Enrico Zimuel
consulenza informatica

Introduzione alla crittografia open source

di Enrico Zimuel

"Se un sistema è veramente sicuro, lo è anche quando i dettagli divengono pubblici"
B.Schneier

Molti di voi si staranno chiedendo cosa diavolo centri la crittografia con l'open source, in quest'articolo cercherò di dimostrare, dopo aver introdotto alcuni concetti di base, che la crittografia per essere realmente utile e quindi sicura deve essere di tipo open source.

Crittografia open source?

Nel 1883 August Kerckhoffs, noto linguista franco-olandese e studioso di crittologia, pubblica un articolo intitolato "La cryptographie militaire" nel Journal des Sciences Militaires sulle tecniche d'utilizzo della crittografia nella strategia militare (riferimento [1] in webografia).

In quest'articolo è presente una frase, diventata poi famosa con il nome di principio di Kerckhoffs, nella quale si afferma che "la sicurezza di un sistema crittografico deve essere legata alla sola conoscenza della chiave".

Secondo questo principio la sicurezza di un sistema crittografico deve essere affidata esclusivamente alla conoscenza della chiave e quindi si deve dare per scontato che il "nemico" sia a conoscenza delle specifiche del cifrario o, per dirla in termini moderni, che sia a conoscenza dei codici sorgenti dell'algoritmo crittografico.

Questo principio, introdotto più di cento anni fa quando i computer erano ancora nei sogni di pochi visionari, come ad esempio Charles Babbage (riferimento [2] in webografia), può essere considerato come il precursore di uno dei principi dell'open source: la libera diffusione dei codici sorgenti. Certo, il principio di Kerckhoffs afferma semplicemente che si deve dare per scontato che le specifiche tecniche dei cifrari siano di dominio pubblico e non parla di libera circolazione del software ma traslando il concetto se ne deduce che solo grazie ad una libera diffusione dei codici sorgenti, con il conseguente studio della validità tecnica da parte dell'opinione pubblica, si può ottenere sicurezza.

"Poiché la sicurezza non ha niente a che vedere con la funzionalità, il beta testing non può in alcun modo rilevare i problemi di sicurezza; l'unico modo di verificare che un sistema è sicuro è sottoporlo all'esame degli esperti per molto tempo e l'unico modo per ottenere il

parere degli esperti è rendere pubblici i dettagli" così afferma Bruce Schneier, uno dei guru della moderna crittografia, nel libro "Sicurezza Digitale" (riferimento[1] in bibliografia).

Da un principio puramente tecnico legato alla sicurezza dei sistemi crittografici segue un principio etico legato alla libertà d'informazione con la libera diffusione del software, a mio avviso questa riflessione può essere considerata come un'ulteriore verifica della validità del fenomeno dell'open source.

Il legame crittografia ed open source risulta quindi inevitabile, non si può affidare la sicurezza di un sistema nascondendo i dettagli tecnici, sono tanti gli esempi di sistemi di sicurezza informatici, basati su presunti algoritmi crittografici proprietari che si sono rivelati totalmente insicuri una volta resi noti i codici sorgenti.

La sfida della crittografia è proprio questa, rendere noti i particolari tecnici, i codici sorgenti, ma essere sicuri che nessuno riuscirà a violare il sistema in tempi utili.

Questa apparente contraddizione può essere spiegata solo attraverso una comprensione profonda dei sistemi crittografici, nei quali la sicurezza è affidata alla matematica, il problema è che la matematica è una scienza pura e che molte volte non si adatta bene alla realtà fatta di persone che interagiscono, computer che sono programmati da persone e sui quali viene affidata la sicurezza delle nostre informazioni, "La matematica è assoluta, mentre la realtà è soggettiva. La matematica è qualcosa di ben definito, mentre i computer sono aleatori. La matematica è logica, mentre le persone sono fallibili, capricciose e spesso incomprensibili." prosegue Schneier per cui è bene tenere presente che la crittografia come scienza può essere considerata sicura ma le applicazioni crittografiche, poiché realizzate da esseri umani e non da numeri e teoremi hanno i loro difetti che possono in parte essere eliminati solo grazie all'open source ed alla libera circolazione dei codici sorgenti.

Le basi della crittografia

La crittografia (dal greco kryptos, nascosto, e graphein, scrivere) è la scienza, in parte arte, delle scritture segrete. Un tempo associata ai servizi militari, alle spie, ai vari agenti 007 sparsi in tutto il mondo oggi la crittografia è entrata a far parte della vita quotidiana di tutti noi grazie all'avvento dell'informatica e della crescente potenza di calcolo dei computer.

Nata come raccolta di tecniche e di sistemi per nascondere messaggi tra regnanti, imperatori, amanti, ecc, la crittografia è maturata definitivamente a rango di scienza solo nei primi del 1900 con l'avvento di nuove teorie e tecniche matematiche legate al concetto di informazione.

Le basi teoriche della moderna crittografia, quella attualmente utilizzata, sono ancora più giovani e risalgono a circa 30 anni fa a partire dal 1969 con le prime ricerche di James Ellis (riferimento [3] in bibliografia) del quartier generale governativo delle comunicazioni britanniche (GCHQ). Successivamente sviluppata ed affinata in America grazie al contributo di Whitfield Diffie e Martin Hellman con la nascita del termine crittografia a chiave pubblica e conseguentemente da tre ricercatori del MIT (Massachusetts Institute of Technology), Ronald Rivest, Adi Shamir e Leonard Adleman con la stesura del cifrario RSA (il cui acronimo rappresenta proprio le iniziali dei tre studiosi) che ha rappresentato il vero punto di svolta per le applicazioni pratiche con alte garanzie di sicurezza ed affidabilità, è da questo momento che nasce il termine strong encryption, crittografia forte (riferimento [3] in webografia).

Abbiamo accennato al fatto che la crittografia si occupa delle “scritture segrete” ma cosa sono realmente tali scritture? Sostanzialmente per scrittura segreta si intende una modalità di scrittura non leggibile da chiunque ma, almeno in teoria, solo da chi è in possesso di una informazione, appunto segreta, indicata con il termine chiave (key).

Consideriamo ad esempio la seguente situazione: vogliamo trasmettere un messaggio segreto ad un nostro amico, dal momento che tutti e due conosciamo una particolare forma di geroglifico decidiamo di utilizzare tale “cifrario” per proteggere le nostre comunicazioni. In questo modo la riservatezza del messaggio sarà garantita perché nessuno, tranne io ed il mio amico sarà in grado di tradurre tali strane scritture (a meno di non incappare in qualche archeologo specializzato in antiche scritture egiziane!).

In questa caso la chiave del cifrario, l’informazione segreta, è rappresentata dalla conoscenza del “vocabolario geroglifico” , soltanto chi è in grado di tradurre documenti da questa antica forma di scrittura potrà leggere il contenuto delle comunicazioni.
In questo semplice esempio abbiamo considerato un cifrario particolare nel quale la chiave non costituisce proprio un’informazione segreta, in effetti chiunque può mettersi a studiare il geroglifico e di conseguenza essere in grado di violare le comunicazioni.

Ovviamente nella crittografia moderna si utilizzano sistemi più complessi delle scritture geroglifiche, le tecniche utilizzate sono di natura matematica.

Nella crittografia moderna le chiavi utilizzate per proteggere i documenti sono costituite da sequenze di caratteri di varia lunghezza, di solito si parte da un minimo di 8 fino ad arrivare a 512 ed oltre (rispettivamente 64 e 4096 bit). Questi caratteri utilizzati per la costruzione delle chiavi vengono convertiti in numeri e manipolati attraverso l’utilizzo di formule matematiche per essere convertiti nuovamente in caratteri. In pratica il testo in chiaro viene tradotto in una sequenza di numeri che vengono “manipolati” matematicamente per essere nuovamente convertiti in una sequenza di caratteri rappresentanti il testo cifrato.

Un cifrario dunque è rappresentato da un algoritmo che consente di trasformare un testo in chiaro (il messaggio da proteggere) in un testo cifrato o crittogramma (il messaggio segreto) e viceversa con l'ausilio di una o più chiavi (figura 1).

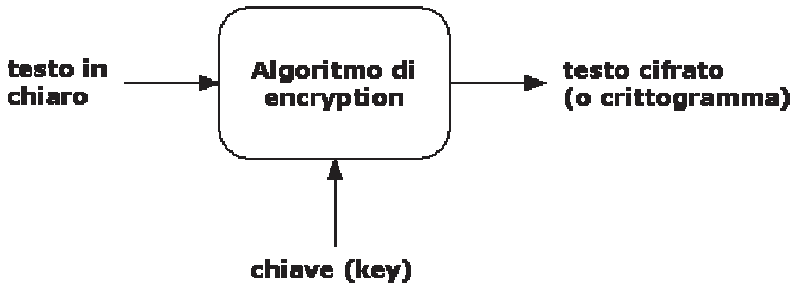


Figura 1

Per poter cifrare documenti di grandi dimensioni il testo in chiaro viene suddiviso in blocchi di caratteri di lunghezza fissa, successivamente ogni blocco viene cifrato ed alla fine i blocchi cifrati vengono riuniti per ottenere l'intero crittogramma, i cifrari di questo tipo vengono chiamati *block chiper*.

La crittografia simmetrica

In crittografia esistono due grandi famiglie di cifrari: quelli simmetrici e quelli asimmetrici o a chiave pubblica. I cifrari simmetrici, storicamente, sono nati per primi grazie allo studio di tecniche manuali di sostituzione e trasposizione delle lettere dell'alfabeto per cercare di "mescolare" il testo in chiaro con l'ausilio di chiavi e tabelle di conversione. I cifrari simmetrici moderni utilizzano, ovviamente, tecniche più sofisticate ma la sostanza dei fatti è la stessa, si tratta sempre di sostituire o permutare un blocco di caratteri, solo che in un cifrario moderno vengono combinate le tecniche decine ed anche centinaia di volte su di uno stesso blocco di caratteri.

Procediamo con ordine con la definizione di cifrario simmetrico: è un sistema di nel quale si utilizza una sola chiave per le operazioni di cifratura e decifrazione.

Nella figura 2 è riportato uno schema a blocchi del funzionamento di un cifrario di tipo simmetrico.

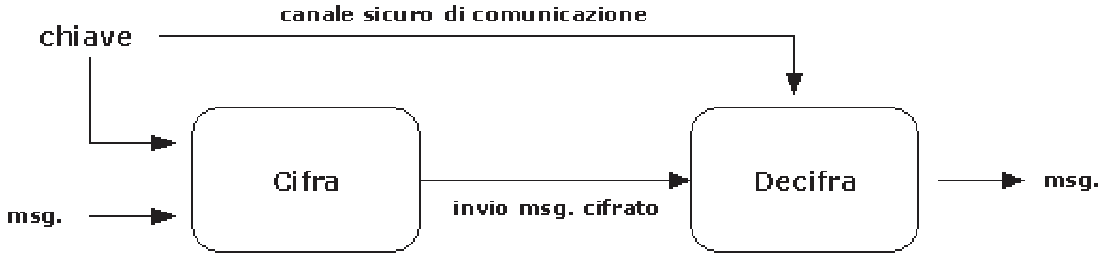


Figura 2

Come è possibile notare affinché il processo di comunicazione vada a buon fine è necessario che sia il mittente che il destinatario siano in possesso della stessa chiave segreta.

La chiave dovrà essere comunicata attraverso un canale sicuro di comunicazione che nella realtà dei fatti non esiste poiché ogni canale di comunicazione è soggetto ad intercettazioni (basti pensare ad Internet o ai sistemi di telefonia mobili).

In realtà lo schema di figura 2 presenta anche un'apparente contraddizione, infatti se siamo in possesso di un canale sicuro di comunicazione poiché mai dovremmo cifrare il messaggio e non trasmetterlo in chiaro utilizzando proprio il canale sicuro di comunicazione?

Supponendo che tale canale sicuro di comunicazione esista di solito esso non avrà la stessa ampiezza di banda di un normale canale di trasmissione e quindi non potrà essere utilizzato per la trasmissione di una grande quantità di informazioni, ecco perché parlo di apparente contraddizione.

Il canale sicuro di comunicazione rappresenta dunque un problema per le comunicazioni sicure punto a punto, ad esempio se due utenti su Internet vogliono comunicare in maniera sicura non possono farlo utilizzando semplicemente un cifrario di tipo simmetrico a meno di incontrarsi fisicamente prima di ogni comunicazione per lo scambio della chiave segreta (anche incontrandosi di persona le cose non cambiano molto, sussurrando la chiave all'orecchio del vostro amico siete sicuri che nessuno intercetti ciò che vi siete detti grazie, ad esempio, ad un potente microfono direzionale? Lo so sono proprio paranoico…).

A maggior ragione pensate ad esempio di dover comunicare con n utenti in maniera sicura attraverso l'utilizzo esclusivo di un cifrario di tipo simmetrico; occorrerà scambiare per ogni comunicazione un numero di chiavi pari a (n-1)*n/2, ad esempio con 100 utenti occorreranno 4950 chiavi differenti.

Come abbiamo visto, i cifrari simmetrici non vengono utilizzati da soli per la protezione delle comunicazioni, per parlare di strong encryption devono essere utilizzati insieme ad altre tecniche crittografiche, come ad esempio i cifrari asimmetrici. Vedremo più avanti un esempio di cifrario ibrido che utilizza proprio una combinazione di queste due tecniche per garantire strong encryption. Il vantaggio dell'utilizzo di cifrari simmetrici è legato soprattutto alla loro elevata velocità di elaborazione.

Esempi di algoritmi simmetrici

Di seguito sono riportati alcuni dei più importanti algoritmi simmetrici utilizzati nell'implementazione delle maggior parte dei sistemi di sicurezza informatica moderni. Se desiderate approfondire i dettagli tecnici di questi algoritmi o conoscerne altri vi consiglio la lettura dei libri [2], [4], [5] e [6] in bibliografia.

DES

Sviluppato dall'IBM nel 1970 è diventato nel 1976 lo standard mondiale per la protezione delle comunicazioni commerciali (FIPS PUB 46-1, 46-2, 81, riferimento[4] in webografia). Il DES è un block chiper a 64 bit, ogni messaggio viene suddiviso in blocchi di 64 bit, ed utilizza una chiave comune di 56 bit. L'algoritmo non introduce alcuna ridondanza nel codice per cui il crittogramma ha la stessa lunghezza del messaggio in chiaro. L'algoritmo di cifratura è basato sull'utilizzo ciclico di 16 reti di Feistel e due permutazioni effettuate all'inizio ed alla fine dell'iterazione principale (figura 3).

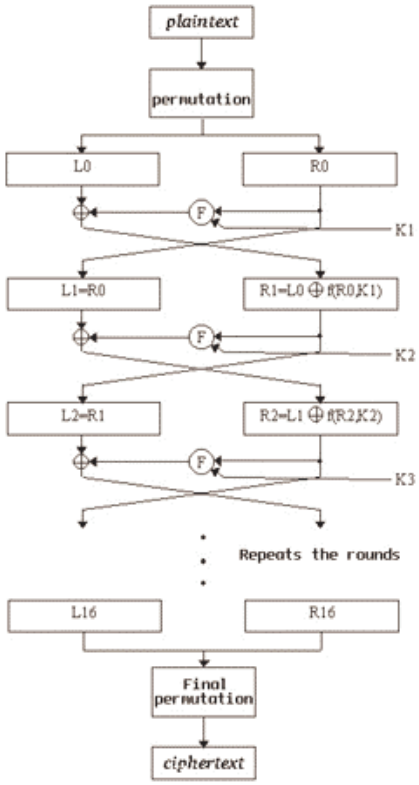


Figura 3

Le reti di Feistel costituiscono uno degli ingredienti fondamentali della maggior parte dei cifrari simmetrici moderni e sono facilmente implementabili poiché costituiti da operazioni di Xor, Shift e loro varianti.

L'idea di base è di suddividere un blocco di 2n bit in due blocchi di n bit denominati L (Left - Sinistra) e R (Right - Destra), successivamente operare delle trasformazioni sul blocco L, lasciando inalterato il blocco R e scambiare alla fine l'ordine dei blocchi. In particolare il risultato di queste

trasformazioni può essere espresso con le seguenti relazioni temporali dove l'indice i indica lo stato attuale e l'indice i+1 lo stato successivo: L(i+1)= R(i) e R(i+1)= L(i) XOR F(R(i),Key), Key è la chiave del cifrario simmetrico (figura 4).

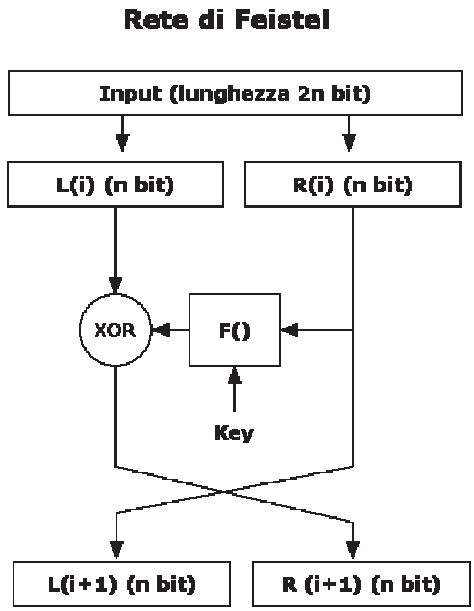


Figura 4

Queste relazioni evidenziano il fatto che il nuovo blocco di sinistra L(i+1) sarà costituito dal vecchio blocco di destra R(i) mentre il nuovo blocco di destra R(i+1) sarà costituito dal risultato dello XOR tra il vecchio blocco di sinistra L(i) e l'output della funzione F(R(i),Key) avente come parametri il vecchio blocco di destra R(i) e la chiave Key.

Operando più volte sui blocchi L e R con più reti di Feistel si ottiene un effetto di "confusione" dei bit del testo in chiaro che vengono mischiati e mascherati più volte con lo scambio dei blocchi L e R.

Il DES è stato utilizzato in tutto il mondo con successo fino al 17 Luglio 1998 quando l'Electronic Frontier Foundation (riferimento [5] in webografia), in collaborazione con la Cryptography Research e l'Advanced Wireless Technologies, ha sviluppato un elaboratore parallelo costituito da 10'368 microchip, denominati Deep Crack, per la generazione a "forza bruta" (brute force) di tutte le possibili chiave a 56 bit utilizzate dall'algoritmo DES (figura 5).

Figura 5

Con l'utilizzo di questo elaboratore dedicato è possibile violare l'algoritmo DES in meno di 56 ore! Il costo della realizzazione dell'intero progetto è stato di circa 250'000 dollari, circa mezzo miliardo delle vecchie lire, costi non proprio alla portata di tutti ma sicuramente alla portata di governi o istituti di ricerca.

Questo potrebbe essere lo spunto per un lungo discorso di tipo politico legato all'e-privacy, qual è lo stato attuale dei progetti Top Secret sulle ricerche crittografiche dei governi di tutto il mondo? O meglio, cosa diavolo

combinano i ricercatori dell'NSA, la National Security Agency americana? Ma questa è un'altra storia (riferimento [6] in webografia).

Morale della favola: non affidate la sicurezza delle vostre informazioni riservate al DES!

3DES

Rappresenta l'evoluzione dell'algoritmo DES attraverso l'utilizzo ripetuto di 3 cifrari DES con una chiave di 112 bit. L'algoritmo è basato sulla tecnica EDE (Encrypt, Decrypt, Encrypt) che consiste nell'utilizzo di un cifrario simmetrico per tre volte consecutive su uno stesso blocco di dati attraverso l'utilizzo di tre operazioni differenti: cifratura con la chiave 1, decifrazione con la chiave 2 e nuova cifratura con la chiave 1 (figura 6).

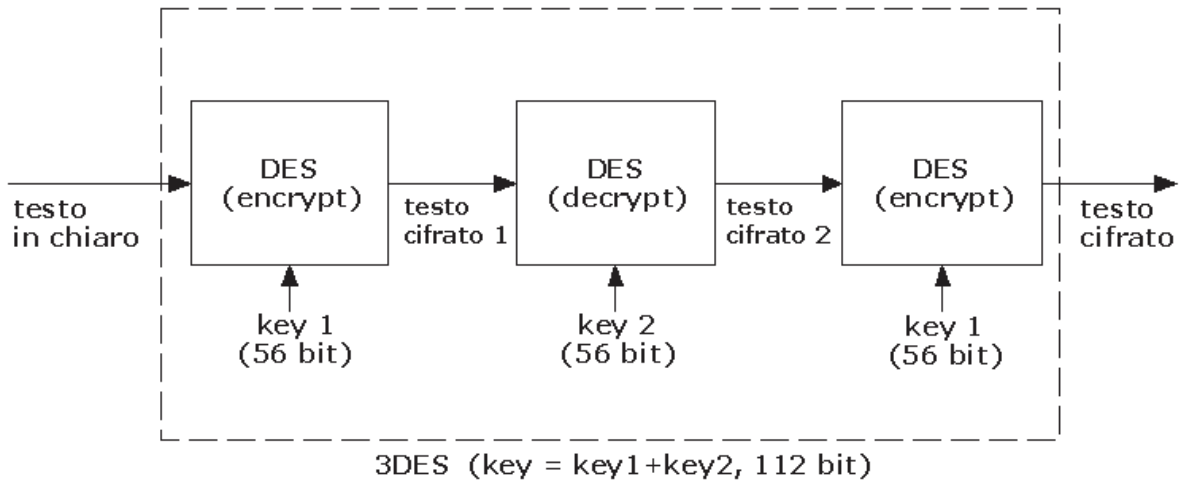


Figura 6

Utilizzando quindi l'algoritmo DES con chiavi di 56 bit, la tecnica EDE consente di ottenere un nuovo cifrario con chiavi di 56+56 = 112 bit. Notate che l'ordine delle operazioni di encryption e decryption all'interno del 3DES non è casuale, si tratta in realtà di tre operazioni distinte di encryption infatti durante la seconda fase di decryption non si ottiene in realtà un testo in chiaro poiché si utilizza la seconda chiave e non la prima per "decifrare", quindi in realtà anche durante la seconda fase dell'algoritmo si può parlare di encryption.

A questo punto si potrebbe pensare di ripetere questa tecnica con più cifrari DES costruendo ad esempio un 13DES in modo da ottenere un cifrario ancora più sicuro con chiavi sempre più lunghe, in realtà non è detto che ripetendo più volte lo stesso algoritmo su di uno stesso blocco di dati si aumenta la sicurezza del sistema. L'affidabilità di un algoritmo di encryption è affidata al modo con il quale vengono "mischiati" i bit e quindi può

essere indipendente dal numero di applicazioni ripetute dell'intero algoritmo, ad esempio con iterazioni eccessive si possono generare degli effetti ciclici sul crittogramma pericolosi per la sicurezza del sistema.

Questa tecnica EDE può essere applicata in realtà anche con algoritmi simmetrici differenti dal DES analizzando di volta in volta i vantaggi sulla sicurezza derivati da queste combinazioni ripetute.

Blowfish

E' un algoritmo ideato nel 1993 da Bruce Schneier per migliorare la velocità di esecuzione del DES. A differenza di quest'ultimo algoritmo il Blowfish non esegue le operazioni di permutazione presenti all'inizio ed alla fine del ciclo di 16 iterazioni del DES e le reti di Feistel utilizzate presentano sostanziali modifiche. E' un algoritmo di encryption veloce, compatto, semplice da implementare e sicuro con chiavi di dimensioni variabili fino a 448 bit. Non si conoscono, attualmente, attacchi efficaci e l'algoritmo può essere utilizzato liberamente essendo non brevettato, ad esempio è utilizzato in molti sistemi open source come OpenBSD.

Rijndael (AES)

Questo algoritmo ha vinto, il 2 Ottobre 2000, la gara indetta dal NIST (National Institute of Standards and Technology) ed è diventato il nuovo standard AES (Advanced Ecnryption Standard) secondo le specifiche FIPS 197 (riferimento [7] in webografia).

Sviluppato Joan Daemen e Vincent Rijmen questo algoritmo utilizza chiavi di lunghezza variabile 128, 192, 256 bit (gli autori hanno dimostrato come è possibile variare le dimensioni delle chiavi con multipli di 32 bit). Lo schema del Rijndael è stato influenzato dall'algoritmo Square.

L'algoritmo lavora su blocchi di dati di 128 bit ed è basato sull'utilizzo di operazioni di XOR e funzioni S-Box ossia delle tabelle di sostituzione. Le operazioni vengono eseguite in un algebra modulo n (campi di Galois) per cui tutte le operazioni possono essere tradotte con le funzioni elementari di XOR e SHIFT. Il numero di cicli dell'algoritmo varia a seconda della lunghezza della chiave: 9 per chiavi da 128 bit, 11 per chiavi da 192 bit, 13 per chiavi da 256 bit (figura 7).

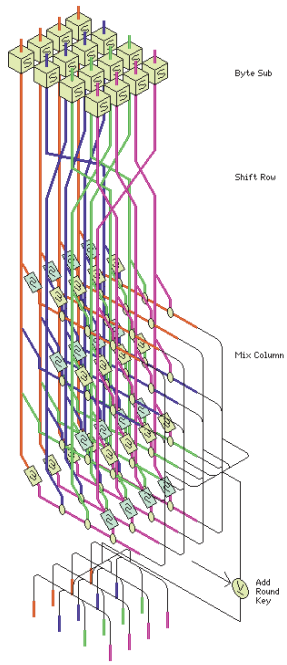


Figura 7

Se siete curiosi di conoscere la pronuncia esatta dell'algoritmo Rijndael potete ascoltarla dalla voce di uno dei due autori all'indirizzo [8] in webografia.

La crittografia asimmetrica o a chiave pubblica

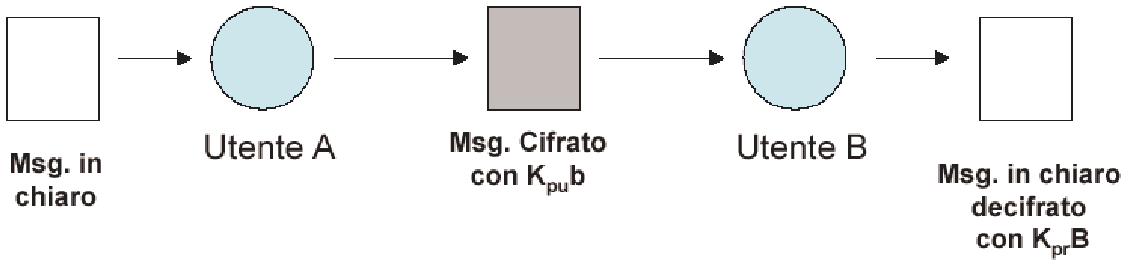
La crittografia asimmetrica è alla base della moderna crittografia ed è particolarmente utilizzata nei sistemi di protezione delle comunicazioni digitali su Internet.

Questo nuovo tipo di approccio consente di eliminare il problema dell'interscambio della chiave segreta, usata nei cifrari simmetrici sia per cifrare che per decifrare i messaggi.

Il problema dello scambio della chiave segreta viene eliminato introducendo una seconda chiave. In pratica si utilizza una chiave per le operazioni di cifratura ed un'altra per le operazioni di decifrazione, la chiave utilizzata per cifrare un messaggio viene resa pubblica (da qui il nome public key) e quindi inviata liberamente a chiunque mentre quella per decifrare rimane privata (da qui il nome private key) e quindi "segreta" di esclusiva proprietà dell'utente.

In questo modo se due persone devono scambiarsi un messaggio non sono costrette a scambiarsi un'informazione segreta, così come avviene per i cifrari simmetrici, basterà inviare le rispettive chiavi pubbliche e cifrare il messaggio utilizzando la chiave del destinatario, non c'è quindi bisogno di tirare in ballo il "canale sicuro di comunicazione" che nella realtà non esiste.

Vediamo di chiarire il concetto utilizzando lo schema di figura 8.



$K_{pu}B$ = chiave pubblica dell'utente B

$K_{pr}B$ = chiave privata dell'utente B

Figura 8

Due utenti A e B vogliono comunicare attraverso un canale insicuro di comunicazione, ad esempio Internet. L'utente A desidera inviare un messaggio protetto all'utente B, per fare ciò utilizza la chiave pubblica dell'utente B (KpuB) cifrando il messaggio con un algoritmo di tipo asimmetrico.

Il messaggio viene spedito sul canale insicuro di comunicazione e quando arriverà all'utente B solo lui potrà decifrarlo utilizzando la propria chiave privata (KprB).

La chiave pubblica di ogni utente può circolare tranquillamente sul canale insicuro di comunicazione un po' come avviene con i numeri di telefono delle persone, esistono infatti dei siti Internet, denominati Key Server, contenenti l'elenco delle chiavi pubbliche di moltissimi utenti un po' come avviene con le pagine gialle telefoniche.

Abbiamo visto come proteggere le comunicazioni da occhi indiscreti cifrando i messaggi con la chiave pubblica del destinatario, la crittografia asimmetrica consente anche operazioni di autenticazione dei messaggi attraverso l'utilizzo di una sorta di "firma digitale" del pacchetto dati da inviare (in realtà non si tratta di una vera e propria firma digitale come viene comunemente intesa, più avanti nell'articolo definiremo meglio questo concetto).

Per illustrare questa tecnica utilizziamo ancora una volta uno schema grafico, la figura 9.

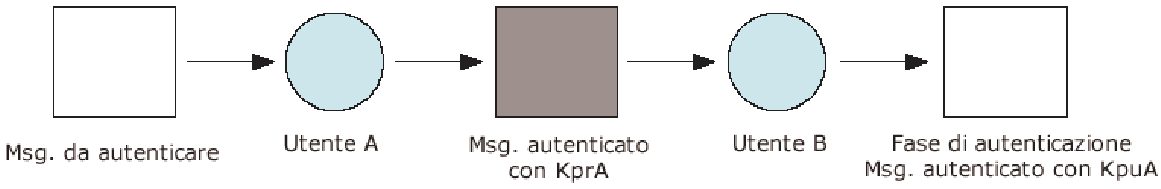


Figura 9

Un utente A vuole inviare un messaggio all'utente B e l'utente B vuole essere sicuro che il messaggio sia stato inviato dall'utente A. Il messaggio non deve essere cifrato occorre solo eseguire un'operazione di autenticazione. Per autenticare il messaggio l'utente A utilizza l'algoritmo asimmetrico con la propria chiave privata KprA, invia il messaggio all'utente B che utilizza la chiave pubblica di A, KpuA, per verificare l'originalità del mittente.

A differenza dello schema di figura 8 nel quale l'utente A cifrava il messaggio con la chiave pubblica dell'utente B in questo schema si utilizzano esclusivamente la chiave privata e quella pubblica dell'utente A. Infatti solo l'utente A è a conoscenza della chiave privata KprA e quindi solo lui potrà utilizzare questa chiave per autenticare il messaggio, d'altronde tutti gli altri utenti, compreso B, possono verificare l'origine del messaggio semplicemente utilizzando la chiave pubblica di A, KpuA (combinando in cascata le operazioni dei due schemi di autenticazione e cifratura, figure 8 e 9, si ottengono i due effetti contemporaneamente, la protezione e l'autenticità del messaggio).

In effetti le due chiavi, pubblica e privata di un utente sono strettamente collegate tra di loro attraverso una relazione matematica sicura, attualmente inespugnabile.

Cerchiamo di approfondire il legame tra chiave pubblica e chiave privata con una domanda provocatoria: dal momento che la chiave privata è legata, in qualche modo, alla chiave pubblica di una persona, altrimenti non sarebbe possibile decifrare il msg. cifrato con la chiave pubblica corrispondente, è possibile ricavare la chiave privata da quella pubblica?

Per rispondere a questa domanda entrano in gioco la matematica e gli algoritmi utilizzati per la costruzione del cifrario asimmetrico. La stabilità di un algoritmo asimmetrico dipende proprio da quest'ultimo punto. L'impossibilità o meglio la complessità computazionale derivante dal calcolo della chiave privata da quella pubblica è strettamente collegata alla risoluzione di un problema matematico complesso come ad esempio la fattorizzazione di numeri "grandi" (il concetto di numero "grande" ovviamente è relativo, al momento in cui scrivo per numero grande, in ambito crittografico, si intende un numero dell'ordine di 200 o più cifre).

Attualmente la sicurezza della maggior parte dei sistemi crittografici è affidata al problema della fattorizzazione dei grandi numeri, allo stato attuale non si conosce un algoritmo efficiente, ossia in grado di dare una risposta in tempi ragionevoli, che riesca a scomporre un numero di dimensioni elevate in fattori primi (per approfondire i concetti matematici legati alla teoria dei numeri in crittografia consiglio i riferimenti [7], [8] e [9] in bibliografia). Questo è un esempio di sicurezza garantita dalla matematica, fin quando non si scoprirà un algoritmo efficiente per la fattorizzazione dei grandi numeri i sistemi crittografici basati su questa tecnica risulteranno sicuri, a meno di non utilizzare calcolatori quantistici in grado di elevare la potenza di calcolo di un fattore esponenziale ma per fortuna questi elaboratori ancora non esistono anche se le ricerche in laboratorio iniziano a dare i primi risultati (vedi i riferimenti [9] e [10] in webografia).

Attraverso l'utilizzo dei cifrari asimmetrici sono nati i moderni sistemi PKI (Public Key Infrastructure) ossia dei sistemi per la protezione delle comunicazioni nei quali vengono utilizzati uno o più cifrari asimmetrici.

All'interno dei sistemi PKI rivestono un ruolo fondamentale le Certification Authority (CA) che garantiscono la corrispondenza biunivoca tra chiavi pubbliche ed utenti.

Chi mi garantisce che la chiave pubblica di una persona, che nella sostanza dei fatti è un normale file, corrisponda effettivamente alla legittima persona?

L'importanza delle CA appare evidente con l'utilizzo della firma digitale. Per poter garantire l'autenticità ed il non ripudio di un messaggio firmato è indispensabile avere a disposizione una CA che garantisca, appunto, che il messaggio firmato appartenga effettivamente alla persona in possesso delle corrispondenti chiavi pubbliche e private.

Nel proseguo dell'articolo verrà introdotto il concetto di firma digitale ed il ruolo delle CA nello specifico attraverso un esempio di utilizzo del certificato digitale nel processo di firma di un documento.

Esempi di algoritmi asimmetrici

Di seguito sono riportati alcuni degli algoritmi asimmetrici più conosciuti basati sull'utilizzo di concetti di teoria dei numeri come il prodotto di numeri primi e il calcolo del logaritmo discreto.

Molti dei nuovi algoritmi asimmetrici moderni utilizzano concetti matematici sempre più sofisticati come la teoria delle curve ellittiche. In questa sede non analizzeremo in dettaglio tutte le tecniche matematiche, il nostro obiettivo è quello di introdurre i concetti essenziali senza però rinunciare al giusto rigore scientifico, la crittografia d'altronde è matematica applicata.

Diffie-Hellman

Questo algoritmo è stato creato nel 1976 da W.Diffie e M.Hellman per risolvere il problema dello scambio delle chiavi senza l'ausilio di un canale sicuro di comunicazione, secondo lo schema già introdotto con la figura 2 quando si parlava di crittografia simmetrica. E' il primo algoritmo a chiave pubblica della storia ed è basato sul calcolo del logaritmo discreto la cui complessità computazionale è improponibile. Senza entrare nei dettagli dell'algoritmo possiamo utilizzare l'esempio della borsa e del lucchetto, famoso in crittografia, per spiegare il principio di funzionamento dell'algoritmo Diffie-Hellman.

Immaginiamo la seguente situazione (figura 10). Due utenti A e B vogliono scambiarsi un messaggio segreto utilizzando un canale di comunicazione insicuro, ad esempio le tradizionali poste italiane. L'utente A inserisce il suo messaggio segreto in una

ventiquattrore e la sigilla inserendo un lucchetto nella maniglia, spedisce poi il pacco all'utente B. L'utente B una volta ricevuto il pacco provvede a sigillare ulteriormente la maniglia della ventiquattrore con un secondo lucchetto di sua proprietà, dopo questa operazione invia il pacco all'utente A. L'utente A ricevuto il pacco provvede a togliere il suo lucchetto attraverso la chiave di sua proprietà e rinvia il pacco all'utente B. A questo punto l'utente B è in grado di aprire finalmente la ventiquattrore utilizzando la sua chiave sull'unico lucchetto rimasto.

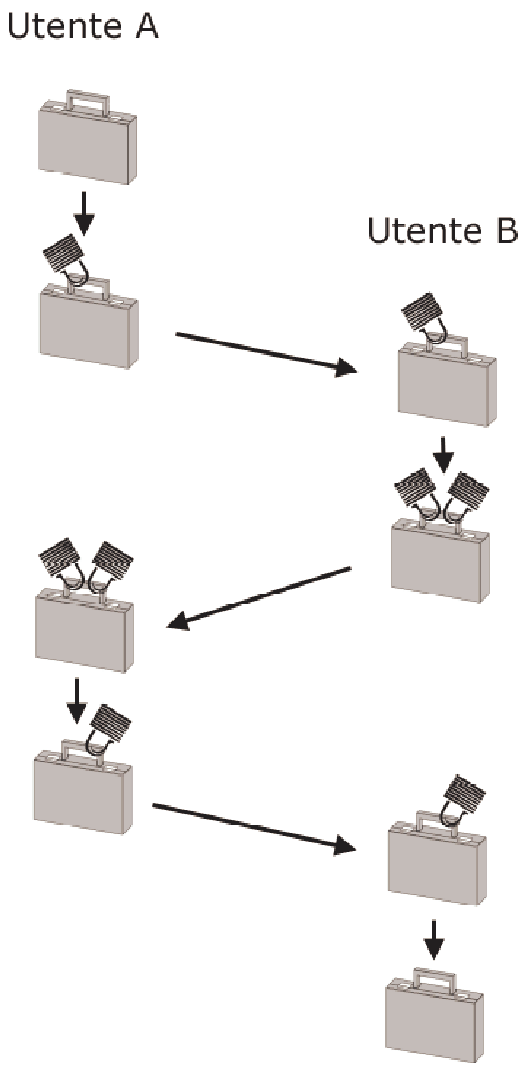


Figura 10

Le implicazioni di questo breve esempio, derivato dal mondo reale, sono profonde. Esso dimostra che un messaggio segreto può essere trasmesso su di un canale insicuro di comunicazione senza dover comunicare preventivamente una chiave, lo schema di figura 2 risulta quindi superato.

Utilizzando relazioni matematiche relative a potenze su di una particolare algebra (campi di Galois) i due ricercatori W.Diffie e M.Hellman sono riusciti a simulare il comportamento dei lucchetti e delle relative chiavi introdotte nell'esempio precedente della ventiquattrore per ottenere uno schema affidabile per l'interscambio di informazioni sensibili (per un'introduzione delle tecniche matematiche utilizzate in quest'algoritmo vedi il riferimento [11] in webografia).

Questo tipo di tecnica viene utilizzata nella pratica per lo scambio di chiavi di cifratura di tipo simmetrico, una volta che due nodi di una rete sono in possesso di una stessa chiave di encryption possono tranquillamente utilizzare un algoritmo veloce di cifratura simmetrica per la protezione delle loro comunicazioni.

RSA

L'algoritmo RSA è il più famoso ed il più utilizzato algoritmo a chiave pubblica nella storia della crittografia. E' stato ideato nel 1977 dai ricercatori del MIT (Massachusetts Institute of Technology) Ronald Rivest, Adi Shamir e Leonard Adleman (figura 11).

Figura 11

L'algoritmo è basato sul problema della fattorizzazione di numeri di dimensioni elevate già introdotto in precedenza. Utilizza chiavi di lunghezza variabile a 512, 1024, 2048, 4096 bit ed oltre.

Applicazioni e servizi Internet utilizzano frequentemente chiavi pubbliche e private RSA di 1024 bit anche se ultimamente, il 15 Aprile 2002, Bruce Schneier in un articolo della sua newsletter Crypto-Gram intitolato "Is 1024 Bits Enough?" suggerisce la lunghezza futura delle chiavi per gli algoritmi a chiave pubblica ed in particolare per l'anno 2005 si parla di 1280, 1536 e 2048 bit per la protezione rispettivamente delle comunicazioni personali, aziendali e governative (riferimento [12] in webografia).

Il 6 Settembre 2000 l'algoritmo RSA è diventato di dominio pubblico, prima era di proprietà dell'omonima azienda RSA Security Inc (riferimento [3] in webografia).

L'algoritmo RSA oltre ai molti vantaggi legati all'affidabilità ed alla sicurezza del sistema presenta almeno uno svantaggio legato alla velocità di elaborazione dei dati, infatti viene utilizzato soprattutto nei sistemi crittografici ibridi che utilizzano contemporaneamente sia algoritmi simmetrici che algoritmi a chiave pubblica (come ad esempio nei software PGP e GNUPG).

Per capire il funzionamento dell'RSA si può far riferimento all'approfondimento "Il cifrario RSA" presente in un riquadro di questa pagina.

I sistemi crittografici ibridi

Nei sistemi di sicurezza informatici vengono utilizzate spesso tecniche combinate di crittografia simmetrica e di crittografia asimmetrica o a chiave pubblica.

Il motivo principale di questo utilizzo di sistemi ibridi è legato alle performance in termini di velocità di elaborazione di questi sistemi.

I cifrari a chiave pubblica sono molto più lenti di quelli a chiave simmetrica, per cui vengono utilizzati molte volte i primi per scambiare o cifrare le chiavi segrete utilizzate dai secondi, gli algoritmi simmetrici, per la cifratura dei dati.

L'encryption dei dati di questi sistemi è affidato quindi agli algoritmi simmetrici mentre l'autenticazione è affidata agli algoritmi a chiave pubblica.

Vediamo, ad esempio, il principio di funzionamento di un sistema crittografico ibrido, uno dei software per la privacy personale più conosciuti al mondo il PGP (Pretty Good Privacy) di Philip Zimmermann (riferimento [13] in webografia).

Utilizziamo sempre come riferimento uno schema grafico riportato in figura 12.

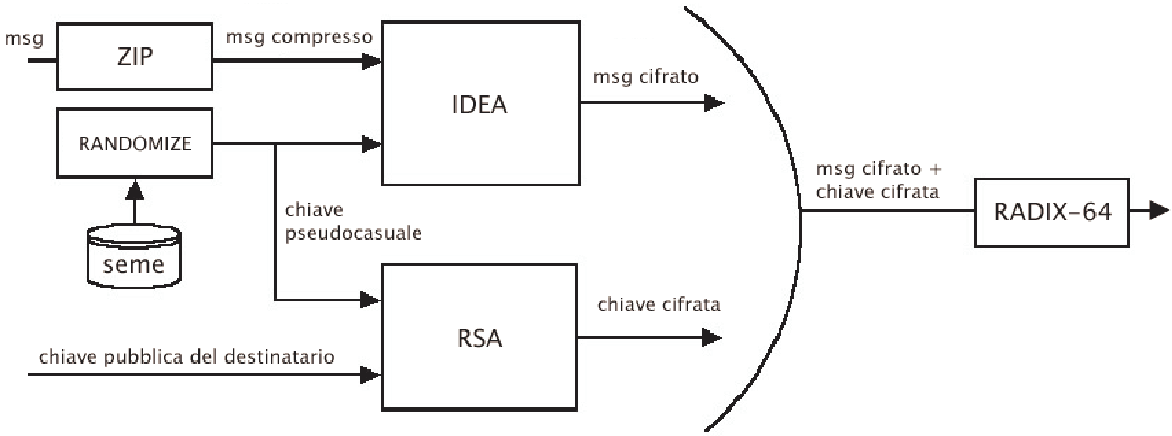


Figura 12

Il PGP è un programma che consente di cifrare documenti e messaggi di posta elettronica utilizzando due algoritmi crittografici, uno di tipo simmetrico (ad esempio l'IDEA) ed uno di tipo asimmetrico (ad esempio l'RSA). Lo schema riportato in figura 12 è relativo ad un'operazione di encryption di un messaggio per l'invio tramite posta elettronica.

Le prima operazione che viene effettuata sul messaggio in chiaro è una compressione tramite l'utilizzo dell'algoritmo ZIP, l'operazione di compressione è tipica dei software di encryption poiché gli algoritmi crittografici sono più sicuri se applicati su sorgenti con maggiore entropia.

Successivamente viene generata una chiave (key) pseudocasuale tramite l'utilizzo di un algoritmo Randomize estrapolando dei dati da una sorgente casuale, il seme che solitamente è un file contenente dei numeri casuali preconfezionati.

La funzione di randomizzazione dei dati utilizzata generalmente in crittografia non è quella classica presente nei linguaggi di programmazione più comuni, ad esempio la funzione rand() del linguaggio C, si preferisce l'utilizzo di algoritmi più complessi per motivi di sicurezza legati al fatto che le funzioni standard non sono particolarmente efficienti nella generazione di numeri pseudocasuali, alla lunga esse presentano delle ciclicità.

La chiave pseudocasuale così generata viene utilizzata tramite un algoritmo di tipo asimmetrico, che nel caso del PGP è rappresentato dall'algoritmo IDEA (un algoritmo a blocchi di 64 bit con chiavi di 128 bit), per cifrare l'intero messaggio compresso.

Contemporaneamente si utilizza l'algoritmo a chiave pubblica RSA per cifrare la chiave pseudocasuale precedentemente generata tramite la chiave pubblica del destinatario del messaggio.

In questo modo solo il destinatario del messaggio, in possesso della sua chiave privata sarà in grado di decifrare la chiave pseudocasuale ed utilizzarla per decifrare il messaggio tramite l'algoritmo IDEA.

Prima di inviare i dati cifrati corrispondenti al messaggio ed alla chiave pseudocasuale il PGP utilizza una funzione, denominata nello schema a blocchi, RADIX-64 che consente di convertire l'alfabeto del messaggio in uno standard internazionale di 64 caratteri per evitare confusioni con altri set di caratteri utilizzati su differenti elaboratori.

Dunque il PGP anche se viene spesso citato come uno dei programmi più famosi di crittografia a chiave pubblica in realtà è un software che utilizza un algoritmo di tipo simmetrico per la cifratura dei dati, l'algoritmo a chiave pubblica presente nel PGP viene utilizzato soltanto per cifrare la chiave simmetrica pseudocasuale generata di volta in volta e differente per ogni comunicazione.

Enrico Zimuel
consulenza informatica

La firma digitale e le funzioni hash sicure

Uno degli aspetti più conosciuti della crittografia a chiave pubblica soprattutto per le sue implicazioni di carattere legale è la firma digitale, vedi il DPR 10 novembre 1997, n. 513 e le nuove direttive europee 1999/93/CE (riferimento [10] in bibliografia, [14] in webografia).

La firma digitale consiste in un procedimento matematico che consente di legare un documento elettronico al suo legittimo proprietario attraverso l'utilizzo di una sequenza univoca di numeri binari, denominata appunto firma digitale.

La firma digitale nasce dall'utilizzo della crittografia asimmetrica poiché per consentire l'autenticità del documento firmato viene utilizzato un algoritmo a chiave pubblica tramite la chiave privata dell'utente firmatario. A differenza quindi del procedimento di cifratura di un documento nel quale l'utente utilizza la chiave pubblica del destinatario del messaggio per la protezione dello stesso, nel caso del processo di firma di un documento l'utente utilizzerà la propria chiave privata per firmare il messaggio. Si tenga presente che in quest'ultimo caso non si ha la cifratura del messaggio, quindi esso non risulta protetto da occhi indiscreti, il processo di firma serve solo per garantire l'integrità e l'autenticità del messaggio, o meglio l'autenticità dell'utilizzo di una particolare coppia di chiavi pubbliche e private.

Dal punto di vista teorico basterà applicare tale procedimento sull'intero documento con la propria chiave privata per generarne una firma elettronica, infatti chiunque attraverso l'utilizzo della chiave pubblica dell'utente firmatario potrà verificare l'autenticità dello stesso attraverso un procedimento matematico inverso rispetto al precedente.

Dal punto di vista pratico la firma digitale viene applicata soltanto ad un "sunto" del documento, questo per evitare dei tempi di elaborazione troppo lunghi nel rilascio della firma, dipendenti dalla complessità computazionale degli algoritmi asimmetrici, e soprattutto per evitare che la firma digitale occupi più spazio, in termini di byte, del documento stesso.

Per questo motivo si utilizzano le funzioni hash che consentono di concentrare l'unicità di un documento in poche centinaia di byte o caratteri. In pratica queste funzioni matematiche consentono di generare una sorta di targa univoca di un documento digitale. L'output di una funzione hash, rappresentato da una sequenza di byte di lunghezza fissa, è strettamente legato, in maniera biunivoca, con il documento digitale in input.

Ciò vuol dire che il risultato di una stessa funzione hash applicata su due documenti diversi deve essere differente anche se i due documenti differiscono di un solo byte.

Una funzione hash per definirsi sicura deve rispettare le seguenti condizioni (indico con H(M) la funzione hash applicata ad un messaggio M):

1) H può essere applicata a un blocco di dati di qualsiasi dimensione;
2) H produce in uscita un risultato di lunghezza fissa (ad esempio 160 bit);
3) per qualunque codice h il calcolo di x tale che H(x)=h deve avere una complessità computazionale improponibile;
4) per qualunque blocco di dati x il calcolo di y diverso da x tale che H(x)=H(y) deve avere una complessità computazionale improponibile;

Inoltre H(x) deve essere relativamente semplice da calcolare (vedi il riquadro di pagina relativo ad un esempio di funzione hash).

Ritornando al discorso della firma digitale possiamo sfruttare il risultato di una funzione hash come "sunto" del documento ed applicare il procedimento di firma, descritto all'inizio del paragrafo, solo su tale sequenza di byte rappresentativa del documento. E proprio questo il procedimento che viene utilizzato dalla maggioranza dei software in circolazione per il rilascio della firma digitale.

In pratica utilizzando questi procedimenti di firma su di un documento digitale otterrò una sequenza di byte di lunghezza fissa strettamente legata al documento stesso. Tale firma potrà essere inviata in chiaro, ad esempio, in coda al documento come se fosse stato firmato in maniera tradizionale utilizzando carta e penna.

Tale procedimento matematico consente di garantire l'autenticità, l'integrità ed il non ripudio del documento firmato elettronicamente.

Per la legge italiana, una firma digitale per risultare valida dovrà essere convalidata attraverso una Certification Authority tramite il rilascio di un certificato digitale che attesti la corrispondenza biunivoca tra la chiave pubblica e le persona fisica.
Vediamo nel dettaglio come avviene il rilascio del certificato digitale con l'utilizzo di una CA.

Il primo passo da realizzare è il rilascio del certificato digitale della CA. Questa operazione viene eseguita dopo aver inviato i dati personali dell'utente, comprensivi della chiave pubblica, alla CA.

La Certification Authority dopo aver verificato la correttezza dei dati relativi all'utente inserirà questi valori nel proprio archivio assegnando all'utente un codice identificativo (ID). Il rilascio del certificato digitale avviene dopo aver apposto la firma, utilizzando la chiave privata della CA, sui dati relativi all'utente comprensivi di ID e chiave pubblica (vedi figura 13).

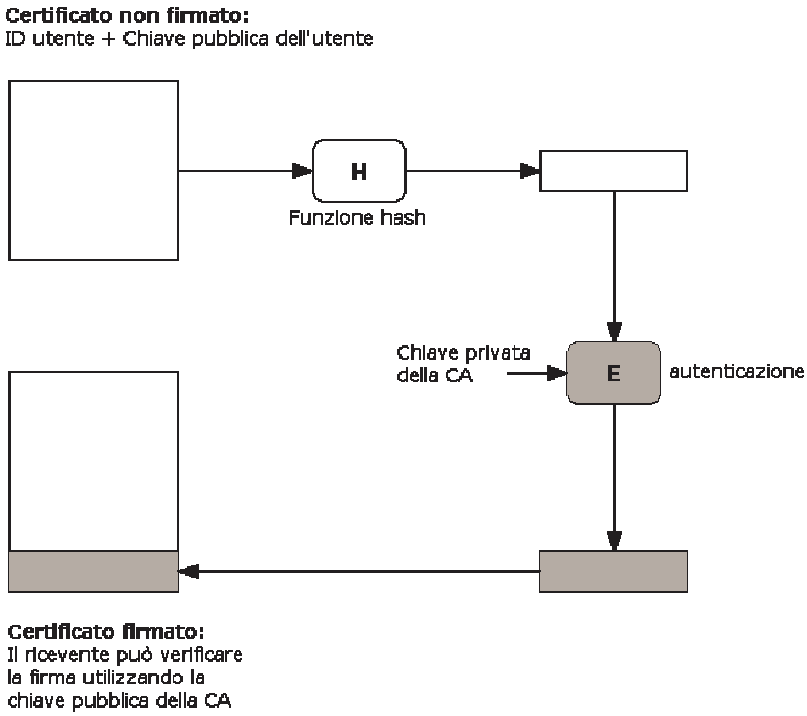


Figura 13

La verifica del certificato digitale da parte del destinatario avverrà semplicemente utilizzando la chiave pubblica della CA, in questo modo si verificherà l'autenticità della chiave pubblica dell'utente (compresa nel certificato) che potrà essere utilizzata per verificare a sua volta la firma digitale del messaggio originale del mittente.

Conclusione

In questo articolo abbiamo introdotto i concetti fondamentali della moderna crittografia cercando di dimostrare che la crittografia deve essere di tipo open source per non violare un principio fondamentale della sicurezza introdotto più di un secolo fa da August Kerckhoffs.

I software open source che si ispirano a questo principio sono sempre più numerosi ed anche gli standard internazionali con codici aperti stanno diventando sempre più frequenti, vedi ad esempio l'AES.

Negli articoli futuri parleremo di alcuni dei progetti crittografici open source più conosciuti come il GnuPG per la privacy personale, l'OpenCA per la creazione di Certification Authority, l'OpenSSL per la protezione delle comunicazioni http, l'OpenSSH per la

protezione delle sessioni telnet, il CIPE Crypto IP Encapsulation per la protezione a livello IP, il FreeS/WAN per l'implementazione dei protocolli IPSEC e IKE, e molti altri ancora.

Bibliografia

[1] "Sicurezza Digitale" di Bruce Schneier, Tecniche Nuove, 2001.

[2] "Applied Cryptography – second edition" di Bruce Schneier, John Wiley & Sons Inc, 1996.

[3] "Codici & Sergeti" di Simon Singh, Rizzoli, 1999.

[4] "Sicurezza delle reti - Applicazioni e standard"di William Stallings, Addison-Wesley Editore, 2001.

[5] "Sicurezza dei sistemi informatici" di M.Fugini, F.Maio, P.Plebani, Apogeo Editore, 2001.

[6] "Internet Security" di M. Cinotti, Hoepli Editore, 2002.

[7] "Aritmetica Superiore" di H. Davenport, Zanichelli Editore, 1994.

[8] "Crittografia" di Andrea Sgarro, Franco Muzzio Editore, 1993.

[9] "A Course in Number Theory and Cryptography" di Neal I. Koblitz, Springer Verlag Editore, 1994.

[10] "La firma digitale e il documento informatico" di Giorgio Rognetta, Simone Editore, 1999

Webografia

[1] "La cryptographie militaire" di August Kerckhoffs, http://www.enricozimuel.net/documenti/crypto_militaire_1.pdf

[2] Charles Babbage Institute, http://www.cbi.umn.edu

[3] Il sito della società Rsa Security Inc. nata grazie alla creazione dell'algoritmo Rsa, http://www.rsasecurity.com/

[4] Informazioni e standard sul DES, http://csrc.nist.gov/cryptval/des.htm

[5] Il progetto DES Cracking dell'EFF, http://www.eff.org/descracker.html

[6] Il sito dell'NSA, la National Security Agency americana, http://www.nsa.gov

[7] AES FIPS 197, http://csrc.nist.gov/encryption/aes/

[8] Pronuncia del Rijndael,http://www.enricozimuel.net/algoritmi/rijndael_pronunciation.wav

[9] Bibliografia sulla crittografia quantistica, http://www.cs.mcgill.ca/~crepeau/CRYPTO/Biblio-QC.html

[10] Alcuni links sui calcolatori quantistici, http://eve.physics.ox.ac.uk/Links/QC.Links.html

[11] L'algoritmo Diffie-Hellman, http://www.enricozimuel.net/documenti/DiffieHellman.pdf

[12] La newsletter Crypto-Gram di Bruce Schneier, http://www.counterpane.com/crypto-gram.html

[13] Il sito di Philip Zimmermann l'ideatore del PGP, http://www.philzimmermann.com/

[14] Informazioni sulla firma digitale della rivista on-line Interlex, http://www.interlex.it/docdigit/indice.htm