Alma Mater Studiorum · Università di Bologna

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI
Corso di Laurea in Matematica

Crittografia e Problema dei Residui Quadratici

Tesi di Laurea in Algoritmi della Teoria dei Numeri e Crittografia

Relatore:
Chiar.mo Prof.
Davide Aliffi

Presentata da:
Giovanni Canarecci

Sessione II
Anno Accademico 2011/2012

Un drago non è una fantasia oziosa
(J.R.R. Tolkien)

Indice

Introduzione

In questo elaborato intendiamo presentare due visioni successive e conseguenti di uno stesso problema: il problema dei Residui Quadratici (e il susseguente problema delle Radici Quadratiche). L'analisi di questi problemi verrà fatta inizialmente da un punto di vista maggiormente qualitativo, a cui seguirà una parte algoritmica e una applicativa.
Nel primo capito ci occuperemo del problema dei Residui Quadratici dal punto di vista della Teoria dei Numeri, presentandolo in una sua generalità e mostrando risultati legati all'esistenza delle soluzioni (nonché al loro effettivo valore).
Nella prima parte del secondo capitolo intendiamo studiare algoritmi atti a risolvere effettivamente entrambi i problemi, sia in casi particolari che in generale. Ovviamente, tutto ciò ove questi algoritmi sono conosciuti.
Infine, nell'ultima parte mostreremo come la difficoltà di risolvere, in generale, questi problemi permetta di creare algoritmi dei quali è possibile dimostrare la sicurezza, secondo certe definizioni e sotto opportune ipotesi. La definizione di sicurezza cui si fa riferimento generalizza l'idea di Shannon della sicurezza perfetta.

di cifratura per criptare messaggi. In questa parte presenteremo tre metodi, mettendo alla prova la loro velocità e la loro sicurezza.

Capitolo 1

Residui Quadratici

In questo capitolo intendiamo definire ed analizzare, dal punto di vista della teoria dei numeri, il problema dei residui quadratici.
Per far questo vedremo vari risultati e arriveremo a trattare i simboli di Legendre e di Jacobi.

1.1 Residui n-esimi modulo p

Iniziamo definendo e trattando la generalizzazione dei residui quadratici: i residui n-esimi che, in questa prima parte, affronteremo sempre modulo un primo p.

Definizione. *(Congruenza binomia)*
Sia p un numero primo. Una *congruenza algebrica (di grado n)* si dice *binomia* se è della forma

$$cx^n \equiv b \bmod p.$$

Osservazione. Se $c \equiv_p 0$, la congruenza binomia è un'identità *se e solo se* $b \equiv_p 0$, altrimenti risulta impossibile. Diversamente risulta $MCD(c,p) = (c,p) = 1$, pertanto c è invertibile e la congruenza può essere ridotta ad $x^n \equiv_p c^{-1}b \equiv_p a$.

Definizione. *(Residui n-esimi)*
Sia p un numero primo. Un numero $a \in \mathbb{Z}_p$ si dice *residuo n-esimo modulo p* se la congruenza

$$x^n \equiv a \bmod p.$$

ha soluzione. In caso contrario si parla di *non residuo n-esimo modulo p*.

Una prima caratterizzazione delle soluzionidi questa congruenza possiamo averla dal seguente teorema.

Teorema. *Sia g un generatore di $\mathbb{Z}_p^*$. Sia $a \equiv_p g^i$. Allora la congruenza:*

$$x^n \equiv_p a$$

ha soluzione se e solo se ha soluzione

$$ny \equiv_{p-1} i.$$

In tal caso, se $y_1, \ldots, y_d$ sono le soluzioni della seconda, allora tutte e sole le soluzioni della prima sono $x_j = g^{y_j}$, $\forall j = 1, \ldots, d$

Dimostrazione. Dato che il caso $a = 0$ si risolve immediatamente, si può considerare $a \in \mathbb{Z}_p^*$. Da qui viene che anche $x \in \mathbb{Z}_p^*$; per cui posso scrivere $x \equiv_p g^y$.
Ora: $g^{ny} \equiv_p x^n \equiv_p a \equiv_p g^i \Leftrightarrow g^{ny-i} \equiv_p 1 \Leftrightarrow ny - i \equiv_{p-1} 0 \Leftrightarrow ny \equiv_{p-1} i$.
La maggior parte dei passaggi qui svolti è di pura sostituzione. Concentriamoci ora sulla penultima doppia implicazione, che si dimostra in questo modo:
Necessità) L'ordine di g è $p-1$ perché g è un generatore di $\mathbb{Z}_p^*$, che ha $p-1$ elementi. Da qui segue che $ny - i$ è maggiore dell'ordine di g, anzi $ny - i$ è un multiplo di $p-1$.
Poniamo, per assurdo, che non sia un suo multiplo. Allora posso scrivere $ny - i = q(p-1) + r$, con $0 < r < p-1$. Ma $g^{ny-i} \equiv_p 1$ e $g^{p-1} \equiv_p 1$, perciò risulterà vero anche che $g^r \equiv_p 1$, che è assurdo perchè r è minore dell'ordine, che è il più piccolo numero per cui deve venire verrificata questa relazione. Trovo così che $ny - i \equiv_{p-1} 0$.
Sufficienza) $ny - 1 = k(p-1)$ e $g^{p-1} \equiv_p 1$ per il Teorema di Fermat. Perciò $g^{ny-i} \equiv_p g^{k(p-1)} \equiv_p 1$.
Questo dimostra che ognuna delle due equazioni di partenza ha soluzione *se e solo se* anche l'altra ha soluzione. In tal caso, sia $d = (n, p-1)/i$, la seconda congruenza ha esattamente d soluzioni $y_1, \ldots, y_d$ (proposizione 3.3.4[1]). Inoltre si vede subito sostituendo che $x_j = g^{y_j}$, per $j = 1, \ldots, d$, sono soluzioni della prima. Per mostrare che sono tutte provo che queste soluzioni non dipendono da g.
Sia g_1 un altro generatore di $\mathbb{Z}_p^*$ e sia $g \equiv_p g_1^j$. Perciò $x \equiv_p g_1^{jy}$ e risulta $g_1^{jmy} \equiv_p g_1^{ji} \Leftrightarrow jmy \equiv_{p-1} ji \Leftrightarrow my \equiv_{p-1} i$ (per l'ultima doppia implicazione utilizzo il fatto $(j, p-1) = 1$, che ci viene da A3.30[1]).
Dato che da ogni g ci si riconduce alla stessa congruenza in y, ne segue che le soluzioni già trovate della prima congruenza sono anche le uniche. □

1.2 Residui quadratici modulo p

Passiamo ora ad analizzare nello specifico il caso $n = 2$.

Definizione. Se $n = 2$, la congruenza $x^n \equiv a \bmod p$ diventa:

$$x^2 \equiv a \bmod p$$

e a si chiama *residuo quadratico modulo p* se posso trovarne una soluzione.

Arrivati a questo punto, possiamo possiamo dare una prima definizione del problema cardine di tutto il capitolo successivo, dedicato alla crittografia.

Definizione. Sia p un primo. Per *problema dei residui quadratici modulo p* si intede il problema di stabilire se, dato $a \in \mathbb{Z}_p$, l'equazione $x^2 \equiv a \bmod p$ abbia soluzione (ovvero scoprire se a è o no un residuo quadratico modulo p).
Si definisce poi *problema delle radici quadrate modulo p* il problema di calcolarne le eventuali soluzioni.

In letteratura i due problemi possono essere definiti in modo diverso, facendo rientrare sotto il nome di *problema dei residui quadratici modulo p* entrambi i problemi, rendendo così il *problema delle radici quadrate modulo p* una sottoparte dell'altro.

Esempio. In $\mathbb{Z}_5$ i residui quadratici sono

$$1 \equiv_5 1^2 \equiv_5 4^2, \quad 4 \equiv_5 2^2 \equiv_5 3^2.$$

In $\mathbb{Z}_7$ sono

$$1 \equiv_7 1^2 \equiv_7 6^2, \quad 2 \equiv_7 3^2 \equiv_7 4^2, \quad 4 \equiv_7 2^2 \equiv_7 5^2.$$

In $\mathbb{Z}_{11}$ sono

$$1 \equiv_{11} 1^2 \equiv_{11} 10^2, \quad 3 \equiv_{11} 5^2 \equiv_{11} 6^2, \quad 4 \equiv_{11} 2^2 \equiv_{11} 9^2,$$

$$5 \equiv_{11} 4^2 \equiv_{11} 7^2, \quad 9 \equiv_{11} 3^2 \equiv_{11} 8^2.$$

Proposizione. Sia p primo e g un generatore di $\mathbb{Z}_p^*$. $a \equiv_p g^j$ è un residuo quadratico modulo p *se e solo se* j è pari. E, se ciò accade, le due soluzioni di $x^2 \equiv_p a$ sono g^i e $-g^i$, con $j = 2i$.

Dimostrazione. Necessità) $x^2 \equiv_p a$ ha soluzione quindi, per il teorema visto, ha soluzione anche $2y \equiv_{p-1} j$, cioè $2y = j + k(p-1)$. Dato che ovviamente $(p-1, 2) = 2$, risulta $2/j$, ovvero j pari.
Sufficienza) $j = 2i$, $a \equiv_p g^{2i}$. Viene subito che a è un residuo quadratico modulo p perché le soluzioni di $x^2 \equiv_p a$ sono g^i e $-g^i$. □

Osservazione. Sia p primo e g un generatore di $\mathbb{Z}_p^*$.

- Il prodotto di due residui quadratici modulo p, a e b, è ancora un residuo quadratico modulo p.

  Dimostrazione. Siano $a \equiv_p g^{2h}$ e $b \equiv_p g^{2k}$. $ab \equiv_p g^{2h}g^{2k} \equiv_p g^{2(h+k)}$, perciò ab è ancora un residuo quadratico. □

- Il prodotto di due non residui quadratici modulo p è un residuo quadratico modulo p.

  Dimostrazione. Siano $a \equiv_p g^{2h+1}$ e $b \equiv_p g^{2k+1}$. $ab \equiv_p g^{2h+1}g^{2k+1} \equiv_p g^{2(h+k+1)}$, perciò ab è un residuo quadratico. □

- Il prodotto di un residuo e di un non residuo è un non residuo.

  Dimostrazione. Siano $a \equiv_p g^{2h}$ e $b \equiv_p g^{2k+1}$. $ab \equiv_p g^{2h}g^{2k+1} \equiv_p g^{2(h+k)+1}$, perciò ab è un non residuo. □

- I residui quadratici modulo p sono tutti e soli

$$1^2, 2^2, 3^2, \ldots, \left(\frac{p-1}{2}\right)^2$$

  .

  Dimostrazione. Gli elementi di $\mathbb{Z}_p$ si possono scrivere come

$$-\frac{p-1}{2}, -\frac{p-3}{2}, \ldots, -1, 0, 1, \ldots, \frac{p-3}{2}, \frac{p-1}{2}.$$

  Poiché siamo in un campo, la relazione $a^2 - b^2 \equiv_p (a-b)(a+b) \equiv_p 0$ implica $a \equiv_p \pm b$; cioé, se due quadrati a^2 e b^2 coincidono modulo p allora $a \equiv_p \pm b$. Ciò implica che $1^2, 2^2, 3^2, \ldots, (\frac{p-1}{2})^2$ sono tutti i residui quadratici e sono a due a due incongrui perché lo sono $1, 2, 3, \ldots, \frac{p-1}{2}$. □

- I non residui quadratici modulo p sono tutti e soli $c, c2^2, c3^2, \ldots, c(\frac{p-1}{2})^2$, ove c è un non residuo quadratico modulo p.

  Dimostrazione. Viene immediatamente dai due punti subito sopra. □

Teorema. *(Caratteristica quadratica di Eulero o Criterio di Eulero)*
Sia p un numero primo e sia $a \in \mathbb{Z}_p$.
a è un residuo quadratico modulo p se e solo se $a^{\frac{p-1}{2}} \equiv_p 1$.
a è un non residuo quadratico modulo p se e solo se $a^{\frac{p-1}{2}} \equiv_p -1$.

Osservazione. La complessità computazionale del calcolo di $a^{\frac{p-1}{2}}$ modulo p è, usando l'algoritmo di esponenziazione modulare, $O(\log^3 p)$, quindi polinomiale.

Esempio. Ad esempio, sia $a = 17$. Se $p = 3$, $a^{\frac{p-1}{2}} \equiv_3 17^{\frac{(3-1)}{2}} \equiv_3 17^1 \equiv_3 2 \equiv_3 -1$, quindi 17 non è un residuo quadratico modulo 3.
Se invece $p = 13$, $a^{\frac{p-1}{2}} \equiv_{13} 17^{\frac{(13-1)}{2}} \equiv_{13} 17^6 \equiv_{13} 1$, quindi 17 è un residuo quadratico modulo 13 (difatti $17 \equiv_{13} 2^2$).

Lemma. Sia $p > 2$ un numero primo e sia $a \in \mathbb{Z}_p$. Se considero le due congruenze $a^{\frac{p-1}{2}} - 1 \equiv_p 0$ e $a^{\frac{p-1}{2}} + 1 \equiv_p 0$, una (e solo una) delle due è vera.

Dimostrazione. (Lemma)
Per il teorema di Fermat si ha $a^{p-1} \equiv_p 1$, ossia $a^{p-1} - 1 \equiv_p 0$. Osservando che p-1 è pari si può scrivere $(a^{\frac{p-1}{2}} - 1)(a^{\frac{p-1}{2}} + 1) \equiv_p 0$. Da ciò si evince subito che una delle due congruenze è verificata.
Ora osservo che se $b \in \mathbb{Z}$, allora $MCD(b-1, b+1) = d \leq 2$.
Infatti siano $h, k \in \mathbb{Z}$ tali che $b - 1 = dh, b + 1 = dk$, allora $2b = d(h + k)$. Se, per assurdo, $d > 2$, succede anche che d/b, ma in questo modo $d/1$, il che è assurdo. Perciò $MCD(a^{\frac{p-1}{2}} - 1, a^{\frac{p-1}{2}} + 1) \leq 2$ e, poiché $p > 2$, non è possibile che entrambi gli interi siano multipli di p. □

Dimostrazione. (Caratteristica quadratica di Eulero)
Dimostreremo solo le due implicazioni verso destra, le altre infatti sono verificate immediatamente grazie alle prime e al lemma. Sia g un generatore di $\mathbb{Z}_p^*$. a è un residuo quadratico *se e solo se* $a \equiv_p g^{2i}$, che, a sua volta, implica $a^{\frac{p-1}{2}} \equiv_p (g^{2i})^{\frac{p-1}{2}} \equiv_p g^{i(p-1)} \equiv_p (g^{p-1})^i \equiv_p 1$ per il teorema di Fermat.
Similmente a è un non residuo quadratico *se e solo se* $a \equiv_p g^{2i+1}$, che implica $a^{\frac{p-1}{2}} \equiv_p (g^{2i+1})^{\frac{p-1}{2}} \equiv_p g^{i(p-1)} g^{\frac{p-1}{2}} \equiv_p g^{\frac{p-1}{2}}$ per il teorema di Fermat.
Dal Fermat so che $(g^{\frac{p-1}{2}})^2 \equiv_p 1$, per cui $g^{\frac{p-1}{2}}$ è sicuramente una soluzione di questa congruenza, ovvero o è 1 o è -1.
Per assurdo sia $g^{\frac{p-1}{2}} \equiv_p 1$; in questo caso però g non potrebbe più generare tutti i $p - 1$ elementi di $\mathbb{Z}_p^*$, ma ne genererebbe alpiù $(\frac{p-1}{2})$, il che è assurdo.
Perciò $a^{\frac{p-1}{2}} \equiv_p g^{\frac{p-1}{2}} \equiv_p -1$ e questo conclude la dimostrazione. □

1.3 Simbolo di Legendre

Un altro metodo per individuare residui quadratici si ottiene dallo studio di un simbolo appositamente definito: il simbolo di Legendre.

Definizione. Sia p un primo dispari e sia a un intero. Si definisce *simbolo di Legendre* il seguente:

$$\left(\frac{a}{p}\right) := \begin{cases} 0, & \text{se } p/a \\ 1, & \text{se } a \text{ è un residuo quadratico modulo p} \\ -1, & \text{se } a \text{ è un non residuo quadratico modulo p} \end{cases}$$

Osservazione. La definizione è ben posta perché i tre casi sono tra loro incompatibili ed esaustivi.

Osservazione. Se $(\frac{a}{p}) = 1$ scriveremo $a \in Q_p$, con Q_p l'insieme dei residui quadratici modulo p.

Osservazione. Il calcolo del simbolo di Legendre ha una complessità computazionale equivalente al calcolo di $a^{\frac{p-1}{2}} \bmod p$, ovvero $O(\log^3 p)$.

Esempio.

$$\left(\frac{2}{11}\right) = -1$$

perché 2 non è un residuo quadratico di 11, mentre

$$\left(\frac{3}{11}\right) = 1$$

perché $3 \equiv_{11} 5^2$.

$$\left(\frac{4}{13}\right) = 1$$

perché $4 = 2^2$. Infine

$$\left(\frac{6}{3}\right) = 0$$

perché 6 é divisibile per 3.

Vediamo ora alcune proprietà del Simbolo di Legendre; esse si dimostrano direttamente dalla definizione e utilizzando la caratteristica quadratica di Eulero.

Proposizione. Sia p un primo dispari e siano a, b interi relativamente primi con p. Allora il simbolo di Legendre gode delle seguenti proprietà:

1. se $a \equiv_p b$, allora $(\frac{a}{p}) = (\frac{b}{p})$;
2. $(\frac{a^2}{p}) = 1$;
3. $(\frac{a}{p}) \equiv_p a^{\frac{p-1}{2}}$;
4. $(\frac{ab}{p}) = (\frac{a}{p})(\frac{b}{p})$;
5. $(\frac{1}{p}) = 1$ e $(\frac{-1}{p}) = (-1)^{\frac{p-1}{2}}$.

Ad esempio, con le proprietà appena viste (e con le due proposizioni che enuncieremo tra poco) si dimostra l'osservazione seguente.

Osservazione. Se p è un primo dispari. Si ha:

$$\left(\frac{2}{p}\right) = \begin{cases} 1, & \text{se } p = 8h \pm 1 \\ -1, & \text{se } p = 8h \pm 3 \end{cases}$$

$$\left(\frac{3}{p}\right) = \begin{cases} 1, & \text{se } p = 12h \pm 1 \\ -1, & \text{se } p = 12h \pm 5 \end{cases}$$

con $h \geq 0$, intero.

Vediamo qui le due proposizioni a cui ci si riferiva sopra.

Proposizione. *(Lemma di Gauss)*
Sia p un primo dispari e sia $a \in \mathbb{Z}$ tale che $MCD(a, p) = 1$. Allora

$$\left(\frac{a}{p}\right) = (-1)^{n(a,p)},$$

ove $n(a, p)$ rappresenta il numero di interi dell'insieme

$$A := \{a, 2a, \ldots, \frac{(p-1)}{2}a\},$$

visto come sottoinsieme di $\mathbb{Z}_p$, maggiori di $p/2$.

Dimostrazione. Dato che $MCD(a, p) = 1$, nessuno degli elementi di A può annullarsi modulo p. Inoltre sono tutti distinti per come sono stati definiti. Perciò si può osservare subito che la cardinalità di A è $(p-1)/2$.
Dividiamo ora gli elementi di A in due sottoinsiemi A_1 e A_2 disgiunti. Poniamo in A_1 gli elementi di A compresi tra 1 e $(p-1)/2$ e in A_2 i rimanenti elementi di A, ossia quelli tra $p/2$ e p. Indichiamo con m il numero di elementi di A_1 mentre il numero di elementi n di A_2 è proprio $n(a, p)$. Ovviamente

si ha $n + m = (p-1)/2$.
Indichiamo ora come r_h gli elementi di A_1, cioè scriviamo $A_1 = \{r_1, r_2, \ldots, r_m\}$, e con s_k quelli di A_2, cioè $A_2 = \{s_1, s_2, \ldots, s_n\}$. Allora gli interi $r_1, \ldots, r_m, p - s_1, \ldots, p - s_n$ sono tali che

$$0 < r_1, \ldots, r_m, p - s_1, \ldots, p - s_n < \frac{p}{2}.(*)$$

Si vuole ora provare che questi numeri sono tutti distinti. Da qui ne seguirà che coincidono, a meno dell'ordine, con gli interi $1, 2, 3, \ldots, \frac{p-1}{2}$, che sono lo stesso numero.
Per assurdo sia, per qualche h e k, $p - s_k = r_h$.
Per la definizione di A esistono due interi α e β, con $1 \leq \alpha, \beta \leq (p-1)/2$ e tali che $r_h \equiv_p \alpha a$ e $s_k \equiv_p \beta a$, da cui $(\alpha + \beta)a \equiv_p r_h + s_k \equiv_p p \equiv_p 0$, il che implica $\alpha + \beta \equiv_p 0$. Ma questo è assurdo perché $0 < \alpha + \beta \leq p - 1$.
Essendo dunque $r_1, \ldots, r_m, p - s_1, \ldots, p - s_n$ tutti distinti, dalla $(*)$ segue che essi coincidono, a meno dell'ordine, con gli interi $1, 2, 3, \ldots, \frac{p-1}{2}$.
Sfruttando questo fatto, si può scrivere

$$\left(\frac{p-1}{2}\right)! = r_1 \cdot r_m \cdot (p - s_1) \cdot \ldots \cdot (p - s_n) \equiv_p (-1)^n r_1 \cdot \ldots \cdot r_m \cdot s_1 \cdot \ldots \cdot s_n.$$

Dato che gli r_h e gli s_k, per $h = 1, \ldots, m$ e $k = 1, \ldots, n$, sono i resti modulo p degli elementi di A, risulta

$$\left(\frac{p-1}{2}\right)! \equiv_p (-1)^n a \cdot 2a \cdot 3a \cdot \cdots \cdot \left(\frac{p-1}{2}\right) a \equiv_p (-1)^n a^{\frac{p-1}{2}} \left(\frac{p-1}{2}\right)!.$$

Notando poi che $MCD((\frac{p-1}{2})!, p) = 1$, si ottiene $a^{(p-1)/2} \equiv_p (-1)^n$. Attraverso il terzo punto della proposizione precedente si vede che questo era ciò che si voleva dimostrare. □

Il teorema seguente ha permesso, storicamente, insieme ai risultati già esposti, di effettuare un calcolo efficiente del simbolo di Legendre.
Questo risultato, inoltre, è di grande importanza per tutta la teoria dei numeri: congetturato da Eulero e da Legendre, venne dimostrato da un diciannovenne Gauss.

Teorema. *(Legge di reciprocità quadratica)*
Siano p e q primi dispari distinti. Allora

$$\left(\frac{p}{q}\right) = \left(\frac{q}{p}\right)$$

a meno che non sia $p \equiv_4 q \equiv_4 3$, nel qual caso

$$\left(\frac{p}{q}\right) = -\left(\frac{q}{p}\right)$$

Equivalentemente

$$\left(\frac{p}{q}\right)\left(\frac{q}{p}\right) = (-1)^{(p-1)(q-1)/4}.$$

Dimostrazione. Consideriamo il rettangolo aperto R del piano xy di vertici $(0,0), (p/2,0), (0,q/2), (p/2,q/2)$. Attraverso un semplicissimo ragionamento geometrico, si vede che ci sono esattamente $(p-1)(q-1)/4$ punti, contenuti in R, le cui coordinate sono numeri interi, cioé tutti i punti del tipo (i,j), con $1 \leq i \leq (p-1)/2, 1 \leq j \leq (q-1)/2$.
Definisco D la diagonale da $(0,0)$ a $(p/2,q/2)$ del rettangolo, di equazione $py = qx$. Dato che $MCD(p,q) = 1$, nessuno dei punti a coordinate intere di R giace su D. Indichiamo ora con R_1 la porzione di R posta sotto D, cioé dove $py < qx$, e con R_2 la porzione di R posta sopra D, ove $py > qx$.
Contiamo i punti interi presenti R_1. Sono tutti e soli i punti del tipo (i,j), con $1 \leq i \leq (p-1)/2, 1 \leq j \leq [qi/p]$. Perciò il numero dei punti interi in R_1 é esattamente $\tau(q,p) = \sum_{i=1}^{(p-1)/2}[qi/p]$. Analogamente, il numero dei punti di R_2 le cui coordinate sono numeri interi è $\tau(p,q) = \sum_{i=1}^{(q-1)/2}[pi/q]$.
Possiamo quindi concludere che

$$\frac{(p-1)(q-1)}{4} = \tau(p,q) + \tau(q,p).$$

Il teorema segue immediatamente dall'ultima Proposizione. □

Osservazione. La definizione del simbolo di Legendre già ci dice che $(\frac{p}{q})$ è legato alla risoluzione di $x^2 = p \bmod q$. Allo stesso modo $(\frac{q}{p})$ è legato alla risoluzione di $x^2 = q \bmod p$. Alla luce della legge di reciprocità quadratica, però, possiamo dire anche un'altra cosa, ovvero che le due congruenze in questione sono strettamente legate tra loro.

1.4 Residui quadratici modulo n

Introduciamo ora la versione più generale del problema dei residui quadratici, che è oltretutto quella veramente oggetto di interesse dal punto di vista della crittografia.

Definizione. Sia $n = pq$ con p e q primi dispari diversi. Per *problema dei residui quadratici modulo n* si intende il problema di stabilire se, dato $a \in \mathbb{Z}_n$,

l'equazione $x^2 \equiv a \bmod n$ ha soluzione.
Anche qui si definisce *problema delle radici quadrate modulo n* il problema di calcolarne le eventuali soluzioni.

Osservazione. Nei problemi di crittografia si prende n come prodotto di due primi diversi e molto grandi. E' possibile scegliere residui quadratici in modo efficiente (alpiù prendendo un intero ed elevandolo al quadrato), ma rimane comunque difficile risolvere totalmente la congruenza associata.

Per risolvere il problema dei residui quadratici modulo n, si frutta in maniera molto consistente quanto detto finora modulo p. Infatti:

Proposizione. Siano $a, b \in \mathbb{Z}_n$. Si ha che:
$a \equiv_{nm} b$ implica che $a \equiv_n b$ e $a \equiv_m b$.
Viceversa, se $(n, m) = 1$, $a \equiv_n b$ e $a \equiv_m b$ implica che $a \equiv_{nm} b$.

Dimostrazione. Se nm divide $a - b$, allora anche n e m dividono $a - b$.
Viceversa se n e m dividono entrambi $a - b$ e non hanno fattori in comune, allora anche nm divide $a - b$. □

Nel nostro caso avremo sempre $(p, q) = 1$, quindi le due condizioni saranno sempre equivalenti. Pertanto possiamo affermare che, per capire se un intero sia un residuo quadratico modulo n, basta chiedersi se sia un residuo quadratico modulo p e modulo q.
Si può notare inoltre che questo discorso vale per un intero positivo n qualunque.
Inoltre, anche per quel che riguarda il calcolo effettivo delle radici modulo n, possiamo fare riferimento al lavoro svolto modulo un primo. Infatti, per calcolare le due radici quadrate modulo n, basta trovare quelle modulo p e modulo q e combinarle insieme con il teorema cinese dei resti. Si otterranno così le quattro soluzioni della congruenza modulo n.
Notare che le soluzioni saranno, in generale, quattro e non solo due perché $\mathbb{Z}_n$ non è un campo.

Esempio. Ad esempio si prenda:

$$x^2 \equiv 25 \bmod 8.$$

Due soluzioni sono ovviamente $x_1 = 5$ e $x_2 = -5 \equiv 3$. Però, dato che $x^2 \equiv 25 \equiv 1 \bmod 8$, ci sono altre due soluzioni, ovvero $x_3 = 1$ e $x_4 = -1 \equiv 7$.

1.5 Il simbolo di Jacobi

Definiamo ora una generalizzazione del simbolo di Legendre, che ci servirà molto nel capitolo successivo: il simbolo di Jacobi. Anche questo simbolo, come vedremo, è legato alla risoluzione del problema dei residui quadratici (modulo n), ma in maniera meno stringente rispetto a Legendre.

Definizione. Sia a un intero qualunque e sia n un intero positivo dispari. Sia $n = p_1^{\alpha_1} \cdot \ldots \cdot p_r^{\alpha_r}$, con $p_1, \ldots, p_r$ numeri primi dispari distinti. Si definisce il *simbolo di Jacobi*:

$$\left(\frac{a}{n}\right) := \left(\frac{a}{p_1}\right)^{\alpha_1} \cdot \ldots \cdot \left(\frac{a}{p_r}\right)^{\alpha_r}.$$

Osservazione. Si osserva che, se n non è primo, $(\frac{a}{n}) = 1$ non significa necessariamente che a è un residuo quadratico modulo n. Questa condizione non è più sufficiente, pur rimanendo necessaria.

Esempio. Come esempio per l'osservazione sopra, vediamo che

$$\left(\frac{2}{15}\right) = \left(\frac{2}{3}\right)\left(\frac{2}{5}\right) = (-1)(-1) = 1,$$

sebbene l'equazione $x^2 \equiv 2 \bmod 15$ non abbia nessuna soluzione.

Definizione. Se $(\frac{a}{n}) = 1$, scriviamo $a \in J_p$, con J_p che è precisamente l'insieme di tutti gli $a \in \mathbb{Z}_n^*$ con simbolo di Jacobi pari a 1.

Definizione. Definiamo poi Q_n come l'insieme dei residui quadratici modulo n ($Q_n \subseteq J_n$) e $\tilde{Q_n} = J_n - Q_n$ l'insieme degli pseudo quadrati modulo n.

Di sicuro su questo si possono dire due cose:

Osservazione. Sia $a \in \mathbb{Z}_n^*$

- Se il simbolo di Jacobi $(\frac{a}{n}) = -1$, allora a è un non residuo quadratico.
- Se $(\frac{a}{n}) = 1$, allora a è un residuo quadratico modulo n, *i.e.* $a \in Q_n$, *se e solo se* $a \in Q_p$ e $a \in Q_q$.

Nonostante ciò, il simbolo di Jacobi gode di alcune proprietà formali analoghe a quelle del simbolo di Legendre, che enunciamo qui in due proposizioni.

Proposizione. Sia n un numero dispari positivo e siano a e b interi relativamente primi con n. Allora si ha:

1. se $a \equiv_n b$, allora $(\frac{a}{n}) = (\frac{b}{n})$;
2. $(\frac{ab}{n}) = (\frac{a}{n})(\frac{b}{n})$;
3. $(\frac{1}{n}) = 1$ e $(\frac{-1}{n}) = (-1)^{(n-1)/2}$.

Dimostrazione. Proviamo solo il terzo punto. Sia $n = p_1^{\alpha_1} \cdots p_r^{\alpha_r}$, con $p_1, \ldots, p_r$ numeri primi dispari distinti. Per definizione e per le proprietà del simbolo di Legendre, si ha

$$\left(\frac{-1}{n}\right) = \left(\frac{-1}{p_1}\right)^{\alpha_1} \cdots \left(\frac{-1}{p_r}\right)^{\alpha_r} = (-1)^{(\alpha_1(p_1-1)+\cdots+\alpha_r(p_r-1))/2}(*)$$

Ora notiamo che, per ogni numero dispari p e per ogni intero positivo α, si ha $p^\alpha = (1 + (p-1))^\alpha \equiv_4 1 + \alpha(p-1)$ (vedi A5.57[1]). Abbiamo allora

$$n \equiv_4 (1 + \alpha_1(p_1 - 1)) \cdot \ldots \cdot (1 + \alpha_r r(p_r - 1))$$

da cui si ottiene

$$n \equiv_4 1 + \alpha_1(p_1 - 1) + \cdots + \alpha_r(p_r - 1)$$

tenendo presente che $p_1 - 1, \ldots, p_r - 1$ sono tutti numeri pari (e che moltiplicandoli tra loro scompaiono mod 4). Riscrivendo posso dire

$$\frac{n-1}{2} \equiv_2 \frac{\alpha_1(p_1 + 1) + \cdots + \alpha_r(p_r - 1)}{2}.$$

Sostituendo questo risultato in $(*)$, si ottiene la tesi. □

Osservazione. Se n è un intero dispari positivo, si ha

$$\left(\frac{2}{n}\right) = (-1)^{(n^2-1)/8}$$

Proposizione. *Analogo della reciprocità quadratica*
Se n e m sono interi dispari positivi, si ha

$$\left(\frac{n}{m}\right) = \left(\frac{m}{n}\right)$$

a meno che non sia $n \equiv_4 m \equiv_4 3$, nel qual caso

$$\left(\frac{n}{m}\right) = -\left(\frac{m}{n}\right).$$

Equivalentemente

$$\left(\frac{n}{m}\right)\left(\frac{m}{n}\right) = (-1)^{(n-1)(m-1)/4}$$

Esempio. Calcolare $\left(\frac{3083}{3911}\right)$.

$$\left(\frac{3083}{3911}\right) = -\left(\frac{3911}{3083}\right)$$

perché entrambi i numeri sono congrui a 3 modulo 4. Dividendo 3911 per 3083 si trova come resto $828 = 4\dot{2}07$. Allora

$$\left(\frac{3083}{3911}\right) = -\left(\frac{4}{3083}\right) = \left(\frac{207}{3083}\right)$$

Concludendo il calcolo:

$$\left(\frac{3083}{3911}\right) = -\left(\frac{207}{3083}\right) = \left(\frac{3083}{207}\right) = \left(\frac{185}{207}\right) = \left(\frac{207}{185}\right) =$$

$$\left(\frac{2}{185}\right)\left(\frac{11}{185}\right) = \left(\frac{11}{185}\right) = \left(\frac{185}{11}\right) = \left(\frac{9}{11}\right) = \left(\frac{11}{9}\right) = \left(\frac{2}{9}\right) = 1$$

Osservazione. La complessità computazionale del calcolo del simbolo di Jacobi $\left(\frac{n}{m}\right)$, con $n > m$, coincide con quella del calcolo del massimo comun divisore di n e m, cioè $O(\log^3 n)$. A questo concorrono le proprietà presentate sul simbolo di Jacobi, prima tra tutte la legge di reciprocità quadratica, che permettono di svolgere il calcolo senza bisogno di fattorizzare i numeri presenti.

Capitolo 2

Cifrari basati sul problema dei residui quadratici

In questo capitolo ci prefiggiamo di illustrare l'utilizzo concreto del problema dei residui quadratici per creare algoritmi di crittografia. Innanzitutto ci concentreremo nel mostrare che il problema dei residui quadratici modulo n è apparentemente computazionalmente intrattabile per n prodotto di primi grandi; successivamente passeremo a presentare algoritmi a chiave pubblica: Rabin, Goldwasser-Micali e Blum-Goldwasser. Per algoritmo di cifratura si intende un elenco di istruzioni grazie alle quali due interlocutori possano scambiarsi messaggi senza che una terza persona (l'avversario) possa intercettarli e leggerli.

Ricordiamo inoltre che, se $n = p$ è primo, è facile sapere se un $a \in \mathbb{Z}_p^*$, sta in Q_n. Ciò succede *se e solo se* $(\frac{a}{p}) = 1$, sapendo anche che tale simbolo di Legendre può essere efficacemente calcolato tramite l'algoritmo 2.149[2]. Il problema è più complesso se n è il prodotto di due primi grandi, come vedremo tra poco.

2.1 Calcolare radici quadrate modulo n

A questo punto intendiamo affrontare il problema del calcolo di radici quadrate modulo n dal punto di vista prettamente algoritmico: tale calcolo risulterà difficile per n composto da due fattori primi grandi e non noti, mentre vedremo essere affrontabili il caso $n = p$ primo e il caso con n prodotto di due primi noti.

2.1.1 Caso $n = p$ primo

Il nostro obiettivo in questo caso è capire se, dato $a \in \mathbb{Z}_p^*$, a sia un residuo quadratico modulo p e, in tal caso, dimostrare la fattibilità pratica della risoluzione del problema dei residui quadratici associato.

Algoritmo 1. *(Algoritmo per trovare le radici quadrate di a modulo un primo p, ovvero per risolvere $x^2 \equiv_p a$)*
INPUT un primo dispari p e un intero a tale che $1 \leq a \leq p-1$, *i.e.* $a \in \mathbb{Z}_p^*$.
OUTPUT la risposta alla domanda se a sia o meno un residuo quadratico e le due radici quadrate di a modulo p, per una risposta positiva alla domanda.

1. Calcolare il simbolo di Legendre $(\frac{a}{p})$ usando l'algoritmo 2.149[2]. Se risulta $(\frac{a}{p}) = -1$ allora a non è un residuo quadratico modulo p e l'algoritmo termina.
2. Prendere un intero b in modo uniformemente casuale, tale che $1 \leq b \leq p-1$, finché non si trova un b tale che $(\frac{b}{p}) = -1$ (b è un non residuo quadratico modulo p).
3. Ripetendo la divisione per 2, scrivere $p - 1 = 2^s t$, con t dispari.
4. Calcolare a^{-1} mod p con l'algoritmo euclideo esteso (2.142[2]).
5. Porre $c = b^t \bmod p$ e $r = a^{(t+1)/2} \bmod p$ (utilizzando l'algoritmo 2.143[2]).
6. Per i che va da 1 a $s-1$:
   (a) $d = (r^2 \cdot a^{-1})^{2^{s-i-1}} \bmod p$.
   (b) Se $d \equiv -1 \bmod p$ allora porre $r = r * c \bmod p$.
   (c) Porre $c = c^2 \bmod p$.
7. Risultato $(r, -r)$

Osservazione. L'algoritmo esposto sopra ha una complessità computazionale di $O(\log^4 p)$; il metodo è quindi polinomiale.

Dimostrazione. Questo risultato è ottenuto osservando che il passo principale, il sesto, viene eseguito $s-1$ volte e che ciascuna iterazione coinvolge un esponenziale in modulo, il cui calcolo costa $O(\log^3 p)$ (Tabella 2.5[2]). Dato che il caso peggiore si presenta quando $s = O(\log p)$, il costo totale risulta essere $O(\log^4 p)$. Quando s è piccolo, il ciclo al sesto passo viene eseguito solo un piccolo numero di volte e il tempo dell'algoritmo è $O(\log^3 p)$. □

Questo algoritmo è non deterministico per via del modo casuale con cui viene scelto b al passo 2. Attualmente non si conoscono algoritmi polinomiali deterministici per trovare un non-residuo quadratico modulo un primo p. Possiamo notare inoltre che, per cercare b in modo non deterministico, basta prendere dei b casuali, che soddisferanno la condizione richiesta nella metà dei casi.

Nel caso $s = 1$, dall'algoritmo 1 si può ricavare il seguente algoritmo deterministico per trovare radici quadrate modulo p, con $p \equiv_4 3$.

Algoritmo 2. *(Algoritmo per trovare le radici quadrate di a modulo un primo p, con $p \equiv_4 3$)*
INPUT un primo dispari p tale che $p \equiv_4 3$ e un quadrato $a \in Q_p$ (ovvero a residuo quadratico mod p).
OUTPUT le due radici quadrate di a modulo p.
1. Calcolare $r = a^{(p+1)/4} \bmod p$ (algoritmo 2.143[2]).
2. Risultato (r, $-$r).

Sempre partendo dall'algoritmo 1, nel caso $s = 2$, si può arrivare ad un altro algoritmo deterministico, che trova radici quadrate modulo p con $p \equiv_8 5$.

Algoritmo 3. *(Algoritmo per trovare le radici quadrate di a modulo un primo p, con $p \equiv_8 5$)*
INPUT un primo dispari p con $p \equiv_8 5$, e un quadrato $a \in Q_p$.
OUTPUT le due radici quadrate di a modulo p.

1. Calcolare $d = a^{(p-1)/4} \bmod p$ (algoritmo 2.143[2]).
2. Se $d = 1$, porre $r = a^{(p+3)/8} \bmod p$.
3. Se $d = p - 1$, porre $r = 2a(4a)^{(p-5)/8} \bmod p$.
4. Risultato $(r, -r)$.

Osservazione. Gli algoritmi 2 e 3 hanno complessità $O(\log^3 p)$, minore quindi di quella dell'algoritmo 1.

Vediamo ora un ultimo algoritmo del caso $n = p$ primo, che chiede come ipotesi $p - 1 = 2^s t$, con t dispari ed s grande. Questo algoritmo è vantaggioso rispetto all'algoritmo 1 perché, quando applicabile, ha una complessità computazionale di $O(\log^3 p)$.

Algoritmo 4. *(Algoritmo per trovare le radici quadrate modulo un primo p)*
INPUT un primo dispari p e un quadrato $a \in Q_p$.
OUTPUT le due radici quadrate di a modulo p.

1. Scegliere in modo uniformemente casuale $b \in \mathbb{Z}_p$, tale che $b^2 - 4a$ sia un non residuo quadratico modulo p, *i.e.* $(\frac{b^2-4a}{p}) = -1$.
2. Sia f il polinomio $x^2 - bx + a$ in $\mathbb{Z}_p[x]$.
3. Calcolare $r = x^{(p+1)/2} \bmod f$ usando l'algoritmo 2.227[2] (si può dimostrare che r è un intero).
4. Risultato $(r, -r)$.

2.1.2 Caso n composto

Affrontiamo ora il caso veramente utilizzabile negli algoritmi di crittografia, ovvero quello del problema della radici quadrate modulo n, con n prodotto di due primi p e q dispari, distinti, grandi e non noti.
Innanzitutto, in questo caso, il problema di determinare se a, $a \in \mathbb{Z}_n^*$, sia un residuo quadratico modulo n è considerato un problema difficile.

Si è visto che, se si conoscono i fattori primi di n (p e q), allora il problema è risolvibile in modo efficace trovando prima le radici quadrate di a modulo p e modulo q e combinandole con il teorema cinese dei resti (2.120[2]), che è polinomiale, ottenendo così le quattro radici quadrate modulo n.
Tutto questo è riassunto nel seguente algoritmo:

Algoritmo 5. *(Algoritmo per trovare le radici quadrate modulo n dati i suoi fattori primi p e q).*
INPUT: un intero n, i suoi fattori primi p e q e $a \in Q_n$.
OUTPUT: Le quattro radici quadrate di a modulo n.

1. Usare uno degli algoritmi precedenti per trovare le due radici quadrate r e $-r$ di a modulo p.
2. Usare uno degli algoritmi precedenti per trovare le due radici quadrate s e $-s$ di a modulo q.
3. Usare l'algoritmo euclideo esteso (2.107[2]) per trovare gli interi c e d tali che $cp + dq = 1$.
4. Porre $x = (rdq + scp) \bmod n$ e $y = (rdq - scp) \bmod n$.
5. Risultato $(\pm x \bmod n, \pm y \bmod n)$.

Osservazione. L'algoritmo 5 ha una complessità computazionale pari a $O(\log^3 p)$, ed è quindi polinomiale.

Algoritmo 6. *(Come trovare radici quadrate di a modulo $n = pq$, con $p \equiv q \equiv 3$ mod 4)*
Se p e q sono entrambi di questo tipo, allora l'algoritmo 5 per calcolare le quattro radici quadrate di c modulo n può essere riscritto come segue:

1. Calcolare $r \equiv a^{(p+1)/4} \bmod p$.
2. Calcolare $s \equiv a^{(q+1)/4} \bmod q$.
3. Usare l'algoritmo euclideo esteso (2.107[2]) per trovare gli interi c e d che soddisfano $cp + dq = 1$.
4. Calcolare $x = (cps + dqr) \bmod n$.
5. Calcolare $y = (cps - dqr) \bmod n$.
6. Le quattro radici quadrate di c modulo n sono $x, -x, y, -y \bmod n$.

Osservazione. L'algoritmo 5 mostra che, se è possibile fattorizzare n, allora si può risolvere il problema delle radici quadrate modulo n intero.

Volendo parafrasare l'ultima ossevazione, si può dire che la difficoltà del problema delle radici quadrate è minore o uguale alla difficoltà del problema della fattorizzazione. Di questa affermazione vale anche l'opposto, come riportato nell'osservazione successiva.

Osservazione. *(Equivalenza del problema delle radici quadrate e della fattorizzazione)*
La difficoltà della fattorizzazione è minore o uguale alla difficoltà del problema delle radici quadrate. Da qui si evince che i due problemi sono computazionalmente equivalenti.
In altre parole, se si possiede un algoritmo che risolve in un tempo polinomiale uno dei due problemi, è possibile risolvere in un tempo polinomiale anche l'altro.

Dimostrazione. Si supponga di possedere un algoritmo A polinomiale per risolvere il problema delle radici quadrate modulo n. Si vuole arrivare a dimostrare che questo algoritmo può essere utilizzato per fattorizzare un intero n dato.
Scegliere un x intero casuale con $MCD(x, n) = 1$, e calcolare $a = x^2 \bmod n$. Ora, l'algoritmo A si applica ad a e n e ritorna y, radice quadrata di a modulo n.
Se $y = \pm x (\bmod\ n)$, allora la prova fallisce e la procedura sopra è ripetuta con un nuovo x casuale finché $y \neq \pm x (\bmod\ n)$, in tal caso $MCD(x - y, n)$ è

sicuramente un fattore non banale di n (vedi 3.18[2]), cioé o p o q.
Dato che a ha quattro radici quadrate modulo n ($\pm x$ e $\pm z$, con $\pm z \neq \pm x \bmod n$), la probabilità di successo, e insuccesso, per ciascun tentativo è $\frac{1}{2}$ (anche se la probabilità di insuccesso può essere resa piccola a piacere prevedendo più tentativi). Quindi il numero atteso di tentativi prima di trovare un fattore di n è 2 e, di conseguenza, l'algoritmo ha una complessità polinomiale. □

L'equivalenza computazionale del problema delle radici quadrate modulo n e della fattorizzazione è alla base della prima chiave pubblica dimostrabilmente sicura per cifrare e degli algoritmi di firma.

Diversamente, per il problema dei residui quadratici si può dimostrare solo che la difficoltà di questo è minore o uguale alla difficoltà della fattorizzazione.

Osservazione. Sapendo risolvere il problema della fattorizzazione, si può risolvere anche il problema dei residui quadratici.

Dimostrazione. Da n si ricavano p e q tramite un algoritmo A, che risolve la fattorizzazione. Attraverso l'algoritmo 1 si stabilisce se un intero a è residuo quadratico modulo p e q e, se lo è per tutti e due, lo è anche per n (seconda osservazione del paragrafo 1.5). □

Inoltre si può far notare che, se la fattorizzazione di n è difficile, allora non si conosce alcun algoritmo efficiente per risolvere il problema dei residui quadratici che sia migliore del semplice indovinare a caso la risposta corretta. Infatti, se $n = pq$, allora la probabilità di una risposta corretta è $\frac{1}{2}$, perché $|Q_n| = |\tilde{Q}_n|$ (2.155[2]). Si suppone che il problema dei residui quadratici sia difficile quanto la fattorizzazione, sebbene di questo non ci sia la prova.

2.2 Algoritmi a chiave pubblica

Solo a questo punto si può cominciare a parlare di vera e propria crittografia. In questa sezione infatti vedremo alcuni algoritmi per mandare messaggi cifrati, che sfruttano tutto ciò che abbiamo detto finora.
Un sistema di crittografia si dice a chiave pubblica quando sono presenti le seguenti componenti: un insieme M dei messaggi possibili (di norma è lo stesso sia per i testi in chiaro che per quelli cifrati), l'insieme K di tutte le chiavi possibili (o degli interi che formano una chiave), una funzione di cifratura e una di decifrazione che sfruttano le chiavi e lavorano sui messaggi (sono definite, di norma, da M in M). Queste funzioni devono essere una l'inverso dell'altra, devono essere sempre facilmente calcolabili e la funzione di decifrazione non deve poter essere calcolabile da quella di cifratura. All'atto pratico queste funzioni sono molto spesso una lista di passaggi all'interno

di un algoritmo.

Il vantaggio di questo metodo, rispetto alla chiave privata, consiste nel fatto che gli interlocutori non hanno la necessità di possedere un segreto in comune, segreto che potrebbe essere intercettato esso stesso.

Definizione. Per poter parlare agilmente di crittografia, diamo alcune definizioni:

- Per *testo in chiaro* si intende un testo (il più delle volte numerico) che non è stato ancora soggetto a cifratura.
- Con *testo cifrato* si intende, viceversa, un testo che è già stato soggetto a cifratura.
- Una *chiave di cifratura* è un intero, o un piccolo vettore di interi, che serve per crittografare messaggi. La chiave può essere sia pubblica che privata.
- Una *chiave di decifrazione* è un intero, o un piccolo vettore di interi, che serve per decifrare messaggi. Questa chiave è sempre privata.
- Un algoritmo *deterministico* è un algoritmo tale che, data una chiave di cifratura, un particolare testo in chiaro m sarà sempre cifrato nello stesso testo cifrato c.
- Allo stesso modo si chiama *non deterministico* un algoritmo che cifra un testo in chiaro m, idealmente, sempre in un diverso testo cifrato c.

In un algoritmo di cifratura, uno dei punti più spinosi da valutare è sicuramente la sua sicurezza. Un requisito per un minimo livello di sicurezza per un tale algoritmo è che sia difficile, essenzialmente in tutti i casi, per un avversario passivo trovare il testo in chiaro partendo dal testo cifrato. Tuttavia è molto meglio richiedere più forti garanzie di sicurezza.

Partiamo ora dagli algoritmi deterministici. In questa categoria rientrano anche gli algoritmi RSA e di Rabin (quest'ultimo lo tratteremo tra poco).

Uno schema deterministico ha alcuni o tutti i seguenti svantaggi:

- L'algoritmo non è sicuro per ogni messaggio possibile. Per esempio, in RSA, i messaggi 0 e 1 vengono sempre cifrati in loro stessi, cosa che li rende facili da scoprire.

- Succede che sia facile trovare informazioni parziali sul testo in chiaro dal testo cifrato. Per esempio, siano m e $c \equiv m^e \bmod n$ rispettivamente testo in chiaro e testo cifrato in RSA. Allora

$$\left(\frac{c}{n}\right) = \left(\frac{m^e}{n}\right) = \left(\frac{m}{n}\right)^e = \left(\frac{m}{n}\right),$$

  dato che e è dispari. In questo caso un avversario O può quindi ricavare un'informazione su m, ovvero il simbolo di Jacobi $\left(\frac{m}{n}\right)$.

- E' semplice capire se lo stesso messaggio viene mandato più volte.

Si può notare che ogni algoritmo deterministico può essere modificato in modo da diventare non deterministico. Un espediente che funziona per ogni algoritmo consiste nel porre alla fine del messaggio una stringa di bit casuali di lunghezza l, con l fissata. Se l è abbastanza grande, in pratica, gli attacchi elencati sopra non funzionano più. Comunque non è provato che una cifratura del genere sia sicura contro ogni altro tipo di attacco.

Diversamente, la cifratura non deterministica utilizza interi casuali per ottenere un livello di sicurezza provabile e molto alto. Ci sono due nozioni forti di sicurezza che si può cercare di raggiungere.

Definizione. Si prendano due messaggi in chiaro m_1 e m_2, se ne cifri uno e tale cifratura si chiami c.
Un algoritmo di cifratura a chiave pubblica è detto *polinomialmente sicuro* se un avversario O, a cui vengono dati m_1, m_2 e c, non è in grado di stabilire da quale messaggio in chiaro c provenga, in un tempo polinomiale e con una probabilità significativamente maggiore di $\frac{1}{2}$.
Addirittura O potrà scegliersi da solo m_1 ed m_2, ma questo non gli darà nessuna informazione in più (pag.63 [4]).

Definizione. Uno schema di cifratura a chiave pubblica è detto *semanticamente sicuro* se, per ogni possibile messaggio, qualunque cosa un avversario passivo possa calcolare in un tempo polinomiale per trovare il testo in chiaro dato il testo cifrato, ciò può essere calcolato in un tempo polinomiale anche senza il testo cifrato.

Intuitivamente, uno schema di cifratura a chiave pubblica è semanticamente sicuro se dal testo cifrato non si può ottenere nessuna informazione parziale sul testo in chiaro in un tempo polinomiale.

Osservazione. *(sicurezza perfetta e sicurezza semantica)* Nella teoria di Shannon (1.13.3.(i)[2]), un algoritmo di cifratura ha sicurezza perfetta se

un avversario passivo, anche con risorse di calcolo illimitate, non può scoprire nulla sul testo in chiaro a partire dal testo cifrato, ad eccezione forse della sua lunghezza. Il poco utilizzo pratico di questa nozione sta nel fatto che la sicurezza perfetta non può mai essere raggiunta a meno che la chiave non sia almeno lunga quanto il messaggio e che venga cambiata per ogni messaggio. Di conseguenza, la nozione di sicurezza semantica può essere vista come la versione polinomiale della sicurezza perfetta: un avversario passivo con risorse polinomiali non può scoprire nulla sul testo in chiaro dal testo cifrato. E, in questo caso, esistono algoritmi di cifratura semanticamente sicuri con le chiavi molto più corte dei messaggi.

Osservazione. Benché la definizione di sicurezza semantica appaia più restrittiva della sicurezza polinomiale, si può dimostrare che le due sono perfettamente equivalenti.

2.2.1 Algoritmo a chiave pubblica di Rabin

L'algoritmo a chiave pubblica di Rabin, che qui presentiamo, è particolarmente importante in quanto si tratta del primo esempio di cifratura con una sicurezza dimostrabile; infatti il problema di decifrare un testo cifrato da questo algoritmo è computazionalmente equivalente al problema dei radici quadrate e alla fattorizzazione. In questo, l'algoritmo di Rabin si discosta da quello di RSA non essendo, per ora, stato provato che la difficoltà di RSA sia equivalente a quella della fattorizzazione, sebbene la si ritenga tale.

Algoritmo 7. *(Algoritmo per generare la chiave per la cifratura a chiave pubblica di Rabin)*
Riassunto: ogni utente crea una propria chiave pubblica e una corrispondente chiave privata.
Ciascun utente A segue questi punti:

1. Selezionare due primi casuali grandi (e distinti) p e q, all'incirca dello stesso ordine di grandezza.
2. Calcolare $n = pq$.
3. La chiave pubblica di A è n, la chiave privata è (p, q).

Algoritmo 8. *(Algoritmo a chiave pubblica di Rabin)*
Riassunto: B cifra un messaggio per A, il quale lo decifra.

I *Cifratura.* B compie i seguenti passaggi:

1. Ottenere la chiave pubblica di A.

2. Scrivere il messaggio come un intero m, con $m \in 0, 1, \ldots, n-1$
3. Calcolare $c = m^2 \bmod n$.
4. Mandare il testo cifrato c ad A.

II *Decifrazione.* Per recupare il testo in chiaro m, A deve:

1. Usare l'algoritmo 5 per trovare le radici quadrate m_1, m_2, m_3 e m_4 di c modulo n.
2. Il messaggio inviato è uno di questi quattro. A decide qual'è attraverso l'uso della ridondanza (la ridondanza viene spiegata in una delle osservazioni sottostanti).

Osservazione. *(Sicurezza dell'algoritmo a chiave pubblica di Rabin)*

- Assumendo che il problema della fattorizzazione sia intrattabile, la sicurezza dell'algoritmo a chiave pubblica di Rabin è provata contro un avversario passivo, perché il fulcro dell'algoritmo è il problema dell'estrazione di radici quadrate, equivalente alla fattorizzazione.

- Diversamente, il cifrario a chiave pubblica di Rabin non resiste a un attacco con testo cifrato scelto (confrontare con oss. 8.14.(*ii*)[2]). Un attacco del genere può essere fatto in questo modo:
  un avversario O sceglie un intero casuale $m \in \mathbb{Z}_n^*$ e calcola $c \equiv m^2 \bmod n$, facendolo poi entrare nella macchina di decifrazione di A e ricavandone un certo testo y. Dato che la macchina di A non conosce m, scelto casualmente, y non è necessariamente m. Con probabilità $\frac{1}{2}$, $y \not\equiv \pm m \bmod n$, nel qual caso $MCD(m-y, n)$ è uno dei fattori primi di n. Se $y \equiv \pm m \bmod n$, allora l'attacco va ripetuto con un altro m.
  Questo attacco è un'applicazione della dimostrazione dell'equivalenza tra la fattorizzazione e il problema delle radici quadrate (ultima osservazione del capitolo 2.1), dove la macchina di decifrazione di A viene utilizzata al posto dell'ipotetico algoritmo polinomiale che risolve il problema delle radici quadrate.

- Il cifrario a chiave pubblica di Rabin è sensibile ad attacchi simili a quelli per RSA, descritti in 8.2.2.(*ii*)[2], 8.2.2.(*iii*)[2] e 8.2.2.(*v*)[2]. Come nel caso di RSA, gli attacchi (*ii*) e (*iii*) possono essere aggirati con il metodo del *salting*[1], mentre l'attacco (*v*) può essere evitato aggiungendo appropriate ridondanze prima della cifratura.

[1]Il *salting* consiste nell'aggiungere a una qualsiaci stringa di bit (tra cui i messaggi) un numero fisso di bit casuali per rendere un processo, ad esempio una cifratura, non deterministico. Se il numero di bit aggiunti non è fisso, si parla invece di *padding*.

Osservazione. *(Uso della ridondanza)*

- L'algoritmo a chiave pubblica di Rabin presenta uno svantaggio, ovvero che il ricevente deve riuscire a riconoscere il corretto testo in chiaro tra i possibili quattro. Una soluzione a questa ambiguità consiste nell'aggiungere una ridondanza prestabilita al messaggio, prima della cifratura (ad esempio una ripetizione degli ultimi 64 bit del messaggio). Allora, con una probabilità molto alta, solamente una delle quattro radici quadrate avrà questa ridondanza e sarà così indentificabile dal ricevente. Se nessuna delle quattro radici presenta la ridondanza cercata, il ricevente potrà scartare il messaggio come fasullo.

- Se utilizziamo la ridondanza come specificato sopra, l'algoritmo di Rabin non è più sensibile all'attacco con testo cifratura scelto dell'osservazione precedente.
  Infatti, se O sceglie un messaggio m con la dovuta ridondanza e manda $c = m^2 \bmod n$ alla macchina di decifrazione di A, molto probabilmente la macchina darà coma valore di ritorno il testo in chiaro m stesso (dato che le altre tre radici, con molta probabilità, non avranno la ridondanza), senza dare nessuna nuova informazione. D'altra parte se O sceglie un m senza la dovuta ridondanza, con molta probabilità nessuna delle quattro radici di $c \equiv m^2 \bmod n$ avrà la ridondanza in questione. In questo caso, la macchina fallirà nel decifrare c e non darà nessuna risposta ad O.
  Quindi, la cifratura a chiave pubblica di Rabin, appositamente modificata aggiungendo ridondanze, è effettivamente interessante a livello pratico.
  Occorre notare che, usando la ridondanza, non è più valida l'equivalenza con la fattorizzazione, che si fonda sul fatto che non è possibile riconoscere quale dei quattro testi è quello cercato.

Esempio. *Generazione delle chiavi.* Un utente A sceglie i primi $p = 277$ e $q = 331$ per calcolare $n = pq = 91687$. La chiave pubblica di A è n, quella privata (p, q).
Cifratura. Supponiamo che, come ridondanza, sia richiesto che gli ultimi 6 bit del messaggio originale vengano ripetuti prima della cifratura. Perciò, per cifrare il messaggio da 10 bit $m = 1001111001$, B compone un messaggio da 16 bit $m' = 1001111001111001$ che, in base 10, è $m' = 40569$.
B calcola poi $c = m^2 \bmod n$, ovvero $c = 40569^2 \bmod 91687 \equiv 62111$, e lo manda ad A.
Decifrazione. A usa l'algoritmo 5 per calcolare le quattro radici di $c \bmod n$:

$$m_1 = 69654,\ m_2 = 22033,\ m_3 = 40569,\ m_4 = 51118$$

che in base 2 diventano

$$m_1 = 10001000000010110,\ m_2 = 101011000010001,$$

$$m_3 = 1001111001111001,\ m_4 = 1100011110101110.$$

Dato che solo m_3 possiede la ridondanza richiesta, A decifra c in m_3 e, togliendo gli ultimi sei bit, ritrova il messaggio originale.

Osservazione. *Efficienza*
La cifratura di Rabin è molto rapida perché include un solo elevamento al quadrato in modulo; perciò questa cifratura è più veloce di quella di RSA che, cifrando con $e = 3$, esige invece una moltiplicazione e un'elevamento al quadrato, in modulo. La decifrazione di Rabin è più lenta della sua cifratura, ma ha circa la stessa velocità della decifrazione di RSA.

2.2.2 Cifratura non deterministica di Goldwasser-Micali

L'algoritmo di Goldwasser-Micali è un cifrario non deterministico a chiave pubblica che risulta essere semanticamente sicuro, assumendo l'intrattabilità del problema dei residui quadratici.

Per prima cosa presentiamo una piccola osservazione che ci servirà immediatamente.

Osservazione. *(Per trovare gli pseudoquadrati)*
Uno pseudoquadrato y modulo n può essere trovato come segue. Per prima cosa trovare un non residuo quadratico a modulo p e un non residuo quadratico b modulo q. (vedi osservazione 2.151[2]). Ora usare l'algoritmo di Gauss (2.121[2]) per calcolare un intero y, $0 \leq y \leq n-1$, che soddisfa simultaneamente le congruenze $y \equiv_p a$ e $y \equiv_q b$. Dato che y, $y \equiv_p a$, è un non residuo quadratico modulo p, è anche un non residuo quadratico modulo n (proposizione del paragrafo 1.4). In più, dalle proprietà dei simboli di Legendre e Jacobi, $(\frac{y}{n}) = (\frac{y}{p})(\frac{y}{q}) = (-1)(-1) = 1$.
Ne segue che y è uno pseudoquadrato modulo n.

Mostriamo ora l'algoritmo per generare le chiavi e l'algoritmo vero e proprio di Goldwasser-Micali.

Algoritmo 9. *(Algoritmo di generazione di una chiave per la cifratura non deterministica di Goldwasser-Micali)*
Riassunto: ogni utente crea una propria chiave pubblica e una corrispondente chiave privata.
Ciascun utente A segue questi punti:

1. Selezionare due grandi primi casuali (e distinti) p e q, all'incirca dello stesso ordine di grandezza.
2. Calcolare $n = pq$.
3. Selezionare, usando l'osservazione precedente, un $y \in Z_n$ tale che y sia un non residuo quadratico modulo n e il simbolo di Jacobi sia $(\frac{y}{n}) = 1$ (y è uno pseudo-quadrato modulo n).
4. La chiave pubblica di A è (n, y); la chiave privata di A è (p, q).

Algoritmo 10. *(Algoritmo non deterministico a chiave pubblica di Goldwasser-Micali)*
Riassunto: B cifra un messaggio m per A, che A decifra.

I *Cifratura.* B svolge i seguenti passi:

1. Leggere la chiave pubblica di A (n, y).
2. Rappresentare il messaggio m come stringa di bit in cifre binarie, $m = m_1 m_2 \cdots m_t$ di lunghezza t.
3. Per i che va da 1 a t:
   (a) Scegliere $x \in \mathbb{Z}_n^*$ in modo uniformemente casuale.
   (b) Se $m_i = 1$ allora porre $c_i = yx^2 \bmod n$; altrimenti porre $c_i = x^2 \bmod n$.
4. Mandare ad A $c = (c_1, \ldots, c_t)$.

II *Decifrazione.* Per recuperare il testo in chiaro m da c, A deve seguire questi passi:

1. Per i che va da 1 a t:
   (a) Calcolare i simboli di Legendre $(\frac{c_i}{p})$ e $(\frac{c_i}{q})$ (usando l'algoritmo 2.149[2]).
   (b) Se $(\frac{c_i}{p}) = 1$ e $(\frac{c_i}{q}) = 1$ allora porre $m_i = 0$; altrimenti porre $m_i = 1$.
2. Il messaggio decriptato è $m = m_1 m_2 \cdots m_t$.

Dimostrazione. Decifrazione
Se $m_i = 0$, allora $c_i = x^2 \bmod n$ è un residuo quadratico modulo n. Se $m_i = 1$, allora y è uno pseudo-quadrato modulo n e $c_i = yx^2 \bmod n$ è anch'esso uno pseudoquadrato modulo n. Dalla proposizione del paragrafo 1.4, c_i è un residuo quadratico modulo n *se e solo se* c_i è un residuo quadratico modulo p e modulo q o, equivalentemente, $(\frac{c_i}{p}) = 1$ e $(\frac{c_i}{q}) = 1$. Dato che A conosce p e q, può calcolare questi simboli di Legendre e da qui recuperare m_i. □

Osservazione. *(Sicurezza della cifratura probabilistica di Goldwasser-Micali)*
Avendo scelto x in modo uniformemente casuale in $\mathbb{Z}_n^*$, $x^2 \bmod n$ è un residuo quadratico casuale modulo n, e $yx^2 \bmod n$ è uno pseudoquadrato casuale modulo n. Perciò un avversario non vede altro che residui quadratici e pseudoquadrati modulo n casuali. Assumendo che il problema dei residui quadratici sia difficile, l'avversario non può fare nulla eccetto tirare a indovinare ogni bit del messaggio. In altre parole, se il problema dei residui quadratici è difficile, allora l'algoritmo di cifratura non determimistica di Goldwasser-Micali è semanticamentre sicuro.

Osservazione. *(Espansione del messaggio)*
Un grande svantaggio dell'algoritmo di Goldwasser-Micali è l'espansione del messaggio di un fattore $\lg n$. Qualche espansione del messaggio è inevitabile in un algoritmo di cifratura non deterministico, perché ci sono molti testi cifrati corrispondenti a ciascun testo in chiaro.
L'algoritmo non deterministico di Blum-Goldwasser (che vedremo tra poco) è un miglioramento di Goldwasser-Micali; in tale algoritmo il testo in chiaro è espanso solo di un fattore costante.

2.2.3 Generatore di bit pseudocasuali di Blum-Blum-Shub

Apriamo ora una veloce parentesi, che ci servirà per l'algoritmo non deterministico a chiave pubblica di Blum-Goldwasser, e parliamo del generatore di bit pseudocasuali Blum-Blum-Shub (conosciuto anche come generatore $''x^2 \bmod n''$ o generatore BBS). Si dimostra che, con una piccola modifica, questo generatore è crittograficamente sicuro, o CSPRBG[2], sotto l'ipotesi che la fattorizzazione sia un problema difficile. La definizione di questa sicurezza viene data subito sotto.

Definizione. Un generatore di bit pseudocasuali (supponendo la difficoltà della fattorizzazione) si dice *crittograficamente sicuro*, o *CSPRBG*, se passa il test del bit successivo: ciò accade se non esiste alcun algoritmo polinomiale che, prendendo in imput l bit di una sequenza s, possa scoprire l'$(l+1)$-esimo bit di s con una probabilità significativamente superiore a $\frac{1}{2}$.

Algoritmo 11. *Generatore di bit pseudocasuali Blum-Blum-Shub*
Riassunto: viene generata una sequenza di bit $z_1, z_2, \ldots, z_l$ di lunghezza l.

[2]La sigla CSPRBG si trova usata in letteratura e significa ″Cryptographically Secure Pseudo-Random Number Generator″, ovvero generatore di numeri pseudo-casuali crittograficamente sicuro

1. Generare due primi grandi segreti p e q casualmente (vedere 8.8[2]), ciascuno dei quali congruo a 3 modulo 4, e calcolare $n = pq$.

2. Selezionare un intero casuale s (il seme) nell'intervallo $[1, n-1]$ tale che $MCD(s, n) = 1$ e calcolare $x_0 = s^2 \bmod n$.

3. Per i che va da 1 a l svolgere:

   - $x_i = x_{i-1}^2 \bmod n$.
   - Prendere come z_i l'ultimo bit di x_i.

4. La sequenza finale è $z_1, z_2, \ldots, z_l$.

Osservazione. *(efficienza del metodo Blum-Blum-Shub)*
Il costo del metodo è di un'elevazione al quadrato in modulo per ogni bit generato. Il metodo risulta quindi efficiente.

2.2.4 Cifratura non deterministica di Blum-Goldwasser

Vediamo ora un ultimo algoritmo di cifratura non deterministico a chiave pubblica: quello di Blum-Goldwasser. Si tratta del più efficiente algoritmo non deterministico conosciuto ed è comparabile all'algoritmo di cifratura di RSA sia in termini di velocità, che in termini di espansione del messaggio.
Si dimostra che anche questo algoritmo è semanticamente sicuro (assumendo l'intrattabilità del problema della fattorizzazione), anche se è vulnerabile ad un attacco con testo cifrato scelto, come vedremo più avanti.
Questo algoritmo opera come segue:
Blum-Blum-Shub genera una sequenza di bit pseudo-casuali che è quindi sommata (modulo 2) con il testo in chiaro. La risultante sequenza di bit, insieme con una cifratura del seme casuale usato, è trasmessa al ricevente che usa la sua chiave privata per recuperare il seme e successivamente ricostruire la sequenza di bit pseudocasuali e, infine, il testo in chiaro.

Algoritmo 12. *(Generazione di una chiave per la cifratura non deterministica di Blum-Goldwasser)*
Riassunto: un soggetto A crea una chiave pubblica e una corrispondente chiave privata.
A effettua questi passi:

1. Scegliere due primi grandi (distinti) casuali p e q, ciascuno congruente a 3 modulo 4.

2. Calcolare $n = pq$.

3. Usare l'algoritmo euclideo esteso (2.107[2]) per trovare due interi a e b tali che $ap + bq = 1$.

4. La chiave pubblica di A è n; la sua chiave privata è (p,q,a,b).

Algoritmo 13. *(Algoritmo non deterministico a chiave pubblica di Blum-Goldwasser)*
Riassunto: B cifra un messaggio m per A, che A decifra.

I *Cifratura.* B esegue le seguenti istruzioni:

1. Ottenere n, chiave pubblica di A.
2. Siano $k = \log n$ e $h = \log k$. Rappresentare un messaggio m come una stringa: $m = m_1 m_2 \cdots m_t$ di lunghezza t, dove ogni stringa binaria m_i abbia lunghezza h.
3. Prendere come seme x_0, un residuo quadratico modulo n casuale (questo può essere fatto selezionando un intero casuale $r \in \mathbb{Z}_n^*$ e ponendo $x_0 = r^2 \bmod n$).
4. Per i che va da 1 a t svolgere i passi seguenti:
   (a) Calcolare $x_i = x_{i-1}^2 \bmod n$.
   (b) Porre come p_i gli ultimi h bit di x_i.
   (c) Calcolare $c_i = p_i \oplus m_i$ (operazione in $\mathbb{Z}_2^h$).
5. Calcolare $x_{t+1} = x_t^2 \bmod n$.
6. Mandare il messaggio cifrato $c = (c_1, c_2, \ldots, c_t, x_{t+1})$ ad A.

II *Decifrazione.* Per recuperare il messaggio in chiaro m da c, A deve fare i passi:

1. Calcolare $d_1 = ((p+1)/4)^{t+1} \bmod (p-1)$.
2. Calcolare $d_2 = ((q+1)/4)^{t+1} \bmod (q-1)$.
3. Calcolare $u = x_{t+1}^{d_1} \bmod p$.
4. Calcolare $v = x_{t+1}^{d_2} \bmod q$.
5. Calcolare $x_0 = vap + ubq \bmod n$.
6. Per i da 1 a t:
   (a) Calcolare $x_i = x_{i-1}^2 \bmod n$.
   (b) Porre come p_i gli ultimi h bit di x_i.
   (c) Calcolare $m_i = p_i \oplus c_i$.

Dimostrazione. Si dimostra qui la correttezza della decifrazione.
Dato che x_t è un residuo quadratico modulo n, è anche un residuo quadratico modulo p. Da qui,

$$x_t^{(p-1)/2} \equiv 1 \bmod p.$$

Osservare che

$$x_{t+1}^{(p+1)/4} \equiv (x_t^2)^{(p+1)/4} \equiv x_t^{(p+1)/2} \equiv x_t^{(p-1)/2} x_t \equiv x_t \bmod p.$$

Similmente,

$$x_t^{(p+1)/4} \equiv x_{t-1} \bmod p$$

e così

$$x_{t+1}^{((p+1)/4)^2} \equiv x_{t-1} \bmod p.$$

Ripetendo questo procedimento si trova:

$$u \equiv x_{t+1}^{d_1} \equiv x_{t+1}^{((p+1)/4)^{t+1}} \equiv x_0 \bmod p.$$

Analogamente, si arriva anche a

$$v \equiv x_{t+1}^{d_2} \equiv x_0 \bmod q.$$

Infine, dato $ap + bq = 1$,

$$vap + ubq \equiv x_0 \bmod p$$

e

$$vap + ubq \equiv x_0 \bmod q.$$

Da qui si ricava $x_0 = vap+ubq \bmod n$, e questo permette ad A di recuperare lo stesso seme casuale che B usò nella cifratura e, di conseguenza, di recuperare anche il testo in chiaro originale. □

Esempio. *Cifratura di Blum-Goldwasser con primi piccoli*
Generazione delle chiavi. L'utente A seleziona i primi $p = 499$ e $q = 547$, entrambi congruenti a 3 modulo 4, e calcola $n = pq = 272953$. Usando l'algoritmo euclideo esteso, A poi trova gli interi $a = -57$ e $b = 52$, che soddisfano $ap+bq = 1$. La chiave pubblica di A è quindi $n = 272953$, mentre la sua chiave privata è (p, q, a, b).
Cifratura. I parametri h e k sono $h = 18$ e $k = 4$. Il messaggio m che B vuole inviare corrisponde a $t = 5$ stringhe: $m = m_1m_2m_3m_4m_5$ con $m_1 = 1001$, $m_2 = 1100$, $m_3 = 0001$, $m_4 = 0000$, $m_5 = 1100$. Ora B prende un residuo quadratico casuale $x_0 = 159201 (= 399^2 \bmod n)$ e calcola la tabella qui riportata

i	$x_i = x_{i-1}^2 \bmod n$	p_i	$c_i = p_i \oplus m_i$
1	180539	1011	0010
1	193932	1100	0000
3	245613	1101	1100
4	130286	1110	1110
5	40632	1000	0100

e $x_6 = x_5^2 \bmod n = 139680$.
Infine B manda ad A il testo cifrato

$$c = (0010, 0000, 1100, 1110, 0100, 139680).$$

Decifrazione. Per decifrare c, A calcola

$$\begin{aligned} d_1 &= ((p+1)/4)^6 \bmod (p-1) = 463 \\ d_2 &= ((q+1)/4)^6 \bmod (q-1) = 337 \\ u &= x_6^{463} \bmod p = 20 \\ v &= x_6^{337} \bmod q = 24 \\ x_0 &= vap + ubq \bmod n = 159201 \end{aligned}$$

Infine, A utilizza x_0 per ritrovare gli x_i e i p_i di prima; dopodiché li utilizza questi per ricostruire i tesi in chiaro m_i attraverso l'addizione modulo 2 tra i p_i e gli c_i.

Osservazione. *(sicurezza della cifratura non deterministica di Blum-Goldwasser)*

- Per prima cosa si può osservare che n è un intero di Blum, ovvero un prodotto di due primi entrambi congrui a 3 modulo 4. Un avversario vede il residuo quadratico x_{t+1}. Assumendo che la fattorizzazione sia difficile, gli ultimi h bit di x_t sono simultaneamente sicuri (definizione 3.82[2], vale a dire che trovare gli ultimi h bit di x_t è altrettanto difficile che trovare tutto x_t). Inoltre un avversario non può fare nulla di meglio che tirare a indovinare i bit pseudocasuali p_i, con $1 \leq i \leq t$. In altre parole si può affermare che se la fattorizzazione è difficile, allora l'algoritmo di Blum-Goldwasser è semanticamente sicuro. Si deve notare, comunque, che per un modulo n di lunghezza in bit fissata questo fatto non è più vero e che l'algoritmo si dimostra essere solo computazionalmente sicuro (ovvero sicuro supponendo la l'impossibilità computazionale della macchina di risolvere il problema).

- A partire dal 1996, il modulo n dovrebbe essere lungo almeno 1024 bit se è desiderata una sicurezza a lungo termine. Se n è un intero di 1025 bit, allora $k = 1024$ e $h = 10$.

- Come per l'algoritmo di cifratura di Rabin (Algoritmo 8), anche quello di Blum-Goldwasser è vulnerabile all'attacco con testo cifrato scelto che recupera la chiave privata dalla chiave pubblica.

Osservazione. *(efficacia di Blum-Goldwasser)*

- Come avevamo già anticipato, a differenza della cifratura di Goldwasser-Micali, un testo cifrato con lo schema di Blum-Goldwasser è solo più lungo del testo in chiaro di un numero costante di bit, vale a dire $k+1$ (la grandezza in bit dell'intero x_{t+1}).

- Il processo di cifratura è piuttosto efficiente, esso richiede solo una moltiplicazione in modulo per cifrare h bit di testo in chiaro. Comparandoli, il processo di cifratura di RSA (Algoritmo 8.3[2]) richiede una esponenziazione ($m^e \bmod n$) per cifrare k bit di testo in chiaro.
  Assumendo che il parametro e sia scelto casualmente e assumendo che una (non ottimizzata) esponenziazione utilizzi $3k/2$ moltiplicazioni modulari, questo si traduce in una velocità di cifratura per RSA di 2/3 di bit per moltiplicazione modulare, contro gli h di Blum-Goldwasser. Se si scegliesse un valore speciale per e, come ad esempio $e = 3$, allora la velocità della cifratura con RSA sarebbe più veloce della cifratura di Blum-Goldwasser.

- La decifratura di Blum-Goldwasser (seconda parte dell'algoritmo 13) è anch'essa piuttosto efficiente, dato che richiede una esponenziazione modulo $p-1$, una modulo $q-1$, una modulo p, una modulo q e t moltiplicazioni modulo n per decifrare ht bit di testo cifrato (il tempo per calcolare il passo II.5 è irrilevante). Comparandole, la decifratura di RSA richiede una esponenziazione modulo n (che può essere fatta attraverso una modulo p e una modulo q) per decifrare k bit di testo cifrato. Inoltre, per messaggi corti ($< k$ bit), la decifratura di Blum-Goldwasser è leggermente più lenta di quella di RSA mentre, per messaggi lunghi, è Blum-Goldwasser a essere il più veloce.

Bibliografia

[1] M.W. Baldoni, C.Ciliberto, G.M. Piacentini Cattaneo, *Aritmetica, Crittografia e Codici*, 2006, ed. Springer

[2] A. Menezes, P. van Oorschot e S. Vanstone, *Handbook of Applied Cryptography*, 1996, ed. CRC Press

[3] Wade Trappe, Lawrence C. Washington, *Crittografia con Elementi di Teoria dei Codici*, 2009, ed. Pearson Education Italia

[4] Jonathan Katz, Yehuda Lindell, *Introduction to Modern Cryptography*, 2007, ed. CRC Press

Ringraziamenti

Ringrazio molto il professor Aliffi per la sua pazienza e la sua disponibilità, nonostante i tempi molto stretti. Ringrazio tutti coloro che, direttamente o indirettamente, mi hanno aiutato a realizzare questo elaborato. Infine ringrazio quella persona che mi è stata accanto anche senza saperlo.