Crittografia

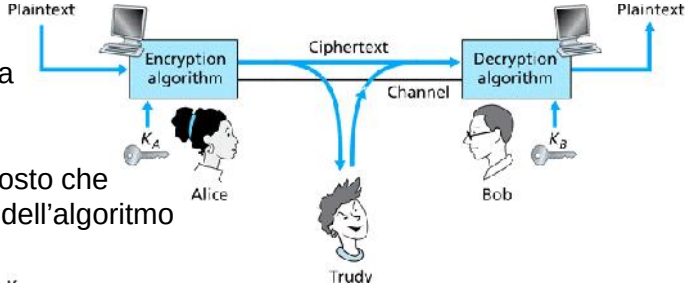
Principi di crittografia
Integrità dei messaggi
Protocolli di autenticazione
Sicurezza nella pila di protocolli di Internet: PGP, SSL, IPSec

Prof. Filippo Lanubile

Elementi di crittografia

- **Crittografia:** procedimento di cifratura e decifratura dei messaggi basato su funzioni parametriche
- **Testo in chiaro**: messaggio originario
- **Algoritmo di cifratura**: effettua sostituzioni e trasformazioni sul testo in chiaro
- **Testo cifrato**: output dell'algoritmo di cifratura da trasmettere sul canale; risulta inintelligibile a un intruso
- **Algoritmo di decifratura**: effettua il lavoro inverso dell'algoritmo di cifratura
- **Chiave**: parametro dell'algoritmo di cifratura o decifratura

La sicurezza della cifratura dipende dalla segretezza della chiave piuttosto che dalla segretezza dell'algoritmo



Classificazione dei sistemi crittografici

- Tipo di operazioni usate per trasformare il testo in chiaro in testo cifrato
  - Sostituzione:
    - Ogni elemento del testo in chiaro è trasformato in un altro elemento
  - Trasposizione:
    - Gli elementi del testo in chiaro sono riorganizzati
- Numero di chiavi (distinte) utilizzate
  - Chiave singola: crittografia a chiave simmetrica (o a chiave segreta)
    - Le chiavi del mittente e del destinatario sono identiche
  - Due chiavi: crittografia a chiave asimmetrica (o a chiave pubblica)
    - La chiave di cifratura è pubblica; la chiave di decifratura è privata
- Il modo in cui il testo in chiaro è elaborato
  - Cifrario a blocchi:
    - Elabora in blocchi di dimensione fissa
  - Cifrario a flusso:
    - Elabora senza una lunghezza predefinita

Esempio: cifrario di Cesare

- Tipo di operazioni:
  - sostituzione
- Numero di chiavi utilizzate:
  - Chiave singola ($1 \leq k \leq 26$)
- Modo in cui il testo in chiaro è elaborato:
  - cifrario a flusso

Plaintext letter:	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

Esempio: cifrario monoalfabetico

- Tipo di operazioni:
  - sostituzione
- Numero di chiavi utilizzate:
  - Chiave singola (26! pattern di sostituzione monoalfabetico)
- Modo in cui il testo in chiaro è elaborato:
  - cifrario a flusso

Plaintext letter:	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
Ciphertext letter:	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

Figure 8.3 ♦ A monoalphabetic cipher

Crittoanalisi

- Processo con cui si tenta di risalire al testo in chiaro o alla chiave usata
  - Diversi attacchi in base alle informazioni a disposizione dell'intruso
- Un algoritmo di cifratura è progettato per resistere a un attacco basato su testo in chiaro conosciuto
  - Per scoprire la chiave occorre provare tutte le chiavi possibili (**attacco a forza bruta**)

Un sistema di cifratura è **computazionalmente sicuro** se il testo cifrato soddisfa uno dei seguenti criteri:

- Il costo per rendere inefficace il cifrario supera il valore dell'informazione cifrata
- Il tempo richiesto per rendere inefficace il cifrario supera l'arco temporale in cui l'informazione è utile

Key Size (bits)	Number of Alternative Keys	Time required at 10^6 Decryption/μs
32	$2^{32} = 4.3 \times 10^9$	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	10 hours
128	$2^{128} = 3.4 \times 10^{38}$	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	5.9×10^{30} years

Crittografia a chiave simmetrica

- Mittente e destinatario condividono una chiave segreta
  - Come si concorda la chiave?
    (problema della distribuzione delle chiavi)

Crittografia a chiave simmetrica: DES

DES: Data Encryption Standard

- Primo standard NIST
- Molto diffuso
  - Attacchi noti solo a forza bruta
- Blocchi di 64 bit
- Chiave di 56 bit
- Operazioni base
  - Permutazione iniziale
  - 16 iterazioni intermedie identiche
  - Permutazione finale

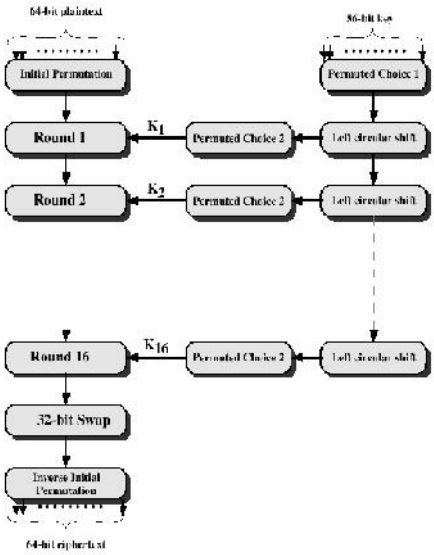


Figure 2.3 General Depiction of DES Encryption Algorithm

Crittografia a chiave simmetrica: 3DES

Triple DES (o TDEA)

- Usa tre chiavi e tre esecuzioni di DES (cifra-decifra-cifra)
- Lunghezza effettiva della chiave: 168 (=3x56) bit
- Modalità *Cipher block chaining* (cifratura a blocchi concatenati)
  - Operazione di XOR sull'i-esimo blocco in ingresso con il precedente blocco di testo cifrato

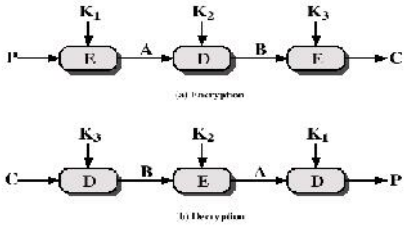


Figure 2.6 Triple DEA

$C = E_{K3}[D_{K2}[E_{K1}[P]]]$

C = ciphertext
P = Plaintext
EK[X] = encryption of X using key K
DK[Y] = decryption of Y using key K

Crittografia a chiave simmetrica: AES

AES: Advanced Encryption Standard
(Algoritmo di Rijndael)

- Scelto da NIST come sostituto di 3DES per migliorare in modo significativo l'efficienza
- Blocchi a 128 bit
- Chiavi a 128, 192 e 256 bit

Crittografia a chiave pubblica

- Le chiavi di cifratura e decifratura sono diverse
  - Mittente e destinatario non condividono una chiave segreta
- La chiave di cifratura **pubblica** è nota a tutti e quindi anche al mittente
- La chiave di cifratura **privata** è nota solo al destinatario

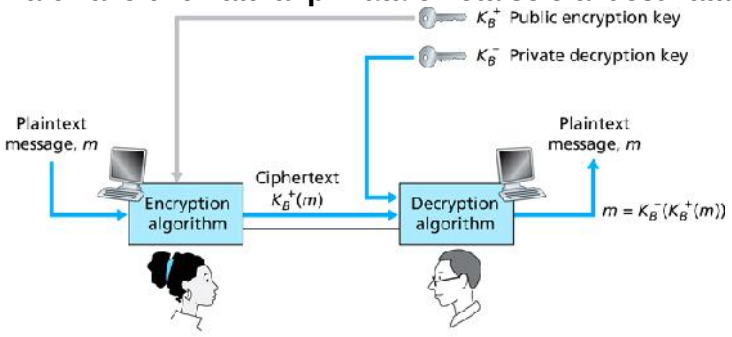


Figure 8.6 ♦ Public key cryptography

Requisiti della crittografia a chiave pubblica

- E' computazionalmente facile generare una coppia di chiavi (chiave pubblica K_B^+ , chiave privata K_B^-), cifrare e decifrare un messaggio tale che $K_B^-(K_B^+(m)) = m$
- Data K_B^+ , deve essere computazionalmente improponibile determinare K_B^-
- Data K_B^+ e il testo cifrato c, deve essere computazionalmente improponibile ricostruire il messaggio originario m
- Ciascuna delle due chiavi può essere utilizzata per la cifratura, usando l'altra per la decifratura: $K_B^-(K_B^+(m)) = m = K_B^+(K_B^-(m))$

Crittografia a chiave pubblica: RSA

- L'algoritmo più diffuso
- Inventato da Ron Rivest, Adi Shamir and Len Adleman al MIT nel 1977
- Si basa sul fatto che la scomposizione in fattori di numeri molto grandi è computazionalmente molto onerosa
- Resistente alla criptoanalisi con chiavi di almeno 1024 bit

RSA: scelta delle chiavi

1. Scegliere due numeri primi di valore elevato: *p, q*
   - Si raccomanda che pq sia di 1024 bit
2. Calcolare *n* = pq, *z* = (p-1)(q-1)
   - z è la funzione di Eulero di n
3. Scegliere *e* (con e < n) tale che non abbia fattori in comune con z
   - e, z sono "relativamente primi"
4. Scegliere *d* tale che ed-1 sia esattamente divisibile per z
   - ovvero: ed mod z = 1
5. K_B^+ = (e, n) ; K_B^- = (d, n)

RSA: cifratura e decifratura

Cifratura

Testo in chiaro: m < n

$K_B^+ = (e, n)$

$$c = m^e \bmod n$$

Decifratura

Testo cifrato: c

$K_B^- = (d, n)$

$$m = c^d \bmod n$$

Si dimostra che $m = (m^e \bmod n)^d \bmod n$

Dimostrazione: $m = (m^e \bmod n)^d \bmod n$

Teorema della teoria dei numeri:
se p e q sono primi e n = pq, allora:

$$x^y \bmod n = x^{y \bmod (p-1)(q-1)} \bmod n$$

$$(m^e \bmod n)^d \bmod n = m^{ed} \bmod n$$

$$= m^{ed \bmod (p-1)(q-1)} \bmod n$$

(usando il teorema sopra)

$$= m^1 \bmod n$$

(perché abbiamo scelto che e e d siano divisibili per (p-1)(q-1) con resto 1)

$$= m$$

RSA: esempio (1)

1. Scegliere due numeri primi di valore elevato: *p, q*
   - **Bob sceglie p=5, q=7**
2. Calcolare *n* = p x q, *z* = (p-1)(q-1)
   - **n=35, z=24**
3. Scegliere *e* (con e < n) tale che non abbia fattori in comune con z
   - **e=5**
4. Scegliere *d* tale che ed-1 sia esattamente divisibile per z
   - **d=29**
5. K_B^+ = (e, n) ; K_B^- = (d, n)
   - **K_B^+ = (5, 35) ; K_B^- = (29, 35)**

RSA: esempio (2)

	lettera	m	m^e	$c = m^e \bmod n$
Cifratura:	l	12	248832	17

	c	c^d	$m = c^d \bmod n$	lettera
Decifratura:	17	48196857210675091509141182522307169 7	12	l

Chiavi di sessione

- La crittografia a chiave pubblica è troppo costosa per cifrare testi lunghi
- E' in genere utilizzata per cifrare chiavi di sessione K_S da utilizzare per la crittografia a chiave simmetrica

$$c = (K_S)^e \bmod n$$

Crittografia

Principi di crittografia
Integrità dei messaggi
Protocolli di autenticazione
Sicurezza nella pila di protocolli di Internet: PGP, SSL, IPSec

Funzione hash crittografica

- Utilizzata per la verifica dell'integrità di un messaggio
- Dato un messaggio in input *m* di lunghezza variabile produce una stringa di lunghezza fissa *h = H(m)*
  - *h* è detta *hash value* o *message digest*
- H è sicura se:
  - è libera da collisioni
    - è computazionalmente impossibile trovare due messaggi x e y tali che H(x) = H(y)
  - è unidirezionale
    - dato h = H(x), (con x sconosciuto), è impossibile determinare x

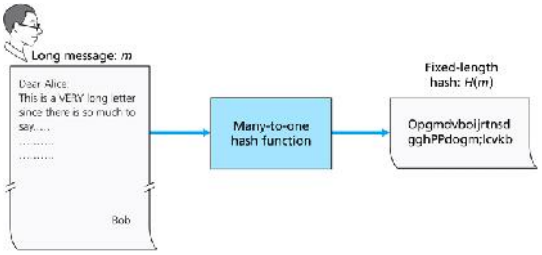


Figure 8.7 ♦ Hash functions

Una funzione hash non crittografica: Internet checksum

E' una funzione hash:

- Crea sintesi di lunghezza fissa (16 bit)
- È molti-a-uno

ma non è sicura

- Non è difficile trovare altri messaggi che utilizzano la stessa checksum del messaggio originale

Message	ASCII Representation	
I O U 1	49 4F 55 31	
0 0 . 9	30 30 2E 39	
9 B O B	39 42 4F 42	
	B2 C1 D2 AC	Checksum

Message	ASCII Representation	
I O U 9	49 4F 55 39	
0 0 . 1	30 30 2E 31	
9 B O B	39 42 4F 42	
	B2 C1 D2 AC	Checksum

Figure 8.8 ♦ Initial message and fraudulent message have the same checksum!

Confronto di funzioni hash crittografiche

MD5

- Ideato da Ron Rivest
- Standard IETF (RFC 1321)
- Blocchi di 512 bit
- Digest di 128 bit
- Processo a 4 fasi di 16 passi ciascuna
- Ormai vulnerabile alla crittoanalisi

SHA-1

Secure hash algorithm

- Standard americano: NIST - FIPS PUB 180-1
- Blocchi di 512 bit
- Digest di 160 bit
- Processo a 4 fasi di 20 passi ciascuna
- Ciascun bit del digest è funzione di ogni bit del messaggio in ingresso
  - Per ottenere due messaggi con lo stesso digest: 2^{80} operazioni

Codice di autenticazione dei messaggi: MAC

E' un valore da allegare al messaggio per verificare contemporaneamente l'autenticità e l'integrità del messaggio

- *s* è la chiave di autenticazione
- *H(m+s)* è il codice di autenticazione del messaggio (MAC)

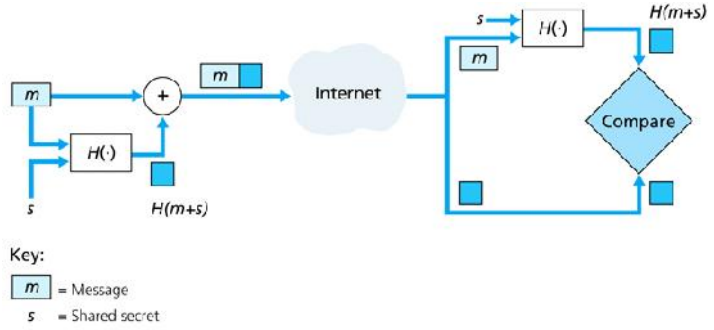
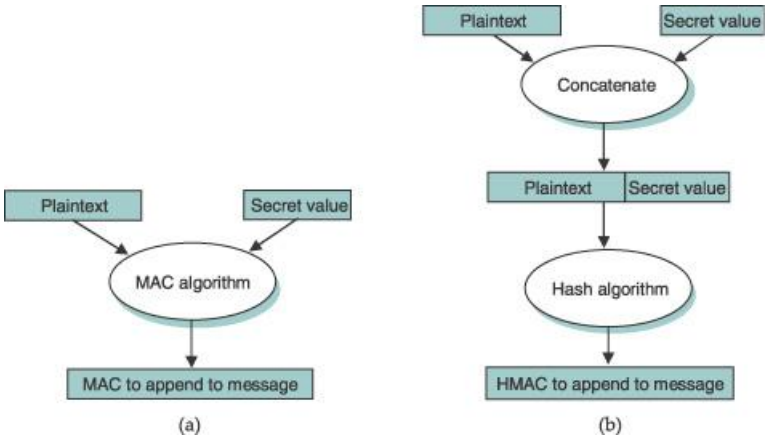


Figure 8.9 ♦ Message authentication code (MAC)

Codice di autenticazione dei messaggi: HMAC

- Variante del MAC con aggiunta di una funzione di hash crittografica
- Utilizzabile anche in combinazione con MD5 e SHA-1



Firma digitale

- Il mittente firma un messaggio con la propria chiave privata applicando un algoritmo di cifratura a chiave pubblica
- La firma è verificabile e non falsificabile:
  - il destinatario può dimostrare che Bob e nessun altro può aver firmato il documento
- Non ripudio: combinazione di integrità e autenticità

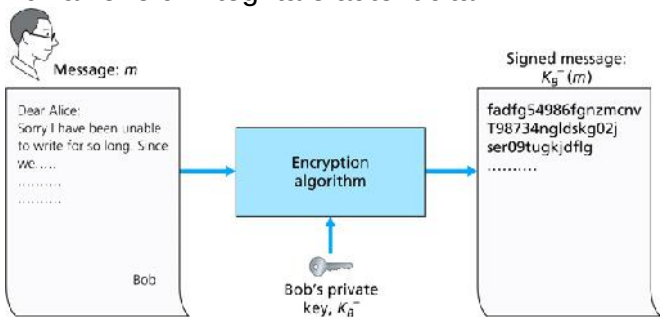


Figure 8.10 ♦ Creating a digital signature for a document

Firma digitale efficiente: spedizione

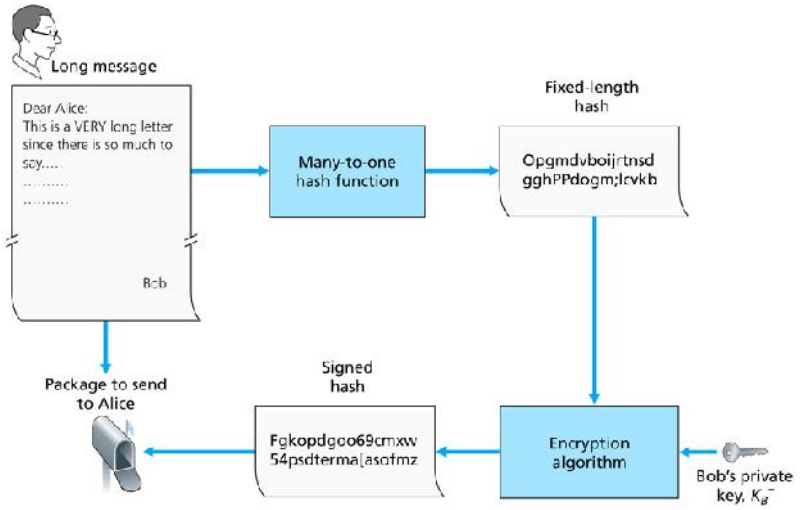


Figure 8.11 ♦ Sending a digitally signed message

Firma digitale efficiente: verifica in ricezione

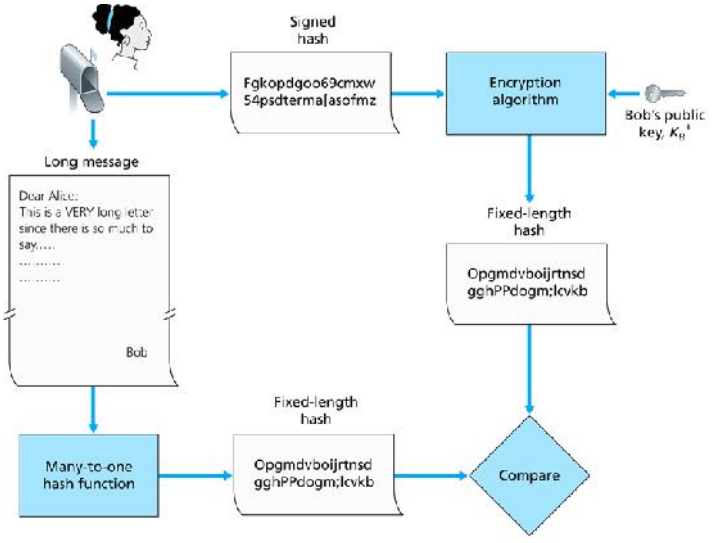


Figure 8.12 ♦ Verifying a signed message

Firma digitale: come fidarsi della chiave pubblica?

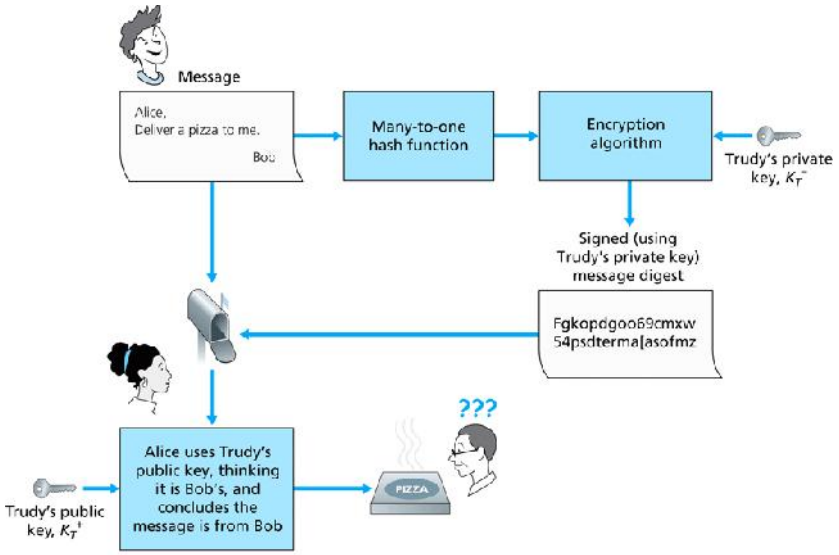
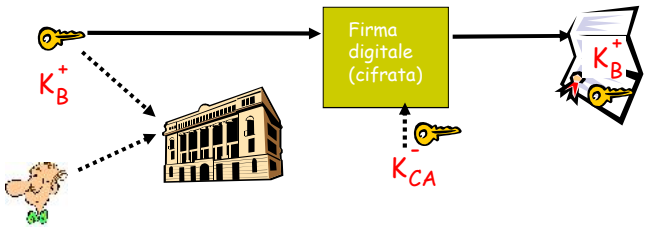


Figure 8.13 ♦ Trudy masquerades as Bob using public key cryptography.

Autorità di certificazione (1)

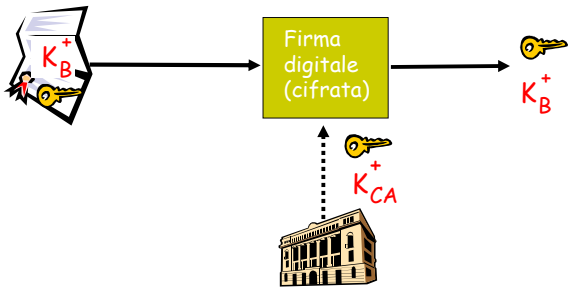
Certification authority (CA)

- Valida l'identità ed emette certificati
- Certificazione
  - Bob fornisce una prova d'identità alla CA
  - La CA crea un certificato che contiene la chiave pubblica di Bob con firma digitale della CA

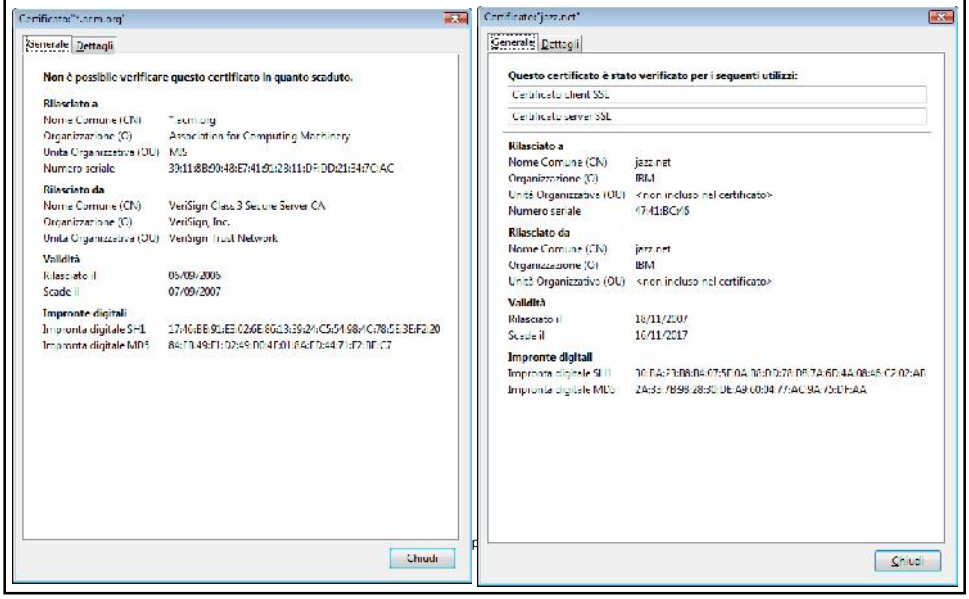


Autorità di certificazione (2)

- Verifica
  - Alice riceve il certificato di Bob
  - Alice applica la chiave pubblica della CA al certificato di Bob e ottiene la chiave pubblica di Bob



Certificati



Crittografia

Principi di crittografia
Integrità dei messaggi
Protocolli di autenticazione
Sicurezza nella pila di protocolli di Internet: PGP, SSL, IPSec

Protocolli di autenticazione (0)

- I sistemi di identificazione personale sono basati su
  - qualcosa che io so
  - qualcosa che io posseggo
  - qualcosa che io sono (sistemi biometrici)
- In rete può essere necessario provare l'identità anche di apparati attivi
  - Router, client, server
- Come provare l'identità di un'entità in rete basandosi esclusivamente sullo scambio di messaggi?
  - Mediante un protocollo di autenticazione
    - Es. Kerberos
  - Scenario: Bob vuole che Alice gli dimostri la sua identità

Protocolli di autenticazione (1)

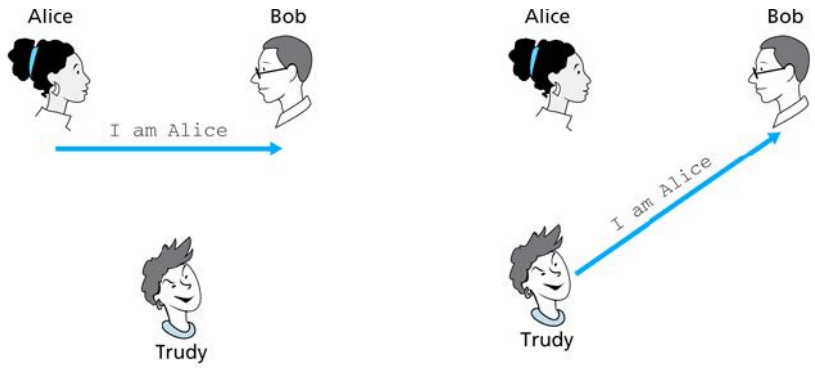


Figure 8.15 ♦ Protocol *ap1.0* and a failure scenario

Protocolli di autenticazione (2)

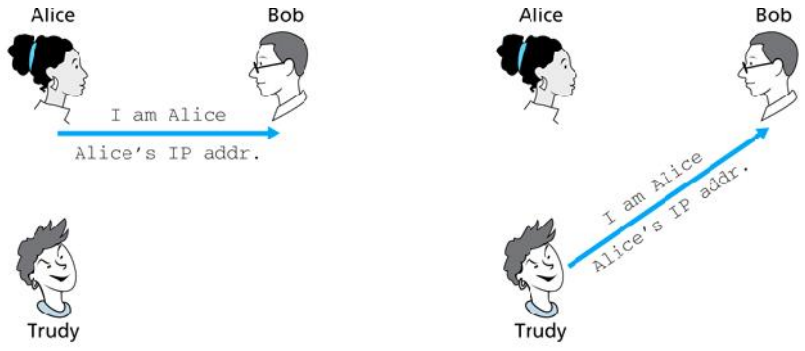
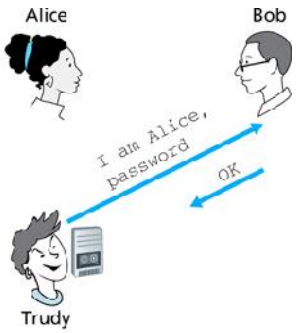


Figure 8.16 ♦ Protocol *ap2.0* and a failure scenario

Protocolli di autenticazione (3)



L'attacco replay funziona anche se la password è cifrata

Figure 8.17 ♦ Protocol *ap3.0* and a failure scenario

Protocolli di autenticazione (4)

- *R* è il **nonce** (number+once)
- Solo Alice (oltre a Bob) conosce la chiave simmetrica per decifrare il nonce

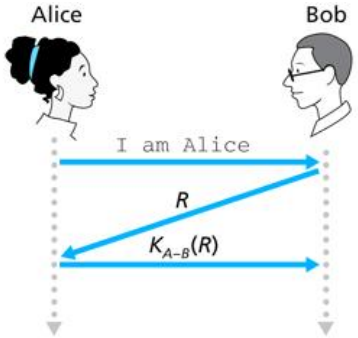
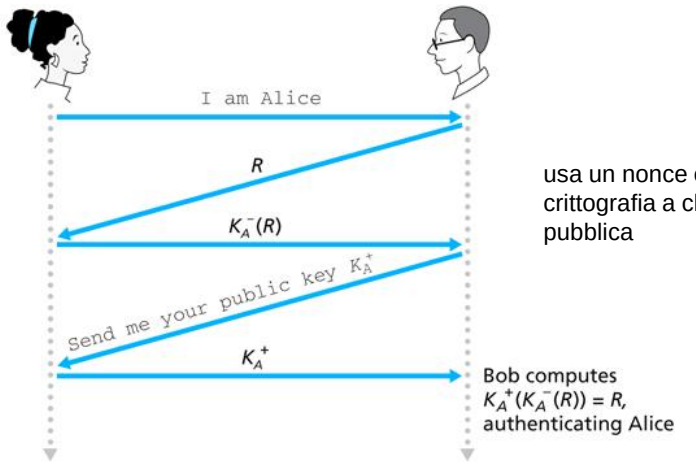


Figure 8.18 ♦ Protocol *ap4.0* and a failure scenario

Protocolli di autenticazione (5)



usa un nonce e la crittografia a chiave pubblica

Figure 8.19 ♦ Protocol *ap5.0* working correctly

Difetto di sicurezza in ap5.0

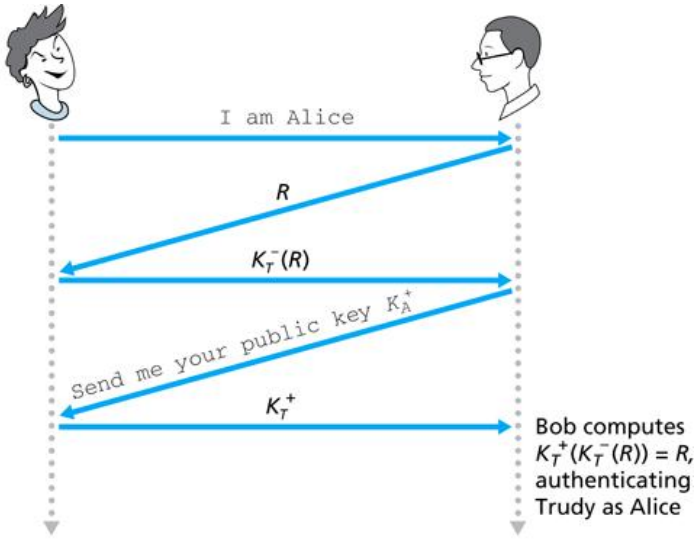


Figure 8.20 ♦ A security hole in protocol *ap5.0*

Connection hijacking in ap5.0

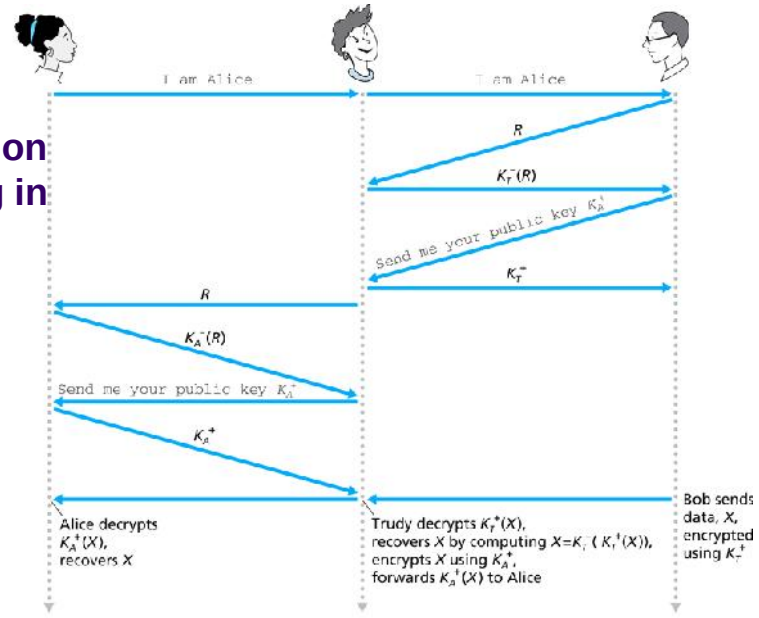
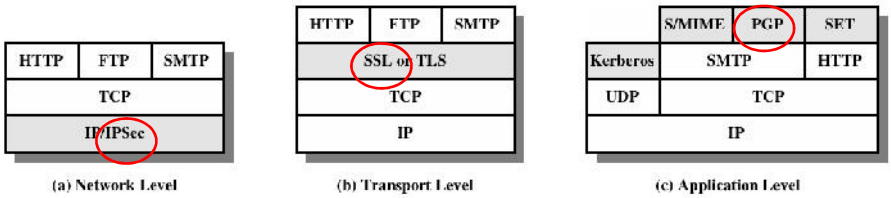


Figure 8.21 ♦ A man-in-the-middle attack

Crittografia

Principi di crittografia
Integrità dei messaggi
Protocolli di autenticazione
Sicurezza nella pila di protocolli di Internet: PGP, SSL, IPSec

Sicurezza nella pila di protocolli di Internet



Sicurezza a livello di applicazione: email (1)

Scenario: Alice e Bob sono interessati alla riservatezza del messaggio

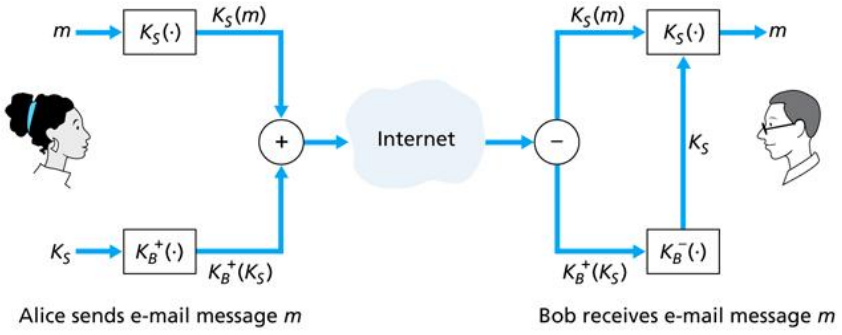


Figure 8.22 ♦ Alice used a symmetric session key, K_S, to send a secret e-mail to Bob.

Sicurezza a livello di applicazione: email (2)

Scenario: Alice e Bob sono interessati all'integrità del messaggio e all'autenticazione del mittente

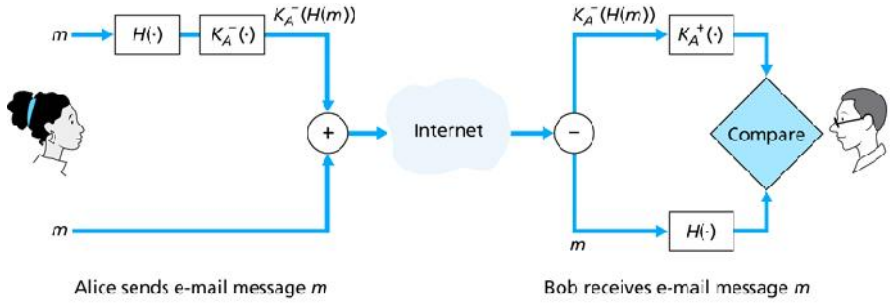


Figure 8.23 ♦ Using hash functions and digital signatures to provide sender authentication and message integrity

Sicurezza a livello di applicazione: email (3)

Scenario: Alice e Bob sono interessati alla riservatezza, integrità del messaggio e all'autenticazione del mittente

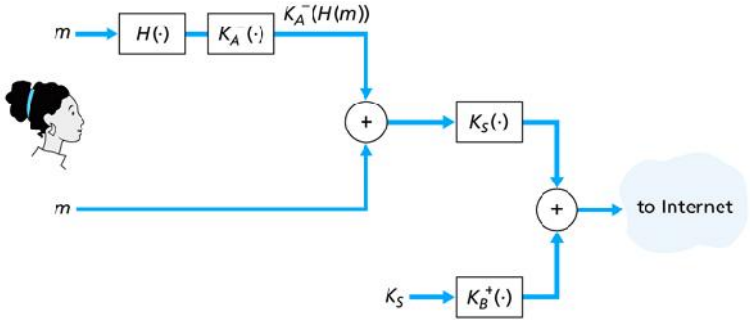


Figure 8.24 ♦ Alice uses symmetric key cyptography, public key cryptography, a hash function, and a digital signature to provide secrecy, sender authentication, and message integrity.

PGP

Pretty Good Privacy

- Ideato da Phil Zimmerman nel 1991 per rendere sicura l'email
- Assicura riservatezza, integrità del messaggio e autenticazione del mittente
  - Usa chiavi simmetriche di crittografia, chiavi pubbliche, funzioni hash
- Meccanismo di certificazione della chiave pubblica mediante rete di fiducia (web of trust)

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash:  SHA1
Bob:
Can I see you tonight?
Passionately yours, Alice
-----BEGIN PGP SIGNATURE-----
Version: PGP for Personal Privacy 5.0
Charset:  noconv
yhHJRHhGJGhgg/12EpJ+lo8gE4vB3mqJhFEvZP9t6n7G6m5Gw2
-----END PGP SIGNATURE-----
```

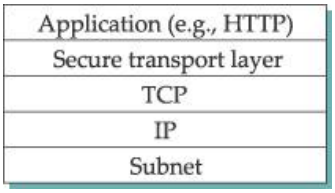
Figure 8.25 ♦ A PGP signed message

```
-----BEGIN PGP MESSAGE-----
Version: PGP for Personal Privacy 5.0
u2R4d+/jKmn8Bc5+hgDsqAewsDfrGdszX68liKm5F6Gc4sDfcXyt
RfdS10juHgbcfDssWe7/K=lKhnMikLo0+1/BvcX4t==Ujk9PbcD4
Thdf2awQfgHbnmKlok8iy6gThlp
-----END PGP MESSAGE
```

Figure 8.26 ♦ A secret PGP message

Secure Socket Layer (SSL)

- Ideato da Netscape per transazioni web
  - HTTPS equivale a HTTP basato su SSL/TLS
  - Well-known service con numero di porta 443
- Servizi di sicurezza
  - Riservatezza
  - Integrità
  - Autenticazione del server
  - Autenticazione del client (opzionale)



- Standard di fatto per la sicurezza a livello di trasporto
  - Una variante di SSLv3 è stata standardizzata da IETF con il nome Transport Layer Security (TLS)

SSL: handshake

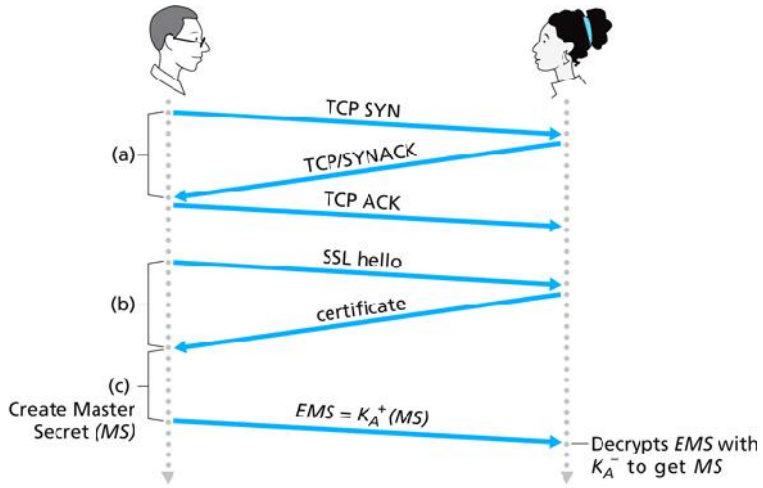
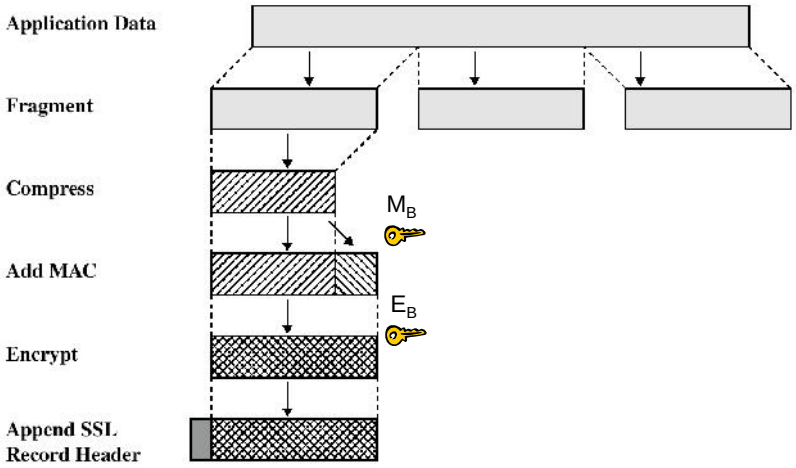


Figure 8.28 ♦ The almost-SSL handshake, beginning with a TCP connection

SSL: derivazione delle chiavi di sessione

- 4 chiavi generate a partire dal segreto condiviso (MS)
  - E_B: chiave di cifratura di sessione per i dati inviati da Bob ad Alice
  - E_A: chiave di cifratura di sessione per i dati inviati da Alice a Bob
  - M_B: chiave MAC di sessione per i dati inviati da Bob ad Alice
  - M_A: chiave MAC di sessione per i dati inviati da Alice a Bob
- Cifratura e algoritmi MAC negoziabili tra Bob e Alice

SSL: trasferimento dati



Record SSL

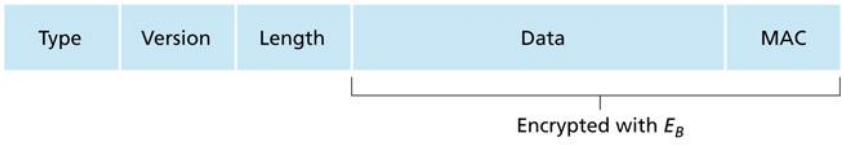


Figure 8.29 ♦ Record format for SSL

IPsec

IP security

- Un framework per consentire servizi di sicurezza a livello rete
  - Progettato sia per IPv4 che per IPv6
  - E' possibile realizzare reti virtuali private (VPN)
- SA: Associazione di sicurezza
  - Relazione unidirezionale tra sorgente e destinazione
    - Identificata da Security Parameter Index (SPI): 32 bit
- Protocollo di sicurezza
  - AH oppure ESP
- Gestione delle chiavi
  - Manuale
  - Automatica
    - Internet Security Association and Key Management Protocol (ISAKMP)
      - Protocollo Internet Key Exchange (IKE)

Prof. Filippo Lanubile

Protocollo AH: intestazione per l'autenticazione

- Fornisce l'autenticazione della sorgente e l'integrità dei dati ma non la riservatezza
- L'intestazione AH comprende:
  - SPI
    - identifica la SA
  - Digest del messaggio firmato dal mittente
    - autentica i dati
    - calcolato in base al datagramma IP originario
  - Campo intestazione successiva
    - specifica il tipo di dati (es.: TCP, UDP, ICMP)

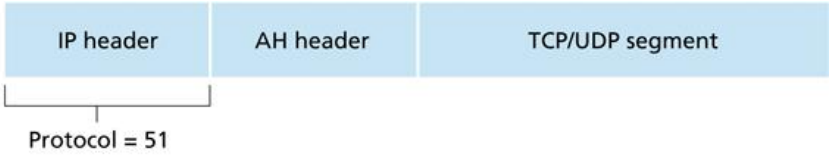


Figure 8.30 ♦ Position of AH header in IP datagram

Protocollo ESP

- Fornisce autenticazione della sorgente, integrità dei dati e riservatezza

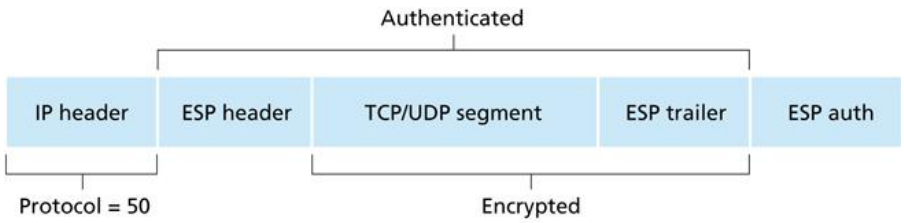


Figure 8.31 ♦ The ESP fields in the IP datagram

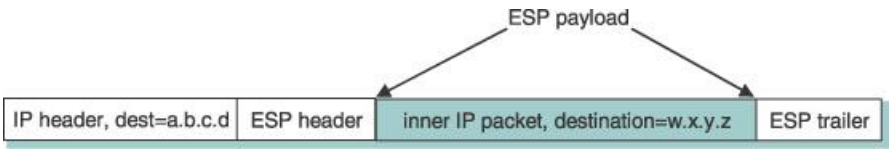
Modalità trasporto e tunnel in IPsec

Modalità trasporto

- Protezione per i livelli superiori
- Utilizzata per comunicazioni end-to-end
  - Es. Client-server

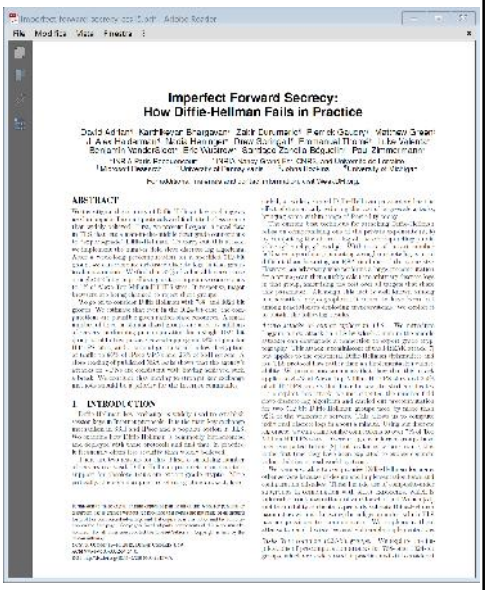
Modalità tunnel

- Protezione di un intero pacchetto IP
  - Nuovo pacchetto IP esterno che contiene il pacchetto ESP
  - Il pacchetto IP originale è interno al pacchetto ESP
- Tipicamente tra router



Diffie-Hellman key exchange

- Utilizzato da TLS e IPsec per la generazione di chiavi di sessione
  - VPN, SSH, HTTPS
- Diffie-Hellman break by NSA



Prof. Filippo Lanubile

https://freedom-to-tinker.com/blog/haldermanheninger/how-is-nsa-breaking-so-much-crypto/