

Biomedical Wearable Technologies
for Healthcare and Wellbeing

Elements of network security

A.Y. 2023-2024
Giacomo Cappon



Agenda

- Definitions of network security and cryptography
- Symmetric-key cryptography
- Asymmetric-key cryptography
- Digital signature algorithms and hash function
- Elements of security in network communication

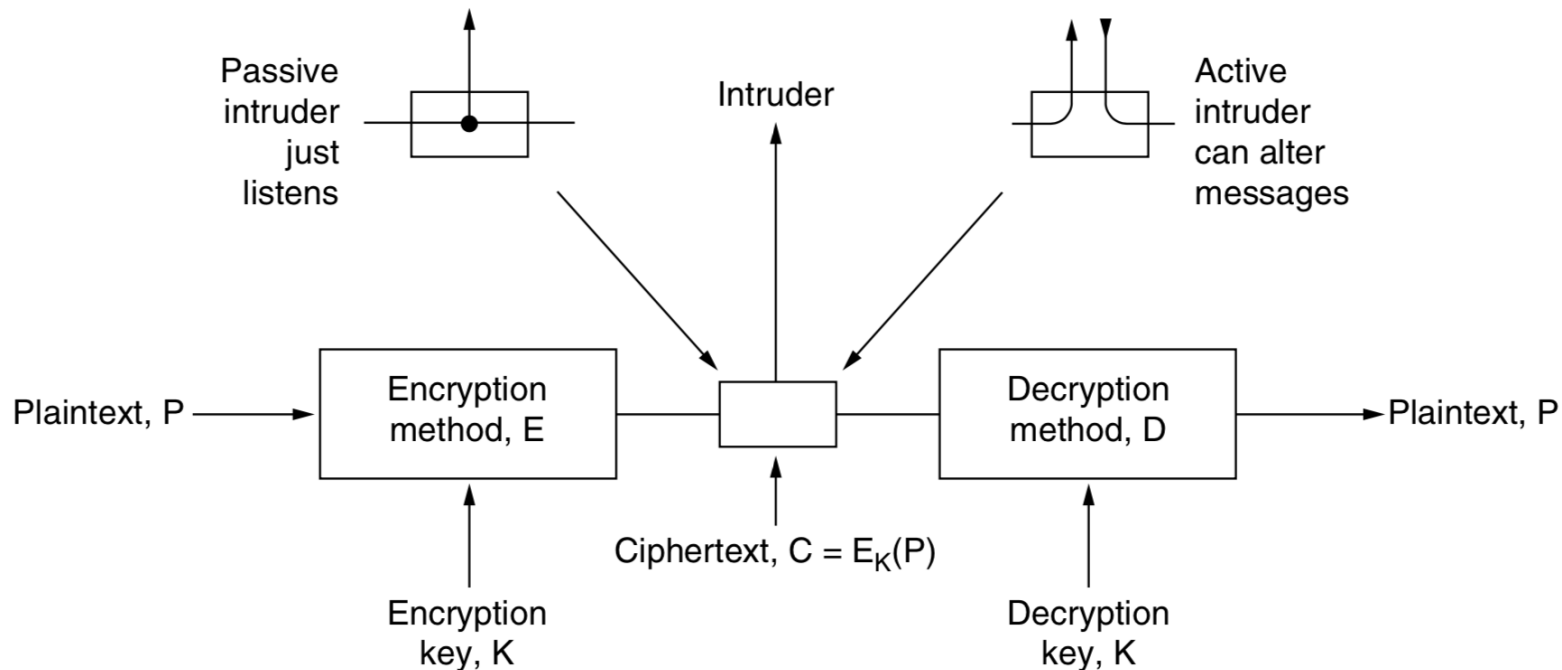
Network security

- Network security problems can be divided roughly into four areas:
 - **Secrecy**, also called **confidentiality** → keeping information secret to unauthorized users.
 - **Authentication** → determining whom you are talking to before revealing sensitive information.
 - **Nonrepudiation** → signatures to ensure that the user you are talking to is who it claims to be.
 - **Integrity control** → how you can be sure that a message you received was really the one sent and not something that a malicious adversary modified in transit or concocted.

- All network security is based on cryptographic principles.

Cryptography

- **Cryptography:** the study of secure communications techniques that allow only the sender and intended recipient of a message to view its contents.
- **Cipher:** a transformation of the message, without regard to the linguistic structure of the message.
- General encryption model (symmetric cryptography):



$$C = E_K(P)$$
$$P = D_K(C)$$

Cryptography

- **Cryptanalysis:** the art of breaking ciphers
- **Cryptology:** the art of devising ciphers
- A fundamental rule of cryptography is that one must assume that the cryptanalyst knows the methods used for encryption (E) and decryption (D).
- **Kerckhoff's principle** (1883): All algorithms must be public; only the keys are secret.
- The secrecy of a cipher stands on the secrecy of the key.
- The key must be sufficiently complex not to allow an intruder to discover the key by exhaustive search!

Two historic ciphers: substitution ciphers

- **Substitution ciphers:** each letter or group of letters is replaced by another letter or group of letters.
- **Caesar cipher:** each letter is substituted with the letter k steps ahead in the alphabet, where k is the key of the cipher. Only 25 possible keys.

Plain text:

A	B	C	D	E	...	W	X	Y	Z
---	---	---	---	---	-----	---	---	---	---

Encrypted text:

D	E	F	G	H	...	Z	A	B	C
---	---	---	---	---	-----	---	---	---	---

 $k = 3$

- **Monoalphabetic substitution cipher:** each letter is substituted with another general alphabet letter. The key is the 26-letter string representing the alphabet permutation. Number of possible keys: $26!$ However, this cipher can be easily broken comparing the frequency of letters in the plain and encrypted text.

Plain text:

A	B	C	D	E	...	W	X	Y	Z
---	---	---	---	---	-----	---	---	---	---

Encrypted text:

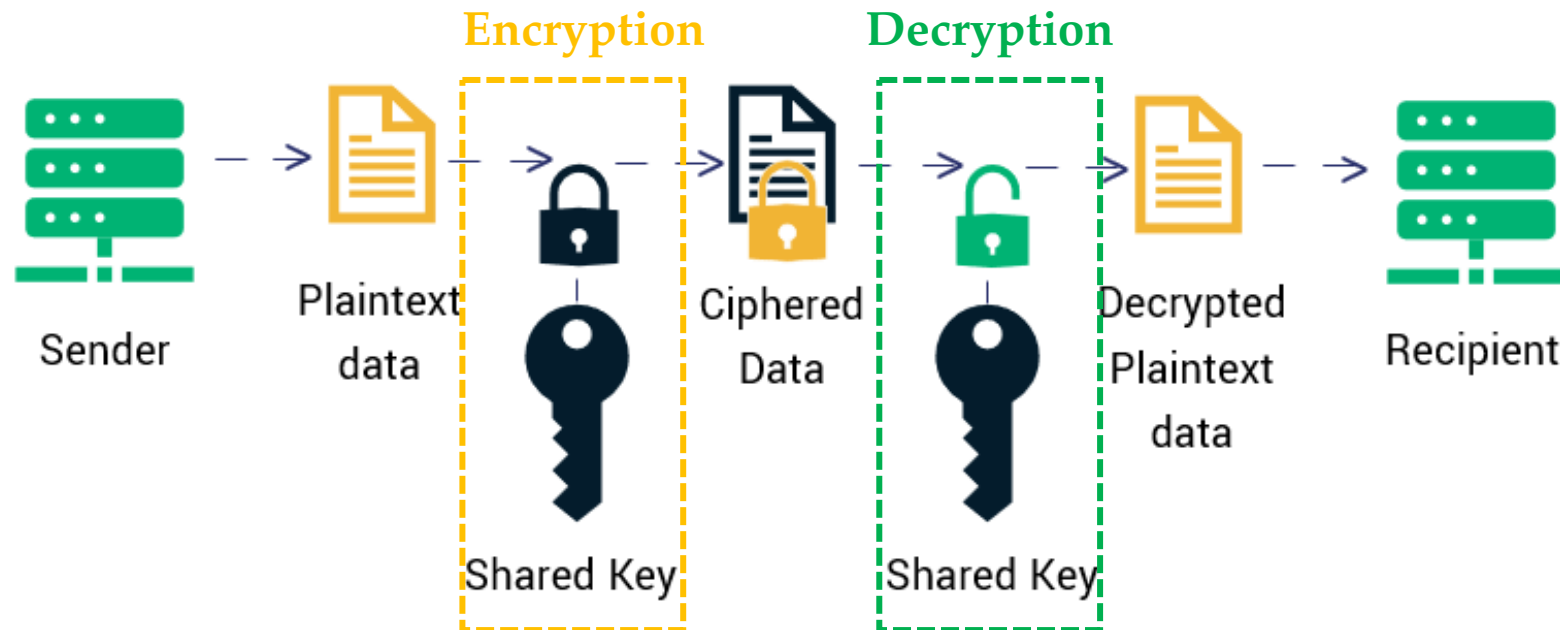
F	M	P	Z	G	...	S	J	L	A
---	---	---	---	---	-----	---	---	---	---

Modern cryptography

- The encryption algorithm is made so complex that, even if the cryptanalyst acquires vast amount of enciphered text, he/she will not be able to make any sense of it without the key.
- There are mainly two classes of cryptography algorithms:
 - **Symmetric-key algorithms** → same key for encryption and decryption
 - **Asymmetric-key algorithms** → different key for encryption and decryption

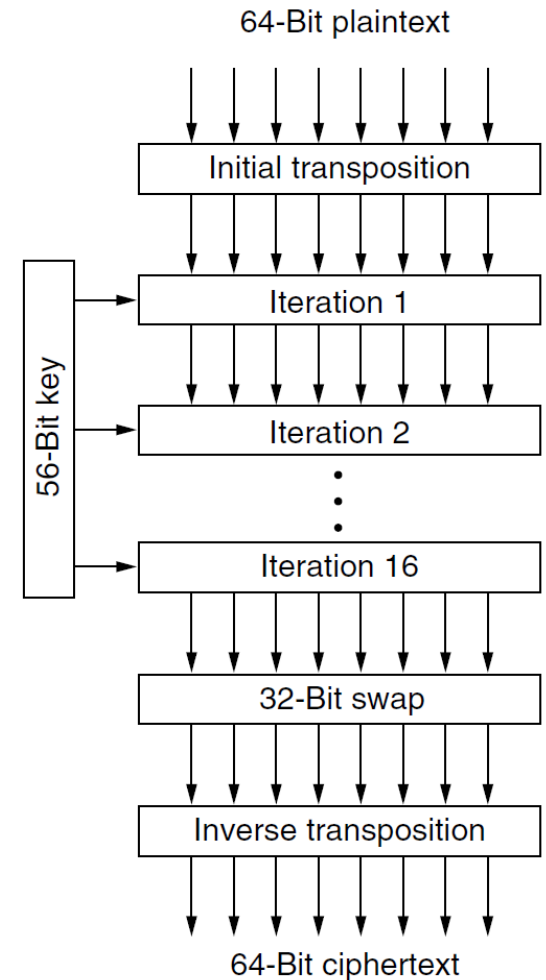
Symmetric-key algorithms

- The key for encrypting the data is the same to be used for decrypting the ciphered message.
- The key must be shared between the sender and the recipient, but it must be protected, because any user having the key can potentially decrypt the message.
- Also called **private-key** cryptography algorithms.



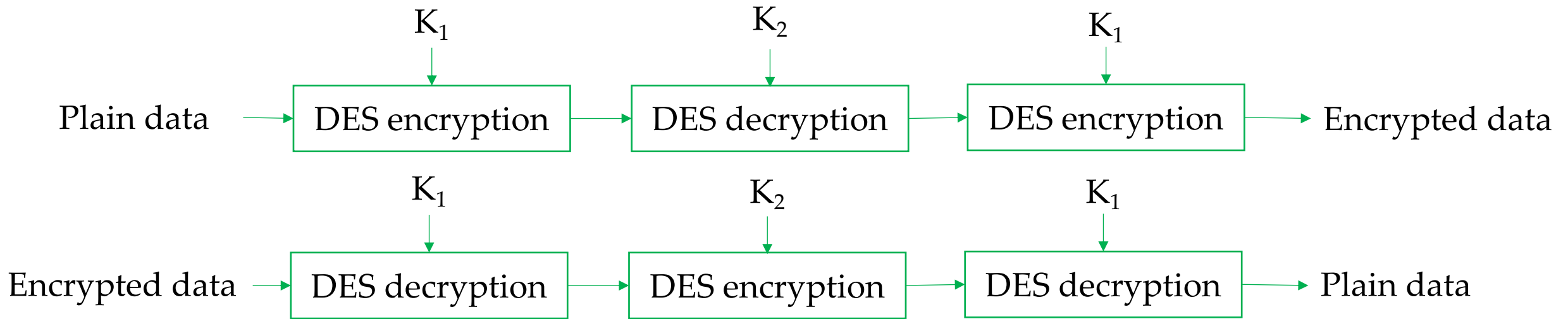
Data Encryption Standard (DES)

- Introduced in the '70 by IBM and adopted as a security standard by the U.S. government for many years.
- Encryption by groups of 64 bits.
- Key K made of 56 bits.
 - From K, creates 16 keys (K_i) of 48 bit using specific functions
- Encryption steps:
 - Key-independent transposition
 - 16 encryption iterations using each K_i
 - Swap of the leftmost 32 bits with the rightmost 32 bits
 - Inverse of the key-independent transposition
- Decryption done with the same steps in reverse order.



Triple DES

- 1977: Diffie and Hellman (Stanford) designed a machine able to break DES by exhaustive search in less than a day and estimated that it could be built for 20 million \$.
- 1979: IBM proposed **Triple DES algorithm**



- The triple DES key is 112-bit long (K_1, K_2).
- With $K_1 = K_2$ the triple DES is equivalent to DES.

Advanced Encryption Standard (AES)

- 1997: the U.S. National Institute of Standards and Technology (NIST) invited researchers from all over the world to submit proposals for a new standard, to be called **AES (Advanced Encryption Standard)**.
- 15 serious algorithms were proposed.
- 2000: NIST proclaimed the winner: **Rijndael**, by Joan Daemen and Vincent Rijmen, two young Belgian cryptographers.
- 2001: Rijndael became the new AES adopted by U.S. government.
- Two variants of AES are generally used:
 - Variant with blocks of 128 bits and a key of 128 bits
 - Variant with blocks of 128 bits and a key of 256 bits

Still considered one of the most secure cryptography algorithms

Limitations of the symmetric-key algorithms

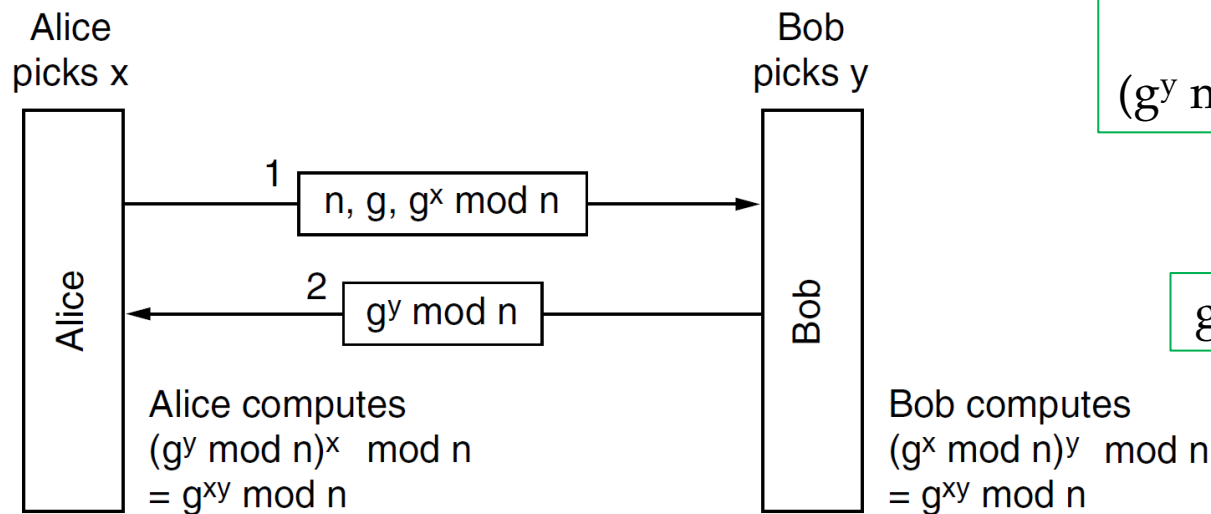
- The privacy of the key is fundamental. No matter how strong a cryptosystem is, if an intruder can steal the key, the system is worthless.
- The key must be shared between the sender and the recipient.

➡ need to distribute the key and at the same time to keep it protected



Diffie-Hellman key exchange protocol

- **Diffie-Hellman key exchange** protocol: two users (Alice and Bob) that never met before can establish a shared secret key (Diffie and Hellman, 1976).
- Alice and Bob have to agree on two large numbers, n and g , where n is a prime, $(n - 1)/2$ is also a prime, and certain conditions apply to g . These numbers may be public.
- Now Alice picks a large (e.g., 1024-bit) number, x , and keeps it secret.
- Similarly, Bob picks a large secret number, y .



By the laws of modular arithmetic:
 $(g^y \bmod n)^x \bmod n = (g^x \bmod n)^y \bmod n = g^{xy} \bmod n$



$g^{xy} \bmod n$ is the secret key for Alice and Bob

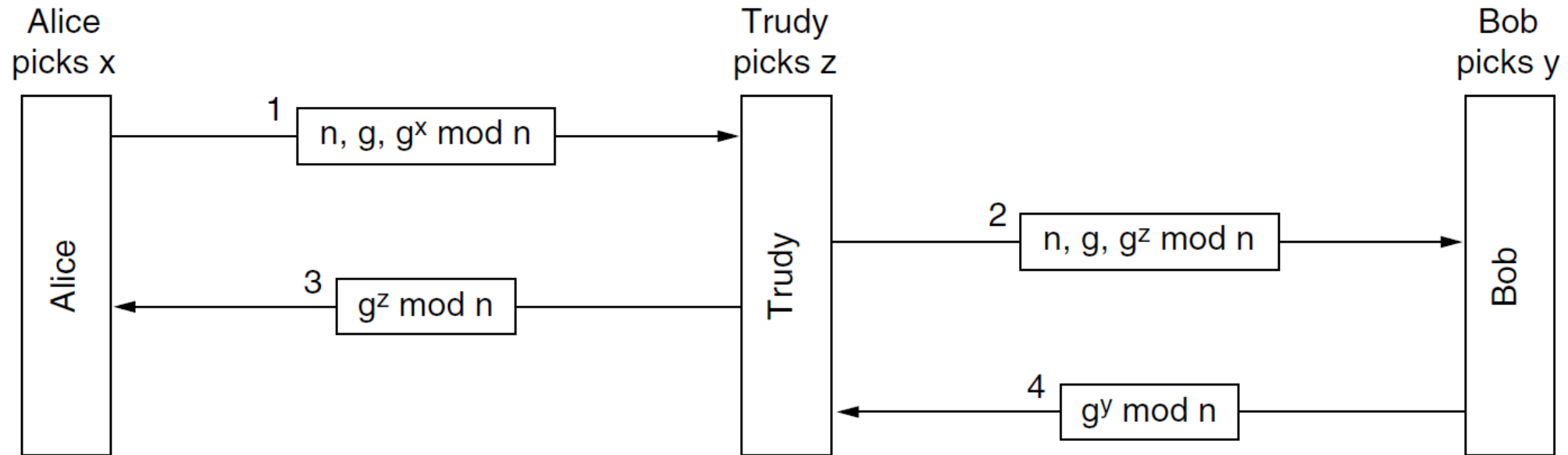
- The strength of the approach is that from only $g^x \bmod n$, n , and g is very difficult to find x .

Example of generation of a secret key

- Alice and Bob publicly agree to use $n = 23$ and $g = 5$ (which is a primitive root modulo 23).
- Alice chooses a secret integer $x = 4$, then sends Bob $A = g^x \bmod n$
 - $A = 5^4 \bmod 23 = 4$
- Bob chooses a secret integer $y = 3$, then sends Alice $B = g^y \bmod n$
 - $B = 5^3 \bmod 23 = 10$
- Alice computes $s = B^x \bmod n$
 - $s = 10^4 \bmod 23 = 18$
- Bob computes $s = A^y \bmod n$
 - $s = 4^3 \bmod 23 = 18$
- Alice and Bob now share a secret key (the number 18).

Man-in-the-middle attack

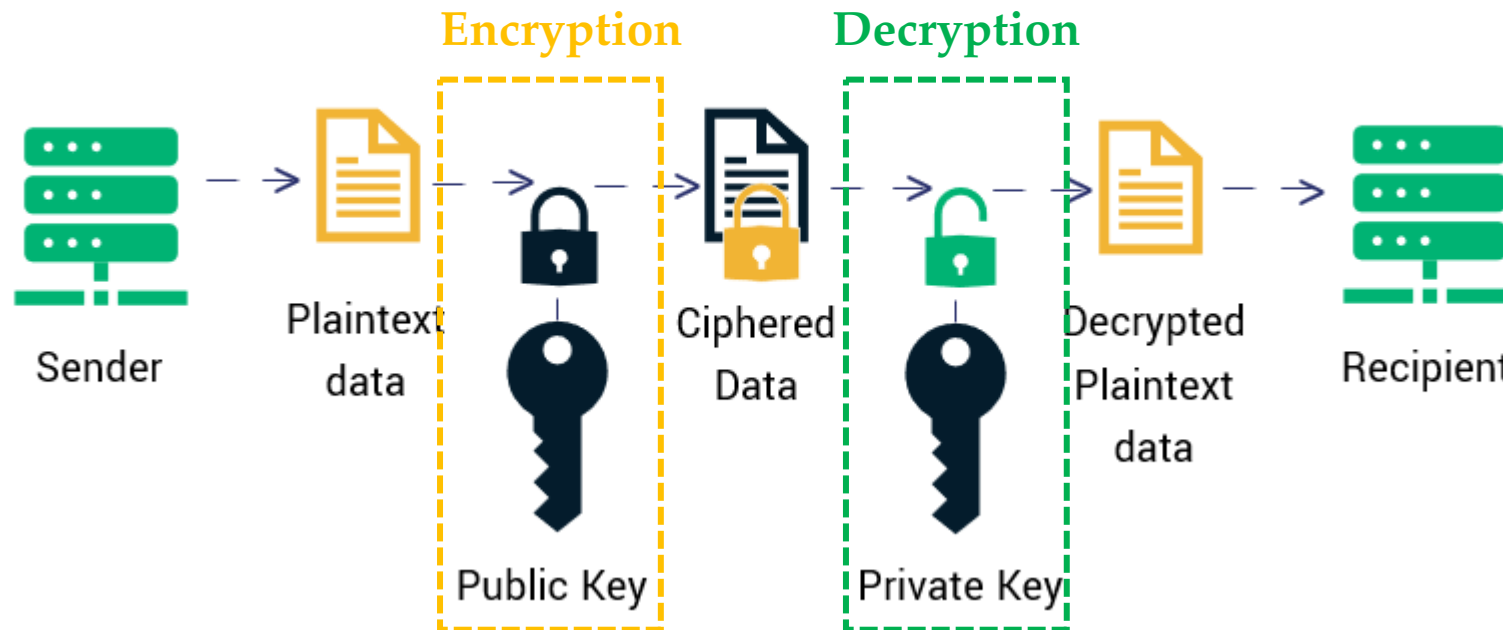
- Critical aspect: this approach can be violated by a man-in-the-middle attack



- 1976: Diffie and Hellman (Stanford University) proposed a new kind of cryptosystem, in which the encryption and decryption keys were different
- So different that the decryption key could not feasibly be derived from the encryption key.

Asymmetric-key algorithms

- The key for encrypting is different from the key to be used for decrypting the message.
- Each recipient has a pair of twin keys, a **public key** and **private key**, such that it is extremely difficult to derive the private key from the public one.
- The sender must encrypt the message using the public key of the recipient.
- The recipient can decrypt the ciphered message using its own private key.
- Also called **public-key cryptographic algorithms**.



The public key can be distributed also in unsafe channels.

RSA

- 1978: Rivest, Shamir and Adleman (MIT) proposed the **RSA** asymmetric-key algorithm.
- The generation of the keys is based on some number principles by these 4 steps:
 1. Choose two distinct large prime numbers, p and q (typically 1024 bits)
 2. Compute $N = p \times q$ and $z = (p - 1) \times (q - 1)$.
 3. Choose an integer number relatively prime to z and call it d .
 4. Find an integer number e such that the rest of $e \times d$ divided by z is 1.
- Public key: (N, e)
- Private key: (N, d)
- p and q are secret, only who generated the keys knows them.
- Strength of RSA: from N it is very difficult to recover p and q
 - from the public key it is very difficult to retrieve the private key

Encryption and decryption with RSA

- The message is divided into blocks of k bits such that $2^k < N$
- Each k -bit number of the plain text, P , is encrypted by the operation

$$C = P^e \bmod N$$

- It can be proven that C can be decrypted using d and N as:

$$P = C^d \bmod N$$

Public key: (N, e)
Private key: (N, d)



Example with public key $(33, 3)$ and private key $(33, 7)$

Plaintext (P)		Ciphertext (C)			After decryption	
Symbolic	Numeric	P^3	$P^3 \bmod 33$	C^7	$C^7 \bmod 33$	Symbolic
S	19	6859	28	13492928512	19	S
U	21	9261	21	1801088541	21	U
Z	26	17576	20	1280000000	26	Z
A	01	1	1	1	01	A
N	14	2744	5	78125	14	N
N	14	2744	5	78125	14	N
E	05	125	26	8031810176	05	E



Sender's computation
Receiver's computation

Symmetric vs asymmetric key algorithms

➤ Symmetric algorithms:

- Their application for encryption and decryption is fast 
- Need to distribute the private key through a safe communication channel 
- A new key must be generated for each pair of communication entities

➤ Asymmetric algorithms:

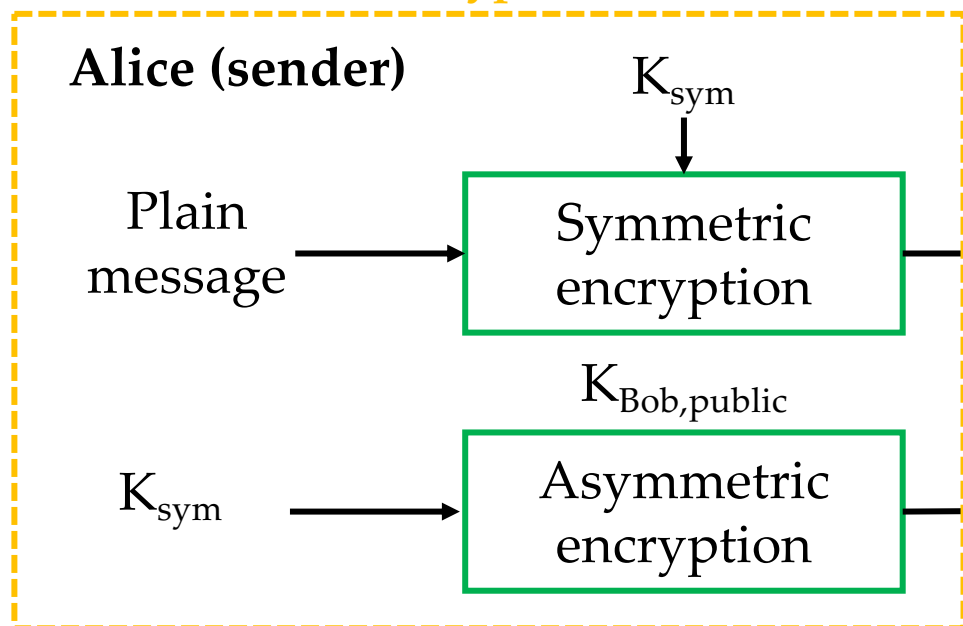
- Very strong security level, no need to distribute the private key. 
- There is a pair of keys for each user.
- It requires keys of at least 1024 bits for good security, which makes the encryption and decryption quite slow. 

Hybrid algorithms

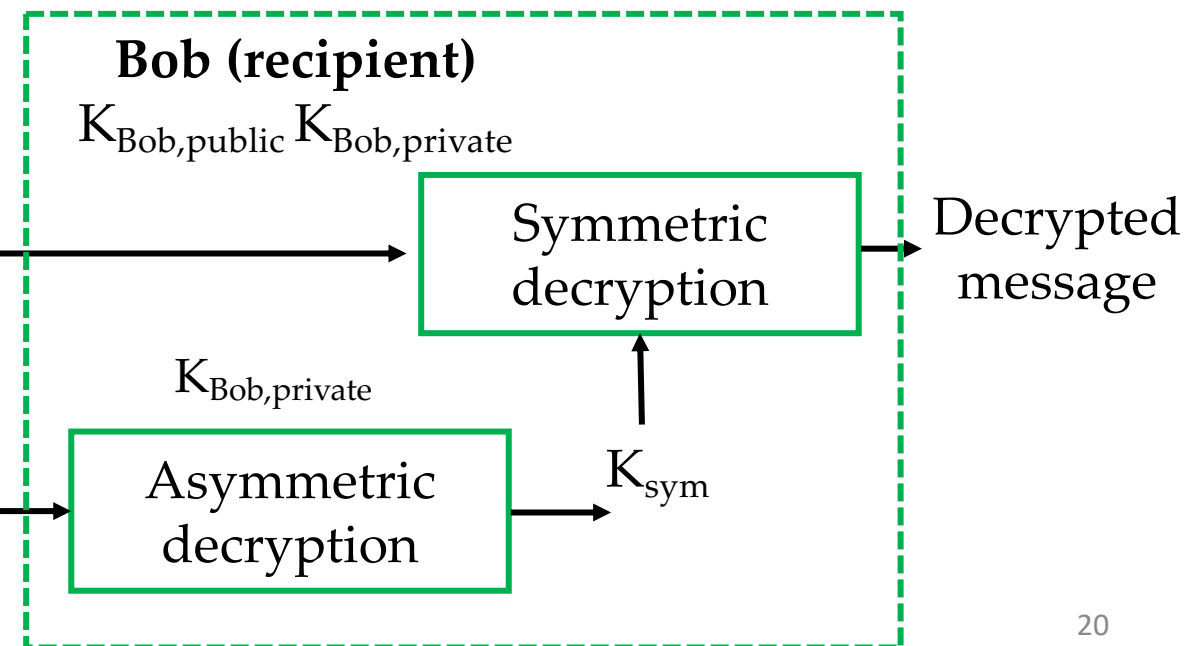
BONUS

- Asymmetric-key and symmetric-key algorithms combined to have the advantages of both approaches.
- Symmetric algorithm (fast) used to encrypt the message M with key K_{sym} .
- K_{sym} , typically much shorter than M , encrypted with the asymmetric algorithm (slow).
- Encrypted $K_{\text{sym},\text{en}}$ securely distributed even using an unsafe communication channel.

Encryption



Decryption



Network security

➤ **Secrecy or confidentiality.**

- Addressed by cryptography algorithms (symmetric-key, asymmetric-key, or hybrid)

➤ **Authentication**

➤ **Nonrepudiation**

➤ **Integrity control**



They can be achieved by digital signature

Digital signatures

- The authenticity of documents is determined by the presence or absence of an authorized handwritten signature.
- **Digital signature** replaces handwritten signature for digital messages.
- Requirements of a digital signature:
 - The receiver can verify the claimed identity of the sender (**authentication**).
 - The sender cannot later repudiate the contents of the message (**non-repudiation**).
 - The receiver cannot possibly have altered the message himself (**integrity**).

Digital signature by symmetric-key cryptography

- $E_K(.)$: Function for symmetric encryption with key K
- Each user is provided a secret key to be used in symmetric-key cryptography.
- A central trusted authority knows the keys of every user and can verify the identify of all users.

Alice (sender)

K_a Plaintext P

Ciphertext C
 $C = E_{K_a}(B, P)$

Bob's identity

Central trusted authority
with secret key K_c

K_a is secret key of Alice
 K_b is the secret key of Bob

The central authority decrypts the message with K_a and thus authenticates Alice.

Bob (recipient)

K_b

$C' = E_{K_b}(A, P, E_{K_c}(A, P))$

Alice's
identity

Needed to Bob to demonstrate he has received the message from the central authority (Alice cannot repudiate the message)

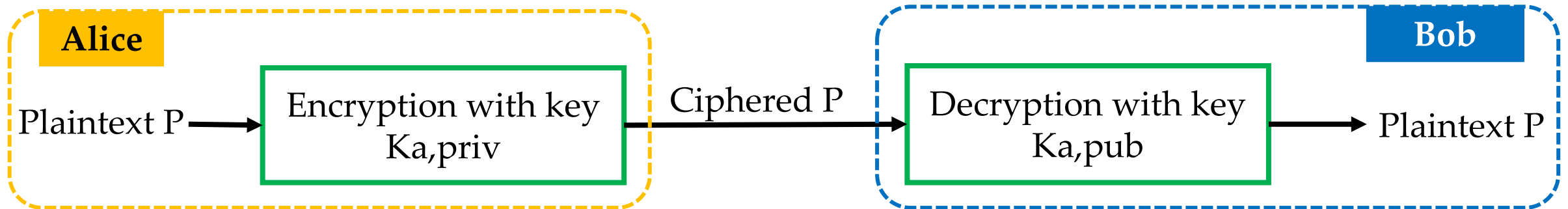
Limitations of the symmetric-key digital signature

- A central trusted authority must be in the middle of any exchange of signed messages.
- The central authority reads all the signed messages.

➡ This problem can be solved using public-key algorithms.

Public-key digital signature

- $K_{a, \text{pub}}$: Alice's public key
 - $K_{a, \text{priv}}$: Alice's private key
- ➔ If Alice encrypts a message with her private key ($K_{a, \text{priv}}$) then Bob can decrypt the message with her public key ($K_{a, \text{pub}}$).



If Bob can open the message by $K_{a, \text{pub}}$, then Bob is sure that the message was sent by Alice, because Alice is the only one having $K_{a, \text{priv}}$

➔ **authentication and non-repudiation are guaranteed**

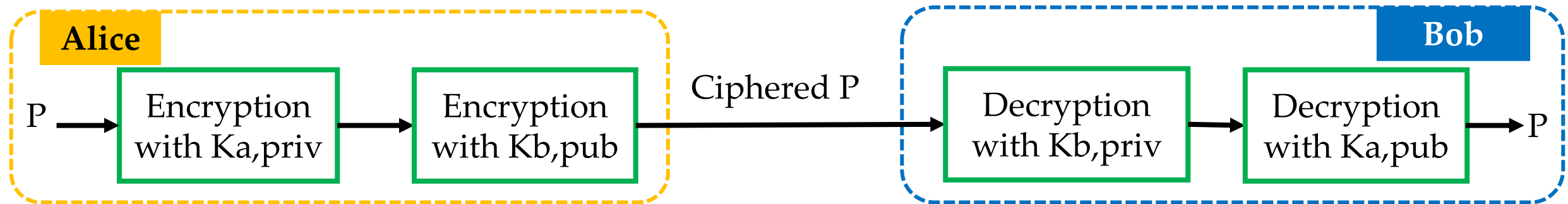
Secrecy is not guaranteed: anyone with $K_{a, \text{pub}}$ can open the message.

Public-key signature with secrecy

- If secrecy of the message must also be guaranteed, public-key cryptography can be applied twice, using both the keys of the sender and the receiver.

$K_{a, \text{pub}}$: Alice's public key
 $K_{a, \text{priv}}$: Alice's private key

$K_{b, \text{pub}}$: Bob's public key
 $K_{b, \text{priv}}$: Bob's private key



Message digest

- Critical aspect of public-key signature: public-key cryptography is slow.
- To make digital signature faster, instead of encrypting the entire message, the signature can be applied on a message digest obtained by a hash function.
- **Hash function:** a function $h(.)$ that takes an arbitrarily long piece of plaintext, P , and from it computes a fixed-length bit string, called **digest** or **fingerprint**.
- Properties that the hash function must have:
 - Given P , it is easy to compute $h(P)$.
 - Given $h(P)$, it is almost impossible to find P .
 - Given P , no one can find P' such that $h(P') = h(P)$.
 - A change to the input of even 1 bit produces a very different output.
- Computing $h(P)$ is much faster than encrypting P with a public-key algorithm
→ digests can be used to speed up digital signature algorithms.

Digital signature with message digest (no secrecy)

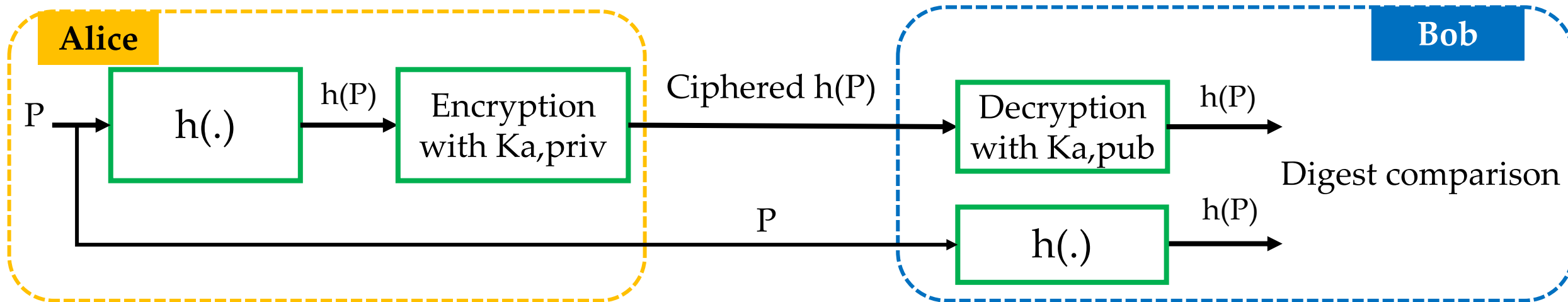
If message secrecy is not required:

$K_{a, pub}$: Alice's public key

$K_{a, priv}$: Alice's private key

$K_{b, pub}$: Bob's public key

$K_{b, priv}$: Bob's private key



If the two digest are the same Bob can be sure that the message was sent by Alice, because otherwise the decrypted hash would differ from the one obtained by Bob.

→ **Authentication** and **non-repudiation** ensured.

Bob is also sure that the message P was not altered during transmission. Any small modification of P would result in a different digest! → **Integrity** ensured.

Digital signature with message digest (with secrecy)

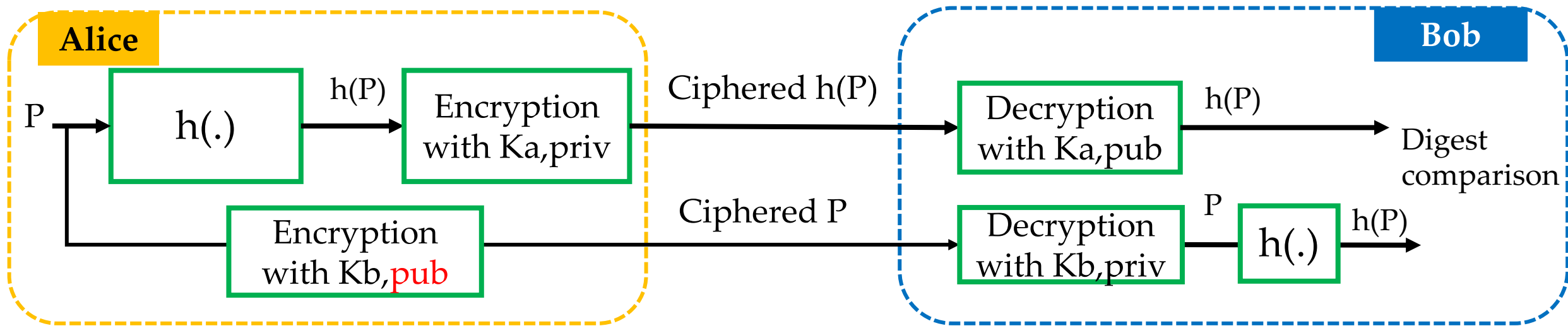
If message secrecy is required:

$K_{a, \text{pub}}$: Alice's public key

$K_{a, \text{priv}}$: Alice's private key

$K_{b, \text{pub}}$: Bob's public key

$K_{b, \text{priv}}$: Bob's private key



P must be encrypted, but the signature can still be done just on the digest because from the digest it is impossible to recover the original message.

Example of hash functions

➤ **SHA-1 (Secure Hash Algorithm 1)**

- it generates a 160-bit message digest

SHA-1(«hello world») → 2aae6c35c94fcfb415dbe95f408b9ce91ee846ed

SHA-1(«hello word») → e0738b87e67bbfc9c5b77556665064446430e81c

➤ **MD5 (Message Digest 5)**

- it generates a 128-bit message digest

MD5(«hello world») → 5eb63bbbe01eeed093cb22bb8f5acdc3

MD5(«hello word») → 13574ef0d58b50fab38ec841efe39df4

Security in network communication

- Main security techniques:
 - Cryptography algorithms to ensure secrecy
 - Digital signature to ensure authentication and non-repudiation
 - Hash functions to ensure integrity
- Security algorithms are implemented at different layers of the TCP/IP network model.
 - Network layer → IPsec
 - Transport-Application layer → SSL/TLS

IPsec (IP security)

- **IPsec (IP security):** secure network protocol suite to implement secure encrypted communication at the network layer.
- IPsec includes protocols for the following services:
 - network-level peer authentication
 - data origin authentication
 - data integrity
 - data confidentiality (encryption)
 - negotiation of cryptographic keys to use during the session
 - Protection against replay attacks (where the intruder replays a conversation)
- IPsec is an algorithm-independent framework that can implement multiple algorithms.
 - If an algorithm that is now thought to be secure may be broken in the future, the framework is still working.
- Multiple granularities: it is possible to protect a single TCP connection or all traffic between a pair of hosts.

Algorithms in IPsec

BONUS

Integrity algorithm:

- HMAC-SHA1/SHA2

Symmetric encryption algorithms for confidentiality:

- TripleDES
- Different versions of AES

Key exchange algorithms:

- Diffie–Hellman
- ECDH (Elliptic-curve Diffie–Hellman)

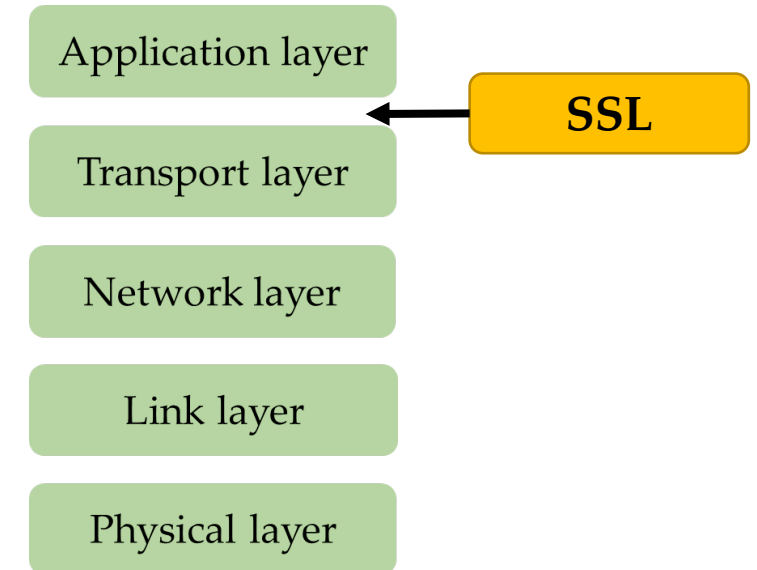
Authentication algorithms:

- RSA
- ECDSA (Elliptic Curve Digital Signature Algorithm)
- PSK (Pre-shared key)

Packets need to be exchanged extremely rapidly so IPsec is mainly based on symmetric encryption.

Secure socket layer (SSL) and Transport Layer Security (TLS)

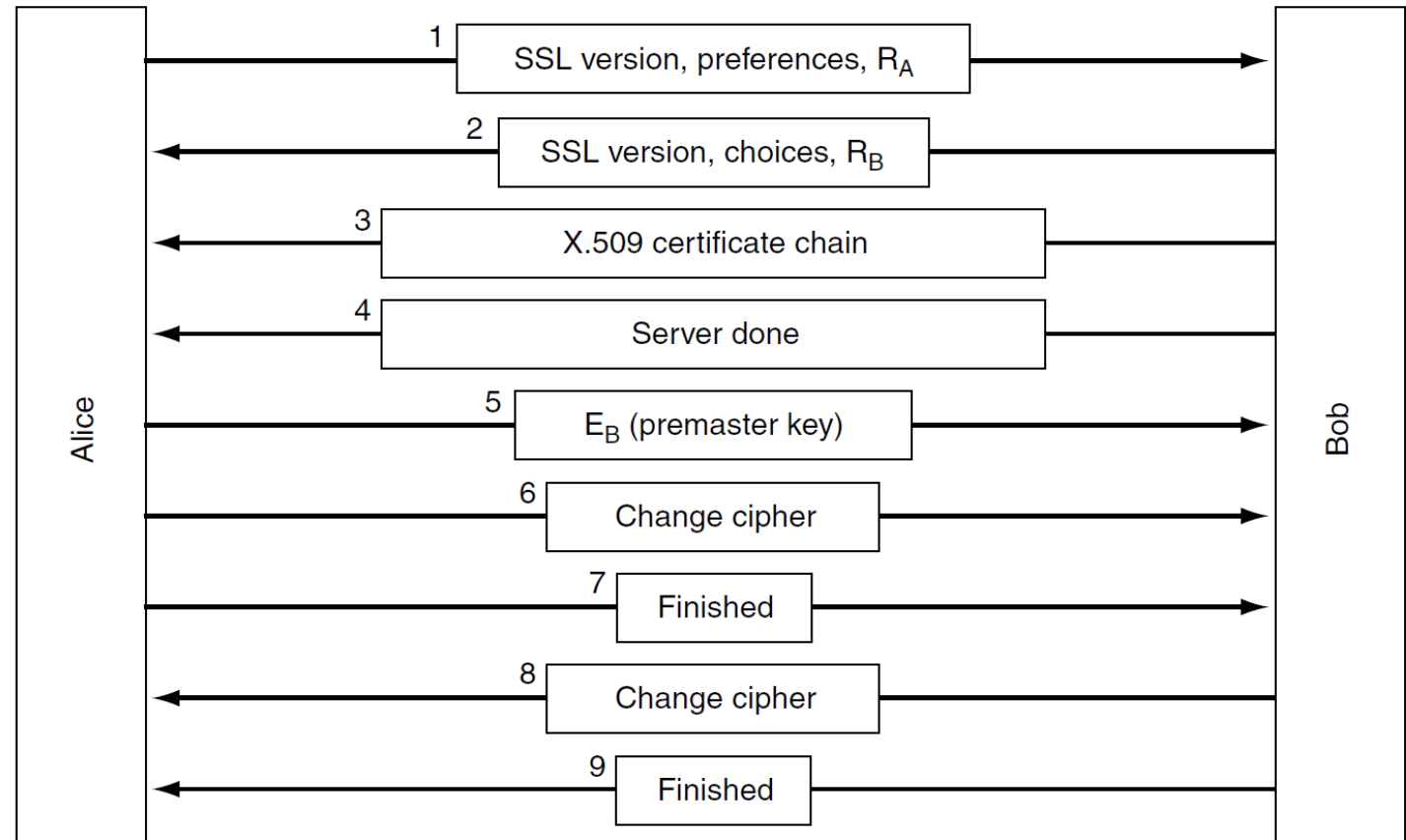
- **Secure Socket Layer (SSL):** A software for building a secure connection between two sockets, including:
 - Parameter negotiation between client and server.
 - Authentication of the server by the client.
 - Secret communication.
 - Data integrity protection.
- SSL is positioned between the application layer and the transport layer
- SSL consists of two subprotocols, one for establishing a secure connection and one for using it.
- When HTTP is used over SSL it is called HTTPS (secure HTTP), even though it is the standard HTTP.
- 2015: SSL was deprecated and substituted by the **Transport Layer Security (TLS)**.



SSL connection establishment protocol

BONUS

1. Alice expresses preferences about cryptography and compression algorithms to use.
2. Bob chooses among Alice preferences.
3. Bob provides the certificate with his public key.
4. Bob communicates his part is done.
5. Alice sends an encrypted premaster key. With the premaster key Alice and Bob are able to calculate their session key (similarly to Diffie-Hellman).
6. Alice tells Bob to switch to the ciphered communication.
7. Alice tells Bob the setup is done.
8. Bob tells Alice he will switch to ciphered communication.
9. Bob confirms that the setup is done.



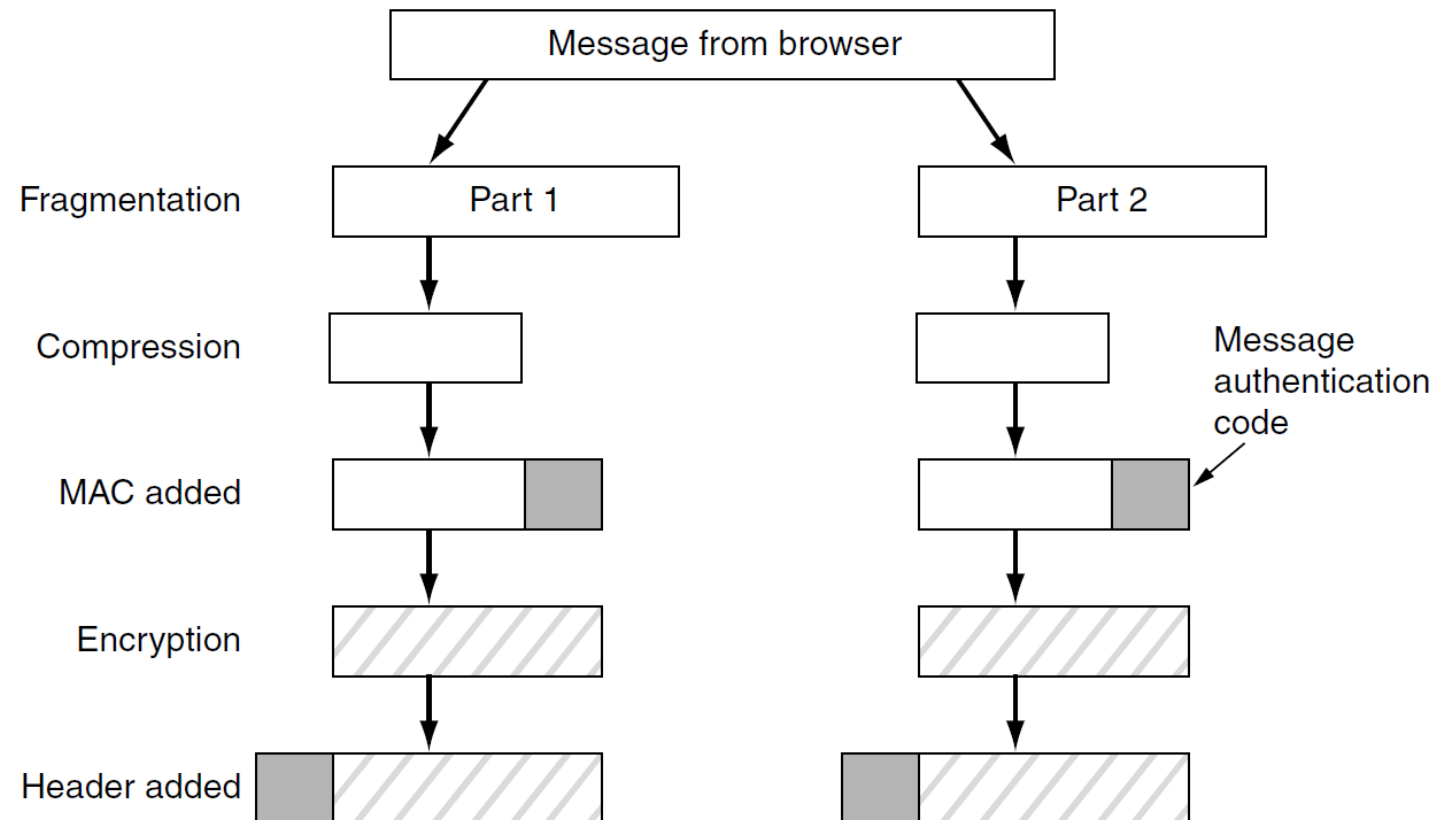
The actual session key used for encrypting data is derived from the premaster key combined with two large random numbers, R_A and R_B , in a complex way.

SSL data transmission protocol

BONUS

- SSL supports multiple cryptographic algorithms. The strongest one uses triple DES with three separate keys for encryption and SHA-1 for message integrity.

The session key is concatenated with the compressed text and the result is hashed with the agreed-on hashing algorithm. The resulting digest is the **Message Authentication Code (MAC)**.



References

- Tanenbaum, Wetherall – Computer Networks – Fifth Edition
 - Chapter 8 – Network security