Schemi di crittografia a chiave pubblica

Schemi di identificazione: Protocolli a conoscenza zero

- FIAT-SHAMIR
- SCHNORR

{ × fattorizzazione
× radici quadrate
× log. discreti
(ecc..., colorazione dei grafi, ecc..) }

challenge-response senza rivelazione del segreto

Cena dei crittografi

Protocollo per verificare se paga NSA o uno dei convitati (anonimo)

Condivisione del segreto (secret splitting & sharing)

$n \le N$: N individui condividono un segreto. Servono almeno n su N per ricostruire il segreto. Se $k<n$ non è possibile

- Schema a soglia di Shamir (interpolatore di Lagrange)
- Schema di BLAKLEY

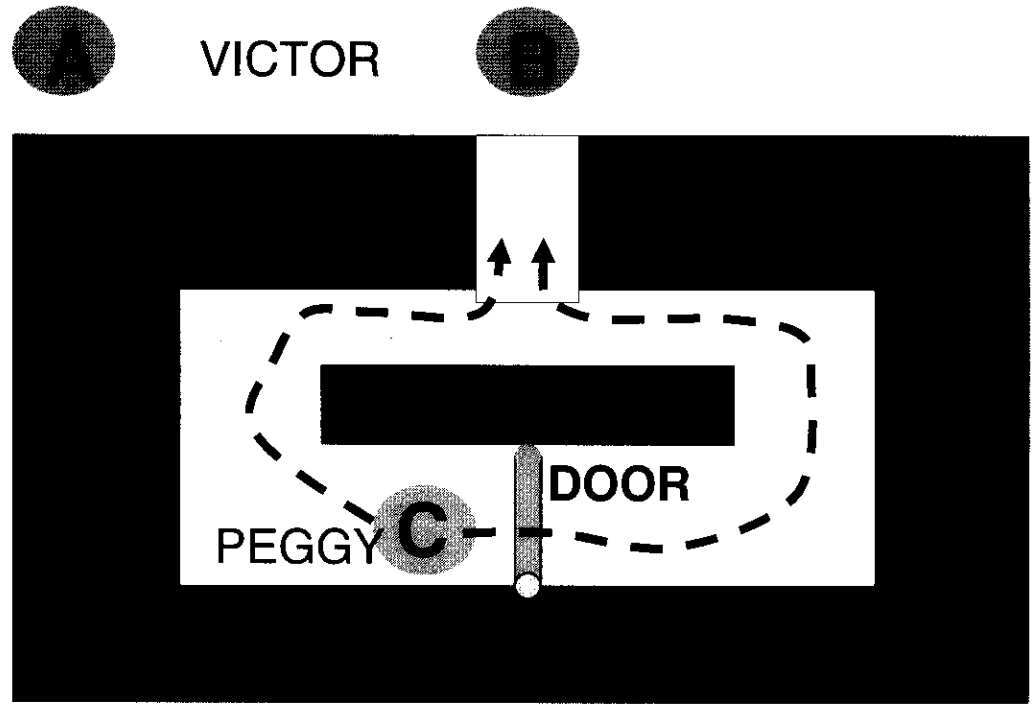
Protocolli a Conoscenza-zero - Zero-Knowledge Protocols

The Cave: Quisquater et alii

PEGGY claims she has the key of the locked door in the cave. VICTOR stands in A as Peggy enters the cave and hides at C, then he moves to B and asks Peggy to come out at random from either the left or the rigth side of the tunnel.
VICTOR goes back to A and repeates the test *m* times.

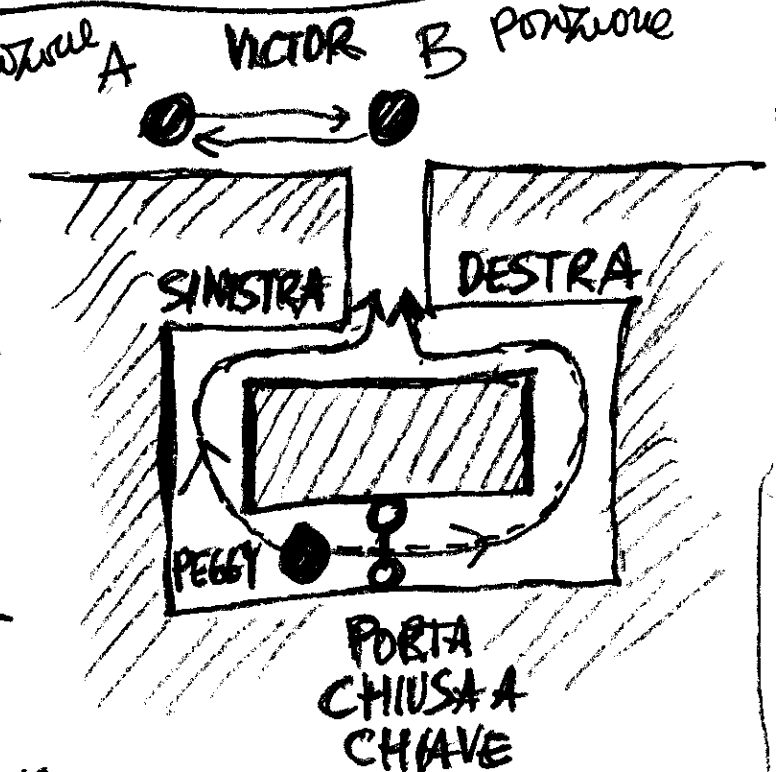
Given that Peggy is cheating (has no key and in each test is lucky in hiding at the requested side of the tunnel) the probability of *m* consecutive positive tests is $1/2^m$. For m=10 or 20, this gives 10^{-3} or 10^{-6}, respectively.

VICTOR, VIVALES = Verifier
PEGGY, PICARA = Prover



Protocolli di verifica della conoscenza o possesso di un segreto, senza scambi di informazione che consentano la rivelazione del segreto stesso nello scambio (anche se crittografato)-

ZERO-KNOWLEDGE PROTOCOLS



LA PROVA DELLA CAVA A CONOSCENZA ZERO

VICTOR ≡ VERIFIER

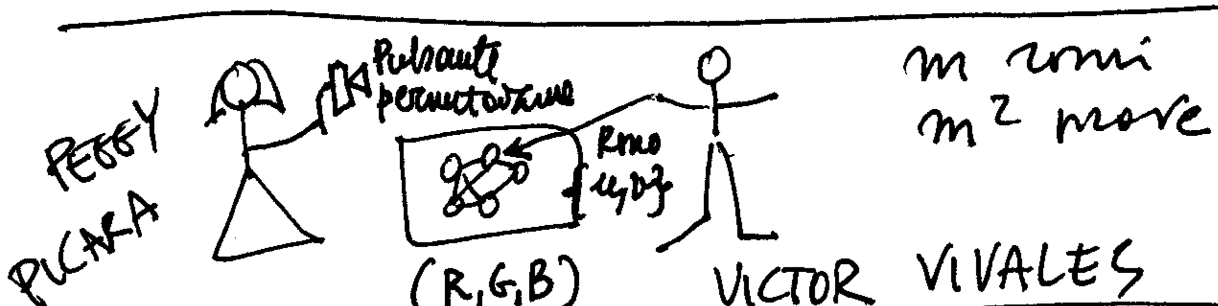
PEGGY ≡ PROVER

Peggy vorrebbe di poterle la chiave della porta

Per verificare Victor esegue le prove. Victor è in A e chiede a Peggy di entrare nella cava. Victor si sposta in B e chiede a caso a Peggy di uscire dal lato sinistro o destro del tunnel. Se Peggy ha la chiave tutte le prove vanno a buon fine. Se Peggy bara e non ha la chiave e si trova a caso a sinistra o a destra da ogni parte, allora dopo k prove positive, $P\{\text{Peggy non ha chiave}\} \approx \frac{1}{2^k}$ $\approx 10^{-3}$ Per $k=10$ $\approx 10^{-7}$ Per $k=20$

ZERO-KNOWLEDGE CON GRAFI COLORATI

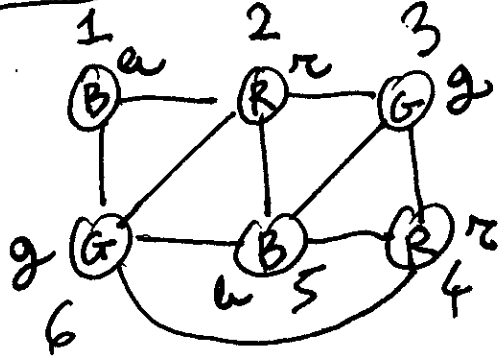
Peggy ha una macchina di simulazione del grafo con 3 lampadine per nodo e accende il colore attribuitole. Una volta trovata la colorazione corretta, Peggy può permutare i 3 colori con un pulsante.
Le lampadine sono accese ma invisibili. Quando Victor tocca un ramo scelto a caso si accendono i due vertici, Victor verifica se sono dello stesso colore o no.



Peggy pretende di aver trovato la funzione $f(i)$ $1 \le i \le n$ di colorazione a 3 colori del grafo G con n vertici e m rami. Victor sfida Peggy scegliendo i rami di volta in volta e verificando i colori dei due vertici.
Ad ogni prova Peggy permuta i tre colori per evitare di mandare informazioni su $f(i)$ $\forall i$.

VIVALES e PICARA = VICTOR e PEGGY

$G \equiv (V, E)$ (1)



RGB (1) π_1

$V = \{1, 2, 3, 4, 5, 6\}$ $|V| = n = 6$

$E = \{u_i, v_i\}\ \forall i, j \in V$ $|E| = m = 10$

3 colori R, G, B Red, Green, Blu

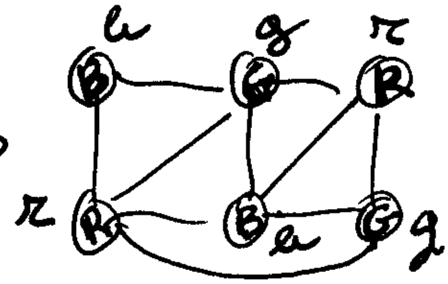
è banale nel fatto che f esiste per tutte le 3! permutazioni RGB.

la funzione $f = \{B, R, G, R, B, G\}$ colora il grafo

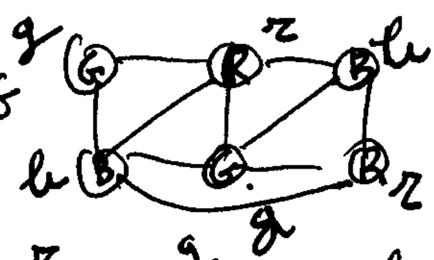
in modo che per ogni $\forall i, j : f(i) \neq f(j),\ i \neq j$

6 permutazioni RGB, GRB, RBG, GBR, BRG, BGR.

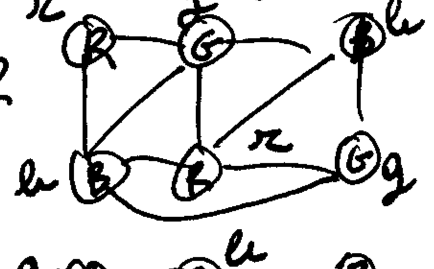
GRB (2) π_2



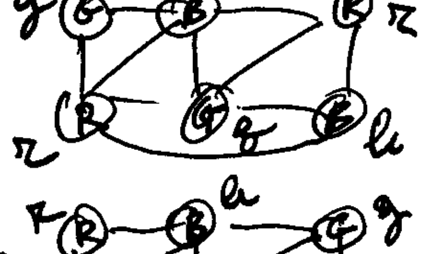
RBG (3) π_3



GBR π_4



BRG π_5



BGR π_6

volta per volta Peggy permuta a caso senza ripetizione e fissa i colori

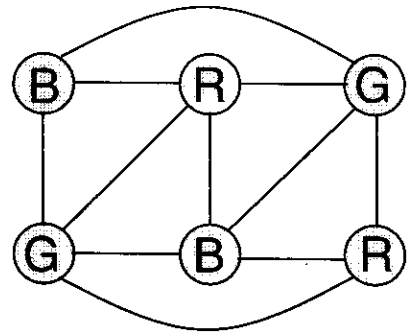
Victor	Peggy permuta prova	i colori per sequenza uno alla volta (max m volte)
{1,2}	π_1	B — R
1,6	π_3	G — B
2,6	π_4	G — B
2,3	π_5	B — R
2,5	π_2	G — B
3,5	π_6	G — R
3,4	π_3	B — R
4,5	π_4	G — R
4,6	π_2	R — B
{5,6}	π_1	B — G
{1,2}	π_6	R — B
{1,6}	π_5	G — R
{2,6}	π_4	

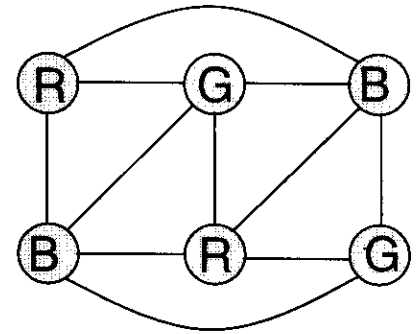
RIPETE

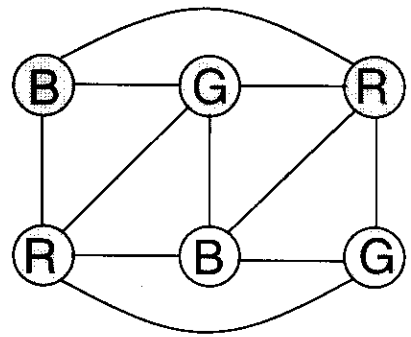
Victor ripete m^2 volte prima di rispondere

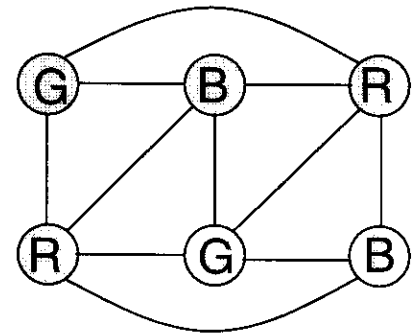
Zero-Knowledge Proof on 3-Colour Graphs

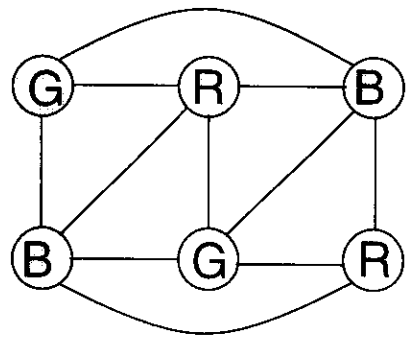
1

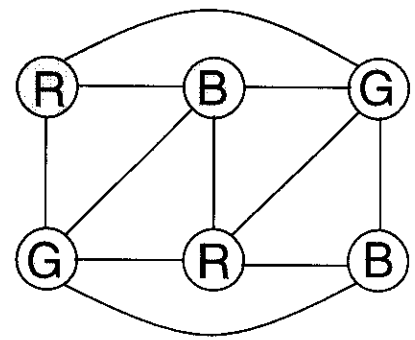












SCHEMA DI IDENTIFICAZIONE FEIGE-FIAT-SHAMIR

$n = p \cdot q$ primi grandi. Peggy ha i numeri segreti $s_1, s_2, \dots s_k$

Sia $v_i = s_i^{-1} \pmod n$; $\text{mcd}(s_i, n) = 1$

I numeri v_i sono variabili a Victor

Victor cerca di verificare se Peggy conosce i numeri $s_1, s_2 \dots s_k$.

1. Peggy sceglie l'intero casuale r, calcola $x \equiv r^2 \pmod n$ e lo manda a Victor
2. Victor manda il numero binario $b_1, b_2 \dots b_k$ $(b_i \in \mathbb{Z}_2)$ a Peggy
3. Peggy calcola $y = r\, s_1^{b_1} s_2^{b_2} \dots s_k^{b_k} \pmod n$ e lo manda a Victor.

4. Victon verifica che

$$x \equiv y^2 \, v_1^{b_1} v_2^{b_2} \dots v_k^{b_k} \pmod n$$

5. I passi 1÷4 sono ripetuti ogni volta con un r differente

Se $k=1$ si incade nel caso precedente qual'è la radice quadrata r o $r s_1$. (prima era r_i o r_1's).

Se $k>1$, Peggy deve fornire il numero $y \equiv r \prod_{i=1}^{k} b_i s_i \quad (b_i \in \mathbb{Z}_2)$

che è la radice quadrata di

$$x \prod_{i=1}^{k} b_i v_i \qquad (b_i \in \mathbb{Z}_2)$$

---

ict CONSULTING

Protocollo a conoscenza-zero basato sulla fattorizzazione $n = p \cdot q$ con p e q primi grandi.

$p \equiv q \equiv 3 \pmod 4$

Victor sceglie un intero casuale $x \bmod n$ e calcola

$$y \equiv x^2 \pmod n$$

e lo manda a Peggy.

Peggy conosce la fattorizzazione di $n = p \cdot q$ e calcola le radici quadrate di $y \bmod p$ e di $y \bmod q$, poi le combina con CRT e ottiene una radice quadrata di $y \bmod n$. Peggy sfrutta $p \equiv q \equiv 3 \pmod 4$. La radice quadrata di y è s e Peggy la manda a Victor.

Victor verifica che $s^2 \equiv y \pmod n$ e ripete la prova per m volte ($m = 20$) scegliendo il numero casuale x ogni prova.

Oscar vede una lista di interi s e il loro quadrato $y \bmod n$, ma riesce a fattorizzare n.

Purtroppo questo NON è un problema di crittografia perché Victor può usare la procedura per fattorizzare n (infatti ci sono 4 radici quadrate di y mod n). Dopo un certo numero di iterazioni Peggy manda un probabile un intero s con $s \neq \pm x \pmod n$. Poiché Victor (e un Oscar!) potrebbe conoscere x e s, può calcolare $\mathrm{mcd}(x-s, n)$ e trovare un fattore non triviale di n.

TECNICHE A CONOSCENZA ZERO

ZERO-KNOWLEDGE TECHNIQUES

Impiego delle radici quadrate

Sia $n = p \cdot q$ prodotto di due primi grandi

ma $y \equiv s^2 \pmod n$ con $mcd(y, n) = 1$

trovare $s \equiv \sqrt{y} \bmod n$ è computazionalmente difficile come fattorizzare n

Peggy pretende di conoscere una radice quadrata s di y. Victor vuole verificare se è vero, ma Peggy non vuole rivelare s. Ecco il metodo di verifica della conoscenza di Peggy senza che riveli l'oggetto della conoscenza (una radice quadrata s di y)

1. Peggy sceglie un intero casuale r_1 e ma $r_2 \equiv s r_1^{-1} \pmod n$

in modo tale che

$$r_1 r_2 \equiv s \pmod n$$

Calcola $x_1 \equiv r_1^2$; $x_2 \equiv r_2^2 \pmod n$

e manda X_1, X_2 a Victor

2. Victor verifica che $X_1 \cdot X_2 \equiv Y \pmod n$ poi sceglie a caso X_1 oppure X_2 e chiede a Peggy di fornirgli la radice quadrata. Victor verifica velocemente se Peggy ha fornito il numero giusto, elevandolo al quadrato.

3. I passi 1 e 2 vengono ripetuti m volte finché Victor si convince che Peggy conosce una $s \equiv \sqrt{y} \pmod n$ ad ogni nuova Peggy cambia x_1 e x_2.

Se Peggy conosce s allora tutte le m prove sono convincenti
Se Peggy non conosce s, allora lei manda due numeri X_1 e X_2 tali che $X_1 \cdot X_2 \equiv Y$
Se conosce la radice quadrata di entrambe, allora conosce quella di y e cioè s. Quindi supponiamo che conosca la radice di X_1 e non quella di X_2.

Se Victor sceglie x_1 e x_2 in modo equiprobabile, allora al 50% Victor domanda una radice quadrata che Peggy non conosce e dopo un po' di prove su (su piccolo) Victor si convince che Peggy non conosce s, radice quadrata di y.

Il sistema di "identificazione" di Peggy (solo lei conosce la risposta alla sfida y - challenge: qual'è la radice quadrata di y (mod n)? - risposte: s.) è sicuro dalle intrusioni e a ogni prova Peggy conosce r_1 e r_2 - Oscar può conoscere di prova in prova una radice di un quadrato casuale (r_i^2) e non una radice di y.

Protocollo a conoscenza-zero basato sui logaritmi discreti

p, primo grande, α radice primitiva e

$$\beta \equiv \alpha^a \pmod p \qquad 1 \le a < p-1$$

PUBLIC: p, α, β

Peggy vuole provare a Victor che lei conosce a, senza rivelarla.

1. Peggy sceglie un numero casuale $r \pmod{p-1}$
2. Peggy calcola
$$h_1 \equiv \alpha^r \pmod p$$
$$h_2 \equiv \alpha^{a-r} \pmod p$$
e manda (h_1, h_2) a Victor
3. Victor sceglie $i = 1$ o $i = 2$ e chiede a Peggy di mandare o $r_1 = r$ oppure $r_2 = a - r \pmod{p-1}$
4. Victor verifica che $h_1 h_2 \equiv \beta \pmod p$
e che $h_i \equiv \alpha^{r_i} \pmod p$, $i = 1, 2$

La procedura 1-4 è ripetuta più volte finché Victor non è convinto che Peggy conosce a.

SCHEMA DI IDENTIFICAZIONE DI SCHNORR

basato sui logaritmi discreti.

p, primo grande, α radice primitiva,

$$\beta \equiv \alpha^a \pmod p \qquad 1 \le a < p-1$$

p, α, β - PUBBLICI

Peggy vuole provare a Victor che la conosce a suo vantaggio.

1. Peggy sceglie k $1 \le k < p-1$ e calcola
$$\gamma \equiv \alpha^k \pmod p$$
e lo manda a Victor

2. Victor sceglie r $1 \le r < p-1$ e lo manda a Peggy

3. Peggy calcola $y \equiv k - ar \pmod{p-1}$
e lo manda a Victor

4. Victor verifica se $\gamma \equiv \alpha^y \beta^r \pmod p$
se sì, Peggy conosce a. Infatti:
$$\alpha^y \cdot \beta^r \equiv \alpha^{k-ar} \cdot \alpha^{ar} \equiv \alpha^k \equiv \gamma.$$

Protocollo a conoscenza zero basato su RSA

Peggy pretende di conoscere il plaintext P corrispondente ad un ciphertext C cifrato con RSA. E cioè "n", "e", e "C" sono pubblici e Peggy pretende di conoscere P tale che

$$C \equiv P^e \pmod{n}$$

essendo $n = p \cdot q$, primi grandi.

Il protocollo di verifica zero-knowledge è

1. Peggy sceglie l'intero casuale r_1 e calcola $r_2 \equiv P \, r_1^{-1} \pmod{n}$

   essendo $\mathrm{mcd}(r_1, n) = 1$.

2. Peggy calcola $X_1 \equiv r_1^e \pmod{n}$

   $X_2 \equiv r_2^e \pmod{n}$

   e manda X_1 e X_2 a Victor

3. Victor verifica che $X_1 \cdot X_2 \equiv C \pmod{n}$

4. Victor sceglie a caso $i = 1$ o $i = 2$ e

chiede a Peggy la radice quadrata di x_i, e cioè r_i $(i = 1, 2)$

5. Victor verifica che la risposta di Peggy sia corretta

$$x_i \equiv r_i^2 \pmod{n}$$

per $i = 1$ e 2

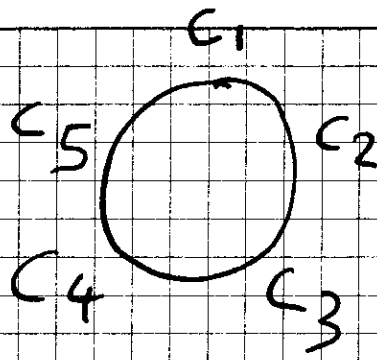
Se il protocollo è fatto 1-5 per m volte allora Victor si convince che Peggy conosce la risposta.

Supponiamo che Peggy non conosca la fattorizzazione di $n = p \cdot q$ non può calcolare le radici quadrate e risponde a caso. La probabilità di successo di Peggy, che non conosce $n = p \cdot q$, e quindi di, rispondere nella risposta a caso sia $1/2$, allora dopo m prove la probabilità di successo nella risposta è $P = \left(\frac{1}{2}\right)^m$ se $m = 7$ $P = 10^{-2}$

ROUND TABLE (mod n)

La cena dei crittografi

A cena ci sono n crittografi (n dispari)



$n = 5$ dispari

ogni crittografo forma due coppie con i due vicini di posto; quindi ci sono $2n$ coppie

Nel caso $n=5$ le coppie sono

C_1	C_2	C_3	C_4	C_5
(C_1, C_2)	(C_2, C_3)	(C_3, C_4)	(C_4, C_5)	(C_5, C_1)
(C_1, C_5)	(C_2, C_1)	(C_3, C_2)	(C_4, C_3)	(C_5, C_4)

le coppie sono 10.

$$N(h,k) \equiv (C_h, C_k)$$

se $\forall\, h, k$ per cui $(1 \le h, k \le n)$

$$h - k = \pm 1 .$$

ora numero le coppie di C_i

$i \to h = i - 1 \pmod n$

$\searrow\ j = i + 1 \pmod n$

Ogni coppia regge un bit b_{hi} noto solo a C_h e C_i.

$C_i \to b_{hi}$

$C_i \searrow b_{ij}$

ognuno riceve due bit

cuoce

ognuno calcola

$$v_i = b_{hi} \oplus b_{ij} \oplus s_i \quad \forall i$$

v_i è pubblico

ove s_i è il segreto anonimo che c_i vuole

lanciare: $\begin{cases} s_i = 0 & \text{se } c_i \text{ non ha pagato} \\ s_i = 1 & \text{se } c_i \text{ ha pagato} \end{cases}$

Se

$$s = s_1 \oplus \dots \oplus s_n = 0$$

ha pagato NSA

Se $s = s_1 \oplus \dots \oplus s_n = 1$

un criptografo anonimo ha pagato

Allora

$$s = v_1 \oplus \dots \oplus v_n = \left(b_{n1} \oplus b_{12} \oplus s_1 \right) \oplus \left(b_{12} \oplus b_{23} \oplus s_2 \right) \oplus$$

$$\dots \oplus \left(b_{n-1,n} \oplus b_{n,1} \oplus s_n \right) =$$

$$= \left(b_{n1} \oplus b_{n1} \oplus s_1 \right) \oplus \dots \oplus \left(b_{n-1,n} \oplus b_{n-1,n} \oplus s_n \right) =$$

$$= s_1 \oplus \dots \oplus s_n = s.$$