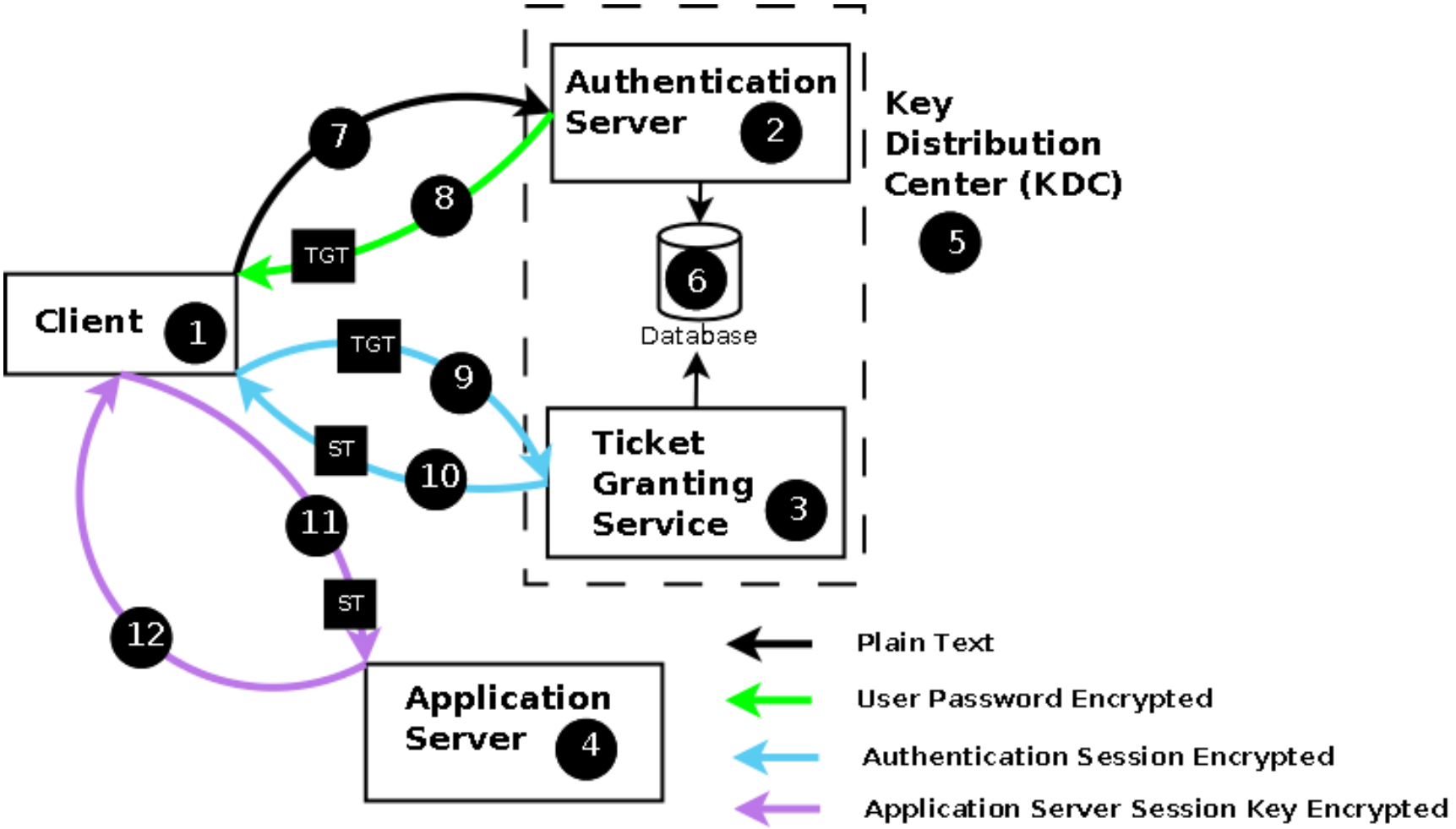
- Terminology
  - Realm
    - means the set of users and application servers that the Key Distribution Center (KDC) covers - or has information about.
  - Principal
    - is the string that fully identifies a user of the Kerberos service and has the form thing@REALM.
  - Ticket
    - is a data structure whose content is known only to the issuer of the Ticket and any party or parties to which the ticket is relevant.

Case study - Kerberos

- How does it work?
  - Instead of client sending password to server:
    - Request Ticket from authentication server
    - Ticket and encrypted request sent to application server
  - Ticket are requested using Ticket Granting Ticket TGT in order to avoid repetition in sending credentials.

Case study - Kerberos

- How does it work?



Source: http://www.zytrax.com/tech/survival/kerberos.html