

**LA GENERAZIONE della CHIAVE PRIVATA
nell'algoritmo crittografico RSA a chiave pubblica**
(How generating PRIVATE KEY in RSA cryptosystem algorithm)

Chiave: Privata : 215848789810065513208302874205921122935334449334942657295536075214506868195979917314395792274
35316895627412498621114596913065463429612455215420383072827555777122991700187967008591109257601820645413373
921927977000310981480265384701404949044705674957052534439113216316015300413232895718188959262368760601426187

Sommario: scopo di questo articolo è quello di descrivere la realizzazione della CHIAVE PRIVATA nell'algoritmo RSA a chiave pubblica con l'utilizzo delle congruenze lineari. In effetti l'uso delle congruenze lineari viene impiegato non solo nel suddetto algoritmo ma anche in diversi altri metodi crittografici, come ad esempio nell'algoritmo DSA e nell'algoritmo di EL GAMAL per la generazione e la verifica della firma digitale. Si inizierà pertanto con l'illustrare un algoritmo per la risoluzione della congruenza lineare $Ax \equiv C \pmod{B}$ o dell'equazione lineare diofantea $A \cdot x - B \cdot y = C$ dove A , B e C sono numeri interi qualsiasi positivi o negativi. Come caso particolare si prenderà in considerazione la congruenza del tipo $Ax \equiv 1 \pmod{B}$ per il calcolo della Chiave Privata. Dopo una breve premessa viene descritto lo sviluppo di un numero razionale in frazione continua arrivando al calcolo del MCD di due numeri. Si passa quindi ad introdurre l'algoritmo riguardante la risoluzione di questo tipo di equazioni o congruenze attraverso i seguenti passi: calcolo delle ridotte; condizioni di risolvibilità dell'equazione; risoluzione delle equazioni $A \cdot x - B \cdot y = \pm 1$ e quindi delle equazione più generale

$A \cdot x - B \cdot y = C$. Viene data poi una succinta panoramica dell'impiego negli algoritmi crittografi citati di questo tipo di congruenza, illustrando poi in dettaglio una sua applicazione riguardante il calcolo della Chiave Privata nell'algoritmo RSA..Per avere dei risultati concreti si sono realizzati due programma in linguaggio Qbasic. Il primo riguarda, sia soluzioni della congruenza lineare generica con valori numerici di A , B e C qualsiasi, ma ciascuno in valore assoluto $< 10^5$, sia il calcolo della chiave privata nell'algoritmo RSA sempre con valori numerici $< 10^5$ per ogni primo impiegato, in quanto ci si limita nei calcoli di questo programma all'uso della doppia precisione..

I risultati ottenibili per la generazione della chiave privata sono quindi da ritenersi di significato unicamente dimostrativo.. Il secondo programma è dedicato espressamente per l'RSA al calcolo di chiavi private, costituite anche da centinaia di cifre decimali. Con l'impiego pertanto di quest'ultimo programma, in cui si utilizza una aritmetica a precisione multipla, si possono ottenere valori di Chiavi Private riguardanti un loro effettivo e reale impiego nel campo della Crittografia.

Abstract: the aim of this paper is to illustrate the PRIVATE KEY implementation in the RSA public key cryptographic method by using linear congruences. Indeed their employment is used not only in the aforesaid algorithm, but also in other important cryptographic systems, such as the DSA and the ElGamal, regarding the Digital Signature generation and verification. So we start explaining an algorithm for solution of the linear congruence $Ax \equiv C \pmod{B}$ or linear diophantine equation $A \cdot x - B \cdot y = C$ where A , B , C are whole positive or negative numbers. After a concise introduction we explain the continued fraction expansion of rational number, attaining the GCD of two numbers. Then we illustrate the algorithm for the solution of the linear congruences by the next steps: the convergents computation; the equation resolvability conditions; the $A \cdot x - B \cdot y = \pm 1$ and $A \cdot x - B \cdot y = C$ solutions.

We give also a short survey of the linear congruences employed in the mentioned cryptographic methods and then we explain the particular computation regarding the private key of the RSA algorithm. We have implemented two programs in Qbasic language. The former considers either the linear congruence solution with A , B , C , each $< 10^5$ or the private key computation in the RSA algorithm with every prime numerical value $< 10^5$. The private key values in this case are only demonstrative values.

The second program considers expressly the private keys computation, composed by several tens or some hundreds of digits. So this program permits, by using a multiple - precision arithmetic, the achievement of numeric value for private keys, that can be of actual utilization in the cryptographic field.

1. Premessa

Per illustrare la realizzazione della **chiave privata** nell' algoritmo RSA risulta necessario illustrare innanzitutto un algoritmo dedicato alla risoluzione delle equazioni lineari diofantee $A \cdot x - B \cdot y = C$ dove A , B e C sono numeri interi qualsiasi positivi o negativi.

Risolvere queste equazioni significa trovare per le incognite x e y dei valori numerici interi che la soddisfano.

Ma perché la risoluzione di questo tipo di equazione può interessare il campo della crittografia?

Basterà per ora accennare che in diversi importanti algoritmi crittografici a chiave pubblica quali l' algoritmo RSA e l' algoritmo DSA (Digital Signature Algorithm) si deve risolvere l' equazione $A \cdot x - B \cdot y = 1$ o l' equivalente congruenza lineare $Ax \equiv 1 \pmod{B}$ con A e B interi positivi per il calcolo di alcune grandezze o parametri riguardanti sia la generazione della **Chiave Privata** nell' algoritmo RSA, sia la generazione e la verifica della **Firma Digitale** nell' algoritmo DSA. Si rimanda nel seguito a maggiori dettagli sull' impiego dell' equazione in questo campo. Sotto il titolo è riportato un esempio di valore numerico di **Chiave Privata** relativo all' algoritmo RSA, valore che può considerarsi di effettivo e reale utilizzo.

Questa nota inizierà illustrando la risoluzione delle equazioni lineari diofantee $A \cdot x - B \cdot y = C$.

2. Sviluppo di un numero razionale in frazione continua

Per poter risolvere l' equazione lineare $A \cdot x - B \cdot y = C$ in questione occorre venire a conoscenza di alcuni argomenti essenziali riguardanti lo sviluppo di un numero razionale in frazione continua, che sono qui di seguito illustrati.

Per i vari tipi di notazioni e simboli utilizzati come pure per la validità delle formule e delle relazioni impiegate si fa riferimento a [Old].

Ogni numero razionale è una frazione della forma $\frac{A}{B}$ con A e B interi e $B \neq 0$

Si dimostra [Old] che ogni frazione, cioè ogni numero razionale lo si può esprimere nella forma seguente:

$$\frac{A}{B} = a_1 + \frac{1}{a_2 + \frac{1}{a_3 + \frac{1}{a_4 + \dots + \frac{1}{a_{n-1} + \frac{1}{a_n}}}}} \quad (1)$$

con un numero limitato di termini a_i (da a_1 a a_n) dove il termine a_1 può essere sia un intero positivo che negativo o nullo e gli altri valori a_i sono degli interi positivi.

I vari a_i si ricavano eseguendo le seguenti divisioni successive:

$$\frac{A}{B} = a_1 + \frac{r_1}{B} \quad \text{con } a_1 \text{ e } r_1 \text{ rispettivamente quoziente e resto della divisione di } A \text{ per } B$$

e quindi $0 < r_1 < B$

$$\frac{B}{r_1} = a_2 + \frac{r_2}{r_1} \quad \text{con } r_2 \text{ resto della divisione di } B \text{ per } r_1 \quad \text{e quindi } 0 < r_2 < r_1$$

$$\frac{r_1}{r_2} = a_3 + \frac{r_3}{r_2} \quad \text{con } r_3 \text{ resto della divisione di } r_1 \text{ per } r_2 \quad \text{e quindi } 0 < r_3 < r_2$$

.....

$$\frac{r_{n-3}}{r_{n-2}} = a_{n-1} + \frac{r_{n-1}}{r_{n-2}} \quad \text{con } r_{n-1} \text{ resto della divisione di } r_{n-3} \text{ per } r_{n-2} \quad \text{e quindi } 0 < r_{n-1} < r_{n-2}$$

$$\frac{r_{n-2}}{r_{n-1}} = a_n + \frac{0}{r_{n-1}} = a_n \text{ con } r_n = 0$$

I resti $r_1, r_2, r_3, \dots, r_{n-1}, r_n$ sono di valore decrescente e costituiscono una successione finita di termini il cui ultimo termine r_n è sempre di valore **0** [Old].

Sarà pertanto finita anche la successione dei termini a_i che prendono il nome di *quozienti parziali*.

La formula (1) si pone per convenzione sotto la seguente forma più pratica e concisa:

$$\frac{A}{B} = [a_1, a_2, a_3, a_4, \dots, a_{n-1}, a_n] \tag{1b}$$

Facciamo un semplice esempio.

Sia da sviluppare in frazione continua il numero razionale $\frac{A}{B} = \frac{1327}{271}$

Eseguendo le divisioni successive come si è sopra indicato si ottengono per i diversi a_i ed r_i i seguenti valori:

$$\begin{array}{ll} a_1 = 4 & r_1 = 243 \\ a_2 = 1 & r_2 = 28 \\ a_3 = 8 & r_3 = 19 \\ a_4 = 1 & r_4 = 9 \\ a_5 = 2 & r_5 = 1 \\ a_6 = 9 & r_6 = 0 \end{array}$$

pertanto si ha: $\frac{A}{B} = \frac{1327}{271} = [4, 1, 8, 1, 2, 9]$ (1c)

Se si volesse invece sviluppare in frazione continua $\frac{271}{1327}$ è facile vedere che si ha:

$\frac{271}{1327} = [0, 4, 1, 8, 1, 2, 9]$ che si differenzia dalla(1c) per un quoziente parziale in più: il primo quoziente che risulta di valore 0.

Questo procedimento delle divisioni successive sopra illustrato viene notoriamente utilizzato con efficacia per il calcolo del Massimo Comun Divisore di due numeri A e B che sarà indicato nel seguito nella seguente maniera: (A,B) . E' questo infatti il ben noto algoritmo Euclideo (Euclidean algorithm) per il calcolo del Massimo Comun Divisore fra due numeri.

In effetti considerati due numeri interi A e B si può dimostrare [Old] che il più piccolo resto non nullo della successione r_1, r_2, r_3, \dots è il loro (A,B) .

3. Equazioni e Congruenze lineari

3.1 Calcolo delle ridotte

Tenendo presente lo sviluppo di $\frac{A}{B}$ consideriamo ora la seguente successione di grandezze c_i che prendono il nome di *ridotte*:

$$\begin{array}{l} c_1 = [a_1]; \quad c_2 = [a_1, a_2]; \quad c_3 = [a_1, a_2, a_3]; \quad c_4 = [a_1, a_2, a_3, a_4]; \dots \\ \dots \dots \dots c_{n-1} = [a_1, a_2, a_3, a_4, \dots, a_{n-1}]; \quad c_n = [a_1, a_2, a_3, a_4, \dots, a_{n-1}, a_n] = \frac{A}{B} \end{array}$$

Esplicitando i due primi termini si ha:

$$c_1 = a_1 = \frac{p_1}{q_1} \text{ dove si è posto } p_1 = a_1 \text{ e } q_1 = 1$$

$$c_2 = a_1 + \frac{1}{a_2} = \frac{a_1 \cdot a_2 + 1}{a_2} = \frac{p_2}{q_2} \quad \text{avendo posto} \quad p_2 = a_1 \cdot a_2 + 1 \quad \text{e} \quad q_2 = a_2$$

Per la successiva ridotta dopo opportuni passaggi e manipolazioni si perviene alla seguente espressione

$$c_3 = a_1 + \frac{1}{a_2 + \frac{1}{a_3}} = \frac{a_1 \cdot a_2 \cdot a_3 + a_1 + a_3}{a_2 \cdot a_3 + 1} = \frac{a_3 \cdot (a_1 \cdot a_2 + 1) + a_1}{a_3 \cdot a_2 + 1} = \frac{a_3 \cdot p_2 + p_1}{a_3 \cdot q_2 + q_1} = \frac{p_3}{q_3} \quad \text{con} \quad p_3 = a_3 \cdot p_2 + p_1 \quad \text{e} \quad q_3 = a_3 \cdot q_2 + q_1$$

analogamente per c_4 e c_5 dopo opportuni passaggi e manipolazioni, si potrà pervenire anche qui rispettivamente ai seguenti risultati:

$$c_4 = \frac{a_4 \cdot p_3 + p_2}{a_4 \cdot q_3 + q_2} = \frac{p_4}{q_4} \quad \text{avendo posto} \quad p_4 = a_4 \cdot p_3 + p_2 \quad \text{e} \quad q_4 = a_4 \cdot q_3 + q_2$$

$$c_5 = \frac{a_5 \cdot p_4 + p_3}{a_5 \cdot q_4 + q_3} = \frac{p_5}{q_5} \quad \text{avendo posto} \quad p_5 = a_5 \cdot p_4 + p_3 \quad \text{e} \quad q_5 = a_5 \cdot q_4 + q_3$$

e così via per tutti le altre ridotte sino a quella in corrispondenza della quale il valore del resto risulta di valore nullo ($r_n = 0$):

$$c_n = \frac{A}{B} = \frac{a_n \cdot p_{n-1} - p_{n-2}}{a_n \cdot q_{n-1} - q_{n-2}} = \frac{p_n}{q_n} \quad \text{con} \quad A = p_n = a_n \cdot p_{n-1} - p_{n-2} \quad \text{e} \quad B = q_n = a_n \cdot q_{n-1} - q_{n-2} \quad (2)$$

In generale si dimostra per induzione [Old] che:

i numeratori p_i e q_i delle ridotte c_i relative alla frazione continua $[a_1, a_2, a_3, a_4, \dots, a_{n-1}, a_n]$ soddisfano le uguaglianze:

$$p_i = a_i \cdot p_{i-1} + p_{i-2} \quad (5)$$

$$q_i = a_i \cdot q_{i-1} + q_{i-2} \quad (6)$$

per $i = 3, 4, 5, \dots, n$

e con i valori iniziali $p_1 = a_1$; $q_1 = 1$; $p_2 = a_2 \cdot a_1 + 1$; $q_2 = a_2$ (7)

Da quanto illustrato si vede pertanto che si può impostare un algoritmo di tipo iterativo, ad esempio con il loop riportato nel riquadro, una volta posti $N = A$; $D = B$ e le condizioni iniziali:

$$i = 2; \quad p_1 = a_1; \quad q_1 = 1; \quad p_2 = a_2 \cdot a_1 + 1; \quad q_2 = a_2$$

```
Inizio loop : i = i + 1
              a_i = ⌊ N / D ⌋ : r_i = N - a_i · D
              p_i = a_i · p_{i-1} + p_{i-2}
              q_i = a_i · q_{i-1} + q_{i-2}
              se r_i = 0 esci dal loop
                N = D
                D = r_i
              p_{i-2} = p_{i-1} : p_i = p_{i-1}
              q_{i-2} = q_{i-1} : q_i = q_{i-1}
              torna a inizio loop
n = i: REM r_n = 0
fine
M.C.D. (A, B) = r_{n-1}
```

Algoritmo per il calcolo dei valori p_i e q_i fino alla iterazione $i = n$ in corrispondenza della quale si ha resto $r_n = 0$

$$4 - \text{Risoluzione della equazione } A \cdot x - B \cdot y = C \quad (3a)$$

4.1 Condizioni di risolvibilità

Si vogliono trovare per x e y i valori interi soddisfacenti la suddetta equazione.

L'equazione può essere scritta così: $A \cdot x = y \cdot B + C$ mettendo con ciò in evidenza che y e C rappresentano rispettivamente il quoziente ed il resto della divisione di $A \cdot x$ per B .

Tale equazione si può esprimere anche come congruenza assumendo in tal caso la seguente forma:

$$A \cdot x \equiv C \pmod{B} \quad (3b)$$

e si enuncia dicendo che $A \cdot x$ è congruo a C modulo B .

Quando si trattano congruenze il valore del modulo B è da considerarsi di valore positivo.

Questa espressione può essere anche messa sotto la seguente forma: $x = C \cdot A^{-1} \pmod{B}$;

per $C = 1$ si ha $x = A^{-1} \pmod{B}$; in questo caso x viene chiamato *inverso moltiplicativo di A*.

Portando C nel primo membro di (3b) si ha $A \cdot x - C \equiv 0 \pmod{B}$; si dice allora che $A \cdot x - C$ è congruo a 0 modulo B .

La prima cosa da appurare è vedere se la (3a) o la (3b) ammettono soluzioni.

L'equazione e quindi la corrispondente congruenza sono risolvibili solo se sussiste la seguente condizione[Old]:

C è divisibile per (A,B)

Questa condizione equivale ad una qualsiasi di queste due condizioni:

A e B sono primi tra loro.

A e B non sono primi tra loro, ma un loro divisore comune è anche divisore di C .

$$4.2 \text{ Risoluzione della equazione } A \cdot x - B \cdot y = 1 \quad (4)$$

Rivolgiamo ora l'attenzione all'equazione $A \cdot x - B \cdot y = 1$ dove A e B possono essere degli interi sia positivi che negativi.

Innanzitutto perché l'equazione sia risolvibile tenendo presenti le condizioni dette sopra, occorre che A e B siano primi tra loro e che quindi sia $(A,B) = 1$ per A e B dello stesso segno e $(A,B) = -1$ per A e B di segno opposto.

Per i valori generici di p_i e di q_i si dimostra sempre per induzione (vedi [Old]) che sussiste la seguente relazione: $p_i \cdot q_{i-1} - p_{i-1} \cdot q_i = (-1)^i$

Applicando l'algoritmo iterativo che è esposto nel riquadro, si esce dal ciclo alla iterazione per cui si ha resto $r_i = 0$; in quest'ultima iterazione che chiameremo iterazione *n-esima* si sono acquisiti i valori di p_{n-1} , q_{n-1} , p_n e q_n per cui vale la relazione $p_n \cdot q_{n-1} - p_{n-1} \cdot q_n = (-1)^n$.

Ma dalla (2) si osserva che $A = p_n$ e $B = q_n$, per cui si perviene alla seguente relazione:

$$p_n \cdot q_{n-1} - p_{n-1} \cdot q_n = A \cdot q_{n-1} - B \cdot p_{n-1} = (-1)^n \quad (5)$$

Se n è pari si ottiene $A \cdot q_{n-1} - B \cdot p_{n-1} = 1$. Si può allora vedere immediatamente dal confronto con la (4) che i valori q_{n-1} e p_{n-1} sono i valori interi rispettivamente di x e di y che soddisfano l'equazione. Tuttavia, per tenere conto che A e B possono essere sia positivi che negativi e quindi anche di segno opposto, gli effettivi

valori soddisfacenti l'equazione sono dati dalle seguenti formule: $x_0 = \frac{q_{n-1}}{(A,B)}$ e $y_0 = \frac{p_{n-1}}{(A,B)}$ con $(A,B) = 1$

se A e B sono dello stesso segno e $(A,B) = -1$ se A e B sono di segno opposto.

Si può mostrare [Old] poi che anche valori di x del tipo $x_0 + B \cdot k$ e i corrispondenti valori di y del tipo $y_0 + A \cdot k$ dove k è un intero positivo o negativo qualsiasi soddisfano anch'essi l'equazione (4). Pertanto tutti gli infiniti valori risolutivi di x e di y si possono ottenere dalle seguenti formule:

$$x = x_0 \pm B \cdot k; \quad y = y_0 \pm A \cdot k \quad \text{con } k=1, 2, 3, \dots$$

Qualora n fosse dispari si può procedere modificando lo sviluppo (1b) nella seguente maniera come mostrato in [Old]:

$$\frac{A}{B} = [a_1, a_2, a_3, a_4, \dots, a_{n-1}, a_n] = [a_1, a_2, a_3, a_4, \dots, a_{n-1}, a_n - 1, 1]$$

senza alterarne il valore, riportando però ad un numero pari lo sviluppo dei quozienti parziali.

4.2 Risoluzione dell'equazione $A \cdot x - B \cdot y = -1$

L'algoritmo per risolvere questa equazione è analogo a quello utilizzato per il termine noto di valore +1.

Pervenendo alla n -esima iterazione alla relazione $p_n \cdot q_{n-1} - p_{n-1} \cdot q_n = A \cdot q_{n-1} - B \cdot p_{n-1} = (-1)^n$ se n è di valore dispari si ottengono $x_0 = q_{n-1}$ e $y_0 = p_{n-1}$ quali soluzioni dell'equazione

Se n risultasse pari si procede anche qui scomponendo il quoziente parziale a_n nei due quozienti parziali $a_n - 1$ e 1 senza alterare il valore dello sviluppo, riportando però ad un valore dispari il numero dei quozienti parziali.

Per tenere conto che A e B possono essere di segno opposto, gli effettivi valori che soddisfano l'equazione sono dati dalle seguenti formule :

$$x_0 = \frac{q_{n-1}}{(A, B)} \quad \text{e} \quad y_0 = \frac{p_{n-1}}{(A, B)}$$

dove risulta $(A, B) = 1$ se A e B sono dello stesso segno e $(A, B) = -1$ se A e B sono di segno opposto.

4.3 Risoluzione dell'equazione $A \cdot x - B \cdot y = C$

Una volta risolta la $A \cdot x - B \cdot y = 1$ con la sua soluzione particolare x_0 e y_0 è facile vedere che per l'equazione generale $A \cdot x - B \cdot y = C$ con A, B, C interi positivi o negativi, la corrispondente soluzione particolare è :

$$x_{00} = C \cdot x_0 = C \cdot \frac{q_{n-1}}{(A, B)} ; \quad y_{00} = C \cdot y_0 = C \cdot \frac{p_{n-1}}{(A, B)} \quad \text{e tutti gli altri infiniti valori di } x \text{ e di } y \text{ che la}$$

soddisfano sono dati da $x_k = x_{00} + B \cdot k ; \quad y_k = y_{00} + A \cdot k$ con $k = 1, 2, 3, 4, \dots$

Se si vuole poi ricercare la soluzione x, y tale per cui x assume il *minimo valore positivo*, essa è data dai seguenti valori:

$$x_m = x_{00} - \left\lfloor \frac{x_{00}}{nx} \right\rfloor \cdot nx \quad \text{con } nx = \frac{B}{(A, B)}$$

$$y_m = y_{00} - \left\lfloor \frac{y_{00}}{ny} \right\rfloor \cdot ny \quad \text{con } ny = \frac{A}{(A, B)}$$

con $(A, B) = 1$ se A e B sono dello stesso segno e $(A, B) = -1$ se A e B sono di segno opposto.

5. Applicazione alla Crittografia

Un'importante applicazione in cui si utilizza l'equazione diofantea del tipo sopra illustrato, in particolare del tipo $A \cdot x - B \cdot y = 1$ con A e B entrambi positivi, si ha nell'ambito della crittografia a chiave pubblica e precisamente nell'algoritmo RSA, nell'algoritmo DSA (Digital Signature Algorithm), dedicato quest'ultimo espressamente alla firma elettronica, e nell'algoritmo di El Gamal come pure in altri algoritmi crittografici meno noti.

Considerando i due importanti metodi crittografici RSA e DSA si riportano qui solo le specifiche riguardanti alcuni loro parametri senza prendere in considerazione una loro descrizione per la quale si può rimandare a vari testi o articoli, vedi ad esempio [FeL], [G.U.],[Hel] [Men], [Sch], [Sga] , i siti Internet [Men],[Oli] e in particolare il sito Internet del Liceo Classico M.Foscatini - Venezia [L.F], per semplicità e chiarezza nell'esposizione del principio di funzionamento dell'algoritmo RSA.

Per quanto riguarda questo algoritmo si illustrerà più avanti come si può trovare la **Chiave privata** con l'ausilio delle congruenze lineari.

5.1 Chiavi nell' Algoritmo RSA

Chiave pubblica: è costituita da due numeri denominati convenzionalmente uno con il simbolo n , l'altro con il simbolo e : n è un numero composto da due numeri primi p, q grandi: $n = p \cdot q$;

e è un numero random positivo $< \Phi$ dove $\Phi = (p-1) \cdot (q-1)$ è denominata funzione di Eulero;

e deve essere tale per cui $(e, \Phi) = 1$. Se si pone e primo non occorre naturalmente verificare la condizione che sia $(e, \Phi) = 1$.

Chiave privata: è un numero d legato ad e ed a Φ dalla seguente relazione: $e \cdot d - \Phi \cdot y = 1$ che possiamo anche porre sotto la forma $e \cdot d \equiv 1 \pmod{\Phi}$ od anche $d = e^{-1} \pmod{\Phi}$.

Pertanto d risulta essere l'inverso moltiplicativo di e modulo Φ .

Per trovare il valore di d , una volta noti i valori di Φ e di e , si dovrà risolvere pertanto l'equazione diofantea. $e \cdot x - \Phi \cdot y = 1$. Il valore di x soddisfacente l'equazione sarà il valore della chiave privata d .

5.2 Algoritmo DSA

Per questo più complesso algoritmo, dedicato alla firma digitale, si accenna brevemente solo a quanto segue, facendo riferimento per la denominazione dei simboli e per maggiori dettagli a [Men].

Chiave pubblica: è costituita da quattro parametri denominati p, q, a, y dove:

q è un numero primo random costituito da 48 cifre decimali;

p è un numero primo costituito da non meno di 154 cifre decimali e tale per cui $(p-1)$ sia divisibile per q ;

$a = g^{\frac{p-1}{q}}$ essendo g un intero positivo random $< p$;

$y = a^a \pmod{p}$ dove a è un intero positivo random $\leq q-1$.

Chiave privata: è costituita dal solo parametro a (sopra definito)

Per la generazione di questi parametri pertanto non risulta implicata nessun'operazione di risoluzione di equazioni lineari diofantee

Questo tipo di operazione viene invece impiegata nei riguardi sia della **generazione** della **Firma digitale** (*Signature generation*) sia in quella per la **verifica** della **Firma** (*Signature verification*).

Senza entrare nei particolari, nel computo dei parametri richiesti per la generazione della firma digitale, uno di essi risulta esser l'inverso moltiplicativo modulo q di un numero segreto k intero positivo scelto in modo random e minore di q ; si deve effettuare quindi il calcolo di $k^{-1} \pmod{q}$ e quindi risolvere una equazione lineare diofantea.

Analogamente anche nelle operazioni di verifica per trovare uno dei suoi parametri occorre effettuare il calcolo dell'inverso moltiplicativo di un parametro definito nelle operazioni dedicate alla generazione.

5.3 Calcolo della Chiave privata nell'algoritmo RSA

Vediamo ora più in dettaglio quali sono le operazioni necessarie per il calcolo Chiave Privata nell'algoritmo crittografico RSA.

Per calcolare la chiave privata d , come si è già detto occorre risolvere l'equazione $e \cdot x - \Phi \cdot y = 1$. Tenendo conto di quanto esposto nei paragrafi precedenti una volta conosciute le grandezze Φ ed e ⁽¹⁾ il valore risolutivo $x_0 = \frac{q_{n-1}}{(e, \Phi)}$ sarà la **Chiave privata** d cercata. Pertanto si ha:

$$d = x_0 = \frac{q_{n-1}}{(e, \Phi)}. \text{ Poiché poi deve essere } (e, \Phi) = 1 \text{ si avrà } d = q_{n-1}.$$

Vengono qui date delle informazioni riguardanti le prestazioni di due programmi realizzati in linguaggio Qbasic con i quali si può ottenere tramite i relativi Eseguibili sia la risoluzione di equazioni lineari diofantee sopra illustrate, sia trovare chiavi private riguardanti l'Algoritmo RSA

Il primo programma utilizza l'aritmetica disponibile sul PC col Qbasic che non va oltre la doppia precisione; quindi con esso per non avere risultati approssimati si devono trattare per i tre parametri e, p, q valori positivi tali per cui la somma complessiva delle cifre che li compongono non superi il valore 16

(ad esempio e composto al massimo da 4 cifre con p e q composti da 6 cifre), i quali quindi si possono considerare solo come esempi esplicativi dell'algoritmo.

Essendo in effetti questo programma rivolto alla risoluzione dell'equazione generale $A \cdot x - B \cdot y = C$ anche i valori dei coefficienti A, B, C , che sono numeri interi sia positivi che negativi, non devono avere in linea di massima ciascuno un valore assoluto maggiore di 10^5 , avvertendo che per ottenere soluzioni esatte qualsiasi risultato ottenibile nei calcoli non deve avere valore $> 10^{15}$ in relazione ad una qualunque delle tre opzioni offerte dal programma:

- la prima dedicata alla risoluzione di una generica equazione lineare diofantea;
- la seconda relativa espressamente al calcolo della chiave privata d una volta introdotti da input la grandezza e ed i due numeri primi p e q , da cui si può ricavare il valore di $\Phi = (p-1) \cdot (q-1)$ e il valore di $n = p \cdot q$ che costituisce insieme ad e la **Chiave pubblica** (n, e) nell'RSA.
- la terza dedicata ad un esempio di calcolo di chiave privata d .

Il secondo programma sempre in Qbasic, è dedicato anch'esso alla risoluzione di equazioni lineari diofantee ma a differenza del precedente programma è in grado di elaborare numeri grandi e quindi di calcolare **effettivi** valori di chiavi private poiché le operazioni di calcolo sono programmate per una loro utilizzazione in aritmetica a precisione multipla.

Esso è dedicato sostanzialmente alla generazione della **chiave privata**, quindi alla risoluzione dell'equazione del tipo $e \cdot x - \Phi \cdot y = 1$ e presenta due opzioni:

Prima opzione: dedicata al calcolo della chiave privata d introducendo dall'esterno e quindi da richiesta di INPUT tre numeri primi grandi, costituiti da stringhe di tipo numerico, riguardanti i seguenti tre parametri:

- e numero dispari e preferibilmente primo random⁽¹⁾;
- p numero primo random grande (il numero p per un suo effettivo impiego dovrebbe essere costituito da almeno 160 cifre);
- q numero primo random (di almeno 140 cifre);

i numeri p e q devono essere tali per cui $n = p \cdot q$ risulti costituito da almeno 309 cifre decimali (digit) pari ad un numero di bit non inferiori a 1024 bit (vedi Art.4 dell'ALLEGATO TECNICO di [G.U.]).

Si suppone che i suddetti tre parametri siano stati preliminarmente creati tramite calcoli già effettuati con opportuni programmi [vedi Nota] e tali per cui $MCD(e, \Phi) = 1$ dove $\Phi = (p-1) \cdot (q-1)$

Seconda opzione: relativa ad un esempio con i valori dei tre parametri suddetti già inseriti nel programma come dati costituiti da stringhe numeriche e quindi immediatamente disponibili.

Nota: i numeri primi grandi e, p, q possono ricavarsi in genere tramite l'ausilio di appositi pacchetti software matematici (vedi ad esempio "Mathematica", "Maple", ecc.). I numeri primi casuali riportati negli esempi e nel testo del programma riportato nell' Allegato 2 sono invece stati creati ciascuno con un tempo di calcolo di qualche minuto, utilizzando un apposito programma in Qbasic dedicato alla generazione e alla verifica di numeri primi relativamente grandi, descritto e riportato in [Teo]. Inoltre per il calcolo dei valori di Φ e del MCD si sono utilizzati opportuni programmi in aritmetica a precisione multipla realizzati dall'autore.

(1) La scelta del più opportuno valore di e richiede particolare attenzione in quanto si deve tener conto di due esigenze fra di loro contrastanti: un valore piccolo di e comporta un più veloce processo di cifratura; d'altra parte è opportuno avere grandi sia il valore di e che quello della chiave privata d [Sch] contro possibili attacchi crittoanalitici.

Si fa presente che nella sezione software dei lavori dell'autore sono disponibili due **Eseguibili**, uno relativo al suddetto primo programma, il secondo all'altro programma e cioè al calcolo e alla realizzazione di chiavi private effettivamente valide per l'algoritmo crittografico RSA.

RIFERIMENTI

- [FeLu] P. Ferragina e F. Luccio, CRITTOGRAFIA, Capp. 8 e 9 – 2001 Bollati Boringhieri editore s.r.l., Torino
- [G.U.] DECRETO DEL PRESIDENTE DEL CONSIGLIO DEI MINISTRI 8 febbraio 1999
GAZZETTA UFFICIALE DELLA REPUBBLICA ITALIANA, Anno 140° - Numero 87
del 15 aprile 1999
- [Hel] M. E. Hellman, *The Mathematics of Public - Key Cryptography*, Scientific American, v. 241 n. 8I Aug. 1979
- [L.F.] <http://www.liceofoscarini.it/studenti/crittografia/critto/rsa/metodo.html>
- [Men] A.J. Menezes, P.C. van Oorschot, S.a Vanstone – HANDBOOK of APPLIED CRYPTOGRAPHY
Ch. 11 – <http://www.cacr.math.uwaterloo.ca/hac/>
- [Old] C.D. Olds, FRAZIONI CONTINUE, Zanichelli, Bologna 1970
- [Sch] B. Schneier, APPLIED CRYPTOGRAPHY, Capp. 12 e 13 – 1994 John Wiley & Sons, Inc.
- [Sga] A. Sgarro, CRITTOGRAFIA, Cap. 7 – Prima edizione 1986, Franco Muzzio & c. editore
- [Teo] C. Teodoro, *Numeri Primi Grandi* - vedi sul Sito "Gruppo Eratostene" la sezione Articoli e la sezione Software dei lavori dell'ing. Teodoro

Esempio di risoluzione ottenuto con un programma in Qbasic realizzato utilizzando una aritmetica a precisione multipla

Si riporta ciò che compare sullo schermo del monitor dopo aver fatto partire il programma ed aver inserito i dati richiesti riguardanti l'opzione scelta.

esempio relativo alla prima opzione (dati introdotti da INPUT)

Siano noti i seguenti valori (vedi Nota a pag. 8):

e : 82132240553708956102000023104321022310483
e è un primo di 41 cifre

p : 5641349579891310252345676789492265610000570123746761347376644766676677102001220
10001234204459998675437549111952738222310519597356425008731040026520870011235561
p è un primo di 160

q : 499289014557779494113201024761641002021230405513312050410123411002435674646789
9110205646789821050746124545449000012457877778744100222354444559445107
q è un primo di 149 cifre

Prima di effettuare il calcolo della chiave privata d si è appurato con un programma ad hoc che i valori numerici random dei due primi p e q introdotti fossero tali da avere $\text{MCD} \left(\frac{p-1}{2}, \frac{q-1}{2} \right) = 1$ come suggerito in [FeL].

Eseguendo il programma si ottiene il risultato riportato nella pagina seguente:

===== calcolo della CHIAVE PRIVATA nell'algoritmo crittografico RSA == =====

(e = CHIAVE PUBBLICA; F = FUNZIONE DI EULERO); d = x CHIAVE PRIVATA nell' RSA)

Sono previste due opzioni:

la PRIMA OPZIONE è relativa ad introdurre come INPUT tre numeri primi: e, p, q

la SECONDA OPZIONE è relativa ad un ESEMPIO di calcolo con e, p, q già inseriti

Se si vuole la prima opzione battere prima su un tasto qualsiasi
e poi sul tasto relativo alla cifra 1

Se si vuole la seconda opzione battere prima su un tasto qualsiasi
e poi sul tasto relativo alla cifra 2

quale opzione? 1

-----ATTENZIONE !!!!!-----

I TRE NUMERI DA INTRODURRE DEVONO ESSERE PRIMI

introdurre parametro e: 8213224055370895610200023104321022310483

cifre di e: 41

introdurre il numero primo p:

5641349579891310252345676789492265610000570123746761347376644766676677102001220100012342

044599986754375491119527382222310519597356425008731040026520870011235561

cifre di p: 160

introdurre il numero primo q:

499289014557779494113201024761641002021230405513312050410123411002435674646789911020564

6789821050746124545449000012457877778744100222354444559445107

cifre di q: 149

Funzione di Eulero: $F = (p-1)*(q-1)$

F = 281 666387 251986 692364 595131 416908 192424 623395 017528 020728

150215 365945 359622 819307 504694 532631 904567 078908 565865 745780 278998

566340 512134 990356 657223 493210 954335 661453 700885 218580 896292 744820

360614 097212 751002 901709 640536 316677 679716 210274 542172 046833 116164

940786 400564 192160 332052 126979 067767 631476 539055 169360

=====

m.c.d.(e,F) = 1

d = 215 848789 810065 513208 302874 205921 122935 334449 334942 657295

536075 214506 868195 979917 314395 792274 353168 956274 124986 211114 596913

065463 429612 455215 420383 072827 555777 122991 700187 967008 591109 257601

820645 413373 921927 977000 310981 480265 384701 404949 044705 674957 052534

439113 216316 015300 413232 895718 188959 262368 760601 426187

il numero d è la Chiave Privata risultando l'inverso moltiplicativo

di e (mod F); essa è composta da 309 cifre

----- VERIFICA -----

e * d = 17 728144 727907 263228 158896 814810 532494 574604 463330 330903

063285 582088 216674 268345 270208 265038 171811 259158 340964 120585 924418

260300 904799 813302 915987 538557 242706 374919 643335 856388 107997 136781

175149 003823 783795 481821 721502 816569 379788 589946 842687 386568 190870

219991 189183 372124 089319 460180 765388 701017 017573 914348 339248 868111

294874 792274 455383 215720 818321

F * y = 17 728144 727907 263228 158896 814810 532494 574604 463330 330903

063285 582088 216674 268345 270208 265038 171811 259158 340964 120585 924418

260300 904799 813302 915987 538557 242706 374919 643335 856388 107997 136781

175149 003823 783795 481821 721502 816569 379788 589946 842687 386568 190870

219991 189183 372124 089319 460180 765388 701017 017573 914348 339248 868111

294874 792274 455383 215720 818320

e * d - F * y = 1 tempo impiegato: .05 secondi