Sicurezza in Rete: Attacchi

- Virus
- Intrusioni
- Sniffing
- Spoofing
- Denial of Service

Attacchi: Intrusioni

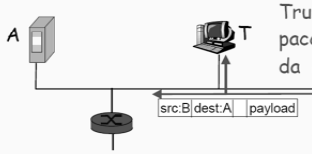
Tipi di intruso (o avversario)

- Intruso passivo: legge il messaggio senza alterarlo (es. password)
- Intruso attivo:
  - Può alterare il messaggio
  - Può inviare messaggi falsi spacciandosi presso il ricevente per il mittente autentico

Attacchi: Sniffing

Packet sniffing (to sniff = "odorare"):

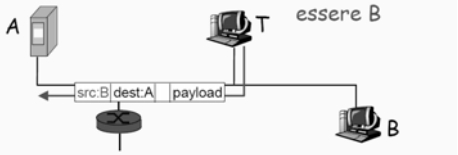
- Evidente in mezzi condivisi
- Un adattatore di rete programmato ad hoc (NIC) legge tutti i pacchetti in transito
- Tutti i dati non cifrati (es.: password) possono essere letti



Attacchi: Spoofing

IP Spoofing:

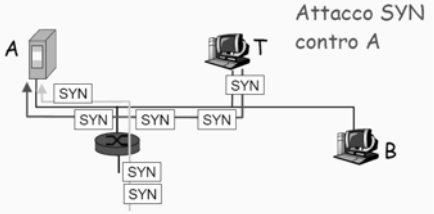
- Un host genera pacchetti IP con indirizzi di sorgente falsi
- Il ricevente non è in grado di stabilire se l'origine dei pacchetti sia quella autentica



Attacchi: Denial of Service

Denial of service (DoS):

- Flusso di pacchetti "maligni" che sommerge il ricevente
- Distributed DoS (DDoS): attacco coordinato multiplo



Sicurezza in Rete: Rimedi

- **Protocolli Sicuri (SSL, IPSEC)***
- **Firewall***
- Antivirus
- Sistemi di Accounting
- **Crittografia e Firma Digitale***

**** Esaminati in dettaglio nel seguito***

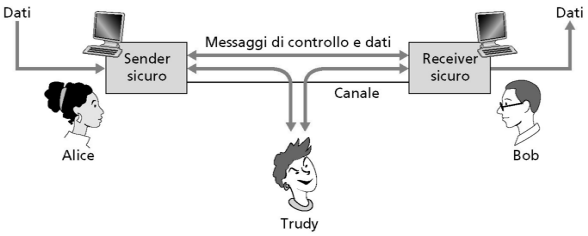
Rimedi: Crittografia

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 manda dei dati possa in futuro negare di averli mandati (firma digitale).

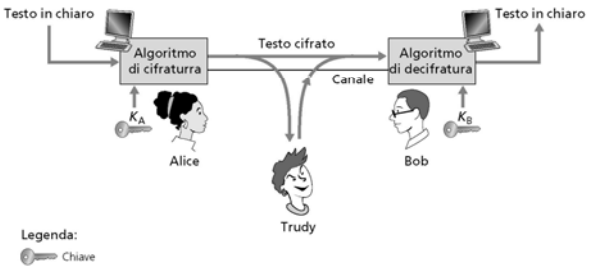
Rimedi: Crittografia

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



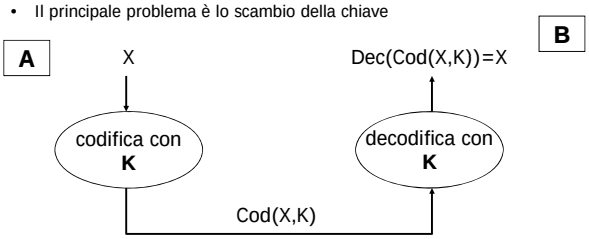
Rimedi: Crittografia

Componenti della crittografia



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



Crittografia: 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

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

Crittografia: 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"

Crittografia: Algoritmi a Chiave Segreta

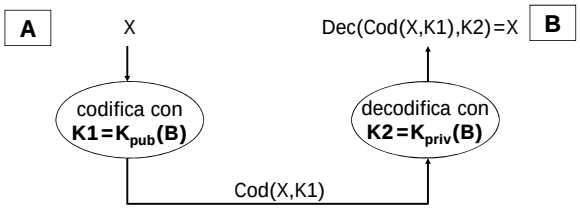
- **Cifrari di trasposizione**
  - le lettere sono trasposte in altre posizioni, secondo una chiave di trasposizione
  - ad esempio, la prima lettera è sostituita con la quarta, la seconda con la terza ecc.
  - anche questo algoritmo è facile da attaccare statisticamente
- **One-Time Pad**:
  - il testo 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

Crittografia: Algoritmi a Chiave Pubblica

- **Chiave Pubblica/Privata**: Ogni soggetto **S** ha
  - una propria *chiave pubblica* $\mathbf{K_{pub}(S)}$, nota a tutti;
  - una propria *chiave privata* $\mathbf{K_{priv}(S)}$ 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 non deve mai essere trasmessa in rete
  - deve essere molto difficile ricavare una chiave dall'altra (in particolare la chiave privata da quella pubblica)

Crittografia: Algoritmi a Chiave Pubblica

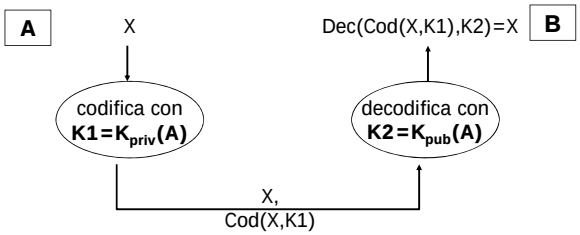
Procedura per garantire la **segretezza** dei dati scambiati.
Solo B può decodificare i dati cifrati da A.



Crittografia: 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.



Crittografia: 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 dati da trasmettere

Crittografia: 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(X)** più corta (detta **impronta**), con la proprietà che, dati due messaggi $M \neq M'$

  Prob(h(M)=h(M')) → 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**

Crittografia: 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

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

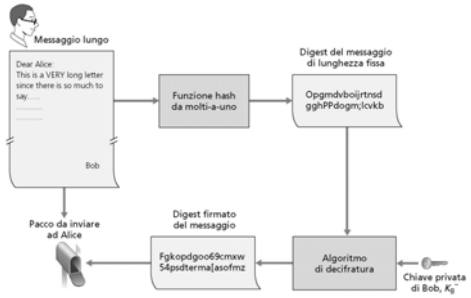
- Per ragioni di velocità e di compattezza dei dati, speso la firma digitale consiste nell'applicazione della chiave privata non al messaggio intero, ma ad **un'impronta hash** (o **digest**) del messaggio.

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).

Firma Digitale

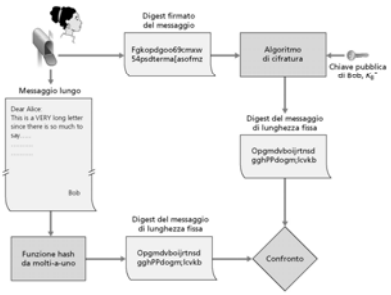
Bob crea il messaggio, ne calcola l'**impronta** (**digest**), **firma** l'impronta con la sua **chiave privata**, ed invia ad Alice **sia** il documento originale, **sia** il digest firmato



Firma Digitale

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.

  $\Rightarrow$ 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.