Reti di Calcolatori

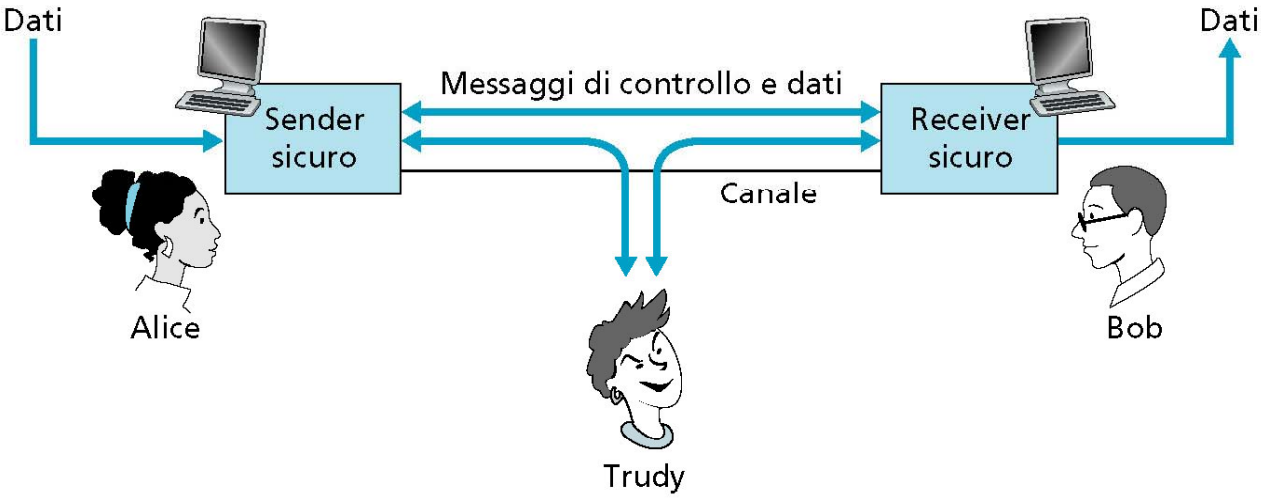
Elementi di Crittografia

Crittografia: Servizi richiesti

- **SEGRETEZZA**: evitare che i dati inviati da un soggetto A a un soggetto B vengano intercettati da un terzo soggetto C.
- **AUTENTICAZIONE**: verificare l'identità di chi manda o riceve i dati (evitare che un intruso si spacci per chi non è).
- **INTEGRITA'**: essere sicuri che i dati ricevuti siano uguali a quelli inviati (evitare cioè che un intruso o un evento accidentale alteri i dati durante la trasmissione).
- **NON RIPUDIO**: evitare che chi invia dei dati possa in futuro negare di averli inviati, e di aver inviato proprio *quei* dati (firma digitale).
- **DISPONIBILITA'**: un servizio deve essere accessibile a chi è legittimamente autorizzato (protezione da attacchi di tipo *denial of service*).

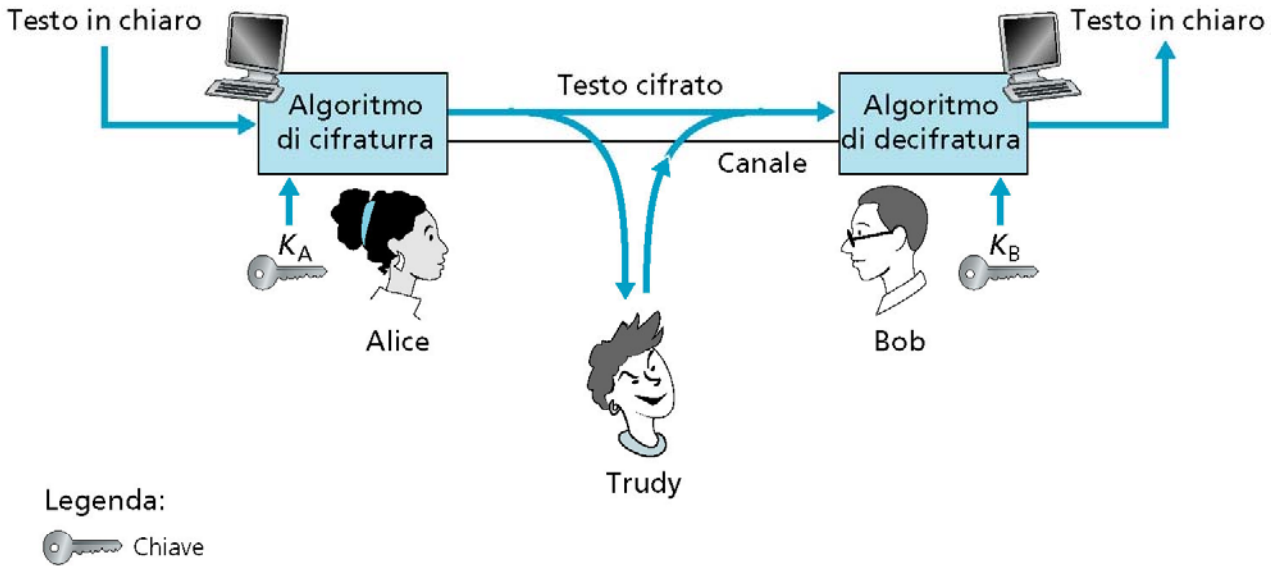
Problema della sicurezza in rete: scenario

Sender, receiver e intruso (Alice, Bob e Trudy)



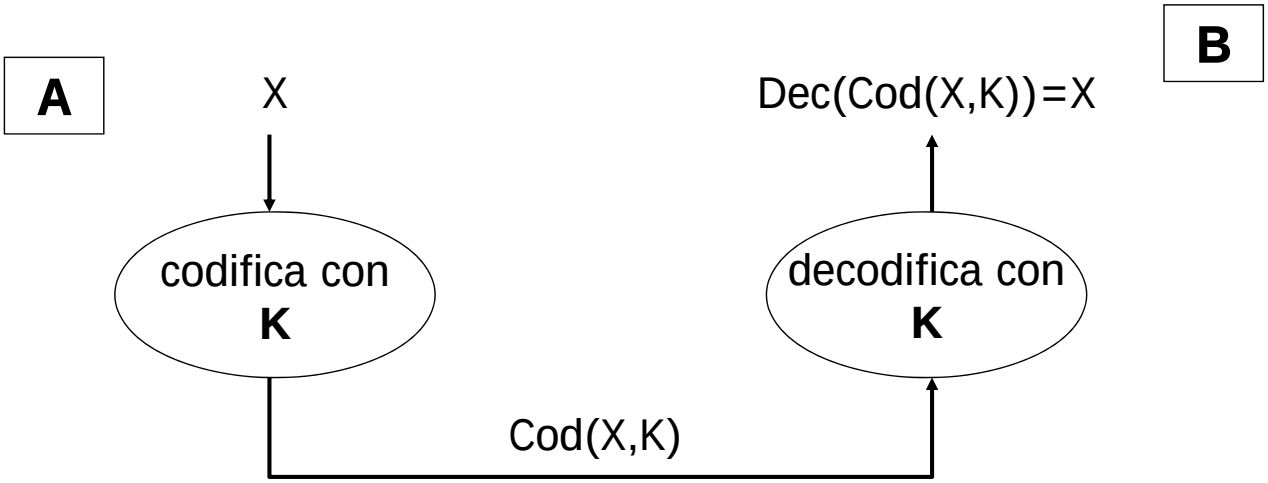
Problema della sicurezza in rete: scenario

Componenti della crittografia



Algoritmi a chiave segreta

- **Chiave Segreta**: i due soggetti (A e B) usano la stessa chiave K per codificare e decodificare i dati
- Gli algoritmi di crittografia sono pubblici, solo la chiave è segreta
- Il principale problema è lo scambio della chiave tra A e B



Algoritmi a chiave segreta

- **Cifrario di Cesare**
  - ogni lettera è sostituita da quella che la segue di **k** posizioni nell'alfabeto
  - **k** è la chiave
  - Se **k=3**, **a** è sostituito con **d**, **b** con **e** ecc.
- **Cifrario di sostituzione mono-alfabetico**:
  - generalizzazione del cifrario di Cesare
  - ogni lettera è sostituita da un'altra, secondo uno schema "libero"
  - facilmente attaccabile con metodi statistici (partendo dalla conoscenza della frequenza delle lettere e delle parole in una lingua)

Lettere in chiaro:	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Lettere cifrate:	m	n	b	v	c	x	z	a	s	d	f	g	h	j	k	l	p	o	i	u	y	t	r	e	w	q

Algoritmi a chiave segreta

- **Cifrario di sostituzione poli-alfabetico**:
  - sono usati due o più cifrari mono-alfabetici
  - le lettere sono sostituite con l'uno o l'altro dei cifrari, secondo uno schema predeterminato

Lettere in chiaro:	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
$C_1(k = 5)$:	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e
$C_2(k = 19)$:	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s

  - es. se lo schema è C1, C2, C2, C1, C2, il testo in chiaro "testo segreto" è crittografato nel testo cifrato "yxlyh xxzwxyh"
  - più lunga è la chiave, più robusto è l'algoritmo, sempre per ragioni statistiche

Algoritmi a chiave segreta

- **Cifrari di trasposizione**
  - le lettere sono trasposte in altre posizioni, secondo una chiave di trasposizione
  - ad esempio, la prima lettera è sostituita con la quarta, la seconda con la terza ecc.
  - anche questo algoritmo è facile da attaccare statisticamente
- **One-Time Pad**:
  - il testo è codificato in binario e viene effettuata un'operazione di **OR esclusivo** con una chiave binaria della stessa lunghezza
  - algoritmo inattaccabile, ma c'è bisogno di chiavi molto lunghe
- **Algoritmi moderni**:
  - chiavi piccole, utilizzo di più operazioni (di trasposizione, sostituzione, di XOR), in cascata, secondo logiche complesse

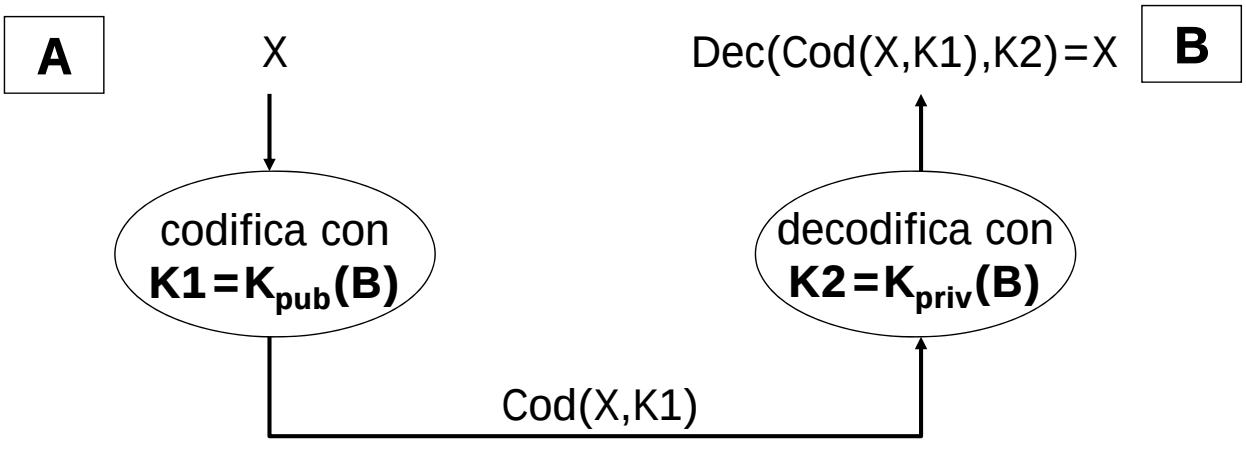
Algoritmi a chiave pubblica

- **Chiave Pubblica/Privata**: Ogni soggetto **S** ha
  - una propria *chiave pubblica* **$K_{pub}(S)$**, nota a tutti;
  - una propria *chiave privata* **$K_{priv}(S)$** nota solo a lui.
- I **requisiti** che un algoritmo a chiave pubblica deve soddisfare sono:
  - i dati codificati con una delle chiavi possono essere decodificati solo con l'altra
  - la chiave privata deve essere riservata, lo smarrimento deve essere denunciato come accade per un documento di identità
  - deve essere molto difficile ricavare una chiave dall'altra (in particolare la chiave privata da quella pubblica)

Algoritmi a chiave pubblica

Procedura per garantire la **segretezza** dei dati scambiati.

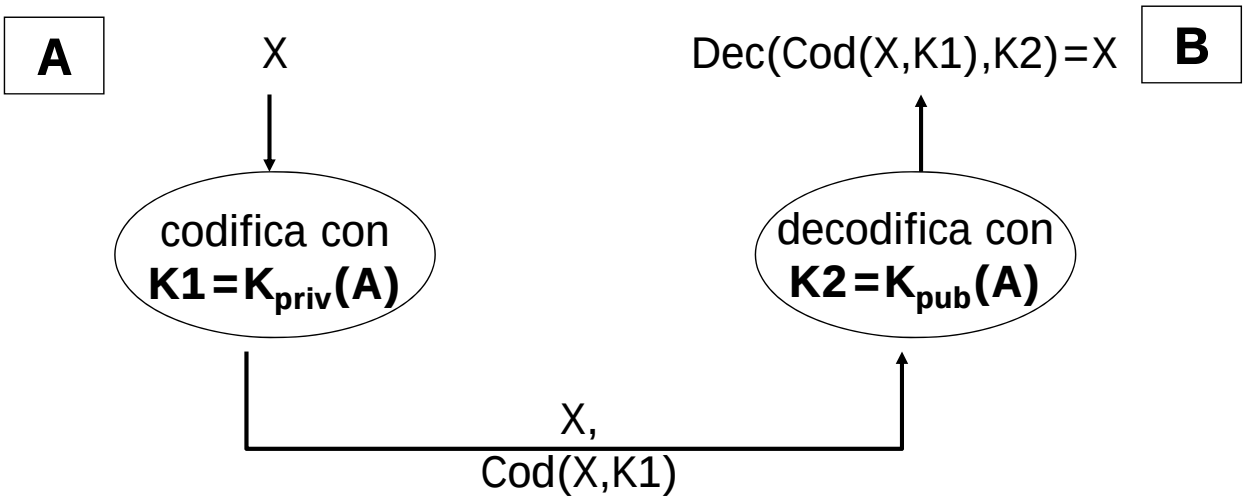
Solo B può decodificare i dati cifrati da A.



Algoritmi a chiave pubblica

Procedura per garantire **l'autenticità** ed il **non ripudio**.

B ha la garanzia che solo A può aver codificato il messaggio X e che in futuro non potrà negare di averlo fatto.



Algoritmi a chiave segreta e a chiave pubblica

Vantaggi degli algoritmi a chiave segreta

- Un vantaggio importante: questi algoritmi sono più veloci di ordini di grandezza rispetto agli algoritmi a chiave pubblica

Vantaggi degli algoritmi a chiave pubblica

- E' eliminato il problema dello scambio della chiave segreta
- Le chiavi pubbliche sono diffuse e accessibili da tutti
- Introduzione della firma digitale e dei certificati digitali (vedi dopo)

Uso congiunto dei due tipi di algoritmo

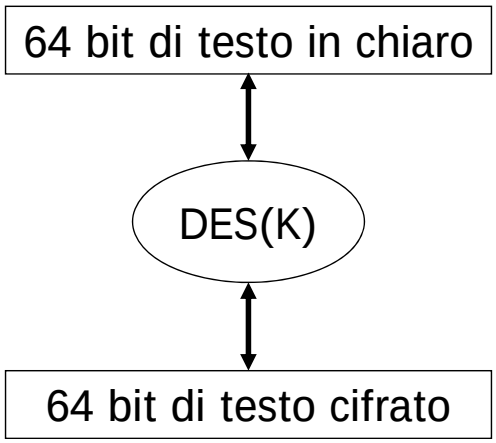
- Spesso si usa un algoritmo a chiave pubblica per scambiare la chiave segreta "di sessione" con cui poi crittografare i successivi dati

Crittografia a chiave simmetrica: DES

DES: *Data Encryption Standard*

- Standard codificato e aggiornato dall'U.S. National Bureau of Standards [NIST 1993]
- Codifica il testo in chiaro in blocchi di 64 bit; la lunghezza effettiva della chiave è di 56 bit
- Ma quanto è sicuro DES?
  - DES Challenge: nel 1997, durante un concorso, la frase "Strong cryptography makes the world a safer place" fu individuata in meno di 4 mesi
- Come rendere DES più sicuro:
  - usare sequenzialmente tre chiavi (3DES, triplo DES)
  - utilizzare il concatenamento dei blocchi cifrati

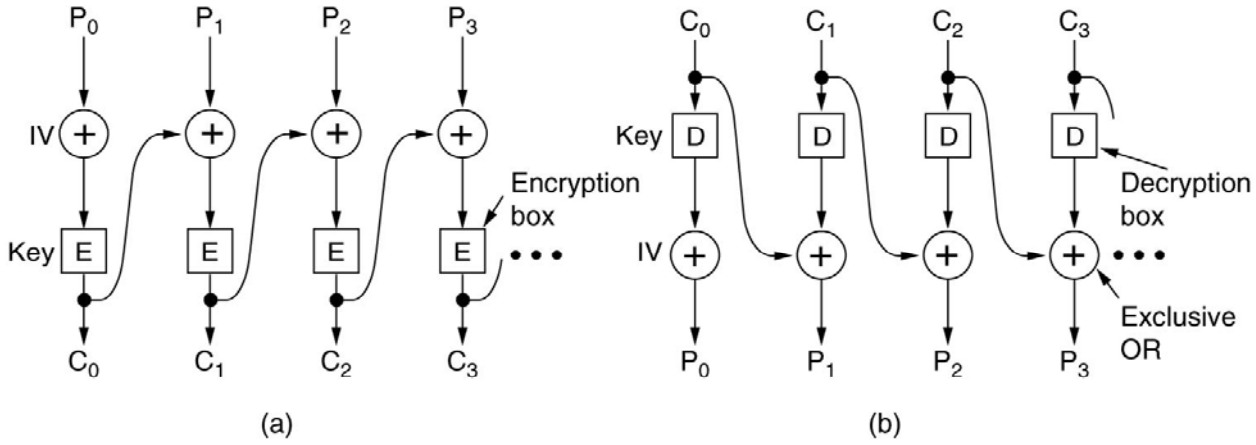
Procedura di base del DES



- K=chiave segreta di **56 bit**
- 19 stadi di codifica
- stesso algoritmo e stessa chiave per codifica e decodifica

Concatenamento dei blocchi cifrati

- codifica//decodifica di messaggi con il metodo del *Cipher Block Chaining* (*CBC*):
- suddivisione del messaggio in *blocchi* di 64 bit: P0, P1, ... , Pn
- ogni blocco, prima di essere codificato con la chiave del DES, è messo in XOR con il risultato della codifica del blocco precedente



Un algoritmo a chiave pubblica: RSA

RSA (Rivest-Shamir-Adleman)

- RSA garantisce l'impossibilità di ricavare la chiave privata da quella pubblica
- RSA sfrutta le proprietà dei numeri primi e la considerazione (consolidata da secoli di studi) che è molto difficile fattorizzare (cioè trovare i divisori di) numeri molto grandi
- Le chiavi possono essere lunghe 512, 1024, 2048 bit
- RSA è molto più lento degli algoritmi a chiave segreta (DES, IDEA) ma in genere è applicato a piccole quantità di dati, ad esempio per la trasmissione delle chiave di una sessione DES

RSA: costruzione delle chiavi

- Si prendono due numeri primi, p e q, di 1024 bit
- Si calcolano $n = p \times q$ e $z = (p-1) \times (q-1)$
- Si sceglie un numero d che sia primo rispetto a z (cioè che non abbia divisori comuni con z).
- Si trova e tale che $(e \times d) \pmod{z} = 1$
- La chiave pubblica è composta da e ed n
- La chiave segreta è composta da d ed n
- *La robustezza deriva dalla difficoltà di scomporre n nei suoi fattori, e quindi di ricavare p e q*
- *In altre parole, pur conoscendo n ed e, è impossibile da questi ricavare d*

RSA: cifratura, decifratura

0. Dati (n,e) e (n,d) calcolati come abbiamo appena visto,

1. Per la codifica di un messaggio m, si calcola:

   $c = m^e \bmod n$

2. Per decifrare il messaggio ricevuto, c, si calcola:

   $m = c^d \bmod n$

Si dimostra che: $m = \underbrace{(m^e \bmod n)}_{c}^{d} \bmod n$

Esempio di funzionamento di RSA

- Si scelgono $p = 3$ e $q = 11$.
- Quindi $n = 33$ e $z = 20$.
- Si può prendere $d = 7$, poiché 7 non ha divisori comuni con 20.
- Con l'equazione $(7 \times e) \pmod{20} = 1$, si trova $e = 3$
- Per cifrare un valore P ($P<33$), si calcola $C=P^3 \pmod{33}$
- Per decifrare C, si calcola $P=C^7 \pmod{33}$

Esempio di cifratura e decifratura

- Poiché $n = 33$, si può cifrare solo un carattere alla volta
- Se n è dell'ordine di 2^{1024}, si possono cifrare blocchi di 1024 bit, cioè 128 caratteri di 8 bit ciascuno

Plaintext (P)			Ciphertext (C)		After decryption	
Symbolic	Numeric	P^3	$P^3 \pmod{33}$	C^7	$C^7 \pmod{33}$	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		

Hashing

E' un metodo che serve a garantire **l'integrità** dei dati:

- L'**hashing** è una funzione che mappa un messaggio **M** (sequenza di bit) in un'altra sequenza di bit **h(M)** più corta (detta **impronta**), con la proprietà che, dati due messaggi $M \neq M'$

  Prob(h(M)=h(M')) $\rightarrow$ 0

- È in pratica impossibile costruire un altro messaggio M' che abbia la stessa impronta di M.
- È quindi facile verificare l'integrità di un messaggio.
- Algoritmi più usati: **MD5**, **SHA**

Firma digitale

- Firma di un documento M da parte di un soggetto S

  M $\rightarrow$ Cod(M, K_{priv}(S))

- Verifica di autenticità di un documento M firmato da un soggetto S

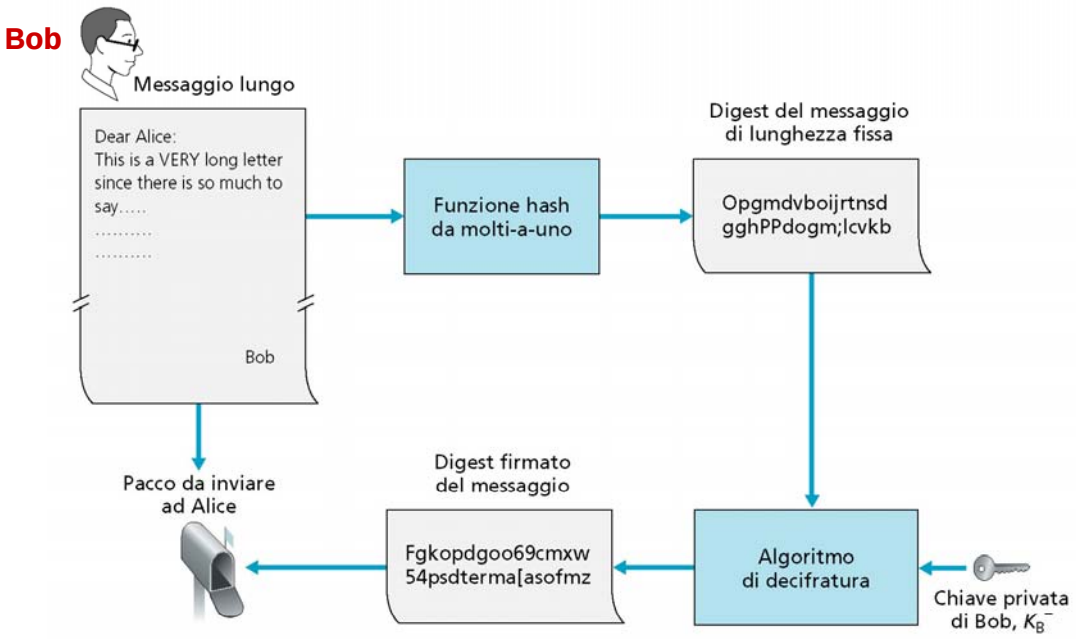
  M = Dec(Cod(M, K_{priv}(S)), K_{pub}(S))

- Per ragioni di velocità e di compattezza dei dati, speso la firma digitale consiste nell'applicazione della chiave privata non al messaggio intero, ma ad **un'impronta hash** (o **digest**) del messaggio.

Considerazioni sulla firma digitale

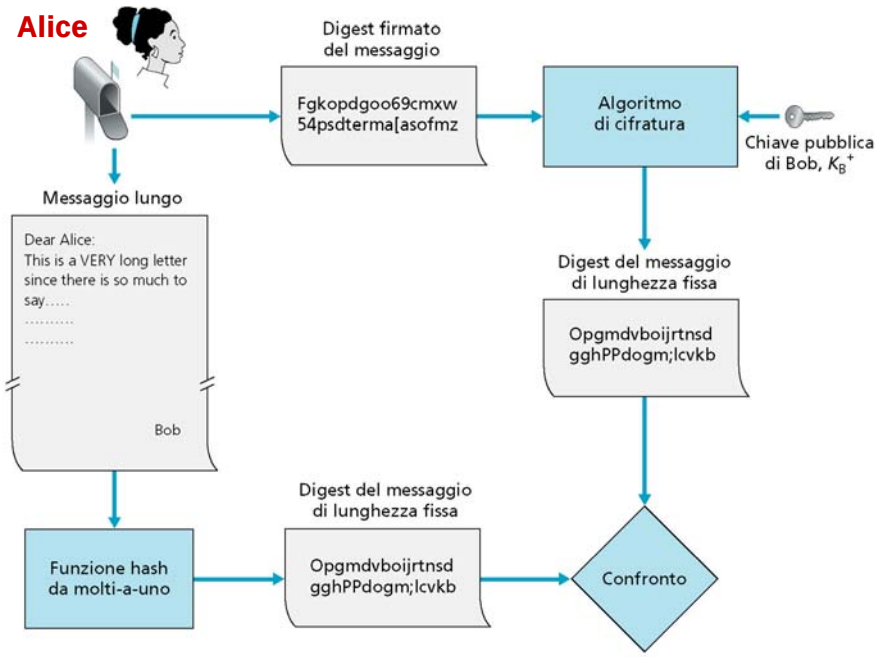
- In genere, quando si appone una firma digitale, non c'è la necessità di garantire la segretezza di un documento, ma bisogna piuttosto assicurarsi che:
  1. il firmatario si **autentichi**
  2. il firmatario **non** possa "**ripudiare**" la sua firma
  3. il documento **firmato** sia integro
- I primi 2 requisiti sono garantiti dalla codifica a chiave pubblica, il secondo dall'uso della funzione di hash
- La normativa attuale stabilisce che un **documento elettronico** firmato con firma digitale abbia la stessa validità di un documento cartaceo, purché alla firma si alleghi il **certificato** che attesta l'autenticità di chi ha apposto la firma (vedi dopo).

- Bob crea il messaggio, ne calcola l'**impronta** (**digest**), **firma** l'impronta con la sua **chiave privata**, ed invia ad Alice **sia** il documento originale, **sia** il digest firmato



Ricezione e verifica di un messaggio firmato

- Alice calcola l'impronta del messaggio originale, e **confronta** tale impronta con il documento ottenuto applicando la chiave **pubblica** di Bob al digest firmato
- Se i due risultati sono identici, il documento originale è integro, ed è stato sicuramente firmato da Bob



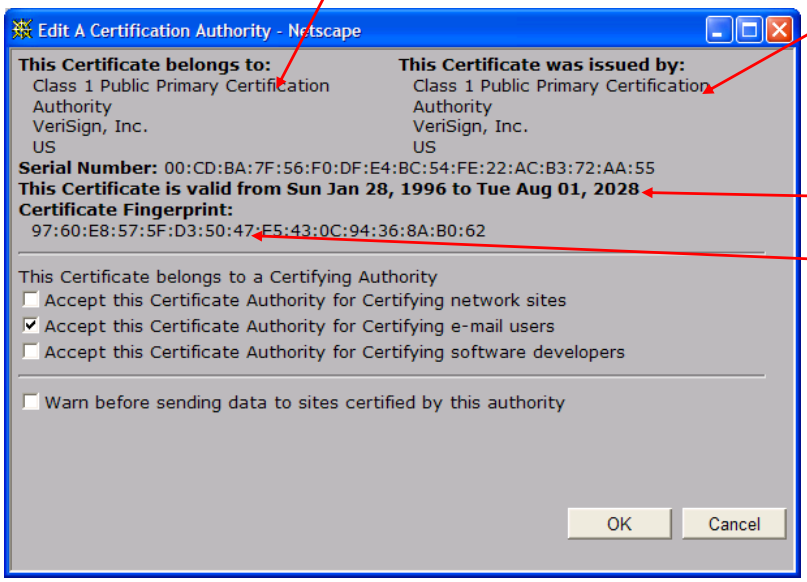
Certificati digitali

- Il **certificato digitale** è un documento emesso da una c**ertification authority (CA)** che associa una chiave pubblica ad un soggetto e si fa garante di questa associazione. Il certificato è firmato (con firma digitale) dalla CA.
- La chiave pubblica della CA è conosciuta e può essere a sua volta certificata da un'altra CA.

  ⇒ Le CA sono organizzate in una gerarchia: ogni CA deve essere certificata da una di livello superiore
- I certificati sono strutturati secondo il modello standard detto **X.509**. In particolare il certificato deve contenere il periodo di validità della chiave pubblica.

Un certificato contiene:

- Numero di serie
- Informazioni sul titolare, compreso il valore della chiave pubblica (non inserita)



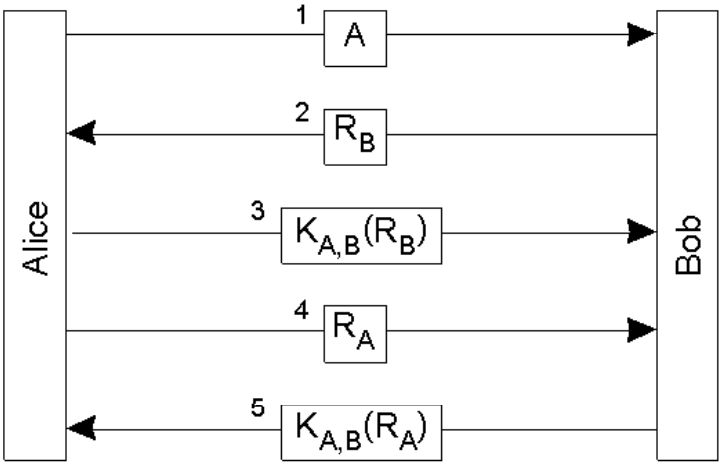
- Informazioni su chi ha emesso il certificato
- Date valide
- Firma digitale di chi lo ha emesso

Protocolli di Autenticazione

- Protocolli di autenticazione basati su:
  - Cifratura a chiave **segreta**
  - Cifratura a chiave **pubblica**
- Scambio di chiavi di sessione

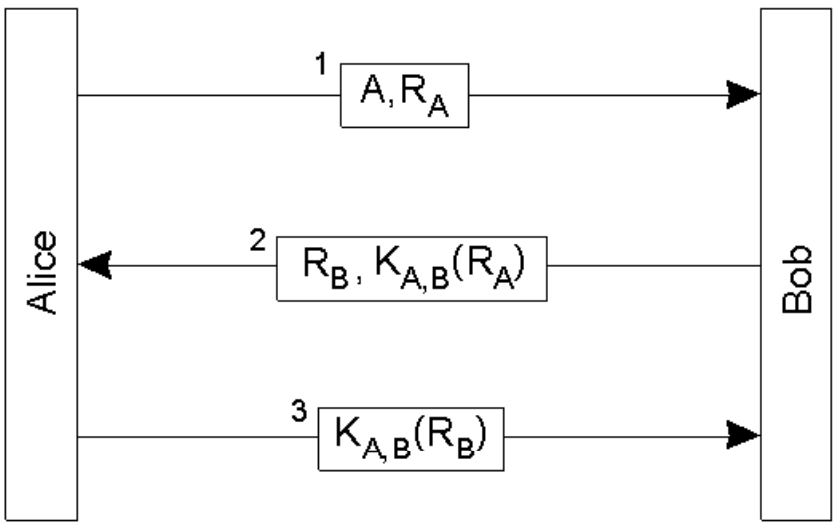
Autenticazione con chiave segreta

- Autenticazione basata su una chiave segreta condivisa
- $K_{A,B}$ è la chiave segreta conosciuta solo da A e da B
- R_B è un “challenge”, cioè un numero casuale che B invia ad A e che A deve cifrare con la chiave segreta



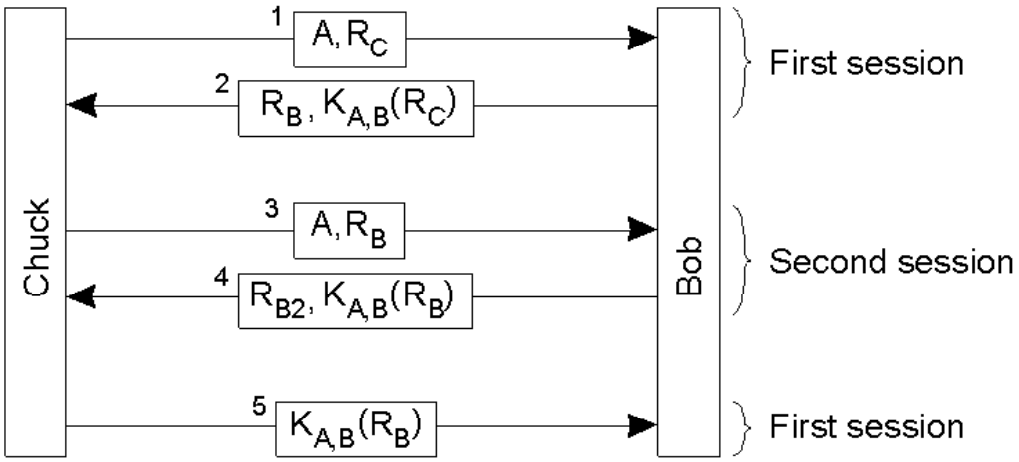
Autenticazione non corretta

- Autenticazione basata su chiave segreta condivisa, utilizzando solo 3 messaggi, invece di 5.
- Questo protocollo è attaccabile con il “reflection attack”



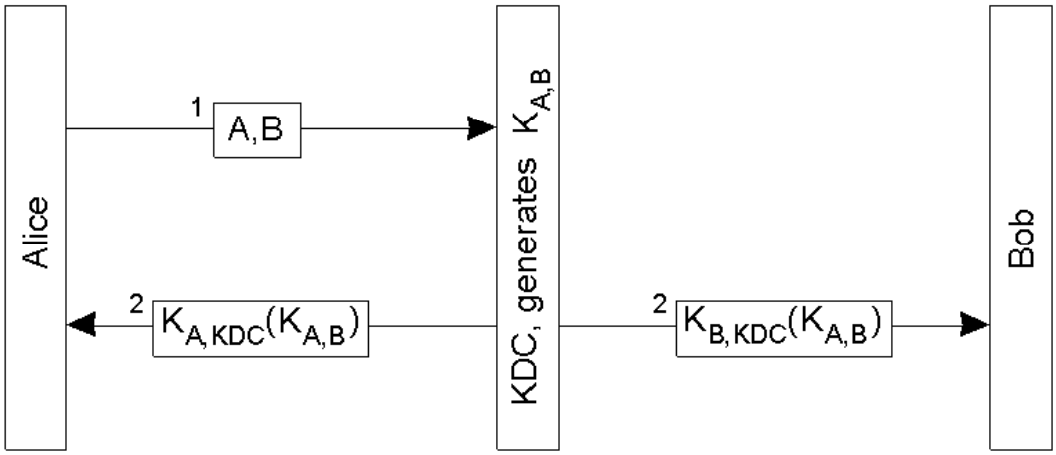
Autenticazione non corretta: possibile attacco

- **Reflection attack**: Chuck fa credere a Bob di conoscere la chiave segreta $K_{A,B}$
- Bob crede che Chuck sia Alice!



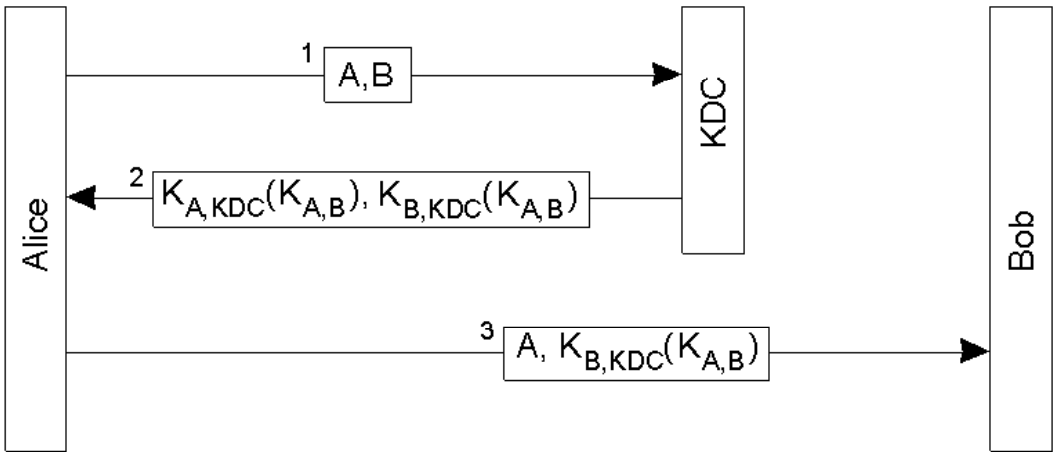
Scambio della chiave segreta

- Uso di un **Key Distribution Center (KDC)**
- Sia A che B condividono con KDC delle chiavi segrete
- Il KDC genera una chiave di sessione $K_{A,B}$ e la invia ad Alice e Bob, cifrandola con le rispettive chiavi segrete



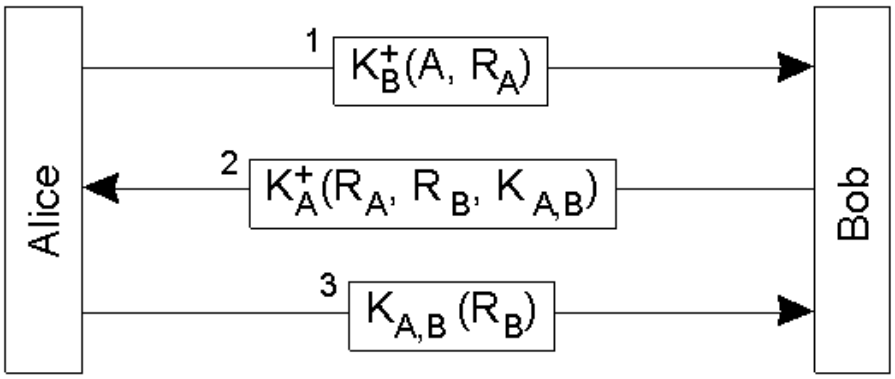
Scambio della chiave segreta con ticket

- KDC invia ad Alice un **ticket**, che poi Alice si incarica di inviare a Bob



Autenticazione con crittografia a chiave pubblica

- Mutua autenticazione con la codifica a chiave pubblica
- Scambio contestuale della chiave di sessione



Secure Socket Layer (SSL)

Protocollo inserito tra il livello trasporto ed il livello delle applicazioni che aggiunge la sicurezza della trasmissione ai servizi offerti dal livello trasporto.

HTTPS (Secure HTTP): protocollo HTTP + uso di SSL.
Porta TCP: 443 invece che 80.

- Nel seguito **S** = Server, **C** = Client
- S invia **$K_{pub}(S)$** a C insieme a un **certificato** che garantisce che $K_{pub}(S)$ è proprio la chiave pubblica di S
- Il certificato è emesso da una CA ritenuta affidabile sia da S sia da C
  - *Se richiesto, anche C invia il proprio certificato ad S*
- C verifica l'autenticità di $K_{pub}(S)$, ed invia a S una chiave segreta **K** da utilizzare durante la transazione (ad es. una chiave di sessione DES). Il messaggio contenente K è codificato con **$K_{pub}(S)$**, in modo che solo S possa decodificarlo
- C e S scambiano messaggi cifrati usando l'algoritmo DES con chiave K