Crittografia è sinonimo di sicurezza?

di Enrico Zimuel (enrico@enricozimuel.net)

5 Luglio 2003 – RoboCup 2003 – Padova

Note sul copyright (copyfree):

Questa presentazione può essere utilizzata liberamente
a patto di citarne la fonte e non stravolgerne il contenuto.

Questa presentazione è stata creata con OpenOffice 1.0
www.openoffice.org
ed è disponibile su internet all'indirizzo
http://www.enricozimuel.net/documenti/conference/robocup2003.sxi

Sommario

- La crittografia é sinonimo di sicurezza?
- I principi della sicurezza e gli ambiti d'utilizzo della crittografia: identificazione, autenticazione, riservatezza, integrità, anonimato
- I limiti dei sistemi proprietari e la crittografia open source
- Lunghezza della chiave e sicurezza
- I sistemi di autenticazione a due fattori: smart card, token, sistemi biometrici
- I protocolli crittografici in Internet
- Esempi di software crittografici open source: GnuPG, OpenSSL, OpenSSH, OpenCA, FreeSWAN

La crittografia è sinonimo di sicurezza?

- No! La crittografia è uno strumento necessario per la sicurezza ma non sufficiente. Sicurezza e crittografia sono due cose diverse.
- La maggior parte dei problemi legati al mondo della sicurezza informatica deriva da problemi di natura umana/organizzativa.
- Uno stupendo algoritmo di crittografia non può nulla contro una pessima programmazione, un sistema operativo scadente o una password inappropriata.
- "La sicurezza di un sistema informatico non è un prodotto ma un processo" Bruce Schneier

I principi della sicurezza e gli ambiti d'utilizzo della crittografia

- Si parla di sicurezza informatica quando è necessario, in generale, proteggere un sistema informatico da possibili manomissioni e/o attacchi interni/esterni.
- La crittografia utilizzata nei sistemi di sicurezza informatica è in grado di garantire, se applicata correttamente, le seguenti operazioni: identificazione, autenticazione, riservatezza, integrità, anonimato.
- I sistemi di sicurezza informatica si basano principalmente su alcune implementazioni più o meno standard di algoritmi crittografici.
- Nella fase dell'implementazione si possono commettere errori!

I limiti dei sistemi proprietari e la crittografia open source

- Come evitare di commettere errori nella fase di implementazione? Basta utilizzare degli algoritmi standardizzati, riconosciuti sicuri dalla comunità internazionale.
- "*Se un sistema è veramente sicuro, lo è anche quando i dettagli divengono pubblici*" Bruce Schneier
- Questa apparente contraddizione può essere spiegata solo grazie all'ausilio della matematica come base teorica della crittografia.
- In che senso la matematica rende la crittografia sicura? Esempio: Sia *n* un numero con 617 cifre decimali prodotto di due numeri primi *p* e *q*, ossia *n*= *p***q*, calcolare *p* e *q*. Allo stato attuale non esiste nessun algoritmo in grado di calcolare *p* e *q* in tempi "accettabili" (in gergo, con complessità al più polinomiale).
- Rsa Inc. mette in palio $200'000 per chi riesce a calcolare *p* e *q*

I limiti dei sistemi proprietari e la crittografia open source

- Matematica + crittografia = 100% sicurezza?
- Purtroppo No! Allo stato attuale possiamo solo dire che non esiste nessun algoritmo in grado di violare un algoritmo robusto di crittografia... ma non ne abbiamo una dimostrazione in senso matematico.
- E' una questione di fiducia. Sempre meglio fidarsi della matematica che di un'azienda di sicurezza con un sistema proprietario chiuso...
- 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).

I limiti dei sistemi proprietari e la crittografia open source

- Conoscenza algoritmo = libera distribuzione codici sorgenti = standard aperti = uno dei principi fondamentali dell'open source!
- I sistemi crittografici proprietari chiusi **non possono essere per definizione sicuri**.
- Come fa un sistema chiuso ad essere considerato sicuro se non si conosco, nel dettaglio, i principi di funzionamento?
- Siete disposti ad affidare il vostro conto in banca ad un sistema informatico sconosciuto? Chi di voi è a conoscenza dei dettagli tecnici di funzionamento del vostro sistema di home-banking?
- Dunque vi fidate della vostra banca e dei loro consulenti tecnici? :-)

Lunghezza della chiave e sicurezza

- Ora che avete scelto il vostro sistema di crittografia “aperto”, dovete utilizzarlo... quasi tutti i sistemi crittografici si basano sull'utilizzo di una password... quale password scegliere?
- La data di nascita di mio figlio? Il nome del mio cane? Il PIN del mio Bancomat che è scritto su di un foglietto conservato nel mio portafoglio? ...
- Il concetto stesso di password si basa su di un ossimoro: la password deve infatti essere una stringa di caratteri casuali, facile da ricordare.
- Ma se deve essere facile da ricordare come può essere casuale?

Lunghezza della chiave e sicurezza

- Chiavi più lunghe = chiavi più sicure? In teoria si, in pratica no.
- Le chiavi vengono generate quasi sempre attraverso delle password di autenticazione scelte dall'utente (ad esempio la *pass phrase* del Pgp/GnuPg).
- Quasi sempre le password di autenticazione sono frasi di senso compiuto o frasi casuali di piccole dimensioni, ad esempio di 10 caratteri, è difficile ricordare a memoria più di 10 caratteri casuali.
- Questo limite umano fa diminuire notevolmente lo spazio delle chiavi ossia l'insieme di tutte le possibili permutazioni di una chiave di n bit.
- Tramite dei semplici programmi di cracking è possibile effettuare con successo un attacco di forza bruta (brute-forcing).

Lunghezza della chiave e sicurezza

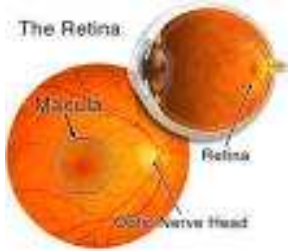
- Ad esempio se la password di un utente è una parola di senso compiuto tramite un attacco di forza bruta basato su un dizionario è possibile violare un sistema in pochi secondi.
- Alcuni test effettuati nel 2000 con un famoso programma di cracking, L0phtcrack, hanno dimostrato che il 90% delle password possono essere determinate in meno di un giorno e circa il 20% nell'arco di pochi minuti.
- Se in un sistema che contiene 1.000 account 999 utilizzano password incredibilmente complicate, L0phtcrack riesce a entrare nel sistema scoprendo l'unica password debole.
- Le password di autenticazione sono dunque insicure proprio perchè subentra nel sistema di sicurezza il fattore umano, le password devono essere ricordate dagli utenti.

I sistemi a doppia autenticazione

- Per evitare di affidare la sicurezza ad un semplice password scelta "casualmente" da un utente si possono utilizzare delle smart card e/o dei token hardware (smart card, usb keys, etc).
- Una smart card/token è un dispositivo di sicurezza in grado di memorizzare una chiave, ed in alcuni casi un algoritmo, in maniera sicura. Utilizzando tale dispositivo possiamo accedere in maniera sicura ad un sistema informatico.
- Oltre al possesso del dispositivo per poterlo utilizzare è necessario ricordare una password, un PIN, di accesso al dispositivo, ecco perchè si parla di doppia autenticazione.

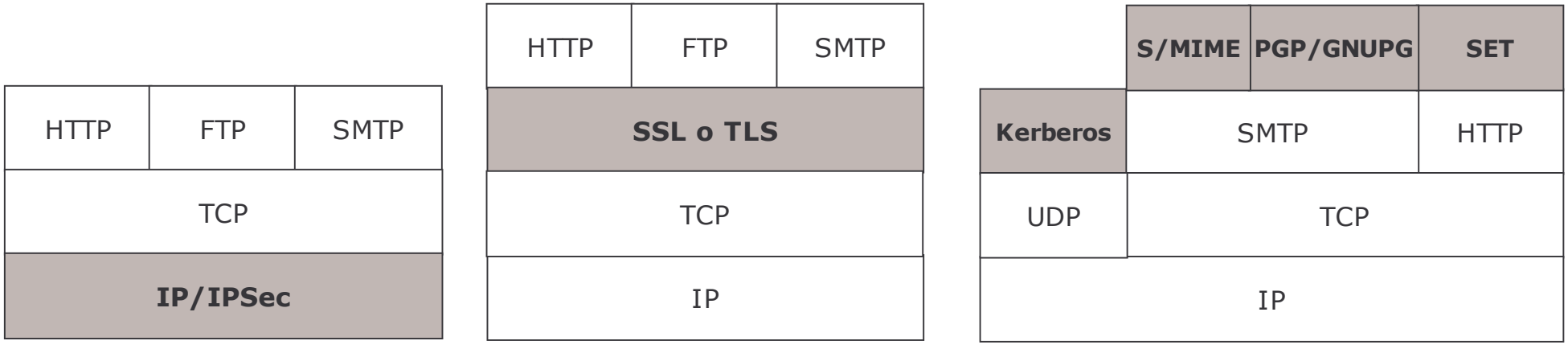
I sistemi di autenticazione biometrica

- E' possibile ottenere un sistema di autenticazione a due fattori senza dover possedere una chiave su di un dispositivo hardware? La chiave può essere costituita da noi stessi!
- Tutti gli esseri umani presentano delle differenze fisiche caratteristiche ed univoche. Ad esempio le impronte digitali, la retina dell'occhio, la voce,etc.
- I sistemi biometrici sfruttano queste diversità fisiche per garantire l'univocità dell'accesso ad un sistema informatico.
- Anche in questo caso è consigliabile implementare sempre il sistema con una doppia autenticazione: una caratteristica fisica + una password/PIN per l'accesso.



I protocolli crittografici in internet

- Per fornire sicurezza del traffico web esistono diversi approcci simili dal punto di vista delle funzionalità ma differenti per quanto concerne l'ambito di applicabilità ed il loro posizionamento all'interno della pila del protocollo TCP/IP.



Matematica + crittografia + tecnologia = sicurezza?

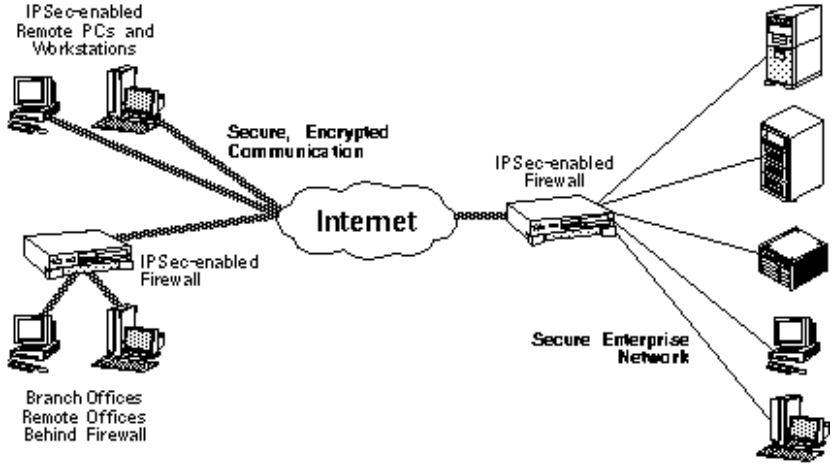
- Attraverso l'utilizzo di questi dispositivi hardware con sistemi a doppia autenticazione siamo dunque sicuri?
- No! Anche i dispositivi precedenti possono essere manomessi.
- Esistono tecniche sofisticate come il DPA (Differential Power Analysis) che consentono di scoprire le informazioni sensibili memorizzate all'interno dei dispositivi hardware variando l'alimentazione del dispositivo.
- Esistono tecniche banali (ma geniali) che consentono di dissimulare i dispositivi biometrici, ad esempio nel 2002 Ysutomu Matsumoto, un crittografo giapponese, utilizzando una particolare gelatina applicata su di un dito, è riuscito a violare un sistema di autenticazione biometrica copiando l'impronta lasciata su di un vetro da un utente autorizzato del sistema.
- Morale della favola la sicurezza informatica al 100% non esiste. Ciò non vuol dire che non si può essere sicuri, personalmente mi fido di più di un sistema crittografico conosciuto e ben progettato che di un cassiere allo sportello di una banca...

Sicurezza IP

- Il protocollo di comunicazione attualmente utilizzato su Internet Ipv4 non prevede la cifratura dei messaggi dati (nell'IPv6 si), chiunque può “sniffare” ossia intercettare dalla rete un pacchetto dati e ricostruire il messaggio originario.
- La sicurezza IP (IPSec, *IP Security*) nasce da una precisa esigenza: proteggere le comunicazioni IP da attacchi di tipo IP spoofing (falsificazione degli indirizzi IP) e di tipo IP sniffing (intercettazione dei pacchetti dati).
- La protezione avviene tramite encryption dei pacchetti IP.
- IPSec fornisce la possibilità di rendere sicure le comunicazioni su LAN, su WAN private e pubbliche e su Internet.
- IPSec fornisce un insieme di servizi di sicurezza a livello IP: controllo dell'accesso, integrità in assenza di connessione, autenticazione della sorgente dati, rifiuto di pacchetti originati da un attacco di replay, riservatezza (cifratura), parziale riservatezza del flusso di traffico.

Sicurezza IP

- Alcuni esempi d'utilizzo di IPSec:
  - Connettività sicura di filiali su Internet;
  - Accessi remoti sicuri su Internet;
  - Possibilità di stabilire connettività extranet e intranet con i partner;
  - Miglioramento della sicurezza del commercio elettronico;
- La caratteristica più importante di IPSec è che può cifrare e/o autenticare tutto il traffico a livello IP, adattandosi quindi a una notevole gamma di applicazioni.



I protocolli SSL e TLS

- SSL (Secure Socket Layer) è stato creato dalla Netscape nel 1994, la versione 3.0 è del 1995.
- Successivamente il progetto è stato sottoposto al processo di standardizzazione Internet, all'interno di IETF () si costituì il gruppo di lavoro TLS (Transport Layer Security).
- La prima versione del TLS può essere considerata come un SSL 3.1, in pratica lo standard TLS è l'evoluzione del protocollo SSL.
- SSL è progettato per fare uso del protocollo TCP al fine di fornire un servizio di sicurezza end-to-end affidabile.
- Viene utilizzato per proteggere le transazioni via web di dati sensibili (https): acquisti legati all'e-commerce, numeri di carte di credito, informazioni aziendali, etc.

Difendere la privacy personale: GNUPG

- Il progetto tedesco GnuPG (GNU Privacy Guard) nasce nel 1997 per opera di Werner Koch, sviluppatore indipendente interessato alla crittografia Open Source.
- L'obiettivo del progetto è la realizzazione di un engine crittografico, alternativo al Pgp, totalmente open source basato su algoritmi crittografici standard e non proprietari.
- Disponibile in più versioni: Gnu/Linux, Ms Windows, FreeBSD, OpenBSD, AIX, Sun Os, BSDI, IRIX, etc.
- Disponbili vari front-end per sistemi GUI: Gnome, KDE, Ms Windows, etc.
- Supporto esteso di algoritmi crittografici ElGamal, DSA, RSA, AES, 3DES, Blowfish, Twofish, CAST5, MD5, SHA-1, RIPE-MD-160 e TIGER
- La versione attuale è la 1.2.1

Transazioni sicure sul web: OpenSSL

- Progetto open source per l'implementazione dei protocolli di sicurezza SSL v2/v3 (Secure Sockets Layer) e TLS v1 (Transport Layer Security).
- Basato sulle librerie SSLeay sviluppate da Eric A. Young e Tim J. Hudson.
- La licenza d'utilizzo non è GPL ma completamente free con disponibilità dei codici sorgenti, può essere utilizzato anche per scopi commerciali (come la licenza Apache).
- La versione attuale è la 0.9.7b (12 Maggio 2003).
- Sviluppo parallelo di un engine crittografico per il supporto di dispositivi hardware 0.9.6-engine.
- Supporto sistemi Gnu/Linux, Ms Windows, VMS.

Comunicazioni sicure su Internet: OpenSSH

- Progetto open source per l'implementazione del protocollo SSH (versioni 1.3,1.5 e 2.0).
- La prima versione 1.2.12 free dell'ssh è stata implementata da Tatu Ylönen, attualmente siamo arrivati alla 3.6.1.
- Nato all'interno del progetto OpenBSD, è stato incluso nel sistema operativo OpenBSD a partire dalla versione 2.6; attualmente il progetto è internazionale.
- Caratteristiche principali: licenza free, strong encryption (3DES, Blowfish), X11 Forwarding, Port Forwarding, Strong Authentication (pki, one-time password, kerberos), Agent Forwarding (Single Sign-one), Compressione dei dati
- La suite OpenSSH comprende anche i seguenti tool: lato client ssh telnet, scp, sftp e lato server sshd, ssh-add, ssh-agent, ssh-keygen, sftp-server.

Certification authority: OpenCA

- Il progetto OpenCA è un servizio di Certification Authority completamente open source.
- OpenCA is basato su diversi altri progetti open source: OpenLDAP, OpenSSL, Apache Project, Apache mod_ssl.
- Il progetto è suddiviso in due gruppi: lo studio degli schemi di sicurezza dei servizi PKI e lo sviluppo delle relative soluzioni software.
- Il progetto è stato fondato da un italiano, Massimiliano Pala, l'attuale LABS Founder & LABS Manager.
- Il progetto è in continua crescita e sono previste anche versioni per il supporto di smart card.
- La versione stabile attuale è la 0.9.1 (Febbraio 2003).

Free/SWAN realizzare sistemi VPN (Virtual Private Network)

- Progetto open source per l'implementazione del protocollo IpSec e IKE per sistemi Linux.
- Può essere utilizzato per implementare reti VPN e per la realizzazione di reti WAN sicure su internet.
- E' utilizzato come modulo WAN su molti sistemi firewall: LinuxMagic, Sentinet, Merilus, LASAT Safepipe, Firecard, Kyzo, PFN ed in molti altri progetti open source di networking come Linux Router Project, Astaro security Linux, Linuxwall, SmoothWall, DevilLinux, Wolverine.
- Gira perfettamente anche su vecchi computer (dal Pentium in su).
- La release attuale è la 2.0 (28 Aprile 2003).

Bibliografia italiana essenziale

- "Sicurezza digitale" di Bruce Schneier, Tecniche Nuove Editore.
- "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.
- "Crittologia" di L. Berardi, A.Beutelspacher, FrancoAngeli Editore.
- "Sicurezza dei sistemi informatici" di M.Fugini, F.Maio, P.Plebani, Apogeo Editore.

Alcuni siti internet sull'argomento...

- www.cryptography.com/resources/researchlinks.html
- www.counterpane.com/crypto-gram-0205.html
- www.counterpane.com/sandl.html
- www.crypto.com
- crypto.stanford.edu
- theory.lcs.mit.edu/~oded/frag.html
- www.iacr.org
- www.smartcardalliance.org
- www.rsasecurity.com/rsalabs/challenges/factoring/numbers.html
- www.dice.ucl.ac.be/crypto