Corso di Sistemi di Elaborazione delle Inforazioni

Corso di Laurea in Infermieristica
Corso di Laurea in Tecniche di Radiologia Medica, per Immagini e Radioterapia
Università degli Studi di Messina
Anno Accademico 2006-2007 - Secondo Semestre

Docente: prof. Salvatore Siracusa

Crittografia e Sicurezza Informatica

La presente dispensa vuole essere una guida per gli argomenti trattati durante il corso.

Introduzione

Le informazioni e i dati digitali, siano essi "veicolati" sulle reti, siano essi memorizzati su proprio personal computer, risultano intrinsecamente esposti al rischio di intercettazione e falsificazione.

Oggi gli algoritmi crittografici, le operazioni di cifratura e di firma digitale che da essi derivano, potendo contare su una base matematica resa "forte" dalla teoria dei numeri e dalla irreversibilità computazionale di opportune condizioni, rendono sicuri canali e strumenti di comunicazione che per propria natura non lo sono, non si limitano a "nascondere" ma garantiscono la riservatezza e l'integrità delle informazioni, l'autenticità di interlocutori che operano a milioni di chilometri di distanza.

Il corso si propone di fornire una panoramica sugli aspetti generali della crittografia, sui contesti applicativi in cui si rivela fondamentale il suo utilizzo, e di illustrare il funzionamento dei principali algoritmi crittografici, le relative prestazioni e, particolarmente, le applicazioni che da essi derivano.

1. Un sistema di comunicazione sicuro

Le reti di comunicazione digitale non rappresentano, un sistema di comunicazione **sicuro** che per essere tale deve poter garantire:

- La **riservatezza**: certezza che la comunicazione non sia stata intercettata, letta e violata da alcuno.
- L'**integrità**: piena conformità della comunicazione rispetto all'originale, ovvero la certezza che la comunicazione non sia stata modificata.
- L'**autenticità**: certezza dell'effettiva provenienza della comunicazione da colui che afferma di esserne il mittente.
- Il **non-ripudio**: l'impossibilità del disconoscimento di una comunicazione, ovvero l'impossibilità da parte del mittente di negare di aver effettivamente trasmesso e del destinatario di negare di aver effettivamente ricevuto.
- La **disponibilità**: certezza di fruibilità delle risorse del sistema.

Ognuno di questi requisiti, sulla rete, può essere facilmente compromesso sfruttando una debolezza intrinseca dei protocolli di comunicazione: tutte le informazioni trasmesse durante una connessione viaggiano in chiaro. Le reti pubbliche non sono state progettate tenendo presente misure di autoprotezione e autodifesa contro abusi intenzionali o accidentali, i protocolli e le architetture che ne stanno alla base sono stati strutturati perché rendessero possibile, efficace e robusta, ma non intrinsecamente sicura, la comunicazione tra computer fisicamente lontani.

La risposta più completa ed esauriente a questa problematica è rappresentata da una sommatoria di fattori di carattere logico (sicurezza logica, quindi contromisure relative alle applicazioni, ai sistemi, alle politiche di implementazione, configurazione e comportamentali), e di carattere fisico (sicurezza fisica, quindi contromisure maggiormente legate alle infrastrutture). La **crittografia** gioca, specie nel primo caso, un ruolo cardine, consentendo di proteggere con un altissimo grado di sicurezza le comunicazioni su rete, **rendendo sicuri canali che intrinsecamente non lo sono**

Esistono due sistemi di crittografia, **simmetrica** o a chiave privata, basata sull'utilizzo di una sola chiave di cifratura, e **asimmetrica** o a chiave pubblica basata sull'utilizzo di due chiavi di cifratura.

2. La crittografia, aspetti generali

La crittografia, classicamente definita come l'arte o la scienza di rendere i messaggi sicuri, fornisce sia una serie di algoritmi e metodi per trasformare reversibilmente un testo e renderlo decifrabile solo a chi dispone di opportune informazioni (garanzia di riservatezza e autenticità), sia algoritmi e metodi per trasformare irreversibilmente un messaggio (garanzia di integrità).

Un protocollo crittografico può essere pensato come un programma costituito da due componenti, la prima quella di cifratura, trasforma il testo in chiaro in cifrato, la seconda trasforma questo nel testo originale; entrambe sono effettuate da algoritmi.

Un algoritmo crittografico (cifrario) è che una funzione matematica che applicata ad un testo lo trasforma realizzando le operazioni di cifratura e decifratura.

Il suo dominio è rappresentato dall'insieme dei messaggi da cifrare o decifrare e dallo spazio delle chiavi.

Le funzioni di cifratura e decifratura possono essere analiticamete diverse ma correlate in modo che detta *E* la funzione di cifratura, *D* quella di decifratura, *M* il messaggio da cifrare (la traduzione numerica del testo), *C* quello cifrato, *Ki* la chiave si abbia:

$$E(M, Ki) = C \qquad D(C, Ki) = M$$

Cifrare e decifrare significa calcolare queste funzioni.

Il presupposto base è che l'operazione di decifratura possa essere realizzata esclusivamente dalla chiave corrispondente a quella di cifratura.

Le chiavi sono sostanzialmente numeri casuali molto grandi, la cui lunghezza si misura in bit, se essa vale n lo spazio delle chiavi comprende 2^n elementi. Quindi maggiore è la lunghezza della chiave, più grande è lo spazio in cui cercarla e maggiore è la difficoltà di trovarla.

Maggiore è la lunghezza della chiave computazionalmente più pesanti saranno le operazioni di cifratura e decifratura, specie per quel che concerne, come verrà meglio chiarito nei paragrafi successivi, gli algoritmi asimmetrici. In generale possono esserci chiavi diverse per la cifratura e la decifratura**.**

La chiave è un parametro critico per la sicurezza: chiunque conosca quella di decifratura può leggere il messaggio, tutta la sicurezza del crittosistema risiede nella chiave e non nell'algoritmo che, pertanto, è noto, analizzabile e soggetto a possibili revisioni. Questo è quanto afferma il principio di Dutchman A. Kerckhoffs: "**Tutta le segretezza della chiave deve risiedere nella chiave e non nel metodo di cifratura**",

Gli algoritmi di cifratura reversibili si suddividono in 2 classi:

- Algoritmi simmetrici o a chiave privata.
- Algoritmi asimmetrici o a chiave pubblica

La differenza fondamentale è che i primi usano la stessa chiave per la cifratura e la decifratura, mentre i secondi usano due chiavi, una detta pubblica, una privata.

Gli algoritmi di cifratura irreversibili comprendono sostanzialmente le funzioni di hash, di seguito trattate.

2.1 Gli algoritmi simmetrici

L'utilizzo della medesima chiave per le operazioni di cifratura e decifratura comporta fondamentalmente il seguente paradigma

Due interlocutori devono:

1. scegliere un algoritmo calcolare e concordare una chiave comune
2. individuare un canale di comunicazione sicuro per lo scambio
3. prestare la massima attenzione e cautela affinché nessun altro al di fuori degli interessati conosca la chiave.

Solo ed esclusivamente la chiave posseduta dai due interlocutori potrà decifrare un messaggio cifrato con la medesima.

Il **vantaggio:** velocità sia per quel che concerne la generazione della chiave che per le operazioni di cifratura e decifratura (le operazioni base, XOR, permutazioni, sostituzioni e shift di bit, sono tipicamente di basso livello e quindi molto veloci), ciò li rende particolarmente idonei a trattare documenti di grossa dimensione.

Lo **svantaggio** è rappresentato dallo scambio e dalla gestione delle chiavi: non si può affidare ad un mezzo di per sé insicuro come la rete lo scambio di una informazione così critica, inoltre il numero delle chiavi cresce esponenzialmente col numero degli utenti (per n utenti esistono complessivamente n*(n-1)/2 chiavi). Un utente che comunica con n interlocutori non può gestire, mantenere segrete e aggiornare n chiavi diverse.

Queste problematiche, che hanno, inizialmente, ostacolato il diffondersi della crittografia, sono state risolte dagli algoritmi a chiave pubblica.

Attualmente l'algoritmo simmetrico più diffuso è il **3-DES** (Data Encryption Standard). L'algoritmo che lo sostituirà **AES** (Advanced Data Encryption Standard). Altri algoritmi simmetrici: **RC4, Blowfish, Idea, Cast**

2.2 Gli algoritmi asimmetrici

Gli algoritmi asimmetrici prevedono che un utente sia in possesso di una coppia di chiavi complementari, una detta pubblica, una privata, generate contestualmente. La chiave pubblica deve essere distribuita, quella privata, come per gli algoritmi simmetrici, deve essere assolutamente mantenuta segreta: solo il titolare della coppia di chiavi, ne deve avere il possesso, nessun altro suo interlocutore la deve conoscere. **In un sistema a chiave pubblica non esistono segreti (chiavi) condivisi**. La crittografia a chiave pubblica, concettualmente, nasce nel 1976, inventata da Whitfield Diffie e Martin Hellman, due anni dopo tre ricercatori del MIT, Ron Rivest, Adi Shamir, Leo Adleman diedero forma a quella che sembrava una pura utopia matematica definendo l'algoritmo **RSA**, il più diffuso e utilizzato.

Altri algoritmi asimmetrici sono il **Diffie-Hellman**, e il **DSS** (Digital Signature Standard), utilizzato solo per le firme digitali.

Dalla relazione matematica alla base delle operazioni di cifratura e decifratura, da quella che intercorre tra le chiavi e soprattutto dalla lunghezza in bit di queste (da 512 bit in poi), derivano 2 importanti proprietà:

- Il testo cifrato da una chiave può essere decifrato esclusivamente da quella corrispondente.
- La conoscenza della chiave pubblica (tutti la possono conoscere), "praticamente" non consente di calcolare la chiave privata, (il termine "praticamente" equivale a "impossibile computazionalmente")

L'utilizzo della medesima chiave per le operazioni di cifratura e decifratura comporta fondamentalmente il seguente paradigma

Due interlocutori devono singolarmente:

1. generare una coppia di chiavi e rendere nota quella pubblica, proteggere quella privata.
2. l'uno cifrerà il messaggio con la chiave pubblica dell'altro
3. ognuno decifrerà il messaggio con la propria chiave privata

Vantaggi:

- nessun problema nello scambio della chiave.
- possibilità di sfruttare la complementarietà delle chiavi per acquisire l'integrità e l'autenticità di un documento elettronico, mediante la realizzazione della **firma digitale**.

Svantaggi:

- la criticità nell'attribuire la chiave pubblica al legittimo proprietario
- la notevole complessità computazionale e i conseguenti tempi elevati di calcolo non solo per le operazioni di generazione delle chiavi (operazione

relativamente poco frequente) ma soprattutto per quelle di cifratura e decifratura (decisamente più frequenti).

Per l'algoritmo RSA, descritto più dettagliatamente nel paragrafo 3, detto k il numero in bit della chiave si hanno le seguenti complessità:

- Operazioni effettuate dalla chiave pubblica (cifratura e verifica della firma digitale): $O(k^2)$.
- Operazioni effettuate dalla chiave privata (firma digitale e decifratura): $O(k^3)$.
- Generazione delle chiavi: $O(k^4)$.

I tempi elevati per il calcolo delle chiavi RSA sono dovuti alla generazione di due numeri primi molto grandi di uguale dimensione (con non meno di 60 cifre decimali) e precisamente agli elevati cicli di calcoli necessari per testarene la primalità. In genere i tempi possono dipendere dalla "fortuna" della ricerca.[1] Vantaggi e svantaggi, in ultima analisi e con le opportune precisazioni, sono riconducibili alla *asimmetria* della base matematica presente nella implementazione degli algoritmi a chiave pubblica. Con particolare riferimento all'algoritmo RSA alla base della generazione delle chiavi c'è il prodotto dei 2 numeri primi, operazione tipicamente *one-way*, ossia rapidamente calcolabile in un senso, ma "computazionalmente irrealizzabile" in senso opposto.

Gli algoritmi asimmetrici risultano fortemente inefficienti per trattare documenti di grossa dimensione, significativamente più lenti di qualsiasi algoritmo simmetrico (l'algoritmo RSA utilizza chiavi 10 volte più lunghe per rendere 1/1000, rispetto al DES)

[1] I 2 numeri primi devono essere quasi di uguale dimensione, per rendere più difficile una eventuale fattorizzazione, se la chiave è lunga 512, veranno scelti da 256 255 bit, con tale numero di bit la "scelta" di numeri primi è vastissima, e tale da non creare collisioni, ossia repliche; si procede generando numeri casuali, sfruttando, per esempio, la "aleatorietà" del movimento del mouse o della velocità del tempo di battitura alla tastiera, combinati con data e ora.
Tra gli algoritmi più utilizzati per il test della primalità è il piccolo teorema di Fermat, algoritmo probabilistico, non assolutamente sicuro ma estremamente veloce.

2.3 La soluzione mista (garanzia di riservatezza)

La soluzione più indicata e perseguibile è quella di combinare i due sistemi per definirne un "**crittosistema misto o ibrido**", che combini i vantaggi di entrambi. L'utilizzo di un sistema misto comporta fondamentalmente il seguente paradigma

1. generare una coppia di chiavi e rendere nota quella pubblica, proteggere quella privata.
2. Generare una chiave simmetrica
3. Cifrare con questa il documento da spedire
4. Utilizzare la chiave pubblica dell'interlocutore per cifrare la chiave simmetrica, operazione perfettamente realizzabile in quanto la lunghezza della chiave da cifrare è di pochi byte.
5. Spedire in un unico documento, sia il testo cifrato che la chiave simmetrica cifrata.

Il testo è al **"sicuro"** grazie all'algoritmo simmetrico. L'algoritmo asimmetrico **"crea il canale di scambio sicuro"** di cui necessita la chiave segreta.

L'interlocutore che riceve dovrà:

1. Utilizzare la propria chiave privata per decifrare la chiave simmetrica,
2. Utilizzare la chiave simmetrica per decifrare e recuperare il testo originale.

2.4 La firma digitale (garanzia di integrità e autenticità)

La crittografia asimmetrica consente di creare un sistema di **firma digitale**. Due sono i fattori base:

- Un messaggio cifrato con una chiave può essere decifrato solo ed esclusivamente dalla chiave corrispondente.
- La segretezza della chiave privata, che deve essere posseduta esclusivamente dal legittimo proprietario.

L'utilizzo della firma digitale comporta fondamentalmente il seguente paradigma

1. generare una coppia di chiavi e rendere nota quella pubblica, proteggere quella privata.
2. Cifrare un messaggio con la chiave privata
3. Inviare sia il documento in chiaro sia il codice cifrato, sia la chiave pubblica. Tutti possono decifrare il codice cifrato (ovvero verificare la firma) con la chiave pubblica.

Problema: inefficienza degli algoritmi asimmetrici

Soluzione: utilizzo delle funzioni di **hash**. Sono particolari funzioni matematiche computazionalmente *one-way* (unidirezionali), che prendono in input un numero arbitrariamente grande (la traduzione in bit del messaggio da firmare) per restituire un numero fisso, costante e piccolo di byte (tipicamente 128 o 160), il cui valore dipende strettissimamente dal valore di input. Il valore di ritorno è detto **impronta** o **hash del messaggio** e costituisce una vera e propria impronta digitale, grazie alle proprietà di cui godono le funzioni di hash:

- l'hash non è computazionalmente invertibile, ossia data l'impronta non si può risalire al testo originale
- L'hash non è computazionalmente soggetto a collisioni, ossia documenti distinti producono impronte distinte.
- Anche una piccolissima modifica al testo originale fornisce un valore di hash totalmente diverso. Calcolare l'hash di un documento significa, quindi, calcolare un "digest" ossia un riassunto che lo identifica in maniera inequivocabile e irripetibile.

Le principali funzioni di hash: SHA-1, MD2, MD5, Ripmode-160

Il paradigma di firma digitale diventa, pertanto, il seguente:

1. generare una coppia di chiavi e rendere nota quella pubblica, proteggere quella privata.
2. Calcolare l'impronta del messaggio
3. Cifrare l'impronta con la chiave privata.(ovvero la firma digitale)

4. Inviare sia il documento in chiaro, sia la firma digitale, sia la chiave pubblica.

Il paradigma di verifica della firma digitale diventa, pertanto, il seguente:

1. acquisire il messaggio firmato
2. calcolare l'impronta sul testo in chiaro
3. essere sicuro dell'appartenenza della chiave pubblica al legittimo firmatario (vd certificati digitali)
4. decifrare la firma digitale
5. confrontare le due impronte: se il confronto è positivo il documento è integro, diversamente è stato modificato.

Firmare digitalmente un documento elettronico, oggi, in Italia, è una operazione legalmente valida. (rif. www.aipa.it)

2.5 I certificati digitali (garanzia di autenticità)

I certificati digitali risolvono il problema di creare una corrispondenza biunivoca e inalterabile tra il legittimo proprietario e la chiave pubblica, e, quindi, per l'associazione matematica e univoca di questa con la chiave privata, una corrispondenza tra il sottoscrittore e la firma digitale. Per legittimo proprietario di una chiave pubblica, può intendersi di un entità del tutto generica, relativamente al contesto che si vuole autenticare: può trattarsi di una persona, di un server o browser Web, di un Ente o di uno sviluppatore software. Occorre che qualcuno garantisca e si prenda la responsabilità dell'appartenenza della chiave pubblica al legittimo proprietario, esattamente, come, per analogia al mondo reale, un funzionario pubblico, o chi per lui, *sottoscrive* la corrispondenza tra la foto e i dati personali contenuti in una carta di identità. Attualmente chi assolve a tale compito sono terze parti fidate, dette Autorità di Certificazione, in Italia ne esistono 14.

Un certificato digitale, in generale, è un insieme di informazioni raggruppabili in 4 tipologie:

- chiave pubblica
- informazioni riferibili all'entità proprietaria della chiave pubblica, i dati anagrafici, per esempio, se il certificato è riferito ad una persona.
- informazioni riferibili alla firma digitale di una terza parte, apposta al certificato.
- informazioni relative agli algoritmi utilizzati per la creazione del certificato, un univoco numero di serie, la data di emissione e di scadenza.

La firma digitale di una terza parte una sorta di sigillo che certifica la chiave pubblica, è l'elemento cardine che sancisce la validità di un certificato intesa come la **misura** dell'appartenenza della chiave pubblica all'entità in possesso delle informazioni in esso contenute. Firmare digitalmente un certificato, pensabile come un normalissimo documento, significa, in generale **accertare prima** e **sottoscrivere poi**, la validità dello stesso. Esistono diverse modalità per acquisire la validità di un certificato, la più sicura è quella della verifica diretta dell'identità di un individuo o della società gestore di un server Web o comunque un metodo che possa riprodurre un equivalente livello di attendibilità.

3. **L'algoritmo RSA (Rivest, Shamir, Adleman)**

Da quanto sopra esposto è evidente il ruolo chiave e fondamentale della crittografia asimmetrica nello sviluppo, applicabilità e integrazione nelle reti pubbliche della Public Key Infrastructure, senza la crittografia asimmetrica difficilmente le tecniche crittografiche avrebbero potuto contribuire a rendere "sicuri" canali di comunicazione che di per se non lo sono. RSA è l'algoritmo asimmetrico più utilizzato, una base matematica semplice e, al tempo stesso, computazionalmente robusta, a cui si deve buona parte dello sviluppo e della diffusione di Internet in settori in cui la "componente" sicurezza rappresenta una *conditio sine qua non*: commercio elettronico, e gestione dei flussi documentali tra i principali.

Descrizione dell'algoritmo RSA

L'algoritmo di questo sistema, inventato nel 1977 da Ron Rivest, Adi Shamir, e Leonard Adelman, è basato sull'operazione di calcolo del modulo di numeri primi. La robustezza di questo algoritmo risiede nella difficoltà computazionale di pervenire alla fattorizzazione di un certo numero n.

Questi i passi fondamentali dell'algoritmo:

1. **Generazione delle chiavi pubblica e privata :**
   - scelta di *p, q* $\forall$ *p, q primi* $\hat{I}$ *N*.

   p, q devono essere numeri casuali di almeno di 1024 bit ciascuno, generati con opportuni fattori (semi) di casualità.

Si verifica che tali numeri siano primi con test di primalità

- calcolo di *n* $n = p*q \ \hat{I} \ N$.
- calcolo di n_1 $n_1 = (p-1)*(q-1) \ \hat{I} \ N$.

a) scelta dell'esponnte pubblico *e* $e \ \hat{I} \ N$: *primo rispetto* n_1

b) ricerca dell'esponente privato *d*

$\exists \ d, k \ \hat{I} \ N : \ e*d = k*n_1+1 => e*d \ mod \ n_1 = 1$

c) chiave pubblica *(n,e)*

d) chiave privata *(n,d)*

2. **cifratura di un messaggio m tramite chiave pubblica (n,e) :**

a) generazione del messaggio *m* in formato numerico

b) cifratura del messaggio $c = (m^e) \ mod \ n$

3. **decifratura di un messaggio cifrato c tramite chiave privata (n,d) :**

a) decifratura del messaggio $m = (c^d) \ mod \ n$

b) generazione del messaggio nel formato originale.

Si ha: $(c^d) = (m^e)^d = (m^{ed}) = (m^{k*(p-1)*(q-1)+1}) = m*(m^{k*(p-1)*(q-1)+1}) = m*1 = m$

//tutto in mod n

4. fase di firma di un messaggio m tramite chiave privata (n,d) :

a) generazione del messaggio *m* in formato numerico

b) generazione firma digitale *s* $s = (m^d) \bmod n$

c) generazione del messaggio firmato $m_1 = m \text{ È } s$

5. fase di verifica di un messaggio firmato m_1 e della relativa firma s tramite chiave pubblica (n,e) :

a) generazione del messaggio m $m = (s^e) \bmod n$

b) verifica che m abbia contenuti uguali a m_1

c) generazione del messaggio m nel formato originale.

5. Specifiche Algoritmo DES

Applied Cryptography, Protocols Algorithms, and Source Code in C Second Edition cap.12

6. Protocollo SSL

http://developer.netscape.com/tech/security/ssl/howitworks.html

Guide su SSL pubblicate su www.trustitalia.it