Cifrari Perfetti: One-Time Pad

Crittografia

Luciano Margara

Unibo

2022

Domanda

Esistono cifrari inviolabili ?

Cifrari inviolabili

Esistono cifrari in grado di nascondere l'informazione con certezza assoluta ma a un costo così alto da mettere in dubbio la loro esistenza pratica. Il loro impiego, o meglio l'impiego di una buona approssimazione di essi, è limitato a comunicazioni sporadiche in casi di estrema segretezza.

Cifrari utilizzati in pratica

Esistono cifrari non inviolabili, ma sufficientemente sicuri ed economici, che vengono utilizzati ogni giorno per comunicazioni di massa.

Modello Matematico

La comunicazione tra un mittente $Mitt$ e un destinatario $Dest$ è modellata come un processo stocastico in cui il comportamento del mittente è descritto da una variabile aleatoria M che assume valori nello spazio Msg dei messaggi, e le comunicazioni sul canale sono descritte da una variabile aleatoria C che assume valori nello spazio $Critto$ dei crittogrammi.
La distribuzione di probabilità della M dipende dalle caratteristiche della sorgente, cioè dalla propensione di $Mitt$ a spedire diversi messaggi.

Modello Matematico

- $Pr(M = m)$ è la probabilità che $Mitt$ voglia spedire il messaggio m a $Dest$
- $Pr(M = m|C = c)$ come la probabilità a posteriori che il messaggio inviato sia effettivamente m dato che c è il crittogramma in transito.

Scenario peggiore

Il crittoanalista che ha intercettato c è in possesso di tutta l'informazione possibile sul sistema tranne la chiave segreta. In particolare egli conosce la distribuzione di probabilità con cui *Mitt* genera i messaggi, il cifrario utilizzato e lo spazio *Key* delle chiavi.

Cifrario perfetto (Shannon)

Un cifrario è perfetto se per ogni $m \in Msg$ e per ogni $c \in Critto$ vale la relazione

$$Pr(M = m | C = c) = Pr(M = m)$$

Cifrario perfetto (Shannon)

Utilizzando un cifrario perfetto la conoscenza complessiva del crittoanalista non cambia dopo che egli ha osservato un crittogramma arbitrario c transitare sul canale.

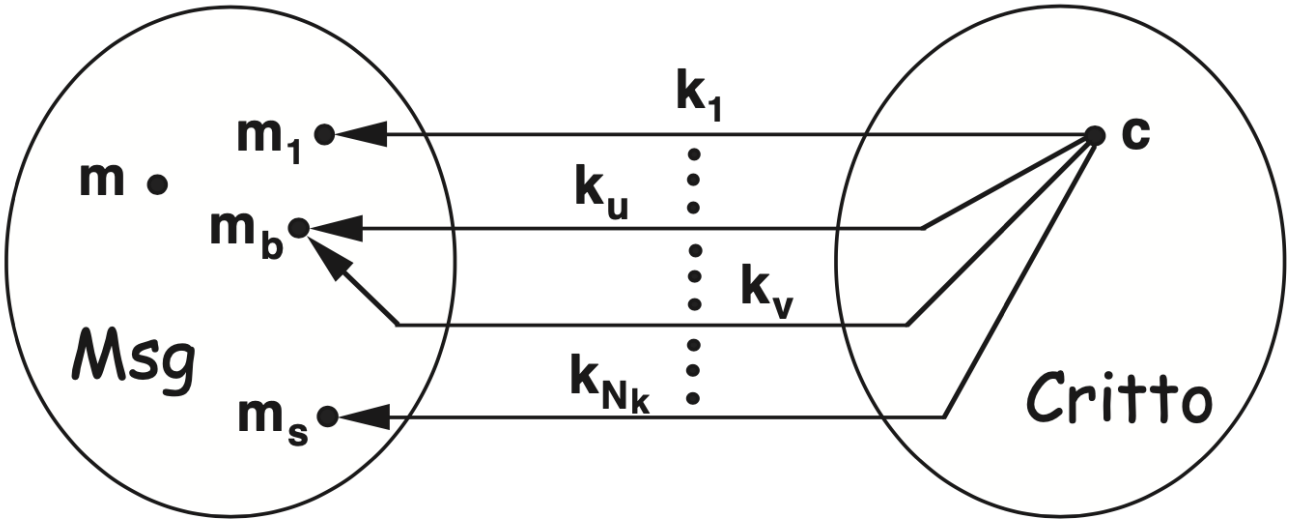
Numero di chiavi in un cifrario perfetto

In un cifrario perfetto il numero delle chiavi deve essere maggiore o uguale al numero dei messaggi possibili.

Numero di chiavi in un cifrario perfetto

Sia N_m il numero di messaggi possibili, cioè tali che $Pr(M = m) > 0$, e sia N_k il numero di chiavi. Poniamo per assurdo che sia $N_m > N_k$. A un crittogramma c, con $Pr(C = c) > 0$, corrispondono $s \leq N_k$ messaggi (non necessariamente distinti) ottenuti decrittando c con tutte le chiavi. Poiché $N_m > N_k \geq s$, esiste almeno un messaggio m con $Pr(M = m) > 0$ non ottenibile da c. Questo implica $Pr(M = m|C = c) = 0 \neq Pr(M = m)$, ovvero per $N_m > N_k$ il cifrario non è perfetto.

Numero di chiavi in un cifrario perfetto



$|Msg| = N_m \qquad N_k \geqslant s \qquad |Critto| = N_c$

Numero di chiavi in un cifrario perfetto

Questo teorema dimostra che l'uso di cifrari perfetti è necessariamente molto costoso poiché richiede chiavi lunghissime per descrivere l'intero spazio Key, più numeroso di quello di tutti i messaggi possibili, con conseguenti difficoltà nella gestione e nello scambio segreto delle chiavi stesse.

One-Time Pad

One-Time Pad è un cifrario perfetto molto semplice e veloce inventato da Mauborgne e Vernam nel 1917. Oltre che per l'efficienza della sua realizzazione, l'interesse per questo cifrario è giustificato da motivi storici poiché fu utilizzato, a quanto si sa, per le comunicazioni diplomatiche tra Washington e Mosca durante la guerra fredda

One-Time Pad

Generazione della chiave segreta:
Si costruisce una sequenza $k = k_1 k_2 \ldots$ di bit, nota a *Mitt* e a *Dest*, avente lunghezza maggiore o uguale a quella del messaggio da scambiare. Ogni bit k_i di k è scelto perfettamente a caso tra i valori 0 e 1.

One-Time Pad

Cifratura:
Se il messaggio da trasmettere è la sequenza di n bit $m = m_1 m_2 \ldots m_n$, il crittogramma $c = c_1 c_2 \ldots c_n$ si genera bit a bit ponendo $c_i = m_i \oplus k_i$, per $i = 1, 2, \ldots, n$.

One-Time Pad

Decifrazione:
Presi gli n bit iniziali $k_1 k_2 \ldots k_n$ della chiave segreta e il crittogramma c, il messaggio m è ricostruito bit a bit ponendo $m_i = c_i \oplus k_i$, per $i = 1, 2, \ldots, n$. (Infatti si ha $c_i \oplus k_i = m_i \oplus k_i \oplus k_i = m_i$)

One-Time Pad: Esempio

msg	0	1	1	0	0	1
key	1	1	0	1	0	1
c	1	0	1	1	0	0
key	1	1	0	1	0	1
msg	0	1	1	0	0	1

One-Time Pad: perfezione

Per semplicità assumeremo che:

- Tutti i messaggi hanno la stessa lunghezza n.
- Tutte le sequenze di n bit sono messaggi possibili.

One-Time Pad: perfezione

Utilizzando una chiave scelta perfettamente a caso per ogni messaggio, il cifrario One-Time Pad è perfetto e impiega un numero minimo di chiavi.

One-Time Pad: perfezione

Dimostrare che il cifrario è perfetto significa provare la validità della relazione $Pr(M = m|C = c) = Pr(M = m)$. Applicando la definizione di probabilità condizionale possiamo riscrivere il termine sinistro della formula come: $Pr(M = m|C = c) = Pr(M = m, C = c)/Pr(C = c)$.

One-Time Pad: perfezione

Osserviamo ora che l'evento $\{M = m, C = c\}$ descrive la situazione in cui *Mitt* ha generato il messaggio m e l'ha cifrato come crittogramma c. Per la definizione di XOR, fissato il messaggio, chiavi diverse danno origine a crittogrammi diversi, e ogni chiave può essere generata con probabilità $(1/2)^n$.

One-Time Pad: perfezione

Dunque, fissato m, risulta $Pr(C = c) = (1/2)^n$ per ogni c, cioè la probabilità dell'evento $\{C = c\}$ è costante e quindi indipendente da m, ovvero gli eventi $\{M = m\}$ e $\{C = c\}$ sono indipendenti tra loro e si ha:
$Pr(M = m, C = c) = Pr(M = m) \times Pr(C = c)$.
Combinando i risultati precedenti otteniamo infine $Pr(M = m | C = c) = Pr(M = m)$, cioè il cifrario One-Time Pad è perfetto.

One-Time Pad: minimalità dell'insieme delle chiavi

Il numero di chiavi non può essere inferiore al numero di messaggi. Nel One-Time Pad il numero di chiavi è 2^n che è uguale al numero di messaggi (non ne possiamo avere di meno).

One-Time Pad

- Perfetto ! ma ...
- Chiavi troppo lunghe
- Chiavi si consumano