Alma Mater Studiorum · Università di Bologna

FACOLTÀ DI SCIENZE
Corso di Laurea in Matematica

ATTACCO DI WIENER AD RSA MEDIANTE FRAZIONI CONTINUE

Tesi di Laurea in Algoritmi della Teoria dei Numeri e Crittografia

Relatore:
Prof. ALIFFI DAVIDE

Presentata da:
DIOMEDI STEFANO

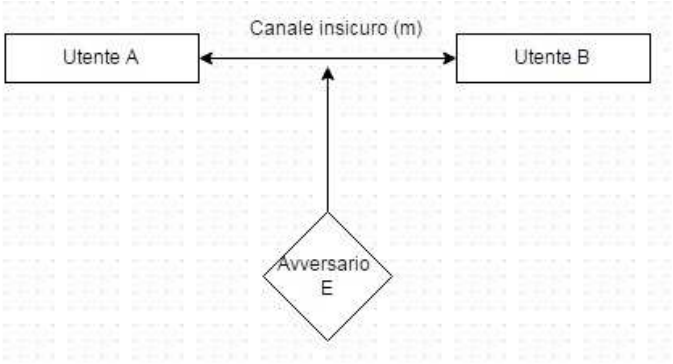
III Sessione
Anno Accademico 2013/2014

Indice

Introduzione

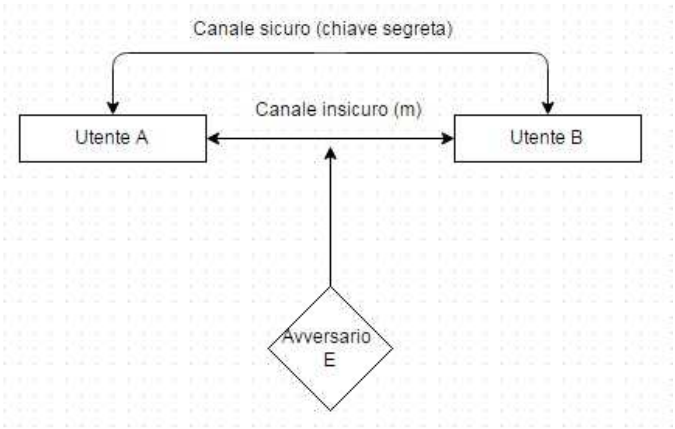
La crittografia è una disciplina che studia la trasformazione dell'informazione con lo scopo di renderla sicura da destinatari o usi non voluti.
L'operazione che trasforma un messaggio in chiaro, cioè leggibile da chiunque, in un messaggio incomprensibile è chiamata **cifratura**, mentre il processo che riconverte il messaggio cifrato in un messaggio comprensibile è detto **decifrazione**. Entrambe queste operazioni, scritte sotto forma di algoritmo, avranno in input una o piu *chiavi segrete*.

Lo schema generale è questo:
due utenti A e B vogliono scambiarsi un messaggio segreto m, su un canale insicuro, senza che questo venga intercettato da un terzo utente che chiameremo E.



Fino a pochi anni fa l'unico metodo crittografico conosciuto era quello della **crittografia simmetrica**, nella quale viene usata solo una chiave sia per cifrare che per decifrare, una per ogni coppia di utenti. Il problema però sta nel condividere la chiave segreta tra gli utenti senza che questa venga intercettata da E, quindi c'è bisogno di un canale sicuro.

Lo schema generale è il seguente:

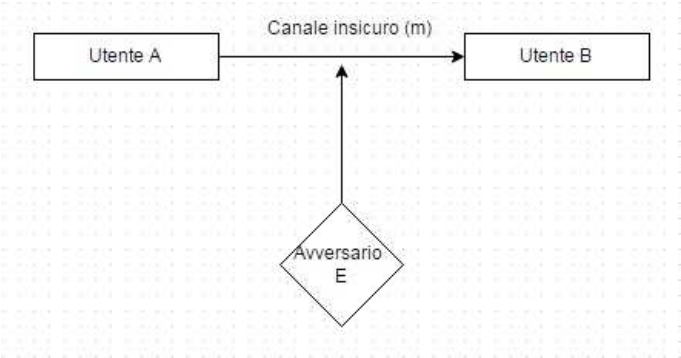


Nel 1976 Diffie e Hellman proposero un nuovo crittosistema, detto **crittosistema asimmetrico** o **a chiave pubblica**, che negli anni , dopo perfezionamenti ad opera di Rivest, Shamir e Adleman, nel 1977 divenne il crittosistema **RSA**, che si basa sulla difficoltà di fattorizzazione degli interi in fattori primi. La differenza fondamentale con i crittosistemi simmetrici è che qui ogni utente ha 2 chiavi differenti, una per cifrare pubblica e una per decifrare che viene tenuta segreta. Questo è un grosso passo avanti in quanto ad esempio viene risolto il problema della *distribuzione delle chiavi*: con la crittografia simmetrica era necessario scambiare l' unica chiave segreta attraverso un canale sicuro, mentre con quella asimmetrica non bisogna più preoccuparsi di scambiare la chiave su un canale segreto in quanto l'unica informazione da scambiare è quella pubblica già accessibile a tutti.

Capitolo 1

RSA

Lo schema generale di RSA è il seguente:
l'utente A vuole mandare un messaggio m a B senza che E lo intercetti.

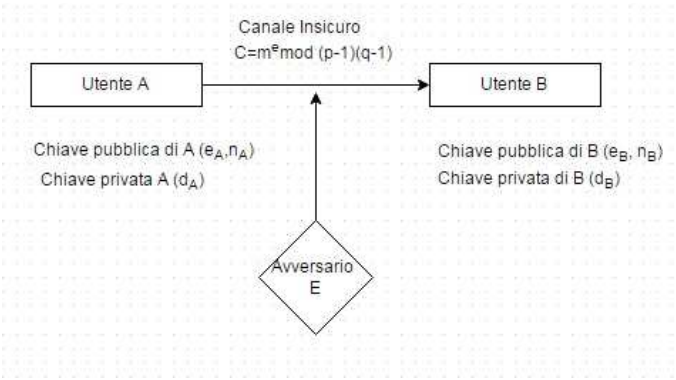


B sceglie due primi p e q grandi e distinti e li moltiplica per formare il numero:

$$n_B = pq.$$

che viene detto *modulo* del nostro sistema RSA, e viene trovato
$\varphi(n_B) = (p-1)(q-1)$ dove φ è la funzione di Eulero. Poi B sceglie il suo esponente di cifratura e_B (esponente pubblico) in modo che questo sia primo con $\varphi(n_B)$, e $e_B < \varphi(n_B)$. Infine viene calcolato l'esponente di decifrazione d_B (esponente privato) tale che il suo prodotto con e_B sia congruo ad 1 modulo $\varphi(n_B)$, ovvero $e_B d_B \equiv 1\,(mod(\varphi(n_B)))$.
La **chiave pubblica** di B sarà il modulo e l'esponente di cifratura ovvero la coppia (e_B, n_B) mentre la sua **chiave privata** sarà l'esponente di decifrazione d_B. Nel caso in cui B voglia mandare un messaggio ad A, quest'ultimo sceglierà a sua volta due primi e seguirà lo stesso procedimento di B generando però le sue due chiavi (e_A, n_A) e d_A.

Quindi lo schema diventa:



Ora vediamo le fasi di Cifratura e Decifrazione nel caso in cui A voglia mandare il messaggio m (che per semplicità consideriamo come un numero)

- **Cifratura** Preso il messaggio in chiaro m, A calcola utilizzando l'esponente pubblico di B:
$$C = m^e \, mod \, n$$
e manda C a B.

- **Decifrazione** Una volta ricevuto il messaggio cifrato C, attravero il suo esponente privato, B calcola:
$$C^d \, mod \, n \, = m$$
e decifra il messaggio criptato.

La decifrazione funziona in quanto:

$$d \cdot e \equiv 1 \, mod \, \varphi(n) \; \Rightarrow \; d \cdot e = 1 + k\varphi(n)$$

da cui:
$$C^d = (m^e)^d = m^{1+k\varphi(n)} = m \cdot m^{k(\varphi(n))} = m \cdot (m^{\varphi(n)})^k$$

con $m^{\varphi(n)} \equiv 1 \, mod \, n$ da cui:

$$C^d = m \, mod \, n.$$

Esempio 1.0.1.
Generazione Chiavi

- *B sceglie due primi p e q:*

$$p = 47, \; q = 71$$

- *B calcola* $n = p \cdot q$ *e* $\varphi(n) = (p-1)(q-1)$*:*

$$n = 47 \times 71 = 3337,\ \varphi(n) = 46 \times 70 = 3220$$

- *B sceglie e tale che* $MCD(e, \varphi(n)) = 1$*:*

$$e = 79$$

- *B calcola* $d = e^{-1} \, mod \, \varphi(n)$*:*

$$d = 79^{-1} \, mod \, 3220 = 1019$$

- *La chiave pubblica è la coppia* (e, n)*:*

$$(e, n) = (79, 3337)$$

- *La chiave privata è* d*:*

$$d = 1019$$

Cifratura

- *Un utente A vuole mandare a B il messaggio* $m = 688$ *(per semplicità consideriamo un numero) usando la chiave pubblica di B* $k_B = (79, 3337)$

- *A calcola* $C = m^e \, mod \, n$*:*

$$C = 688^{79} \, mod \, 3337 = 1570$$

Decifrazione

- *B riceve il messaggio cifrato* $C = 1570$

- *B si ricava* m *con la formula* $m = C^d \, mod \, n$*:*

$$m = 1570^{1019} \, mod \, 3337 = 688$$

1.1 Sicurezza di RSA

La sicurezza di RSA si basa fondamentalmente su un problema intrattabile, ovvero quello della fattorizzazione di n, da cui segue la conoscenza di $\varphi(n)$. Infatti E conosce la chiave pubblica, quindi sia l'esponente pubblico che il modulo, e se fosse in grado di fattorizzare quest'ultimo, troverebbe i due primi p e q con cui ricaverebbe $\varphi(n) = (p-1)(q-1)$ e troverebbe l'esponente segreto risolvendo:

$$d\,e + y\,\varphi(n) = 1$$

Mostriamo ora che l'unico modo per decifrare un messaggio è conoscere l'esponente segreto d:

Proposizione 1.1.1.
Conoscere $\varphi(n)$ è equivalente a conoscere p e q

Dimostrazione. Poichè p e q sono primi allora posso calcolare:

$$\varphi(n) = (p-1)(q-1) \Rightarrow \varphi(n) = pq - (p+q) + 1 = n - (p+q) + 1$$

ne viene che p e q sono le radici del polinomio $x^2 - (p+q)x + pq$. □

Definizione 1.1.
Dato il *modulo* n, si dice che r è un *esponente universale* se:

$$a^r \equiv 1 \, mod \, n \quad \forall a \in \mathbb{Z} : MCD(a, n) = 1$$

Osservazione 1.
Se n è un primo allora $n-1$ è un esponente universale per il teorema di Fermat, inoltre in generale per il teorema di Eulero $\varphi(n)$ è sempre un esponente universale e un esponente universale è sempre pari

Proposizione 1.1.2.
Esiste un algoritmo propbabilistico che, dato un esponente universale del modulo $n = p \cdot q$ permette di trovare p e q.

Proposizione 1.1.3.
La conoscenza di d permette la fattorizzazione di n

Dimostrazione. Basta scrivere $ed = 1 + k\varphi(n)$ da cui $ed - 1$ sarà esponente universale, quindi utilizzando l'algoritmo per gli esponenti universali, fattorizzo n. □

1.2 Attacchi base ad RSA

Ci sono due strategie generali per attaccare l'algoritmo RSA:

- trovare un algoritmo che, dato un messaggio cifrato C e la chiave pubblica (e, n), restituisce il messaggio decifrato m senza però recuperare la chiave privata d, ma nessuno ha mai trovato un algoritmo efficiente.

- ricavare la chiave privata da quella pubblica.

Come abbiamo visto nella sezione riguardante la sicurezza, ci sono 3 alternative equivalenti per ricavare la chiave privata da quella pubblica:

1. fattorizzare n

2. Calcolare $\varphi(n)$

3. ricavare il valore d dalla chiave pubblica

rese però inutili dal problema, intrattabile, della fattorizzazione di n. In alcuni casi particolari però è facile forzare il sistema, a causa di una scelta errata dei parametri. Questo succede ad esempio:

- se m ed e sono così piccoli che $m^e < n$; allora risulta facile trovare la radice e-esima di C poichè $C = m^e$ e poi con un algoritmo di approssimazione numerica si può trovare m

- se $p - q < n^{1/4}$ ci sono algoritmi veloci per trovare p e q.

- se $p - 1$ e $q - 1$ hanno solo fattori piccoli, n si fattorizza velocemente con l'algoritmo di Pollard $p - 1$

- se, come è stato trovato da Wiener, (lo vedremo in maniera piu approfondita nel capitolo dedicato), $q < p < 2q$ e $d < \frac{1}{3}n^{\frac{1}{4}}$.

Capitolo 2

Frazioni Continue

In questo capitolo ci occuperemo di spiegare cosa è una frazione continua e di esporre le sue proprietà fondamentali sfruttate dall'attacco dell' esponente segreto piccolo. Ci soffermeremo più sulla parte delle frazioni continue per i razionali che per gli irrazionali, perchè questi ultimi non sono utilizzati dalla crittografia.

2.1 Algoritmo Euclideo

L'algoritmo di Euclide è un algoritmo ricorsivo risalente al 300 a.C. che permette, dati due numeri interi (a, b) di calcolare il loro massimo comun divisore d ($MCD(a, b) = d$) senza richiedere la fattorizzazione in fattori primi di a e di b che, per numeri troppo grandi, risulta essere praticamente impossibile.

Vediamo come funziona:
siano $a, b \in \mathbb{Z}, a > b : a, b \geq 2$ per semplicità.
Vale che $MCD(a, b) = MCD(b, r_1)$ con r_1 resto della divisione di a per b. Applichiamo la divisione con resto iterativamente fino ad ottenere un resto nullo:

$$\textbf{PASSO 1} : a = b*q_1+r_1 \begin{cases} \text{se } r_1 = 0 \Rightarrow MCD(a, b) = b \\ \text{se } r_1 \neq 0 \\ \quad \text{PASSO 2 (cioè si riapplica il procedimento con } a = b \, e \, b = r_1) \end{cases}$$

$$\textbf{PASSO 2} : b = r_1 * q_2 + r_2 \begin{cases} \text{se } r_2 = 0 \Rightarrow MCD(b, r_1) = r_1 \\ \text{se } r_2 \neq 0 \text{ PASSO 3} \end{cases}$$

[...]

PASSO i-esimo ; $r_{i-2} = r_{i-1} * q_i + r_i$ $\begin{cases} \text{se } r_i = 0 \Rightarrow MCD(r_{i-2}, r_{i-1}) = r_{i-1} \\ \text{se } r_i \neq 0 \text{ PASSO i+1} \end{cases}$

L'ultimo resto non nullo sarà l' $MCD(a, b)$

Esempio 2.1.1.

Calcoliamo l'MCD dei numeri 345786 e 7432:

$$345786 = 7432 * 46 + 3914$$
$$7432 = 3914 * 1 + 3518$$
$$3914 = 3518 * 1 + 396$$
$$3518 = 396 * 8 + 350$$
$$396 = 350 * 1 + 46$$
$$350 = 46 * 7 + 28$$
$$46 = 28 * 1 + 18$$
$$28 = 18 * 1 + 10$$
$$18 = 10 * 1 + 8$$
$$10 = 8 * 1 + 2$$
$$8 = 2 * 4 + 0$$

$\Rightarrow$ *MCD(345786,7432)=2*

2.2 Frazioni continue semplici finite e numeri razionali

Un'applicazione molto significativa dell' algoritmo euclideo è quella delle frazioni continue, che sono un modo alternativo di rappresentare i numeri reali. Infatti, se vogliamo ad esempio trovare la frazione continua del numero $\frac{68}{15}$, applichiamo l'algoritmo con a=68 e b=15 ottenendo:

$$68 = 15 * 4 + 8 \tag{2.1}$$
$$15 = 8 * 1 + 7 \tag{2.2}$$
$$8 = 7 * 1 + 1 \tag{2.3}$$
$$7 = 1 * 7 + 0 \tag{2.4}$$

ora dividendo primo e secondo membro di (2.1) per 15. Otteniamo

$$\frac{68}{15} = 4 + \frac{8}{15} \tag{2.5}$$

Se notiamo che il numero razionale è compreso tra 4 e 5 in quanto $\frac{8}{15} < 1$ e in piu scriviamo $\frac{8}{15}$ come inverso di un numero maggiore di 1, la formula (2.5) diventa

$$\frac{68}{15} = 4 + \frac{1}{\frac{15}{8}} \tag{2.6}$$

Notiamo ora che dividendo la (2.2) per 8 otteniamo la frazione al denominatore di 1 che abbiamo nella (2.6)

$$\frac{15}{8} = 1 + \frac{7}{8} \tag{2.7}$$

e la riscriviamo nella forma

$$\frac{15}{8} = 1 + \frac{1}{\frac{8}{7}} \tag{2.8}$$

Infine, applicando un analogo ragionamento alla (2.3) troviamo

$$\frac{8}{7} = 1 + \frac{1}{7} \tag{2.9}$$

Abbiamo così ottenuto la frazione continua di $\frac{68}{15}$:

$$\frac{68}{15} = 4 + \frac{1}{1 + \frac{1}{1+\frac{1}{7}}} \tag{2.10}$$

Abbiamo visto un esempio di frazione continua. Ora formalizziamo il tutto e iniziamo dando la seguente definizione:

Definizione 2.1 (Frazione Continua)**.**
Si dice *frazione continua finita* una frazione della forma:

$$a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{a_4 + \ddots \cfrac{1}{a_{n-1} + \frac{1}{a_n}}}}} \tag{2.11}$$

con $a_1, a_2, ..., a_n$ numeri reali, tutti positivi, ad eccezione al più di a_1.
I numeri $a_2, ..., a_n$ si chiamano denominatori parziali o quozienti parziali della frazione. Una frazione continua finita si dice semplice se tutti i suoi quozienti parziali sono interi.

Proposizione 2.2.1.
Una qualunque frazione continua semplice finita è uguale ad un numero razionale. Viceversa, un qualunque numero razionale si può scrivere come frazione continua semplice finita.

Dimostrazione. La prima parte è ovvia. Sia ora $\frac{a}{b}$ il numero razionale, $b > 0$.Applichiamo l'algoritmo di euclide per trovare il $MCD(a, b)$

$$\begin{aligned}
&a = b * a_1 + r_1 && 0 < r_1 < b\\
&b = r_1 * a_2 + r_2 && 0 < r_2 < r_1\\
&r_1 = r_2 * a_3 + r_3 && 0 < r_3 < r_2\\
&\vdots && \vdots\\
&r_i = r_{i+1} * a_{i+2} + r_{i+2} && 0 < r_{i+2} < r_{i+1}\\
&\vdots && \vdots\\
&r_{n-3} = r_{n-2} * a_{n-1} + r_{n-1} && 0 < r_{n-1} < r_{n-2}\\
&r_{n-2} = r_{n-1} * a_n + r_n && 0 < r_n < r_{n-1}\\
&r_{n-1} = r_n * a_{n+1} + 0
\end{aligned}$$

Dato che i resti sono tutti positivi, anche i quozienti a_i, ad eccezione eventualmente del primo, saranno positivi. Riscriviamo le equazioni dell'algoritmo euclideo dividendo la prima per b, la seconda per r_1, la terza per r_2,e così via fino all' ultima per r_n. Avremo

$$\begin{aligned}
\frac{a}{b} = a_1 + \frac{r_1}{b} &= a_1 + \frac{1}{\frac{b}{r_1}}\\
\frac{b}{r_1} = a_2 + \frac{r_2}{r_1} &= a_2 + \frac{1}{\frac{r_1}{r_2}}\\
\frac{r_1}{r_2} = a_3 + \frac{r_3}{r_2} &= a_3 + \frac{1}{\frac{r_2}{r_3}}\\
\frac{r_2}{r_3} = a_4 + \frac{r_4}{r_3} &= a_4 + \frac{1}{\frac{r_3}{r_4}}\\
&\vdots\\
\frac{r_{n-1}}{r_n} &= a_{n+1}
\end{aligned}$$

i primi membri delle precedenti uguaglianze sono dei numeri razionali, che abbiamo scritto come somma di un intero e di una frazione con numeratore 1.

Con eliminazioni successive si ottiene

$$\frac{a}{b} = a_1 + \frac{1}{\frac{b}{r_1}} = a_1 + \frac{1}{a_2 + \frac{\frac{1}{r_1}}{r_2}}$$

da cui si arriva all' espressione

$$\frac{a}{b} = a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{a_4 + \ddots \cfrac{1}{a_{n-1} + \frac{1}{a_n}}}}} \tag{2.12}$$

Abbiamo così rappresentato il numero razionale $\frac{a}{b}$ come frazione continua finita semplice. □

Osservazione 2.
La scrittura (2.12) non è molto comoda dal punto di vista della notazione, quindi si preferisce indicare la stessa frazione nel modo seguente:

$$[a_1; a_2, a_3, \ldots, a_{n-1}, a_n]$$

ossia come successione finita dei suoi quozienti parziali. Ad esempio la frazione dell' esempio precedente si scrive

$$\frac{68}{15} = [4; 1, 1, 7]$$

Si osservi che l'intero a_1 sarà zero se e solo se la frazione è positiva e minore di 1. Inoltre osserviamo che a_1 è il valore intero approssimante per difetto a/b, cioè $a_1 = [a/b]$, a_2 il valore approssimante per difetto di b/r_1, cioè $a_2 = [b/r_1]$, in generale

$$a_i = \left[\frac{r_{i-2}}{r_{i-1}}\right]$$

2.3 Frazioni continue semplici e numeri irrazionali

Abbiamo visto che ogni numero razionale ha la sua rispettiva frazione continua; questo vale anche per gli irrazionali. Questi avranno la particolarità di avere frazioni continue infinite, e poichè non possiamo usare con questi l'algoritmo di Euclide per ricavare la frazione, si utilizzerà il seguente metodo. Facciamo prima un esempio: cerchiamo la frazione continua di $\sqrt{2}$. Dato che esso è compreso tra 1 e 2 (uno è la sua parte intera) potrò scrivere:

$$\sqrt{2} = 1 + \frac{1}{x}$$

per qualche numero reale $x > 1$. Precisamente si ricava $x = \sqrt{2} + 1$ da cui:

$$x = \sqrt{2} + 1 = \left(1 + \frac{1}{x}\right) + 1 = 2 + \frac{1}{x}$$

da questa si deducono le seguenti altre:

$$\frac{1}{x} = \frac{1}{2 + \frac{1}{x}} = \frac{1}{2 + \frac{1}{2 + \frac{1}{x}}} = \cdots = \frac{1}{2 + \cfrac{1}{2 + \cfrac{1}{2 + \ddots \cfrac{1}{2 + \cfrac{1}{2 + \frac{1}{x}}}}}}$$

al tendere di n a infinito si ottiene:

$$\sqrt{2} = 1 + \frac{1}{2 + \cfrac{1}{2 + \cfrac{1}{2 + \cfrac{1}{2 + \ldots}}}}$$

Questo intuitivamente è il modo di scrivere $\sqrt{2}$ come Frazione Continua.
In generale, sia f un irrazionale, si ha che i convergenti q_i della frazione continua di f di ottengono cosi:

$$q_0 = \lfloor f \rfloor, \qquad r_0 = f - q_0, \qquad e \tag{2.13}$$

$$q_i = \left\lfloor \frac{1}{r_{i-1}} \right\rfloor, \qquad r_i = \frac{1}{r_{i-1}} - q_i, \qquad per\ i = 1, 2, \ldots, m \tag{2.14}$$

2.4 L'approssimazione di razionali mediante i convergenti

Definiamo ora i convergenti di una frazione continua:

Definizione 2.2 (Convergente di una frazione continua)**.**
Sia $[a_1; a_2, a_3, \ldots, a_n]$una frazione continua semplice finita. La frazione continua che si ottiene troncando la frazione continua al k-esimo quoziente parziale si chiama k-esimo convergente e si denota nel modo seguente:

$$C_k = [a_1; a_2, a_3, \ldots, a_k], \quad per\ ogni\ 1 \leq k \leq n$$

Si noti che C_{k+1} si ottiene da C_k sostituendo a_k con $a_k + \frac{1}{a_{k+1}}$

Osservazione 3.
Ogni $C_k = [a_1; a_2, a_3, \ldots, a_k]$ è un numero razionale quindi usiamo la notazione $C_k = \frac{p_k}{q_k}$, dove $MCD(p_k, q_k) = 1$. Possiamo ora trovare delle formule

per il calcolo del numeratore p_k e del denominatore q_k a partire dai quozienti parziali. Chiaramente se $C_1 = [a_1] = \frac{p_1}{q_1}$, allora $p_1 = a_1$ e $q_1 = 1$. Inoltre, se:

$$C_2 = [a_1; a_2] = a_1 + \frac{1}{a_2} = \frac{a_1 a_2 + 1}{a_2} = \frac{p_2}{q_2}$$

allora $p_2 = a_1 a_2 + 1$ e $q_2 = a_2$. Allo stesso modo , se:

$$C_3 = [a_1; a_2, a_3] = a_1 + \frac{1}{a_2 + \frac{1}{a_3}} = \frac{a_3(a_2 a_1 + 1) + a_1}{a_3 a_2 + 1},$$

allora $p_3 = a_3(a_2 a_1 + 1) + a_1 = a_3 p_2 + a_1 e\, q_3 = a_3 a_2 + 1 = a_3 q_2 + q_1$. Da ciò si può ottenere una formula ricorsiva per il calcolo del numeratore e del denominatore di un convergente:

$$\boxed{p_k = a_k p_{k-1} + p_{k-2}, \qquad q_k = a_k q_{k-1} + q_{k-2}} \tag{2.15}$$

Da queste si ottiene, con semplici calcoli,

$$p_k q_{k-1} - q_k p_{k-1} = -(p_{k-1} q_{k-2} - q_{k-1} p_{k-2})$$

Dato che

$$p_2 q_1 - q_2 p_1 = (a_1 a_2 + 1) \cdot 1 - a_2 a_1 = 1,$$

ne segue:

$$\boxed{p_k q_{k-1} - q_k p_{k-1} = (-1)^k} \tag{2.16}$$

Da qui segue in particolare che, per ogni $k = 1, \ldots, n$, i numeri p_k e q_k sono primi tra loro. Dividendo l'ultima relazione per $q_k q_{k-1}$ otteniamo

$$\frac{p_k}{q_k} - \frac{p_{k-1}}{q_{k-1}} = \frac{(-1)^k}{q_k q_{k-1}},$$

ovvero:

$$\boxed{C_k - C_{k-1} = \frac{(-1)^k}{q_k q_{k-1}}} \tag{2.17}$$

valida per ogni $k \geq 1$. In modo del tutto analogo si verifica la relazione

$$\boxed{C_k - C_{k-2} = \frac{(-1)^k a_k}{q_k q_{k-2}}.} \tag{2.18}$$

Esempio 2.4.1.
Calcolare tutti i convergenti di 68/15. *Dall' esempio possiamo scrivere direttamente lo sviluppo della frazione come* $\frac{68}{15} = [4; 1, 1, 7]$. *Utilizzando le formule ricorsive* (2.15) *si ottengono i vari convergenti:*

$$C_1 = \frac{4}{1}$$
$$C_2 = \frac{a_1 a_2 + 1}{a_2} = \frac{5}{1}$$
$$C_3 = \frac{a_3(a_2 a_1 + 1) + a_1}{a_3 a_2 + 1} = \frac{9}{2}$$

Una approssimazione abbastanza buona di 68/15 *sarà* 9/2.

Osserviamo che i convergenti C_k di una frazione semplice continua finita hanno carattere oscillante. Infatti vale il seguente lemma

Lemma 2.4.2.
Sia $a/b = [a_1; a_2, a_3, \ldots, a_n]$ *una frazione continua semplice. Allora i convergenti verificano le seguenti proprietà:*

- $C_1 < C_3 < C_5 < \ldots$,
- $C_2 > C_4 > C_6 > \ldots$,
- $C_{2j} > C_{2j-1}$, *per ogni* $j \geq 1$.

Di qui si deduce che:

$$C_1 < C_3 < C_5 < \cdots \leq \frac{a}{b} \leq \cdots < C_6 < C_4 < C_2, \tag{2.19}$$

ossia i convergenti approssimano la frazione continua ma in modo oscillante, cioè quelli con indice dispari l'approssimano per difetto, e formano una successione crescente, quelli con indice pari per eccesso e formano una successione decrescente, ed ogni convergente di indice pari è maggiore del convergente precedente.

Dimostrazione. Dalla formula (2.16) dividendo per $q_{k-1}q_k$ otteniamo

$$\frac{p_k}{q_k} - \frac{p_{k-1}}{q_{k-1}} = \frac{(-1)^{k-1}}{q_{k-1}q_k}$$

Ora
se $k = 1$ si ha $\frac{p_1}{q_1} = \frac{p_o}{q_o} + \frac{1}{q_0 q_1} > \frac{p_o}{q_o} \Rightarrow \frac{p_1}{q_1} > \frac{p_o}{q_o}$
se $k = 2$ si ha $\frac{p_2}{q_2} = \frac{p_1}{q_1} - \frac{1}{q_1 q_2} < \frac{p_1}{q_1}$ con $\frac{1}{q_1 q_2} < \frac{1}{q_0 q_1} \Rightarrow \frac{p_2}{q_2} > \frac{p_o}{q_o}$ *e* $\frac{p_2}{q_2} > \frac{p_1}{q_1}$
Interando questo ragionamento si arriva a conclusione. □

Vediamo ora il principale risultato che permette l'attacco di Wiener al sistema RSA.

Teorema 2.4.3 (Teorema di Legendre)**.**

Sia α un razionale e sia $\alpha = [a_1; a_2, \ldots, a_n]$ il suo sviluppo in frazione continua semplice e ne siano $C_n = p_n/q_n$ i convergenti. Se p,q sono interi con $q > 0$, e se n è un intero positivo tale che

$$|q\alpha - p| < |q_n\alpha - p_n|, \tag{2.20}$$

allora $q > q_{n+1}$. Inoltre, se

$$\left|\alpha - \frac{p}{q}\right| < |\alpha - C_n| \tag{2.21}$$

allora $q > q_n$. In altri termini, ogni convergente $C_n = p_n/q_n$ approssima il valore α meglio di qualunque frazione il cui denominatore non superi q_n. Notiamo che tale risultato è valido anche per α irrazionale.

Dimostrazione. Supponiamo valga la (2.20) e supponiamo per assurdo che sia $q < q_{n+1}$. Consideriamo il sistema

$$\begin{cases} p_n x + p_{n+1} y = p \\ q_n x + q_{n+1} y = q \end{cases}$$

Usando la (2.16), si trova

$$y = (-1)^n(pq_n - qp_n), \quad x = (-1)^n(qp_{n+1} - pq_{n+1}).$$

Notiamo che $x \neq 0$, altrimenti si avrebbe $q = q_{n+1}y \geq q_{n+1}$, contro l'ipotesi. Allo stesso modo, $y \neq 0$ altrimenti avremmo $p = p_n x$, $q = q_n x$ e allora

$$|q\alpha - p| = |x||q_n\alpha - p_n| \geq |q_n\alpha - p_n|$$

contro la (2.20).
Verifichiamo che x e y hanno segni opposti. Sia $y < 0$. Allora $q_n x = q - q_{n+1}\, y > 0$ e pertanto $x > 0$ perchè $q_n > 0$. Sia $y > 0$. Poichè $q_n + 1y \geq q_{n+1} > q$ si ha $q_n x = q - q_{n+1}y < 0$, sicchè $x < 0$. Dall'andamento oscillante dei convergenti segue subito che $q_n\alpha - p_n$ e $q_{n+1}\alpha - pn + 1$ hanno segni opposti. Pertanto $x(q_n\alpha - p_n)$ e $y(q_{n+1}\alpha - pn + 1)$ hanno lo stesso segno. Poichè

$$|q\alpha - p| = |(q_n x + q_{n+1}y)\alpha - (p_n x + p_{n+1}y)|$$

si ha

$$|q\alpha - p| = |x||q_n\alpha - p_n| + |y||q_{n+1}\alpha - p_{n+1}| \geq |q_n\alpha - p_n|$$

contro la (2.20), Ció prova la prima parte del teorema.
Supponiamo ora valga la (2.21) e supponiamo per assurdo che sia $q < q_n$. Allora si ha:

$$q\left|\alpha - \frac{p}{q}\right| < q_n|\alpha - C_n|$$

cioè la (2.20). Ma allora deve essere $q \geq q_{n+1}$ e dunque avremo $q_n \geq q_{n+1}$ che contraddice la (2.15) □

Teorema 2.4.4.
Sia α un razionale e sia $\alpha = [a_1; a_2, a_3, \ldots, a_n]$ il suo sviluppo in frazione continua, e ne siano $C_n = p_n/q_n$ i convergenti. Se p,q sono interi primi tra loro, con $q > 0$ e tali che

$$\left|\alpha - \frac{p}{q}\right| < \frac{1}{2q^2},$$

allora p/q è un convergente della frazione continua $[a_1; a_2, a_3, \ldots, a_n]$.

Dimostrazione. Supponiamo che p/q non sia un convergente. Possiamo trovare due convergenti successivi C_n, C_{n+1} tali che $q_n \leq q \leq q_{n+1}$. Per il Teorema (2.4.3) si ha

$$|q_n\alpha - p_n| \leq |q\alpha - p| = q\left|\alpha - \frac{p}{q}\right| < \frac{1}{2q},$$

da cui

$$|\alpha - C_n| < \frac{1}{2qq_n}$$

Poichè $p/q \neq C_n$, si ha $|qp_n - pq_n| \geq 1$ e dunque

$$\frac{1}{qq_n} \leq \frac{|qp_n - pq_n|}{qq_n} = \left|C_n - \frac{p}{q}\right| \leq |\alpha - C_n| + \left|\alpha - \frac{p}{q}\right| < \frac{1}{2qq_n} + \frac{1}{2q^2},$$

il che dà

$$\frac{1}{2qq_n} < \frac{1}{2q^2},$$

cioè $q_n > q$, contro l'ipotesi. □

Capitolo 3

Attacco di Wiener

3.1 Scopo dell'attacco

Ci troviamo nella situazione descritta nel primo capitolo in cui A cerca di mandare un messaggio segreto m a B, utilizzando il sistema crittografico a chiave pubblica RSA, ed in cui E cerca di intercettare e decifrare il messaggio. Quindi avremo che B sceglie due primi p e q grandi e distinti e li moltiplica tra di loro ottenendo

$$n = pq$$

detto *Modulo*. Viene poi scelto l'*esponente di cifratura* e scelto in modo che:

$$MCD(e, \varphi(n)) = 1$$

dove $\varphi(n)$ è la funzione di Eulero che ci da l'ordine del gruppo moltiplicativo degli elementi invertibili dell'anello $\mathbb{Z}_n$ e se si conosce la fattorizzazione di n, ovvero in questo caso p e q, si calcola facilmente con la formula $\varphi(n) = (p-1)(q-1)$. Viene poi calcolato d l'*esponente di decifrazione* d in modo che

$$de \equiv 1 \, mod \, \varphi(n) \quad \Rightarrow \quad d \equiv e^{-1} mod \varphi(n)$$

(si può calcolare con l'algoritmo esteso di Euclide); quindi per B si avranno le seguenti chiavi:

- **CHIAVE PUBBLICA**: (n, e)
- **CHIAVE PRIVATA**: d

Ora, se A vuole mandare un messaggio m a B, calcola:

$$C = m^e \, mod \, n$$

e manda C a B, che potrà decifrare il messaggio grazie alla sua chiave segreta d calcolando:

$$m \equiv C^d \, mod \, n \equiv m^{de} \, mod \, n.$$

Lo scopo dell'avversario E sarà quello di ottenere la chiave segreta d di B a partire da ciò che conosce, cioè la chiave pubblica (n, e).

3.2 Quando si può realizzare

L'attacco di Wiener con esponente di cifratura basso, come ogni altro attacco crittografico, si basa su una debolezza del sistema, in questo caso sarà la lunghezza dell'esponente di decifrazione o cifratura. Quindi tale attacco sarà realizzabile se l'esponente sarà scelto abbastanza piccolo. Infatti in alcuni casi è preferibile usare un piccolo esponente di decifrazione in quanto viene così ridotto il tempo di decriptazione. Questo si ha perchè per un modulo di lunghezza fissata, il tempo di decifrazione o cifratura è proporzionale al numero di bit dell'esponente scelto.
Uno di questi casi, in cui scegliere un esponente corto porta un vantaggio, si verifica quando c'è una grosssa differenza di potenza di calcolo tra due apparecchi, ad esempio quando RSA viene usato nella comunicazione tra una carta di credito e un computer. In questo caso, sarebbe preferibile per il calcolatore della carta avere un esponente in grado di ridurre i processi computazionali richiesti. Un altro caso si ha quando c'è una esigenza di maggiore velocità di decriptazione da parte dell'utente B, che lo porterebbe a scegliere un esponente piu piccolo.

3.3 Come funziona

In breve questo attacco utilizza principalmente tre cose: il teorema di Wiener, il teorema di Legendre e l'algoritmo delle frazioni continue.
Studieremo l'attacco relativo all'esponente di decifrazione d.

3.3.1 Teorema di Wiener e Convergenti

L'Attacco di Wiener si basa sul seguente teorema che ci da una condizione fondamentale sulla lunghezza dell'esponente di decifrazione in relazione agli altri valori n, p, q, e ci dice se è possibile o no usare tale attacco:

Teorema 3.3.1 (Teorema di Wiener)**.**
Siano $n = p \cdot q$ il modulo di un crittosistema RSA, dove p e q sono numeri

primi tali che $q < p < 2q$. *Siano* $1 \leq d$, $e < \varphi(n)$ *tali che* $ed \equiv 1\,(mod\,\varphi(n))$. *Allora se*

$$d < \frac{1}{3}n^{\frac{1}{4}}$$

allora è possibile calcolare d in maniera efficiente, cioè è possibile rompere il sistema RSA.

Dimostrazione.
Poichè $ed \equiv 1\,(mod\,\varphi(n))$, ne segue che $\exists$ un intero t tale che:

$$ed - t\varphi(n) = 1$$

Poichè $n = pq > q^2$, si ha che $q < \sqrt{n}$, quindi:

$$0 < n - \varphi(n) = p + q - 1 < 2q + q - 1 < 3q < 3\sqrt{n}$$

ora vediamo che:

$$\left|\frac{e}{n} - \frac{t}{d}\right| = \left|\frac{ed - tn}{dn}\right| = \left|\frac{1 + t(\varphi(n) - n)}{dn}\right| < \frac{3t\sqrt{n}}{dn} = \frac{3t}{d\sqrt{n}}$$

Poichè $t < d$, si ha che $3t < 3d < n^{\frac{1}{4}}$, dunque:

$$\left|\frac{e}{n} - \frac{t}{d}\right| < \frac{1}{dn^{\frac{1}{4}}}$$

Infine se $3d < n^{\frac{1}{4}}$ si ha:

$$\left|\frac{e}{n} - \frac{t}{d}\right| < \frac{1}{3d^2} < \frac{1}{2d^2}$$

Quindi per il teorema di Legendre abbiamo che $\frac{e}{n}$ è un convergente della frazione continua di $\frac{t}{d}$ e quindi, con l'algoritmo delle frazioni continue che descriveremo fra poco, saremo in grado di ricavarci d. □

3.3.2 Algoritmo per il calcolo di Frazioni Continue

Abbiamo visto nel Secondo Capitolo due modi per trovare la frazione continua di un numero, quello per i razionali con l'algoritmo di Euclide e quello per gli irrazionali. Useremo qui le notazioni del secondo metodo che può comunque essere utilizzato per i razionali:

la frazione continua di un numero razionale positivo f è ottenuta sottraendogli la sua parte intera, invertendo il resto che si ottiene e sottraendo ancora la parte intera, finchè il resto non è zero, cioè:

$$q_0 = \lfloor f \rfloor, \qquad r_0 = f - q_0, \qquad e \tag{3.1}$$

$$q_i = \left\lfloor \frac{1}{r_{i-1}} \right\rfloor, \qquad r_i = \frac{1}{r_{i-1}} - q_i, \qquad per\ i = 1, 2, \ldots, m \tag{3.2}$$

e avremo che $f = [q_0; q_1, q_2, \ldots, q_n]$

Esempio 3.3.2.
$f = \frac{68}{5} = [4, 1, 1, 7]$ *infatti:*

$$q_0 = \left\lfloor \frac{68}{15} \right\rfloor = 4 \qquad r_0 = \frac{68}{15} - 4 = \frac{8}{15}$$

$$q_1 = \left\lfloor \frac{1}{r_0} \right\rfloor = \left\lfloor \frac{15}{8} \right\rfloor = 1 \qquad r_1 = \frac{15}{8} - 1 = \frac{7}{8}$$

$$q_2 = \left\lfloor \frac{1}{r_1} \right\rfloor = \left\lfloor \frac{8}{7} \right\rfloor = 1 \qquad r_1 = \frac{8}{7} - 1 = \frac{1}{7}$$

$$q_3 = \left\lfloor \frac{1}{r_2} \right\rfloor = \lfloor 7 \rfloor = 7 \qquad r_3 = 0$$

Osservazione 4 (Proprietà).
Riscriviamo alcune proprietà già viste per i quozienti e i convergenti, ma con una nuova notazione:

- Si ha che $\forall m \quad q_m \geq 2$
- Si ha che i convergenti approssimano la frazione continua in modo oscillante, quindi per ogni x vale che:

$$\begin{array}{lcll} [q_0; q_1, \ldots, q_m] & < & [q_0; q_1, \ldots, q_{m-1}, q_m + x] & \text{se m è pari} \\ [q_0; q_1, \ldots, q_m] & > & [q_0; q_1, \ldots, q_{m-1}, q_m + x] & \text{se m è dispari} \end{array}$$

Osservazione 5 (Metodo per calcolare f a partire dai quozienti).
Mostriamo ora come ricostruire f a partire dalla sua frazione continua. Se n_i e d_i, per $i = 0, 1, \ldots, m$, sono i numeratori e i denominatori dei convergenti i-esimi ne seguirà:

$$\frac{n_i}{d_i} = [q_0; q_1, \ldots, q_i], \qquad MCD(n_i, d_i) = 1 \qquad \forall i = 0, 1, \ldots, m.$$

e si avrà che:

$$n_0 = q_0, \qquad d_0 = 1, \tag{3.3}$$
$$n_1 = q_0 q_1 + 1, \qquad d_1 = q_1, \tag{3.4}$$
$$n_i = q_i n_{i-1} + n_{i-2}, \qquad d_i = q_i d_{i-1} + d_{i-2} \qquad per\, i = 1, 2, \ldots, m \tag{3.5}$$

in questo modo la frazione $f = \frac{n_m}{d_m}$ può essere ricostruita.
Osservazione 6.
Ricordiamo anche la relazione

$$n_i d_{i-1} - n_{i-1} d_i = -(-1)^i \qquad \text{per } i = 1, 2, \ldots, m \tag{3.6}$$

Andiamo ora a descrivere l'algoritmo:
sia f' un convergente e una buona approssimazione di f, si ha:

$$f' = f(1 - \delta) \quad per\, qualche\, \delta \geq 0 \tag{3.7}$$

Siano q_i, r_i e q'_i, r'_i i quozienti parziali (che per semplicità chiameremo quozienti) e i resti rispettvamente di f e f'. Se δ è abbastanza piccolo, allora posso trovare il numeratore e il denominatore di f usando questo algoritmo iterativo:
si ripetono i seguenti punti fino a trovare f:

1. Vengono generati i quozienti (q'_i) dell' espansione in frazione continua di f'

2. Usiamo la Costruzione a partire dalla frazione continua vista prima per costruire la frazione

$$[q'_0, q'_1, \ldots, q'_{i-1}, q'_i + 1] \qquad \text{se i è pari} \tag{3.8}$$
$$[q'_0, q'_1, \ldots, q'_{i-1}, q'_i] \qquad \text{se i è dispari} \tag{3.9}$$

3. Si controlla se la frazione costruita è proprio f.

Osservazione 7.
La ragione per cui si aggiunge 1, se i quozienti sono in numero pari, è che la stima di f che andiamo a cercare deve essere maggiore di f', poichè $f \geq f'$; infatti per l'oscillazione dei convergenti si ha:

$$[q'_0; q'_1, \ldots, q'_{i-1}, q'_i] < f' = [q'_0; q'_1, \ldots, q'_{i-1}, q'_i + r'_i] \quad se\, m\, pari$$

Proposizione 3.3.3.
L'algoritmo delle frazioni continue ha successo se:

$$[q_0; q_1, \ldots, q_{m-1}, q_m - 1] \quad < \quad f \quad \leq \quad [q_0; q_1, \ldots, q_{m-1}, q_m] \qquad \textit{se } m \textit{ è pari} \tag{3.10}$$

$$[q_0; q_1, \ldots, q_{m-1}, q_m + 1] \quad < \quad f \quad \leq \quad [q_0; q_1, \ldots, q_{m-1}, q_m] \qquad \textit{se } m \textit{ è dispari} \tag{3.11}$$

da cui equivalentemente basta che

$$\delta < \frac{1}{\frac{3}{2} n_m d_m} \tag{3.12}$$

Dimostrazione.
Prima di tutto scriviamo δ come:

$$\delta = 1 - \frac{f'}{f}. \tag{3.13}$$

Vediamo i casi $m = 0, 1$; m pari ≥ 2 e m dispari ≥ 3

Caso 1 **m=0**
Usando (3.10)(3.11) per sostituire f' in (3.13) otteniamo:

$$\delta < 1 - \frac{[q_0 - 1]}{[q_0]} = \frac{1}{q_0} = \frac{1}{n_0 d_0}$$

Caso 2 **m=1**
Come prima otteniamo:

$$\delta < 1 - \frac{[q_0; q_1 + 1]}{[q_0; q_1]} = \frac{1}{(q_0 q_1 + 1)(q_1 + 1)}$$

Sappiamo che $q_m \geq 2$ quindi in questo caso avremo che $\frac{3}{2} q_1 \geq q_1 + 1$ da cui:

$$\delta < \frac{1}{\frac{3}{2} n_1 d_1}$$

Caso 3 **m pari, m $\geq$ 2**
Analogalmente agli altri casi otteniamo in questo caso:

$$\delta < 1 - \frac{[q_0; q_1, \ldots, q_{m-1}, q_m - 1]}{[q_0; q_1, \ldots, q_{m-1}, q_m]}$$

Usando (3.3) otteniamo due equivalenze:

$$[q_0; q_1, \ldots, q_{m-1}, q_m - 1] = \frac{(q_m - 1)n_{m-1} + n_{m-2}}{(q_m - 1)d_{m-1} + d_{m-2}}$$

$$[q_0; q_1, \ldots, q_{m-1}, q_m] = \frac{q_m n_{n-1} + n_{m-2}}{q_m d_{m-1} + d_{m-2}}$$

che sostituite nella disuguaglianza danno:

$$\delta < \frac{n_{m-1}d_{m-2} - n_{m-2}d_{m-1}}{(q_m n_{m-1} + n_{m-2})(q_m d_{m-1} + d_{m-2} - d_{m-1})}$$

da cui usando le espressioni per n_m e d_m si ottiene:

$$\delta < \frac{1}{n_m(d_m - d_{m-1})} = \frac{1}{n_m d_m} - \frac{1}{n_m d_{m-1}} < \frac{1}{n_m d_m}$$

Caso 4 **m dispari, m $\geq$ 3**
Come fatto per gli ultimi tre casi, otteniamo

$$\delta < \frac{1}{n_m(d_m + d_{m-1})}$$

Poichè $d_m = q_m d_{m-1} + d_{m-2}$ e $q_m \geq 2$, abbiamo che $d_m + d_{m-1} \leq (3/2)d_m$ si ha

$$\delta < \frac{1}{\frac{3}{2}n_m d_m}$$

□

Osservazione 8 (Considerazioni sul tempo di esecuzione).
Se $x = max(n_m, d_m)$, si può dimostrare che il numero di quozienti dell' espansione in frazione continua di f e $O(log(x))$. Per ogni quoziente, una approssimazione di f è generata e testata. Il tempo di calcolo per l'approssimazione di f è polinomiale in $log(x)$. Se assumiamo che il test per la correttezza di f abbia la stessa velocità di calcolo, si avrà che questo algoritmo ha una velocità di calcolo polinomiale in $log(x)$.

3.3.3 Algoritmo applicato a RSA e Criteri di Esattezza

Consideriamo la solita relazione tra l'esponente pubblico e e quello privato d tenendo conto che possiamo calcolare $\varphi(n) = \varphi(pq) = mcm(p-1, q-1)$ da cui

$$ed \equiv 1 \,(mod\, mcm(p-1, q-1)) \tag{3.14}$$

da questa otteniamo che deve esistere un intero K tale che:

$$ed = K \cdot mcm(p-1, q-1) + 1 \tag{3.15}$$

ora se prendiamo $G = MCD(p-1, q-1)$ e usiamo il fatto che $mcm(p-1, q-1) = (p-1)(q-1)/G$ otteniamo:

$$ed = \frac{K}{G}(p-1)(q-1) + 1, \tag{3.16}$$

ma è possibile che K e G abbiano fattori comuni. Quindi definiamo $k = K/MCD(K, G)$ e $g = G/MCD(K, G)$ e avremo che $k/g = K/G$ e $MCD(k, g) = 1$. Quindi si ha che:

$$ed = \frac{k}{g}(p-1)(q-1) + 1 \tag{3.17}$$

e dividendo tutto per dpq otteniamo

$$\frac{e}{pq} = \frac{k}{dg}(1-\delta), \quad dove \quad \delta = \frac{p+q-1-\frac{g}{k}}{pq} \tag{3.18}$$

Notare che abbiamo scritto le due frazioni come $f' = f(1-\delta)$, punto di partenza del nostro algoritmo e che la frazione $\frac{e}{pq}$ è formata interamente a partire da informazioni pubbliche e che è una buona stima di $\frac{k}{dg}$.
Prima di utilizzare l'algoritmo delle frazioni continue, ricordiamo che questo algoritmo trova sempre delle frazioni ridotte ai minimi termini, quindi dall'equazione $ed = K \cdot mcm(p-1, q-1) + 1$ vediamo che $MCD(K, d) = 1$. Poichè k divide K, abbiamo che $MCD(k, d) = 1$. Anche $MCD(k, g) = 1$ per definizione. Quindi $MCD(k, dg) = 1$ e possiamo usare l'algoritmo delle frazioni continue per trovare k e dg purchè δ sia abbastanza piccolo. Ora usando il fatto che $\delta < \frac{1}{\frac{3}{2}n_m d_m}$ e che $\delta = \frac{p+q-1-\frac{g}{k}}{pq}$ otteniamo che:

$$kdg < \frac{pq}{\frac{3}{2}(p+q)} \tag{3.19}$$

basta per dire che k e dg possono essere trovati. Notiamo che $(-1 - g/k)$ non compare nell'aspressione per δ perchè è piccolo rispetto a $(p+q)$.

Vediamo ora come possiamo essere sicuri che i valori di k e dg siano giusti.
In modo da semplificare il test, assumeremo che $ed > pq$. Questa condizione non è una restrizione in quanto se fisso e o d, il valore atteso per l'altro sarà approssimativamente $\frac{pq}{G}$. A meno che G sia scelto molto grande, è molto

probabile che $ed > pq$. Dall'equazione (3.17), una conseguenza del fatto che $ed > pq$, è che $k > g$. Riscrivendo la (3.17) come:

$$edg \;=\; k(p-1)(q-1) + g \tag{3.20}$$

vediamo che dividendo edg per k otteniamo un quoziente di $(p-1)(q-1)$ e un resto di g finchè $k > g$. Questo ci dà una stima per $(p-1)(q-1)$e per g. Se la stima di $(p-1)(q-1)$ è zero, k e dg calcolati non sono quelli corretti. Questo caso va escluso. Ora, la stima per $(p-1)(q-1)$ può essere usata per ottenere una stima di $\frac{p+q}{2}$ usando la seguente identità:

$$\frac{pq - (p-1)(q-1) + 1}{2} \;=\; \frac{p+q}{2}. \tag{3.21}$$

Se la stima di $\frac{p+q}{2}$ non è un intero, allora le stime di k e dg sono sbagliate. Analogamente usando l'identità:

$$\left(\frac{p+q}{2}\right)^2 - pq \;=\; \left(\frac{p-q}{2}\right)^2 \tag{3.22}$$

dalla stima di $\frac{p+q}{2}$ otteniamo una stima per $\left(\frac{p-q}{2}\right)^2$ e se questa stima è un quadrato perfetto, allora le stime di k e dg sono incorrette. Una volta trovati i valori esatti di k e dg possiamo scoprire l'esponente segreto d dividendo dg per g. Ricordiamo che g era il resto della divisione di edg per k. Volendo si possono anche ricavare facilmente p, q da $\frac{p+q}{2}, \frac{p-q}{2}$.

Se non viene fatto nulla per prevenire questo attacco basato sulle frazioni continue, allora uno può aspettarsi che g sia piccolo e $k < dg$. Sotto questa condizione, possiamo vedere da (3.19) che un esponente segreto con circa un quarto dei bit rispetto al numero di bit del modulo, può essere trovato in tempo polinomiale. Questo attacco non può essere applicato nel caso in cui l'esponente ha un numero di bit pari al modulo. Questo perchè l'attacco si basa sulle informazioni che l'esponente pubblico ci da per poter fattorizzare il modulo e nel caso normale, l'esponente pubblico può essere scelto indipendentemente dal modulo.

3.3.4 Come contrastare l'attacco

Poichè è usuale scegliere un modulo di 1024 bit, ne segue che l'esponente deve essere almeno di 256 bit per evitare questo attacco. Ma questo, come abbiamo detto, non è un bene per i dispositivi con potenza di calcolo ridotta, dove la scelta di un piccolo esponente diminuiva il tempo di calcolo. Ma non tutto è perduto in quanto Wiener stesso ci da due metodi che permettono comunque una decriptazione veloce e la protezione dall'attacco :

Modo 1 **Scelta di e grande**: poichè l'algoritmo calcola d se

$$kdg < \frac{pq}{\frac{3}{2}(p+q)},$$

basterà aumentare k o g per bloccarlo e ad esempio se io volessi aumentare k basterebbe aumentare e poichè abbiamo visto che

$$ed = \frac{k}{g}(p-1)(q-1)+1 \Rightarrow k = \frac{edg-1}{(p-1)(q-1)}$$

e ciò può essere ottenuto aggiungendo ad e un multiplo di $\varphi(n)$ e quindi lavorare con $e' = e + k\varphi(n)$ con k intero.

Modo 2 **Teorema Cinese del Resto**: qui utilizziamo il famoso Teorema Cinese dei Resti per velocizzare la decifrazione senza dover scegliere l'esponente d troppo piccolo. Supponiamo di aver scelto d tale che:

$$d_p := d \, mod \, (p-1)$$

$$d_q := d \, mod \, (q-1)$$

con d_p e d_q abbastanza piccolo, ad esempio di 128 bit, allora posso andare a decriptare il messaggio criptato C calcolando prima:

$$M_p \equiv C^{d_p} \, mod \, p$$

$$M_q \equiv C^{d_q} \, mod \, q$$

e usando poi il Teorema Cinese dei Resti per trovare l'unico valore M che risolve il sistema:

$$\begin{cases} M \equiv C^{d_p} \, mod \, p \\ M \equiv C^{d_q} \, mod \, q \end{cases}$$

Di positivo in questo metodo c'è che d può essere abbastanza grande anche se d_p e d_q non lo sono. Ma questi devono essere comunque non troppo piccoli o sarà possibile fattorizzare n.

3.3.5 Esempio di attacco di Wiener a RSA

Facciamo ora un esempio di questo attacco. Ho utilizzato un algoritmo e due funzioni implementati da me in Matlab (per facilitare i conti) che inserirò

e spiegherò passo per passo. L'utente B sceglie i numeri primi $p = 2323259$ e $q = 3434351$ ottenendo:

$$n = 2323259 \cdot 3434351 = 7978886869909$$

$$\varphi(n) = 2323258 \cdot 3434350 = 7978881112300$$

e sceglie come esponente pubblico $e = 35943202454$ e come esponente privato $d = 313$. Per prima cosa notiamo che le condizioni del teorema di Wiener sono rispettate infatti:

$$d < \frac{1}{3} n^{\frac{1}{4}} = 560.227$$

quindi è possibile eseguire l'attacco. Abbiamo bisogno di due funzioni : uno per calcolare una frazione a partire dai suoi quozienti che ho chiamato $CalcoloFC$, rappresentata nel seguente script:

```
function [ FC ,NUM,DEN ] = CalcoloFC( QUOZ )
NUM=[];
DEN=[];
L=length(QUOZ);
if L==1
    NUM(1)=QUOZ(1);
    DEN(1)=1;
elseif L==2
    NUM(1)=QUOZ(1);
    DEN(1)=1;
    NUM(2)=QUOZ(1)*QUOZ(2)+1;
    DEN(2)=QUOZ(2);
else
    NUM(1)=QUOZ(1);
    DEN(1)=1;
    NUM(2)=QUOZ(1)*QUOZ(2)+1;
    DEN(2)=QUOZ(2);
    for i=3:1:L
            NUM(i)=QUOZ(i)*NUM(i-1)+NUM(i-2);
        DEN(i)=QUOZ(i)*DEN(i-1)+DEN(i-2);
        NUM(i)
        DEN(i)
    end
end
format rat
```

```
FC=sym(NUM(L))/sym(DEN(L))
end
```

in cui in input va inserito un vettore di quozienti ad esempio $QUOZ = [q_1, q_2, q_3] = [4, 56, 21]$. L'algoritmo si basa sul processo iterativo (3.3) e dà in output la frazione relativa ai quozienti ed i numeratori e denominatori calcolati per trovarla. In questo caso la funzione dà come risultato FC ovvero la frazione $\frac{4729}{1177}$ e DEN e NUM ovvero i vettori con i numeratori calcolati 4,225,4729 e i denominatori 1,56,1177.
L'altra funzione è chiamata $CalcoloQZ$ e calcola i quozienti a partire da una frazione, implementata nel seguente modo:

```
function [ q,dr,nr ] = CalcoloQZ( nf,df )
nr=[];
dr=[];
q(1)=floor(nf/df);
dr(1)=df;
nr(1)=nf-df*q(1);
i=1;
while nr(i)~=0
    i=i+1;
    q(i)=floor(dr(i-1)/nr(i-1));
    dr(i)=nr(i-1);
    nr(i)=dr(i-1)-q(i)*nr(i-1);
end
end
```

Questa funzione è basata sulla formule (2.13),(2.14), prende in input il valore del numeratore nf e del denominatore della frazione df che ci interessa e dà in output il vettore dei quozienti q, quello dei numeratori dei vari resti ottenuti nr e dei denominatori dr. Quindi mettendo in input la frazione $\frac{4729}{1177}$ otterremo i suoi quozienti $4, 56, 21$ e i numeratori dei resti che sono $21, 1, 0$ e i denominatori $1177, 21, 1$.

Ora abbiamo bisogno dello script in cui è descritto l'attacco di Wiener che richiamerà più volte le funzioni descritte qui sopra:

```
disp('ATTACCO DI WIENER');
nf=input('Inserire l ESPONENTE PUBBLICO: ');
df=input('Inserire il MODULO: ')
format rat
%f è la mia frazione di partenza di cui voglio trovare i quozienti e resti,
%che usero per calcolare l'esponente privato
f=nf/df;
```

```
%calcolo i quozienti e i resti della frazione nf/df
[q,dr,nr ]=CalcoloQZ(nf,df);
disp('i quozienti zono')
q
disp('i numeratori dei resti sono')
nr
disp('i numeratori dei resti sono')
dr

%calcolo NUM il vettore con tutti i numeratori e DEN quello con i
%denominatori
[FC,NUM,DEN]=CalcoloFC(q);
disp('i numeratori sono')
NUM
disp('i denominatori sono sono')
DEN

%ora scrivo tutte le frazioni ni/di in un vettore FCS
FCS=[];
for i=1:1:length(NUM)
    FCS(i)=NUM(i)/DEN(i);
end
disp('tutte le frazioni n/d sono')
FCS
%PQ è il vettore in cui inserirò le stime di (p-1)(q-1)
PQ=[]

%troviamo i quozienti per stimare k/dg e li metto nel vettore QFC
for i=1:1:length(FCS)
    QFC=q(:,[1:i]);
    if mod(i,2)~=0
        QFC(i)=QFC(i)+1;
    end
    QFC
%qui calcolo una stima di k/dg cioè KDG
    [KDG,NUM,DEN]=CalcoloFC(QFC);
    KDG
    V=length(DEN);
    K=NUM(V)
    DG=DEN(V)
```

```
    EDG=nf*DG
PQ=fix(EDG/K)
%primo criterio di esattezza
    if PQ==0
        continue
    end
    G=mod(EDG,K);
    PPQ=(df-PQ+1)/2
    t=PPQ-floor(PPQ);
%secondo criterio di esattezza
    if t~=0
        continue
    end
    PMQ=(PPQ^2)-df;
    m=sqrt(PMQ)-floor(sqrt(PMQ));
%terzo criterio di esattezza
if m~=0
        continue
    else
        d=DG/G
        break
    end
end
```

Questo algoritmo chiede in input la chiave pubblica di B e va a calcolare la chiave privata di B d.
Andiamo a studiare l'algoritmo passo per passo e portiamo avanti il nostro esempio.
I numeri da inserire in input sono appunto la chiave pubblica e quindi $e = 3594320245477$ e $n = 7978886869909$.
La prima cosa da fare è calcolare i quozienti della frazione $f = \frac{3594320245477}{7978886869909}$ quindi usiamo la funzione $CalcoloQZ(3594320245477, 7978886869909)$ e otteniamo i seguenti valori:

Quozienti	**Numeratori resti**	**Denominatori resti**
0	7978886869909	3594320245477
2	790246378955	3594320245477
4	433334729657	790246378955
1	356911649298	433334729657
1	76423080359	356911649298
4	51219327862	76423080359
1	25203752497	51219327862
2	811822868	25203752497
31	37243589	811822868
21	29707499	37243589
1	7536090	29707499
3	7099229	7536090
1	436861	7099229
16	109453	436861
3	108502	109453
1	951	108502
114	88	951
10	71	88
1	17	71
4	3	17
5	2	3
1	1	2
2	0	1

Ora attravero la funzione:

$$CalcoloFC([0, 2, 4, 1, 1, 4, 1, 2, 31, 21, 1, 3, 1, 16, 3, 1, 114, 10, 1, 4, 5, 1, 2])$$

otteniamo i vari denominatori d_i e numeratori n_i delle frazioni $[0], [0, 2], [0, 2, 4] \ldots$. Dopo di ciò c'è il ciclo for, dove seguendo la costruzione (3.8) vengono calcolate la varie stime di k/dg e poi usando le formule (3.21),(3.22) calcoliamo $edg, g, (p+q)/2, ((p-q)/2)^2$ e verifichiamo attraverso 3 if i criteri di esattezza da cui seguirà il risultato:

n_i/d_i	k/dg	edg	g
0	1	3594320245477	0
1/2	1/2	7188640490954	0
4/9	5/11	39537522700247	2
5/11	5/11	39537522700247	2
9/20	14/31	111423927609787	9
41/91	41/91	327083142338407	38
50/111	91/202	726052689586354	39
141/313	141/313	1125022236834301	1

(p-1)(q-1)	(p+q)/2	$((p-q)/2)^2$	d
3594320245477	2192283312217	stop	
7188640490954	395123189478	stop	
7907504540049	35691164931	stop	
7907504540049	35691164931	stop	
7958851972127	10017448892	stop	
7977637618009	624625951	stop	
7978600984465	142942723	stop	
7978881112300	2878805	308631358116	313

L'attacco ad RSA da quindi come risultato :

$$d = 313,\ p = 2323259,\ q = 3434351\, k = 141\, g = 1$$

infatti si ha che:

$$de \equiv 1\, mod\, 7978881112300.$$

Bibliografia

[1] M. J. Wiener,*Cryptanalysis of short RSA secret exponents* (1990)

[2] A. Dujella,*Continued Fractions and RSA whit small secret exponent*, (2004)

[3] D. Boneh*Twenty years of attacks on the RSA cryptosystem* (1999)

[4] Baldoni, Ciliberto, Piacentini Cattaneo, *Aritmetica, Crittografia e Codici* (2006). Springer-Verlag, Milano

[5] Douglas R. Stinson *Cryptography theory and practice 3ed* (2006)