Un'introduzione alla crittografia Open Source

di Enrico Zimuel (cerin0@blackhats.it)

Smau 2002 Ethical Hacker's Speech II

26 Ottobre 2002

(www.blackhats.it)

Copyright

Sommario

- Perchè crittografia ed open source?
- Definizioni di base
- La crittografia simmetrica o a chiave segreta
- Il problema della trasmissione della chiave
- Esempi di cifrari simmetrici: DES, 3DES, Blowfish, Rijndael
- La crittografia asimmetrica o a chiave pubblica
- Esempi di cifrari asimmetrici: RSA, Diffie-Hellman.
- La firma digitale e le funzioni hash sicure
- La crittoanalisi, tecniche, Des Cracking, Security Flaw

La crittografia è Open Source

- Risulterà strano ma uno dei principi fondamentali della crittografia, utilizzato ancora nei moderni sistemi crittografici è stato individuato nel lontano 1883 dal linguista franco-olandese August Kerckhoffs nel suo celebre articolo “La cryptographie militaire” apparso nel Journal des sciences militaires.
- Principio di Kerckhoffs: *“La sicurezza di un sistema crittografico è basata **esclusivamente** sulla conoscenza della chiave, in pratica si presuppone noto a priori l’algoritmo di cifratura e decifrazione.”*
- Conoscenza algoritmo = libera distribuzione codici sorgenti = Open Source!

La crittografia è Open Source

- “Se un sistema è veramente sicuro, lo è anche quando i dettagli divengono pubblici” B.Schneier
- Sicurezza = Trasperenza !
- Questa apparente contraddizione può essere spiegata solo grazie all'ausilio della matematica come base teorica della crittografia.
- La sicurezza di un sistema crittografico è intrinseca al sistema poiché basata su principi matematici.
- Sicurezza teorica = Sicurezza pratica? Purtroppo NO, i problemi sorgono in fase di applicazione dei concetti teorici (ad esempio esiste un algoritmo teoricamente sicuro, l'algoritmo di Vernam, ma non può essere implementato correttamente).

Che cos'è un cifrario?

- Un cifrario è un sistema, di qualsiasi tipo, in grado di trasformare un testo in chiaro (messaggio) in un testo inintellegibile (testo cifrato o crittogramma).

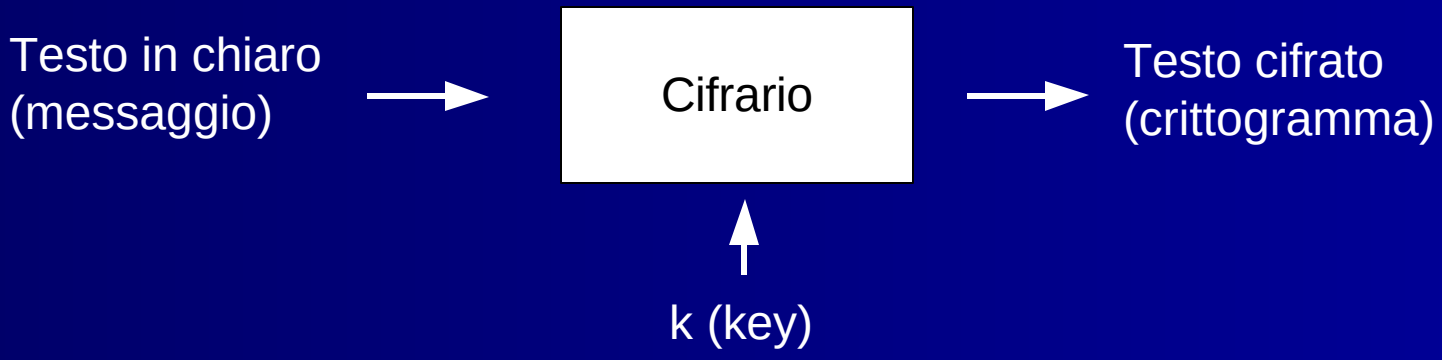
Per poter utilizzare un cifrario è necessario definire due operazioni: la cifratura del messaggio e la decifrazione del crittogramma.

Le operazioni di cifratura e decifrazione

- Definiamo con **Msg** “l'insieme di tutti i messaggi” e con **Critto** “l'insieme di tutti i crittogrammi”.
- ***Cifratura***: operazione con cui si trasforma un generico messaggio in chiaro **m** in un crittogramma **c** applicando una funzione **C**: Msg->Critto.
- ***Decifrazione***: operazione che permette di ricavare il messaggio in chiaro **m** a partire dal crittogramma **c** applicando una funzione **D**: Critto -> Msg.
- Matematicamente D(C(m))=m le funzioni C e D sono una inversa dell'altra e la funzione C deve essere iniettiva, ossia a messaggi diversi devono corrispondere crittogrammi diversi.

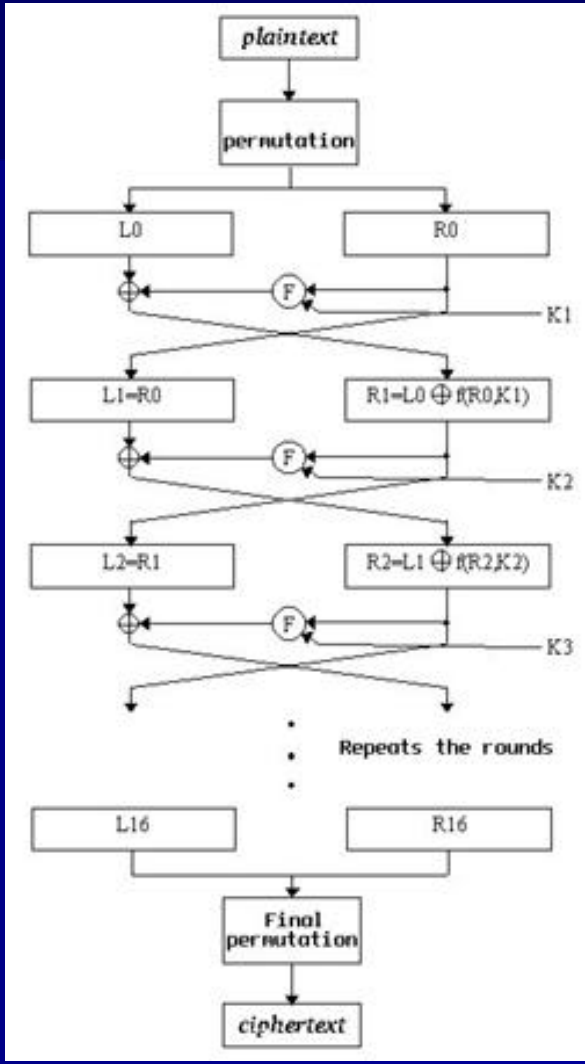
La crittografia simmetrica

- Introduciamo un parametro chiamato **k** (key= chiave) all'interno delle funzioni di cifratura **C(m,k)** e decifrazione **D(c,k)**.
- Si parla di crittografia simmetrica perchè si utilizza la stessa chiave k per le operazioni di cifratura e decifrazione.
- La robustezza del cifrario dipende, a differenza del modello precedente, solo dalla segretezza della chiave **k**.



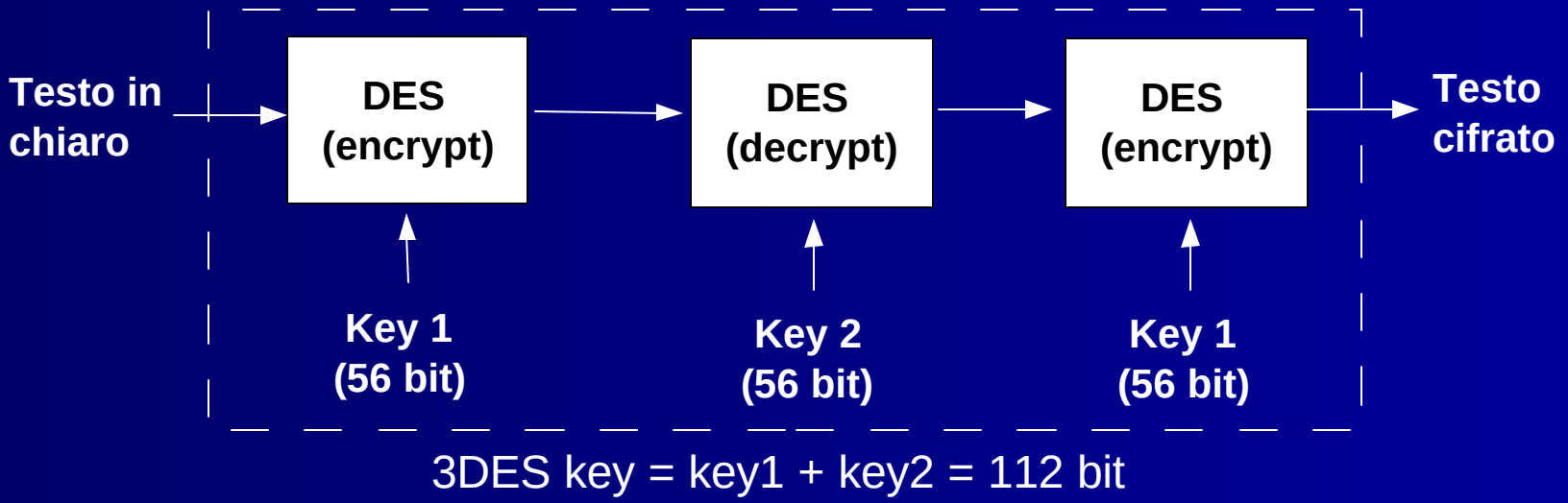
DES (Data Encryption Standard)

- Sviluppato dall'IBM nel 1970 diventato standard nel 1976.
- Utilizza chiavi di 56 bit, divide il testo in chiaro in blocchi di 64 bit, effettua delle permutazioni iniziali e finali ed un ciclo di 16 iterazioni di permutazioni e xor (Feistel network, tecniche di confusione e diffusione).
- Il 17 Luglio 1998, l'EFF (Electronic Frontier Foundation) costruisce un sistema dedicato in grado di violareill DES in meno di 3 giorni, tramite un attacco di tipo "brute-force".
- Morale della favola: non utilizzate sistemi di cifratura basati sul DES!



3DES

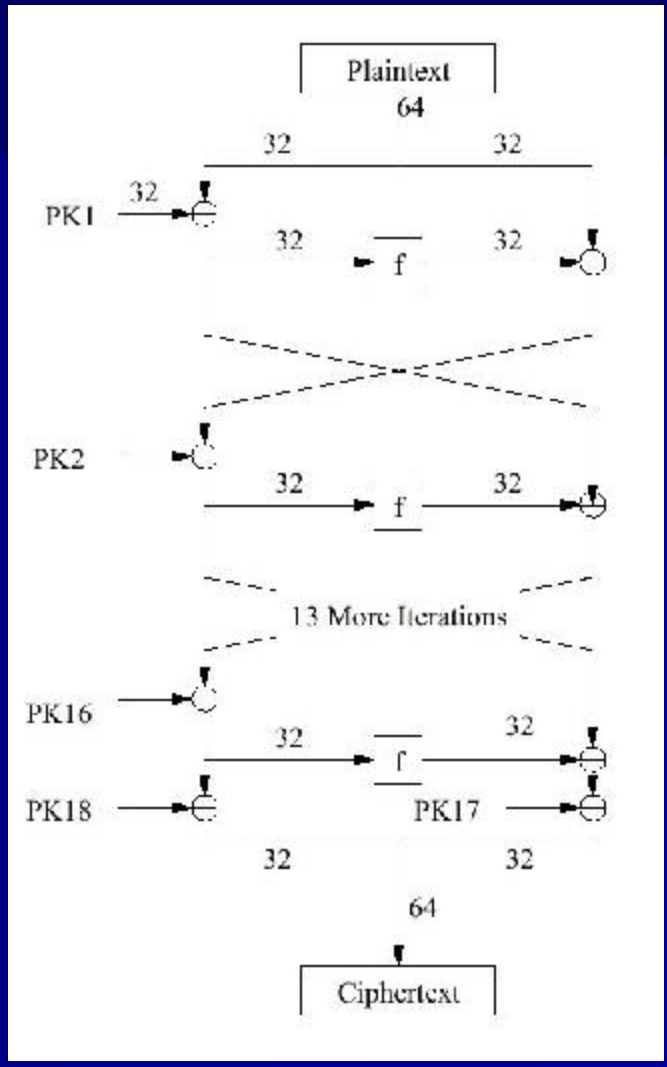
- Evoluzione del DES, è basato su un utilizzo del cifrario DES ripetuto, chiavi di 112 bit.
- Si utilizza la tecnica della codifica-decodifica-codifica (EDE, Encrypt-Decrypt-Encrypt) utilizzando il cifrario DES.



3DES key = key1 + key2 = 112 bit

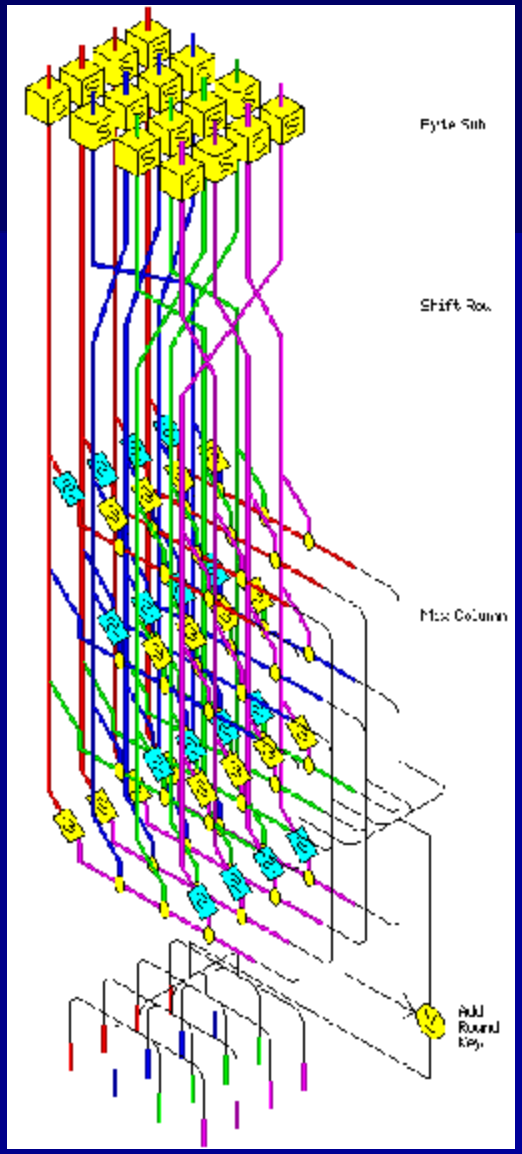
Blowfish

- Ideato nel 1993 da Bruce Schneier.
- E' stato sviluppato come algoritmo di encryption: veloce, compatto, semplice da implementare e sicuro con chiavi di dimensioni variabili fino a 448 bit.
- E' un cifrario a blocchi di 64 bit, basato sulle reti di Feistel.
- Non si conoscono attacchi efficaci.
- E' un algoritmo non patentato, utilizzato in molti sistemi open source (come ad esempio in OpenBSD).



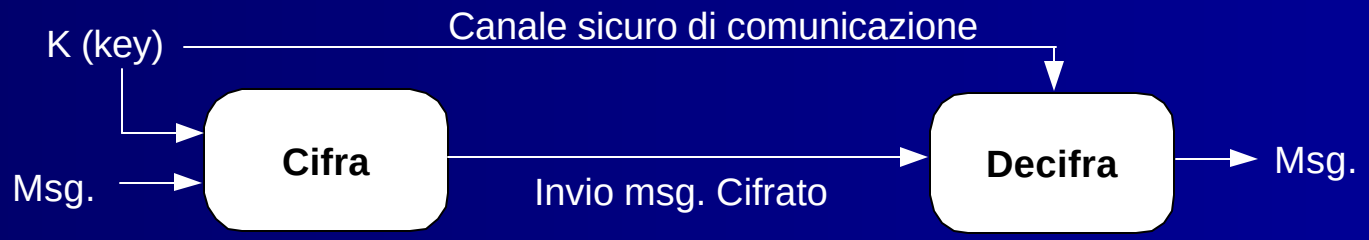
Rijndael

- Sviluppato Joan Daemen e Vincent Rijmen.
- Questo algoritmo ha vinto la selezione per l'Advanced Encryption Standard (**AES**) il 2 Ottobre 2000. Ufficialmente il Rijndael è diventato lo standard per la cifratura del XXI secolo.
- Il cifrario 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.



Il problema della trasmissione della chiave

- Volendo utilizzare un cifrario simmetrico per proteggere le informazioni tra due interlocutori come posso scambiare la chiave segreta? Devo utilizzare una “canale sicuro” di comunicazione.



Ma tale “canale sicuro” esiste nella realtà?

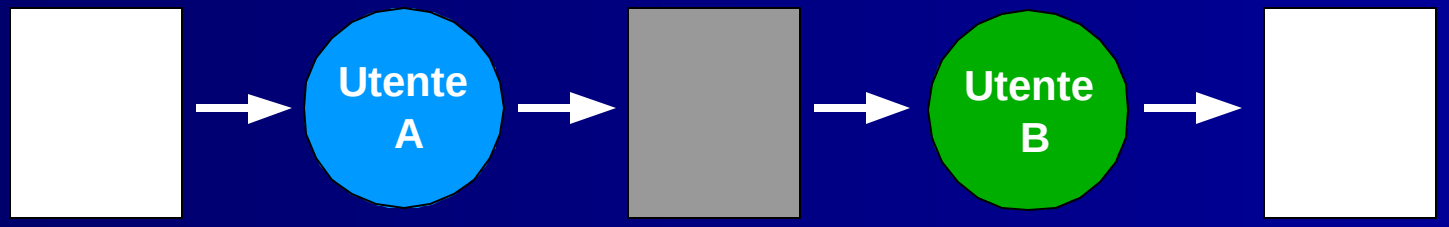
Per una comunicazione sicura tra n utenti si dovranno scambiare in tutto (n-1)*n/2 chiavi, ad esempio con 100 utenti occorreranno 4950 chiavi, il tutto per ogni comunicazione!

La crittografia a chiave pubblica

- Utilizza una coppia di chiavi per le operazioni di cifratura (*encryption*) e decifrazione (*decryption*).
- Una chiave detta pubblica (**public key**) viene utilizzata per le operazioni di encryption.
- L'altra chiave, detta privata (**private key**), viene utilizzata per le operazioni di decryption.
- A differenza dei cifrari simmetrici non è più presente il problema della trasmissione delle chiavi.
- Sono "intrinsecamente sicuri" poiché utilizzano tecniche di tipo matematico basate sulla teoria dei numeri, sulla teoria delle curve ellittiche, etc.

La crittografia a chiave pubblica

- Esempio di encryption (trasmissione sicura):



Msg. In chiaro

Msg. Cifrato con $K_{pu}B$

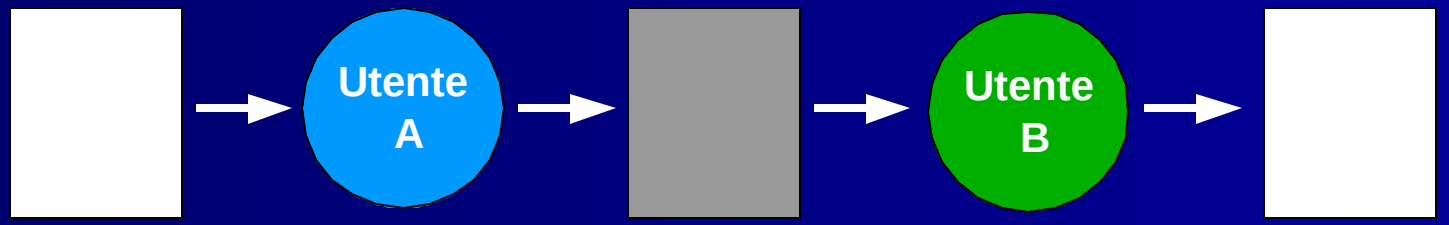
Msg. In chiaro decifrato con $K_{pr}B$

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

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

La crittografia a chiave pubblica

- Esempio di autenticazione:



Msg. Da autenticare ("firmare")

Msg. Autenticato con $K_{pr}A$

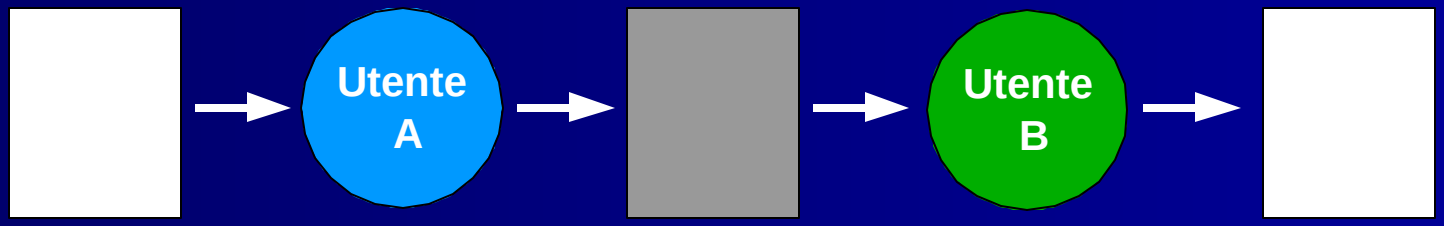
Fase di autenticazione Msg. Autenticato con $K_{pu}A$

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

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

La crittografia a chiave pubblica

- Esempio di encryption ed autenticazione:



Msg. In chiaro

Msg. Autenticato con $K_{pr}A$ e cifrato con $K_{pu}B$

Msg. In chiaro decifrato con $K_{pr}B$ ed autenticato con $K_{pu}A$

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

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

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

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

La nascita dei sistemi PKI (Public Key Infrastructure)

- Dove trovo le chiavi pubbliche dei miei destinatari?
- Creazione di “archivi di chiavi pubbliche”, i public key server.
- Ma chi mi garantisce la corrispondenza delle chiavi pubbliche con i legittimi proprietari?
- Nascita delle **certification authority** (**CA**).
- A questo punto chi garantisce la validità delle certification authority?
- Atto di fede!

Il cifrario RSA (Ronald **Rivest**, Adi **Shamir** e Leonard **Adleman**)

- E' basato su tecniche di teoria dei numeri: prodotto di due numeri primi di dimensioni elevate (ad esempio con 300 cifre decimali).
- Definiamo alcuni concetti di teoria dei numeri per poter analizzare il funzionamento del cifrario RSA:
  - Un numero p>1 si dice **primo** se è divisibile solo per ±1 e ±p.
  - Dati tre interi a,b=0 e n>0, si dice che **a è congruo a b modulo n** se esiste un intero k per cui a = b + kn (o equivalentemente se a mod n = b mod n, dove l'operatore **mod** indica il resto della divisione intera tra a e n, b e n).
  - Per un intero n>1 si definisce la funzione di Eulero **F (n)** come il numero di interi minori di n e relativamente primi con esso.
    Se n è un numero primo si ha che F(n)=n-1.

Il cifrario RSA

- Le chiavi pubbliche e private vengono determinate con il seguente algoritmo:
  - Si scelgono due numeri primi **p** e **q** molto grandi;
  - Calcolo **n= p*q**, e la funzione di Eulero **F (n)**= (**p**-1)*(**q**-1);
  - Scelgo un intero **e** minore di **F (n)** e primo con esso;
  - Calcolo l'intero **d**, inverso di **e** modulo **F (n)** (ossia tale che **e***d** = k***F (n) + 1,** con k numero intero);
  - La chiave pubblica è costituita dalla coppia di valori **<e,n>**, la chiave privata dai valori **<d,n>**.
- Le operazioni di encryption e decryption sono:

$$C=M^e \pmod n \quad , \quad M=C^d \pmod n=(M^e \pmod n)^d \pmod n$$

dove M= blocco di testo in chiaro, C= crittogramma.

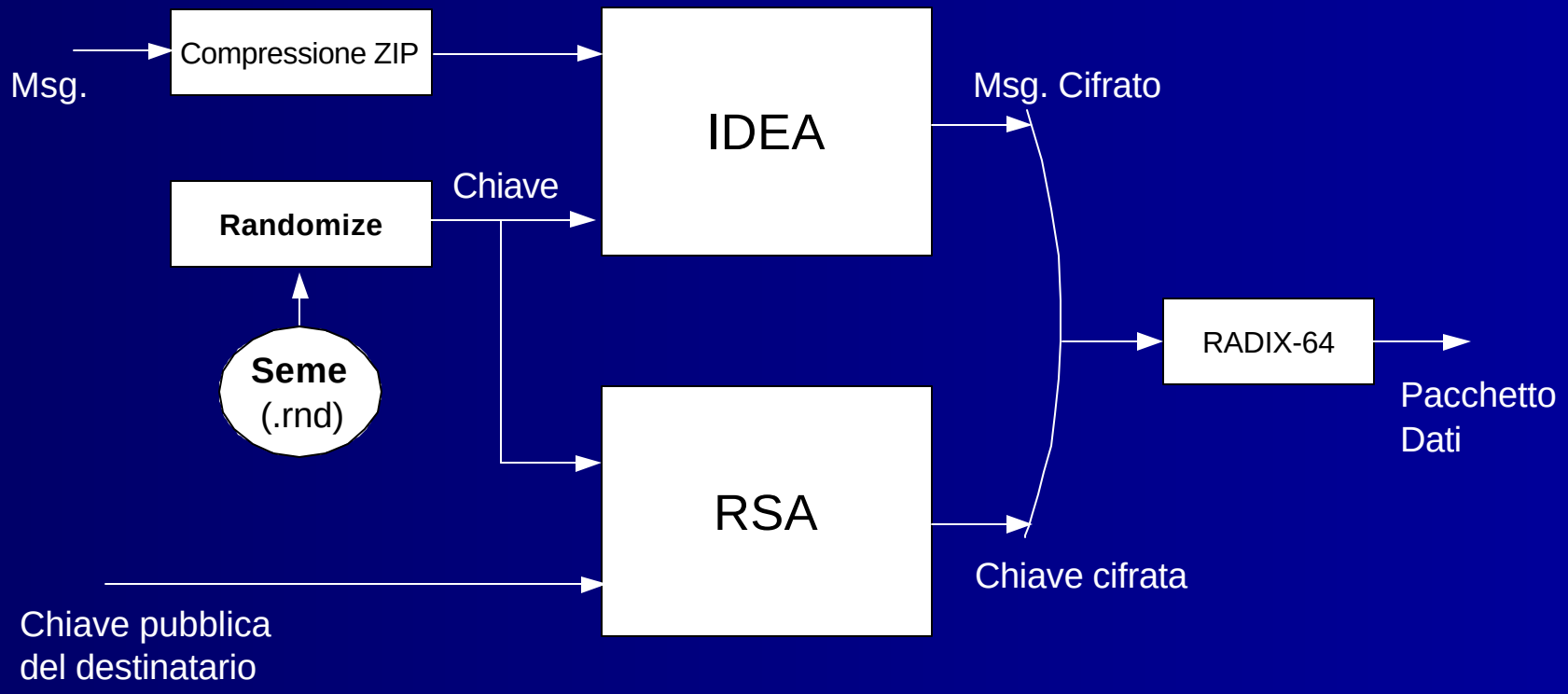
Il cifrario RSA

- La sicurezza del sistema è basata sul fatto che è difficile fattorizzare un prodotto di due numeri primi di dimensioni elevate (allo stato attuale).
- La lunghezza delle chiavi è variabile: 512, 1024, 2048, 4096 bit ed oltre.
- Svantaggio: l'algoritmo RSA non è veloce, 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).
- Il 6 Settembre 2000 l'algoritmo RSA è diventato di dominio pubblico, prima era di proprietà dell'omonima azienda RSA Security Inc.

Esempio di un sistema crittografico ibrido: il PGP

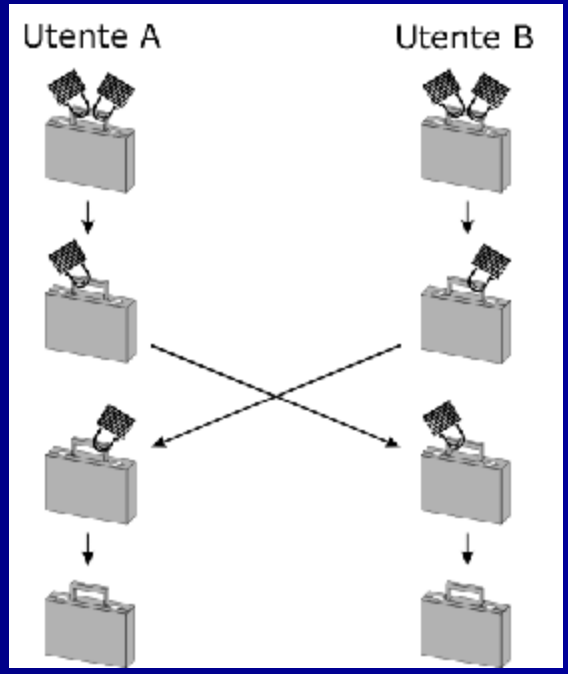
- PGP (Pretty Good Privacy) è un software di pubblico dominio creato da Phil Zimmermann nel 1991.
- E' un software per la privacy personale: protezione delle email, dei files, firma digitale.
- Utilizza gli algoritmi di crittografia a chiave pubblica RSA, Diffie-Hellman, DSA e gli algoritmi simmetrici IDEA, CAST, 3-DES.
- E' basato su di un sistema di crittografia “ibrido” nel senso che utilizza crittografia simmetrica per le operazioni di encryption sui dati generando delle chiavi di sessione pseudo-casuali cifrate con un algoritmo a chiave pubblica.
- Attualmente il progetto PGP è alla versione 8.0 public beta, grazie alla creazione di una nuova società PGP Corporation.

Il funzionameno del PGP



L'algoritmo Diffie-Hellman per lo scambio delle chiavi

- Creato nel 1976 dai ricercatori W.Diffie e M.Hellman è il primo algoritmo a chiave pubblica della storia.
- E' stato creato per eliminare il problema dello scambio delle chiavi di cifratura su di un canale insicuro di comunicazione.
- L'efficacia dell'algoritmo Diffie-Hellman dipende dalla difficoltà di calcolare logaritmi discreti.
- Il sistema lavora su delle strutture albebriche particolari, i campi di Galois con prodotti e potenze di numeri interi.

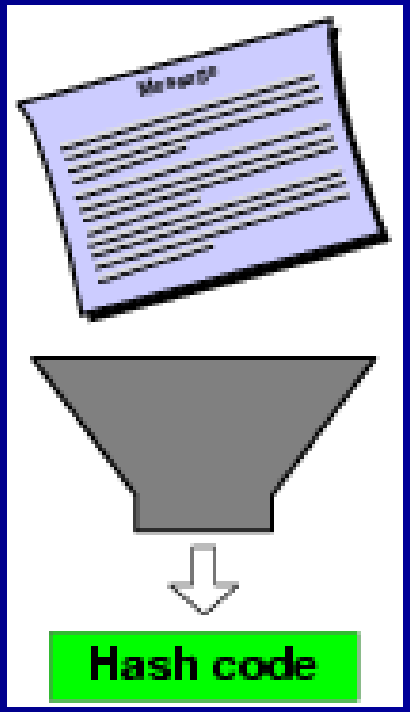


La firma digitale e le funzioni hash sicure

- Nasce come applicazione dei sistemi a chiave pubblica.
- Viene utilizzata per autenticare la paternità di un documento informatico e la sua integrità.
- Si utilizza un cifrario a chiave pubblica e si “cifra” un documento (file) con la propria chiave segreta. Chiunque può verificare la paternità del documento utilizzando la chiave pubblica dell'utente firmatario.
- Problema: per l'autenticazione di un documento di grandi dimensioni con un algoritmo a chiave pubblica occorre molto tempo.
- Soluzione: posso autenticare solo un “riassunto” del documento tramite l'utilizzo di una funzione hash sicura.

Le funzioni hash sicure

- Vengono utilizzate per generare un sorta di “riassunto” di un documento informatico (file).
- Una funzione hash accetta in ingresso un messaggio di lunghezza variabile M e produce in uscita un digest di messaggio H(M) di lunghezza fissa.
- Questo digest (impronta digitale, targa, riassunto) è strettamente legato al messaggio M, ogni messaggio M genera un H(M) univoco.
- Anche considerando due messaggi M ed M' differenti solo per un carattere le loro funzioni hash H(M) e H(M') saranno diverse.



Requisiti di una funzione hash sicura H(x)

- H può essere applicata a un blocco di dati di qualsiasi dimensione;
- H produce in uscita un risultato di lunghezza fissa (ad esempio 160 bit);
- Per qualunque codice h il calcolo di x tale che H(x)=h deve avere una complessità computazionale improponibile;
- Per qualunque blocco di dati x deve essere il calcolo di y?x tale che H(x)=H(y) deve avere una complessità computazionale improponibile.
- Ai fini pratici H(x) deve essere relativamente semplice da calcolare.

Esempio di funzione hash

- Tutte le funzioni hash operano sulla base del seguente principio: i dati in ingresso sono considerati come una sequenza di blocchi di n bit, essi vengono elaborati un blocco alla volta iterativamente per produrre una funzione hash di n bit.
- Una delle più semplici funzioni hash è quella che esegue lo XOR (+) bit a bit di ciascun blocco, ossia:

$$C_i = b_{i1} + b_{i2} + \ldots + b_{im}$$

- Dove C_i rappresenta l'i-esimo bit del codice hash, m il numero di blocchi di n bit, b_{ij} l'i-esimo bit all'interno del j-esimo blocco e l'operatore + l'operazione di XOR.
- La probabilità che un errore nei dati produca lo stesso valore hash è 2^{-n}, con n=128 bit $2^{-128} \sim 2{,}9387 * 10^{-39}$.

La crittoanalisi

- La scienza che si occupa dell'analisi e della validità degli algoritmi crittografici.
- Analizzando il contenuto di un testo cifrato, attraverso tecniche statistico/matematiche si possono ottenere informazioni sul testo in chiaro.
- Per fortuna ciò non è sempre possibile, la maggior parte dei cifrari moderni è ancora al sicuro da tecniche di crittoanalisi.
- La storia ci insegna che non esistono cifrari inviolabili.

Le basi della crittoanalisi

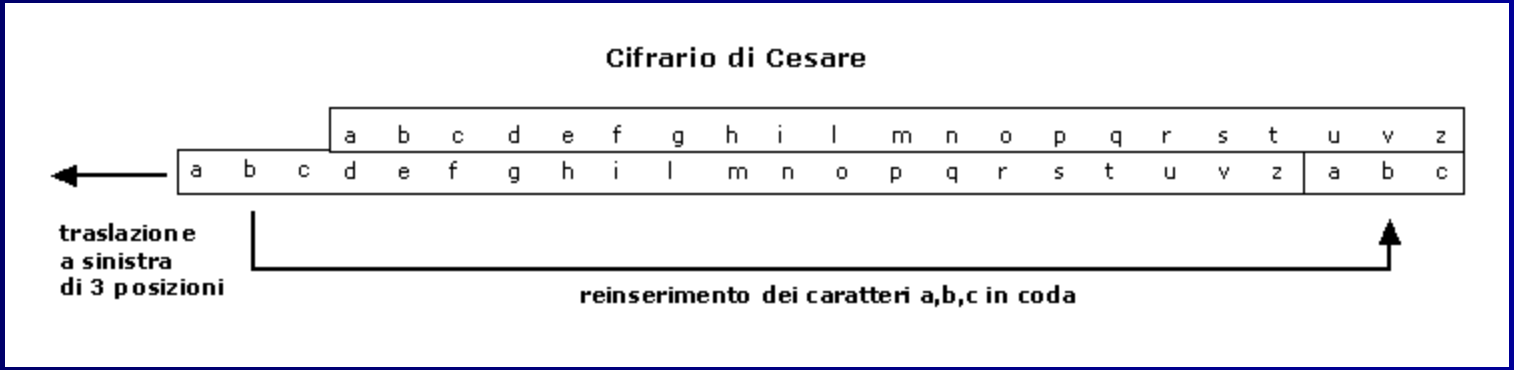
- L'attacco ad un sistema crittografico ha l'obiettivo di forzare il sistema, il metodo scelto e il suo livello di pericolosità dipendono dalle informazioni in possesso del crittoanalista.
- Fondamentalmente esistono queste tipologie di attacchi:
  - **Cipher Text Attack** (il crittoanalista è in possesso solo di alcuni crittogrammi)
  - **Known Plain-text Attack** (il crittoanalista è venuto a conoscenza di una serie di testi in chiaro e di crittogrammi)
  - **Chosen Plain-Text Attack** (il crittoanalista ha scelto una serie di testi in chiaro e di crittogrammi)

La crittoanalisi statistica

- Tramite l'utilizzo di tecniche statistiche sulla frequenze dei caratteri o sottostringhe del testo cifrato si ottengono informazioni utili sul testo in chiaro.

Un primo esempio: il cifrario di Cesare

- Consideriamo l'alfabeto italiano, costruiamo un cifrario che sostituisce ad ogni lettera di questo alfabeto la lettera che si trova 3 posizioni in avanti.



- Ad esempio il testo in chiaro “prova di trasmissione” viene cifrato nel crittogramma “surbd gn zudvpnvvnrqh”.

Crittoanalisi statistica del cifrario di Cesare

- Il cifrario di Cesare, come la maggior parte dei cifrari storici basati tu trasposizioni e traslazioni, può essere facilmente violato utilizzando tecniche statistiche (crittoanalisi statistica).
- Si analizzano le frequenze relative dei caratteri nel testo cifrato e le si confrontano con quelle di una lingua conosciuta, ad esempio l'italiano.
- Le frequenze relative al testo cifrato “surbd gn zudvpnvvnrqh” risultano *s* (1/19), *u* (2/19), *r* (2/19), *b* (1/19), *d* (2/19), *g* (2/19), *n* (3/19), *z* (1/19), *v* (3/19), *p* (1/19), *h* (1/19).
- Si confrontano tali frequenze con quelle della lingua italiana: *a* (0,114), *e* (0,111), *i* (0,104), *o* (0,099), *t* (0,068), *r* (0,065),...
- Con queste informazioni ottengo una prima approssimazione del testo in chiaro “s**ro**b**a** g**i** z**ra**vp**i**vv**io**qh”, procedo per tentativi ripetendo il procedimento.

Altre tecniche di crittoanalisi

- **Brute-force**, ossia tramite il calcolo di tutte le possibili combinazioni di chiavi del cifrario. Con l'aumento della potenza di calcolo degli elaboratori questa tecnica banale stà diventando sempre più efficace. Basti pensare al Cracking del DES a 56 bit con un computer multiprocessore costruito dall'EFF in grado di violare l'algoritmo in meno di 3 giorni di calcolo.
- **Crittoanalisi differenziale**, tramite l'analisi delle "distanze" numeriche dei caratteri presenti nel testo cifrato e l'ausilio di sofisticate tecniche matematiche unite ad algoritmi sempre più veloci.
- **Man-in-the-middle**, sfruttando il sistema delle infrastrutture PKI un eventuale intruso può posizionarsi tra un mittente ed un destinatario e scambiare le loro chiavi pubbliche e private con altre opportunamente modificate.

DES Cracking

- Il 17 Luglio 1998 la Cryptography Research, l'Advanced Wireless Technologies, e l'EFF annunciano a tutto il mondo di aver costruito un computer in grado di violare l'algoritmo DES in meno di 56 ore!
- Il DES a 56 bit genera un numero di chiavi possibili pari a 72'057'594'037'927'936.
- Cracking DES: 6 cabinet riciclati SUN-2, ognuno contenente 27 schede circuito contenenti 64 microchip, ogni microchip contiene 24 unità di ricerca delle chiavi, il tutto pilotato da un PC.
- Performance: 92 bilioni di chiavi DES al secondo!
- Costo del progetto: < 250'000 $

Quanto devono essere lunghe le chiavi?

- **Bruce Schneier** il 15 Aprile 2002 pubblica un articolo nella sua newsletter intitolato “Is 1024 Bits Enough?” dove suggerisce la lunghezza delle chiavi, per la sicurezza dei prossimi anni, con l'utilizzo di algoritmi a chiave pubblica.

Lunghezza chiavi (in bits)			
Anno	**Privacy personale**	**Aziende**	**Governi**
1995	768	1280	1536
2000	1024	1280	1536
2005	1280	1536	2048
2010	1280	1536	2048
2015	1536	2048	2048

Il security flaw nel formato OpenPGP (24 Marzo 2001)

- Oltre alla crittoanalisi bisogna fare i conti con i bug dei software crittografici.
- E' un attacco che sfrutta un bug sul formato aperto internazionale **OPENPGP**.
- Scoperto da due crittologi della Repubblica Ceca, Vlastimil Klima e Tomas Ros nel 2001.
- L'attacco scoperto sul formato OpenPGP è basato sul fatto che alcune informazioni "delicate" sulla chiave pubblica e privata di un utente non sono protette adeguatamente nel file di configurazione del programma crittografico che si sta utilizzando.
- Modificando queste informazioni con dei dati prestabiliti si possono ottenere dei valori numerici utilizzabili per il calcolo della chiave privata dell'utente.

Vulnerabilità sugli ID Multipli del PGP (5 Settembre 2001)

- L'attacco scoperto da Sieuwert van Otterloo consente di "falsificare" le firme digitali generate dal programma attraverso l'utilizzo di User Id fasulli.
- Questo tipo di attacco funziona su tutte le versioni del PGP a partire dalla 5.x fino alle più recenti 7.x.
- L'attacco può compromettere il sistema di “Web of Trust” ossia di fiducia nell'utilizzo delle chiavi pubbliche tra utenti web.
- E' basato principalmente su di un errore “logico” di impostazione delle chiavi trusted nel software PGP quando più utenti hanno più user ID, tipicamente più caselle di posta elettronica.
- Per maggiori info: http://www.bluering.nl/pgp/

Bibliografia italiana essenziale

- “Sicurezza delle reti - Applicazioni e standard” di William Stallings, Addison-Wesley Editore.
- “Crittografia - Principi, Algoritmi, Applicazioni” di P. Ferragina e F. Luccio, Bollati Boringhieri Editore.
- “Crittografia” di Andrea Sgarro, Franco Muzzio Editore.
- “Segreti, Spie e Codici Cifrati” di C.Giustozzi, A.Monti, E.Zimuel, Apogeo Editore.
- “Codici & Segreti” di Simon Singh, Rizzoli Editore.
- “Sicurezza Digitale” di Bruce Schneier, Tecniche Nuove Editore.
- “Sicurezza dei sistemi informatici” di M.Fugini, F.Maio, P.Plebani, Apogeo Editore.
- “Internet Security – seconda edizione” di Maurizio Cinotti, Hoepli Editore.

Webografia essenziale

- http://www.crypto.com
- http://www.pgpi.org
- http://www.pgp.com
- http://www.counterpane.com/crypto-gram.html
- http://www.rsasecurity.com/
- http://theory.lcs.mit.edu/~rivest/
- http://www.swcp.com/~iacr/
- http://www.ieee-security.org/cipher.html
- http://www.kluweronline.com/issn/0925-1022
- http://world.std.com/~franl/crypto.html
- http://www.enricozimuel.net
- http://www.dia.unisa.it/~ads/