Appunti del corso di CyberSecurity e Reti dei calcolatori (0312212INF01II), a.a. 2023/24 – Corso di Laurea L-31 UNIPEGASO

RETI DI CALCOLATORI

- **Host:** sistemi periferici
- **Communication link:** collegamenti
- **Packet Switch:** commutatori di pacchetto

Velocità di trasmissione: bit per secondo

Commutatore di pacchetto: prende un pacchetto in in e lo converte in un pacchetto in out

I principali commutatori di pacchetto sono i **router (utilizzati all'interno di una rete)** e i **commutatori a livello di collegamento (utilizzati nelle reti periferiche, quelle di accesso al sistema).**

INTERNET SERVICE PROVIDER: insieme di collegamenti e di commutatori a livello di pacchetto gestiti da enti commerciali; **sono organizzati in una gerarchia a livelli.**

Sistemi periferici, commutatori ed ISP fanno uso di regole di comunicazione comuni detti **PROTOCOLLI.**

TCP (transmission control protocol): rende affidabile la comunicazione dei dati in rete tra mittente e destinatario.

IP (Internet Protocol): specifica il formato dei pacchetti scambiati tra router e sistemi periferici.

Gli standard per Internet vengono sviluppati dalla **IETF** (Internet Engineering Task Force); le pubblicazioni che ne trattano vengono chiamate **RFC** (request for comments). Anche L'**IEEE** (Institute for electrical and Electronic Engineers) si occupa di standard.

RETI DI ACCESSO: connettono un sistema periferico al suo EDGE ROUTER (il primo router sul percorso)

- **DSL** (Digital Subsciber Line): utilizza la **linea analogica telefonica**, con opportuni split sulle frequenze e tramite un **modem (mod**ulatore**, dem**odulatore di **frequenze)** dedicato, per la trasmissione di dati. La compagnia telefonica assume anche il ruolo di **ISP**. Tramite un **splitter**, che divide le frequenze e viene installato a casa dell'utente, utilizza un canale di **downstream** (in download) ad **alta velocita**, che viaggia sulla banda 50 Khz – 1 Mhz, e un canale **upstream** (verso l'esterno) a **media velocità**, su frequenze tra 4 e 50 Hz. Nelle frequenze tra 0 e 4 Hz è ospitato il **canale telefonico**.
  La linea dell'abitazione raggiunge un dispositivo definito DSLAM (Digital Subscriber Line Access Multiplexer), ospitato presso le cabine della compagnia telefonica, che fa appunto da Multiplexer sulle varie connessioni dati delle abitazioni connesse. DSL è un protocollo asimmetrico, in funzione delle differenti velocità di download ed upload.
- **FTTH (Fiber To The Home)**: collegamento in fibra ottica fino alla casa dell'utente, con specifiche **PON (Passive Optical Network)**. Utilizza un cavo ed apparecchiature dedicate. In ogni abitazione è presente un **ONT (Optical Network Terminator), che converte il segnale ottico in digitale.** Ogni ONT è connesso ad uno splitter ottico di quartiere che gestisce una serie di connessioni portate tramite una **fibra ottica condivisa**; essa è a sua volta collegata ad un terminale chiamato **OLT** (Optical Line Terminator) **situato nella cabina dell'ISP che svolge sostanzialmente il ruolo di MODEM**.

LAN Ethernet: connessione di più dispositivi all'edge router tramite **switch ethernet**; i terminali sono connessi con doppino in rame (RJ45).

WLAN (Wireless Logical Area Network): rete locale gestita tramite onde radio a bassa frequenza; i terminali si connettono ad un **WAP (Wireless Access Point)**; si seguono le specifiche IEEE 802.11.

Accesso satellitare: usa gli stessi segnali della trasmissione radio cellulare (3, 4, 5 G).

PACCHETTO: parte di una trasmissione digitale (bit) comprensiva di intestazione dedicata;

TRASMISSIONE

STORE AND FORWARD: il router può trasmettere in out il primo bit del pacchetto ricevuto **solo quando si completa la ricezione del pacchetto stesso;**

Pertanto, visto un tempo di ricezione L del pacchetto, una velocità di trasmissione del router di R bps, e visto che il pacchetto non potrà essere trasmesso fino a quando non è stato ricevuto tutto il pacchetto, la velocità di trasmissione sarà:

2L/R secondi

In caso di 3 pacchetti, sarà 4L/R secondi e così via; quindi, in maniera più generale (N+1)L/R secondi.

Il **buffer di output** è la struttura che servirà a contenere i pacchetti in out fino al momento in cui potranno essere inviati. **-> (OVERFLOW del BUFFER di OUTPUT: perdita di pacchetti dovuta alla saturazione del buffer in uscita)**

Ogni router possiede una **tabella di inoltro**, che mette in corrispondenza gli indirizzi **IP (Internet Protocol address)** di destinazione scritti negli header dei pacchetti con i collegamenti in uscita del router stesso.

In una rete a commutazione di pacchetto, possono crearsi vari tipi di ritardo:

- ritardo di **ELABORAZIONE:** ritardo legato all'esame dell'header e alla determinazione di verso quale uscita dirigerlo;
- ritardo di **ACCODAMENTO:** ritardo legato alla coda di pacchetti precedentemente ricevuti ed accodati in attesa dell'instradamento; è possibile stimare l'intensità del traffico tramite **I = A/R, dove A indica il numero di bit in arrivo nella coda al secondo e R è la velocità di trasmissione del router in uscita;**
- ritardo di **TRASMISSIONE: dati L come numero di bit che compongono il pacchetto e R velocità in uscita del router, il ritardo di trasmissione è L/R secondi;**
- ritardo di **PROPAGAZIONE: è il tempo impiegato dal pacchetto, una volta mandato in out, a percorrere la strada tra il router A e il router B;** dipende dalla velocità di propagazione V in m/s e la distanza D da percorrere **(D/V);**

Il ritardo totale del router è dato dalla somma di questi ritardi. Esistono programmi come TRACEROUTE che permettono di verificare il percorso di un pacchetto sui vari nodi e visualizzare i tempi, stimando eventuali ritardi.

TROUGHPUT MEDIO End-to-End: è relativo alla trasmissione tra due sistemi End-to-End. È il tempo T necessario a trasferire il file F tra A e B:

T = (F / Troughput) secondi

Non è mai una misura fissa, in quanto dipende dalla velocità di interconnessione dei collegamenti che dal traffico della rete.

Possiamo quindi dire che:

Troughput medio EndtoEnd = minimo (Rs/Rc)

Dove Rs = connessione tra Server e Router ed Rc = connessione tra client e Router;

Quindi T = F/minimo(Rs, Rc)

Pertanto, spesso, il Troughput medio sarà direttamente collegato al collegamento più lento presente fra i vari host che la compongono.

Bisognerà, naturalmente, tenere in conto non solo le velocità di connessione ma anche i vari ritardi intervenuti.

TRE LIVELLI DI ISP

- **ISP di accesso (livello basso)**
- **ISP regionali (livello medio)**
- **ISP Tier-1 (livello alto, AOL, AT&T, Deutsche Telekom, Telecom Italia Sparkle etc...)**

Gli ISP di Tier-1 non pagano a nessuno i loro servizi (cartello); i livelli inferiori, invece, saranno costretti a noleggiare servizi ed infrastrutture dagli ISP di livello più alto **-> MULTIHOMING**, cioè la possibilità per gli ISP di connettersi a più ISP di livello superiore.

PoP (Point of Presence): sono gruppi di router che connettono i vari ISP; gli ISP forniscono spesso ad altri ISP spazio nei loro data-center per i device hardware dello stesso, con un servizio detto **HOUSING**. In questo modo, è possibile ridurre costi di manutenzione e vigilanza.

Gli **ISP tier-1** sono connessi tra di loro tramite una modalità detta **PEERING** (a pari livello); questo significa che non si addebitano nessun costo l'un l'altro

per lo scambio dati tra le loro reti; questa modalità viene ormai sfruttata anche da coppie di ISP di livelli inferiori. Gli stessi procedono alla gestione e alla manutenzione dell'infrastruttura dedicata allo scambio del traffico, tramite gli **IXP (Internet Exchange Point).**

A fianco di questa struttura gerarchica, si aggiungono fornitori di reti privati che si occupano di distribuire contenuti, come ad esempio Google. Essa trasporta dati solo da e verso server di proprietà di Google stessa; per evitare gli ISP di alto livello, essa sfrutta il peering con ISP di livello inferiore, collegandosi a loro direttamente o tramite IXP.

ARCHITETTURA A LIVELLI

Il livello più basso gestisce l'invio dei segnali lungo le connessioni fisiche tra dispositivi; gli altri livelli:

- eseguono azioni caratteristiche del proprio livello;
- forniscono servizi al livello superiore;

Si definisce una organizzazione a PILA (STACK) di protocolli parallela a quella dei livelli.

I pacchetti gestiti da un protocollo di un livello dell'architettura è diviso in:

- **PAYLOAD: contiene il pacchetto gestito dal livello superiore;**
- **HEADER: contiene informazioni aggiuntive gestite dal protocollo del livello;**

SUITE DI PROTOCOLLI ISO/OSI

7 LIVELLI:

1. **Livello FISICO: flusso di dati su connessioni tra dispositivi (doppino di rame, etc);**
2. **Livello di COLLEGAMENTO: servizi di trasferimento di dati tra nodi vicini;**
3. **Livello di RETE: comunicazione tra i nodi che compongono la rete, tra i vari sistemi;**
4. **Livello di TRASPORTO: livello che permette un trasferimento affidabile, con controllo degli errori e perdite dei pacchetti;**
5. **Livello di SESSIONE: sincronizzazione e scambio dei dati;**

6. **Livello di PRESENTAZIONE: interpreta il significato semantico dei dati forniti dal livello di applicazione indipendentemente dal formato in cui sono rappresentati;**
7. **Livello di APPLICAZIONE: ha funzione di interfacciare e fornire servizi per i processi relativi all'esecuzione delle applicazioni;**

SUITE DI PROTOCOLLI TCP/IP

4 o 5 LIVELLI (RFC 1122 del 1989):

1. **FISICO:** traferisce i **singoli bit**; dipende dal mezzo fisico (doppino in rame, onde radio etc...)
2. **COLLEGAMENTO:** trasferisce i **FRAME**, pacchetti del livello superiore (datagrammi), da un nodo a quello adiacente; utilizza i **protocolli Ethernet, Wifi etc...;**
3. **INTERNET o DI RETE:** trasporta i **DATAGRAMMI**, pacchetti del livello superiore (segmenti), per tutti i nodi che compongono la rete, tramite il **protocollo IP.**
4. **TRASPORTO:** il pacchetto si chiama **SEGMENTO**, **protocolli TCP che controlla il flusso e la consegna dei pacchetti tramite una connessione che rimane attiva, ed UDP che fornisce una trasmissione senza curarsi troppo della connessione, quindi non controlla pacchetti persi etc...**
5. **APPLICAZIONE:** i pacchetti si chiamano **MESSAGGI**, si parla di livello applicativo, quindi **protocolli HTTP, FTP, SMTP, DNS etc..**

Alcune volte, il livello COLLEGAMENTO e FISICO vengono riuniti nel livello collegamento.

L'inserimento del pacchetto del livello superiore nel payload di quello sottostante è detto INCAPSULAMENTO.

TIPI DI DdoS

- **Attacchi alla vulnerabilità del sistema:** blocco del servizio o crash del sistema tramite pacchetti specifici;
- **Bandwidth flooding:** l'attaccante manda una quantità di pacchetti che satura la capacità di banda del difensore;

- **Connection flooding:** l'attaccante stabilisce un enorme numero di connessioni e le lascia tutte aperte, esaurendo la possibilità di effettuare connesioni;

X.509 E I CERTIFICATI

X.509 è la definizione di un framework per fornire servizi tramite una auth su directory basata sull'uso di certificati; implementa l'uso della crittografia a chiave pubblica consigliando RSA come standard, e l'uso degli hash.

OGNI CERTIFICATO CONTIENE LA CHIAVE PUBBLICA DEL POSSESSORE ED È FIRMATO CON LA CHIAVE PRIVATA DELLA C.A. CHE LO HA EMESSO.

Alcuni esempi italiani sono Verisign, Comodo ed Actalis.

Il certificato non ancora firmato contiene le informazioni dell'utente, la sua chiave pubblica e le informazioni sulla C.A.; di questi dati viene effettuato un hash che viene firmato con la chiave privata della C.A. Tale firma viene concatenata con la parte del certificato non firmato. In questo modo, chi lo riceve potrà ricalcolare l'hash sulla parte non firmata e confrontarlo con quanto estratto dalla firma, usando per decifrarla la chiave pubblica della C.A.

Il certificato contiene:

1. **La versione, che dipende dal tipo di certificato;**
2. **Il numero di serie, assegnato dalla CA;**
3. **L'Id dell'algoritmo di firma;**
4. **Nome della C.A. emettitrice;**
5. **Periodo di validità;**
6. **Il nome del soggetto titolare del certificato;**
7. **La chiave pubblica del soggetto;**
8. **Estensioni (previste dalla Versione 3 in avanti)**
9. **Firma, cioè il contenuto degli altri campi crittografato con la chiave privata della C.A.**

Esistono liste di certificati revocati, rilasciati dalle C.A.; esse contengono:

1. **ID dell'algortimo di firma;**
2. **Nome dell'emettitore;**
3. **Data di creazione della lista;**
4. **Data del prossimo aggiornamento;**

5. **Un campo per ogni certificato revocato, composto da numero seriale e data di revoca;**
6. **Firma della lista di revoca;**

Esiste uno standard denominato PKIX (Public Key Infrastructure X.509) per la messa in opera su internet di una architettura basata su certificati. Essa definisce le entità in gioco e le funzioni di gestione (registrazione, revupero del certificato, pubblicazione, liste di revoca, etc...)

Le entità in gioco sono:

- l'utente finale;
- La C.A.
- L'autorità di registrazione (opzionale): supporta la C.A. per la registrazione degli utenti;
- L'emittente delle liste di revoca (opzionale);
- Il repository di certificati e di liste di revoca;

IPSEC

È un layer di sicurezza che lavora al livello del protocollo IP. Serve a:

- **rendere sicura l'infra dal monitoraggio non autorizzato del traffico;**
- **rendere sicuro il traffico fra gli utenti finali con meccanismi di autenticazione e crittografia;**

Contrasta gli attacchi come IP spoofing; il suo uso principale è creare le VPN.

Per garantire la sicurezza, si usano due protocolli basati su Extension Header che seguono l'intestazione principale IP:

- un protocollo di Auth denominato **Authentication Header (AH)**
- Un protocollo combinato di crittografia ed auth denominato **Encapsulated Security Payload (ESP)**

Quando si parla di sicurezza ed Auth IP, è necessario introdurre il concetto di **Security Association (SA); è una relazione monodirezionale tra un mittente e un destinatario che riguarda servizi di sicurezza relativi al traffico trasportato.**

Ogni SA è identificata da 3 parametri:

- **SPI (Security Parameters Index):** identifica l'associazione di sicurezza SA ed è trasportato nelle intestazioni AH e ESP;
- **Indirizzo IP di destinazione;**
- **Identificatore del protocollo di sicurezza:** identifica se l'associazione è di tipo AH o ESP;

A sua volta, ogni SPI è definito in un Security Association Database; di seguito i più utilizzati:

- **Sequence Counter Number:** valore a 32 bit, utilizzato per generare il campo sequence number in un header AH o ESP;
- **Anti-reply window:** indica se il pacchetto in ingresso è un reply;
- **AH-Information**: informazioni sull'intestazione AH;
- **ESP-Information:** informazioni sull'intestazione ESP;
- **Lifetime of this SA:** durata di questa SA;
- **IPSec protocol mode:** trasporto, tunnel o wildcard;

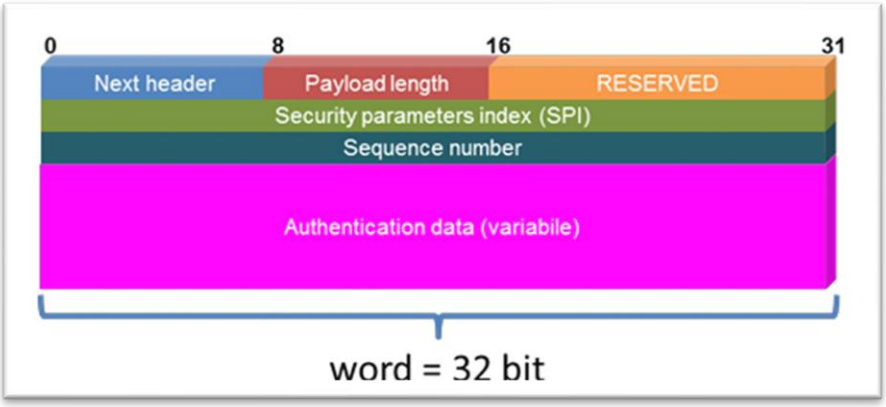
AH ed ESP supportano due modalità di funzionamento: trasporto e tunnel.

- **TRASPORTO:** si realizza la protezione dei protocolli di livello superiore; AH autentica il payload IP e di determinate parti dell'header IP, ESP esegue la crittografia, e opzionalmente autentica il payload IP.
- **TUNNEL:** si fornisce la protezione all'intero pacchetto IP; una volta che ad esso sono stati aggiunti i campi AH o ESP, l'intero pacchetto viene trattato come payload del nuovo pacchetto IP con destinazione esterna;

Authentication Header (AH)

Fornisce supporto per l'integrità dei dati e l'autenticazione dei pacchetti IP; non è possibile modificare il contenuto dei dati in transito senza che la modifica venga rilevata. L'autenticazione è basata su codice MAC, pertanto le due chiavi devono condividere una chiave segreta.

L'AH è così strutturato:

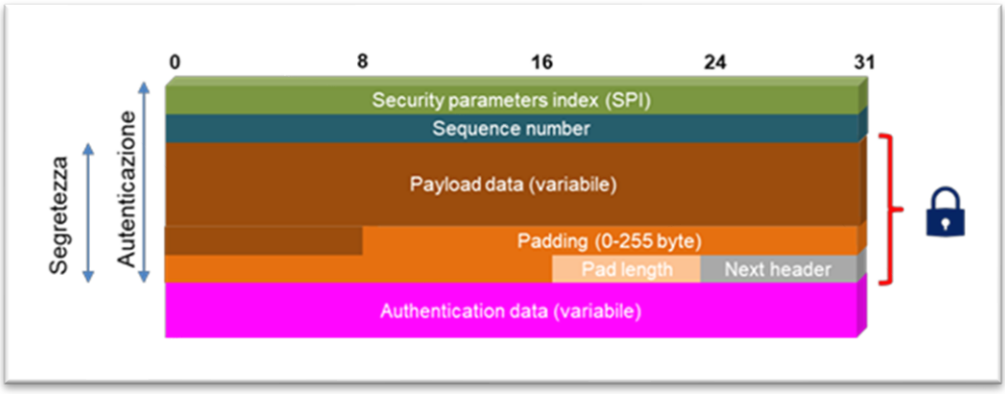


- **Next header (8 bit):** identifica il tipo di intestazione che segue immediatamente questa intestazione;
- **Payload Length:** la lunghezza di AH in WORD – 2;
- **Reserved:** campo riservato;
- **SPI:** identifica una associazione di sicurezza;
- **Sequence number:** contatore incrementale, **impedisce attacchi reply**;
- **Auth data:** ha lunghezza variabile, contiene il valore Integrity check Value o il codice MAC;

ENCAPSULATING SECURITY PAYLOAD (ESP)

Fornisce servizi di segretezza.

ESP è così strutturato



- **SPI:** identifica una associazione di sicurezza;
- **Sequence number:** contatore incrementale, **impedisce attacchi reply**;
- **Payload data:** è il segmento di **trasporto dei dati (modalità trasporto) o il pacchetto IP (modalità tunnel)** protetto dalla crittografia;
- **Padding e Pad Length:** gestiscono il riempimento in relazione alla dimensione della WORD, che è sempre 32 bit.
- **Next header:** identifica il tipo di dati contenuti nel campo payload data;
- **Authentication data:** contiene il valore Integrity Check come in AH calcolato sul pacchetto ESP senza, ovviamente, il campo Authentication Data;

I campi Payload, Padding, Pad Length e Next Header sono cirttografati tramite algoritmi Triple DES, IDEA, etc...

COMBINAZIONI DI ASSOCIAZIONI DI SICUREZZA

Possiamo combinare tra loro le associazioni di sicurezza, utili a fa coesistere AH ed ESP ad esempio, principalmente in due modi:

- **ADIACENZA DI TRASPORTO: consiste nell'applicare allo stesso pacchetto IP più protocolli di sicurezza senza utilizzare modalità tunnel;**
- **TUNNEL ITERATO: consiste nell'applicare più protocolli di sicurezza tramite la modalità tunnel; ogni tunnel può avere diversa origine o destinazione;**

GESTIONE DELLE CHIAVI IN IPSEC

Scelta fondamentalmente tra due protocolli:

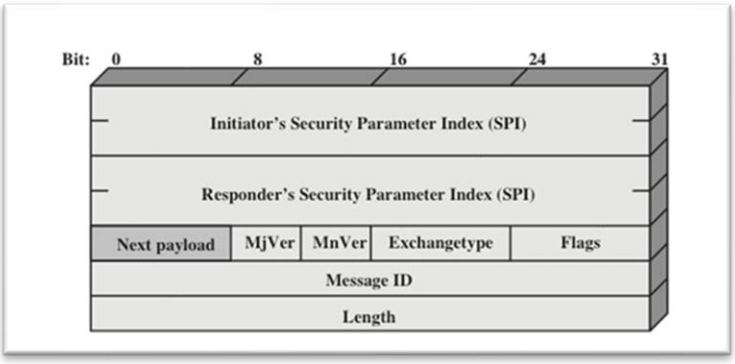
- **Oakley Key Determination Protocol:** protocollo di scambio chiavi basato su Diffie-Hellman, ma con sicurezza superiore;
- **Internet Security Association and Key Management Protocol (ISAKMP):** specifica una architettura per la gestione delle chiavi ed il supporto per la negoziazione delle associazioni di sicurezza;

Un messaggio ISAKMP è costituito da:

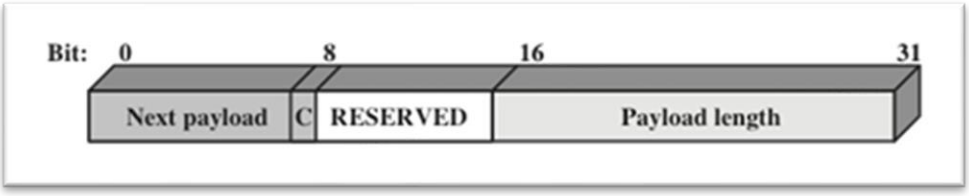
- una intestazione ISAKMP
- uno o più payload ISAKMP di differente tipologia che contengono, a loro volta, una intestazione generica;

L'intestazione ISAKMP contiene i seguenti campi:

- **Initiator cookie** (64 bit);
- **Responder cookie (**64 bit);
- **Next payload:** indica il tipo del primo payload;
- **Major/minor version:** versione usata;
- **Exchange type:** ne esistono 5 tipi;
- **Flags:** indica le opzioni per lo scambio;
- **Message ID:** codice univoco;
- **Length:** lunghezza totale del messaggio;



I Payload ISAKMP sono sempre preceduti da una intestazione generica:

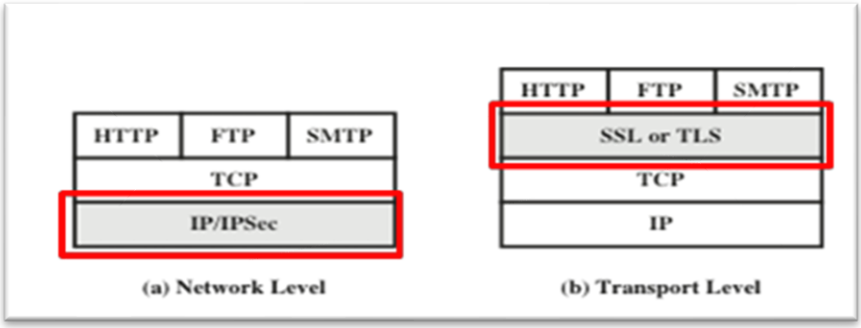


- **Next Payload**: vale 0 se è l'ultimo payload;

- **il nono bit è un bit critico**, indica al destinatario se questo payload deve essere scartato o meno sotto certe condizioni;
- **Payload Length:** lunghezza del payload compresa l'intestazione generica;

L'ARCHITETTURA SSL

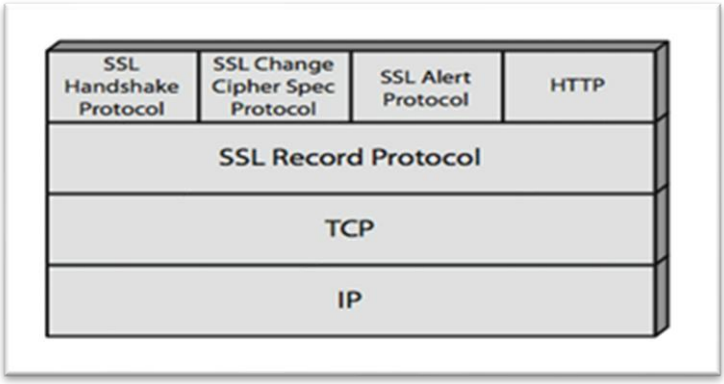
Mentre IPSec implementa soluzioni di sicurezza a livello di rete, SSL/TLS offrono sicurezza a livello di trasporto.



SSL (Secure Socket Layer) sviluppato da netscape, poi diventa

TLS (Transport Security Layer).

SSL è un insieme di protocolli crittografici che permettono una comunicazione sicura end-to-end su reti TCP/IP; un esempio è HTTPS. Esso è costituito da due strati di protocolli:



- **SSL record protocol:** offre servizi di sicurezza di base per i protocolli di livelli superiori;
- **SSL handshake Protocol, Change Chiper Spec Protocol e Alert Protocol** sono protocolli di più alto livello;

Essi si basano su due concetti distinti ed importanti:

Connessione SSL: forma di trasporto che fornisce un determinato servizio; le connessioni sono transitorie ed **ogni connessione è associata ad una sola sessione;**

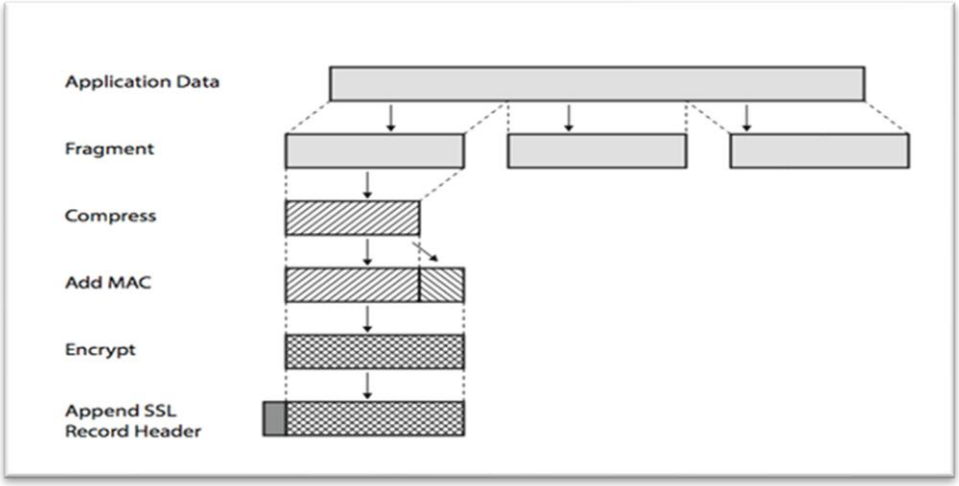
Sessione SSL: associazione tra client e server; vengono definite da un protocollo handshake e **definiscono un insieme di parametri di sicurezza che possono essere condivisi fra più connessioni;**

SSL RECORD PROTOCOL

Fornisce due servizi:

Integrità del messaggio: basato su codice MAC e K segreta condivisa tramite handshake;

Segretezza: realizzata tramite cifratura simmetrica con K segreta condivisa tramite handshake; il messaggio viene compresso prima di essere cifrato.



SSL CHANGE CHIPER SPEC ED ALERT

SSL CHANGE CHIPER SPEC: è composto da un singolo byte impostato ad 1, indica la richiesta da parte del mittente di variare la cifratura da usare per l'attuale connessione.

SSL ALERT: è costituito da 2 byte, il primo specifica il valore di severità dell'alert (warning o fatal), il secondo identifica il tipo di alert.

SSL HANDSHAKE

Consente di autenticarsi tra due host vicendevolmente, negoziare algortimi per crittografia e MAC, negoziare chiavi crittografiche; è composto da una serie di messaggi con la seguente struttura:

- **Tipo (1 byte):** indica il tipo di messaggio;
- **Lunghezza (3 byte):** indica la lunghezza del messaggio in byte;
- **Contenuto(>= 0 byte):** indica i parametri associati al messaggio;

Si svolge in 4 fasi:

1. **inizializzazione delle funzionalità di sicurezza;**
2. **Auth del server e scambio chiavi;**
3. **Auth del client e scambio chiavi;**
4. **Fine;**

IL PROTOCOLLO TLS

Evoluzione di SSL

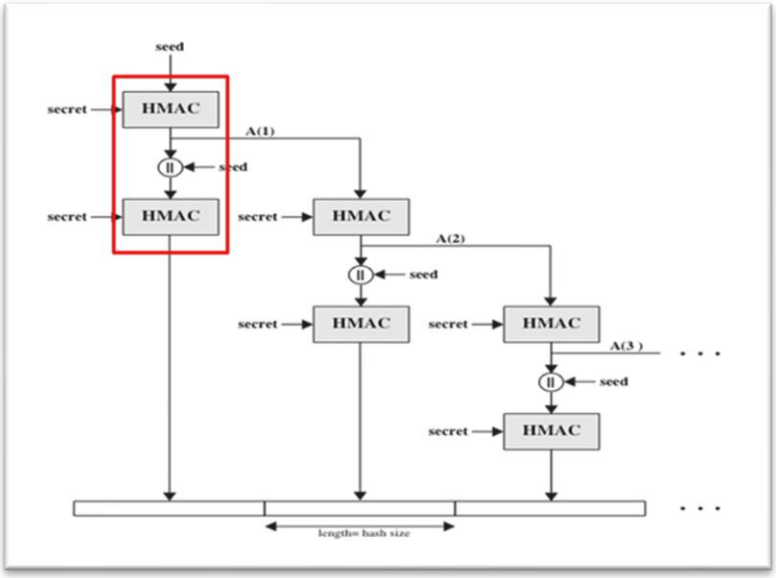
Espande e rende più solido il protocollo TLS; aggiunge anche alcune novità (funzionalità pseudocasuale, usa HMAC, codici allarme addizionali, padding sui contenuti dei campi).

LA FUNZIONE PSEUDORANDOM

Utilizzata per espandere i valori segreti in blocchi di dati per la generazione o la convalida della chiave; **l'obbiettivo è utilizzare un valore segreto condiviso piuttosto compatto per generare lunghi blocchi di dati sicuri.**

In ingresso percepisce due valori:

- **un seed;**
- **il valore segreto;**



l'iterazione dei due valori su sotto blocchi della lunghezza dell'hash di dimensione del codice HMAC produce blocchi di dati casuali della lunghezza necessaria. La PRF prevede l'utilizzo di algoritmi SHA-1 (digest 20 byte) ed MD5 (digest 16 byte) come algoritmi di base per la funzione HMAC.

IL PROTOCOLLO HTTPS

Hypertext Transfer Protocol Over Secure Socket Layer = http + SSL

Generalmente utilizza la porta 443 al posto della 80;

Cifra:

- **URL della risorsa richiesta;**
- **contenuti della risorsa;**
- **contenuti del form del browser;**
- **I cookies inviati in entrambe le sessioni (server-browser, browser-server)**
- **I contenuti dell'header http;**

PGP: INTRODUZIONE

Due standard per la posta elettronica:

- Pretty Good Privacy **(PGP)**
- Secure/multipurpose Internet Mail Extension **(S/MIME)**

PGP -> RFC 4880

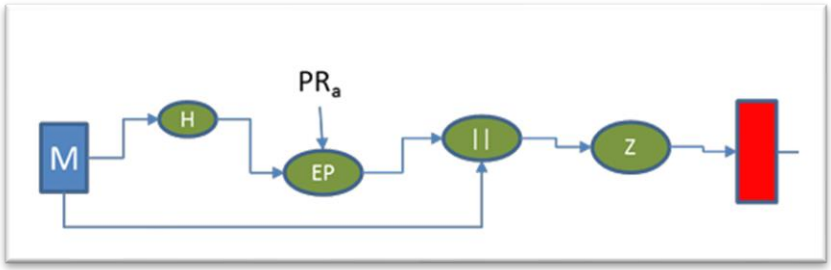
Basato su:

- **RSA, DSS e Diffie-Hellman per crittografia pubblica;**
- **CAST-128, IDEA e 3DES per crittografia simmetrica;**
- **SHA-1 per codifica hash;**

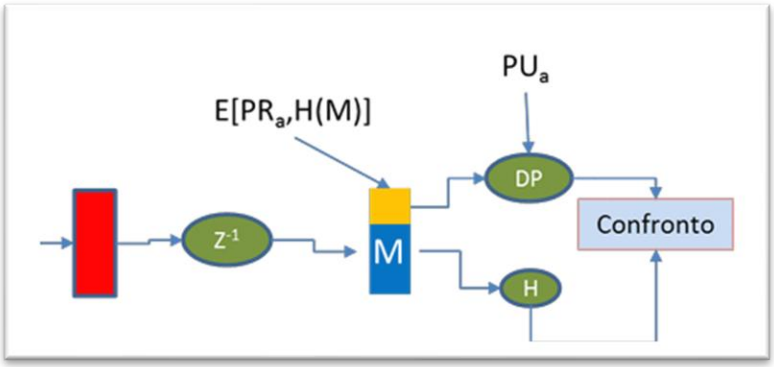
Comprende 5 servizi:

- **Autenticazione (firma digitale):** SHA1 e DSS o RSA;
- **Segretezza (crittografia):** cifratura simmetrica del messaggio con chiave di sessione che viene cifrata con crittografia a chiave pubblica;
- **Compressione:** basata su algoritmo ZIP;
- **Compatibilità con la posta elettronica:** basata su conversione ASCII radix 64;
- **Segmentazione (riassemblaggio)**

AUTENTICAZIONE



- Il mittente A genera un hash del messaggio tramite SHA1 e poi lo cifra con la propria chiave privata; il risultato viene concatenato al messaggio e zippato prima di inviarlo.



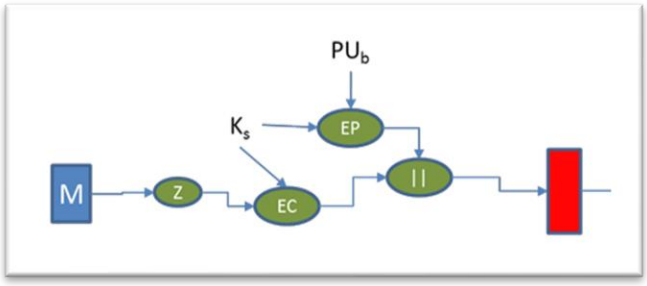
- Il destinatario B decomprime il pacchetto e riottiene il messaggio M e l'hash dello stesso cifrato con la chiave privata di A; a questo punto, può usare la chiave pubblica di A per decifrare l'hash, ricalcolare lo stesso tramite la funzione SHA1 sul messaggio non zippato e confrontarli; in quel

caso, l'autenticazione del mittente è garantita (a discapito della segretezza del messaggio).

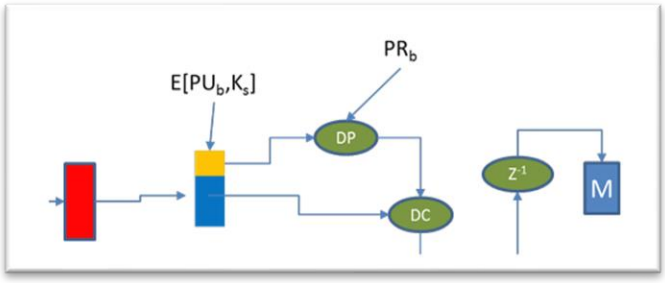
SEGRETEZZA

La segretezza viene garantita tramite crittografia simmetrica con cifratura CFB (chiper feedback) a 64 bit. La chiave K è una chiave monouso.

- Il mittente A comprime il messaggio M con ZIP; poi lo cifra con algoritmo simmetrico (i.e. 3DES) con la chiave di sessione Ks; tale chiave viene poi cifrata con RSA usando la chiave pubblica del destinatario Pub e concatenata con il messaggio cifrato prima di inviarlo:



- Il destinatario utilizza la sua chiave privata PRb per decifrare la chiave Ks, e poi usa questa per decifrare il messsaggio e decomprimerlo.



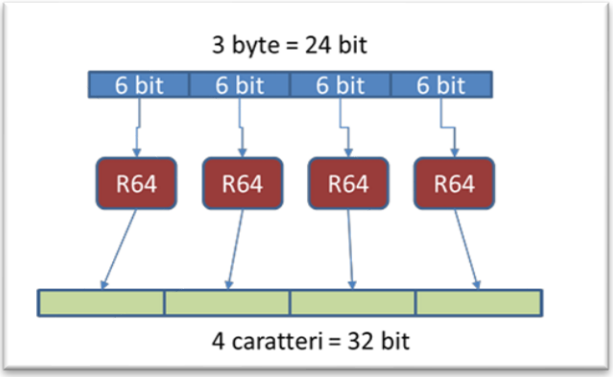
Per garantire sia segretezza che autenticità, sarà necessaria una doppia cifratura del messaggio utilizzando prima una cifratura tramite chiave privata del mittente e poi tramite chiave pubblica del destinatario.

LA COMPRESSIONE E LA CODIFICA

La compressione in ZIP del messaggio viene effettuata dopo aver applicato la firma, ma prima della crittografia a chiave simmetrica; questo perché:

- è preferibile effettuare la funzione di firma su un messaggio non compresso per facilitare la verifica della stessa;
- La crittografia viene applicata dopo la compressione, in modo da ridurre la ridondanza e la debolezza ad attacchi basati sulla frequenza;

La conversione in radix-64 mappa **gruppi da 8 bit in caratteri ASCII stampabili** e **gruppi di 3 byte in 4 caratteri ASCII, con una espansione del 33%.**



Ogni gruppo di 3 byte (24 bit) viene suddiviso in 4 gruppi da 6 bit; ognuno di essi viene mappato su uno dei 64 caratteri ASCII, che sono però costituiti da 8 bit l'uno.

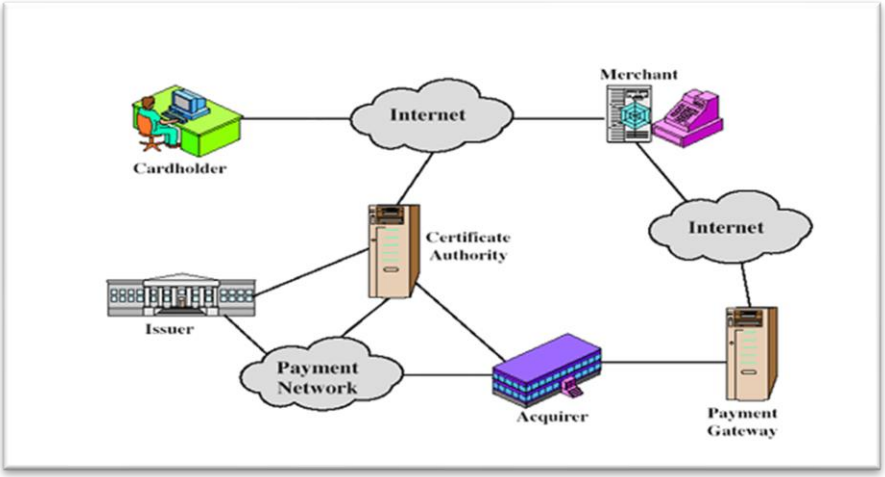
SECURE ELETRONIC TRANSACTION (SET)

CARATTERISTICHE:

- Segretezza tramite DES;
- Garanzia tramite Firma Digitale RSA e Hash SHA1;
- Corrispondenza tra possessore di carta di credito e intestatario del conto tramite firma digitale e certificati X509;
- Interoperabilità fra software e provider grazie a protocolli e formati indipendenti dalla piattaforma su cui si opera;

ATTORI COINVOLTI:

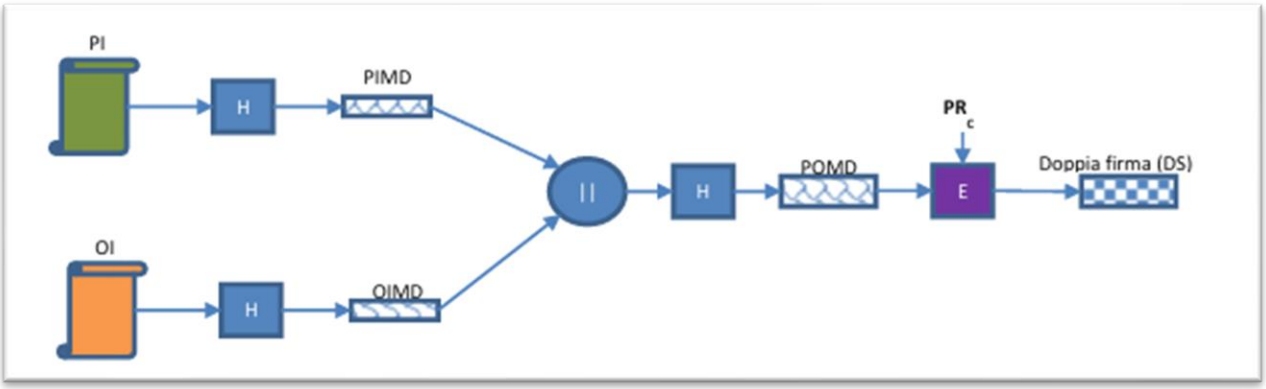
- Titolare della Carta di credito (**Cardholder**);
- Venditore (**Merchant**);
- Emettitore-banca (**Issuer**): responsabile del pagamento
- Acquisitore (**Acquirer**): istituzione finanziaria che tutela il venditore;
- Gateway di pagamento (**Payment Gateway**): elabora i messaggi di pagamento per il venditore;
- **Certification Authority**: emette i certificati;



Si utilizza il sistema di **doppia firma: il cliente emette due diversi messaggi destinati a due differenti destinatari:**

1. Le informazioni di ordine **(OI, Order Information) vengono inviate al venditore ma non il numero di carta di credito;**
2. Le informazioni di credito **(PI, Payment Information) devono essere inviate alla banca, ma non i dettagli dell'ordine;**

La doppia firma viene generata tramite un uso congiunto di hash SHA1 e della crittografia asimmetrica:



Vengono quindi calcolati i due hash, che vengono poi concatenati e sui quali viene realizzato un nuovo hash, cifrato poi tramite la chiave privata del cliente.

DS=E(PRC,[H(H(PI)||H(OI))])

A questo punto:

Il **venditore** riceve la doppia firma, le info dell'ordine OI e l'hash PIMD, quindi effettua tre operazioni:

1. Calcolare H(PIMD || H(OI));
2. Decifrare la doppia firma utilizzando la chiave pubblica del cliente (Puc, DS);
3. Confrontare i due risultati;

La banca riceve la doppia firma, le info di pagamento PI e l'hash OIMD, quindi effettua tre operazioni:

1. Calcolare H(H(PI) || OIMD);
2. Decifrare la doppia firma utilizzando la chiave pubblica del cliente (Puc, DS);
3. Confrontare i due risultati;

Così le informazioni di pagamento e di ordine vengono collegate dal cliente non possono essere alterate successivamente da banca o venditore;

LA RICHIESTA DI ACQUISTO

È suddivisa in fasi:

- **Initiate request: da cliente a venditore** il cliente richiede i **certificati del venditore e del gateway di pagamento**; invia anche un **ID della coppia request/response e un nonce per impedire reply attack**;
- **Initiate response: da venditore a cliente** invia il **nonce i**nviato dal cliente, un altro **nonce ed un IDr della transazione**, il tutto **cifrato** con la **propria chiave privata**; **include il suo certificato e quello del gateway di pagamento** come richiesto dal cliente;
- **Purchase request: da cliente a venditore verifica con la CA** i due certificati ricevuti, **crea le informazioni OI e PI e genera una chiave simmetrica monouso Ks con cui cifra la concatenazione di PI, hash OIMD e della doppai firma**; il risultato viene concatenato con **una busta digitale che contiene la chiave monouso cifrata con la chiave pubblica PUb del gateway di pagamento**;
- **Purchase response: da venditore a cliente** il venditore **verifica** con **la CA il certificato del cliente, la doppia firma, elabora l'ordine e inoltra le PI al gateway;** invia un **blocco di conferma che contiene IDt cifrato con la sua chiave privata** e, di nuovo, il suo certificato;

AUTORIZZAZIONE DI PAGAMENTO

È suddivisa in fasi:

- **Autorization request: da venditore a gateway di pagamento** inoltra un messaggio con le **PI e la busta digitale ricevuta dal cliente**; invia un **blocco di autorizzazione** che contiene **IDt firmato con la sua chiave privata e cifrato con una nuova chiave simmetrica monouso Kvs** del venditore;
- **Autorization response da gateway di pagamento a venditore verifica** i certificati; **decifra** la busta digitale inviata dal **venditore**, **estrae Kvs**, decifra il blocco di autorizzazione e **verifica la firma del venditore**; **decifra** la busta digitale proveniente dal **cliente**, **estrae Ks**, decifra il blocco di pagamento e **verifica la doppia firma**; **verifica IDt** ricevuto dal venditore **con quello contenuto nelle PI** ricevute indirettamente dal cliente; **richiede e riceve una autorizzazione dall'emettitore**.
  Invia un **blocco di autorizzazione cifrato con la chiave privata del gateway**, tutto cifrato con la **chiave Kgs**, inoltre include una **busta digitale contenente la chiave simmetrica Kgs, cifrata con la PU del venditore**; invia un **token di cattura del pagamento** che servirà nella fase successiva;

CATTURA DEL PAGAMENTO

È suddivisa in fasi:

- **Capture request: da venditore a gateway** invia il **blocco di cattura(somma del pagamento e IDt), firmato con doppia chiave privata e cifrato con la chaive pubblica del gateway;** reinvia il token ricevuto durante la Autorization response; **invia il proprio certificato e quello del gateway;**
- **Capture response: da gateway a venditore** decifra e verifica la firma sul blocco di cattura, decifra e verifica il token di cattura, **esegue il confronto tra richiesta di cattura e token,** invia una richiesta di liquidazione all'emettitore che esegue il trasferimento fondi a favore del venditore; **invia un blocco di risposta della cattura firmato con la sua chiave privata insieme al suo certificato al venditore, che memorizza la ricevuta di pagamento;**

FIREWALL

SCOPI:

- **proteggere la rete interna dagli attacchi esterni;**
- **fornire un unico punto di accesso su cui imporre sicurezza ed effettuare auditing;**

Esistono numerosi tipi di firewall, che offrono una grande gamma di servizi (auditing di rete sulla sicurezza, punto per vpn); per contro, frapponendosi lui fra interno ed esterno, non è utile contro attacchi provenienti dalla rete interna.

TIPI DI FIREWALL:

- **a filtraggio di pacchetti IP;**
- **a ispezione di stati:** controlla non solo i pacchetti, ma anche i protocolli e le connessioni;
- **Gateway a livello applicazione (proxy):** se il proxy non implementa una specifica applicazione, tale servizio non viene supportato;
- **Host bastione:** sistema identificato come punto critico per la sicurezza di rete che serve da gateway;
- **Host-based firewall:** modulo software che protegge un singolo host;

Sicurezza multilivello -> segmentazione dei privilegi ->

- **no read up**
- **no write down**

COMUNICAZIONI ANONIME

Proteggere le identità dei soggetti coinvolti:

- **reciprocamente;**
- **a terzi;**

PROTOCOLLO CROWDS

- **Un protocollo orientato all'anonimato:**
- **Ogni utente implementa un proxy sul proprio sistema;**
- **Tale proxy è definito jondo;**
- **Jondo si registra presso un server (blender) e viene ammesso al crowd;**
- **ogni richiesta in partenza passa attraverso il jondo;**
- **Jondo seleziona casualmente un altro Jondo sulla rete e inoltra la richiesta;**

- **Quando questo Jondo riceve la richiesta, c'è una probabilità del 50% che la inoltri al server finale, oppure ad un altro jondo;**
- **I collegamenti sono cifrati con le chiavi pubbliche dei vari jondo, e vengono stabilite quando il primo jondo entra nel crowd;**

PROTOCOLLO MIX

Basato su:

- **onion routing;**
- **mixing del traffico;**
- **generazione di traffico "dummy";**

LA RETE TOR

- **Perfect Forward Secrecy per le chiavi;**
- **Usa il concetto di "guardie" per aumentare l'anonimato dell'utente;**
- **Server accessibili solo tramite rete Tor;**
- **Sessioni per comunicare a lungo termine;**

Esistono dei **directory server** che possiedono il **Tor consensus file**; esso contiene la lista di tutti i relay con tutte le info su di essi; **I relay di entrata** ed **uscita** sono diversi da quelli nel mezzo.

Protocolli e software dedicati per accedere al livello più profondo del deep-web.

CYBERSECURITY

4 ATTACCHI ATTIVI:

- **MASCHERAMENTO**
- **RIPETIZIONE**
- **MODIFICA DEI MESSAGGI**
- **DOS**

RFC 2828 e IUT-T X800 – SERVIZI DI SICUREZZA

5 CATEGORIE DI SERVIZI E 14 SERVIZI SPECIFICI:

- **AUTENTICAZIONE**
  - **Autenticazione dell'identità peer -> garantisce origine in modalità connessa**

    - Autenticazione dell'origine dei dati -> garantisce origine in modalità non connessa
- CONTROLLO DEGLI ACCESSI

- SEGRETEZZA DEI DATI
    - Segretezza in modalità connessa -> protegge il contenuto del singolo pacchetto
    - Segretezza in modalità non connessa -> protegge il contenuto del singolo pacchetto
    - Segretezza selettiva a campi -> garantisce la segretezza solo di alcuni campi selezionati
    - Segretezza del traffico -> protezione dell'intero flusso di traffico

- INTEGRITÀ DEI DATI
    - Integrità in modalità connessa con ripristino -> rileva le modifiche e, eventualmente, ripristina
    - Integrità in modalità connessa senza ripristino -> rileva le modifiche ma non ripristina
    - Integrità selettiva a campi in modalità connessa
    - Integrità selettiva a campi in modalità non connessa
    - Integrità in modalità non connessa
- NON RIPUDIABILITÀ
    - Non ripudiabilità, origine
    - Non ripudiabilità, destinazione

MECCANISMI DI SICUREZZA

SPECIFICI:

- CRITTOGRAFIA
- FIRMA DIGITALE
- CONTROLLO ACCESSI
- INTEGRITÀ DEI DATI
- SCAMBIO DI AUTENTICAZIONE
- TECNICHE DI RIEMPIMENTO DEL TRAFFICO (I.E. PADDING)
- CONTROLLO DELL'INSTRADAMENTO
- AUTENTICAZIONE

PERVASIVI:

- **FUNZIONALITÀ FIDATA**
- **ETICHETTA DI SICUREZZA**
- **RILEVAMENTO DEGLI EVENTI**
- **AUDIT TRAIL**
- **RIPRISTINO DELLA SICUREZZA**

CRITTOGRAFIA

OPERAZIONI UTILIZZATE:

- **SOSTITUZIONE: mappatura di un elemento su un altro**
- **TRASPOSIZIONE: cambiamento di posizione degli elementi**

TIPOLOGIE DI CIFRATURA:

- **A BLOCCHI: elaborazione di un blocco alla volta**
- **A FLUSSI: elaborazione continuativa che produce un elemento alla volta**

ATTACCHI AD UN SISTEMA CRITTOGRAFICO:

- **ANALISI CRITTOGRAFICA -> cosa conosco? Plaintext, Cyphertext, Plaintext scelto e risultato…**
- **ATTACCO A FORZA BRUTA -> in media, funziona dopo aver provato metà delle chiavi, che dipendono dai bit di cui è composta la chiave stessa e dalla velocità di prove al secondo.**

CIFRARIO DI GIULIO CESARE: tecnica di sostituzione, sposto tutto l'alfabeto di un tot di posizioni.

TECNICHE DI SOSTITUZIONE MONO ALFABETICA E DI PLAYFAIR: mappo una lettera dell'alfabeto su di un'altra (mono), oppure uso digrammi cifrati con tabelle di cifratura 5x5 (I e J contano come una sola lettera).

TECNICHE DI SOSTITUZIONE VERNAM E ONE-TIME PAD: vernam prevede la scelta di una chiave lunga quanto il testo in chiaro e fa un XOR tra la cifra del plaintext ad ogni posizione, la corrispondente nella chiave e la scrive nel testo cifrato; la chiave può essere più cor del plaintext e quindi viene ripetuta, questo espone ad attacchi sulla frequenza delle ripetizioni. Nel One-time Pad, il meccanismo si evolve, generando una chiave univoca ogni volta, lunga quanto il

testo da cifrare e senza ripetizioni. È inviolabile, in quanto chiavi diverse potrebbero comunque produrre risultati sensati, senza indicare quale dei due è corretto.

TECNICHE DI TRASPOSIZONE:
ROLL-FENCE: il testo viene scritto su delle diagonali e poi letto in orizzontale.
CIFRARIO A RIGHE: dispongo il testo in un rettangolo seguendo le righe e poi lo si rilegge con l'ordine delle colonne indicato dalla chiave.
ENIGMA: più cilindri con 26 in e 26 out; ad ogni in rotazione di un cilindro, cambia l'abbinamento degli out di esso con gli in di quello successivo; esistono, con 3 cilindri, 17567 alfabeti disponibili.

SONO TUTTI TIPI DI CIFRATURA A FLUSSO
ORA VEDIAMO QUELLI A BLOCCHI
CIFRATURA DI FEISTEL

Si utilizza una cifratura a blocchi di grandezza n, in quanto se la chiave fosse grande quanto il blocco stesso (cifratura a blocchi ideale), le operazioni sarebbero troppo onerose per il sistema.

La cifratura di FEISTEL segue il seguente schema:

- **cifratura a blocchi di dimensione n bit**
- **Lunghezza della chiave di k bit**
- **Si ottengono 2^k possibili trasformazioni**
- **Alterna sostituzioni e permutazioni**

E implementa due concetti base:

- **DIFFUSIONE: la struttura del testo in chiaro viene "espansa" in un più ampio intervallo**
- **CONFUSIONE: si cerca di complicare il più possibile le statistiche tra il testo cifrato**

PASSAGGI:

divido il testo in due metà, LE e RE

la cifratura consta di 16 fasi; alla i-esima fase, la parte REi vieni sostituita sia nella parte LEi+1 che elaborata attraverso una trasformazione attraverso la func di fase F secondo la sottochiave Ki; successivamente, si ha un XOR con la parte

LEi per generare infine REi+1. La parte REi diventa semplicemente LEi+1 (nella fase successiva subirà lei la stessa trasformazione sopra descritta).

Le sottochiavi Ki sono tutte derivate da K e diverse tra loro.

CARATTERISTICHE:

- **dimensione blocco 64 bit;**
- **dimensione chiave 128 bit;**
- **16 fasi;**

Il processo di decrittografia è identico, ma si applicano le chiavi Ki in ordine inverso.

DES (Data Encryption Standard, 1997 - NIST)

CARATTERISTICHE:

- **dimensione blocco 64 bit;**
- **dimensione chiave 56 bit;**
- **stessi passi in cifratura e decifratura;**
- **uso di una serie di blocchi basati sui blocchi di Feistel:**
  - **Permutazione iniziale (IP)**
  - **16 ripetizioni (Round)**
  - **Swap seguito dall'inverso della permutazione iniziale (1P^-1)**
- **Anche la chiave segreta K subisce delle elaborazioni:**
  - **Permutazione iniziale**
  - **Generazione delle 16 sottochiavi Ki tramite la combinazione di uno scorrimento a sinistra e una permutazione**

PERMUTAZIONE INIZIALE: viene realizzata tramite tabelle di bit:

DES: Initial Permutation (IP)

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Quindi il bit in posizione 1 va in posizione 58, il bit in posizione 2 va in 57 etc...

Quando viene eseguita la permutazione finale, si ristabiliscono le posizioni originali.

DETTAGLIO DI UNA FASE:

il blocco di 64 bit viene diviso in 2 metà, Li ed Ri, che vengono elaborate tramite la cifratura di Freistel:

- **Li=Ri-1**
- **Ri=Li-1 XOR F(Ri-1, Ki)**

La funzione F(Ri-1, Ki) prevede:

- **Una espansione/permutazione per passare da 32 a 48 bit;**
- **Un XOR con la chiave a 48 bit Ki**

Con la seguente tabella di espansione:

Table 1: Expansion Permutation

32	01	02	03	04	05
04	05	06	07	08	09
08	09	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	01

La prima e l'ultima colonna sono ripetizioni di 16 bit, per estendere il blocco a 48.

Nella successiva funzione avviene una sostituzione basata su S-box, in particolare **8 S-box** (48 bit del blocco / 6 Bit in ingresso che richiede l'S-box), ognuna delle quali **accetta 6 bit in input e ne produce 4 in output**, basati su una matrice di permutazione costituita da 4 righe e 16 colonne (64 posizioni):

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	4	1	14	8	13	6	2.	11	15	12	9	7	3	10	5	0
	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Nei 6 bit, il primo e il sesto determinano il valore della riga, mentre gli altri 4 quello della colonna. I numeri risultanti vanno da 1 a 15, quindi rappresentabili tramite 4 bit.

Alla fine, avremo un nuovo blocco di 32 bit (4 bit di output X 8 S-box)

Viene eseguita la **permutazione P** prima di andare in XOR con Li-1, con la seguente mappatura:

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

GENERAZIONE DELLA CHIAVE:
la chiave a 64 bit subisce una prima permutazione **(Permuted Choice 1),** dove vengono scartati i bit multipli di 8, **ottenendo una chiave a 56 bit.**
Successivamente, essa viene suddivisa in 2 parti da 28 bit che subiscono trasformazioni indipendenti; ad ognuna delle 16 fasi, esse subiscono uno scorrimento a sinistra di 1 o 2 bit e poi vengono passate come input ad una ulteriore permutazione **(Permuted choice 2), che trasforma i 56 in 48 bit, che rappresentano la sottochiave Ki.**

Ogni piccolo cambiamento nel testo in chiaro porta a enormi differenze nel testo cifrato anche dopo pochissimi round di cifratura, grazie ad un effetto "valanga".

1999 -> TRIPLE DES -> blocchi a 128 bit, chiavi a 128, 192 o 256 bit., ottima sicurezza ma piuttosto lento

AES (Advanced Encryption Standard, Rijndael 2001 – NIST)

Algoritmo molto sicuro e performante, facilmente implementabile.

CARATTERISTICHE:

- **dimensione blocco 128 bit;**
- **dimensione chiave 128, 192 o 256 bit;**
- **Basato su 4 funzioni fondamentali:**
  - **Add round key**
  - **Byte substitution**
  - **Shift rows**
  - **Mix columns**

L'operazione di decifratura è diversa da quella di cifratura

La struttura dei due algoritmi è la stessa, ma le funzioni sono inverse; quindi serviranno inevitabilmente due moduli software distinti.

La dimensione della chiave influisce sul numero di fasi utilizzate (più è grande, più sono le fasi)

LE FUNZIONI IN DETTAGLIO:

Byte substitution

Il blocco da 128 bit viene considerato come una matrice di 16 byte (4X4); su **ogni byte** viene operata una permutazione con una S-Box contenente tutti i valori che può assumere il byte stesso (8x8 bit= 256). Gli 8 bit vengono mappati utilizzando i primi 4 come indice riga e secondi 4 come indice colonna.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Shift Rows

Il blocco da 128 bit viene considerato come una matrice di 16 byte (4X4); avvengono spostamenti circolari sulle righe con i seguenti criteri:

- **la prima riga rimane uguale;**
- **la seconda riga si sposta di un byte a sinistra;**
- **la terza riga si sposta di due byte a sinistra;**
- **la quarta riga si sposta di tre byte a sinistra;**

I byte che escono a sinistra rientrano a destra sulla stessa riga.

Mix Columns

Il blocco da 128 bit viene considerato come una matrice di 16 byte (4X4); viene eseguita una moltiplicazione fra i valori presenti nella matrice e la matrice seguente:

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}$$

Add round key

Il blocco da 128 bit viene considerato come una matrice di 16 byte (4X4); viene eseguito uno XOR bit a bit del testo in chiaro con la chiave.

Esiste poi una ulteriore operazione, **Expand Key,** che però riguarda solo la chiave; espande la chiave a 128 bit (4 WORD da 32 bit) a 44 WORD, utilizzando una serie di XOR e operazioni G (basata su scorrimenti e sostituzioni) delle singole WORD.

2DES

Consiste nell'applicare 2 volte in sequenza il DES utilizzando due chiavi diverse K1 e K2; è stato dimostrato che è differente cifrare due volte in sequenza con chiavi da 56 bit piuttosto che cifrare una volta sola con chiave a 112.

L'attacco MitM (Meet in the middle)

3DES

Consiste nell'applicare 3 volte in sequenza il DES utilizzando sole due chiavi diverse K1 e K2; infatti, cifrare e decifrare sono concettualmente equivalenti, potrei quindi:

- **prendere il testo in chiaro, cifrarlo con K1, "decifrarlo" con K2 e cifrarlo nuovamente con K1;**
- **e poi prendere il testo cifrato, decifrarlo con K1, cifrarlo con K2 e decifrarlo nuovamente con K1;**

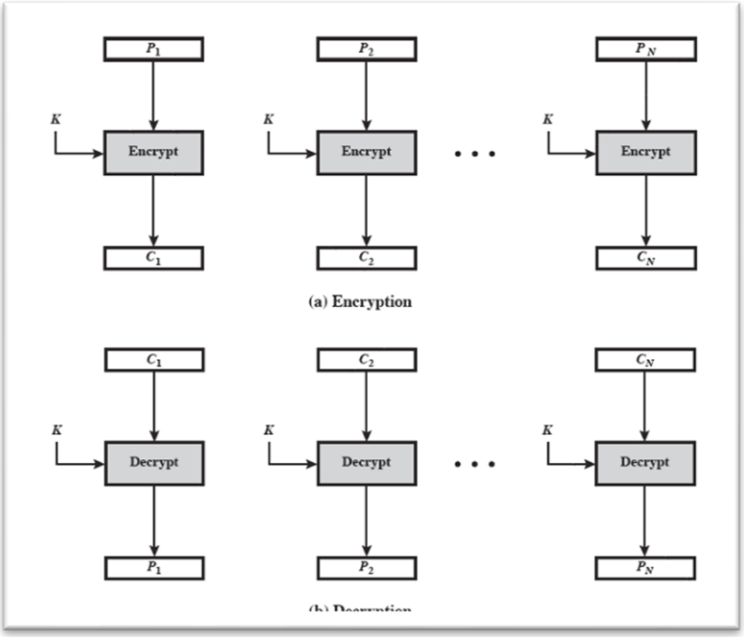
Viene comunque anche spesso utilizzata una chiave a 168 bit, unione di 3 chiavi a 56 bit.

MODALITÀ DI FUNZIONAMENTO DELLA CIFRATURA A BLOCCHI

1. **Electronic Codebook (ECB)**
2. **Cipher Block Chaining (CBC)**
3. **Cipher Feedback (CFB)**
4. **Output Feedback (OFB)**
5. **Counter (CTR)**

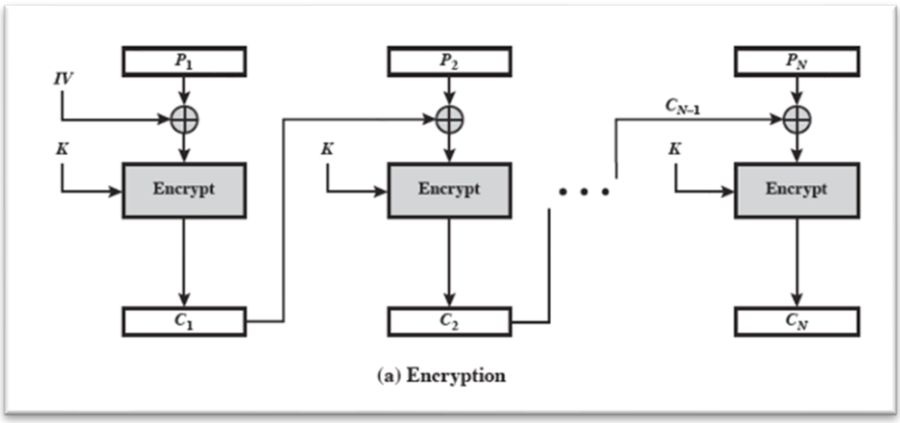
ELECTRONIC CODEBOOK (ECB)

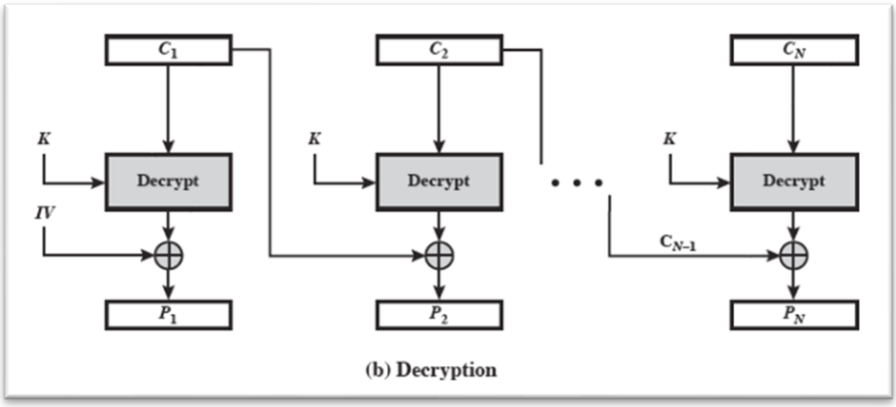
- **un blocco alla volta**
- **sempre con la stessa chiave.**



CIPHER BLOCK CHAINING (CBC)

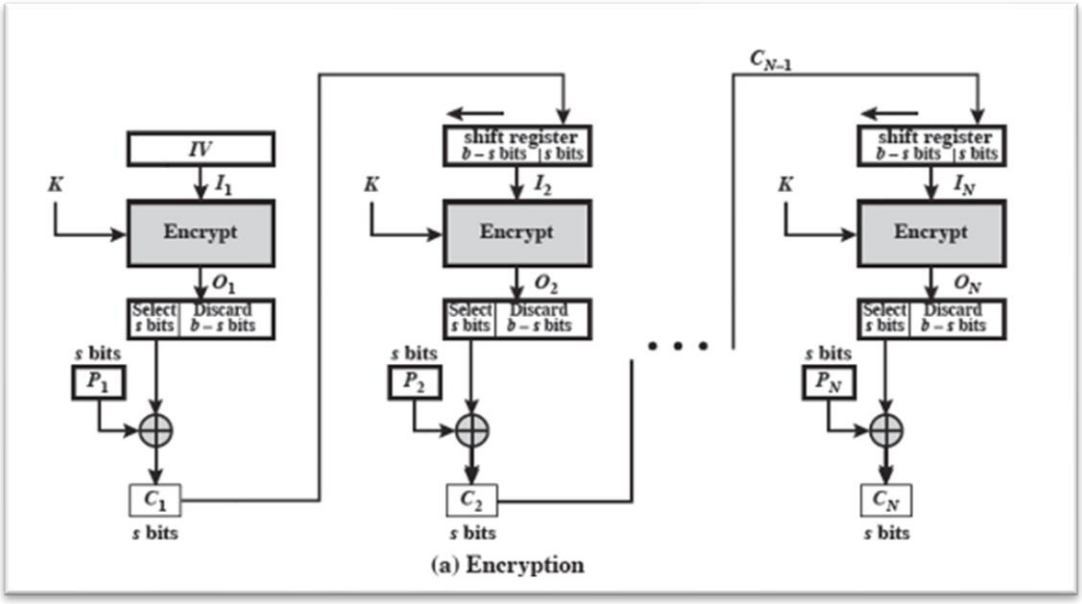
- **In-out chaining tra blocchi precedenti e successivi**
- **L'input è l'XOR tra il blocco di testo in chiaro e il blocco di testo cifrato al passo precedente.**

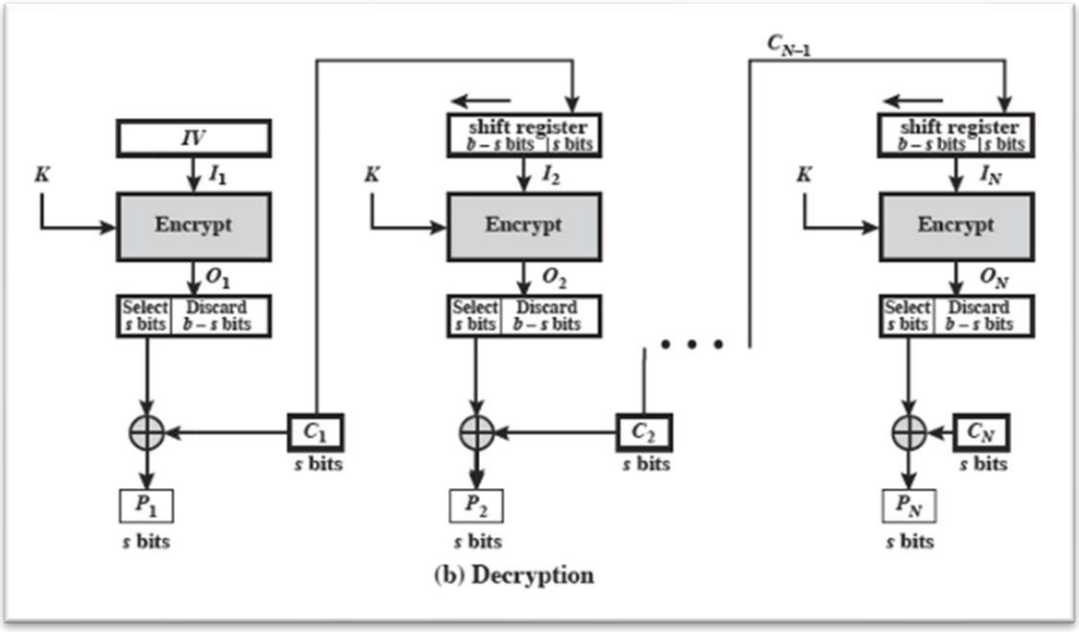




CIPHER FEEDBACK (CFB)

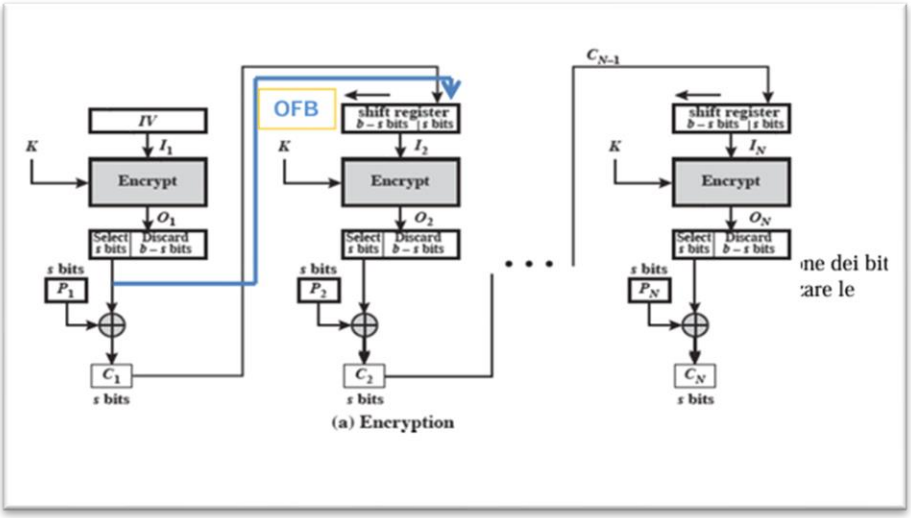
Permette di eseguire una cifratura a flussi; l'input della funzione è dato da un **registro a scorrimento di b bit (dimensione del blocco), inizializzato con un vettore di inizializzazione (IV) e la chiave K.** Successivamente, gli s bit (s = dimensione dei segmenti in cui è suddiviso il blocco, solitamente 8) più significativi del testo cifrato vanno in XOR con gli s bit del testo in chiaro e generano la prima unità di testo cifrato C1, che viene trasmessa. In seguito, il registro a scorrimento viene fatto scorrere a sinistra di s bit e nei bit meno significativi si inserisce la prima unità di testo cifrato C1. Il ciclo riparte.





OUTPUT FEEDBACK (OFB)

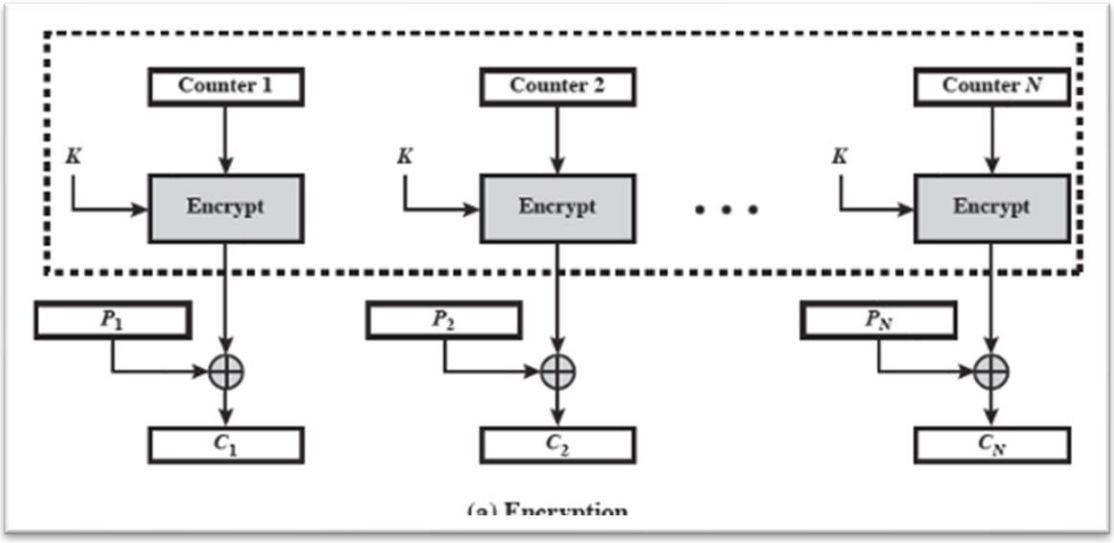
Molto simile a CFB, **ma nel registro a scorrimento vengono inviati gli s bit cifrati che vanno in XOR con gli s bit del testo in chiaro, anzi che con C1.**

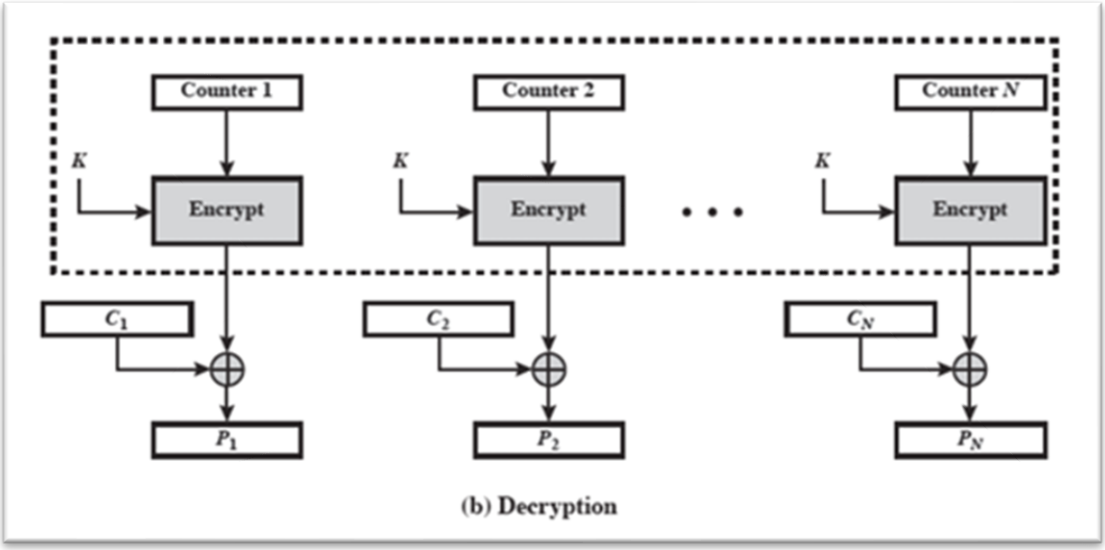


COUNTER (CTR)

Viene implementato un contatore della dimensione del blocco in chiaro; il valore del contatore deve essere diverso per ogni blocco che viene cifrato, solitamente stabilendo un valore di partenza ed incrementandolo nel blocco successivo. Tale valore viene cifrato e messo in XOR con il blocco del testo in chiaro.

Questa modalità, non creando legami tra i blocchi, permette l'elaborazione dei singoli in parallelo.





(b) Decryption

TIPOLOGIE DI CRITTOGRAFIA:

- **crittografia di canale**
- **crittografia end-to-end**

Sono entrambe utilizzate, a livelli diversi del protocollo OSI

IL PROBLEMA DI SCAMBIO DELLE CHIAVI:

Scambio diretto tramite altro canale, scambio di una nuova chiave tramite canale cifrato con quella vecchia, scambio tramite Key Distribuition Center, la terza parte può consegnare una chiave generata ad-hoc alle parti per cifrare la loro comunicazione.

SCAMBIO DI CHIAVI TRAMITE KDC, utilizzo del NONCE e delle identità degli interlocutori. Il KDC funge da "garante" dell'identità degli intelocutori.

CRITTOGRAFIA ASIMMETRICA

- **Chiave pubblica: viene distribuita**
- **Chiave privata: rimane nella esclusiva disponibilità del proprietario**

Sono chiavi complementari tra loro: se ne uso una per cifrare un messaggio, potrò decifrarlo solo ed esclusivamente con l'altra.

Se uso la chiave pubblica di qualcuno per cifrare un messaggio, in modo che lui possa decifrarlo solo con la chiave privata, garantisco la segretezza del messaggio ma non l'identità di chi lo trasmette.

Se invece cifro il messaggio con la mia chiave privata, permetto a tutti quello con il possesso della mia chiave pubblica di decifralo (nessuna segretezza), ma ne autentico in maniera inequivocabile la provenienza.

Per creare una cifratura efficace, sarà necessario cifrare prima un messaggio con la chiave pubblica di chi lo dovrà ricevere (segretezza) e poi con la propria chiave privata (autenticazione).

REQUISITI DELLA CRITTOGRAFIA A CHIAVE PUBBLICA:

- **deve essere computazionalmente facile generare la coppia di chiavi;**
- **deve essere computazionalmente facile cifrare il messaggio con le varie chiavi;**
- **deve essere impossibile, conoscendo una chiave, risalire all'altra;**
- **deve essere impossibile, conoscendo una chiave e il testo cifrato, risalire al testo in chiaro;**
- **Le chiavi devono poter essere applicate in qualsiasi ordine;**

ALGORITMO RSA (RIVER-SHAMIR-ADLEMAN, 1977, MIT)

GENERAZIONE DELLE CHIAVI

- **Ogni utente sceglie due numeri primi p e q (che tiene privati);**
- **Calcolo n = p*q;**
- **Calcolo ϕ(n) = (p-1)*(q-1);**
- **Selezionare un numero e tale che sia primo relativo di ϕ(n) e minore di ϕ(n), quindi MCD ϕ(n), e = 1; ϕ(n) ed e devono essere quindi primi tra loro;**
- **Determinare d tale che d*e=1 mod160 e d<160 (esistono algoritmi dedicati a questo calcolo);**
- **Rendere pubblica la chiave pubblica PU={e, n};**
- **Tenere segreta la chiave privata PR={d, n};**

Per ottenere il testo cifrato C partendo dal messaggio M:

C = M^e mod n

Per decifrare il messaggio M partendo dal cifrato C:

M = C^d mod n

È opportuno scegliere p e q sufficientemente grandi per ridurre le possibilità di attacco.

MODALITÀ DI DISTRIBUZIONE DELLE CHIAVI PUBBLICHE

- **Annuncio pubblico;**
- **Elenco pubblico (deve essere gestito da una autorità fidata che lo aggiorni);**
- **Autorità di distribuzione delle chiavi (è la CA a distribuire direttamente le chiavi sotto richiesta);**
- **Certificati a chiavi pubbliche (include chiave pubblica, identificatore del proprietario e firma di garanzia dell'autorità di certificazione);**

GENERALMENTE, LA CRITTOGRAFIA ASIMMETRICA È UTILIZZATA PER STABILIRE UN CANALE SICURO PER CONDIVIDERE CHIAVI DI CRITTOGRAFIA SIMMETRICA.

Distribuzione semplice di chiave segreta:

invio diretto della chiave da un interlocutore all'altro.

Distribuzione della chiave segreta con segretezza ed autenticazione:

Negli scambi iniziali tra due interlocutori, dove si utilizzano le rispettive chiavi pubbliche, viene incluso un NONCE (ad esempio, un timestamp) da entrambe le parti, in modo da verificare la segretezza del canale.

Per lo scambio sicuro di chiavi, è possibile utilizzare l'algoritmo di Diffie-Hellman; si basa su logaritmi discreti e radici primitive.

La radice primitiva di un numero primo **p** è un numero che genera potenze in modulo p comprese tra 1 e p-1 (se a è radice primitiva di p, allora a mod p, a^2 mod p... a^p-1 mod p sono distinti e compresi tra 1 e p-1)

Dato poi un qualsiasi intero **b** ed una radice primitiva **a** di un numero primo **p**, è possibile trovare un esponente univoco **i** tale che:

b=a^1 mod p dove **0<= i <=(p-1)**

L'esponente **i** prende il nome di **logaritmo discreto.**

Esempio:

abbiamo le quantità **q=353** e **a=3**, selezioniamo le chiavi private **Xa=97** e **Xb=233**. Calcoliamo le chiavi pubbliche:

Ya=3^97 mod 353 =40

Yb=3^233 mod 353=248

Possiamo poi calcolare da entrambe le parti la chiave segreta **K** eseguendo le operazioni:

K=248^97 mod 353=160

K=40^233 mod 353=160

L'algoritmo di Diffie-Hellman è suscettibile ad attacco MitM in quanto non autentica i partecipanti.

- **AUTENTICAZIONE DEI MESSAGGI: verifica che i messaggi ricevuti vengano effettivamente da una determinata sorgente e non siano stati modificati.**
- **FIRMA DIGITALE: una tecnica di autenticazione che certifica la non ripudiabilità del messaggio da parte del mittente.**

L'AUTENICATORE (il valore che autentica il messaggio) può essere generato principalmente tramite 3 modalità:

- **La crittografia dei messaggi (simmetrica ed asimmetrica)**
- **Il codice MAC (Message Authentication Code)**
- **La funzione Hash**

CODICE MAC: viene generato con una chiave privata condivisa, passata sul testo che genera una stringa di lunghezza fissa; è suscettibile a brute force, in quanto qualcuno che conosce il testo in chiaro e il codice MAC, può forzare utilizzando tutte le 2^k chiavi possibili.

FUNZIONI HASH: viene generato con una funzione f su un testo che genera una stringa di lunghezza fissa. Esso viene allegato al messaggio M e inviato, in modo che il destinatario possa ricalcolare l'hash e verificare l'integrità del messaggio stesso.

La funzione deve essere facile a livello computazionale, in modo da poter esser calcolata in modo veloce, ma impossibile da invertire, in modo che sia

impossibile, partendo dalla stringa generata, risalire al testo che l'ha generata.

La funzione di hash deve avere due proprietà fondamentali:

- **Proprietà di resistenza debole alle collisioni: dato un valore x, è impossibile trovare un valore y diverso da x tale che l'hash di x sia uguale all'hash di y.**
- **Proprietà dii resistenza forte alle collisioni: è impossibile trovare una coppia di valori tali che i loro hash siano uguali.**

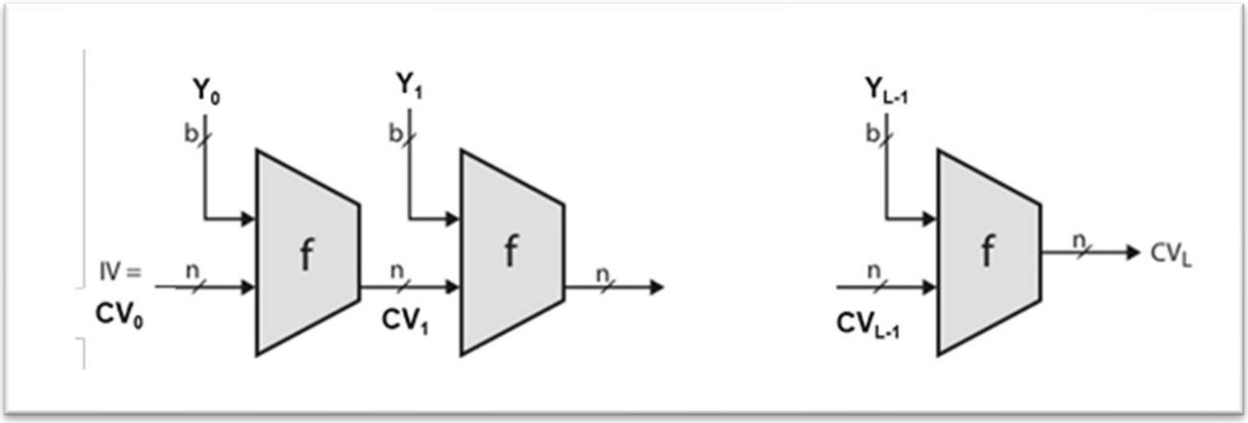
ATTACCO A COMPLEANNO

Vogliamo sostituire il messaggio originale con uno alterato e siamo in possesso del messaggio originale e conosciamo la funzione di hash utilizzata. Ipotizziamo di avere un messaggio cifrato con un hash a 64 bit; proviamo a generare un numero x di messaggi variati, dove x è uguale a 2^(64/2). Successivamente, volendo sostituire il messaggio, genero anche 2^32 versioni del messaggio fraudolento.

Confronto gli hash di tutte le varianti dei messaggi originali generati e di quelli fraudolenti generati, fino ad individuarne una coppia con lo stesso hash. Per il paradosso del compleanno, la probabilità di successo è superiore a ½. Se la corrispondenza non viene individuata, è possibile generare ulteriori varianti del testo e procedere con la comparazione.

L'ALGORITMO SHA1

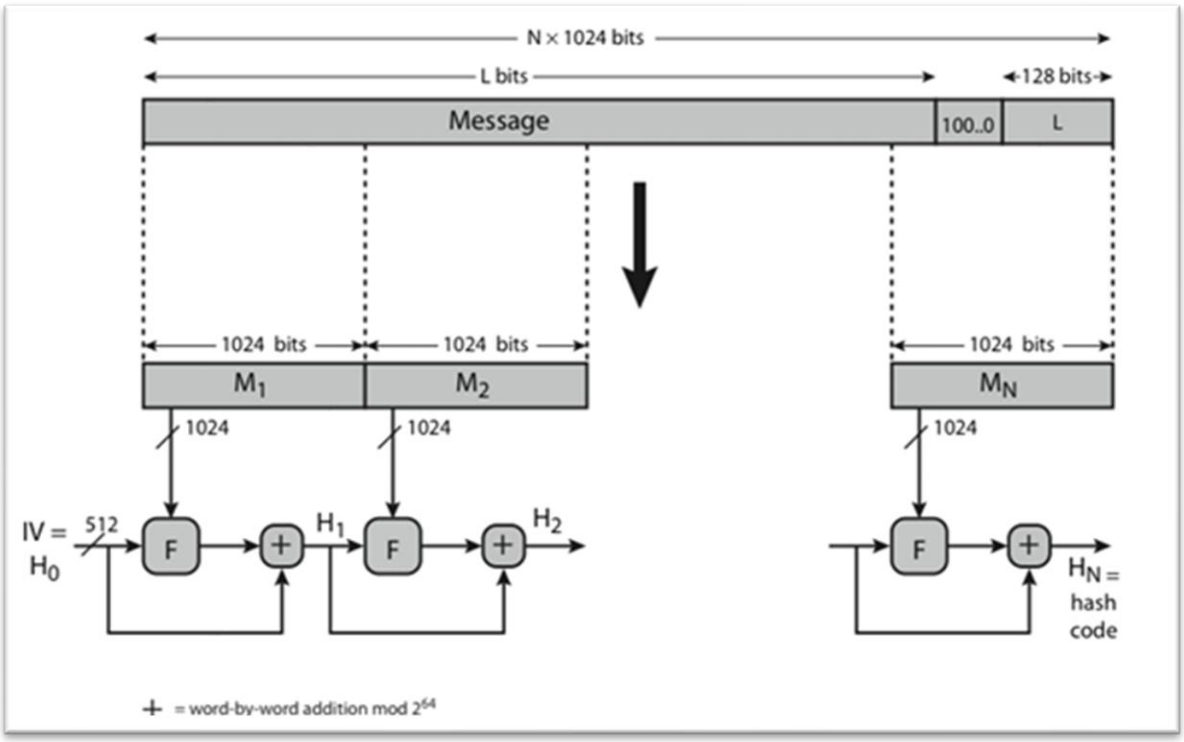
Il concetto che sta alla base di SHA 1 è la ripetizione di una **funzione di compressione f** che riceve in input **un blocco di n bit**, tratto dal passo precedente e chiamato **variabile di concatenamento.**



IV: valore iniziale
CV_i = valore di concatenamento
Y_i = blocco di input i-esimo
f = algoritmo di compressione
L = numero di blocchi
n = lunghezza codice hash
b = lunghezza blocchi di input

- **1995, SHA-1: digest a 160 bit**
- **Poi SHA2: SHA-256, SHA-384 e SHA-512**
- **2015, SHA3**

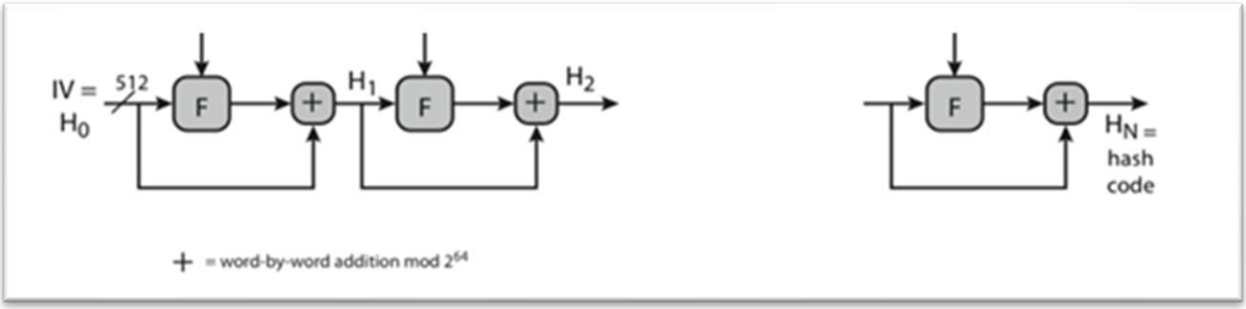
L'ALGORITMO SHA512



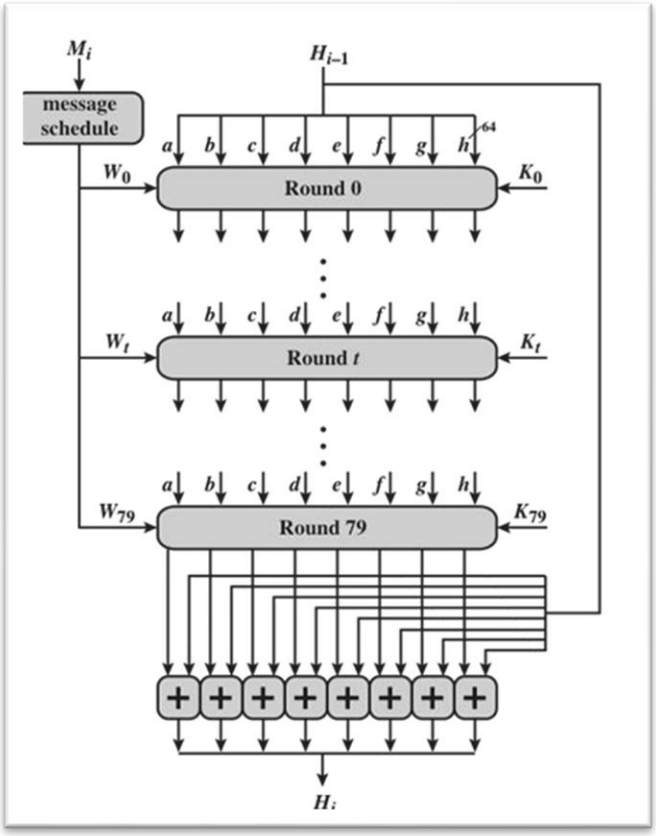
FASI:

1 **e 2) AGGIUNTA BIT DI RIEMPIMENTO:** si aggiungono da 1 a 1024 bit di rimepimento per ottenere N blocchi da 1024 bit.

3) INIZIALIZZAZIONE DEL BUFFER DI HASH: è un buffer di appoggio da 512 bit, quindi 8 da 64 bit; ognuno di essi è indicato da una lettera (A, B, C...) e contiene un valore che sono i 64 bit iniziali della parte frazionaria della radice quadrata dei primi 8 numeri



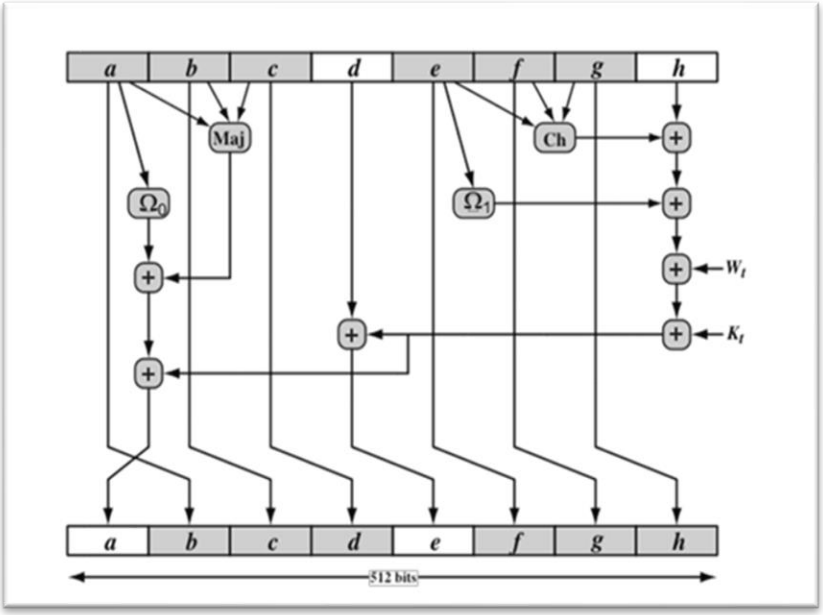
4) ELABORAZIONE DEL MESSAGGIO A BLOCCHI DA 1024 BIT: viene eseguita dal modulo f e consta di **80 fasi tramite la ripetizione della funzione di fase f**. Le 80 word vengono generate in base al blocco Mi, K0-K79 sono generate prendendo i primi 64 bit della parte frazionaria della radice cubica dei primi 80 numeri primi.



5) OUTPUT A 512 BIT: dopo l'elaborazione di tutti gli N blocchi da 1024 bit, l'output Hn costituisce il digest da 512 bit.

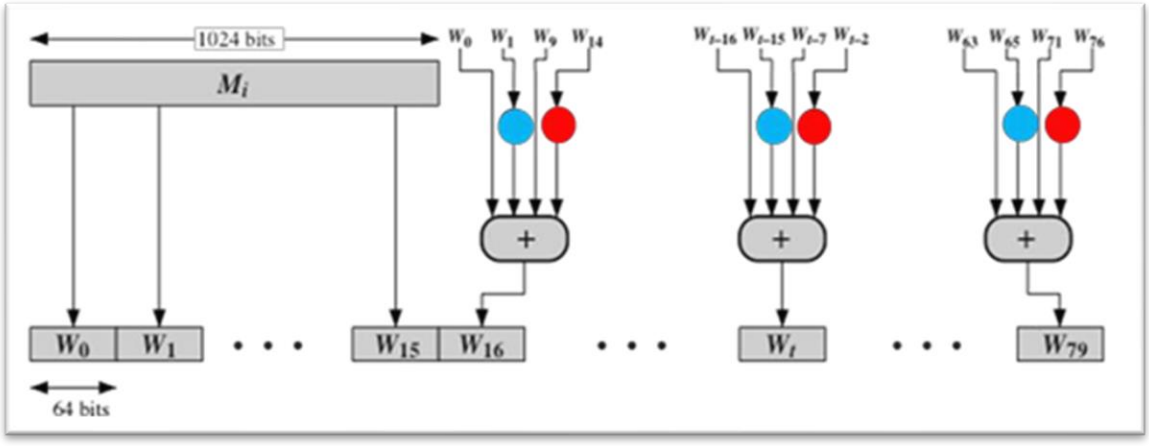
FUNZIONE DI FASE F

Prende in input il buffer di hash (8 registri da 64 bit), la word Wt (64 bit) e la costante di fase Kt (64 bit), e produce in out il buffer di hash aggiornato (512 bit), da usare nella fase successiva.



Le 80 word (W0-W79) vengono generate così:

- **W0-W15: sono una copia del blocco da 1024 bit.**
- **W16-W79: sono generate tramite la somma in modulo 2^64 delle word precedenti come indicato nello schema, dove i cerchi colorati sono rotazioni a destra e shift e sinistra con riempimenti di zeri a differenti passi e XOR logici.**

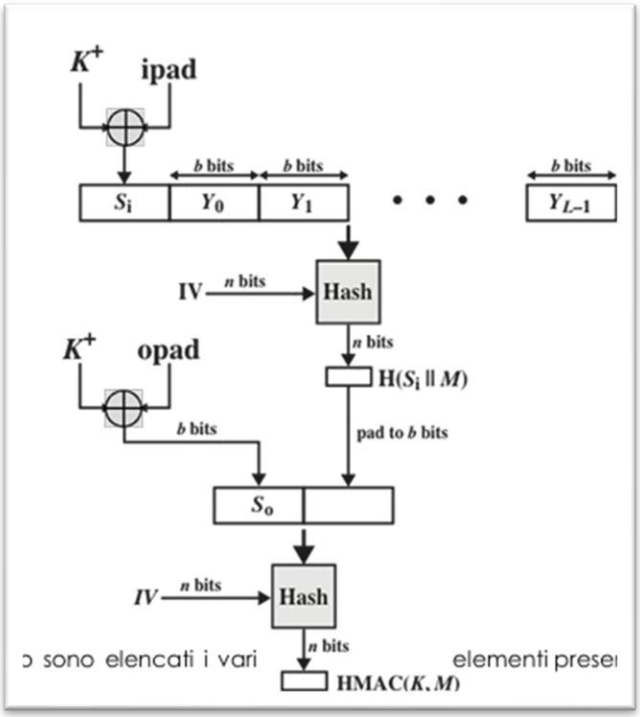


L'ALGORITMO HMAC

RFC2104 – **HMAC (keyed-Hash Message Authentication Code)** è un algoritmo che, al contrario del tradizionale MAC che deriva dalla cifratura simmetrica, implementa al suo interno funzioni di HASH. La sua sicurezza è indipendente dall'algoritmo di hash che implementa.

I passi sono i seguenti:

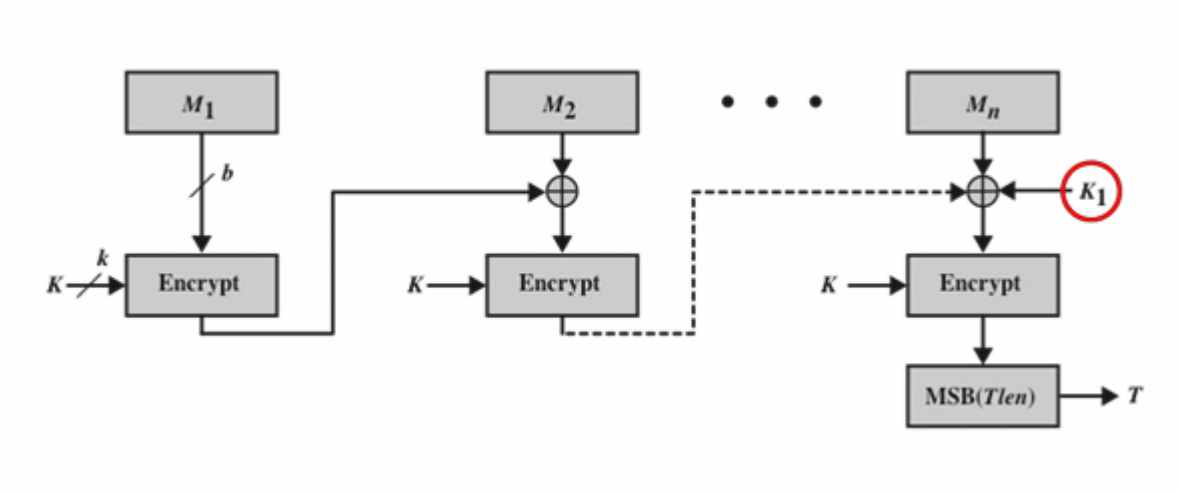
- **calcolo Si eseguendo uno XOR tra la chiave segreta K estesa alla lunghezza del blocco e il valore ipad (8 bit fissi ripetuti per tutta la lunghezza del blocco);**
- **Concateno il messaggio M con il valore Si calcolato sopra**
- **Calcolo l'hash e poi lo estendo da n bit a b bit;**
- **Eseguo uno XOR fra la chiave segreta estesa alla lunghezza del blocco e il valore opad (altri 8 bit fissi ripetuti per tutta la lunghezza del blocco), producendo il blocco di b bit S0;**
- **Concateno S0 con il codice hash esteso;**
- **Applico nuovamente la funzione di hash ed ottengo il codice HMAC;**

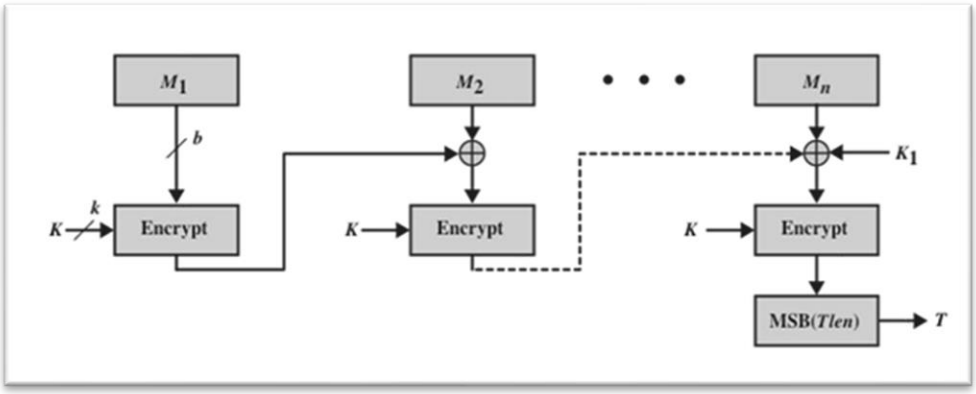


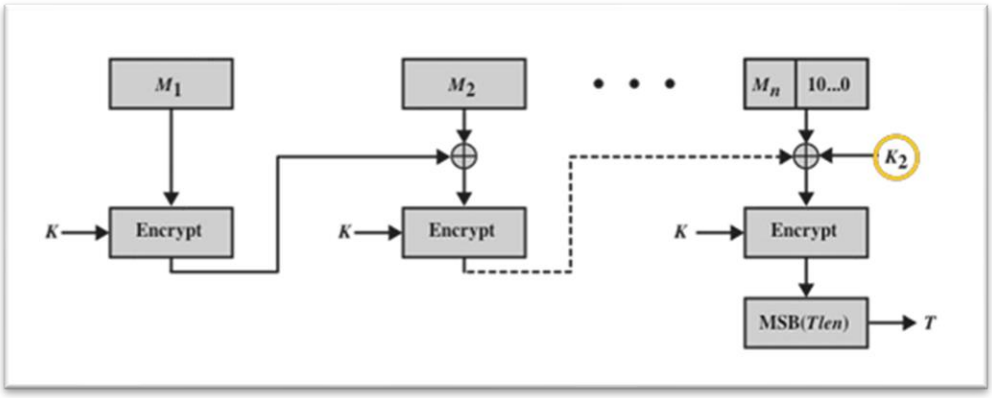
L'ALGORITMO CMAC

I codici MAC possono essere realizzati anche tramite algoritmi di crittografia a blocchi, come nel caso del **CMAC (Chiper Block Chaining Mac).**

Vengono derivate 2 chiavi, K1 e K2, da usare In caso il messaggio sia un multiplo intero della lunghezza b del blocco di cifratura (64 bit per 3DES, 128 bit per AES) o nel caso non losia; nel caso non lo sia, l'ultimo blocco viene esteso con 100...0 quanti ne servono per riempire il blocco.



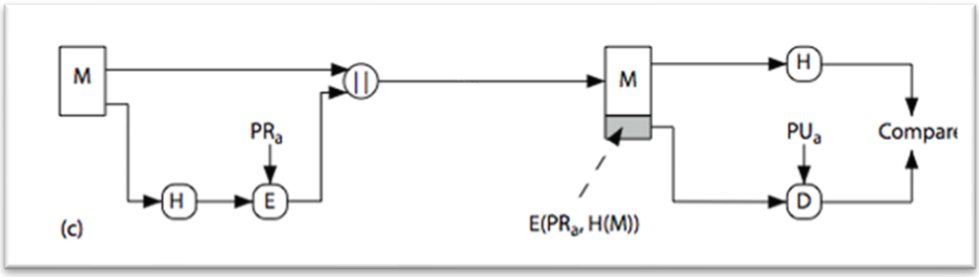




LE FIRME DIGITALI

Caratteristiche:

- **Devono certificare L'autore, la data e l'ora;**
- **Autenticare il contenuto nel momento in cui viene apposta la firma;**
- **Deve essere verificabile da terzi;**

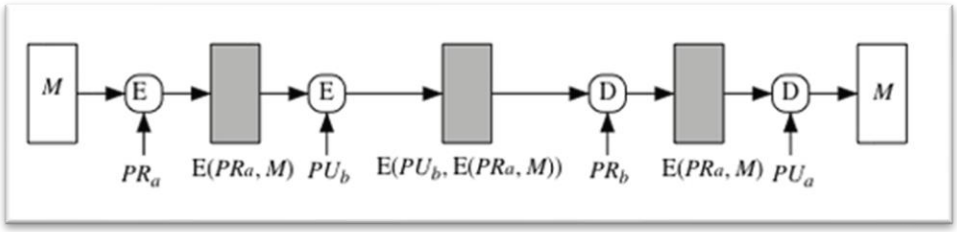


Requisiti:

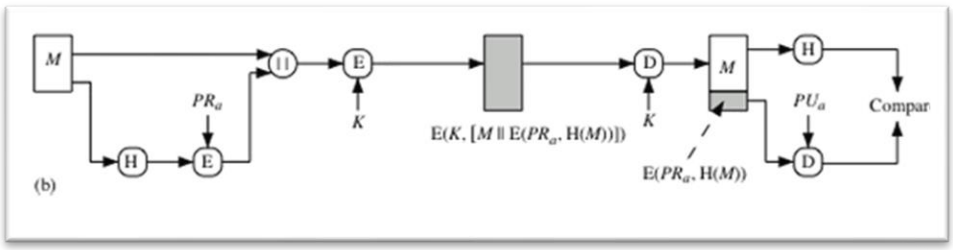
- **deve dipendere dal messaggio**
- **deve utilizzare informazioni specifiche del mittente;**
- **deve essere facile da produrre e da verificare;**
- **deve essere computazionalmente impossibile contraffarla;**
- **deve essere possibile conservarne una copia;**

Esistono due tipololgie fondamentali di firma digitale:

- **FIRMA DIRETTA: mittente e destinatario si scambiano direttamente la firma;**



Con crittografia asimmetrica



Con crittografia simmetrica

- **FIRMA ARBITRATA: mittente e destinatario si affidano ad un soggetto terzo fidato (l'arbitro) che verifichi la validità della firma stessa e dei messaggi inviati;**

In questo caso, implementando una soluzione a crittografia simmetrica, si utilizzeranno 3 chiavi segrete distinte: una tra X e l'arbitro A, una tra Y e l'arbitro A e una tra X e Y.

In questo caso, ad esempio:

X->A : 3 quantità concatenate:

- IDx: identificativo di X;
- E(Kxy, M): messaggio cifrato con la chiave XY;
- E(Kxa[IDx || E(Kxy, M)]): firma del mittente A cifrata con la chiave tra lui e l'arbitro;

L'arbitro A decifra la firma, verifica l'identificativo di X e la corrispondenza del messaggio cifrato (non può però leggerlo in quanto non possiede Kxy)

A->Y:

- IDx || E(Kxy, M)
- E(Kya[IDx || E(Kxy, M) || T]): si è aggiunto il Timestamp alla fine

A questo punto Y, conoscendo Kay, può decifrare il messaggio inviato dall'arbitro e, una volta ricavato il IDx, decifrare il contenuto di E(Kxy, M) e conservare la firma del mittente X, contenente il timestamp, per eventuali dispute.

AUTENTICAZIONE IN AMBIENTI DISTRIBUITI

KERBEROS -> sviluppato dal MIT, basato su DES ->V4 e poi V5

Soddisfa i seguenti requisiti:

- **SICUREZZA (basato su crittografia simmetrica)**
- **AFFIDABILITÀ (utilizza una architettura a server distribuiti e ridondati)**
- **TRASPARENZA (l'utente deve solo inserire la password)**
- **SCALABILITÀ (supporta un gran numero di client e server)**

Kerberos V4 implementa 2 tipi di server di autenticazione:

- **AS, Authentication Server:** serve ad un client per autenticare una sessione verso un server;
- **TGS, Ticket-granting server:** serve a proteggere le password e ad evitare che l'utente debba reintrodurre la password per ogni servizio all'interno della stessa sessione;

I PASSAGGI:

FASE A

1) **Il client C invia un messaggio al server AS richiedendo l'accesso al TGS;** invia il suo ID, l'ID del server TGS a cui vuole accedere ed un timestamp:

Ida || IDtgs || TS1

2) **Il server AS risponde con il seguente messaggio cifrato con la chiave del client Kc, derivata dalla password del client stesso:**

E(Kc [Kc tgs || IDtgs || TS2 || Lifetime2 || Ticket tgs])

Il messaggio contiene la chiave Kc tgs, condivisa tra client C e server Tgs, l'id del server stesso, un Timestamp TS2, un valore di durata e il ticket Tgs. Il ticket Tgs contiene:

E(Ktgs[Kc tgs || IDc || ADc || IDtgs || TS2 || Lifetime2])

Esso è cifrato con la chiave Ktgs (nota anche al server AS), quindi solo il server Tgs può aprire il ticket; tale ticket contiene la chiave tra il client C e il server Tgs, che viene quindi ricevuta in modalità sicura e permette quindi la comunicazione diretta con il client C.

FASE B

1) **Il client C contatta il server TGS;** invia:

IDv || TicketTgs || Autenticatore c

L'autenticatore è così costituito:

Autenticatore c = E(Kc tgs[IDc || ADc || TS3])

A questo punto, il server Tgs risponde al client C inviando il seguente messaggio cifrato con la chiave di sessione Ktgs:

E(Kc tgs[Kc v || IDv || TS4 || Ticket v])

Il messaggio contiene la chiave di sessione Kc v, condivisa tra il client C e il server V, il timestamp e un ticket per il server V cifrato con la chiave Kv del server V:

Ticket v = E(Kv [Kcv || IDc || ADc || IDv || TS4 || Lifetime])

FASE C
A questo punto, il client C possiede un ticket V con cui presentarsi al server V inviando:

Ticketv || Autenticatore

Il server V tramite la chiave K decifra il ticket e recupera la chiave di sessione Kcv; con tale chiave, il server può decifrare l'autenticatore e verificare l'identità del client C con quanto ricevuto nel Ticket.

Alla fine del protocollo, il client C e il server V possiedono una chiave segreta condivisa.

DdoS

- **SYN FLOOD: multipli SYN con indirizzo di ritorno falso -> il server apre migliaia di sessioni con riposte con non arrivano -> Risorse esaurite;**
- **ICMP ECHO: pacchetti ICMP con host uguale a quello della macchina da attaccare; il server fa ECHO REPLY su se stesso;**

2 TIPI DI DDOS:

DdoS diretto (usa gli host infetti) e **DdoS riflesso** (uso di macchine bot che a loro volta instradano verso macchine non infette, usate come riflettori).

MULTIMEDIA FORENSICS

- **BLIND:** non si dispone del contenuto in forma originale, ma solo di quello da analizzare;
- **PASSIVO:** si può soltanto osservare il contenuto senza avervi accesso prima per inserire informazioni utili in seguito (i.e. watermarking) per l'analisi;

- **IDENTIFICAZIONE DELLA SORGENTE E DELL'INTEGRITÀ**
- **ANALISI DELL'AUTENTICITÀ**

3 Tipologie di alterazione di una immagine digitale:

- **Fotoritocco (modifica di una immagine tramite strumenti o filtri)**
- **Fotomontaggio (unione di più immagini diverse)**
- **Copy-move (clonazione di una parte di immagine al suo interno)**

IDENTIFICAZIONE DELLA SORGENTE

- **Attraverso metadati**
- **Analizzando le caratteristiche dell'immagine**
  - **Sistema ottico**
  - **Sensore CCD/CMOS**
  - **Elaborazione e salvataggio del file**

SISTEMA OTTICO: lenti e Color Filter Array (CFA) specifici e riconoscibili, ad esempio per specifica distorsione ottica dovuta alle lenti. Il CFA, tramite algoritmi di DEMOSACKING specifici, converte i colori reali in spazio RGB;

SENSORE CCD/CMOS: cattura la luce -> genera rumore specifico PRNU(Photo Response Non-Uniformity), riconoscibile e caratteristico per ogni sensore. Serve un campione di fotografie un minimo consistente dal punto di vista numerico, con bassa varianza e ad alta luminosità, per stimare il PRNU di un device.

Il passaggio su alcuni social network lascia alcune “tracce” sulle immagini (ricompressioni specifiche o filtri dedicati, watermark)

ANALISI ALTERAZIONI BASATE SU ELEMENTI FISICI

- **Inconsistenza delle luci**
- **Inconsistenza delle ombre**
- **Inconsistenza delle traiettorie (nei video): legate alle traiettorie degli oggetti in base alla prospettiva dell’immagine, quindi sfasameto dei piani.**

ANALISI ALTERAZIONI BASATE SUI DESCRITTORI: anomalie basate sulla descrizione del contenuto

Usate per nascondere o duplicare qualcosa; analisi legata a più fasi:

- **Descrizione**, ad esempio dei punti salienti, tramite specifici descrittori quali SIFT, SURF, ORB e RISK;
- **Matching** tra i descrittori;
- **Clustering e localizzazione** delle eventuali zone duplicate;

ANALISI ALTERAZIONI BASATE SUI FORMATI

Legati ad effetti o artefatti tipici di alcuni tipi di formati, quali blocking e blurring per il JPEG, oppure ricompressioni multiple di aree della stessa immagine.

APPLICAZIONI

- Cyber-bullismo
- Adversarial Machine Learning
- DeepFake

LA BLOCKCHAIN

Struttura decentralizzata basata su strumenti crittografici, protocolli condivisi, DLT (ditribuited Ledger Technology) e peer-to-peer, utile a vari utilizzi. È formata da blocchi concatenati tra loro; ogni blocco è collegato con il precedente ed il successivo.

Ogni blocco può contenere diversi tipi di informazioni; nel caso dei Bitcoin, contiene informazioni sulle transazione (mittente (identificato tramite chiave pubblica), destinatario (chiave pubblica), ammontare inviato).

Ogni blocco quindi contiene:

- **i dati delle transazioni;**
- **l'hash del blocco precedente;**
- **l'hash del blocco attuale (SHA 256, ne garantisce l'integrità);**
- **l'hash del blocco successivo;**

PROOF-OF-WORK

Garantisce la sicurezza della blockchain; è un processo computazionalmente complesso ma di semplice verifica. **Consiste nel trovare un numero casuale (nonce) tale che l'hash dei dati delle transazioni, l'hash del blocco precedente e il nonce stesso abbiano un numero definito di 0 iniziali.**

Se una transazione della catena viene "alterata", il suo Pow viene alterato e quindi sarà facile capire da che punto in avanti di essa la stessa non è più valida.

IL LEDGER

È un libro mastro, utilizzato nelle reti P2P; ogni nodo ne contiene una copia e su di esso sono memorizzate tutte le informazioni delle transazioni. Garantisce in parte la sicurezza, in quanto una versione modificata dello stesso e distribuita da un nodo, prima di diventare quella ufficiale, andrebbe accettata almeno dal 50% dei nodi presenti.

IL MINING

Nel caso dei bitcoin, è il processo che produce nuova moneta; risolvere il proof-of-work attuale premia i nodi della rete che partecipano (miner) con il rilascio di una certa quantità di nuova moneta (in base al loro contributo; essi possono infatti aggregarsi in mining pool dedicate). Gli altri miner, una volta risolto il Pow attuale, lo verificano tramite il nonce e aggiornano la blockchain.

Schema di una transazione

1) **A richiede una transazione a B**
2) **La transazione viene inviata a tutti i nodi della blockchain**
3) **I nodi validano la transazione**

4) **Le transazioni validate sono aggregate in “mempool”, in attesa di essere inserite in un nuovo blocco**
5) **Il blocco viene validato tramite la Pow**
6) **Il primo miner che risolve il Pow attuale invia la soluzione agli altri miner per la verifica ed il blocco viene aggiunto alla blockchain**