Sicurezza

Leonardo Ganzaroli

Indice

Introduzione

Questi appunti del corso *Sicurezza* sono stati creati durante la laurea Triennale di informatica all'università "La Sapienza".

1 Elementi introduttivi

La sicurezza informatica si occupa delle risorse informatiche soggette a diversi tipi di minacce e le misure necessarie a garantirne la protezione.

I 3 obiettivi chiave sono:

1. **Confidenzialità**
   - Dei dati
   - Privacy
2. **Integrità**
   - Dei dati
   - Dei sistemi
3. **Disponibilità**

Altri 2 fattori importanti sono:

1. **Autenticità**
2. **Responsabilità**

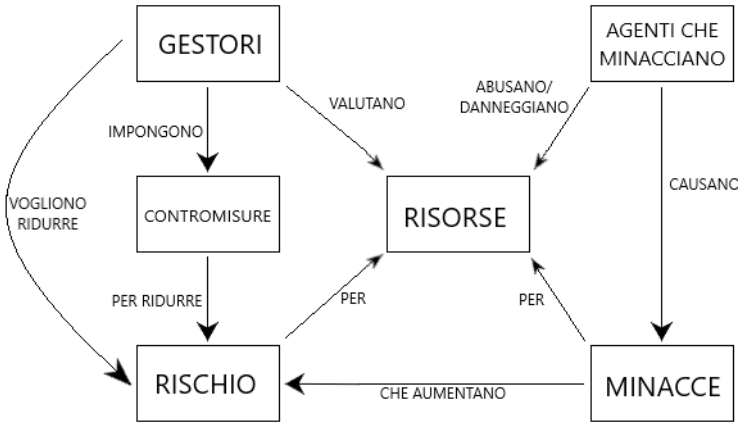


Figura 1: Modello generale

Definizione (Agenti che minacciano) Avversario = individuo/gruppo/organizzazione che vuole condurre o conduce attività dannose.

Definizione Attacco = qualsiasi tipo di attività malevola che tenta di raccogliere/negare/distruggere/... le risorse o i dati di un sistema informativo.

Definizione Contromisura = tecnica o dispositivo che compromette l'efficacia di un'attivita indesiderata.

Definizione Rischio = misura della portata di una minaccia su un'entità.

Definizione Politica di sicurezza = insieme di criteri per la fornitura dei servizi di sicurezza.

Definizione Minaccia = circostanza o evento con il potenziale di avere un impatto negativo.

Definizione Vulnerabilità = Debolezza di un sistema informativo, delle sue procedure, dei suoi controlli interni o della sua implementazione che potrebbe essere sfruttata da una minaccia.

Definizione Attacco = minaccia che viene eseguita ed eventualmente porta ad una violazione.

Le principali conseguenze degli attacchi (ed i rispettivi attacchi) sono:

- Divulgazione non autorizzata
  - Esposizione
  - Intercettazione
  - Inferenza
  - Intrusione
- Inganno
  - Mascheramento
  - Falsificazione
  - Ripudio
- Usurpazione
  - Appropriazione indebita
  - Uso improprio
- Interruzione
  - Interdizione
  - Corruzione
  - Ostruzione

Gli attacchi si possono dividere in:

- **Attivi** se cercando di influenzare il funzionamento del sistema
- **Passivi** se cercano di apprendere informazioni senza toccare le risorse

2 Autenticazione degli utenti

2.1 Modello

Definizione L'autenticazione digitale degli utenti è un processo atto a stabilire la fiducia nelle identità fornite dagli utenti ad un sistema informativo.

Il processo si divide in:

1. **Identificazione**

   L'utente fornisce una presunta identità.

2. **Autenticazione**

   Si stabilisce la validità dell'identità presentata.

Un modello per rappresentare il procedimento è l'SP800-63-3:

Requisiti base
Identificazione
Autenticazione
Requisiti derivati
Uso autenticazione a più fattori
Usare meccanismi resistenti
Evitare riutilizzo identificatori per tot. tempo
Disabilitare identificatori in caso di inattività
Imporre complessità minima password
Proibire riutilizzo password per tot. generazioni
Uso password temporanea per cambio password
Memorizzare/trasmettere solo password cifrate
Oscurare il risultato delle informazioni di autenticazione

Tabella 1: Requisiti del modello

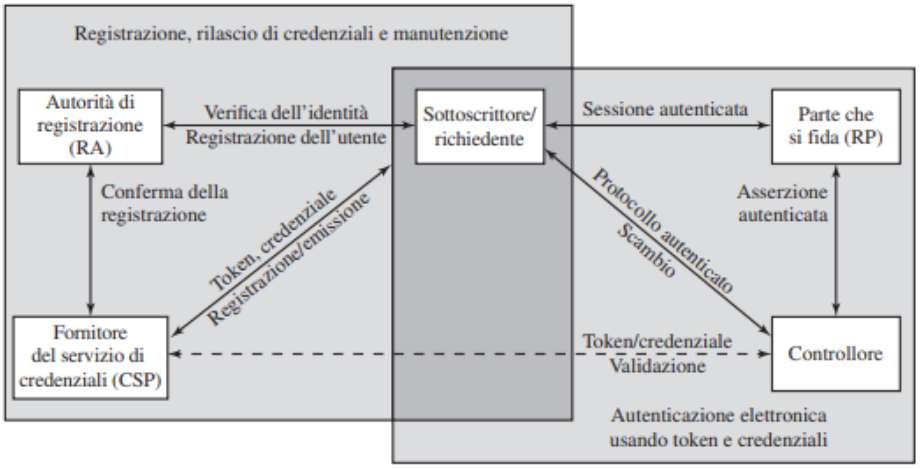


Figura 2: Schema del modello

Procedimento di registrazione:

1. Il richiedente si rivolge a RA per diventare sottoscrittore in CSP
2. Scambio di informazioni tra CSP e richiedente
3. CSP rilascia una credenziale elettronica al richiedente (diventa sottoscrittore)
4. Il sottoscrittore può adesso autenticarsi

Definizione Il verificatore è un insieme di sistemi che effettuano autenticazione e autorizzazione.

Procedimento di autenticazione (previa registrazione):

1. Il richiedente fa richiesta al verificatore
2. Verifica tramite protocollo autenticativo
3. Se OK il verificatore invia un'asserzione sull'identità a RP
4. RP usa le informazioni per decidere accesso e autorizzazioni

2.2 Mezzi di autenticazione

I mezzi principali per autenticare un utente sono 4:

1. Qualcosa che conosce
2. Qualcosa che possiede
3. Qualcosa che è
4. Qualcosa che ha

Definizione L'autenticazione multifattore è l'uso di almeno 2 mezzi.

2.2.1 Password

Le password restano il metodo più diffuso, l'autenticazione avviene tramite la coppia (ID, password) presentata dall'utente al sistema. L'ID inoltre:

- Determina se l'utente può accedere al sistema
- Determina i privilegi dell'utente
- Si usa nel controllo degli accessi discrezionale

Solitamente le password non vengono salvate in chiaro, si salva l'hash (la funzione usata è volutamente lenta) (della password + un valore detto *salt*) all'interno di un file che conterrà la riga (ID, salt, hash). Quando qualcuno deve autenticarsi si recupera nel file la riga corrispondente all'ID, si calcola l'hash con la password fornita ed il *salt* salvato e se il risultato corrisponde a quello presente avviene l'autenticazione.

Un tipo di attacco comune è il *cracking* in cui si cerca di indovinare la password di un utente, tradizionalmente avviene in 2 modi:

1. Si usa un dizionario contenente le password più comuni
2. Si usa la *Rainbow Table* che contiene gli hash precalcolati per ogni possibile *salt*

Per cercare di rafforzare le password ma allo stesso tempo cercare di mantenerle memorizzabili ci sono diverse strategie:

- Educare gli utenti
- Imporre delle regole sulla complessità
- Far generare le password al computer

- Mantenere un dizionario di password scadenti
- Effettuare un cracking "interno" per scovare le password deboli
- Usare il filtro di Bloom

  Dato un dizionario delle password deboli. Un filtro di Bloom di ordine k consiste in k funzioni di hash indipendenti dove:

  $$H_i(x_j) = y \text{ con } 1 \leq i \leq k,\ 1 \leq j \leq D,\ 0 \leq y \leq N-1$$

  In cui:

  - x_j è la j-esima parola nel dizionario
  - D è il numero di parole nel dizionario
  - N è un certo valore

  Funzionamento:

  1. Si definisce un'array di N bit impostati a 0
  2. Per ogni password nel dizionario si calcolano i k valori e usandoli come indici si settano a 1 quei bit nell'array

  Se viene presentata una nuova password e tutti i suoi valori di hash conducono a bit dell'array con valore 1 viene rifiutata. C'è comunque una possibilità di falsi positivi che si approssima con:

  $$\left(1 - e^{-(kD)/N}\right)^k$$

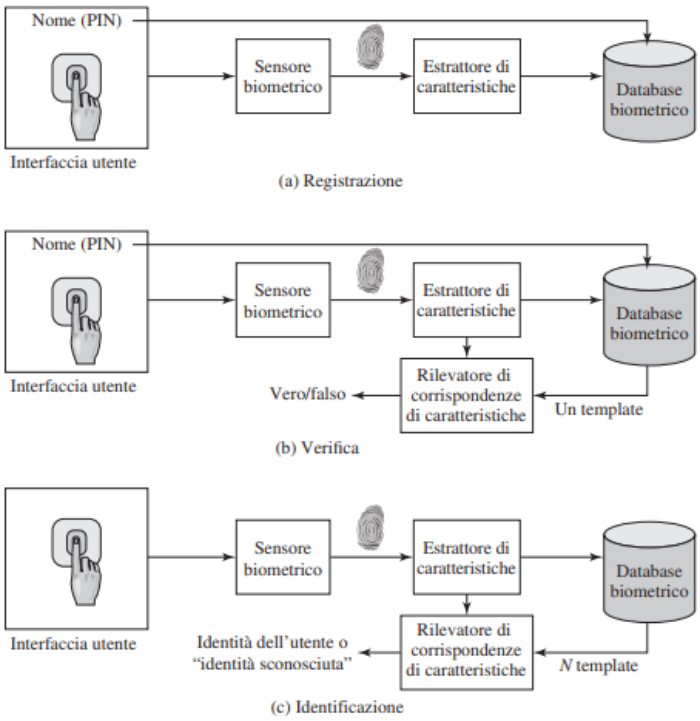
  Questo metodo permette di avere un dizionario enorme senza necessità di mantenerlo in memoria, inoltre velocizza il processo di controllo delle nuove password.

2.2.2 Biometria

I sistemi biometrici sono basati sulle caratteristiche fisiche uniche degli utenti, alcune sono:

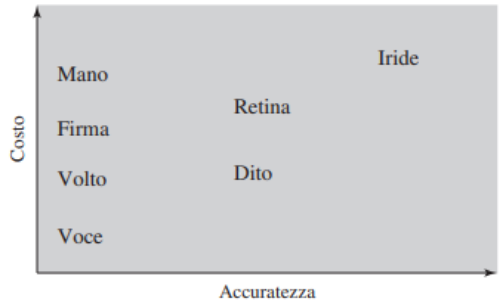
- Impronte digitali
- Geometria della mano
- Schema della retina
- Firma
- Voce

Funzionamento:



Come si può vedere può funzionare sia in modo simile alla password (Verifica) sia da solo (Identificazione).

Essendo le caratteristiche fisiche molto complesse non ci si aspetta che ci sia una corrispondeza perfetta con la loro rappresentazione digitale, quindi si usano degli algoritmi che forniscono un punteggio di somiglianza tra il modello presentato e quello salvato. Questo ovviamente porta a falsi positivi/negativi.



3 Controllo degli accessi

Definizione Il controllo degli accessi è quel processo con cui si concede/nega una richiesta riguardante l'ottenimento/uso di informazioni e relativi servizi o si permette di far entrare un individuo in una struttura.

Definizione L'autorizzazione è la concessione di un diritto/permesso ad un'entità di accedere ad una risorsa.

Definizione Il controllo è la revisione/verifica delle attività e dei registri di sistema.

Definizione CUI = informazioni non classificate controllate.

Un modello per rappresentare il procedimento è l'SP800-171:

Requisiti base
Limitare l'accesso ai soli utenti autorizzati, i loro processi o i loro dispositivi
Limitare l'accesso ai tipi di transazioni/funzioni in base alle autorizzazioni dell'utente
Requisiti derivati
Controllare il flusso di CUI in base alle autorizzazioni
Separare i compiti degli individui
Usare il principio del minimo privilegio
Usare account/ruoli non privilegiati per funzioni non di sicurezza
Impedire agli utenti "normali" di usare funzioni privilegiate
Limitare i tentativi di accesso non riusciti
Fornire avvisi su Privacy e sicurezza in base alle norme in vigore
Interrompere le sessioni in caso di inattività prolungata
Terminare una sessione in presenza di certe condizioni
Controllare le sessioni remote
Usare la crittografia nelle sessioni remote
Indirizzare l'accesso remoto a dei punti di controllo
Autorizzare l'esecuzione remota di comandi privilegiati
Autorizzare l'accesso wireless prima di accettare le connessioni
Usare crittografia e autenticazione per gli accessi wireless
Controllare le connessioni mobili
Cifrare le CUI sui dispositivi mobili
Verificare/controllare/limitare le connessioni esterne
Limitare l'uso di dispositivi di archiviazione interni su sistemi esterni
Controllare le CUI sui sistemi accessibili al pubblico

Tabella 2: Requisiti del modello

Definizione Una politica di controllo definisce i tipi di accesso consentiti in quali condizioni e da chi.

3.1 Discrezionale

Il metodo tradizionale, si basa su 3 elementi:

1. **Soggetto**
2. **Oggetto**
3. **Permesso d'accesso**

Questi vengono organizzati in una matrice in cui le colonne sono gli oggetti, le righe i soggetti e le singole celle contengono i permessi del soggetto sull'oggetto.

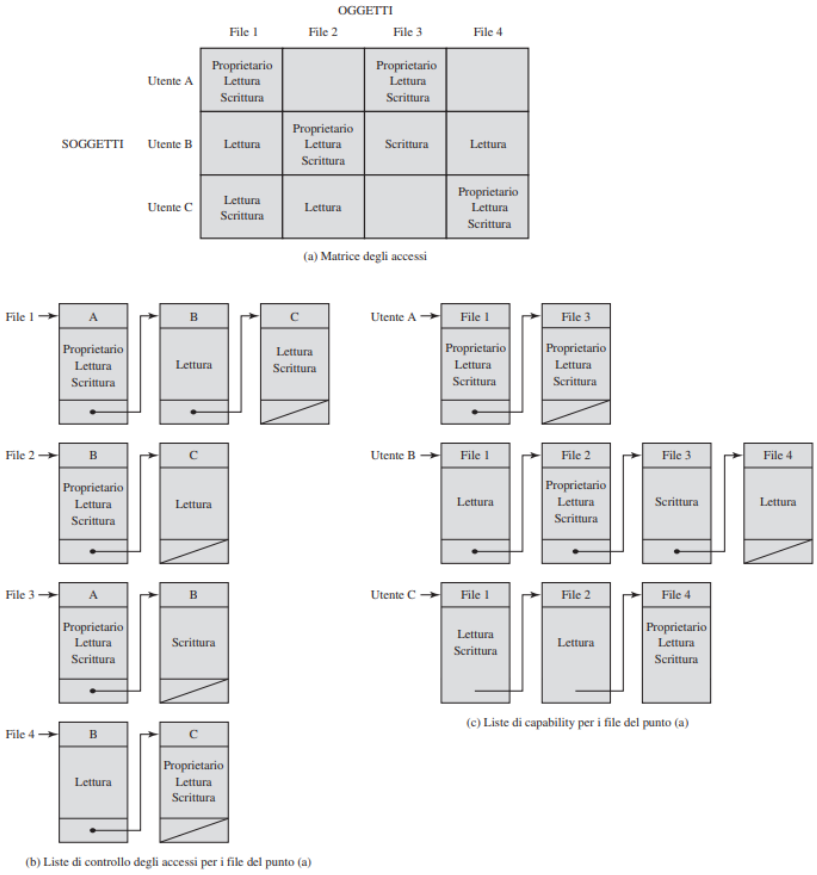


Figura 3: Esempio con file

La decomposizione per colonne fornisce le liste d'accesso dei file, quella per righe le capability list degli utenti.

Un'alternativa è la tabella delle autorizzazioni:

Soggetto	Modalità d'accesso	Oggetto
A	Proprietario	F1
A	Scrittura	F3
B	Lettura	F3

Il tutto si può estendere anche ad altri elementi:

- **Processi** (Cancellarli, bloccarli, riattivarli)
- **Dispositivi** (Lettura, scrittura, controllo)
- **Locazioni/Regioni di memoria** (Lettura, scrittura)
- **Soggetti** (Cancellare, concedere permessi)

Ottenendo così una matrice estesa, per eseguire l'effettivo controllo è presente un controllore per ogni tipo di oggetto che usa le Entry della matrice per decidere:

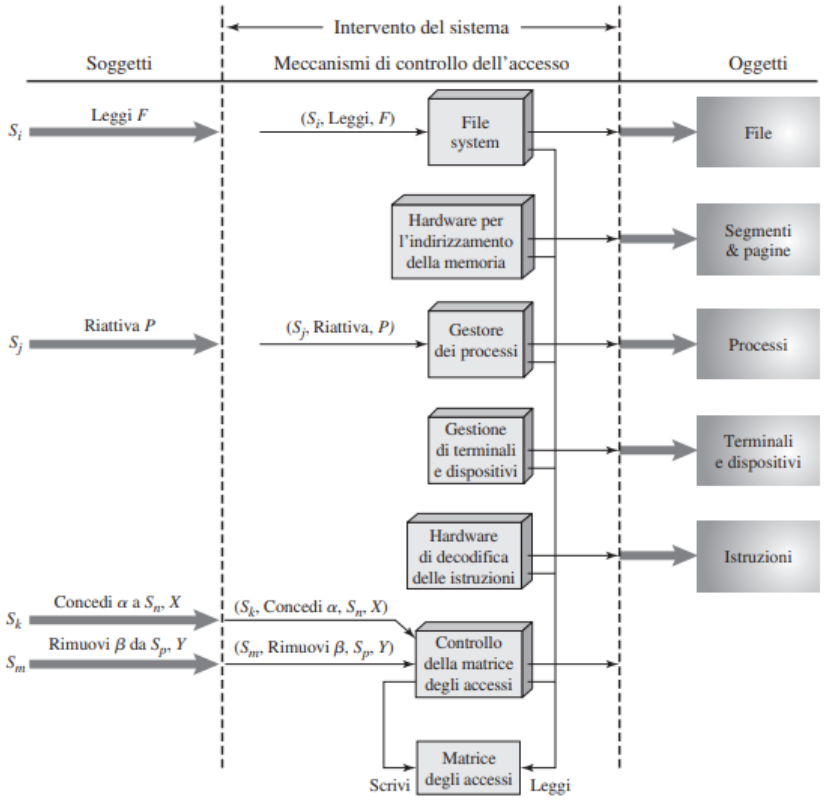


Figura 4: Esempio di funzionamento

3.2 Basato su ruoli

Questo sistema si basa sui ruoli che gli utenti possono assumere in un certo sistema (tipicamente con ruolo si intende una funzione lavorativa in un'organizzazione), l'assegnazione di questi ruoli avviene in modo dinamico mentre i ruoli tendono ad essere statici.

Una famiglia di modelli è la seguente:

- $RBAC_0$

  Contiene:

  - **Utenti**
  - **Ruoli**
  - **Permessi** (associati ai ruoli)
  - **Sessione** (mappatura tra utente e sottoinsieme dei suoi ruoli)

- $RBAC_1$

  Come il precedente ma si aggiunge una gerarchia tra i ruoli usando l'ereditarietà.

- $RBAC_2$

  Come $RBAC_0$ ma con l'aggiunta di vincoli:

  - **Cardinalità**, max utenti con certo ruolo
  - **Mutua esclusività**, un utente non può avere più di 2 ruoli esclusivi contemporaneamente
  - **Prerequisiti**, un utente ha un ruolo x solo se ha anche il ruolo y

- $RBAC_3$

  Unione di $RBAC_1$ e $RBAC_2$.

In breve:

Modello	Gerarchia	Vincoli
$RBAC_0$		
$RBAC_1$	x	
$RBAC_2$		x
$RBAC_3$	x	x

Anche in questo caso si possono usare le matrici, una per associare i permessi ai ruoli e un'altra per associare i ruoli agli utenti.

3.3 Basato su attributi

Un attributo rappresenta una certa caratteristica di soggetti, oggetti e condizioni ambientali, per controllare gli accessi vengono presi in considerazioni gli attributi di tutti e 3 gli elementi:

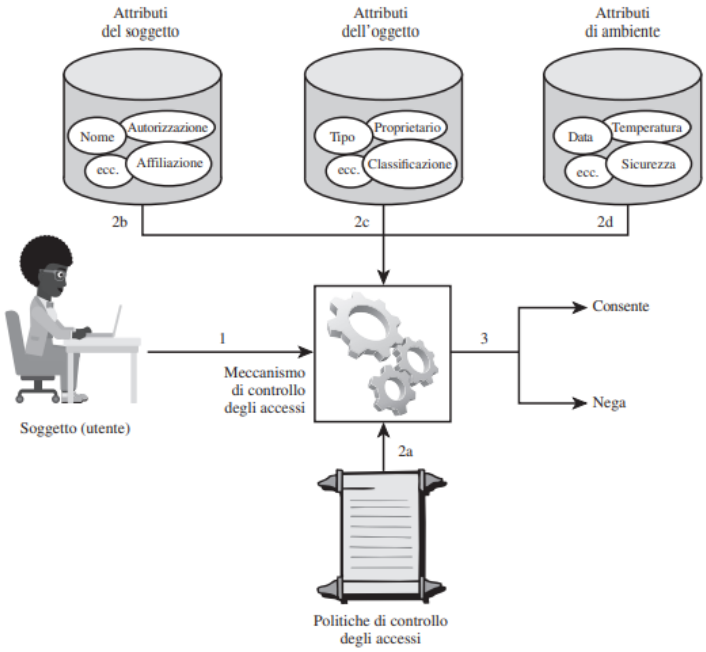


Figura 5: Funzionamento

Esempio:

Classificazione Film	Età minima
R	17
PG-13	13
G	-

Agli utenti viene assegnato un attributo riguardante la loro età durante la registrazione, per garantire che un utente acceda ad un film solo se ha almeno l'età minima basta una sola regola (ignorando l'ambiente):

$$\begin{aligned} R1 : can\_access(u, m, e) \leftarrow\ & (Age(u) \geq 17 \wedge Rating(m) \in \{R, PG-13, G\}) \vee \\ & (Age(u) \geq 13 \wedge Rating(m) \in \{PG-13, G\}) \vee \\ & (Age(u) < 13 \wedge Rating(m) \in \{G\}) \end{aligned}$$

---

4 Database

4.1 SQL Injection

Si tratta di attacchi progettati per estrarre/cancellare/modificare dei dati in un DB, solitamente sfruttano l'interazione tra il DB e le pagine web per inserire istruzioni malevole.

L'attacco consiste nell'inserimento di specifiche stringhe che vanno a modificare la Query costruita nel back-end, così facendo è possibile prelevare informazioni che normalmente dovrebbero essere private.

Esistono diversi modi per inviare le istruzioni:

- Input utente
- Variabili del server
- Cookie
- Input fisici

E diversi tipi di attacchi:

- In banda (stesso canale per invio e ricezione)
  - **Tautologia**
    Uso di istruzioni condizionali sempre vere.
  - **Commento a fine riga**
    Inserimento di '- -' a fine riga, tutto ciò che viene dopo diventa un commento.
  - **Query Piggybacked**
    Concatenazione di una Query a quella leggittima.
- Fuori banda (canali diversi)
- Inferenziale (ricostruzione delle informazioni)
  - **Query illecite/sbagliate**
    Ricavo di informazioni dalla pagina di errore.
  - **Blind SQL**
    Invio di Query di tipo Vero/Falso ed analisi dei risultati.

Definizione L'inferenza è un processo che consiste nell'effettuare richieste leggittime e dedurre informazioni riservate partendo dai risultati ottenuti.

Le principali contromisure sono:

- Programmazione difensiva
  - **Pratiche manuali**
    Controllo del tipo di dato, riconoscimento sequenze anomale.
  - **Inserimento parametrizzato**
    Definizione dettagliata della Query, inserimento sequenziale dei parametri.
  - **SQL DOM**
    Insieme di classi per la validazione automatica.
- Rilevazione
  - **Su firma**
    Individuazione delle specifiche sequenze di attacco.
  - **Su anomalia**
    Definizione di modello comportamentale, individuazione di attività che si discostano da esso.
  - **Analisi del codice**
    Uso di specifici test per rilevare le vulnerabilità.
- Prevenzione a Run-time
  Controllo delle Query a run-time con appositi strumenti.

4.2 Controllo degli accessi

I DBMS implementano dei sistemi di controllo discrezionali e a ruoli, e si basano sul presupposto che il sistema autentichi ogni singolo utente.

Lo stesso linguaggio SQL ha 2 comandi che permettono di concedere/revocare ruoli agli utenti e permessi ai ruoli:

- $GRANT$
- $REVOKE$

Scegliendo alcune opzioni è possibile usarli in cascata, nel caso della revoca ad un utente che ha ricevuto lo stesso permesso da più utenti si segue la regola "*Quando un utente revoca un permesso verrano revocati tutti a cascata, a meno che un certo permesso sarebbe esistito anche se l'utente non avesse concesso il permesso*".

4.3 Cifratura

La cifratura rappresenta l'ultimo sistema di difesa per un DB, essa può essere applicata a diversi "livelli" (attributi, record, campi, ...), porta però a 2 problemi:

1. **Gestione delle chiavi**

   Ogni utente deve conoscere la chiave per accedere ai dati e gli utenti interessati possono essere molti.

2. **Rigidità**

   La ricerca dei dati diventa più complessa.

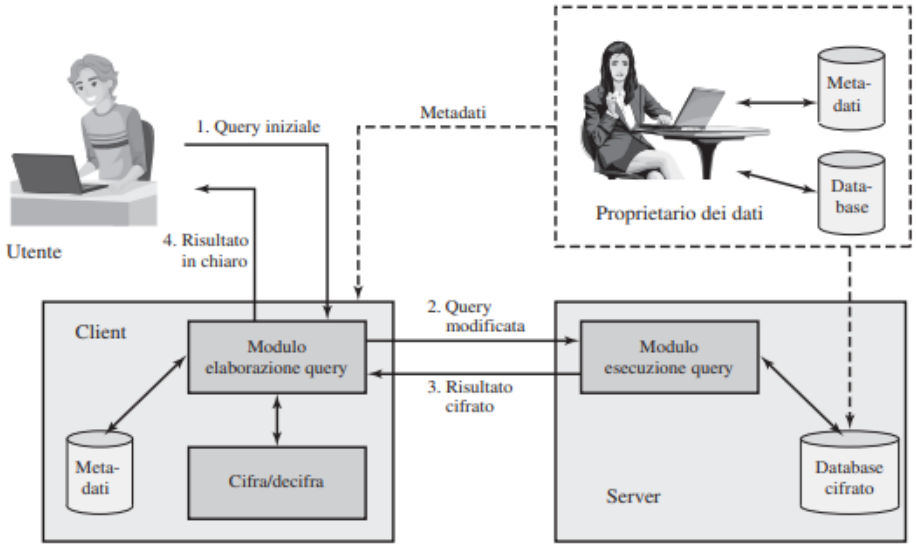


Figura 6: Funzionamento

Le 2 opzioni principali sono:

1. **Cifrare tutto**

2. **Cifrare i record**

   Ogni riga viene cifrata come fosse un blocco unico e vengono aggiunti degli indici per gli attributi (non necessariamente tutti) $[E(k, (x_1|x_2|\ldots)), I_1, I_2, \ldots]$

   Gli indici servono per semplificare la ricerca, i valori assumibili da essi rappresentano ognuno un sottointervallo dei valori assumibili da quell'attributo, ovviamente il valore che assume in una riga dipende dal valore di quell'attributo nella stessa.

5 Software malevolo

Definizione Un malware è un programma che viene inserito in un sistema (solitamente di nascosto) con l'intento di compromettere il sistema stesso o i suoi dati.

Definizione Un kit di attacco è un insieme di strumenti che può generare automaticamente malware.

Definizione Un ATP è un tipo di attacco che si distingue dagli altri per la sua selezione accurata del bersaglio e la sua persistenza nel tempo.

Definizione Il payload sono le azioni malevole che svolge un malware.

5.1 Virus

Definizione Un Virus è un frammento software che infetta altri programmi inserendo in essi del codice che generi delle copie di sè stesso.

Solitamente è inserito in un programma e viene eseguito ogni volta che si esegue il programma relativo operando con i suoi privilegi. Gli elementi costitutivi sono:

- **Vettore di infezione**
- **Trigger**
- **Payload**

Attraversa 4 fasi durante la sua vita:

1. **Dormiente**
2. **Propagazione**
3. **Attivazione**
4. **Esecuzione**

Si può dare una classificazione per target:

- **Boot Sector infector**
- **File infector**
- **Macro Virus**
- **Virus multipartito**

Ed una in base al camuffamento usato:

- **Criptato**
- **Furtivo**
- **Polimorfo**

  Le sue copie hanno stesse funzioni ma pattern diversi.
- **Metamorfico**

  Ad ogni iterazione si ridefinisce totalmente.

In particolare i Macro Virus si "attaccano" ai documenti e sfruttano le operazioni Macro delle applicazioni apposite, questo li rende indipendenti dalla piattaforma e facilmente creabili.

5.2 Worm

Definizione Un Worm è un programma che cerca attivamente nuovi sistemi da infettare, sfrutta quelli già infetti come "trampolino di lancio".

I principali mezzi usati sono:

- E-mail
- Chat istantanee
- MMS
- Bluetooth
- Condivisione file
- Trasferimento/Accesso remoto

Le sue fasi sono simili a quelle del virus, nel dettaglio la propagazione tramite rete prevede 4 possibili strategie:

- **Casuale**

  Prova tutti i possibili indirizzi.
- **Hit-list**

  Redige una lista dei possibili bersagli.
- **Topologica**

  Sfrutta le informazioni della macchina.
- **Sottorete locale**

  Sfrutta la struttura della sottorete se riesce ad attraversare il firewall.

Un fatto importante è che la sua percentuale di diffusione presenta delle similitudini con i virus biologici, l'andamento si può esprimere con la formula:

$$\frac{d\ I(t)}{d\ t} = \beta I(t) S(t)$$

In cui:

- $I(t)$ è il numero di individui infetti al tempo t
- $S(t)$ è il numero di individui suscettibili all'infezione al tempo t
- β è il tasso di infezione
- $N = I(t) + S(t)$ è la dimensione della popolazione

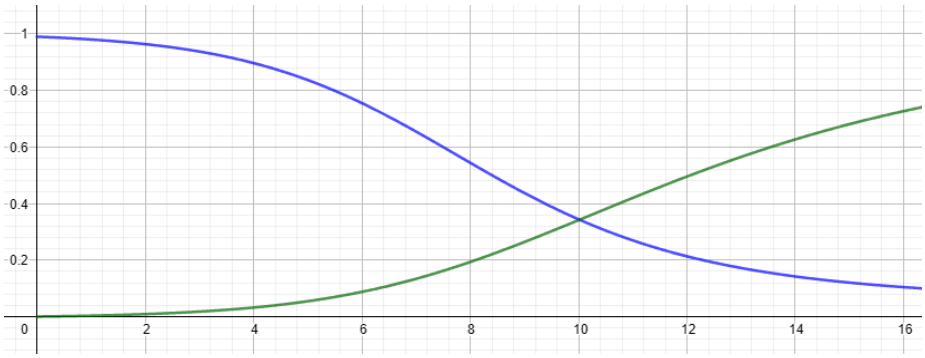


Figura 7: Esempio di andamento

Definizione Il mobile code è l'insieme di programmi che possono essere trasmessi invariati ad un insieme eterogeneo di piattaforme con la stessa semantica.

Alcuni metodi di diffusione più specifici sono:

- **Drive-by-download**

  Exploit di bug nelle applicazioni utente per installare malware, per esempio un sito web che scarica un file senza il consenso.

- **Watering-hole**

  Versione mirata del precedente, è preceduto da uno studio della vittima. In alcuni casi si sfrutta il Malvertising.

- **Clickjacking**

  Acquisizione dei click, forza l'utente a svolgere certe azioni.

5.3 Trojan

Definizione Un Trojan è un/una programma/procedura in apparenza utile che contiene codice malevolo nascosto.

I modelli principali sono 3:

1. Svolge sia la funzione normale che quella malevola
2. Svolge la funzione normale ma essa viene modificata
3. Sovverte completamente la funzione normale

Il loro mezzo di diffusione principale è lo Spam di e-mail, con l'ascesa dei telefoni cellulari però si è iniziato a diffonderli tramite i Marketplace delle applicazioni. I Trojan per smartphone risultano estremamente dannosi data la presenza di molte informazioni personali su di essi.

5.4 Payload

I principali tipi di Payload sono:

- **Ransomware**

  Criptano i dati del sistema e chiedono un riscatto per la chiave.

- **Distruzione dati**
- **Distruzione apparecchiature fisiche**
- **Creazione zombie/bot/droni**

  Essi sono sistemi infetti le cui risorse vengono usate per lanciare attacchi, i principali sono:

  - DDOS
  - Spamming
  - Sniffing del traffico
  - Keylogging
  - Diffusione malware

- **Keylogging**

  Lettura dei tasti premuti dall'utente.

- **Creazione Backdoor**
- **Creazione Rootkit**

  Come il precedente ma mantiene l'accesso con privilegi di amministratore.

5.5 Contromisure

Il metodo migliore per contrastare la diffusione dei malware resta la prevenzione, le misure sono piuttosto semplici:

- Avere i sistemi sempre aggiornati
- Controllare gli accessi ad applicazioni e dati
- Sensibilizzare gli utenti

Nel caso non basti si cerca di mitigare i danni:

- Eliminare i file infetti
- Usare i backup
- Cancellazione completa e successivo ripristino (in extremis)

A livello dei singoli dispositivi si fa uso di Antivirus, esistono diverse generazioni:

1. Identificazione con firma
2. Uso di criteri euristici e controllo integrità dei programmi
3. Analisi dinamica
4. Combinazione dei precedenti

Un tipo di analisi particolare è quella Sandbox, essa consiste nell'eseguire il codice in ambiente controllato (VM, emulatori) e monitorare l'andamento.

A livello di rete invece si usa una scansione perimetrale:

- Monitor di ingresso (tra rete interna ed internet)
  - Ricerca firme
  - Honeypot
- Monitor di uscita (tra sottoreti)
  - Controllo traffico in uscita
  - "Data-loss"

Il metodo migliore è una combinazione dei due, definendo un sistema centrale dedito all'analisi la rete diventa simile ad un sistema immunitario.

6 DOS

Definizione Un attacco DOS previene/incapacita l'uso di reti/sistemi/applicazioni esaurendo le loro risorse.

Definizione DDOS = attacco DOS distribuito, solitamente tramite net-bot.

I tipi principali sono:

- **Flooding**

  Il metodo più semplice, si inviano una grande quantità di pacchetti al bersaglio.

- **Spoofing indirizzo**

  Vengono "forgiati" degli indirizzi IP e poi usati nei pacchetti, rende il rintracciamento più difficile.

- **Spoofing Syn**

  Come il precedente ma il bersaglio è la tabella delle connessioni TCP, si cerca di mantenerla sempre piena per non permettere nuove connessioni. Dato il funzionamento di TCP l'uso di indirizzi non esistenti genera ulteriore traffico che va a congestionare ulteriormente la rete.

- **VoIP**

  Sfrutta le richieste *INVITE* (dispendiose a livello di risorse) del protocollo SIP, un flooding di queste richieste non permette al bersaglio di ricevere telefonate.

- **HTTP**

  Ci sono 2 varianti non banali:

  1. **Spidering**
     Si tratta di un flood di richieste HTTP ricorsivo, opera su una pagina e i link presenti in essa ricorsivamente.
  2. **Slowrolis**
     Cerca di mantenere la connessione con il server aperta per più tempo possibile, fa ciò inviando inizialmente una richiesta parziale e successivamente linee di header incomplete in modo periodico senza però finire mai la richiesta.

- **Di riflesso**

  Usando l'IP del bersaglio si fanno richieste continue ad un intermediario che invierà i risultati al destinatario.

- **Di amplificazione**

  Come il precedente ma si usano indirizzi di intere reti per generare ancora più traffico.

6.1 Difesa

Le linee di difesa sono 4:

1. Prevenire gli attacchi
2. Filtrare il traffico
3. Identificare la sorgente
4. Eliminare gli effetti dell'attacco

Per prevenire ci sono diversi modi:

- Limitare l'uso di indirizzi "forgiati"
- Assicurarsi che la via di ritorno del pacchetto sia quella giusta
- Limitare la quantità di certi tipi di pacchetti
- Gestire in modo diverso le connessioni TCP
- Bloccare gli indirizzi di broadcast
- Usare i Captcha
- Usare server specchiati/replicati

In ogni caso bisogna definire un piano di risposta:

1. Come contattare il personale ISP senza la rete
2. Imporre un filtraggio del traffico
3. Come rispondere

In particolare:

- Identificare il tipo di attacco
- Cercare di identificare la sorgente
- In caso di attacco duraturo avere un piano di contingenza
- Aggiornare continuamente il piano di risposta

7 Rilevamento intrusioni

I sistemi di rilevamento intrusioni sono composti da:

- Sensori
- Analizzatori
- Interfaccia grafica

Sono basati sul fatto che il comportamento di un utente legittimo differisce da quello di un attaccante, è comunque presente un certo grado di incertezza che può portare a falsi rilevamenti.

Esecuzione continua	Tolleranza ai guasti	Resistenza alla sovversione
Overhead minimo	Configurazione con *policies*	Adattamento ai cambiamenti
Scalabilità	Degradazione graziosa	Riconfigurazione dinamica

Tabella 3: Requisiti IDS

Funzionamento:

1. Creazione del modello comportamentale

   3 metodi:

   (a) **Statistico**
   Uso di modelli univariati/multivariati/a serie di tempo.

   (b) **Basato su conoscenza**
   Si usa un sistema esperto basato su certe regole.

   (c) **Basato su Machine Learning**
   Ci sono molti modi:
   - Reti Bayesiane
   - Logica Fuzzy
   - Reti neurali
   - Algoritmi genetici
   - ...

2. Rilevazione

   - **Euristica**
     Si usano delle regole prefissate.
   - **Con firma**
     Si comparano i pattern conosciuti con quelli presenti in un sistema o transitanti sulla rete.

7.1 Tipi

In base a dove operano si distinguono:

- **HIDS** (host)

  Possibili sorgenti di dati sono:

  - Chiamate di sistema
  - Log file
  - Checksum d'integrità
  - Accessi ai registri (Windows)

  Nelle LAN si possono creare dei sistemi HIDS distribuiti, alcuni nodi fungeranno da punti di raccolta ed analisi dati.

- **NIDS** (rete)

  2 tipi:

  1. **Inline**
     Dei veri e propri nodi, il traffico li attraversa.
  2. **Passivi**
     Ricevono una copia del traffico.

  Solitamente i punti in cui si inseriscono sono:

  - Sul Firewall esterno
  - Tra Firewall esterno ed Internet
  - Tra LAN e rete interna
  - Tra Backbone e rete interna

- **IDS ibridi/distribuiti**

  Quando i singoli nodi rilevano dell'attività sospetta fanno "gossip" agli altri nodi, se un nodo ne riceve una quantità considerevole presume ci sia un attacco e risponde di conseguenza.

Definizione Gli Honeypot sono sistemi esca creati appositamente per adescare gli attaccanti, hanno il compito di deviare gli attacchi lontano dai sistemi critici e allo stesso tempo incoraggiare gli attaccanti a restare più tempo possibile. Per fare ciò contengono delle informazioni fabbricate e fanno in modo che ogni attacco verso di loro vada sempre a buon fine.

Ce ne sono 2 tipi:

1. **A bassa interazione**

   Pacchetti software che emulano servizi/sistemi in modo realistico.

2. **Ad alta interazione**

   Sistemi veri e propri.

8 Firewall

Definizione Il Firewall è un componente hardware/software inserito tra la rete interna ed internet, funziona essenzialmente da filtro.

I tipi principali sono 4:

1. **A filtro di pacchetti**

   Si basa su delle regole riguardanti i campi dei pacchetti, in particolare:

   - IP sorgente/destinazione
   - Porta sorgente/destinazione
   - UDP/TCP
   - Interfaccia in/out

2. **Con filtraggio stateful**

   Mantiene una tabella delle connessioni TCP in uscita, permette ai pacchetti di entrare solo se riguardanti una connessione attiva.

3. **Gateway di applicazione**

   Fa da relay per per il traffico a livello applicativo, segue questo procedimento:

   (a) Utente chiede servizio al Firewall
   (b) Firewall chiede l'Host a cui accedere
   (c) L'Utente invia il nome dell'Host
   (d) L'utente fornisce il suo ID e le informazioni di autenticazione
   (e) Il Firewall contatta l'Host e poi inizia a trasmettere

4. **Gateway di circuito**

   "Spezza" le connessioni TCP in 2 e fa da intermediario:

   $$\text{Host interno} \longleftrightarrow^{TCP} \text{Firewall} \longleftrightarrow^{TCP} \text{Host esterno}$$

In base al posizionamento si identificano:

- **Bastion Host**

  Si trova in un punto critico per la rete.

- **Host-based**

  Posto su i singoli host, solitamente sui server.

- **Personali**

  Si trova sui PC "personali".

Usando 2 firewall (interno ed esterno) è possibile creare una DMZ in cui sono presenti i dispositivi che devono essere protetti ma allo stesso tempo devono essere accessibili dall'esterno.

Definizione Un VPN è un insieme di PC connessi tramite una rete non sicura che usano cifratura ed appositi protocolli per comunicare in modo sicuro, questo lavoro viene svolto solitamente tramite software dai router o dai firewall.

9 Buffer Overflow

Definizione Il buffer overflow è una condizione d'errore in cui vengono scritti in un buffer più dati di quanti ne possa contenere.

Solitamente si verifica a causa di un errore di programmazione, considerando inoltre che il buffer si trova nell'Heap/Stack/Sezione dati del programma c'è il rischio che gli elementi vicino ad esso vengano modificati.

Nel caso il buffer si trovi nella stack l'attacco classico punta a modificare l'indirizzo di ritorno nello stack frame, il modo più semplice è inserire una quantità di dati nel buffer tale che essi vadano a "risalire" fino all'indirizzo e modificarlo. Generalmente questo porta ad un crash del programma, in casi più sofisticati si inserisce nel buffer la "traduzione" di codice macchina insieme ad un *NOP-SLED* e si cerca di modificare l'indirizzo facendolo puntare al codice inserito per eseguirlo.

Alcuni metodi per prevenirlo sono:

- Scegliere un linguaggio di programmazione adatto
- Scrivere codice sicuro
- Usare meccanismi di protezione della stack
- Usare delle pagine di guardia
- Proteggere lo spazio degli indirizzi
- Randomizzare lo spazio degli indirizzi

10 Sicurezza del software

Nel mondo del Software ci sono 3 principali elementi che portano a degli errori:

1. Interazione non sicura tra i componenti
2. Gestione non oculata delle risorse
3. Difese deboli

Definizione La programmazione difensiva è il processo di programmazione e implementazione che garantisce il funzionamento del programma anche sotto attacco, esso deve continuare la sua esecuzione o fallire elegantemente.

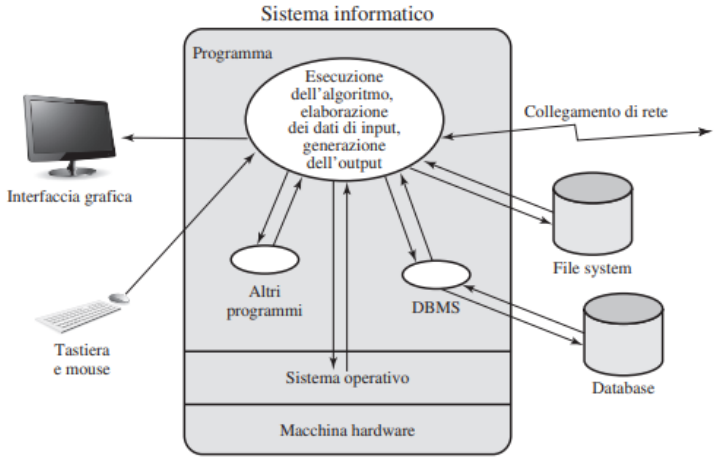


Figura 8: Modello di un programma

Gli elementi chiave da gestire sono:

- Codice stesso

  Implementare correttamente gli algoritmi, gestire opportunamente la memoria e interpretare correttamente i valori usati.

- Interazione con altri programmi/S.O.

  Fornire i privilegi minimi possibili, prevenire le race condition e controllare le librerie usate.

- Input

  Controllare le dimensioni, il tipo e accertarsi che sia conforme.

- Output

  Filtrare i dati definendo quelli ammissibili ed accertarsi della loro conformità.

11 Crittografia

Uno schema di cifratura è computazionalmente sicuro se:

- Il costo per rompere la cifratura è maggiore del valore delle informazioni cifrate
- Il tempo necessario per rompere la cifratura è maggiore della vita utile delle informazioni cifrate

11.1 Simmetrica

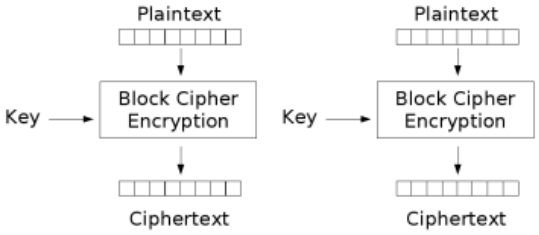
Questo tipo necessita che la chiave usata sia uguale ai 2 capi della comunicazione, si distinguono:

- **A blocchi** (DES,3DES,AES)

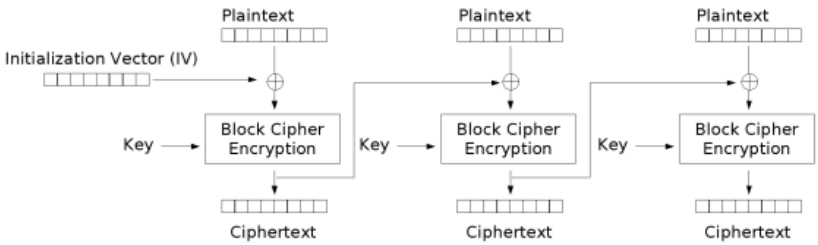
  Il messaggio viene diviso in blocchi che verranno poi cifrati ed inviati, sono presenti diversi modi per procedere:

  - **ECB**

    Si usano blocchi da 64 bit.



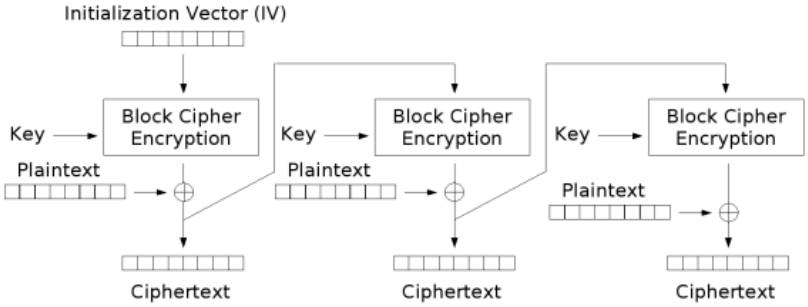
  - **CBC**



    Per riavere il blocco in chiaro si deve decifrare il blocco e poi fare lo XOR con il blocco cifrato precedente.

– **CFB**

Serve a trasformare la cifratura in blocchi in una cifratura di flusso.

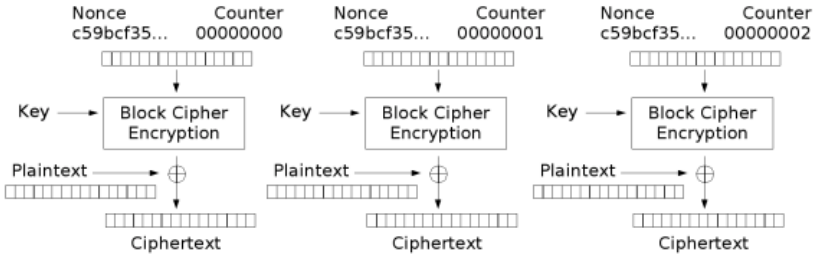


– **OFB**

Come il precedente ma l'input della prossima cifratura è preso prima dello XOR.

– **CTR**

Simile a ECB ma l'input della cifratura è un contatore incrementato ad ogni blocco ed il risultato è dato dallo XOR dell'output e del blocco in chiaro.



- **Di flusso** (RC4)

  Viene usata per cifrare un flusso continuo di informazioni, gli elementi vengono dati in output uno alla volta.

11.1.1 DES e 3DES

Questo schema è basato su una versione leggermente modificata della struttura di Feistel:

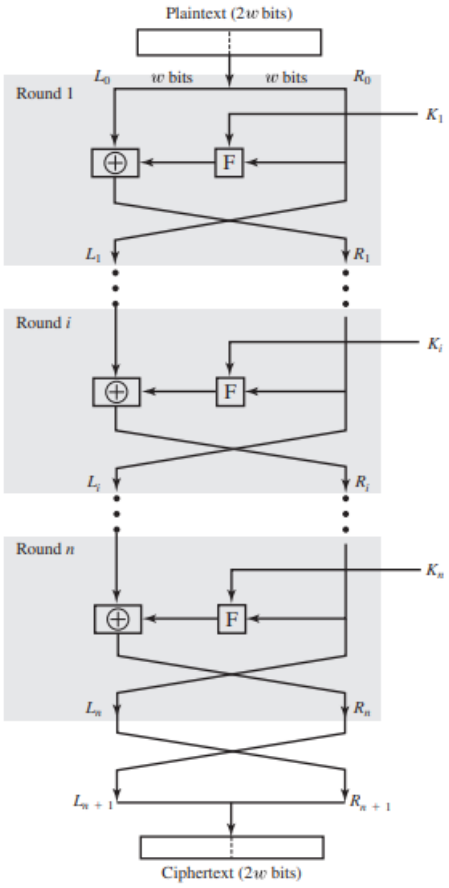


Figura 9: Struttura di Feistel

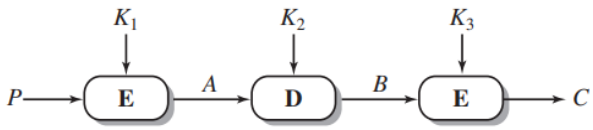
In cui:

- F è la funzione di round
- Le sottochiavi sono generate dalla chiave con una certa funzione

In particolare usa:

- Blocchi da 64 bit
- Chiave da 56 bit
- 16 round

3DES usa 3 chiavi ed applica DES 3 volte di fila nel seguente modo:



11.1.2 AES

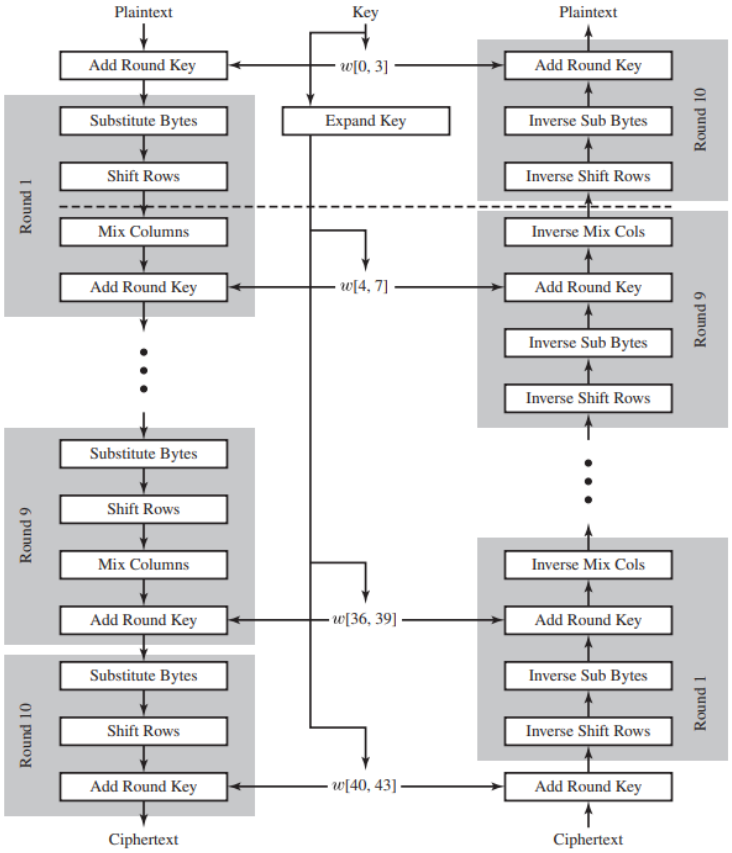


Figura 10: Funzionamento

Nel dettaglio:

- Blocchi da 182 bit
- Chiave da 128/192/256 bit
- 10/12/14 round in base alla chiave

- La chiave viene espansa tramite una funzione in un array lineare con grandezza dipendente dalla chiave, nei primi posti ci sarà la chiave stessa
- Il blocco viene copiato in un array detto stato su cui si eseguirà il procedimento e costituirà alla fine l'output
- **Substitute bytes**

  Si usa una *S-Box* di 16×16 bit contenente una permutazione dei valori assumibili da un byte (256), usando i 4 bit a sx e dx dello stato si trova il nuovo valore che costituirà lo stato
- **Shift Row**

  Su ogni riga dello stato si fa uno shift a sx circolare di $n-1$ posizioni con n numero di riga.
- **Mix columns**

  Ogni byte di ogni colonna ottiene un nuovo valore dato da tutti i byte presenti nella colonna stessa.
- **Add round key**

  XOR tra stato e chiave di round (per colonna)

11.1.3 RC4

Algorithm 1 Generazione stream

```
for i ∈ [0, 255] do                                   ▷ Inizializzazione
    S[i] = i
    T[i] = key[i  mod (len(key))]
end for

j = 0                                                 ▷ Permutazione iniziale
for i ∈ [0, 255] do
    j = (j+S[i]+T[i])  mod (256)
    Swap(S[i],S[j])
end for

i,j = 0                                               ▷ Generazione stream
while 1 do
    i = (i+1)  mod (256)
    j = (j+S[i])  mod (256)
    Swap(S[i],S[j])
    t = (S[j]+S[i])  mod (256)
    x = S(t)
end while
```

In breve:

- La chiave ha lunghezza $[1, 256]$ byte

1. S contiene $[0, 255]$
2. T contiene la chiave intera o ripetuta
3. S diventa una permutazione di se stesso
4. Si itera sugli elementi in S ed in base alla sua configurazione si effettua lo Swap

Per cifrare basta fare lo XOR tra x ed il byte del testo in chiaro correnti, per decifrare basta rifare lo XOR del byte cifrato corrente con x.

11.2 Asimmetrica

Si distinguono chiave pubblica e privata, con quella pubblica chiunque può cifrare un messaggio che potrà essere decifrato solamente con la chiave privata.

11.2.1 RSA

Il testo in chiaro M ed il testo cifrato C vengono visti come 2 interi tra 0 e $n-1$:

$$C = M^e \mod n$$

$$M = C^d \mod n = (M^e)^d \mod n = M^{ed} \mod n$$

Si devono soddisfare i seguenti requisiti:

- Si possono trovare e, d, n tali che $\forall\ M < n \quad M^{ed} \mod n = M$
- $\forall\ M < n \quad M^e, C^d$ sono facilmente calcolabili
- Non si può calcolare d partendo da e o n

La chiave pubblica sarà $\{e, n\}$ e quella privata $\{d, n\}$.

Il primo punto tiene solo se e, d sono inversi moltiplicativi modulo $\phi(n)$, quindi devono entrambi essere coprimi a $\phi(n)$.

Algorithm 2 Trovare le chiavi

Scegliere $p, q \mid p \neq q \wedge$ entrambi primi
$n = pq$
$\phi(n) = (p-1)(q-1)$
Selezionare $e \mid MCD(\phi(n), e) = 1 \wedge 1 < e < \phi(n)$
Calcolare d sapendo che $de \mod \phi(n) = 1$

11.2.2 Diffie-Hellman

Definizione La radice primitiva di un numero primo p è un numero a tale che:

$$a \mod p, a^2 \mod p, \ \ldots \ , a^{p-1} \mod p$$

Sono tutti distinti e generano tutti i numeri da 1 a $p-1$.

Definizione Dato p ed a sua radice primitiva. Per ogni $b < p$ si può trovare un numero i detto logaritmo discreto per cui:

$$b = a^i \mod p \ \text{ con } 0 \leq i \leq (p-1)$$

Questo algoritmo si basa sui logaritmi discreti e viene usato per scambiare delle chiavi che verranno poi usate per la cifratura simmetrica.

---

q numero primo ▷ Elementi pubblici
α radice primitiva di q

Scegliere un intero $X_A < q$ ▷ Privato
$Y_A = \alpha^{X_A} \mod q$ ▷ Pubblico
L'altro utente farà altrettanto ottenendo X_B e Y_B

Chiave segreta generata da $A = (Y_B)^{X_A} \mod q$
Chiave segreta generata da $B = (Y_A)^{X_B} \mod q$

Esempio:

- $q = 353$
- $\alpha = 3$
- $X_A = 97$
- $X_B = 233$

Si ha:

- $Y_A = 40$
- $Y_B = 248$
- $K_A = 248^{97} \mod 353 = 160 = 40^{233} \mod 353 = K_B$

11.3 Autenticazione messaggi

11.3.1 SHA

Con SHA si intende una famiglia di funzioni di hash sviluppate dall'NSA.

Funzionamento di SHA-512:

1. **Aggiunta di padding**

   Vengono aggiunti **sempre** dei bit di padding in modo che la lunghezza del messaggio sia congruente a 896 $\mod 1024$.

2. **Aggiunta della lunghezza**

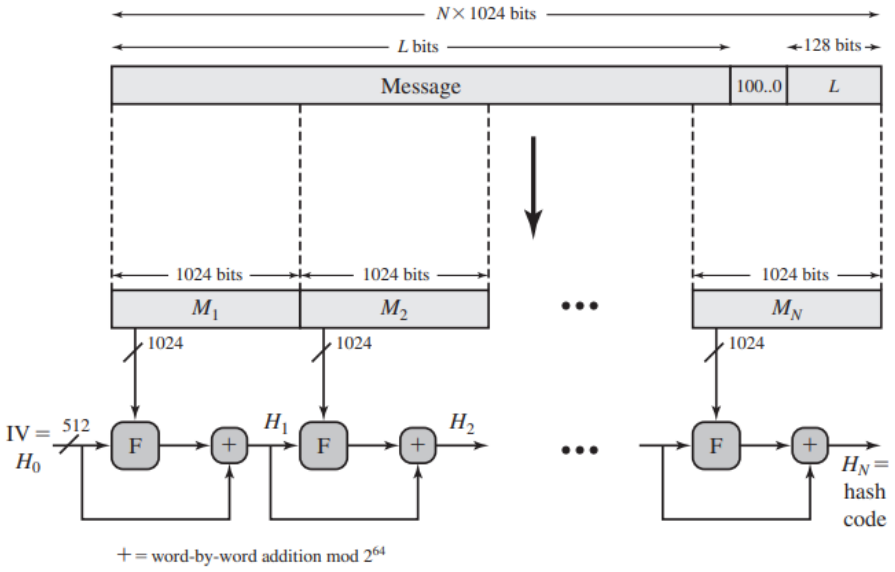
   Si aggiunge un blocco lungo 128 bit in cui si inserisce la lunghezza originale del messaggio nei bit più a sx (intero unsigned).

3. **Inizializzazione buffer**

   Si usano 8 buffer da 512 bit, vengono posti:

   - $a = 6A09E667F3BCC908$
   - $b = BB67AE8584CAA73B$
   - $c = 3C6EF372FE94F82B$
   - $d = A54FF53A5F1D36F1$
   - $e = 510E527FADE682D1$
   - $f = 9B05688C2B3E6C1F$
   - $g = 1F83D9ABFB41BD6B$
   - $h = 5BE0CD19137E2179$

4. **Processo dei blocchi**



5. **Output finale**

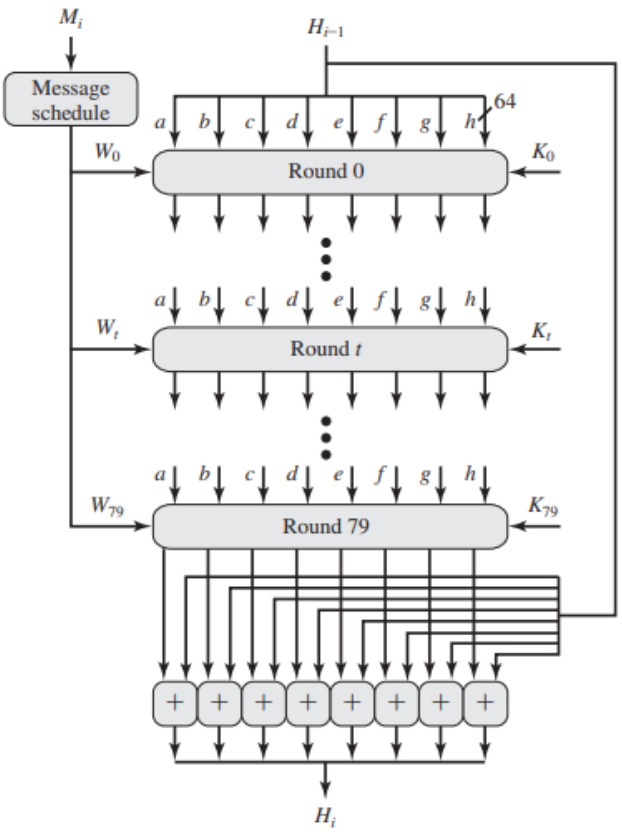


Figura 11: Processo di un blocco nel dettaglio

- W_i deriva dal blocco corrente
- K_i è una costante additiva
- Le operazioni svolte nel round sono quelle booleane primitive

11.3.2 Firma digitale

La firma digitale consiste nell'aggiungere alla fine del messaggio l'output di una specifica funzione (DSA,ECDSA) che richiede la chiave privata del mittente e l'hash del messaggio stesso.

Il destinatario potrà verificare che il messaggio è autentico facendo lo stesso procedimento con la chiave pubblica e confrontando l'output ottenuto con la firma presente.

11.3.3 HMAC

Si tratta di un approccio basato su hash che serve ad autenticare i messaggi, permette di usare qualsiasi funzione di hash esistente senza modificare il procedimento.

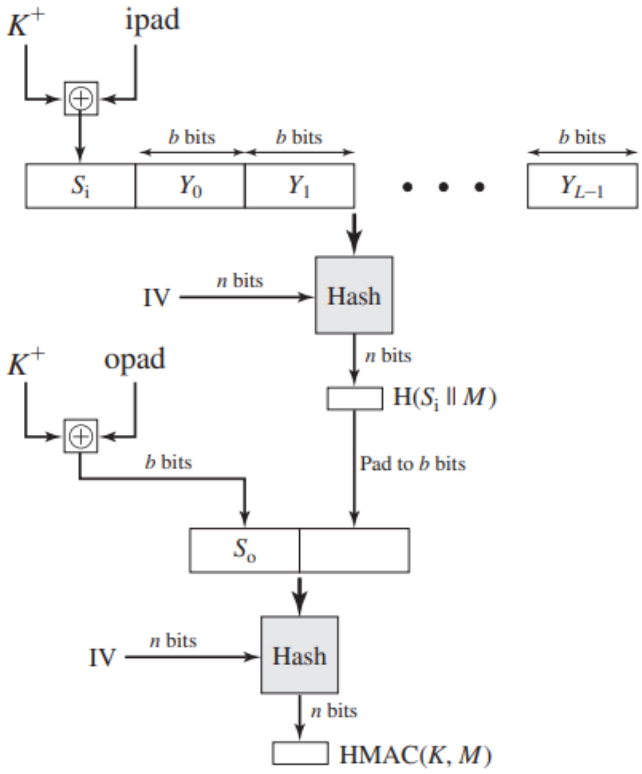


Figura 12: Funzionamento

- H è la funzione di hash usata
- M è il messaggio (compreso di padding se la funzione lo richiede)
- Y_i è l'i-esimo blocco del messaggio
- L è il numero di blocchi di M
- b è il numero di bit per blocco
- n è la lunghezza dell'hash prodotto
- K^+ è la chiave con eventuale padding di zeri a sx lunga b
- $ipad = 00110110$ ripetuto $\frac{b}{8}$ volte
- $ipad = 01011100$ ripetuto $\frac{b}{8}$ volte

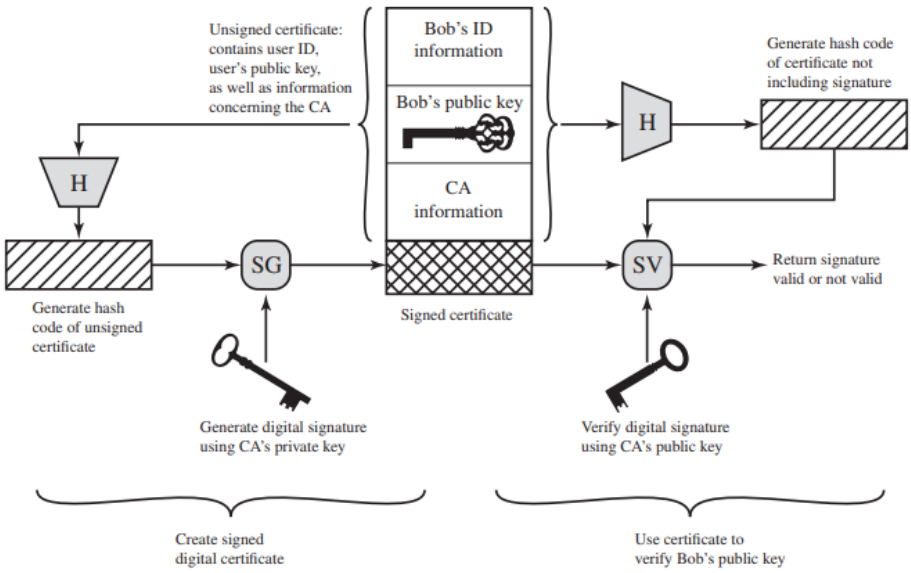
11.4 Altro

11.4.1 Certificazione della chiave pubblica

Per garantire che una chiave pubblica sia effettivamente associata ad un individuo si usa un certificato composto da:

- Chiave
- ID del proprietario
- Firma di un terzo
- Altre informazioni

Il terzo è un'autorità certificata CA che ha la fiducia della comunità.

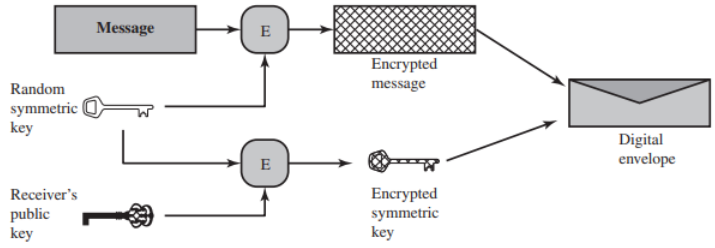


Procedimento:

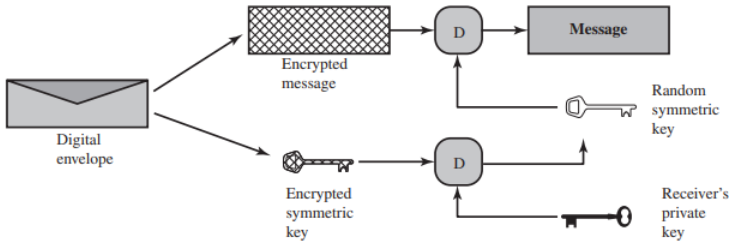
1. Il client genera un certificato non firmato e lo invia in modo sicuro
2. La CA genera una firma digitale e la applica al certificato
3. La CA invia il certificato firmato al client

11.4.2 Digital envelope

Una tecnica che permette di inviare un messaggio cifrato combinando la cifratura asimmetrica con quella simmetrica usando una chiave simmetrica monouso:



(a) Creation of a digital envelope



(b) Opening a digital envelope

11.4.3 Cifratura dei dati memorizzati

Alcuni casi di applicazione sono:

- **Dispositivo HW backend**

  Viene posto tra i server e i sistemi di memorizzazione, cifra il traffico dai primi ai secondi e decifra quello opposto.

- **Cifratura basata su libreria**

  Si inserisce un coprocessore embedded con chiave integrata nei dispositivi appositi per i nastri magnetici.

- **PC**

  Esistono dei programmi che permettono di cifrare dischi o partizioni di essi.