CRITTOGRAFIA

La crittografia è un sistema sofisticato per far viaggiare in rete delle informazioni private, al riparo da occhi indiscreti che potrebbero intercettarle. Cifrare è, infatti, una parola proveniente dal greco cryptos, che significa nascosto. Per proteggere un messaggio il mittente trasforma il testo originale (**plaintext**) in un messaggio criptato (**ciphertext**).
Il messaggio criptato viene trasmesso attraverso il canale di comunicazione. Se il messaggio viene intercettato, l'intruso non è in grado di ricostruirlo; solo il destinatario è in grado di riportarlo nella forma originale.
Queste due operazioni sono dette rispettivamente cifratura e decifratura.

I TEMI CHE ANDREMO A TRATTARE:

- Definizione di chiave
- Algoritmi basati su chiavi private (o simmetriche)
- Algoritmi basati su chiavi pubbliche (o asimmetriche)
- Come si garantisce la riservatezza e l'autenticità
- Sistemi crittografici
- Protocolli crittografici
- Utilizzo delle funzioni di HASH

Algoritmi

Attraverso Internet viaggiano ormai sempre più spesso informazioni riservate, la cui protezione richiede tutte le attenzioni normalmente dedicate ai corrispondenti documenti cartacei.
Scambi di posta elettronica, operazioni bancarie e commerciali dal proprio pc rappresentano dei progressi telematici ormai alla portata di mano di qualsiasi cittadino. Un'evoluzione dei tempi che però trascina con sé i rischi che altri individui entrino in possesso dei nostri segreti e dei nostri soldi. La protezione della segretezza digitale diventa allora una delle sfide più grandi che si trovano a dover affrontare quei paesi che stanno investendo nella comunicazione a distanza.

Il passaggio da tali documenti tradizionali al relativo documento elettronico deve venire gestito in maniera tale da conservare, ed eventualmente migliorare, le tradizionali politiche di sicurezza al fine di consentire un sistema di comunicazione sicuro.
La scienza che si occupa di effettuare questa trasposizione, la crittografia, offre una controparte elettronica a tutti i dispositivi e le procedure tradizionalmente adottati. La crittografia moderna si occupa della individuazione e messa a punto di dispositivi di sicurezza o algoritmi per i documenti elettronici, in maniera del tutto analoga ai corrispondenti documenti cartacei, che vanno ben oltre l'individuazione di un semplice codice di cifratura per proteggere i propri dati.

Definizione di chiave

Gli algoritmi utilizzati per la cifratura e la decifratura del documento sono delle funzioni matematiche che oltre al messaggio stesso, hanno come argomento quella che viene definita **chiave**. **La *chiave* è infatti un insieme di parametri** (tipicamente una famiglia di numeri casuali grandi) con cui si completa l'algoritmo il quale potrebbe anche essere reso pubblico, l'importante è mantenere la segretezza della chiave.
La chiave deve dunque essere trasmessa mediante un canale sicuro. La sua distribuzione diventa un problema serio quando si deve mantenere la privatezza su una rete estesa.

Algoritmi simmetrici

La crittografia classica si basa sull'uso di chiavi simmetriche, ovvero chiavi uguali utilizzate sia per la cifratura che per la decifratura del testo.
Se l'utente A vuole comunicare con B, insieme devono concordare un algoritmo di cifratura ed una chiave K. Se A vuole trasmettere un messaggio riservato a B dovrà cifrarlo usando l'algoritmo e la chiave prescelta, in questo modo il testo sarà trasformato in una sequenza incomprensibile di caratteri. L'utente B potrà decifrare il testo utilizzando lo stesso algoritmo e la stessa chiave.

Tali algoritmi hanno però diversi problemi:

- La chiave segreta deve essere scambiata tra i due interlocutori, per cui esiste almeno un momento in cui una terza persona potrebbe impossessarsene durante la trasmissione.
- Se la chiave viene compromessa da una terza persona, questa non solo potrà decifrare tutto il traffico cifrato con quella chiave ma anche produrre dei messaggi falsi o alterare gli originali senza che il destinatario se ne renda conto.
- Ogni coppia di interlocutori deve definire una chiave per qui c'è la necessità di un grande numero di chiavi.

Algoritmi asimmetrici

Questo tipo di algoritmi garantiscono maggiormente:

- Riservatezza e inviolabilità del documento
- Autenticità della provenienza del documento
- Impossibilità di disconoscere la trasmissione o la ricezione del messaggio

Essi si basano sull'uso di una coppia di chiavi l'una pubblica e l'altra privata, le quali possono essere utilizzate indifferentemente per la cifratura o la decifratura.
A seconda dell'uso che ne viene fatto, si potrà garantire la riservatezza e l'autenticità dei dati trasmessi.

Il sistema a chiave asimmetriche ha i seguenti vantaggi:

- Non richiede una informazione segreta per la cifratura e la decifratura che deve essere scambiata tra i due soggetti che vogliano comunicare;
- Non richiede di dover tenere una chiave segreta per ogni soggetto con il quale si vuole comunicare, infatti la chiave utilizzata da un qualsiasi utente per inviare messaggi segreti allo stesso destinatario è uguale per tutti.

Come si garantisce la riservatezza e l'autenticità

Per garantire la riservatezza, se l'utente A vuole trasmettere un documento che deve rimanere riservato, cioè comprensibile solamente a B, dovrà cifrare il documento con la chiave pubblica del destinatario così che, l'unica persona in grado di decifrarlo sarà proprio il destinatario del messaggio (in quanto solo lui possiede la sua chiave privata ed un messaggio cifrato con chiave pubblica può essere decifrato solo con la corrispondente chiave privata).
Invece, se l' utente A vuole garantire l'autenticità del documento da lui trasmesso, ovvero vuole garantire al destinatario che quel documento proviene effettivamente da lui, dovrà cifrarlo con la propria chiave privata ed inviarlo insieme al documento originale al destinatario. Il destinatario potrà a sua volta verificarne la provenienza confrontando il messaggio originale con quello decifrato tramite chiave pubblica del mittente (ciò basta a garantire l'autenticità perché il mittente è l'unico a conoscere la propria chiave privata ed un messaggio cifrato con chiave privata può essere decifrato solo con la corrispondente chiave pubblica).

La chiave pubblica degli utenti sarà resa disponibile a tutti coloro che la necessitano e da essa non si potrà in alcun modo risalire alla corrispondente chiave privata. Le chiavi non devono essere troppo corte perché in tal caso il processo di decifrazione sarebbe favorito dalle potenze di calcolo ormai molto elevate.

Sistemi crittografici

I più moderni sistemi crittografici, si basano su quelle che potremmo chiamare "funzioni pseudounidirezionali". Una funzione pseudounidirezionale è una funzione facilmente computabile, la cui funzione inversa non può essere computata a meno che non si posseggano certe informazioni particolari utilizzate nella sua costruzione, che fungono da parola d'ordine o da "molla segreta".
La ricerca di funzioni pseudounidirezionali conduce a quella classe di problemi che la teoria della complessità ha caratterizzato come non-deterministici. Ai fini dei crittosistemi la proprietà di maggiore interesse dei problemi non deterministici è data dal fatto che attualmente tutti gli algoritmi noti, atti a darne una soluzione generale, richiedono un tempo di calcolo che cresce rapidamente benché il controllo di una particolare soluzione proposta si compia velocemente.
Questi problemi si presentano ottimamente alla costruzione di funzioni unidirezionali, inoltre è stato possibile introdurre nelle funzioni certe molle segrete; uno dei più conosciuti algoritmi che si basa su un problema non deterministico (la fattorizzazione di un numero grande in numeri primi) è l'RSA.

Protocolli crittografici

Un protocollo crittografico in generale non specifica gli algoritmi da usare nei vari passi, ma piuttosto:

- quali tecniche adottare (ad esempio: crittografia a chiave pubblica e/o privata);
- quale successione di passi deve essere seguita;
- quale tecnica va adottata in ogni passo.

Esistono vari protocolli crittografici, che si differenziano per:

- il contesto iniziale (ad esempio: i due partecipanti hanno una chiave segreta in comune o no? Conoscono le rispettive chiavi pubbliche o no?);
- gli scopi da raggiungere (ad esempio: autenticazione, segretezza, o entrambi?).

Un primo problema da affrontare e risolvere mediante un protocollo crittografico è il seguente: in un contesto distribuito come il Web, è impensabile che ogni potenziale coppia di comunicatori disponga di una chiave segreta. Dunque, bisogna trovare un protocollo per concordare, all'inizio della sessione, la chiave segreta da usare durante il resto della sessione, detta per questo **chiave segreta di sessione.**
Un protocollo di per se molto semplice sfrutta la crittografia a chiave pubblica:

1. Alice invia la sua chiave pubblica a Bob;
2. Bob genera una nuova chiave segreta, la cifra con la chiave pubblica di Alice e la invia ad Alice;
3. Alice riceve la chiave segreta (cifrata) e la decifra con la propria chiave privata;
4. Bob e Alice a questo punto condividono la chiave segreta di sessione per mezzo della quale possono comunicare in tutta sicurezza.

Utilizzo delle funzioni di HASH

La funzione HASH, converte un testo di qualsiasi dimensione in una stringa binaria di lunghezza fissa (in genere 128 o 160 bit) molto più breve del documento stesso, ciò serve a garantire una maggiore efficienza dell'algoritmo di cifratura.

L'hash è l'attuale moderno strumento per la verifica di integrità, nel trasferimento sicuro dei messaggi.
Le funzioni di hash possono essere sfruttate per affrontare i seguenti problemi:

- Trasferimento sicuro di messaggi
- Firma digitale
- Memorizzazione sicura di una chiave segreta
- Certificazione di una chiave pubblica
-

Una problematica legata a quella del trasferimento sicuro di messaggi è quella della loro verifica di integrità.
E' nel caso di firma digitale, però, che l'hash si rivela, per questioni computazionali, indispensabile.
L'hash si rivela utile per la memorizzazione di chiavi segrete (passo presente anche nelle problematiche di certificazione).

Un esempio di algoritmo asimmetrico: RSA

Uno degli algoritmi asimmetrici più conosciuti è l'algoritmo RSA, acronimo formato dalla prima lettera dei cognomi di coloro che lo inventarono nell'aprile del 1977: Ronald L. Rivest, Adi Shamir e Leonard M. Adleman.

Le due chiavi possono essere ricavate l'una dall'altra, ma la garanzia fornita da RSA è che l'operazione di derivare la chiave segreta da quella pubblica è troppo complessa per venire eseguita in pratica, anche su un calcolatore molto potente; infatti, RSA sfrutta il fatto che è facile calcolare il prodotto di due numeri primi anche molto grandi, ma dato un numero è particolarmente oneroso a livello computazionale scomporlo, cioè trovare i numeri primi il cui prodotto è proprio il numero dato.

Per capire meglio facciamo un esempio pratico che dovrebbe chiarire l'idea su cui si basa l'RSA:

- si scelgono a caso due numeri primi, p e q , l'uno indipendentemente dall'altro, abbastanza grandi da garantire la sicurezza dell'algoritmo;
- si calcola il loro prodotto $n=n \times p$, chiamato *modulo*(ricordate l'aritmetica dell'orologio);
- si sceglie poi un numero e (chiamato *esponente pubblico)*, più piccolo e primo (cioè il cui unico divisore comune è 1) con $(p-1)(q-1)$;
- si calcola il numero d (chiamato *esponente privato*) tale che $e \bullet d = 1 \bmod (p-1)(q-1)$.

La chiave pubblica è (n, e), mentre la chiave privata è (n, d).

Il punto di forza dell'algoritmo sta nel fatto che con numeri molto grandi il calcolo di d è molto lungo e difficoltoso.

Lo scambio di chiavi rimane comunque un problema grave, nonostante l'uso della crittografia asimmetrica. Infatti, anche se non viene trasmessa la chiave privata, è comunque necessario inviare alcuni dati (i numeri n ed e) che qualcuno, diverso dal legittimo destinatario, potrebbe intercettare ed utilizzare per ricostruire l'intera chiave.

ESEMPIO ALGORITMO RSA IN JAVA

```
public RSAService() {
	// Genero i 2 numeri primi necessari per comporre le chiavi
	p = BigInteger.probablePrime(512, new SecureRandom()); // Numero primo p
	q = BigInteger.probablePrime(512, new SecureRandom()); // Numero primo q
	// Ottengo n moltiplicando p e 1
	n = p.multiply(q);
	// Ottengo p-1 e q-1
	BigInteger p1 = p.subtract(BigInteger.ONE);
	BigInteger q1 = q.subtract(BigInteger.ONE);
	// Calcolo fi moltiplicando tra loro p-1 e q-1
	BigInteger fi = p1.multiply(q1);


public String crypt(BigInteger text) {
	BigInteger crypto = text.modPow(d, n); // Cifratura
	return crypto.toString();
}


public String decrypt(BigInteger text) {
	BigInteger result = text.modPow(e, n); // Decifratura
	return result.toString();
}
```

Crittografia AES

Advanced Encryption Standard (AES) è uno dei più frequentemente usati e più sicuri algoritmi di crittografia disponibili ad oggi.

AES è l'algoritmo avanzato di crittografia migliore esistente grazie alle seguenti caratteristiche:

- **Sicurezza:** Gli algoritmi AES hanno la capacità di resistere agli attacchi meglio di altri metodi di crittografia.
- **Costo:** Destinato a essere rilasciato in modo globale,l'algoritmo AES è efficiente su base computazionale e di memoria.
- **Implementazione:** L'algoritmo AES è flessibile e altamente adattabile quando implementato in hardware e software, oltre ad essere semplice da implementare.

L'algoritmo è basato su diverse sostituzioni, permutazioni e trasformazioni lineari, ognuna eseguita su blocchi di dati da 16 byte - da qui il termine cifrario a blocchi. Queste operazioni sono eseguite più volte, chiamate "rounds".

Durante ogni round, una chiave "roundkey" viene calcolata dalla chiave di crittografia, ed incorporata nei calcoli.

Il cambio di un singolo bit, che sia sulla chiave, o nel blocco di testo in chiaro, risulta in un blocco di testo cifrato completamente differente - un chiaro vantaggio sui flussi cifrati tradizionali.

La differenza tra AES-128, AES-192 e AES-256 infine è la lunghezza della chiave: 128, 192 o 256 bit

Per illustrare: craccare una chiave AES a 128bit con un supercomputer di ultima generazione impiegherebbe più tempo della presunta età dell'universo.

Ad oggi, non esiste un attacco praticabile contro AES. Quindi, AES rimane lo standard di crittografia preferita per governi, banche e sistemi di alta sicurezza in tutto il mondo.

DES

Possiamo considerare DES come il padre di tutti gli algoritmi simmetrici moderni.

La sua principale proprietà è di essere un algoritmo semplice sia dal punto di vista teorico che implementativo.

Il DES è un codice cifrato a blocchi. La chiave usata per cifrare è un blocco di 64 bit suddivisa in 8 sotto blocchi di 8 bit ciascuno; l'ultimo bit di ogni sotto blocco è di controllo, di conseguenza i bit liberi che costituiscono in pratica la chiave sono 56.

Il testo da cifrare viene suddiviso in blocchi di 64 bit ciascuno e vengono cifrati uno dopo l'altro in successione con uguale procedimento.

La semplicità è una proprietà molto importante perché consente di provare matematicamente la sicurezza dell'algoritmo, o meglio poter dimostrare che tutti gli attacchi noti sono almeno veloci quanto l'attacco di forza bruta.

La semplicità implementativa ovviamente rende più facile evitare errori nella implementazione in software e hardware e potenzialmente favorisce anche la velocità di esecuzione.

Il grande punto debole del DES è la lunghezza della chiave simmetrica, di soli 56 bit. Una delle ipotesi principali è che l'NSA, che richieste espressamente l'adozione di questa chiave, fosse in grado di decrittare un testo cifrato con una chiave di 56 bit.

In generale DES è comunque un ottimo algoritmo, oggi non è più sicuro solo perché è possibile fare degli attacchi di forza bruta(*brute-force)* a qualunque

BIBLIOGRAFIA

- http://www.cardano.pv.it/studenti/matedida/crittografia/crittografia.htm
  Informazioni riguardanti l'introduzione alla crittografia, la definizione di chiave e le funzioni di HASH.

- http://matematica-old.unibocconi.it/interventi/bovetti/bovetti3.htm
  Informazioni riguardanti i vantaggi e gli svantaggi dell'algoritmo RSA.

- https://www.boxcryptor.com/it/encryption/
  Informazioni riguardanti i vantaggi e gli svantaggi dell'algoritmo AES.

- http://www.science.unitn.it/~sala/cryptowars/Q01_web.pdf
  Dati riguardanti DES.

- Paolo Ollari, *Corso di sistemi e reti vol.3*, Zanichelli, Bologna,2013
  Per gli esempi di funzionamento dell'algoritmo(Alice e Bob).