Crittografia

Indice

Capitolo 1

Introduzione

La crittografia fa parte, insieme alla crittoanalisi, della crittologia.
La **crittografia** è la scienza che studia tecniche e metodologie per **cifrare un testo in chiaro**, al fine di produrre un **testo cifrato comprensibile solo ad un ricevente legittimo**, il quale possiede l'informazione sufficiente (detta chiave) per decifrarlo, recuperando il testo in chiaro.
La **crittoanalisi** studia come decrittare un testo cifrato per ottenere il testo in chiaro: il verbo decrittare indica l'azione compiuta da un'entità che non possiede la chiave per recuperare, in modo legittimo, il testo in chiaro. In letteratura, suddetta entità viene indicata con il termine ascoltatore, oppure avversario o anche nemico. Lo scopo della crittografia è di produrre un messaggio cifrato m in modo che nessun avversario sia in grado di decrittarne il contenuto.
La **stegonografia** (dal greco *scrivere nascosto*), indica l'insieme delle tecniche che permettono a due (o più) entità in modo tale da occultare, agli occhi di un ascoltatore, non tanto il contenuto (come nel caso della crittografia), ma l'estistenza stessa di una comunicazione. In altre parole, è l'arte di nascondere un messaggio all'interno di un altro *messaggio contenitore*. Alcuni esempi sono:

- disposizione dei caratteri
- inchiostro invisibile
- contrassegno dei caratteri
- nascondere messaggi all'interno di bit di file multimediali
- ...

Proprietà di sicurezza di un messaggio

1. **Confidenzialità**
2. **Autenticazione**
3. **Non ripudio**

4. **Integrità**

5. **Anonimia** (canali in cui le entità comunicanti devono poter nascondere la loro identità)

Capitolo 2

Crittografia classica

2.1 Cifrari simmetrici

Con crittografia simmetrica, o crittografia a chiave privata, si intende una tecnica di cifratura che consiste nel cifrare un testo in chiaro dove la chiave di crittazione è la stessa chiave di decrittazione. Non è previsto uno scambio di chiavi, dunque le due parti devono esserne già in possesso.
Possiamo dividere i cifrari simmetrici in due categorie:

- **Cifrari a flusso:** trasformano pochi bit alla volta (un singolo bit, o un byte); si rivelano adatti quando si deve proteggere singoli dati generati uno dopo l'altro. Al flusso di dati in input in input corrisponde un *flusso di chiavi*
- **Cifrari a blocchi:** trasformano blocchi di più bit (64, 128, o più); si rivelano adatti quando si devono proteggere intere strutture dati

Si può fare una distinzione tra due tecniche per la cifratura del testo in chiaro:

- **Sostituzione:** i valori in chiaro vengono sostituiti con altri simboli
- **Trasposizione:** vengono scambiate le posizioni dei valori in chiaro

Queste tecniche possono essere combinate tra loro, a patto che siano reversibili.

2.1.1 Cifrari di shift

Sono una generalizzazione del *cifrario di Cesare*, senza fissare a 3 lo spostamento delle lettere.
Con questo approccio, lo schema di cifratura deve avere uno spazio delle chiavi non vulnerabile ad un attacco di *forza bruta*; tuttavia, questa è una condizione necessaria ma non sufficiente!
Anziché semplicemente shiftare le lettere, si potrebbe effettuare una permutazione casuale delle lettere dell'alfabeto; la chiave, diventa dunque una stringa di 26 lettere.

```
Plain:  abcdefghijklmnopqrstuvwxyz
Cipher: DKVQFIBJWPESCXHTMYAUOLRGZN   ← chiave
```

Con un alfabeto di 26 lettere, ci sono 26! possibili chiavi.
Potremmo aver pensato di aver ottenuto un adeguato livello di sicurezza, ma le caratteristiche del linguaggio naturale rendono possibili attacchi alternativi, come ad esempio la frequenza dei caratteri (che non cambia tra testo in chiaro e cifrato).
Alcune contromisure sono:

- usare più simboli per cifrare i caratteri più frequenti, in modo da abbassare la frequenza
- aggiungere simboli meno frequenti nel testo in chiaro, in modo da non compromettere il significato
- usare parole in codice oltre ai simboli dell'alfabeto

2.1.2 Cifrario affine

Sono un caso particolare di cifrario a sostituzione monoalfabetico. Per trovare la sostituzione si usa un'espressione, detta **affine**:

$$c_i = E(p_i) = (k_1 p_i + k_2) mod 26$$

$\rightarrow$ la chiave è quindi data da **due costanti**.
La *decrittazione* avviene invece secondo la formula:

$$p_i = D(c_i) = (c_i - k_2) \cdot k_1^{-1}$$

con k_1^{-1} inteso come l'inverso modulo 26 di k_1, ovvero quel numero x che soddisfa l'equazione:

$$(k_1 \cdot x) mod 26 = 1$$

Affinché questo sia possibile è necessario che k_1 e 26 siano primi tra loro.

2.1.3 Cifrario Alberti

Fu il primo meccanismo che usò più alfabeti cifranti che si sostituiscono durante la cifratura. Utilizza un apparecchio composto da due dischi concentrici, rotanti in maniera indipendente, contenenti un alfabeto ordinato per il testo in chiaro, ed uno disordinato per il testo cifrato.

Si sceglie una lettera come chiave, la si fa corrispondere alla A del disco esterno, e per ogni lettera cifrata si muove in senso orario il disco interno.

2.1.4 Cifrario Porta

È un cifrario che si occupa di cifrare i **digrammi**, ovvero coppie di lettere. Ad ogni digramma viene assegnato un numero; la chiave è data da una permutazione arbitraria di numeri del cifrato e lettere su righe e colonne.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
B	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
C	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77
D	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103
E	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129
F	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155
G	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181
H	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
I	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233
J	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259
K	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285
L	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311
M	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337
N	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363
O	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389
P	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415
Q	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441
R	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467
S	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493
T	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519
U	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545
V	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571
W	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597
X	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623
Y	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649
Z	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675

2.1.5 Cifrario di Playfair

È una tecnica di cifratura simmetrica manuale basata su un **cifrario monoalfabetico a due lettere** (vengono cifrate due lettere alla volta). L'analisi delle frequenze può ancora essere intrapresa, ma invece che 26 monografi esistono 600 possibili digrafi! È dunque molto più difficile.
Il cifrario Playfair si basa sull'uso di una matrice 5×5 contenente una parola chiave; la tabella viene costruita inserendo le lettere della parola chiave senza lettere duplicate, per poi riempire gli spazi rimanenti con le lettere non utilizzate dell'alfabeto (in ordine).

Essendo disponibili 25 spazi, una lettera viene esclusa (di norma la Q o la W, oppure facendo collassare la I e la J nello stesso spazio). La chiave può essere scritta nelle celle della tabella seguendo una *logica arbitraria*: la chiave è dunque **l'ordine di riempimento** della parola chiave nella tabella, oltre che la parola stessa.
La cifratura avviene separando le lettere del testo in chiaro in digrafi ed applicando a ciasuno di essi quattro regole:

1. se entrambe le lettere sono uguali, si aggiunge una X (o una lettera poco comune) tra di loro e poi si ripete il procedimento
2. se le lettere appaiono nella stessa riga, vengono codificate con quelle alla propria destra
3. se le lettere appaiono nella stessa colonna, vengono codificate con quelle immediatamente sotto
4. altrimenti, si identifica un rettangolo che ha le due lettere come vertici opposti; le lettere vengono codificate con quella sulla stessa riga in corrispondenza dell'altra colonna

Per decifrare, si usano regole inverse a queste.

2.1.6 Cifrario di Hill

Un cifrario a **sostituzione polialfabetica** basato sull'algebra lineare.

- In *primis*, viene fatto corrispondere ad ogni lettera un numero ($A = 0, B = 1, \dots$)
- Viene fissato un blocco di n lettere, considerate come uno spazio vettoriale di dimensione n
- Il vettore ottenuto viene moltiplicato per una matrice M di dimensione $n \times n$ in modulo 26
- Segue che m lettere in chiaro vengono sostituite con altrettante lettere cifrate secondo m equazioni lineari

La chiave è data dalla matrice M; deve essere casuale a patto che sia invertibile in $\mathbb{Z}_2 6$ affinché la decrittazione sia possibile.
Ad esempio, il messaggio `TUO` verrebbe cifrato con la chiave `YYPRZWIZJ` come:

$$\begin{pmatrix} 24 & 24 & 15 \\ 17 & 25 & 22 \\ 08 & 25 & 09 \end{pmatrix} \times \begin{pmatrix} 19 \\ 20 \\ 14 \end{pmatrix} = \begin{pmatrix} 1146 \\ 1131 \\ 778 \end{pmatrix} (mod 26) \equiv \begin{pmatrix} 02 \\ 13 \\ 24 \end{pmatrix}$$

2.1.7 Cifrario di Vigenère

È un il più semplice tra i cifrari **polialfabetici**, è considerabile come una **generalizzazione del cifrario di Cesare**: invece di spostare sempre dello stesso numero di posti la lettera da cifrare, essa viene spostata di un numero di posti variabile, ma ripetuto, determinato attraverso una **parola chiave**.
Rispetto ai cifrari monoalfabetici, si ha il *vantaggio* di avere n alfabeti cifranti (n lunghezza della chiave). La *debolezza* consiste che si tratta di n cifrari di Cesare, motivo per cui è possibile, a partire dalle ripetizioni, stimare la lunghezza della chiave per poi analizzare le frequenze in ognuno degli alfabeti cifranti.

2.1.8 Cifrario di Vernam

È un cifrario di Vigenère a cui viene imposto che la chiave abbia la stessa lunghezza del messaggio da cifrare.
Questo metodo è potenzialmente ***unconditionally secure*** (ossia sicuro indipendentemente dal tempo e risorse), ma ha il problema di necessitare di una chiave casuale (già di per sé difficile siccome le chiavi sono pseudo-casuali) ed eventualmente di grandi dimensioni.

2.2 Probabilità

Possiamo definire un cifrario come:

- Gen l'algoritmo di generazione dlele chiavi, con K lo spazio delle chiavi
- M spazio dei messaggi in chiaro
- C spazio dei messaggi cifrati, dove $Enc_k(m) = c$ è la funzione di cifratura, mentre $Dec_k(c) = m$ è la funzione di decifratura

Consideriamo ora i messaggi M e i testi cifrati C; possiamo definire:

- $P(M)$ la probabilità di osservare il messaggio M per l'attaccante
- $P(C)$ la probabilità di osservare il testo cifrato C per l'attaccante

Possiamo ora definire uno schema di cifratura composto da Gen, Enc, Dec su uno spazio di messaggi M come **perfettamente sicuro** se: *per ogni distribuzione di probabilità su M (per ogni messaggio m) e per ogni distribuzione di probabilità su C (per ogni cifrato c) si ha che:*

$$Prob(M|C) = Prob(M)$$

ovvero che l'avversario non ha alcuna informazione sul messaggio in chiaro M osservando il cifrato C.
È possibile dimostrare che se lo schema è perfetto, allora $|K| \geq |M|$.

Capitolo 3

Crittoanalisi

Il *principio di Kerchoffs* afferma che **la sicurezza di un sistema crittografico non deve dipendere dalla segretezza dell'algoritmo crittografico**, ma solo da tenerne celata la chiave. In altre parole, il sistema deve rimanere sicuro anche nell'ipotesi che il nemico conosca l'algoritmo di criptazione.
Esistono diverse situazioni di attacco, che corrispondono a gradi di sicurezza; più un sistema resiste a livelli di attacco alti, più è robusto:

- *Known Ciphertext Attack:* l'avversario cononosce solo il cifrato
- *Known Plaintext Attack:* l'avversario conosce anche il testo in chiaro
- *Chosen Plaintext Attack:* l'avversario può ottenere la cifratura di un testo in chiaro a sua scelta
- *Chosen Ciphertext Attack:* l'avvversario può ottenere la decifratura di un testo cifrato a sua scelta
- *Chosen Text Attack:* l'avversario può ottenere la cifratura e la decifratura di coppie di testi chiaro/cifrato

$\Rightarrow$ se un cifrario viene rotto conoscendo solo il testo cifrato, allora è molto debole ...

3.1 Cifrari a shift

Dato che prendono una lettera e la spostano di posizione, la struttura sottostante del testo rimane la stessa; in particolare, la frequenza delle lettere cifrate corrisponde alla frequenza delle lettere in chiaro nella lingua in cui è stato scritto il testo.
Basta mappare le lettere fino a che viene trovata la chiave; computazionalmente parlando è molto semplice.

3.2 Cifrario di Hill

Nella sua versione standard, il cifrario di Hill è vulnerabile al ***known plaintext attack***, poiché completamente lineare: nel caso in cui un attaccante intercetti n^2 coppie chiaro-cifrato, può impostare un sistema lineare che può essere risolto. Può capitare che il sistema risulti indeterminato, ma è sufficiente aggiungere qualche altra coppia per renderlo risolvibile.
Per questa ragione, è richiesto poco tempo per rompere il cifrario.

3.3 Cifrario di Vigenerè - Metodo Kasiski

Il cifrario di Vigenerè resiste all'analisi delle frequenze, dato che una lettera cifrata corrisponde a più simboli in chiaro ed esiste un grande numero di possibli chiavi. Un possibile attaco è il ***known ciphertext attack***, poiché permette di usare metodi statistici per individuare la lunghezza della chiave, per poi applicare un'analisi delle frequenze in ognuno degli alfabeti cifranti corrispondenti alle lettere della chiave.
Il metodo di Kasiski, applicabile a Vigenerè e simili, consiste nell'osservare che ci sono frequenze di caratteri identiche, poste ad una certa distanza di loro; questa distanza può corrispondere, con una certa probabilità, alla lunghezza della chiave o ad un suo multiplo.
Individuando tutte le sequenze ripetute (facile con un testo lungo) è probabile che la chiave sia pari al MCD tra le distanze delle frequenze ottenute, o un suo multiplo.
A questo punto, si può svolgere un'analisi delle frequenze.

3.3.1 Indice di coincidenza

L'indice di coincidenza di una stringa $x_1x_2 \dots x_n$, rappresentato con la notazione $IC(x_1x_2 \dots x_n)$, rappresenta la probabilità che due caratteri presi a caso nella stringa siano uguali. Lo possianmo definire come

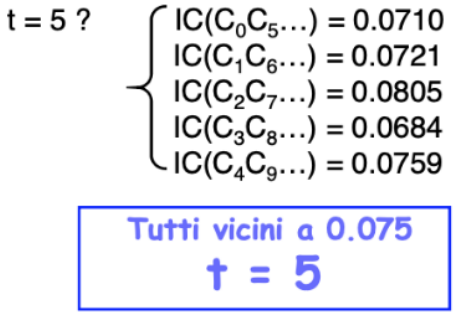
$$IC(x_1, x_2, ...x_n) = \frac{\sum_{i=0}^{25} n_i(n_i - 1)}{n(n-1)}$$

dove n_i rappresenta il numero di occorenze della i-esima lettera dell'alfabeto.
Con il calcolo dell'indice di coincidenza per una lettera su un testo cifrato, possiamo stabilire con quale lingua abbiamo a che fare (in base agli indici di coincidenza dei vari alfabeti); se vogliamo determinare la lunghezza t della chiave (t rappresenta il periodo di ripetizione), possiamo procedere in questo modo:

- Se $t = 1$, significa che si sta usando una sostituzione monoalfabetica; posso dunque calcolare l'indice di coincidenza sul cifrato, e se ottengo valori distanti da quelli delle varie lingue, posso scartare l'ipotesi di $t = 1$

- Proseguo ipotizzando $t = 2$; per poter calcolare l'IC, divido il mio cifrato in due sottotesti con lo shift pari a 2 $\langle (x_0 x_2 \dots), (x_1 x_3 \dots) \rangle$
- ...
- Con $t = n$, si devono calcolare n indici di coincidenza che avranno come caratteri quelli con uno shift pari a n

$\Rightarrow$ si prosegue fino a che si trovano tutti gli indici di coincidenza simili ad un indice di una lingua conosciuta



3.3.2 Indice mutuo di coincidenza

Per trovare i caratteri della chiave si usa l'indice mutuo di coincidenza (IMC) che ci dice qual è la probabilità di estrarre casualmente da 2 stringhe 2 caratteri uguali. L'espressione che permette di calcolare IMC è

$$IMC(x_1, x_2, ..., x_n; y_1, y_2, ..., y'_n) = \frac{\sum_{i=0}^{25} f_i \times f'_i}{n \times n'}$$

dove f_i rappresenta il numero di occorrenze del carattere i-esimo nella prima stringa, mentre f'_i rappresenta il numero di occorenze del carattere i-esimo nella seconda stringa.

Capitolo 4

Cifratura Simmetrica - DES

In crittografia un algoritmo di cifratura a blocchi è un algoritmo a chiave simmetrica che opera su un gruppo di bit di lunghezza finita organizzati in un blocco. I blocchi vengono cifrati contemporaneamente.
Con blocchi da n bit in chiaro, si ottengono 2^n possibili input; quando avviene la trasformazione da blocco in chiaro a blocco cifrato, occore che essa sia univoca e **reversibile**; significa che ogni blocco in chiaro deve produrre un blocco cifrato univoco.
La chiave è ciò che effettua questa mappatura tra blocchi in chiaro e blocchi cifrati; se ho un blocco di n bit, la chiave deve avere una lunghezza di almeno $n \cdot 2^n$ affinché possa coprire tutte le possibili mappature.
L'idea è che ogni valore del blocco venga mappato su un altro valore del blocco (quindi con una mappatura 1:1). Ad esempio, con un blocco di due bit:

Testo chiaro	cifrato
00	11
01	10
10	01
11	01

I blocchi piccoli equivalgono ad una cifratura a sostituzione (vulnerabili dunque ad analisi statistica), mentre i blocchi di grandi dimensioni portano a problemi relativi alla dimensione della chiave.
Le operazioni di **cifratura e decifratura** vengono svolte consultando la tabella di mappatura.

4.1 Cifratura di Feistel

Un cifrario di Feistel è un **algoritmo di cifratura a blocchi** con una particolare struttura, usata da molti algoritmi. Ha il vantaggio che **la cifratura e**

decifratura sono molto simili, spesso identiche, e che basta invertire il funzionamento del gestore della chiave per ottenere l'operazione inversa (permettendo di usare gli stessi circuiti).
L'idea è quella di usare **cifrature in sequenza** per ottenere cifrature più complesse di una singola componente, alternando operazioni di sostituzione e permutazione; queste operazioni, reiterate più volte (*round*), conferiscono all'algoritmo le proprietà di **confusione** e **diffusione**:

- **confusione:** consiste nel rendere la correlazione tra la chiave e il testo cifrato il meno correlata possibile; cerca di non permettere di risalire alla chiave dal cifrato
- **diffusione:** è la capacità dell'algoritmo di distribuire le correlazioni statistiche del testo lungo tutto l'alfabeto usato, rendendo quanto più difficile un attacco statistico

Queste sono due proprietà che un agoritmo di cifratura **deve possedere affinché sia considerabile robusto**, ovvero scarsamente attaccabile attraverso la crittoanalisi perché in grado di contrastare attacchi statistici.

Caratteristiche

Le caratteristiche dei cifrari di Feistel sono le seguenti:

- **Dimensione del blocco:** maggiore è la dimensione, maggiore sarà la sicurezza (al prezzo di prestazioni peggiori)
- **Dimensione della chiave:** analogo alle considerazioni sulla dimensione del blocco
- **Numero di fasi:** tutte le fasi hanno la stessa struttura
- **Algoritmo di schedulazione della chiave:** a partire dalla chiave iniziale, vengono prodotte tante sottochiavi quanti sono i round
- **Funzione da usare ogni round:** maggiore è la complessità, maggiore sarà la resistenza alla crittoanalisi

Poniamo F la funzione dei passaggi e $K_0, K_1, \ldots, K_n$ le sottochiavi dei passaggi $0, 1, \ldots, n$. Il procedimento è:

- l'input viene diviso in due parti uguali L_0 e R_0
- per ogni round i viene calcolato

  $$L_i = R_{i-1} \text{ e } R_i = L_{i-1} \bigoplus f(R_{i-1}, K_{i-1})$$

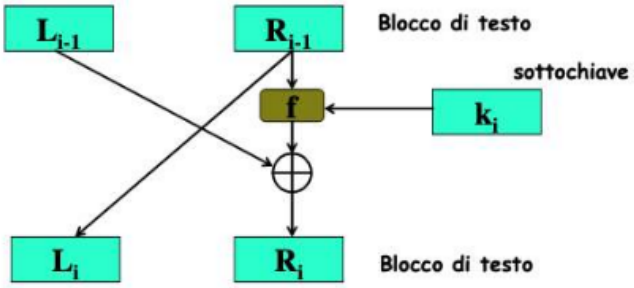
  con f funzione di round e K_i chiave di sessione

$\Rightarrow$ così facendo si ottiene il testo cifrato (L_n, R_n)

Senza considerare la funzione f, la decifrazione si ottiene con

$$R_{i-1} = L_i \text{ e } L_{i-1} = R_i \bigoplus f(L_i, K_i)$$

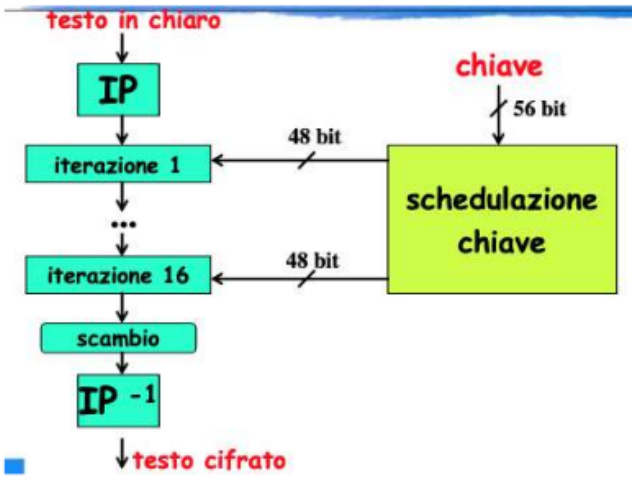
$\rightarrow$ il procedimento è invertibile indipendentemente da f, permettendo di scegliere funzioni f non invertibili e molto complesse.



4.2 DES

È un algoritmo di cifratura a blocchi che usa:

- dimensione del blocco di 64 bit
- chiave di 64 bit, di cui 56 bit usati effettivamente dall'algoritmo e 8 usati per il controllo di parità (per ciascun byte, i primi 7 bit sono parte della chiave mentre l'ultimo è usato per il controllo (corrisponde allo XOR dei precedenti 7))



Come schematizzato nell'immagine, ci sono 16 fasi identiche dette *round*, necessarie per ottenere una diffusione sufficiente. Ci sono poi due operazioni di

permutazione, IP all'inzio e IP^{-1} alla fine, che si annullano (sono dunque inutili ai fini di cifratura) probabilmente usate per facilitare il caricamento dei blocchi sull'hardware degli anni '70.
Prima del ciclo principale, il **blocco viene suddiviso in due metà** e processao alterativamente; questo incrocio viene detto *rete di Feistel*. La struttura della rete assicura che cifratura e decifratura siano processi simili. La funzione (che è quella vista in precedenza) non fa altro che mescolare una metà del blocco con una parte della chiave; il risultato viene poi combinato con l'altra metà del blocco e le due metà vengono scambiate prima del ciclo successivo.
La singola funzione f (vista in precedenza) opera su 4 passi:

- **Espansione:** il mezzo blocco di 32 bit viene espanso fino a 48 bit usando una permtuazione di espansione (E nel blocco che segue)
- **Miscelazione con la chiave:** il risultato è combinato con una sottochiave usando un'operazione di XOR; 16 sottochiavi di 48 bit, una per ogni ciclo, sono derivate dalla chiave principale usando il gestore della chiave
- **Sostituzione:** il blocco viene diviso in 8 parti da 6 bit ciascuna che vengono processati con la *substitution box* (s-box); ognuna delle 8 parti sostituisce 6 bit in input con 4 in output mediante uan trasformazione non lineare; le s-box forniscono il **principale elemento di sicurezza di DES**, siccome senza di esse la cifratura sarebbe lineare
- **Permutazione:** i 32 bit risultanti dalle s-box sono riordinati in base alle permutazioni fisse (p-box)

→ l'alternanza di s-box e p-box garantiscono sia diffusione che confusione
Il **gestore delle chiavi di cifratura**, ovvero quello che genera le sottochiavi a partire dalla chiave iniziale, funziona secondo i seguenti passi:

- vengono selezionati 56 dei 64 bit iniziali tramite una funzione detta **permuted choice 1**; i restanti 8 sono scartati e usati come controllo di parità; nella funzione *PC1*, i bit di parità sono multipli di 8
- i 56 bit sono divisi in due parti di 28 bit che verranno trattate separatamente; nei cicli vengono fatte un totale di 28 permutazioni
- a questo punto vengono scelti, per ogni round, 24 bit da ciascuno dei due sottoinsiemi ottenuti mediante la funzione **permuted choice 2**

4.2.1 Modalità operative

In crittografia la *modalità operativa* di un cifrario a blocchi è una serie di procedimenti standard per garantire la sicurezza; le modalità operative di DES sono:

- **Electronic Codebook Chaining (ECB):** è la più semplice; il testo in chiaro viene suddiviso in blocchi da 64 bit; a parità di testo e di chiave, il cifrato sarà sempre lo stesso; non è adatto a messaggi lunghi

- **Cipher Block Chaining (CBC):** supera il limite di ECB (a parità di testo e di chiave, si ottiene un cifrato differente). L'input è dato dallo XOR tra il blocco di testo corrente e il blocco cifrato precedente (per ciascun blocco viene usata la stessa chiave).

  Il primo blocco viene cifrato con un *inizialization vector*, che deve essere noto anche al destinatario per permette la decifratura; per evitare attacchi di *replay*, l'iv viene modificato per ogni istanza di esecuzione.

  La dipendenza tra i blocchi genera un **rallentamento** e rende l'algoritmo soggetto alla **propagazione degli errori**.

- **Cipher Feedback (CFB):** idealmente si vuole **convertire una cifratura a blocchi in una a flusso**; il processo di cifratura avviene utilizzando un registro di scorrimento a 64 bit che, inizialmente, contiene l'iv. I primi s bit del testo in chiaro vengono posti a XOR con i primi s bit dell'iv; a questo punto, si fa scorrere di s bit il contenuto del registro e gli ultimi s bit, che rimmarrebbero vuoti, sono riempiti con i bit cifrati appena calcolati.

  Il valore di s può essere scelto a piacimento. Questo approccio ha lo svantaggio di essere soggetto alla **propagazione degli errori** e di non essere efficiente per valori di s piccoli

- **Counter (CTR):** viene usato un contatore; il requisito fondamentale è che sia diverso per ogni blocco. Per la cifratura, viene fatto lo XOR tra il contatore (cifrato) e il testo in chiaro; per la decifratura si usa la stessa sequenza di valori del contatore con il testo cifrato. Ha i vantaggi di:

  - *efficienza hw e sw;* l'esecuzione può essere fatta in parallelo su più blocchi
  - *pre-elaborazione;* è possibile precalcolare l'output
  - *accesso casuale*
  - *sicurezza dimostrabile*
  - *semplice*, siccome richiede solo l'algoritmo crittografico

4.3 Sicurezza dei cifrari moderni

Con **sicurezza perfetta** intendiamo che un attaccante non possa ricavare alcuna informazione del testo cifrato.
Con **sicurezza computazionale** intendiamo che se dotato di abbastanza tempo e risorse, un attaccante possa rompere il cifrario (è quindi un rilassamento della sicurezza perfetta).
La sicurezza perfetta prevede quindi che l'avversario sia dotato di capacità computazionali e temporali illimitate, ma che non riesca comunque a ottenere alcuna informazione dal testo cifrato. Nella pratica, uno schema che rivela informazioni con probabilità $2^{-}60$ ad avversari che possono investire 200 anni di sforzo computazionale è considerato molto buono.

Esistono due approcci che mirano a garantire la sicurezza di un cifrario:

- concreto
- asintotico

4.3.1 Approccio concreto

Limita la probabilità di successo massima di qualsiasi avversario che esegua un attacco per una specifica quantità di tempo.
Uno schema (t, ϵ) è sicuro se qualsiasi avversario che ha a disposizione tempo t riesce a rompere lo schema con probabilità ϵ.
I cifrari moderni sono considerati ottimali se, con una chiave di lunghezza n, un avversario riesce a rompere lo schema con probabilità $1/2^n$ adottando un attacco di forza bruta.

4.3.2 Approccio asintotico

Nell'ambito della sicurezza computazionale, uno schema è considerato sicuro se un avversario che opera in tempo polinomiale rompe lo schema con probabilità trascurabile. Introduce un parametro di sicurezza n intero, utilizzato per parametrizzare sia lo schema che lo parti coinvolte; è noto agli avversari, e i tempi di esecuzione degli avversari e la probabilità di successo sono espresse in funzione di n:
Il parametro n permette di calibrare la sicurezza dello schema di cifratura al livello desiderato, considerando n come la lunghezza della chiave; il tempo di ricerca della chiave deve crescere in maniera esponenziale rispetto alla lunghezza delle chiave.

4.4 Crittoanalisi di DES

Il metodo più efficaciente per violare DES è un attacco di forza bruta (avendo una chiave di 56 bit). Esistono però altre due strategie:

- crittoanalisi differenziale
- crittoanalisi lineare

Sono delle tecniche generali di crittoanalisi che hanno successo con molti cifrari.

4.4.1 Forza bruta

Un attacco di forza bruta è il metodo di attacco più semplice, in quanto consiste nel provare tutte le possibili chiavi; in DES, avendo una chiave a 56 bit, ci sono 2^56 possibili chiavi. Nel caso di *known plain text attack*, in genere bastano 2^55 tentativi.
L'approccio a questo tipo di problema è legato al compromesso tra memoria e tempo di calcolo: si può efficientarne uno aumentando il costo dell'altro. È

possibile creare una tabella di ricerca che tracci le corrispondenze tra testo in chiaro e cifrato per ogni chiave possibile; si può calcolare l'intera tabella (aumentando i costi di memoria) o solo alcune voci (aumentando il tempo di calcolo).
Nel momento in cui si ottiene il cifrato, la decifrazione si limiterà alla sola ricerca della chiave nella tabella; questo processo, che ha tempo di esecuzione costante o al più logaritmico, richiede un tempo di precomputazione e spazio di memoria pari a 2^56. L'idea per costruire la tabella è quella di usare una funzione di riduzione che, data una stringa da 64 bit, ne ricava una di 56 bit.
$\rightarrow$ data una stringa di 64 bit x, posso applicare la funzione di riduzione e usare quei 56 bit come chiave

fanculo è da finire sta parte