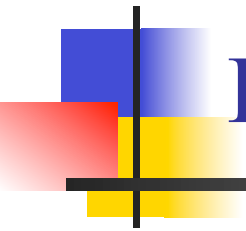
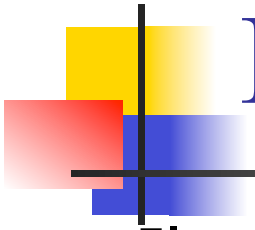


Corso di Crittografia

Prof. Dario Catalano

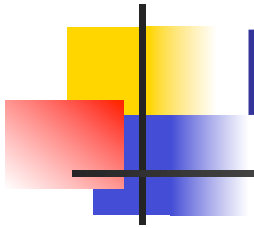
Funzioni e Permutazioni Pseudo-casuali: Applicazioni ai Cifrari a Blocchi



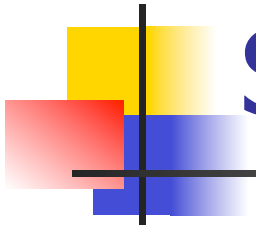
Introduzione

- Il motivo per cui abbiamo introdotto funzioni (PRF) e permutazioni (PRP) pseudo casuali e' per potere studiare e analizzare adeguatamente I cifrari a blocchi.
- Inizialmente, ci siamo occupati di un tipo di sicurezza piuttosto debole: la resistenza agli attacchi di tipo key recovery.
- Abbiamo visto che tale livello di sicurezza, se certamente e' necessario, non puo' definirsi sufficiente in pratica.

La proprieta' universale

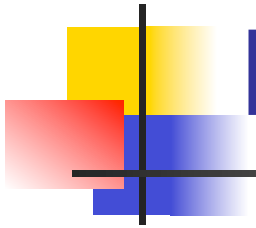


- Vogliamo una proprieta' "universale" che se soddisfatta, possa permetterci di dichiarare un dato cifrario a blocchi sicuro.
- Oggi candideremo a tale ruolo la proprieta' di essere una PRP sicura rispetto ad attacchi di tipo CPA o CCA.



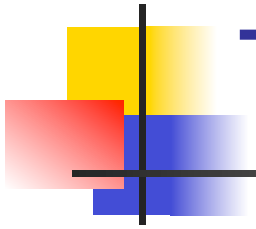
Sfortunatamente…

- Non possiamo dimostrare che un determinato cifrario E abbia tale proprieta'.
- Possiamo *assumere* che E abbia tale proprieta' e procedere sulla base di tale assunto.



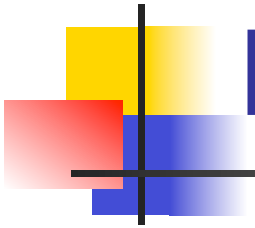
Precisazioni

- La precedente e' solo una congettura
- Ad es. potrebbe esistere un attacco che rompe AES (o DES) come PRF senza riuscire a trovare la chiave.
- Non conosciamo alcun attacco di questo tipo, ma non moltissima attenzione e' stata rivolta a questo problema.
- Ipotizzare che un cifrario a blocchi sia una PRF e' una assunzione MOLTO piu' forte rispetto ad assumerlo sicuro contro key recovery.



Tuttavia…

- Le motivazioni che abbiamo illustrato in favore di PRF e PRP rimangono.
- Se un dato cifrario dovesse rivelarsi insicuro come PRF, allora sarebbe bene considerarlo insicuro in pratica (e cambiarlo quanto prima)



Esempi di Attacchi

- Prima di procedere ulteriormente soffermiamoci sulle definizioni studiate le scorsa volta.
- In particolare cerchiamo di capire meglio questo concetto di "indistinguibilita' tra mondi" guardando ad un semplice esempio.

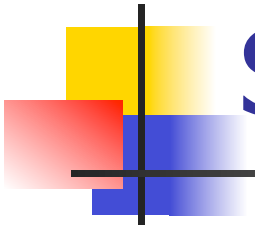
Definizione

- F: K × D → R, famiglia di funzioni (pubblica)

Esp_F^{prf-1} (A)	Esp_F^{prf-0} (A)
$K \leftarrow_R K$	$g \leftarrow_R Func(D,R)$
$b \leftarrow A^{F_K}$	$b \leftarrow A^g$
Return b	Return b
// Funz. PseudoCasuale	// Funz. Casuale

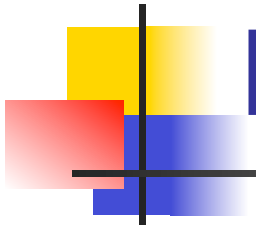
$$Adv(A)=|Pr[Esp_F^{prf-1}(A)=1] - Pr[Esp_F^{prf-0}(A)=1]|$$

- La prf e' sicura se tale vantaggio e' "piccolo"



Sicurezza contro Key Recovery

- Abbiamo visto che questo livello di sicurezza non e' sufficiente in pratica (pur essendo necessario)
- Un cifrario a blocchi per essere sicuro dovrebbe comportarsi come una prf (o prp)
- E' giunto il momento di cominciare a verificare questo fatto



Formalizziamo meglio

- Cosa intendiamo per sicurezza contro attacchi di tipo key recovery?
- L'avversario B ha a disposizione un certo num di coppie (M,C) e deve trovare la chiave.
- Definiamo un adeguato esperimento in cui il vantaggio di B e' la sua probabilita' di trovare la chiave.

Definizione

- F: K × D → R, famiglia di funzioni.

Esp_F^{kr} (B)

$K \leftarrow_R K$

$K' \leftarrow B^{F_K}$

If K'=K Return 1 else return 0

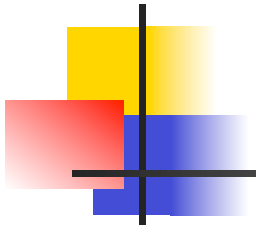
$$Adv(B)=Pr[Esp_F^{kr}(B)=1]$$

- La funzione e' sicura se tale vantaggio e' "piccolo"

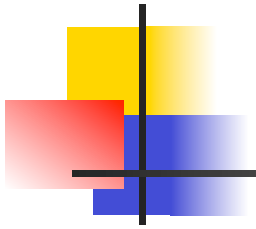
Formalmente

- F: K × D → R, famiglia di funzioni (qualunque!).
- B avvers. key recovery, che lavora in tempo t e fa (al piu') q domande.
- Esiste A avv. prf che lavora in tempo t e fa al piu' (q+1) domande

$$Adv(B) \leq Adv(A) + 1/|R|$$



- Il teorema dimostra che se F e' una famiglia di funzioni sicura in senso prf allora F e' sicura contro ogni avversario (limitato) di tipo key recovery.



- L’avversario A deve determinare se una data funzione g appartiene alla famiglia F (pseudorandom) o e’ una funzione casuale da D a R.
- A ha la possibilita’ di “utilizzare” l’altro avversario B.
- Deve essere in grado di sfruttare le capacita’ di B, per risolvere il suo problema.

Descrizione dell'avversario A

i=0;
Run B
when B asks for x do
{ i++; $x_i = x$;
$y_i = \mathcal{O}^A(x_i)$;
Return y_i to B; }
until B stops and outputs k'
Let $x \leftarrow_R D - \{x_1, \ldots, x_q\}$
$y=\mathcal{O}^A(x)$;
If (F(k',x)==y) return 1
else return 0

Definizione

- F: K × D → R, famiglia di funzioni (pubblica)

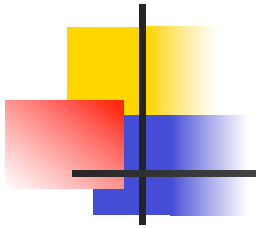
Esp_F^{prf-1} (A)	Esp_F^{prf-0} (A)
$K \leftarrow_R K$	$g \leftarrow_R Func(D,R)$
$b \leftarrow A^{F_K}$	$b \leftarrow A^g$
Return b	Return b
// Funz. PseudoCasuale	// Funz. Casuale

$$Adv(A)=|Pr[Esp_F^{prf-1}(A)=1] - Pr[Esp_F^{prf-0}(A)=1]|$$

- La prf e' sicura se tale vantaggio e' "piccolo"

L'attacco del compleanno

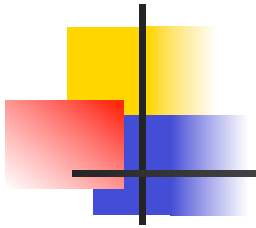
- Sia $E : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ una famiglia di perm (es cifrario a blocchi)
- Supponiamo di ricevere accesso (black box) una funzione g che e' o un'istanza di E o una funzione casuale.
- Come distinguere i due casi?
  - Invochiamo la funzione su q punti, se g e' una perm gli output devono essere diversi.
  - Se g e' una funzione tale condizione puo' non essere vera.



- Sorprendentemente tale strategia non e' male.
- Bastano (piu' o meno) $q = \sqrt{2^\ell}$ domande per avere un vantaggio prossimo a 1.
- Tale fenomeno e' dovuto al cosiddetto "paradosso del compleanno"

Paradosso del compleanno

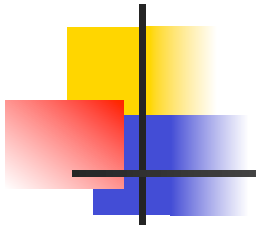
- Supponiamo che i compleanni siano uniformemente distribuiti sui giorni dell'anno.
- Quale e' la probabilita che in un gruppo di q persone vi siano almeno 2 individui che fanno il compleanno nello stesso giorno?
- Sorprendentemente se $q \approx \sqrt{365}$, tale probabilita' e' piuttosto alta (circa ½)



- Supponiamo di avere q palline e N buche.
- Se vi sono troppe palline, le collisioni sono inevitabili.
- Sia D_i l'evento che non vi e' alcuna collisione dopo aver tirato l'i-esima pallina

$$\Pr[D_1]=1 \quad \Pr[D_{i+1} \mid D_i]=(N-i)/N$$

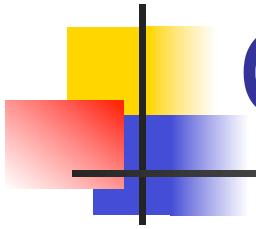
- Dobbiamo valutare $\Pr[D_q]$



$$e^{-x} = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} \cdots$$

Dunque

$$1 - x \leq e^{-x}$$



Conseguenza

- Dunque un cifrario a blocchi puo' essere distinto da una funzione casuale guardando un numero $2^{\ell/2}$ coppie input-output.