Cenni di teoria dell'informazione

Gregorio D'Agostino

18 Maggio 2021

"Risultati base per il cifrario trasposizionale a blocchi"

Mersenne Twister

Esercizi

Il cifrario di trasposizione a blocchi è ideale

Data una sorgente stazionaria e senza memoria (ξ_i indipendenti tra loro ed egualmente distribuite), il cifrario a blocchi è ideale. Se $C^{n,b}$ sono le sequenze di n blocchi di lunghezza b, cioè i possibili crittogrammi:

$$H(K) = H(K|C^{n,b}).$$

Lo dimostriamo ricordando che la cifratura e la decifrazione sono deterministiche e dunque: $P(M|K,C) = 1$ e $P(C|M,K) = 1$; $P(M'|K,C) = 0$ per $M' \neq M$ e $P(C'|M,K) = 0$ per $C' \neq C$:

$$P(K,C) = P(M|K,C) \cdot P(K,C) = P(M,K,C) =$$

$$= P(M,K) \cdot P(C|M,K) = P(M,K).$$

Confrontando le due eguaglianze:

$$P(K,C) = P(M,K).$$

Il cifrario di trasposizione a blocchi è ideale - cont

Ricordando le probabilità condizionate $P(K, C) = P(K) \cdot P(C|K)$:

$$P(K, M) = P(K, C) = P(K) \cdot P(C|K);$$

essendo M e K indipendenti tra loro $P(K, M) = P(K) \cdot P(M)$:

$$P(K, C) = P(K) \cdot P(C|K) = P(K) \cdot P(M).$$

cioè, semplificando $P(K)$ nel secondo e terzo membro:

$$P(C|K) = P(M),$$

che non dipende da K e quindi:

$$P(C|K) = P(C).$$

Decrittare il cifrario di trasposizione a blocchi

- Due lezioni fa abbiamo fatto un esercizio in cui conoscendo un numero crescente di coppie messaggio-crittogramma riuscivamo a scoprire la chiave. Si trattava di un Known Plaintext Attack. In quel caso un soggetto terzo sceglieva i messaggi da testare.

Decrittare il cifrario di trasposizione a blocchi

- Due lezioni fa abbiamo fatto un esercizio in cui conoscendo un numero crescente di coppie messaggio-crittogramma riuscivamo a scoprire la chiave. Si trattava di un Known Plaintext Attack. In quel caso un soggetto terzo sceglieva i messaggi da testare.
- Scegliendo tutti uno o tutti zero non si ottenevano informazioni. Ma negli altri casi sì.
  Dimostreremo che: data una sorgente stazionaria e senza memoria (ξ_i indipendenti tra loro ed egualmente distribuite), l'entropia delle chiavi tende a zero all'aumentare delle coppie messaggio crittogramma note. Siano $M^{n,b}$ le sequenze di blocchi di lunghezza b non codificate e $C^{n,b}$ i loro crittogrammi:

$$\lim_{n\to\infty} H(K|M^{n,b}, C^{n,b}) = 0.$$

  Dire che l'entropia tende a zero equivale a dire che la chiave è nota.

Il cifrario di trasposizione a blocchi - Stime della chiave

Supponiamo di avere una regola per stimare la chiave. Non importa quale regola, purché sia compatibile con le coppie messaggio in chiaro crittogramma. Vogliamo domandarci qual è la probabilità che la stima non sia esatta. In ogni blocco di lunghezza b ci devono essere ripetizioni di caratteri altrimenti il confronto delle due sequenze in chiaro e crittata fornisce la permutazione giusta. Quindi calcoliamo la probabilità che vi siano ripetizioni di caratteri.

Se l'alfabeto ha dimensione L e caratteri $\alpha_1, \alpha_2, \ldots, \alpha_L$ con probabilià $p_1, p_2, \ldots, p_L$. La probabilità che il carattere α_k si ripeta nelle posizioni i e j è $(p_k)^2$. La propabilità che si ripeta uno qualunque dei caratteri nelle posizioni i e j è la somma delle probabilità di ognuno degli eventi disgiunti:

$$p(rip(i,j)) = \sum_{k=1}^{L} p_k^2 \stackrel{def}{=} v.$$

Vediamo un esempio

- Supponiamo di cifrare blocchi da 8 con un metodo puramente trasposizionale. La sequenza $a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8$ sarà permutata nella stringa $a_{i_1} a_{i_2} a_{i_3} a_{i_4} a_{i_5} a_{i_6} a_{i_7} a_{i_8}$.

Vediamo un esempio

- Supponiamo di cifrare blocchi da 8 con un metodo puramente trasposizionale. La sequenza $a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8$ sarà permutata nella stringa $a_{i_1} a_{i_2} a_{i_3} a_{i_4} a_{i_5} a_{i_6} a_{i_7} a_{i_8}$.
- Ad esempio:
  10010110 va in 01110110.
  i_1 può essere 2,3,5 e 8 ma non 1,4, 6 e 7
  analogamente per i_5 e i_8 (tutti quelli che hanno uno zero nella sequenza cifrata).
  i_2 può essere 1,4, 6 e 7 ma non 2,3,5 e 8
  analogamente per i_3 i_4 i_6 e i_7 (tutti quelli che hanno un 1 nella sequenza cifrata).

Vediamo un esempio

- Supponiamo di cifrare blocchi da 8 con un metodo puramente trasposizionale. La sequenza $a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8$ sarà permutata nella stringa $a_{i_1} a_{i_2} a_{i_3} a_{i_4} a_{i_5} a_{i_6} a_{i_7} a_{i_8}$.
- Ad esempio:
  10010110 va in 01110110.
  i_1 può essere 2,3,5 e 8 ma non 1,4, 6 e 7
  analogamente per i_5 e i_8 (tutti quelli che hanno uno zero nella sequenza cifrata).
  i_2 può essere 1,4, 6 e 7 ma non 2,3,5 e 8
  analogamente per i_3 i_4 i_6 e i_7 (tutti quelli che hanno un 1 nella sequenza cifrata).
- Qualunque sia la regola che utilizziamo per stimare K con $\hat{K}$ deve rispettare questi vincoli cioè le coppie sequenza in chiaro-crittogramma conosciute.

Vediamo un esempio con 3 caratteri

- Supponiamo di cifrare blocchi da 8 con un metodo puramente trasposizionale. La sequenza $a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8$ sarà permutata nella stringa $a_{i_1} a_{i_2} a_{i_3} a_{i_4} a_{i_5} a_{i_6} a_{i_7} a_{i_8}$.

Vediamo un esempio con 3 caratteri

- Supponiamo di cifrare blocchi da 8 con un metodo puramente trasposizionale. La sequenza $a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8$ sarà permutata nella stringa $a_{i_1} a_{i_2} a_{i_3} a_{i_4} a_{i_5} a_{i_6} a_{i_7} a_{i_8}$.
- Ad esempio:
  12012112 va in 21210210.
  i_1 può essere 2,5 e 8 ma non gli altri.
  analogamente per i_3 e i_6 (tutti quelli che hanno un due nella sequenza cifrata).
  i_2 può essere 1,4, 6 e 7 ma non 2,3,5 e 8
  analogamente per i_4 e i_7 (tutti quelli che hanno un 1 nella sequenza cifrata).
  ecc.

Vediamo un esempio con 3 caratteri

- Supponiamo di cifrare blocchi da 8 con un metodo puramente trasposizionale. La sequenza $a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8$ sarà permutata nella stringa $a_{i_1} a_{i_2} a_{i_3} a_{i_4} a_{i_5} a_{i_6} a_{i_7} a_{i_8}$.
- Ad esempio:
  12012112 va in 21210210.
  i_1 può essere 2,5 e 8 ma non gli altri.
  analogamente per i_3 e i_6 (tutti quelli che hanno un due nella sequenza cifrata).
  i_2 può essere 1,4, 6 e 7 ma non 2,3,5 e 8
  analogamente per i_4 e i_7 (tutti quelli che hanno un 1 nella sequenza cifrata).
  ecc.
- in generale non ricaviamo informazioni quando due caratteri sono uguali. I gruppi di caratteri uguali possono provenire dalla stessa posizione iniziale.

Il cifrario di trasposizione a blocchi - Stime della chiave -cont

La probabilità che almeno un carattere si ripeta in ognuno degli n blocchi di lunghezza b in posizioni i e j è il prodotto delle probabilità

$$p(rip(i,j), n - blocchi) = \left(\sum_{k=1}^{L} p_k^2.\right)^n = v^n.$$

La probabilità che la stima $\hat{K}$ della chiave K sia errata è minorata dalla probabilità che per tutte le coppie di posizioni i e j in tutti i blocchi vi siano caratteri ripetuti. Questa è maggiorata da

$$P(K \neq \hat{K}) \leq \frac{1}{2} b \cdot (b-1) v^n.$$

Il cifrario di trasposizione a blocchi - Stime della chiave

La grandezza v è minore di 1:

$$v \stackrel{def}{=} \sum_{k=1}^{L} p_k^2 < \left(\sum_{k=1}^{L} p_i \right) = 1;$$

perché il quadrato è una funzione strettamente convessa.
La probabilità che la stima della chiave possa essere errata tende quindi a zero:

$$P(K \neq \hat{K}) < \frac{1}{2} b \cdot (b-1) v^n < \epsilon$$

per $n > N_\epsilon$

$$N_\epsilon = \frac{1}{-log(v)} \left(-log(\epsilon) + log \left(\frac{1}{2} b \cdot (b-1) \right) \right)$$

Il cifrario di trasposizione a blocchi - Stime della chiave

- Se la probabilità che la stima differisca dalla chiave tende a zero, l'entropia della chiave fissate le coppie messaggio-crittogramma tende a zero:

$$H(K|M^{n,b}, C^{n,b}) \leq H(K|\hat{K}) \to 0.$$

Il cifrario di trasposizione a blocchi - Stime della chiave

- Se la probabilità che la stima differisca dalla chiave tende a zero, l'entropia della chiave fissate le coppie messaggio-crittogramma tende a zero:

$$H(K|M^{n,b}, C^{n,b}) \leq H(K|\hat{K}) \to 0.$$

- Quindi disponendo di sequenze scelte a caso corredate dei crittogrammi associati, all'aumentare del numero di messaggi la chiave diviene nota.

Il cifrario di trasposizione a blocchi - Stime della chiave

- Se la probabilità che la stima differisca dalla chiave tende a zero, l'entropia della chiave fissate le coppie messaggio-crittogramma tende a zero:

$$H(K|M^{n,b}, C^{n,b}) \leq H(K|\hat{K}) \to 0.$$

- Quindi disponendo di sequenze scelte a caso corredate dei crittogrammi associati, all'aumentare del numero di messaggi la chiave diviene nota.
- Anche se il cifrario è ideale si può violare con un attacco a testi noti. Ovviamente se potessimo scegliere i testi (Chosen Plaintext Attack) scopriremmo la chiave molto prima: basta scegliere tutti i caratteri uguali tranne uno in tutte le posizioni diverse.

Cifratura Sequenziale

- La cifratura sequenziale si ottiene senza suddividere le sequenze in blocchi.

Cifratura Sequenziale

- La cifratura sequenziale si ottiene senza suddividere le sequenze in blocchi.
- La cifratura One Time Pad si può estendere al caso sequenziale se si dispone di una sorgente indefinita di bit casuali. Cioè di un generatore di numeri casuali.

Cifratura Sequenziale

- La cifratura sequenziale si ottiene senza suddividere le sequenze in blocchi.
- La cifratura One Time Pad si può estendere al caso sequenziale se si dispone di una sorgente indefinita di bit casuali. Cioè di un generatore di numeri casuali.
- Se disponessimo di una sorgente casuale privata (cioè accessibile solo dai i soggetti autorizzati) si potrebbe realizzare una cifratura perfetta per qualsiasi messaggio.

Cifratura Sequenziale

- La cifratura sequenziale si ottiene senza suddividere le sequenze in blocchi.
- La cifratura One Time Pad si può estendere al caso sequenziale se si dispone di una sorgente indefinita di bit casuali. Cioè di un generatore di numeri casuali.
- Se disponessimo di una sorgente casuale privata (cioè accessibile solo dai i soggetti autorizzati) si potrebbe realizzare una cifratura perfetta per qualsiasi messaggio.
- In pratica questo è impossibile, ma possiamo utilizzate un generatore di sequenze pseudo-casuali per avvicinarci a questa situazione ideale.

Cifratura Sequenziale

- La cifratura sequenziale si ottiene senza suddividere le sequenze in blocchi.
- La cifratura One Time Pad si può estendere al caso sequenziale se si dispone di una sorgente indefinita di bit casuali. Cioè di un generatore di numeri casuali.
- Se disponessimo di una sorgente casuale privata (cioè accessibile solo dai i soggetti autorizzati) si potrebbe realizzare una cifratura perfetta per qualsiasi messaggio.
- In pratica questo è impossibile, ma possiamo utilizzate un generatore di sequenze pseudo-casuali per avvicinarci a questa situazione ideale.
- Il problema si sposta sulla decifrazione dell'algoritmo di generazione delle sequenza pseudo-causali.

Generatori Pseudo-casuali

- Per ottenere cifrari perfetti è necessario disporre di generatori di sequenze di numeri casuali molto lunghe. Per ottenere sequenze genuine di tal genere occorre attendere tempi molti lunghi, quindi si preferisce utilizzare generatori deterministici di numeri pseudo-casuali a velocissima computazione.

Generatori Pseudo-casuali

- Per ottenere cifrari perfetti è necessario disporre di generatori di sequenze di numeri casuali molto lunghe. Per ottenere sequenze genuine di tal genere occorre attendere tempi molti lunghi, quindi si preferisce utilizzare generatori deterministici di numeri pseudo-casuali a velocissima computazione.
- I generatori pseudo random (PRGN) sono funzioni iterative che generano flussi di dati simili alle sequenze casuali. Per molti anni si sono utilizzate varianti dei generatori congruenziali.

Generatori Pseudo-casuali

- Per ottenere cifrari perfetti è necessario disporre di generatori di sequenze di numeri casuali molto lunghe. Per ottenere sequenze genuine di tal genere occorre attendere tempi molti lunghi, quindi si preferisce utilizzare generatori deterministici di numeri pseudo-casuali a velocissima computazione.
- I generatori pseudo random (PRGN) sono funzioni iterative che generano flussi di dati simili alle sequenze casuali. Per molti anni si sono utilizzate varianti dei generatori congruenziali.
- Makoto Matsumoto e Takuji Nishimura (iniziali MT), nel 1997 introdussero il loro famoso algoritmo denominato MT19937. Fu battezzato in onore di Mersenne Mersenne Twister, ma il nome reca anche le iniziali degli autori.

Generatori Pseudo-casuali

- Per ottenere cifrari perfetti è necessario disporre di generatori di sequenze di numeri casuali molto lunghe. Per ottenere sequenze genuine di tal genere occorre attendere tempi molti lunghi, quindi si preferisce utilizzare generatori deterministici di numeri pseudo-casuali a velocissima computazione.
- I generatori pseudo random (PRGN) sono funzioni iterative che generano flussi di dati simili alle sequenze casuali. Per molti anni si sono utilizzate varianti dei generatori congruenziali.
- Makoto Matsumoto e Takuji Nishimura (iniziali MT), nel 1997 introdussero il loro famoso algoritmo denominato MT19937. Fu battezzato in onore di Mersenne Mersenne Twister, ma il nome reca anche le iniziali degli autori.
- Il numero 19937 ricorda la periodicità della funzione: 2^{19937}.

Il generatore di numeri casuali Mersenne Twister

- Si basa su un numero primo di Mersenne. Un numero di Mersenne è della forma $M = 2^p - 1$ in cui p è primo. Un numero primo di Mersenne è un numero di Mersenne primo.

Il generatore di numeri casuali Mersenne Twister

- Si basa su un numero primo di Mersenne. Un numero di Mersenne è della forma $M = 2^p - 1$ in cui p è primo. Un numero primo di Mersenne è un numero di Mersenne primo.
- Il periodo del Mersenne twister è $2^{19937} \approx 10^{6000}$ che allo stato attuale è insondabile.

Il generatore di numeri casuali Mersenne Twister

- Si basa su un numero primo di Mersenne. Un numero di Mersenne è della forma $M = 2^p - 1$ in cui p è primo. Un numero primo di Mersenne è un numero di Mersenne primo.
- Il periodo del Mersenne twister è $2^{19937} \approx 10^{6000}$ che allo stato attuale è insondabile.
- L'algoritmo funziona secondo una ricorrenza lineare che utilizza sequenze X_i di w bit:

$$X_{k+n} = X_{k+m} \oplus (X_k^u || X_{k+1}^l) \cdot A$$

in cui il simbolo $||$ indica la concatenazione delle stringhe; A è una matrice (w x w); e le sequenze X_i^u e X_i^l sono definite come segue:

Il generatore di numeri casuali Mersenne Twister

- Si basa su un numero primo di Mersenne. Un numero di Mersenne è della forma $M = 2^p - 1$ in cui p è primo. Un numero primo di Mersenne è un numero di Mersenne primo.
- Il periodo del Mersenne twister è $2^{19937} \approx 10^{6000}$ che allo stato attuale è insondabile.
- L'algoritmo funziona secondo una ricorrenza lineare che utilizza sequenze X_i di w bit:

$$X_{k+n} = X_{k+m} \oplus (X_k^u || X_{k+1}^l) \cdot A$$

in cui il simbolo $||$ indica la concatenazione delle stringhe; A è una matrice (w x w); e le sequenze X_i^u e X_i^l sono definite come segue:

  - X_i^u è la sotto sequenza dei primi $w - r$ bit della sequenza X_i;

Il generatore di numeri casuali Mersenne Twister

- Si basa su un numero primo di Mersenne. Un numero di Mersenne è della forma $M = 2^p - 1$ in cui p è primo. Un numero primo di Mersenne è un numero di Mersenne primo.
- Il periodo del Mersenne twister è $2^{19937} \approx 10^{6000}$ che allo stato attuale è insondabile.
- L'algoritmo funziona secondo una ricorrenza lineare che utilizza sequenze X_i di w bit:

$$X_{k+n} = X_{k+m} \oplus (X_k^u || X_{k+1}^l) \cdot A$$

in cui il simbolo $||$ indica la concatenazione delle stringhe; A è una matrice (w x w); e le sequenze X_i^u e X_i^l sono definite come segue:

  - X_i^u è la sotto sequenza dei primi $w - r$ bit della sequenza X_i;
  - X_i^l è la sotto sequenza dei ultimi r bit della sequenza X_i;

Il generatore di numeri casuali Mersenne Twister

- Si basa su un numero primo di Mersenne. Un numero di Mersenne è della forma $M = 2^p - 1$ in cui p è primo. Un numero primo di Mersenne è un numero di Mersenne primo.
- Il periodo del Mersenne twister è $2^{19937} \approx 10^{6000}$ che allo stato attuale è insondabile.
- L'algoritmo funziona secondo una ricorrenza lineare che utilizza sequenze X_i di w bit:

$$X_{k+n} = X_{k+m} \oplus (X_k^u || X_{k+1}^l) \cdot A$$

in cui il simbolo $||$ indica la concatenazione delle stringhe; A è una matrice (w x w); e le sequenze X_i^u e X_i^l sono definite come segue:

- X_i^u è la sotto sequenza dei primi $w - r$ bit della sequenza X_i;
- X_i^l è la sotto sequenza dei ultimi r bit della sequenza X_i;
- L'indice m soddisfa la diseguaglianza $1 \leq m \leq n$.

Il generatore di numeri casuali Mersenne twister - cont

- La matrice A può essere scelta nella forma:

$$A = \begin{pmatrix} & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \\ a_{w-1} & a_{w-2} & \cdots & \cdots & a_0 \end{pmatrix};$$

Il generatore di numeri casuali Mersenne twister - cont

- La matrice A può essere scelta nella forma:

$$A = \begin{pmatrix} & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \\ a_{w-1} & a_{w-2} & \cdots & \cdots & a_0 \end{pmatrix};$$

- Dal punto di vista operativo il calcolo del prodotto di X per A si riduce a degli scorrimenti (shift) di registro e o esclusivi (XOR). Ponendo $X = X_{w-1}, X_{w-2}, \ldots, X_0$ e $a = a_{w-1}, a_{w-2} \ldots, a_0$:

$$XA = \begin{cases} rightshift(X) & se \quad x_0 = 0; \\ rightshift(X) \oplus a & se \quad x_0 = 1. \end{cases}$$

Il generatore di numeri casuali Mersenne twister - cont

- Il valore che si fornisce in uscita viene moltiplicato per un'altra matrice T (anch'essa $w \, x \, w$) detta di "tempering":

$$z_k = x_k \cdot T$$

Il generatore di numeri casuali Mersenne twister - cont

- Il valore che si fornisce in uscita viene moltiplicato per un'altra matrice T (anch'essa $w \, x \, w$) detta di "tempering":

$$z_k = x_k \cdot T$$

- Questo è un espediente simile a scegliere i bit centrali per un generatore congruenziale per ottenere le ripetizioni che altrimenti sarebbero impossibili.

Il generatore di numeri casuali Mersenne twister - cont

- I valori tipici del MT sono: $w = 32$, $n = 624$, $m = 397$ ed r=31. $p = 19937 = 624 * 32 - 31 = n * w - r$. In generale funziona quando $n * w - r = p$ è primo e $2^p - 1$ anche.

Il generatore di numeri casuali Mersenne twister - cont

- I valori tipici del MT sono: $w = 32$, $n = 624$, $m = 397$ ed r=31. $p = 19937 = 624 * 32 - 31 = n * w - r$. In generale funziona quando $n * w - r = p$ è primo e $2^p - 1$ anche.
- Il generatore appena descritto si chiama MT-19937 ed è utilizzato da moltissimi sistemi inclusi Octave, Mathlab, gcc, gfortran, pyton, Mathematica. La std (standard lib) del c++ possiede una call MT19937 a 32 bit, ma è possibile scaricare anche una versione a 64 bit.

Il generatore di numeri casuali Mersenne twister - cont

- I valori tipici del MT sono: $w = 32$, $n = 624$, $m = 397$ ed r=31. $p = 19937 = 624 * 32 - 31 = n * w - r$. In generale funziona quando $n * w - r = p$ è primo e $2^p - 1$ anche.
- Il generatore appena descritto si chiama MT-19937 ed è utilizzato da moltissimi sistemi inclusi Octave, Mathlab, gcc, gfortran, pyton, Mathematica. La std (standard lib) del c++ possiede una call MT19937 a 32 bit, ma è possibile scaricare anche una versione a 64 bit.
- Al sito della GNU sono disponibili versioni open source sia a 32 che a 64 bit per vari linguaggi di programmazione https://www.gnu.org/software/gsl/manual/html_node/Random-number-generator-algorithms.html

Altri generatori

- MT-19937 supera tutti i test statistici ed è eseguibile rapidissimamente (pochi cicli per ogni bit random); pertanto ha praticamente soppiantato i generatori precedenti.

Altri generatori

- MT-19937 supera tutti i test statistici ed è eseguibile rapidissimamente (pochi cicli per ogni bit random); pertanto ha praticamente soppiantato i generatori precedenti.
- Abbiamo visto il caso dei generatori congruenziali (modulari) che continuano ad essere usati per applicazioni non crittografiche tipo MC o altro.

Altri generatori

- MT-19937 supera tutti i test statistici ed è eseguibile rapidissimamente (pochi cicli per ogni bit random); pertanto ha praticamente soppiantato i generatori precedenti.
- Abbiamo visto il caso dei generatori congruenziali (modulari) che continuano ad essere usati per applicazioni non crittografiche tipo MC o altro.
- Dimostrare le proprietà probablistiche di MT-19937 e gli altri codici tipo il congruenziale di fibonacci, va al di là del corso.

Altri generatori

- MT-19937 supera tutti i test statistici ed è eseguibile rapidissimamente (pochi cicli per ogni bit random); pertanto ha praticamente soppiantato i generatori precedenti.
- Abbiamo visto il caso dei generatori congruenziali (modulari) che continuano ad essere usati per applicazioni non crittografiche tipo MC o altro.
- Dimostrare le proprietà probablistiche di MT-19937 e gli altri codici tipo il congruenziale di fibonacci, va al di là del corso.
- Si segnala solo l'ibridizzazione che consiste nell'utilizzare un generatore congruenziale in cui periocamente si cambia il seme con un numero veramente casuale. Questo stesso stratagemma può essere utilizzato con tutti i generatori di numeri pseudo-casuali.

Riassunto: Cifrario Sostituzionale

- La cifratura sostituzionale si viola tramite l'analisi delle frequenze. Un attacco al solo testo cifrato (cyphertext only attack) è sufficiente. I messaggi devono essere lunghi ed essere composti da elementi (ad esempio i caratteri) ripetitivi.

Riassunto: Cifrario Sostituzionale

- La cifratura sostituzionale si viola tramite l'analisi delle frequenze. Un attacco al solo testo cifrato (cyphertext only attack) è sufficiente. I messaggi devono essere lunghi ed essere composti da elementi (ad esempio i caratteri) ripetitivi.
- Non è un cifrario ideale perché dal solo crittogramma si può dedurre la chiave.

Riassunto: Cifrario Sostituzionale

- La cifratura sostituzionale si viola tramite l'analisi delle frequenze. Un attacco al solo testo cifrato (cyphertext only attack) è sufficiente. I messaggi devono essere lunghi ed essere composti da elementi (ad esempio i caratteri) ripetitivi.
- Non è un cifrario ideale perché dal solo crittogramma si può dedurre la chiave.
- Se si usa una cifratura polialfabetica bilanciata (in cui ad un carattere corrispondono più caratteri cifrati) si può rendere uniforme la distribuzione dei caratteri cifrati, ma il crittogramma diviene molto più lungo del messaggio originale.

Riassunto: Cifrari alla Vernam

- Se si esegue un o esclusivo (o aut o xor) con una sequenza random lunga quanto il messaggio (OTP), si ottiene un cifrario perfetto:

$$c = f(m) \oplus k$$

in cui il crittogramma possiede una distribuzione uniforme, qualunque sia la trasformazione f e la struttura dei messaggi.

Riassunto: Cifrari alla Vernam

- Se si esegue un o esclusivo (o aut o xor) con una sequenza random lunga quanto il messaggio (OTP), si ottiene un cifrario perfetto:

$$c = f(m) \oplus k$$

in cui il crittogramma possiede una distribuzione uniforme, qualunque sia la trasformazione f e la struttura dei messaggi.

- Se si riusa la stessa chiave il cifrario non è più perfetto. In un cifrario a blocchi in cui si usa sempre la stessa chiave:

$$c_i = f(m_i) \oplus k$$

basta considerare la somma dei messaggi con l'o esclusivo per far sparire la chiave:

$$c_i \oplus c_j = (f(m_i) \oplus k) \oplus (f(m_j) \oplus k) = f(m_i) \oplus f(m_j)$$

quindi basta studiare la regolarità di $f(m_i) \oplus f(m_j)$. Si chiama la crittografia differenziale.

Riassunto: Cifrari a blocchi trasposizionali

- In questo caso se si esegue una sola trasposizione globale (una permutazione) su tutto un messaggio si ottiene una cifratura ideale.

Riassunto: Cifrari a blocchi trasposizionali

- In questo caso se si esegue una sola trasposizione globale (una permutazione) su tutto un messaggio si ottiene una cifratura ideale.
- Se si usa la stessa permutazione in un cifrario a blocchi, il cifrario è vulnerabile rispetto ad un attacco a testi noti casuali (known random plaintext attack). Se si dispone di un numero sufficiente di coppie casuali testo in chiaro-crittogramma si scopre sempre la chiave.

Riassunto: Cifrari a blocchi trasposizionali

- In questo caso se si esegue una sola trasposizione globale (una permutazione) su tutto un messaggio si ottiene una cifratura ideale.
- Se si usa la stessa permutazione in un cifrario a blocchi, il cifrario è vulnerabile rispetto ad un attacco a testi noti casuali (known random plaintext attack). Se si dispone di un numero sufficiente di coppie casuali testo in chiaro-crittogramma si scopre sempre la chiave.
- Se si possono scegliere i messaggi in chiaro si scopre la chiave in un numero minore di passi (chosen plaintext attack).

Riassunto: Composizioni di permutazioni e o aut

- Un cifrario a blocchi in cui si esegue un qualsiasi numero di permutazioni e o esclusivi (con varie chiavi) equivale ad un altro in cui si esegue una sola permutazione ed un solo o esclusivo (con una sola chiave).

Riassunto: Composizioni di permutazioni e o aut

- Un cifrario a blocchi in cui si esegue un qualsiasi numero di permutazioni e o esclusivi (con varie chiavi) equivale ad un altro in cui si esegue una sola permutazione ed un solo o esclusivo (con una sola chiave).
- Questi cifrari si attaccano con le tecniche della crittografia differenziale (per eliminare l'o esclusivo) e l'analisi della regolarità

Riassunto: Composizioni di permutazioni e o aut

- Un cifrario a blocchi in cui si esegue un qualsiasi numero di permutazioni e o esclusivi (con varie chiavi) equivale ad un altro in cui si esegue una sola permutazione ed un solo o esclusivo (con una sola chiave).
- Questi cifrari si attaccano con le tecniche della crittografia differenziale (per eliminare l'o esclusivo) e l'analisi della regolarità
- Anche questi cifrari sono attaccabili tramite attacchi a testi noti casuali.

Riassunto: Cifratura alla Feistel

- Il modulo di Feistel introduce delle trasformazioni che non commutano con l'o esclusivo.

Riassunto: Cifratura alla Feistel

- Il modulo di Feistel introduce delle trasformazioni che non commutano con l'o esclusivo.
- La reiterazione di operazioni elementari alla Feistel con permutazioni e o esclusivi con chiavi produce la cifratura all Feistel.

Riassunto: Cifratura alla Feistel

- Il modulo di Feistel introduce delle trasformazioni che non commutano con l'o esclusivo.
- La reiterazione di operazioni elementari alla Feistel con permutazioni e o esclusivi con chiavi produce la cifratura all Feistel.
- Non si conoscono metodi crittoanalitici per violare la cifratura alla Feistel

Riassunto: Cifratura alla Feistel

- Il modulo di Feistel introduce delle trasformazioni che non commutano con l'o esclusivo.
- La reiterazione di operazioni elementari alla Feistel con permutazioni e o esclusivi con chiavi produce la cifratura all Feistel.
- Non si conoscono metodi crittoanalitici per violare la cifratura alla Feistel
- Le cifrature a chiave simmetrica attualmente in uso sono tutte alla Feistel pricipalmente AES e Triple DES

Messaggio

- Abbiamo visto che i cifrari puramente trasposizionali a blocchi si violano sempre al crescere della lunghezza del messaggio se si dispone delle coppie messaggio in chiaro-crittpgramma.

Messaggio

- Abbiamo visto che i cifrari puramente trasposizionali a blocchi si violano sempre al crescere della lunghezza del messaggio se si dispone delle coppie messaggio in chiaro-crittpgramma.
- Abbiamo visto un esempio di cifrario perfetto: One Time Pad (OTP) o cifrario di Wernam (cifratura "usa e getta"). Un altro cifrario perfetto è quello di Augusto finché non si riutilizzano gli stessi numeri.

Messaggio

- Abbiamo visto che i cifrari puramente trasposizionali a blocchi si violano sempre al crescere della lunghezza del messaggio se si dispone delle coppie messaggio in chiaro-crittpgramma.
- Abbiamo visto un esempio di cifrario perfetto: One Time Pad (OTP) o cifrario di Wernam (cifratura "usa e getta"). Un altro cifrario perfetto è quello di Augusto finché non si riutilizzano gli stessi numeri.
- Utilizzando il cifrario OTP i crittogrammi non sono decrittabili con metodi crittoanalitici e il codice è inviolabile perché la chiave viene usata una sola volta e dunque anche scoprendola non si ottiene alcuna informazione sulle sequenze future cifrate.

Messaggio

- Abbiamo visto che i cifrari puramente trasposizionali a blocchi si violano sempre al crescere della lunghezza del messaggio se si dispone delle coppie messaggio in chiaro-crittpgramma.
- Abbiamo visto un esempio di cifrario perfetto: One Time Pad (OTP) o cifrario di Wernam (cifratura "usa e getta"). Un altro cifrario perfetto è quello di Augusto finché non si riutilizzano gli stessi numeri.
- Utilizzando il cifrario OTP i crittogrammi non sono decrittabili con metodi crittoanalitici e il codice è inviolabile perché la chiave viene usata una sola volta e dunque anche scoprendola non si ottiene alcuna informazione sulle sequenze future cifrate.
- Affinché il codice funzioni sono necessarie due condizioni: imprevedibilità (distribuzione casuale uniforme) della chiave e conoscenza (condivisone) delle stessa da parte di tutti i soggetti autorizzati.

Messaggio

- Abbiamo visto che i cifrari puramente trasposizionali a blocchi si violano sempre al crescere della lunghezza del messaggio se si dispone delle coppie messaggio in chiaro-crittpgramma.
- Abbiamo visto un esempio di cifrario perfetto: One Time Pad (OTP) o cifrario di Wernam (cifratura "usa e getta"). Un altro cifrario perfetto è quello di Augusto finché non si riutilizzano gli stessi numeri.
- Utilizzando il cifrario OTP i crittogrammi non sono decrittabili con metodi crittoanalitici e il codice è inviolabile perché la chiave viene usata una sola volta e dunque anche scoprendola non si ottiene alcuna informazione sulle sequenze future cifrate.
- Affinché il codice funzioni sono necessarie due condizioni: imprevedibilità (distribuzione casuale uniforme) della chiave e conoscenza (condivisone) delle stessa da parte di tutti i soggetti autorizzati.
- Il Mersenne Twister consente la generazione di numeri pseudo-casuali utilizzabili anche come succedanei dei numeri

Ridondanza ciclica

- Trovare il digest del messaggio 11010101 con polinomi generatore $g(x) = x^2 + 1$

Ridondanza ciclica

- Trovare il digest del messaggio 11010101 con polinomi generatore $g(x) = x^2 + 1$
- 11010101-¿ 11010101 00

Ridondanza ciclica

- Trovare il digest del messaggio 11010101 con polinomi generatore $g(x) = x^2 + 1$
- 11010101-¿ 11010101 00
- $x^2 + 1$ -¿ 101

Ridondanza ciclica

- Trovare il digest del messaggio 11010101 con polinomi generatore $g(x) = x^2 + 1$
- 11010101-¿ 11010101 00
- $x^2 + 1$ -¿ 101
- Il resto di 11010101 è 10
- 101 è divisore esatto di 11010101 10

Generatori random Congruenziali

- $n = 2^{64}$, trovare a e b per un generatore congruenziale. Es. $a = 5,\ b = 1$

Generatori random Congruenziali

- $n = 2^{64}$, trovare a e b per un generatore congruenziale. Es. $a = 5$, $b = 1$
- $n = 2^6 = 64$ calcolare una serie completa generata con $a_0 = 2$. Verificare che copre tutti i valori.

Generatori random Congruenziali

- $n = 2^{64}$, trovare a e b per un generatore congruenziale. Es. $a = 5$, $b = 1$
- $n = 2^6 = 64$ calcolare una serie completa generata con $a_0 = 2$. Verificare che copre tutti i valori.
- Che succede se si sceglie $b = 2$? e se si sceglie $b = 1$ e $a = 3$?

Generatori random Congruenziali

- $n = 2^{64}$, trovare a e b per un generatore congruenziale. Es. $a = 5,\ b = 1$
- $n = 2^6 = 64$ calcolare una serie completa generata con $a_0 = 2$. Verificare che copre tutti i valori.
- Che succede se si sceglie $b = 2$? e se si sceglie $b = 1$ e $a = 3$?
- Travare un generatore congruenziale per $n = 49$: Sol: $a = 8 = 7 + 1, 15, 22, \ldots$ $b = 1, 3, 5, ..$

Generatori di anelli ciclici

- in $\mathbb{Z}_n^*$,con $n = 81$ trovare un generatore. Sol. $g = 2$ è generatore con p=3 ($81 = p^4 = 3^4$).

Generatori di anelli ciclici

- in $\mathbb{Z}_n^*$,con $n = 81$ trovare un generatore. Sol. $g = 2$ è generatore con p=3 ($81 = p^4 = 3^4$).
- Con $n = 343$? Sol $g = 3$ è generatore con p=7 ($343 = p^3 = 7^3$). 2 non è generatore.

Generatori di anelli ciclici

- in $\mathbb{Z}_n^*$,con $n = 81$ trovare un generatore. Sol. $g = 2$ è generatore con p=3 ($81 = p^4 = 3^4$).
- Con $n = 343$? Sol $g = 3$ è generatore con p=7 ($343 = p^3 = 7^3$). 2 non è generatore.
- Trovare tutti i generatori di $\mathbb{Z}_n^*$ con $n = 49$. Sol sono le potenze del 3 prime con $\phi(49) = 7 * 6$: 3,3^5, 3^7 ... Sono 12 generatori in tutto.

Esercizi Entropia

- Due loci con due alleli aA bB. Dati P(a,b), p(a,B), pA,b), pA,B).

Esercizi Entropia

- Due loci con due alleli aA bB. Dati P(a,b), p(a,B), pA,b), pA,B).
- Trovare distribuzioni marginal, entropie, entropia totale mutua informazione.

Esercizi Entropia

- Due loci con due alleli aA bB. Dati P(a,b), p(a,B), pA,b), pA,B).
- Trovare distribuzioni marginal, entropie, entropia totale mutua informazione.
- Sono loci indipendenti o c'è un "linkage"?