Università degli Studi di Verona

Crittografia

Riassunto dei principali argomenti

Davide Bianchi

October 15, 2018

Contents

1 Introduzione

Iniziamo dando alcune definizioni fondamentali. Si useranno i termini *ciphertext* e *plaintext* per indicare rispettivamente il testo cifrato e quello in chiaro.

Definizione 1.0.1 (Crittografia) *Insieme dei metodi per rendere un messaggio non leggibile ad altri.*

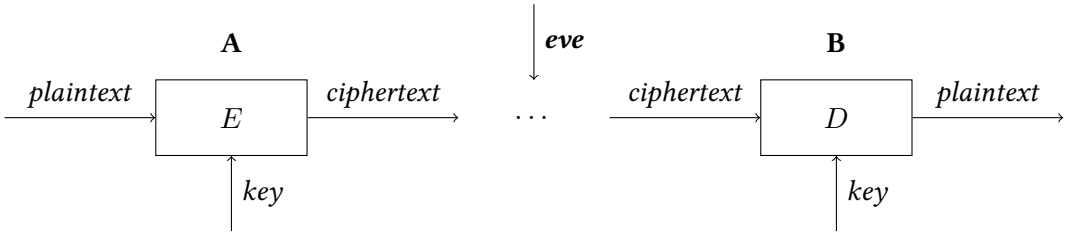
Definizione 1.0.2 (Steganografia) *Insieme dei metodi per nascondere l'esistenza di un messaggio in un altro contenuto.*

Definizione 1.0.3 (Crittoanalisi) *Analisi del ciphertext per ottenere il plaintext corrispondente.*

Definizione 1.0.4 (Bit) *Conoscenza di un evento che ha probabilità* 0.5 *di verificarsi. Da ciò si ricava che*

$$I = -log_2 P$$

Un generico sistema crittografico è strutturato come:



dove:

- **A** e **B** sono i due enti che si devono scambiare il messaggio cifrato;
- ***eve*** è un terzo che tenta di decrittare il messaggio;
- E e D sono i sistemi di *encrypt* e *decrypt*;
- i 3 punti (. . .) rappresentano il mezzo impiegato per la trasmissione del messaggio (qualunque esso sia).

Ovviamente, per *eve*, che tenta di decrittare il messaggio, non ha senso tentare di insistere se il costo della decrittazione è maggiore del valore dell'informazione da proteggere. In tal senso il sistema di cifratura è "sicuro".

La funzione E, inoltre, per essere sufficientemente affidabile, deve essere invertibile (in caso contrario non si potrebbe decrittare il messaggio), ma deve essere difficile fare ciò (in tal caso si dice che E è una *one-way function*). Con la chiave la funzione diventa facile anche da invertire (*one-way trapdoor*).

2 Cifrari storici

2.1 Cifrario di Cesare

Il messaggio viene cifrato sostituendo ogni lettera l del messaggio con la $l + k$ esima lettera dell'alfabeto; la chiave quindi è data dalla coppia $(l, l + k)$.

Il cifrario di Cesare è facile da attaccare in quanto basta un attacco *bruteforce*, quindi è sufficiente provare tutte le combinazioni (che sono in totale 26).

2.2 Permutazione casuale

Supponiamo di usare come cifrario una permutazione casuale dell'alfabeto, ovvero sostituendo ad ogni lettera dell'alfabeto un'altra lettera, in modo totalmente casuale. In tal caso l'attacco bruteforce richiederebbe tempo eccessivo (ci sono 26! pssibili combinazioni da provare, che sono decisamente troppe).

La tecnica usata per attaccare questo tipo di crittografia è l'*analisi delle frequenze*, ovvero l'analisi delle lettere che capitano di più in una data lingua, e associare la lettera del messaggio cifrato con una data frequenza con quella nella lingua del messaggio con una frequenza simile.

2.3 Cifrario a coppie di lettere

Supponiamo di avere come chiave la parola *MONARCHY*. Generiamo una tabella $5x5$, che viene riempita inserendo la chiave prima e poi le altre lettere dell'alfabeto in ordine, escludendo le lettere della chiave.

M	O	N	A	R
C	H	Y	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

Per iniziare è necessario dividere il plaintext in coppie di lettere. Per codificare una lettera ci si sposta lungo la riga fino ad incolonnarsi sotto l'altra lettera della coppia. La lettera del ciphertext sarà quella in cui ci si trova.

Attaccare il cifrario è semplice: è sufficiente analizzare le frequenze di coppie di lettere, anche se è leggermente più complesso che analizzare le singole lettere.

2.4 Cifrario di Vigenère

Il cifrario di Vigenère riprende l'idea del cifrario di Cesare. L'idea è la seguente: presa una chiave (es. *key*), si ripete la chiave tante volte quanto è lungo il testo (eventualmente troncando l'ultima ripetizione), e si codifica la lettera con il corrispondete cifrario di Cesare.

K E Y K E Y K E Y K E Y
P R O V A D I T E S T O

La prima lettera del cipher text sarà la lettera ottenuta dal cifrario di Cesare di chiave (K, P), la seconda con la chiave (E, R) e così via.

Anche questo cifrario è semplice da attaccare, si parte dalla divisione del ciphertext in gruppi di lunghezza pari a quella della chiave, e si esegue l'analisi delle frequenze su ogni gruppo.

2.5 Enigma

Consideriamo ora la rappresentazione binaria di un cifrario di Cesare. Esistono solamente due chiavi possibili:

$$(0,1);(0,0)$$

Consideriamo quindi un semplice esempio, con $K = 0011, P = 0101$. Applicando Vigenère otteniamo che $C = 0110$. Si può notare che, date K_i una qualsiasi lettera della chiave e P_i una qualsiasi lettera del plaintext, $C_i = K_i \oplus P_i$. Pertanto si può dedurre che $C = K \oplus P$. Vale in maniera analoga anche $P = K \oplus C$.

È stato già verificato che il cifrario di Vigenère sia semplice da attaccare, ma se $|K| = |P|$ allora non è più applicabile la crittoanalisi data per il cifrario di Vigenère.

Tuttavia, se vale $|K| = |P|$, allora significa che la chiave è sostanzialmente una sequenza di bit casuali. In tal caso il ciphertext non avrebbe *nessuna* relazione con il plaintext, e non sarebbe possibile ricavare alcuna informazione utile. Il problema di questo metodo sta nel fatto che ogni chiave è utilizzabile una sola volta, in caso contrario sarebbe possibile ricavare informazioni dai vari ciphertext ottenuti con la stessa chiave. (si parla in questo caso di *one-time pad*)

La macchina Enigma (usata dai tedeschi nella II G.M.) usava questo metodo: cambiando chiave ogni giorno garantiva ciphertext difficilissimi da decifrare. Infatti generava una sequenza di permutazioni per una lettera, poi ruotava una permutazione, poi quella successiva e coosì via per ogni lettera.

Ciò che rese decifrabili le comunicazioni generate da questa macchina fu il fatto che ogni messaggio iniziasse con la stessa sequenza di caratteri (un saluto ad Hitler).

3 Crittografia moderna

Consideriamo per un attimo i possibili attaccanti che un sistema crittografico deve affrontare per essere affidabile:

- *known cypertext attacker:* questo attaccante è il meno aggressivo e conosce solamente il testo cifrato;
- *known plaintext attacker:* conosce entrambi i tipi di testo;
- *choosen plaintext:* può scegliere il plaintext da codificare e analizzare il ciphertext ottenuto;
- *adaptive choosen plaintext:* può liberamente scegliere il plaintext da far codificare e comportarsi di conseguenza, sulla base del risultato appena ottenuto.

Fino ad ora per attaccare i cifrari visti è bastato solamente conoscere il ciphertext. Passiamo a scenari più complessi.

3.1 Data Encryption Standard (DES)