Checksum

Gruppo di bit di controllo associati al messaggio. Di solito, è posizionato alla fine del messaggio e calcolato come **complemento a uno del risultato della somma dei dati**. In questo modo gli errori possono essere rilevati sommando l'intera parola contenente sia i bit dati sia il checksum.

CRC

È una tecnica di error detection basata sull'**aritmetica polinomiale** in base 2. Una stringa di m bit viene interpreta come un polinomio di grado $m-1$ che ha come coefficienti i bit della stringa.

La sorgente e la destinazione si accordano su un **polinomio generatore** $G(x)$, che deve avere come primo ed ultimo bit 1. Il frame da controllare $M(x)$ deve essere di ordine maggiore di $G(x)$. L'idea è aggiungere una specie di checksum alla fine del frame in modo che il polinomio rappresentato dal frame, checksum compreso, sia divisibile per $G(x)$.

Il calcolo del checksum avviene in questo modo:

1. sia r il grado di $G(x)$. Si aggiungono r zeri alla fine del frame, in modo da ottenere una sequenza $x^r M(x)$ composta da $m+r$ bit;
2. si divide tale sequenza per $G(x)$ tramite la divisione in modulo 2;
3. si sottrae il resto ottenuto dalla divisione a $x^r M(x)$, sempre in modulo 2;
4. la sequenza risultante sarà il frame con il checksum pronto per la trasmissione.

La destinazione riceve il polinomio e prova a dividerlo per $G(x)$, ossia eseguo uno shift a sinistra di r bit. Se c'è **resto allora si è verificato un errore**. Dato che un errore di trasmissione può essere visto come sommare un polinomio E(x) al frame, allora è possibile fare error detection per tutti gli errori E(x) che non sono divisibili per G(x).

Esempio:

$$M(x) = 10011101$$

$$G(x) = x^4 + x + 1 = 10011$$

$$F(x) = x^4 M(x) + R(x) = 100111010000 + 1111 = 100111011111$$

2.3 Protocolli per il controllo di flusso

2.3.1 Stop-and-wait

È piuttosto elementare e si usa in canali simplex o half duplex. Il principio è il seguente: il mittente trasmette un frame e **aspetta** che chi è dall'altro capo

del canale invii una **conferma di ricezione**, detta **ACK**, *Acknowledgement*. Un chiaro svantaggio di questo protocollo è l'**elevato tempo di attesa** tra le trasmissioni dei frame: Stop-and-wait **non funziona bene in presenza di roundtrip delay elevato**.

Si possono verificare due errori:

Il frame non arriva a destinazione Il mittente si trova bloccato nell'**aspettare l'ACK, che non arriverà mai**, rendendo necessaria la presenza di un **timeout** dopo il quale il frame venga inviato nuovamente.

L'ACK non arriva al mittente Il destinatario riceve il frame ma al mittente non arriva la conferma: dopo il timeout descritto precedentemente i dati vengono ritrasmessi. Tuttavia, in questo modo **il destinatario riceve due volte lo stesso frame**. La soluzione consiste nel dotare i frame di un campo che lo etichetti con un numero; basta anche un solo bit, per indicare il precedente dal successivo.

2.3.2 I protocolli sliding window

Ogni partecipante alla conversazione deve tener sotto controllo **due finestre**: quella dei frame in **entrata** e quella dei frame in **uscita**. Ogni frame in uscita contiene un numero di sequenza e il destinatario deve tener traccia di questi per la ricezione, mentre il mittente per l'invio.

Quando il mittente **invia un frame, resta nella finestra** finché non viene ricevuto il corrispondente ACK. Dopodiché la finestra verrà aggiornata. Quando il destinatario **riceve il frame**, controlla che il numero sia uguale a quello che si aspettava e ne invia l'ACK. Se quest'ultimo contiene il numero che la sorgente si aspettava, prosegue nella trasmissione inviando un nuovo frame. Altrimenti, invia nuovamente quello segnato nel buffer.

Si possono **inviare più frame contemporaneamente** prima di entrare in attesa (pipelining). Il destinatario aggiorna la finestra non appena riceva il frame e invia l'ACK. In caso di pipelining, i protocolli **go back n** e **selective repeat** permettono di gestire i problemi dovuti a questo tipo di trasmissione.

Go back n

E' il protocollo a finestra scorrevole più semplice. La finestra di chi riceve ha ampiezza 1 mentre quella di chi trasmette ha ampiezza N. Vengono inviati fino a N pacchetti alla volta con un'etichetta che indica lo slot della finestra, anche se il ricevente li gestisce comunque uno alla volta.

La destinazione, una volta accortasi che un frame è corrotto, **scarta a prescindere tutti i frame successivi** (per numero di sequenza) già ricevuti. Per i frame scartati non invia ACK, ma aspetta che scadano i timeout nel mittente, il quale provvederà a trasmetterli nuovamente.

Go back n **funziona bene quando il roundtrip delay è elevato e il canale è affidabile**. Si noti che il mittente deve essere in grado di gestire N timer per rispedire i pacchetti e avere un buffer capiente in cui tenerne una copia da ritrasmettere.

Selective repeat

Evoluzione del Go back n, nella quale anche il ricevente ha un buffer per contenere più frame contemporaneamente. Il destinatario conserva tutti i frame validi e li salva in un buffer. La sorgente continua ritrasmettere i frame per i quali, prima della scadenza del proprio timeout, non ha ricevuto l'ACK. Se la destinazione riceve un frame corrotto, invia un **NACK** (*Not ACK*) per sollecitare la ritrasmissione prima della scadenza del timeout da parte del mittente.

Si può inoltre inviare un NAK per indicare al mittente che un frame potrebbe essere stato perso e diminuire ulteriormente lo spreco di tempo (Il timer del NAK deve essere minore del timer di rinvio). Questo protocollo **sfrutta il più possibile il canale**, però richiede maggiori risorse dato che è necessario gestire un timer e una parte di buffer per ogni slot aperto della finestra. Un **problema** di questo protocollo (delle sliding windows in generale) è che l'**apertura massima della finestra deve essere al più uguale alla metà delle etichette disponibili per evitare la perdita di sincronizzazione**.

Piggybacking

È una tecnica che consiste nello **sfruttare un messaggio del destinatario al mittente come *passaggio* per il messaggio di conferma** ACK. Il campo ACK è posto nell'header del frame. Chiaramente, il piggybacking dell'ACK si usa solo se vi è un messaggio del destinatario al mittente nell'immediata possibilità d'inviare la conferma, altrimenti si invia l'ACK separatamente. Di conseguenza questa tecnica funzione bene quando la comunicazione tra le due parti è bilanciata.

2.4 PPP

Il protocollo PPP, *Point to Point Protocol*, viene utilizzato per gestire la configurazione della rilevazione d'errore di una linea, supportare molteplici protocolli, permettere l'autenticazione e molto altro. Provvisto di numerose opzioni, il protocollo PPP ha le seguenti tre caratteristiche principali:

- un metodo di framing che permette di delimitare in modo non ambiguo la fine di un frame e l'inizio del successivo. Il formato del frame permette di gestire anche la rilevazione di errori;

- un protocollo per gestire la connessione, il test della linea, negoziare le opzioni di collegamento e gestire la disconnessione in modo pulito quando la linea non serve più. Questo protocollo è chiamato **LCP**, *Link Control Protocol*;

- una modalità per negoziare le opzioni relative al livello di rete, in modo indipendente dall'implementazione di tale livello che verrà usata per la comunicazione. Il metodo scelto avrà un diverso NCP (*Network Control Protocol*), un protocollo di controllo della rete, per ogni livello di rete supportato.

Questo protocollo viene largamente utilizzato nelle connessioni DSL (per il collegamento utente-provider) e GPRS.

Nome	Numero di bytes	Descrizione
Flag	1	indica l'inizio o la fine del frame
Address	1	indirizzo broadcast
Control	1	byte di controllo
Protocol	2	indica il protocollo del campo data
Data	variabile (da 0 a 1500)	campo di dati
FCS	2 (o 4)	somma di correzione

Figura 2.1: Frame PPP

Il campo più importante di un frame PPP è il campo **Protocol** che specifica il tipo di protocollo PPP in uso: se il primo bit è a 1 si sta usando LCP mentre se è a 0 NCP. Ci sono 11 tipi di frame LCP, 4 di configurazione 2 di terminazione, 2 di rifiuto, 2 di echo e 1 di test.

Configurazione: configure-request, configure-Ack, configure Nak, e configure-reject, vengono usati per stabilire e configurare la connessione, si può scegliere la lunghezza del campo dati, che livello di error-detection usare e se trasmettere o meno i campi Address e Control.

Terminazione: terminate-request e terminate-ack.

Rifiuto: code-reject (richiesta sconosciuta) e protocol-reject (protocollo richiesto non supportato).

Echo: echo-request e echo-reply, per il test della qualità della rete.

Test: discard-request

I frame PPP poi possono essere incapsulati in frame Ethernet (PPPoE) o ATM (PPPoA).

4.3.6 Principali differenze tra IPv4 e IPv6

- indirizzi più lunghi: 16 byte di IPv6 contro i 4 di IPv4. Questo si traduce nel **supporto a miliardi di host anche con un'allocazione di indirizzi altamente inefficiente**;
- **semplificazione** dell'intestazione. Ad esempio, si è deciso di togliere l'error detection per rendere la trasmissione più veloce. Dopotutto, era ridondante e rallentava l'instradamento dato che il decremento del campo TTL per ogni nodo attraversato rendeva necessario il ricalcolo del checksum;
- opzioni del protocollo: in IPv4 erano limitate a 40 byte mentre con IPv6, grazie al campo *Next Header*, sono potenzialmente illimitate. Lascia quindi **spazio ad evoluzioni future**;
- IPv6 fornisce un grado di sicurezza maggiore;
- IPv6 permette a un host di spostarsi senza cambiare il suo indirizzo;
- Il vecchio protocollo può coesistere con quello nuovo.

4.4 Altri protocolli

4.4.1 DHCP

DHCP, *Dynamic Host Configuration Protocol*, è un protocollo che permette l'assegnazione manuale o automatica dell'indirizzo IP. L'idea di base è semplice: esiste un server *speciale* che assegna gli indirizzi alle macchine che lo richiedono. Questo server può trovarsi sulla stessa LAN del richiedente.

Quando una macchina vuole ottenere un indirizzo IP, manda in broadcast un particolare pacchetto chiamato *DHCP DISCOVER*. Il gestore di rete centralizzato risponde alla macchina inviando un pacchetto contenente un indirizzo. Dato che nella stessa rete ci possono essere più gestori DHCP, l'host richiedente deve confermare al gestore che ha accettato l'IP proposto, il quale dovrà chiudere l'accordo con un ACK.

Essendo la rete dinamica, le informazioni dopo un po' di tempo devo essere rinnovate mediante un meccanismo di fading chiamato **leasing** secondo il quale sia il gestore sia la macchina interessata sanno quando scade l'informazione, in modo da poterla rinnovare prima che scada.

4.4.2 ARP

ARP, *Address Resolution Protocol*, è un protocollo di servizio appartenente alla suite del protocollo Internet IPv4, il cui compito è fornire la *mappatura* tra indirizzo IP e MAC di un terminale in una rete locale. Il suo analogo in IPv6 è NDP, *Neighbor Discovery Protocol*.

Per inviare un pacchetto IP, è necessario incapsularlo in un frame di livello data link, che dovrà avere come indirizzo di destinazione il MAC address dell'host a cui lo si vuole inviare. ARP viene utilizzato per ottenere questo indirizzo. Se il pacchetto deve essere inviato ad un'altra sottorete, l'indirizzo MAC da richiedere sarà quello del gateway o del router.

Ogni macchina connessa alla rete conserva in memoria una tabella, detta *ARP Cache*, con le corrispondenze IP-MAC già precedentemente richieste, in modo da evitare d'interrogare continuamente la rete per inviare ogni pacchetto. Le informazioni contenute in questa cache vengono cancellate dopo un certo periodo di tempo, tipicamente 5 minuti.

Quando un host deve inviare un pacchetto ad un IP non presente in tabella, manda in broadcast un messaggio detto *ARP REQUEST*, contenente il proprio indirizzo MAC e l'indirizzo IP della macchina di cui si vuole conoscere l'indirizzo IP. Tutti gli host della rete ricevono la richiesta e, in ciascuno di essi, il protocollo ARP verifica se viene richiesto il proprio indirizzo MAC confrontando il proprio IP con quello ricevuto. L'host che verifica positivamente questa corrispondenza provvede ad inviare in **unicast** una risposta, *ARP Reply*, contenente il proprio MAC.

Per ottimizzare il protocollo, una macchina, quando si collega alla rete, manda in broadcast la richiesta della propria associazione IP-MAC, così tutte le altre macchine la possono memorizzare. Inoltre, se qualcuno risponde significa che si è verificato un errore nell'assegnazione degli indirizzi IP.

Si noti che l'ARP Request ad un nodo aggiorna completamente la tabella ARP presente nella cache, senza rispetto per le voci già esistenti. Particolare molto importante perché causa di diversi attacchi chiamati *ARP Spoofing*, che verranno approfonditi nella sezione dedicata alla sicurezza delle reti.

4.4.3 NAT

Il NAT, *Network Address Translation*, costituisce una soluzione adottata per risolvere il problema dell'esaurimento degli indirizzi IP, **simulando una rete in un unico indirizzo**. Ad esempio, un'azienda può avere un solo indirizzo IP per il traffico Internet. Internamente, ogni host riceve un indirizzo univoco per instradare i dati nella rete aziendale. Quando il traffico esce da questa rete avviene una traduzione dell'indirizzo a quello di Internet, effettuato dal dispositivo NAT.

Il problema sta nel decidere a chi inviare internamente la risposta inviata dal Web, che come indirizzo ha quello generico aziendale. Siccome, la quasi totalità dei pacchetti IP trasporta carichi utili TCP e UDP, si è pensato di sfruttare queste informazioni per implementare l'inoltro. TCP e UDP, come si vedrà in seguito, utilizzano e includono nei propri header una porta sorgente e una di destinazione del pacchetto.

Ogni volta che un pacchetto diretto verso l'esterno raggiunge l'apparto NAT, oltre a sostituire l'indirizzo, viene sostituito anche il campo *Source*

port, con un indice che punta alla tabella di traduzione NAT. Al contrario, quando viene ricevuta la risposta, si usa il campo *Source port* ricevuto viene usato per ottenere l'indirizzo e la porta del mittente originale all'interno della rete NAT.

Sebbene questo schema in un certo senso risolva il problema dell'esaurimento degli indirizzi IP, molti della comunità lo considerano un vero e proprio abominio.

Viola il modello gerarchico di IP Non è più vero che ogni indirizzo IP identifica a livello mondiale una singola macchina.

Rompe il modello di connettività end-to-end Poiché l'associazione viene effettuata dai pacchetti in uscita, i pacchetti di entrata non possono essere accettati se non dopo quelli in uscita. Esistono comunque alcune tecniche per arginare questo problema.

Trasforma Internet in una rete orientata alla connessione Questo perché l'apparato NAT deve conservare le informazioni relative a ogni connessione che lo attraversa. **Se l'apparato NAT si blocca, tutte le sue connessioni TCP vanno perse**. Questo non succede in assenza di NAT.

Viola la regola principale della stratificazione dei protocolli Di norma, il livello k non deve essere costretto a formulare alcuna ipotesi su ciò che il livello $k + 1$ ha inserito nel payload. NAT distrugge questa indipendenza perché si basa sui livelli superiori TCP e UDP. Se in futuro TCP venisse aggiornato a TCPv2, con uno schema d'intestazione diverso, NAT non funzionerebbe più.

I processi su Internet non sono obbligati a usare TCP e UDP Se due utenti si accordassero per usare un nuovo protocollo di trasporto, il meccanismo di NAT non funzionerebbe più.

Alcune applicazioni usano più connessioni TCP o porte UDP NAT non sa gestire queste situazioni, a meno che non vengano prese speciali precauzioni.

Nonostante ciò, NAT è molto usato nella pratica, specialmente per reti domestiche o in piccole aziende, in quanto è l'unica tecnica che riesca a gestire il problema della mancanza di indirizzi IP. Per questa ragione è inverosimile che sparisca, anche qualora IPv6 fosse ampiamente adottato.

Capitolo 7

Sicurezza

7.1 Introduzione

7.1.1 Glossario

Cifrario Trasformazione del testo originale **carattere per carattere**.

Codice Rimpiazzo di una **parola con un'altra**

I messaggi da cifrare sono detti **testo in chiaro**. L'output è il **testo cifrato**. L'arte di forzare i cifrari, chiamata **criptoanalisi** e l'arte di inventarli, **crittografia**, sono note sotto il nome collettivo di **crittologia**. **Decriptare** è l'attività di decifrazione da parte di un criptoanalista (intruso), mentre **decifrare** è l'operazione legittima di lettura di un messaggio cifrato.

7.1.2 Principio di Kerckhoff

Il principio di Kerckhoff afferma che **tutti gli algoritmi devono essere pubblici, solo le chiavi devono essere tenute segrete**. Tenere invece segreto l'algoritmo è una forma di sicurezza detta **per occultamento**. Il segreto sta quindi in un buon algoritmo con chiavi lunghe per aumentare il fattore lavoro.

7.1.3 Principi crittografici fondamentali

Ridondanza Tutti i messaggi cifrati devono contenere informazioni ridondanti, ossia non necessarie alla comprensione del messaggio.

Attualità È necessario avere la possibilità di verificare che ogni messaggi ricevuto sia attuale.

7.2 Chiave Condivisa

7.2.1 Cifrari a sostituzione

In un cifrario a sostituzione, **ogni lettera o gruppo di lettere viene rimpiazzato da un'altra lettera o gruppo di lettere** per mascherare il messaggio. Una semplice generalizzazione consiste nello **spostare l'alfabeto del testo cifrato di k lettere**. In questo caso k diventa la chiave del metodo generale, che sta nell'avere un alfabeto spostato circolarmente. Il miglioramento successivo consiste nel far sì che ognuno dei simboli del testo in chiaro corrisponda ad altri simboli. Il sistema generale per la sostituzione simbolo a simbolo viene chiamato **sostituzione monoalfabetica**, dove la chiave è la stringa di lettere che corrisponde all'interno alfabeto.

Per decriptare i testi cifrati con questo metodo, si utilizza un **approccio statistico** chiamato *frequency analysis*, che si basa sul fatto che le singole lettere, i digrammi (coppie di lettere) e i trigrammi (terne di lettere) in ogni lingua hanno una certa frequenza ben nota. Un altro approccio è quello di provare con una parola che dato il contesto ha una buona probabilità di essere nel testo e da lì ricavare le varie lettere.

7.2.2 Cifrari a trasposizione

I cifrari a sostituzione conservano l'ordine dei simboli del testo in chiaro, limitandosi a mascherare la loro apparenza. I **cifrari a trasposizione, invece, riordinano le lettere** senza mascherarle. Si utilizza una matrice in cui il testo in chiaro viene scritto orizzontalmente, per righe, fino a riempire la matrice, eventualmente usando alcuni caratteri per occupare celle rimaste vuote. Il **testo cifrato viene trasmesso per colonna**, secondo un ordine stabilito in base ad una chiave usata: può essere anche una stringa con l'ordine determinato tramite lessicografia.

Per poter attaccare un cifrario a trasposizione si vede dapprima se le **frequenze delle lettere nel testo cifrato corrispondono alle frequenze statistiche** nella lingua presa in esame. Da ciò si può capire se il cifrario è a trasposizione. Il passo successivo è **scoprire di quante colonne e formata la matrice e il loro ordine**.

7.3 Algoritmi a chiave simmetrica

7.3.1 DES

Sono dei sistemi di crittografia che si basano su product cipher (combinazioni di P-Box e S-Box [1]). DES sfrutta 19 passaggi che lavorano su **blocchi da**

[1] I P-Box sono delle funzioni che prendono in input un blocco di bit e lo permutano. Gli S-Box prendono in input un blocco di bit e li alterano. Tutte le modifiche dipendono da una chiave

64 bit e con una chiave da 56 bit:

- il primo e l'ultimo sono trasposizioni, una il contrario dell'altra;
- il penultimo consiste nello scambiare i 32 bit di destra con quelli di sinistra;
- i passaggi intermedi sono funzionalmente uguali ma parametrizzati con diverse funzioni della chiave.

7.3.2 Triplo DES

Triplo DES si basa sull'**utilizzo a cascata di DES**. Per criptare il testo:

1. lo si codifica con una chiave K_1;
2. il risultato viene decodificato con una chiave K_2;
3. il tutto viene poi ricodificato ancora con K_1,

Questo procedimento è stato adottato per **retrocompatibilità con DES**: infatti, se si usano due chiavi uguali, la prima e la seconda operazione si annullano ed il terzo passaggio codifica il testo secondo l'algoritmo classico.

7.3.3 AES

AES, *Advanced Encryption Standard* è l'algoritmo, nato nel 1997 a seguito di un concorso pubblico, che ha sostituito DES come standard ufficiale del governo statunitense. Si tratta di un **cipher block** con blocchi e chiavi da 128 e 256 bit.

7.3.4 Cipher Modes

Electronic Code Book

Il testo viene diviso in blocchi che vengono poi **cifrati uno dopo l'altro** usando la stessa chiave. L'ultimo pezzo di testo in chiaro, se necessario, viene riempito per fargli raggiungere la lunghezza dei blocchi precedenti.Ogni blocco può dunque essere visto come pagina di un libro.

Crudelia può tuttavia memorizzarsi tutte le pagine e inviarle a Bob con un **ordine casuale rischiando di modificare lo stato** di Bob. Questo problema viene aggirato dai Cipher Modes che aggiungo ad un blocco criptato della dipendenze al blocco precedente.

Chaining Mode

Per evitare gli attacchi che possono capitare con ECB, si **collegano tutti i blocchi cifrati in diversi modi**, in modo che un eventuale operazione di scambio o rimpiazzo renda i dati senza significato a partire dal punto in cui è stata operata la sostituzione.

Ogni blocco di testo in chiaro è messo in XOR con il precedente blocco cifrato, prima di eseguire la cifratura vera e propria. Per il primo blocco, lo XOR viene calcolato con un blocco di dati casuali detto vettore di inizializzazione (IV). Un chiaro vantaggio di questo metodo sta nel fatto che **non produce lo stesso testo cifrato a partire da blocchi di testo in chiaro uguali**. Lo **svantaggio** principale consiste nel dover aspettare che un **intero blocco di testo cifrato arrivi a destinazione prima che la decifrazione possa cominciare**.

Feedback Mode

Usato in certi casi in cui è richiesta una decifrazione *al volo*, prima che tutto il blocco sia ricevuto, come nel caso dei terminali interattivi. Si usa quindi una cifratura byte a byte mediante un registro di shift ausiliario. Inizialmente, se non sono presenti dei byte già cifrati, il registro viene riempito con il vettore di inizializzazione. Per ottenere un byte criptato, si deve:

1. criptare il contenuto del registro;
2. prendere il byte più a sinistra e metterlo in XOR con il byte da criptare;
3. inserire il byte criptato nel registro.

Allo stesso modo per decriptare un byte, prima si cripta il registro e poi si fa lo XOR per ottenere il byte in chiaro. Infine si inserisce nel registro il byte criptato, cioè quello ricevuto.

Stream Mode

Anziché cifrare il testo, **si cifra il vettore di inizializzazione con una chiave crittografica e poi si usa il risultato per cifrare il testo mediante XOR**. Il risultato della cifratura di IV viene ulteriormente cifrato (keystream) per produrre il secondo blocco in uscita, quindi si prosegue analogamente per il terzo e via dicendo.

Counter Mode

Simile allo stream mode, anziché cifrare continuamente il vettore di inizializzazione **si cifra il vettore di inizializzazione più un numero sequenziale**. Se il vettore di inizializzazione non cambia tra una trasmissione e l'altra c'è il rischio di un attacco del tipo keystream reuse (vale anche per lo stream mode).

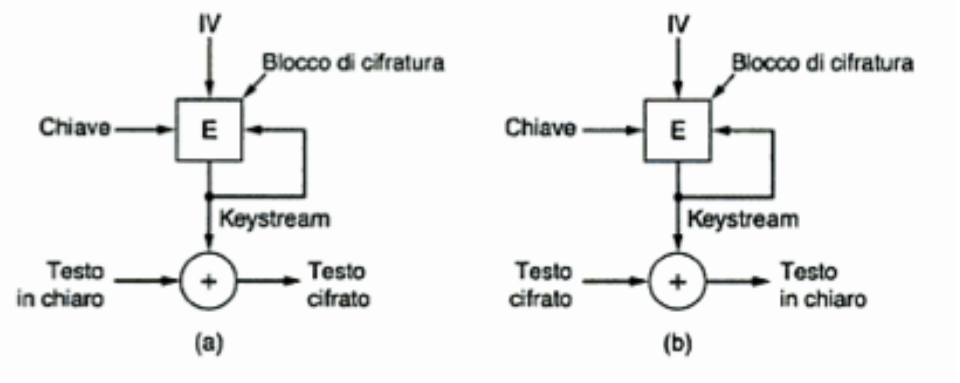

Figura 7.1: Stream Mode

7.4 Algoritmi a chiave pubblica

7.4.1 Diffie-Hellman

1. Alice e Bob si scambiano due numeri molto grandi, P e G in chiaro.
2. Alice e Bob scelgono un numero random ciascuno, A e B.
3. Alice invia a Bob $G^A \mod P$ (chiave pubblica di Alice), lo stesso vale per Bob ($G^B \mod P$)
4. Entrambi si calcolano $G^{A^B} \mod P$ che è la loro chiave segreta.
5. Alice e Bob possono ora comunicare in modo criptato usando la chiave segreta

Con i dati che può ottenere Crudelia, il calcolo della chiave segreta è un **problema intrattabile risolvibile solo a forza bruta**.

7.4.2 RSA

Alice vuole comunicare con Bob in modo segreto.

1. Bob sceglie due numeri P e Q molto grandi e primi.
2. Bob calcola:

   prod $= P \cdot Q$.
   ino $= (P-1) \cdot (Q-1)$.
   dec un numero coprimo di *ino*
   enc tale che $enc \cdot dec$ è congruo a $1 \mod ino$

3. Bob invia ad Alice la sua chiave pubblica, cioè la coppia *(end,prod)*

4. Alice divide il messaggio in blocchi di dimensione minore di *prod* e lo cripta con $C = P^{enc} \mod prod$

5. Bob decripta il messaggio in $P = C^{dec} \mod prod$

Crudelia per decifrare il testo ha bisogno della chiave privata di Bob e per calcolarla deve trovare P e Q. Il calcolo di P e Q è un **problema intrattabile e risolvibile solo mediante forza bruta**.

7.5 Firme digitali

7.5.1 Hash crittografico

Una funzione, per essere considerata un hash crittografico, deve avere le seguenti proprietà:

1. Dato P è facile calcolare $H(P)$;

2. Dato $H(P)$ è praticamente impossibile trovare P;

3. Dato P è praticamente impossibile trovare Q tale che $H(P) = H(Q)$;

4. Una piccola variazione di P fa variare completamente $H(P)$

Alcuni esempi di hash crittografici sono MD5, SHA-1 e SHA-2.

7.5.2 Hash Message Auth Code

E' un sistema di autenticazione a chiave condivisa. Assieme al messaggio viene inviato l'hash del messaggio e della chiave. Tramite HMAC è possibile garantire sia l'integrità che l'autenticità del messaggio.

7.5.3 Preimage Attack

È un attacco in cui si cerca di **trovare un messaggio che ha uno specifico valore hash**. Se l'hash è di n bit servono circa 2^n tentativi a forza bruta.

7.5.4 Birthday Attack

L'attacco del compleanno consiste, data una funzione f, nel trovare due numeri x_1 e x_2 tali che $f(x_1) = f(x_2)$. Tale coppia di valori (x_1, x_2) è chiamata **collisione**. A causa del paradosso del compleanno, quest'attacco può risultare efficiente: applicato agli hash, per trovare due messaggi che hanno lo stesso hash con una probabilità del 50% bastano $2^{\frac{n}{2}}$ tentativi.

7.6 Gestione delle chiavi pubbliche

7.6.1 Certificati

Vengono usati per garantire l'identità del proprietario e possono essere rilasciati solo da un Autorità di Certificazione. Uno standard per i certificati è X.509, il quale prevede vari campi tra cui:

- il numero di serie del certificato
- l'autorità che l'ha emesso
- il proprietario
- la sua chiave pubblica
- la firma della CA.

Per sapere che una CA è veramente una CA e non Crudelia che si finge tale, è necessario che ogni computer abbia al suo interno i certificati che certificano l'autenticità delle CA.

7.7 Sicurezza delle comunicazioni

7.7.1 IPsec

E' una variante del protocollo IP che aggiunge un sistema di autenticazione a chiave condivisa mediante Diffie-Hellman. IPsec può essere implementato in due modalità:

Transport I dati relativi all'autenticazione vengono inseriti all'inizio del campo dati del pacchetto IP.

Tunnel Viene creato un nuovo pacchetto IP contenente al suo interno i dati per l'autenticazione e il vecchio pacchetto IP.

Ci sono due protocolli supportati da IPsec ed entrambi possono funzionare in modalità transport o tunnel: **AH** e **ESP**.

Entrambi gli header sono simili e contengono i seguenti campi.

NextHeader Indica il tipo di protocollo usato per trasmettere i dati.

SPI Identifica, in combinazione con l'indirizzo IP, la Security Association.

Sequence Number Relativo alla connessione, serve per evitare attacchi di tipo replay.

La differenza tra i due protocolli sta nella parte autenticata (con HMAC):

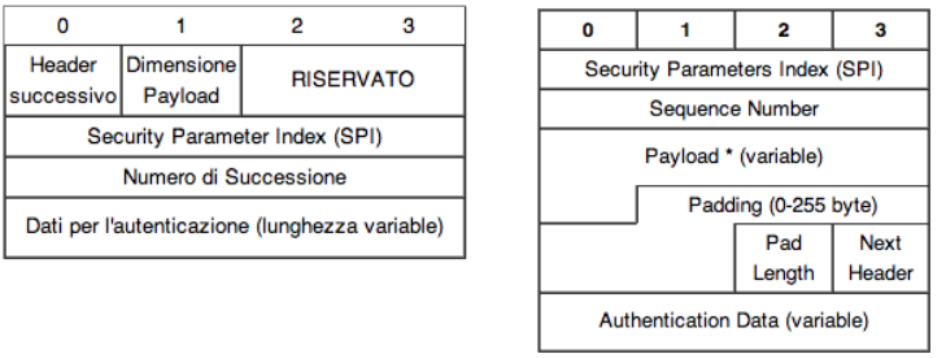

Figura 7.2: A sinistra l'header del protocollo AH mentre a destra quello ESP

- con AH viene autenticato l'intero pacchetto (i campi variabili vengono calcolati come se fossero 0);
- con ESP invece vengono autenticati solamente l'header ESP e il contenuto del pacchetto. Con ESP i dati trasmessi nel campo payload vengono criptati.

7.8 Protocolli di autenticazione

7.8.1 3-Way Handshake

E' un protocollo di autenticazione a chiave condivisa, usato anche nel sistema GSM. Alice e Bob vogliono comunicare tra loro, per essere sicuri di parlare l'uno con l'altro:

1. Alice sfida Bob mandandogli un numero casuale da criptare con la chiave condivisa;
2. Bob risponde alla sfida e ne lancia una analoga ad Alice;
3. Alice risponde alla sfida.

Dato che la chiave condivisa è conosciuta solo da Alice e Bob solo loro possono rispondere alla sfida.

Reflection Attack e soluzione

Crudelia sfrutta Bob per farsi passare la risposta:

1. Crudelia sfida Bob;
2. Bob risponde alla sfida e a sua volta sfida Crudelia con un numero X;

3. Crudelia, che non può rispondere alla sfida, sfida nuovamente Bob usando però il numero X;

4. Bob risponde alla sfida fornendo a Crudelia la risposta alla prima sfida;

5. Crudelia risponde alla prima sfida di Bob.

In questo modo Bob crede di comunicare con Alice quando in realtà sta comunicando con Crudelia. Nella realtà viene quindi usata una versione autenticata:

1. Alice sfida Bob mandandogli R_a

2. Bob risponde con R_b,$HMAC(R_a, R_b, A, B, K_{ab})$

3. Alice risponde con $HMAC(R_a, R_b, K_{ab})$

Quando Alice riceve la risposta di Bob può calcolarsi l'HMAC e controllare se coincidono. Crudelia non potrà mai generare lo stesso HMAC dato che non conosce la chiave condivisa.

7.9 Minacce alla sicurezza

7.9.1 Attacco Man in the Middle

È un attacco nel quale **l'attaccante è in grado di leggere, inserire o modificare a piacere messaggi tra due parti, senza che nessuna delle due sia in grado di sapere che il collegamento è stato compromesso**. L'attaccante deve essere in grado di osservare e intercettare il transito dei messaggi tra le due vittime.

Supponiamo che Alice voglia comunicare con Bob, e che Giacomo voglia spiare la conversazione, e se possibile consegnare a Bob dei falsi messaggi.

1. Per iniziare, Alice deve chiedere a Bob la sua chiave pubblica. Se Bob invia la sua chiave pubblica ad Alice, ma Giacomo è in grado di intercettarla, può iniziare un attacco Man in the middle.

2. Giacomo può semplicemente inviare ad Alice una chiave pubblica della quale possiede la corrispondente chiave privata.

3. Alice poi, credendo che questa sia la chiave pubblica di Bob, cifra i suoi messaggi con la chiave di Giacomo ed invia i suoi messaggi cifrati a Bob.

4. Giacomo quindi li intercetta, li decifra, ne tiene una copia per sé, e li cifra nuovamente, dopo averli alterati se lo desidera, usando la chiave pubblica che Bob aveva originariamente inviato ad Alice.

5. Quando Bob riceverà il messaggio cifrato, crederà che questo provenga direttamente da Alice.

Questo tipo di attacco può essere fatto anche all'interno di una rete locale mediante ARP Poisoning.

7.9.2 DoS

È una tipologia di attacco volta a far collassare la macchina vittima. Un attacco DoS, *Denial of Service*, può essere fatto a livello network (*smurf*) con le richieste ECHO di ICMP, oppure a livello Transport (*SYN attack*) inviando in continuazione richieste di connessione TCP senza confermarle.

Un attacco DoS può essere fatto anche in versione *distribuita* (DDoS) mediante l'utilizzo di più macchine zombie che attaccano contemporaneamente la vittima. Esiste inoltre una versione *reversed* effettuata mediante *IP spoofing*, nella quale l'attaccante invia un elevato numero di pacchetti ECHO modificati che hanno come mittente l'IP della vittima.

7.10 Sicurezza nelle reti WireLess

7.10.1 FHSS

FHSS, *Frequency Hopping Spread Spectrum*, consiste nell'avere la **frequenza di trasmissione che varia ad intervalli regolari**, rendendo più difficile intercettare la trasmissione.

7.10.2 Bluetooth

La sicurezza con Bluetooth si basa su FHSS e un sistema di crittografia a chiave simmetrica con blocchi da 128 bit.

7.10.3 802.11

WEP

In WEP, *Wired Equivalent Privacy*, le trasmissioni vengono cifrate mediante una chiave simmetrica da 40-104-232 bit (WEP-40/104/232) e un cifrario a blocchi RC4 secondo la modalità stream chyper.

Il **punto debole del sistema WEP è il vettore di inizializzazione che, essendo composto da soli 24 bit, rende probabile il riutilizzo di una sua configurazione** (attacco di tipo *keystream reuse*). Infatti, già nel 2001 si riusciva a bucare una rete WEP in circa 15 minuti. Da segnalare, inoltre, l'attacco ai magazzini Marshall che ha permesso ai malintenzionati di rubare 45.7 milioni di carte di credito (evento registrato come record del mondo).

WPA

WPA sfrutta TKIP, un protocollo di cifratura basato su RC4 con una chiave (*passphrase*) di 128 bit (da 8 a 32 caratteri ASCII). Introduce tuttavia un nuovo problema di origine sociale: **gli utenti tendono ad usare chiavi corte e simili a parole**, creando una vulnerabilità agli attacchi di tipo *dictionary*.

Nel 2008 viene scoperta una falla che ha portato allo standard **WPA2**, in cui viene usato CCMP (*Counter Mode with Cipher Block Chaining Message Authentication*); chiavi e blocchi restano però di 128 bit.