9 Metodi di Attacco al Crittosistema RSA

9.1 Radici quadrate modulo un intero

I principali metodi di attacco al crittosistema RSA sono quelli che si basano sulla fattorizzazione di n. Prima di vedere questo, vediamo alcune informazioni di aritmetica modulare che saranno utili per il seguito.

Siano a, n due interi, n dispari tale che $\gcd(n, a) = 1$. Vogliamo determinare il numero N delle soluzioni $y \in \mathbb{Z}_n$ tale che

$$y^2 \equiv a \ mod \ n. \tag{9.1}$$

Abbiamo già visto che, se n è primo allora N è 2 o 0 a seconda che il simbolo di Legendre $\left(\frac{a}{n}\right)$ sia uguale a 1 o -1, rispettivamente.

Proposizione 9.1. Se $n = p^e$, con p primo dispari tale che $\gcd(a, p) = 1$ ed e intero positivo, allora N è 2 o 0 a seconda che il simbolo di Legendre $\left(\frac{a}{p}\right)$ sia uguale a 1 o -1, rispettivamente.

Dimostrazione. Da (9.1) si ha $y^2 \equiv a \ mod \ p$, quindi $N = 0$ se $\left(\frac{a}{p}\right) = -1$. Pertanto, supponiamo che $\left(\frac{a}{p}\right) = 1$ e sia y_0 un intero tale che $y_0^2 \equiv a \ mod \ p$. Siccome $y_0^2 \equiv a \ mod \ p^{e-1}$, allora esiste $k \in \mathbb{Z}$ tale che $y_0^2 = a + kp^{e-1}$. Siccome $\gcd(2y_0, p) = 1$, allora sia $d \in \mathbb{Z}$ tale che $2y_0 d \equiv 1 \ mod \ p^{e-1}$ e sia quindi

$$y_1 = y_0 - kdp^{e-1}.$$

Quindi

$$\begin{aligned}
y_1^2 &= \left(y_0 - kdp^{e-1}\right)^2 = \\
&= y_0^2 + k^2d^2p^{2(e-1)} - kp^{e-1} = \\
&= a + kp^{e-1} + k^2d^2p^{2(e-1)} - kp^{e-1} = \\
&= a + k^2d^2p^{2(e-1)}
\end{aligned}$$

e quindi $y_1^2 \equiv a \ mod \ p^e$. Pertanto, $N = 2$.

□

Teorema 9.2. *Sia $n > 1$ un intero dispari avente fattorizzazione*

$$n = \prod_{i=1}^{\ell} p_i^{e_i},$$

dove p_i è primo e e_i è un intero positivo, $i = 1, ..., \ell$. Se $\gcd(n, a) = 1$, allora N è 2^ℓ se per ogni $i = 1, ..., \ell$ il simbolo di Legendre $\left(\frac{a}{p_i}\right) = 1$, e 0 altrimenti.

Dimostrazione. Se esiste $i_0 \in \{1, ..., \ell\}$ tale che $\left(\frac{a}{p_{i_0}}\right) = -1$, allora $N = 0$ siccome (9.1) implica $y^2 \equiv a \ mod \ p_{i_0}$. Pertanto, $\left(\frac{a}{p_i}\right) = 1$ per ogni $i = 1, ..., \ell$. Pertanto, $y^2 \equiv a \ mod \ p_i^{e_i}$ ammette 2 soluzioni b_{i1} e b_{i2} per la **Proposizione 9.1**. Quindi, Per il **Teorema Cinese dei Resti**, per ogni $(b_{1j_1}, ..., b_{\ell j_\ell})$ il sistema congruenziale

$$\begin{cases} y \equiv b_{1j_1} \ mod \ p_1^{e_1} \\ \vdots \\ y \equiv b_{\ell j_\ell} \ mod \ p_\ell^{e_\ell} \end{cases}$$

ha un unica soluzione modulo n. Pertanto, $N = 2^\ell$ siccome il numero di tali ℓ-ple $(b_{1j_1}, ..., b_{\ell j_\ell})$ è proprio 2^ℓ.

□

9.2 Metodi di attacco al crittosistema RSA

Il metodo più ovvio per violare il crittosistema RSA consiste nel fattorizzare. Vediamo alcuni metodi utilizzati per fattorizzare n.

Divisione: Siccome n è composto, esiste un primo p tale che $p \leq \lfloor\sqrt{n}\rfloor$. Pertanto, tale metodo consiste nel dividere n con ogni intero dispari minore o uguale a $\lfloor\sqrt{n}\rfloor$. Siffatto metodo è considerato ragionevole se $n < 10^{12}$.

Algoritmo $p-1$ di Pollard (1974): Siano n, B interi. Supponiamo che esista un primo p divisore di n tale che se $p - 1 = \prod_{i=1}^{k} q_i^{\alpha_i}$, allora $q_i^{\alpha_i} \leq B$. Pertanto $(p-1) \mid B!$. Quindi, esiste un $2 \leq j_0 \leq B!$ tale che $j_0 = k(p-1)$. Pertanto, per il **piccolo Teorema di Fermat**, vale che $2^{j_0} \equiv 1 \ mod \ p$. Quindi, $p \mid \gcd(2^{j_0} - 1, n)$. Se $\gcd(2^{j_0} - 1, n) < n$, allora $\gcd(2^{j_0} - 1, n)$ è un fattore proprio di n.

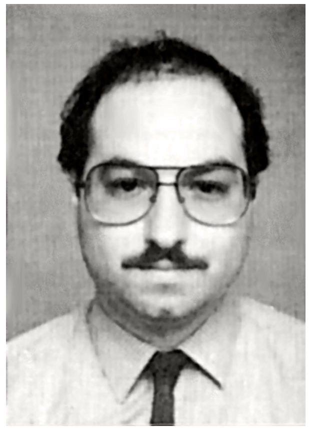


Figura 9.1: Jonathan Jay Pollard (1954)

Algoritmo 9.3. **($p-1$ di Pollard (n, B))**

$a \longleftarrow 2$
for $j \longleftarrow 2$ **to** B
do $\begin{cases} a \longleftarrow a^j \ mod\ n \\ d \longleftarrow \gcd(a^j \ mod\ n - 1, n) \end{cases}$
if $1 < d < n$
then return (d)
else return ("insuccesso")

Chiaramente, l'**Algoritmo 9.3** è di tipo Las Vegas. Infatti, perché esso funzioni, n deve ammettere un divisore primo p tale che $p-1$ sia costituito da potenze di primi piccoli, minori o uguali al bound B prefissato.

Esempio 9.4. Si consideri $n = 15770708441$ e $B = 180$ allora $a = 11620221425$ e $\gcd(a-1, n) = 135979$. Quindi

$$15770708441 = 135979 \times 115979$$

e l'algoritmo funziona siccome $135978 = 2 \times 3 \times 131 \times 173$. Quindi, in realtà il metodo funziona per ogni $B \geq 173$.

Nell'algoritmo $p-1$ si eseguono $B-1$ esponenziazioni modulari e $B-1$ calcoli di gcd. Ogni esponensiazione modulare richiede $\ln B \ln^2 n$ bit e ogni calcolo di gcd richiede $\ln^2 n$. Pertanto, la complessità dell'algoritmo $p-1$ è $O(B \ln B \ln^2 n)$. Chiaramente per B vicino a $\sqrt{n}$ la probabilità di successo aumenta ma la velocità di esecuzione e circa quella delle divisioni .
Si noti che, per rendere l'RSA immune da siffatti tipi di attacco si considera $n = pq$ con $p = 2p_1 + 1$ e $q = 2q_1 + 1$ con p_1, q_1 primi elevati.

- **Algoritmo di Lenstra.** È un generalizzazione del metodo di $p-1$ di Pollard basato sulle curve elittiche (lo vedremo più avanti).
- **Algoritmo Rho di Pollard.** L'idea che sta alla base dell'algoritmo è che se p il minimo divisore primo di n ed esistono due interi x, x' minori di n tali che $x \neq x'$ e $x \equiv x' \ mod\ p$, allora

  $$p \leq \gcd(x - x', n) < n.$$

  E' così determinato un fattore non banale di n.

Più pecisamente, si sceglie $X \subseteq \mathbb{Z}_n$ e per ogni coppia (x, x') di elementi distinti di X si calcola $\gcd(x - x', n)$. L'algoritmo ha successo se l'applicazione $x \longmapsto x \ mod\ p$ ammette una collisione in X. Utilizzando il **paradosso del compleanno**, ciò si verifica con una probabilità de 50% se $|X| \simeq 1.17\sqrt{p}$. Tuttavia, siccome p è sconosciuto, la collisione è determinata solo valutando $\gcd(x - x', n)$ per ogni coppia (x, x') di elementi distinti di X.

Pertanto, per determinare una collisione, il numero dei gcd che bisogna calcolare è $\binom{|X|}{2} > p/2$ che è un numero elevato. Per ridurre sia il numero dei gcd da calcolare che la memoria da utilizzare, l'algoritmo Rho di Pollard procede come segue:

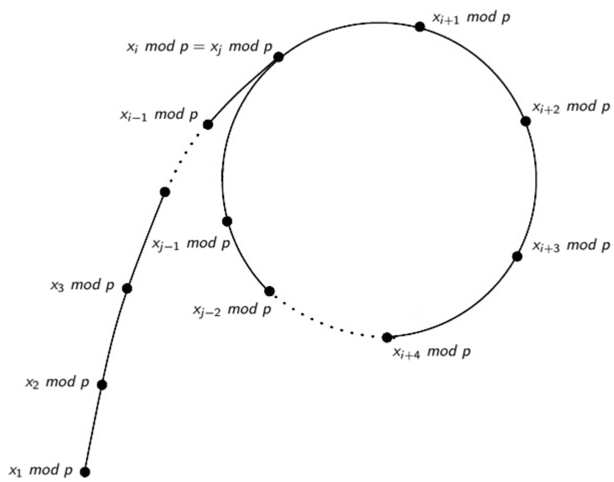
1. Si considera $f \in \mathbb{Z}[x]$ e $x_1 \in \mathbb{Z}_n$ (generalmente si considera $f(x) = x^2 + a$ e nella maggior parte dei casi $a = 1$);
2. Si considera la sequenza definita da $x_{j+1} \equiv f(x_j) \ mod\ n$ per $2 \leq j \leq m-1$ (si determina così $X = \{x_1, ..., x_m\}$ di m elementi che assumiamo essere tutti distinti);
3. Ogni volta che si determina un x_j si valuta $\gcd(x_j - x_i, n)$ per ogni $j < i$.

Lemma 9.5. *Vale che*

$$x_i \equiv x_j \ mod\ p \Rightarrow \forall \delta \in \mathbb{N} : x_{i+\delta} \equiv x_{j+\delta} \ mod\ p. \tag{9.2}$$

Dimostrazione. Siccome esiste una collisione $x_i \equiv x_j \ mod\ p$ per qualche $2 \leq i, j \leq m-1$ allora $f^{\delta}(x_i) \equiv f^{\delta}(x_j) \ mod\ p$, ovvero $x_{i+\delta} \equiv x_{j+\delta} \ mod\ p$, per ogni $\delta \geq 0$.

□



Vediamo un esempio.

Esempio 9.6. Sia $n = 7171 = 71 \times 101$ e $f(x) = x^2 + 1$ e $x_1 = 1$ allora la sequenza degli x_i è

1	2	5	26	667	6557	4105
6347	4903	2218	219	4936	4210	4560
4872	375	4377	4389	2016	5471	88

e ridotti modulo 71 si ha

1	2	5	26	28	25	58
28	4	17	6	37	21	16
44	20	46	58	28	4	17

e quindi la prima collisione è $x_7 \ mod \ 71 = x_{18} \ mod \ 71 = 58$.
Infatti, $\gcd(x_7 - x_{18}, 7171) = \gcd(4105 - 4389, 7171) = 71$.

Lemma 9.7. ***(Trucco di Floyd)***
Se $x_i \equiv x_j \ mod \ p$ allora esiste $i' \in \{i, i+1, ..., j-1\}$ tale che $x_{i'} \equiv x_{2i'} \ mod \ p$.

Dimostrazione. Sia $\ell = j - i$, poichè $\{i, i+1, \cdots, j-1\}$ è un sistema completo di residui modulo ℓ, esiste $i' \in \{i, i+1, \cdots, j-1\}$ tale che $i' = k_0\ell$. Sia $\delta = i' - i$, allora vale

$$x_{i'} \equiv x_{i'+\ell} \ mod \ p.$$

Ora applicando (9.2) con i' al posto di i, $i' + \ell$ al posto di j ed ℓ al posto di δ, si ha $x_{i'+\ell} \equiv x_{i'+2\ell}$ *mod* p. Pertanto, $x_{i'} \equiv x_{i'+2\ell}$ *mod* p e induttivamente

$$x_{i'} \equiv x_{j'} \ mod\ p \text{ con } j' = i' + k\ell, k \in \mathbb{N}.$$

Se $k = k_0$, allora $j' = 2i'$ e quindi vale

$$x_{i'} \equiv x_{2i'} \ mod\ p.$$

□

Algoritmo 9.8. **(Rho di Pollard** (n, x_1)**)**

esternal f
$x \longleftarrow x_1$
$x' \longleftarrow f(x)\ mod\ n$
$p \longleftarrow \gcd(x - x', n)$
while $p = 1$

$$\textbf{do} \begin{cases} \textbf{commenti: } \text{nella } i\text{-esima iterazione} \\ x \longleftarrow f(x)\ mod\ n \\ x' \longleftarrow f(x')\ mod\ n \\ x' \longleftarrow f(x')\ mod\ n \\ p \longleftarrow \gcd(x - x', n) \end{cases}$$

if $p = n$
then return ("insuccesso")
else return (p)

Sulla base del **Lemma 9.7** si cercano collisioni del tipo $x_{i'} \equiv x_{2i'}$ *mod* p per qualche $i' \in \{i, i+1, ..., j-1\}$. Quindi, all'iterazione i-esima si calcola solo $\gcd(x_{2i} - x_i, n)$. Essendo, il numero di iterazioni per determinare un fattore p primo di n e di circa $\sqrt{p}$. Si noti che in questo modo non si determina la prima collisione, se ne esistono, a vantaggio del minor numero di gcd da calcolare.

Ritornando all'esempio precedente, si ha che la prima collisione per $i = 7$ e $j = 18$ quindi $\ell = 11$, il primo $i' \geq i$ tale che si abbia una collisione è $i = 11$. Infatti $\gcd(x_{22} - x_{11}, n) = \gcd(7745 - 219, 7171) = 71$.

Si prova che la probabilià di successo è circa $p/n < 1/\sqrt{n}$ e la complessità dell'**Algoritmo Rho di Polard** è $O(\sqrt[4]{n} \ln n)$.

9.3 Algoritmo di Dixon per i quadrati casuali

L'idea che sta alla base è molto semplice: è quella di determinare due interi x, y tali che $x \not\equiv \pm y \; mod \; n$ ma $x^2 \equiv y^2 \; mod \; n$. Pertanto $n \mid (x+y)(x-y)$ ma non divide $x-y$ o $x+y$. Pertanto $\gcd(n, x-y)$ e $\gcd(n, x+y)$ sono fattori non banali di n.

L'algoritmo opera come segue:

(1) Si fissa $\mathcal{B} = \{p_1, ..., p_b\}$ un sottoinsieme (**base**) di primi, eventualmente unito a $\{-1\}$.

(2) Si determinano $z_1, ..., z_c$ interi casuali con $c > b$ e di ognuno di essi calcola il quadrato e lo si riduce modulo n. Generalmente, interi della forma $j + \lceil \sqrt{kn} \rceil$ con $j = 0, 1, 2...$ e $k = 1, 2, ..$ che tendono produrre corrispondenti $z^2 \; mod \; n$ relativamente piccoli e quindi completamente fattorizzabili rispetto i primi della base. Oppure interi oppure della forma $\lfloor \sqrt{kn} \rfloor$, tali che $z^2 \; mod \; n$ è vicino ad n e quindi $-z^2 \; mod \; n$ è potenzialmente completamente fattorizzabili rispetto i primi della base $\mathcal{B}$.

(3) Per ogni $1 \leq j \leq c$

$$z_j^2 \equiv p_1^{\alpha_{1j}} \times p_2^{\alpha_{2j}} \cdots \times p_b^{\alpha_{bj}} \; mod \; n$$

e si considera il vettore di $\mathbb{Z}_2^b$ definito da

$$\vec{a}_i = (\alpha_{1j} \; mod \; 2, \alpha_{2j} \; mod \; 2, \ldots, \alpha_{bj} \; mod \; 2)$$

(4) Poichè $c > b$, i vettori $\vec{a}_1, \ldots \vec{a}_c$ sono linearmente dipendenti. Quindi esiste $X \subseteq \{1, .., c\}$ tali $\sum_{j \in X} \vec{a}_j = \vec{0}$ e pertanto

$$\prod_{j \in X} z_j^2 \equiv \prod_{i=1}^{b} p_i^{\sum_{j \in X} \alpha_{ij}} \; mod \; n$$

siccome $\sum_{j \in X} \vec{a}_j = \vec{0}$, allora $\sum_{j \in X} \alpha_{ij} = 2k_i$ per ogni $j \in X$. Quindi, posto $x = \prod_{j \in X} z_j$ e $y = \prod_{i=1}^{b} p_i^{k_i}$ vale che $x^2 \equiv y^2 \; mod \; n$. Da qui si procede al calcolo di $\gcd(n, x-y)$ o di $\gcd(n, x+y)$.

Esempio 9.9. Fattorizziamo $n = 1829$ attraverso i quadrati casuali di Dixon.

Sia $\mathcal{B} = \{-1, 2, 3, 5, 7, 11, 13\}$ calcoliamo $\sqrt{n} = 42.77$, $\sqrt{2n} = 60.48$, $\sqrt{3n} = 74.03$ e $\sqrt{4n} = 85.53$. Consideriamo gli interi

$$z = 42, 43, 61, 62, 74, 75, 85, 86.$$

Allora

$$\begin{aligned}
z_1^2 &\equiv 42^2 \equiv -65 \equiv -1 \times 5 \times 13 \ \ mod\ 1829\\
z_2^2 &\equiv 43^2 \equiv 20 \equiv 2^2 \times 5 \ \ mod\ 1829\\
z_3^2 &\equiv 61^2 \equiv 63 \equiv 3^2 \times 7 \ \ mod\ 1829\\
z_4^2 &\equiv 74^2 \equiv -11 \equiv -1 \times 11 \ \ mod\ 1829\\
z_5^2 &\equiv 85^2 \equiv -91 \equiv -1 \times 7 \times 13 \ \ mod\ 1829\\
z_6^2 &\equiv 86^2 \equiv 80 \equiv 2^4 \times 5 \ \ mod\ 1829
\end{aligned}$$

Quindi

$$\begin{aligned}
\vec{a}_1 &= (1,0,0,1,0,0,1)\\
\vec{a}_2 &= (0,0,0,1,0,0,0)\\
\vec{a}_3 &= (0,0,0,0,1,0,0)\\
\vec{a}_4 &= (1,0,0,0,0,1,0)\\
\vec{a}_5 &= (1,0,0,0,1,0,1)\\
\vec{a}_6 &= (0,0,0,1,0,0,0).
\end{aligned}$$

Chiaramente $\vec{a}_2 + \vec{a}_6 = \vec{0}$ ma non produce alcuna fattorizzazione di n, invece $\vec{a}_1 + \vec{a}_2 + \vec{a}_3 + \vec{a}_5 = \vec{0}$ produce

$$(42 \times 43 \times 61 \times 85)^2 \equiv (2 \times 3 \times 5 \times 7 \times 13)^2 \ \ mod\ 1829$$

ovvero $1459^2 \equiv 901^2$ mod 1829 da cui si ricava che $\gcd(1459 + 901, 1829) = 59$ e quindi $1829 = 31 \times 59$.

□

9.4 Altri metodi di attacco

9.4.1 Calcolo di $\varphi(n)$

Proposizione 9.10. Sia $n = pq$, con p, q primi distinti allora valgono i seguenti fatti:

1. la conoscenza della fattorizzazione n è equivalente alla conoscenza di $\varphi(n)$.
2. La complessità del calcolo di $\varphi(n)$ a partire dalla conoscenza di n è $O(\log n)$.
3. La complessità del calcolo della fattorizzazione di n a partire dalla conoscenza di $\varphi(n)$ è $O(\log^3 n)$.

Dimostrazione. Infatti, se $n = pq$, segue da

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

e quindi la complessità del calcolo di $\varphi(n)$ è $O(\log n)$.

Viceversa, noto $\varphi(n)$, allora $p + q = n - \varphi(n) + 1$, allora p, q sono le soluzioni di

$$X^2 - (n - \varphi(n) + 1)\, X + n = 0.$$

Quindi,

$$p, q = \frac{(n - \varphi(n) + 1) \pm \sqrt{(n - \varphi(n) + 1)^2 - 4n}}{2}$$

da cui si ricava che la sua complessità $O(\log^3 n)$.

□

Esempio 9.11. Siano noti $n = 84773093$ e $\varphi(n) = 84754668$. Allora

$$x^2 - 18426x + 84773093 = 0$$

ha come soluzioni $p = 8887$ e $q = 9539$.

9.4.2 L'esponente di decifratura

In questa sezione mostriamo che la conoscenza dell'esponente di decifratura d comporta la fattorizzazione di n in tempi polinomiali.

Sia $n = pq$ con p e q primi distinti e siano e e d le chiavi di cifratura e decifratura di un generico utente che utilizza il crittosistema RSA. Allora $ed - 1 = 2^s r \equiv 0 \ mod\ \varphi(n)$ (r dispari). Se w è un intero tale che $\gcd(w, n) = 1$, dal **Teorema di Eulero** discende che

$$w^{2^s r} \equiv 1 \ mod\ n.$$

Sia $t = \min\{0, ..., s\}$ tale che $w^{2^t r} \equiv 1 \ mod\ n$. Se $w^{2^{t-1} r} \not\equiv -1 \ mod\ n$ per $t > 0$, allora $w^{2^{t-1} r}$ è una radice quadrata non banale di 1 in $\mathbb{Z}_n$. Pertanto

$$1 < \gcd(w^{2^{t-1} r} + 1, n) < n,$$

cioè $\gcd(w^{2^{t-1} r} + 1, n) \in \{p, q\}$ e quindi n è fattorizzato.

Ciò non si verifica nel caso in cui w sia un intero tale che $\gcd(w, n) = 1$ e vale una delle seguenti congruenze:

1. $w^r \equiv 1 \ mod\ n$
2. $w^{2^t r} \equiv -1 \ mod\ n$ con qualche intero t tale che $0 \le t \le s - 1$.

Cioè non si verifica nel caso in cui w è una base rispetto alla quale n è uno pseudoprimo di Eulero. Pertanto, la probabilità di fattorizzare n corrisponde alla probabilità di trovare una base rispetto alla quale n non è uno pseudoprimo di Eulero. Tale probabilità, abbiamo visto essere maggiore uguale ad 1/2 per il **Teorema 8.26** .

Esempio 9.12. Siano $n = 89855713$, $e = 34986517$, $d = 82330933$ e il valore casuale $w = 5$. Allora

$$ed - 1 = 2^3 \times 360059073378795.$$

Quindi $s = 3$ e $r = 360059073378795$. Per $t = 1$

$$\begin{aligned} w^r \ mod\ n &\equiv 85877701 \\ w^{2r} \ mod\ n &\equiv 1, \end{aligned}$$

quindi

$$\gcd(85877700, 89855713) = 9871$$

e pertanto $n = 9103 \times 9871$.

Algoritmo 9.13. (RSA-Factor (n, a, b))

commento: supponiamo che $ed \equiv 1 \ mod\ \varphi(n)$
scriviamo $ed - 1 = 2^s r$ con r dispari.
scegliamo un intero causale w compreso tra 1 e $n - 1$
$x \longleftarrow \gcd(w, n)$
if $1 < x < n$
then return (x)
commento: x è un fattore di n
$v \longleftarrow w^r \ mod\ n$
if $v \equiv 1 \ mod\ n$
then return ("insuccesso")
while $v \not\equiv 1 \ mod\ n$
do $\begin{cases} v_0 \longleftarrow v \\ v \longleftarrow v^2 \ mod\ n \end{cases}$
if $v_0 \equiv -1 \ mod\ n$
then return ("insuccesso")
else $\begin{cases} x \longleftarrow \gcd(v_0 + 1, n) \\ \quad \text{return } (x) \end{cases}$
commento: x è un fattore di n

Chiaramente, l'**Algoritmo 9.13** è un $(m, 1 - \frac{1}{2^m})$-algoritmo di tipo Las vegas con complessità $O(\log^3 n)$.

9.4.3 Attacco di Wiener

In questa sezione presentiamo un attacco al crittosistema RSA dovuto a Norbert Wiener.

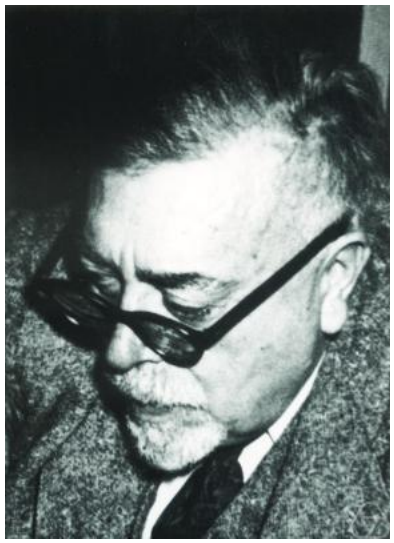


Figura 9.2: Norbert Wiener (1894 – 1964)

L'attacco permette di determinare la chiave di decifratura d nel caso in cui $n = pq$ con $q < p < 2q$ e $3d < \sqrt[4]{n}$. Siccome (n, e) è la chiave pubblica, l'idea che sta alla base è determinare d attraverso lo sviluppo in frazione continua di $\frac{e}{n}$.

Per comprendere il suddetto attacco è necessario fare una brevissima introduzione sulle frazioni continue.

Definizione 9.14. **(Frazione Continua)**
Siano $a_0, a_1, ..., a_n$ interi tali che $a_1, ..., a_n > 0$, allora un'espressione della forma

$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{\dots + a_n}}}}$$

si dice **frazione continua finita** e la si denota con $[a_0, a_1, ..., a_n]$.

Siano a, b interi positive tali che $\gcd(a, b) = 1$ e siano $(q_1, ..., q_m)$ gli interi positivi determinati attraverso l'**Algoritmo Euclideo**, ovvero, posto $r_0 = a$ e $r_1 = b$, vale che

$$\begin{array}{ll} r_0 = q_1 r_1 + r_2 & 0 < r_2 < r_1 \\ r_1 = q_2 r_2 + r_3 & 0 < r_3 < r_2 \\ \vdots & \vdots \\ r_{m-2} = q_{m-1} r_{m-1} + r_m & 0 < r_m < r_{m-1} \\ r_{m-1} = q_m r_m & \end{array}$$

Proposizione 9.15. Vale che
$$\frac{a}{b} = [q_1, ..., q_m].$$

Dimostrazione. Proviamo il seguente asserto per induzione su m. Possiamo supporre che $b \neq 1$. Quindi $m \geq 2$.

(1) $m = 2$: Quindi $r_0 = q_1 r_1 + r_2$ e $r_1 = q_2 r_2$. Quindi
$$\frac{a}{b} = \frac{r_0}{r_1} = \frac{q_1 r_1 + r_2}{r_1} = q_1 + \frac{r_2}{r_1} =$$
$$= q_1 + \frac{1}{\frac{r_1}{r_2}} = q_1 + \frac{1}{q_2} = [q_1, q_2].$$

(2) Supponiamo vero l'asserto per $m = k$ e proviamolo per $m = k + 1$. Quindi,

$$\begin{array}{ll} r_0 = q_1 r_1 + r_2 & 0 < r_2 < r_1 \\ r_1 = q_2 r_2 + r_3 & 0 < r_3 < r_2 \\ \vdots & \vdots \\ r_{k-2} = q_{k-1} r_{k-1} + r_k & 0 < r_k < r_{k-1} \\ r_{k-1} = q_k r_k + r_{k+1} & 0 < r_{k+1} < r_k \\ r_k = q_{k+1} r_{k+1} & \end{array}$$

Si noti che $\gcd(r_1, r_2) = \gcd(r_0, r_1) = 1$ e quindi applicando l'ipotesi induttiva a $\frac{r_1}{r_2}$ si ha che $\frac{r_1}{r_2} = [q_2, ..., q_{k+1}]$. D'altra parte, $\frac{a}{b} = \frac{r_0}{r_1} = q_1 + \frac{1}{\frac{r_1}{r_2}}$ e quindi $\frac{a}{b} = [q_1, q_2, ..., q_{k+1}]$.

□

- Se $\frac{a}{b} = [q_1, ..., q_m]$, allora $[q_1, ..., q_m]$ è detta **frazione continua** di $\frac{a}{b}$.
- Per ogni $1 \leq j \leq m$, definiamo **convergente j-esimo** il numero razionale $C_j = [q_1, ..., q_j]$.

Lemma 9.16. *Per ogni* $1 \leq j \leq m$ *il convergente* j*-esimo il numero razionale* $C_j = \frac{c_j}{d_j}$, *dove*

$$c_j = \begin{cases} 1 & se\ j = 0 \\ q_1 & se\ j = 1 \\ q_j c_{j-1} + c_{j-2} & se\ j \geq 2 \end{cases} \qquad d_j = \begin{cases} 1 & se\ j = 0 \\ 1 & se\ j = 1 \\ q_j d_{j-1} + d_{j-2} & se\ j \geq 2 \end{cases}$$

La **Proposizione 9.15** e il **Lemma 9.16** forniscono un metodo efficiente per determinare lo sviluppo di un qualsiasi razionale in frazione continua e per il calcolo dei convergenti, rispettivamente. Vediamo degli esempi.

Esempio 9.17. Determinare lo sviluppo di $\frac{34}{99}$ mediante frazione continua e le relative frazioni continue convergenti.

Attraverso l'**Algoritmo Euclideo** si ottiene

$$\begin{aligned} &34 = 0 \times 99 + 34 \\ &99 = 2 \times 34 + 31 \\ &34 = 1 \times 31 + 3 \\ &31 = 10 \times 3 + 1 \\ &3 = 3 \times 1 \end{aligned}$$

Quindi, lo sviluppo di $\frac{34}{99}$ mediante frazione continua è

$$\frac{34}{99} = \frac{1}{2 + \frac{1}{1 + \frac{1}{10 + \frac{1}{3}}}} = [0, 2, 1, 10, 3].$$

Le frazioni convergenti convergenti sono

$$\begin{aligned} &[0] = 0 \\ &[0, 2] = 1/2 \\ &[0, 2, 1] = 1/3 \\ &[0, 2, 1, 10] = 11/32 \\ &[0, 2, 1, 10, 3] = 34/99. \end{aligned}$$

□

Esempio 9.18. Determinare lo sviluppo di $\frac{60728973}{160523347}$ mediante frazione continua e le relative frazioni continue convergenti.

Attraverso l'**Algoritmo Euclideo** si ottiene

$$\begin{aligned}
&60728973 = 0 \times 160523347 + 60728973\\
&160523347 = 2 \times 60728973 + 39\,065\,401\\
&60728973 = 1 \times 39\,065\,401 + 21\,663\,572\\
&39\,065\,401 = 1 \times 21\,663\,572 + 17\,401\,829\\
&21\,663\,572 = 1 \times 17\,401\,829 + 4261\,743\\
&17\,401\,829 = 4 \times 4261\,743 + 354\,857\\
&4261\,743 = 12 \times 354\,857 + 3459\\
&354\,857 = 102 \times 3459 + 2039\\
&3459 = 1 \times 2039 + 1420\\
&2039 = 1 \times 1420 + 619\\
&1420 = 2 \times 619 + 182\\
&619 = 3 \times 182 + 73\\
&182 = 2 \times 73 + 36\\
&73 = 2 \times 36 + 1\\
&36 = 36 \times 1
\end{aligned}$$

Quindi,

$$\frac{60728973}{160523347} = [0, 2, 1, 1, 14, 12, 102, 1, 1, 2, 3, 2, 2, 36].$$

I primi convergenti sono

$$\begin{aligned}
&[0] = 0\\
&[0, 2] = 1/2\\
&[0, 2, 1] = 1/3\\
&[0, 2, 1, 1] = 2/5\\
&[0, 2, 1, 1, 14] = 3/8\\
&[0, 2, 1, 1, 14, 12] = 14/37\\
&\vdots
\end{aligned}$$

□

Teorema 9.19. *Siano a, b, c, d interi tali che* $\gcd(a, b) = \gcd(c, d) = 1$. *Se*

$$\left|\frac{a}{b} - \frac{c}{d}\right| < \frac{1}{2d^2}$$

allora $\frac{c}{d}$ è uno dei convergenti di $\frac{a}{b}$.

Sia $n = pq$ con p, q primi distinti e siano e, d, dove $1 < e, d < \varphi(n)$ gli esponenti di cifratura e decifratura del generico utente che utilizza il crittosistema RSA. Sia $k \in \mathbb{Z}$ tale che $ed - 1 = k\varphi(n)$.

Teorema 9.20. ***(Teorema di Wiener)***
Se $q < p < 2q$ e $3d < \sqrt[4]{n}$ allora $\frac{k}{d}$ è uno dei convergenti dello sviluppo mediante frazioni continue di $\frac{e}{n}$.

Dimostrazione. Siccome $ed - 1 = k\varphi(n)$ implica $k\varphi(n) < ed < \varphi(n)d$ e quindi $k < d$, allora

$$3k < 3d < \sqrt[4]{n}. \tag{9.3}$$

Poiché $q < p$, allora $q^2 < pq = n$ e quindi $q < \sqrt{n}$.
Inoltre, da $p < 2q$, segue che

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

Da (9.3) segue

$$0 < k(n - \varphi(n)) < 3k\sqrt{n} < \sqrt[4]{n^3}. \tag{9.4}$$

Quindi,

$$\begin{aligned}
\left|\frac{e}{n} - \frac{k}{d}\right| &= \left|\frac{ed - kn}{dn}\right| = \left|\frac{ed - 1 + 1 - kn}{dn}\right| = \left|\frac{k\varphi(n) + 1 - kn}{dn}\right| \\
&= \left|\frac{1 - k(n - \varphi(n))}{dn}\right| \leq \frac{k(n - \varphi(n))}{dn} \leq \frac{\sqrt[4]{n^3}}{dn} \\
&= \frac{1}{d\sqrt[4]{n}} \leq \frac{1}{3d^2} < \frac{1}{2d^2}.
\end{aligned}$$

Pertanto, $\frac{k}{d}$ è uno dei convergenti dell'espansione mediante frazioni continue di $\frac{e}{n}$ per il **Teorema 9.19**.

□

Algoritmo 9.21. **(Wiener (n, e))**
$(q_1, ..., q_m; r_m) \longleftarrow$ **Algoritmo Euclideo** (n, b)
$c_0 \longleftarrow 1$
$c_1 \longleftarrow q_1$
$d_0 \longleftarrow 0$
$d_1 \longleftarrow 1$
for $j \longleftarrow 1$ **to** m
do {
  $n' \longleftarrow (d_j e - 1)/c_j$
  commento: $n' = \varphi(n)$ if c_j/d_j è il convergente corretto
  if n' è un intero
  then {
    siano p e q le soluzioni dell'equazione
      $x^2 - (n - n' + 1)x + n = 0$
    if p e q sono interi positivi minori di n
    then return (p, q)
  }
  $j \longleftarrow j + 1$
  $c_j \longleftarrow q_j c_{j-1} + c_{j-2}$
  $d_j \longleftarrow q_j d_{j-1} + d_{j-2}$
}
return ("Insuccesso")

Vediamo con un esempio come funziona l'attacco di Wiener.

Esempio 9.22. Siano $n = 160523347$ e $e = 60728973$, quindi $\frac{e}{n} = \frac{60728973}{160523347}$. Si ricordi che,

$$\frac{60728973}{160523347} = [0, 2, 1, 1, 14, 12, 102, 1, 1, 2, 3, 2, 2, 36]$$

e le prime frazioni convergenti sono

$$0, \frac{1}{2}, \frac{1}{3}, \frac{2}{5}, \frac{3}{8}, \frac{14}{37}, \dots$$

$\frac{c_j}{d_j}$	$n' = \frac{d_j e - 1}{c_j}$	$x^2 - (n - n' + 1)x + n = 0$	Fattorizzazione di n
$\frac{1}{2}$	121 457 945	$x^2 - 39\,065\,403x + 160523347 = 0$	no
$\frac{1}{3}$	182 186 918	$x^2 + 21\,663\,570x + 160523347 = 0$	no
$\frac{2}{5}$	151 822 432	$x^2 - 8700\,916x + 160523347 = 0$	no
$\frac{3}{8}$	$\frac{485\,831\,783}{3}$	-	no
$\frac{14}{37}$	160 498 000	$x^2 - 25\,348x + 160523347 = 0$	$n = 12347 \times 13001$

Si noti che

$$d = e^{-1}\ mod\ (12346 \times 13000) = 37 < \frac{\sqrt[4]{n}}{3} = 37.52.$$

□