ITS ICT Piemonte

Corso di Sicurezza Informatica

Appunti di Algebra per la crittografia a chiave pubblica RSA.

Prof. Marco Farina

marco.farina@its-ictpiemonte.it

Ultima revisione: 18 gennaio 2017.

Indice

Il presente testo è tratto alle lezioni del prof. Alberto Albano presso il Dipartimento di Informatica dell'Università di Torino.
Questo testo è stato scritto usando LaTeX, sistema originariamente sviluppato da Leslie Lamport, basato su TeX creato da Donald Knuth.

1 Insiemi di numeri

1.1 Introduzione

I numeri sono così pervasivi del nostro mondo che se ne comincia a fare conoscenza nei primi anni di vita e se ne continua a fare un uso via via più approfondito nella vita quotidiana e nel percorso scolastico fino alle scuole superiori. Resta tuttavia indispensabile per gli studenti che hanno deciso di intraprendere studi di carattere scientifico o tecnico ritornarci di nuovo nei corsi che utilizzano concetti matematici di base. L'esigenza nasce dalla necessità di fare alcuni chiarimenti su alcuni aspetti delicati e profondi dei numeri, che giocano poi un ruolo fondamentale in tutta quanta la matematica in generale e la crittologia in particolare. Non si tratta tanto di questioni fondazionali sul concetto di numero, che in questa sede non verranno affrontate, quanto di questioni concrete da tenere ben presenti da chiunque voglia utilizzare lo strumento matematico con perizia e sicurezza.

1.2 Naturali, interi, razionali

I primi numeri che si incontrano sono gli interi positivi, detti anche numeri naturali: $1, 2, 3, \ldots$. L'insieme dei numeri naturali si indica con il simbolo $\mathbb{N}$. Sono i numeri che servono a contare e che hanno fatto la prima comparsa nelle società umane svariate migliaia di anni fa. Per fare misure di quantità fisiche come lunghezze, aree, tempi, temperature, ecc., è tuttavia necessario poter disporre di sottoparti dell'unità e considerare quindi numeri frazionari m/n dove $m, n \in \mathbb{N}$ con $n \neq 0$. È poi conveniente anche introdurre i numeri con il segno per poter trattare grandezze negative come possono essere la temperatura, la velocità e molte altre grandezze fisiche. Si ottengono così i seguenti insiemi numerici:

$$\mathbb{Z} = \Big\{0, \pm 1, \pm 2, \ldots\Big\} \qquad \text{numeri interi relativi,}$$
$$\mathbb{Q} = \Big\{\frac{m}{n} \mid m, n \in \mathbb{Z},\, n \neq 0\Big\} \qquad \text{numeri razionali.}$$

Si hanno le evidenti inclusioni $\mathbb{N} \subseteq \mathbb{Z} \subseteq \mathbb{Q}$. Talvolta è anche utile considerare gli insiemi

$$\mathbb{Z}^+ = \Big\{m \in \mathbb{Z} \mid m \geq 0\Big\} \qquad \text{numeri interi non negativi,}$$
$$\mathbb{Q}^+ = \Big\{q \in \mathbb{Q} \mid q \geq 0\Big\} \qquad \text{numeri razionali non negativi.}$$

2 L'algoritmo di Euclide

Da bambini abbiamo imparato che quando si divide 27 per 6 il *quoziente* è 4 ed il *resto* è 3, cioè che

$$27 = 6 \cdot 4 + 3.$$

La cosa importante da notare è che il resto deve essere un intero maggiore o uguale a 0 e minore in senso stretto di 6. Anche se è vero che, ad esempio

$$27 = 6 \cdot 3 + 9,$$

si prende come resto 3 e non 9 perché 3 è il più piccolo intero positivo per cui vale l'identità.

La proprietà fondamentale di cui ci stiamo occupando si può enunciare in modo preciso con il seguente teorema.

Teorema 2.1 (della divisione)**.** *Se $a \geq 0$ e $b > 0$ sono interi, esistono e sono univocamente determinati gli interi $q \geq 0$ ed r tali che*

$$a = bq + r \qquad e \qquad 0 \leq r < b.$$

q ed r sono detti rispettivamente il quoziente *e il* resto *della divisione.*

La determinazione del quoziente e del resto si può fare utilizzando l'algoritmo della divisione che abbiamo imparato nella scuola elementare. Ad esempio, si provi ad applicare l'algoritmo per ottenere quoziente e resto della divisione di 2731 diviso 43.

Definizione 2.1. Dati due interi a e b diciamo che b è un **divisore** di a, e scriviamo $b|a$ se, dividendo a per b si ottiene resto 0, cioè se $a = qb$. Equivalentemente si dice che b **divide** a, oppure che a è divisibile per b, oppure ancora che a è un **multiplo** di b.

Definizione 2.2. Dati due interi a e b, il numero positivo d tale che

1. $d \mid a$ e $d \mid b$,
2. se $c \mid a$ e $c \mid b$ allora $c \mid d$

è detto il **massimo comun divisore** di a e b e denotato con

$$d = (a, b).$$

La condizione 1 dice che d è un divisore comune di a e b, e la condizione 2 dice che ogni divisore comune di a e b è anche un divisore di d. Per esempio, 6 è un divisore comune di 60 ed 84, ma non è il massimo comun divisore, dato che $12 \mid 60$ e $12 \mid 84$, ma $12 \nmid 6$ (il simbolo $\nmid$ significa "non divide").

Definizione 2.3. Due numeri a, b sono detti **coprimi** se $(a, b) = 1$.

Esiste un famoso metodo per calcolare il massimo comun divisore di due interi, basato sull'uso del quoziente e del resto.

Teorema 2.2 (Algoritmo di Euclide)**.** *Il massimo comun divisore di due interi a, b si ottiene con il metodo delle divisioni successive seguenti:*

$$\begin{aligned} a &= bq + r_0 \\ b &= r_0 q_0 + r_1 \\ r_0 &= r_1 q_1 + r_2 \\ &\vdots \\ r_{n-2} &= r_{n-1} q_{n-1} + r_n \\ r_{n-1} &= r_n q_n + 0 \end{aligned}$$

Poiché i resti r_i, con $0 \leq i \leq n$, sono positivi e decrescenti, dopo un numero finito di passi la divisione dà resto 0. L'ultimo resto non nullo r_n è il massimo comun divisore di a, b.

Se già il primo resto è uguale a zero, b divide a e quindi $(a, b) = b$.

Esempio. Determinare il massimo comun divisore di 1776 e 1492.
Soluzione. Si ha:

$$\begin{aligned} (1776, 1492) &= (1492, 284), && \text{essendo} && \mathbf{1776} = 1 \cdot \mathbf{1492} + \mathbf{284} \\ (1492, 284) &= (284, 72), && \text{essendo} && \mathbf{1492} = 5 \cdot \mathbf{284} + \mathbf{72} \\ (284, 72) &= (72, 68), && \text{essendo} && \mathbf{284} = 3 \cdot \mathbf{72} + \mathbf{68} \\ (72, 68) &= (68, 4), && \text{essendo} && \mathbf{72} = 1 \cdot \mathbf{68} + \mathbf{4} \\ (68, 4) &= 4, && \text{essendo} && \mathbf{68} = 17 \cdot \mathbf{4}. \end{aligned}$$

Teorema 2.3 (Identità di Bezout)**.** *Se $d = (a, b)$ è il massimo comun divisore di a e b, allora esistono due interi m ed n tali che*

$$d = ma + nb.$$

*$ma + nb$ è detta una **combinazione lineare** di a e b con coefficienti m ed n.*

3 I numeri primi

Un concetto di fondamentale importanza nella storia e nella pratica della matematica è quello di **numero primo**. Un numero intero p si dice primo se

- $p \neq 0, 1, -1$;
- per ogni $a \in \mathbb{Z}$ se a divide p allora $a \in \{1, -1, p, -p\}$.

In altre parole, un intero è un primo se è diverso da 0, 1,-1, e non ha divisori propri.

Il Lemma seguente descrive un'importante proprietà dei numeri primi.

Lemma 3.1. *Siano $a, b, p \in \mathbb{Z}$ con p primo:*

$$se \quad p \mid ab \quad allora \quad p \mid a \quad o \quad p \mid b.$$

Dimostrazione. supponiamo che $p|ab$. Se $p|a$ la dimostrazione è banale; assumiamo quindi anche che $p \nmid a$. Allora $(a, p) = 1$, quindi esistono $x, y \in \mathbb{Z}$ tali che $xa + yp = 1$, da cui si ottiene

$$b = 1 \cdot b = xab + ypb;$$

poiché $p|ab$, segue che p divide b. □

Dimostriamo ora il cosiddetto Teorema Fondamentale dell'Aritmetica.

Teorema 3.1. *Sia $z \in \mathbb{Z}$ un intero diverso da 0,1,-1. Allora esistono numeri primi $p_1, p_2, \ldots, p_n$ tali che*

$$z = p_1 \cdot p_2 \cdot p_3 \cdot \ldots \cdot p_n.$$

Inoltre tale fattorizzazione è unica a meno del segno dei numeri primi e del loro ordine nel prodotto.

Dimostrazione. (esistenza) Supponiamo prima $z > 0$ (quindi $z \geq 2$) e applichiamo il principio di induzione nella seconda forma.

Se $z = 2$ allora la dimostrazione è banale. Supponiamo ora che $z \geq 3$ e che, per ipotesi induttiva, una fattorizzazione in prodotto di primi esista per ogni $2 \leq k \leq z - 1$.

Se z è primo, allora è già fattorizzato (con un solo fattore). Supponiamo quindi che z non sia primo. Allora z ha almeno un divisore proprio k; quindi

$z = kb$ con $2 \leq k, b \leq z-1$. Ma, per ipotesi induttiva, k e b sono un prodotto di numeri primi, e quindi anche z è tale.

Sia ora $z < 0$; allora $-z > 0$ e quindi, per quanto appena visto, $-z = p_1 \cdot p_2 \cdot p_3 \cdot \ldots \cdot p_n$, con $p_1 \cdot p_2 \cdot \ldots \cdot p_n$ numeri primi; quindi $z = (-p_1) \cdot p_2 \cdot p_3 \cdot \ldots \cdot p_n$. La prova di esistenza è completata. □

Dimostrazione. (unicità) Supponiamo che $p_1, p_2, \ldots, p_n$ e $q_1, q_2, \ldots, q_s$ siano primi tali che

$$p_1 \cdot p_2 \ldots p_n = z = q_1 \cdot q_2 \ldots q_s.$$

Allora $p_1 | z = q_1 \cdot q_2 \ldots q_s$, quindi per l'osservazione che segue il Lemma precedente, p_1 divide almeno uno dei q_i A meno di riordinare $q_1 \cdot q_2 \ldots q_s$ possiamo supporre che $p_1 | q_1$, ma allora, essendo primi, $p_1 = q_1$ oppure $p_1 = -q_1$.

Dividendo ora z per p_1 si ottiene dunque

$$p_2 \cdot p_3 \ldots p_n = \frac{z}{p_1} = \pm q_2 \cdot q_3 \ldots q_s.$$

Procedendo in questo modo alla fine si ricava $n = s$, ed anche l'unicità dei primi nelle due fattorizzazioni, a meno dell'ordine e dei segni. □

Teorema 3.2 (di Euclide)**.** *Esistono infiniti numeri primi positivi.*

Dimostrazione. Supponiamo per assurdo che l'insieme dei numeri primi positivi sia finito, e siano allora $p_1, p_2, \ldots, p_t$ tutti i numeri primi positivi distinti. Consideriamo il numero intero

$$N = p_1 \cdot p_2 \cdot p_3 \ldots p_t + 1.$$

Allora $N \geq 2$ e c'è un fattore primo q di N. Essendo primo, q deve essere uno dei p_i; ma allora $q | p_1 \cdot p_2 \ldots p_t$ e quindi q divide $N - p_1 \cdot p_2 \ldots p_t = 1$, che è assurdo. □

4 Relazioni di congruenza

Definizione 4.1. Fissato un intero n, $(n \geq 2)$, due interi $a, b \in \mathbb{Z}$ sono detti **congruenti modulo** n se $n \mid b - a$. In questo caso si scrive

$$a \equiv b \pmod{n}.$$

Teorema 4.1. *Due interi sono congruenti modulo n se e solo se divisi per n danno lo stesso resto.*

Dimostrazione. Consideriamo gli interi a e b. Dividendoli per n, otteniamo

$$a = q_1 n + r_1, \qquad b = q_2 n + r_2, \qquad \text{con} \qquad 0 \leq r_1, r_2, < n.$$

Se $r_1 = r_2$, allora $(b - a) = (q_2 - q_1)n$, cioè a e b sono congruenti modulo n
Viceversa, se $a \equiv b \pmod{n}$, allora $b - a = kn$, da cui

$$b = a + kn = q_1 n + r_1 + kn = (q_1 + k)n + r_1.$$

Dal Teorema della Divisione il resto è univocamente determinato e quindi, confrontando $b = q_2 n + r_2$ e $b = (q_1 + k)n + r_1$, si ha che $r_1 = r_2$. □

Il teorema precedente assicura che si può scegliere come rappresentante di ogni classe il resto della divisione per n. Possiamo allora usare la notazione seguente:

$$\mathbb{Z}_n = \{\overline{0}, \overline{1}, \overline{2}, \ldots, \overline{n-1}\},$$

essendo $0, 1, 2, \ldots, n-1$ i possibili resti della divisione di un numero per n.

Esempio. Dato $n = 3$, $\mathbb{Z}_3$ è formato dalle 3 classi di congruenza seguenti:

$$\begin{aligned}
\overline{0} &= \{\ldots, -6, -3, 0, 3, 6, 9, \ldots\}, \\
\overline{1} &= \{\ldots, -5, -2, 1, 4, 7, 10, \ldots\}, \\
\overline{2} &= \{\ldots, -4, -1, 2, 5, 8, 11, \ldots\}.
\end{aligned}$$

È possibile definire delle operazioni di somma e prodotto in $\mathbb{Z}_n$.

5 Il teorema di Euler-Fermat

In questa sezione vedremo una sorprendente applicazione della teoria delle congruenze. Nel 1736 Euler diede una dimostrazione di una affermazione di Fermat (1601-1665) che riguardava la relazione fra le potenze di un numero e le sue classi di congruenza modulo un numero primo. Euler riuscì anche a dimostrare una generalizzazione di questa affermazione, valida per ogni numero intero n, e non solo per i numeri primi. Ed è proprio usando questo teorema (o meglio, una sua variante) che recentemente, Rivest, Shamir e Adleman hanno inventato un metodo di crittografia estremamente efficiente e sicuro. Il metodo RSA è uno dei più usati oggi (per esempio, è implementato nel programma di pubblico dominio PGP – Pretty Good Privacy) ed è usato da molte autorità di certificazione riconosciute in rete.

Cominciamo enunciando quello che è comunemente conosciuto come il *Piccolo Teorema di Fermat.*

Teorema 5.1 (Piccolo Teorema di Fermat)**.** *Sia p un numero primo, e sia a un intero non divisibile per p. Allora*

$$a^{p-1} \equiv 1 \mod p.$$

Dimostrazione. Consideriamo i multipli di a

$$m_1 = a, \quad m_2 = 2a, \quad m_3 = 3a, \quad \ldots, \quad m_{p-1} = (p-1)a.$$

Nessuna coppia di questi numeri interi può essere congrua modulo p: infatti, in tal caso, p sarebbe un divisore di $m_r - m_s = (r-s)a$, dove r ed s sono numeri interi $1 \leq r < s \leq p-1$, e poiché $r-s$ è minore di p ed inoltre p non divide a per ipotesi, questo non è possibile. Dunque i numeri $m_1, m_2, \ldots, m_{p-1}$ sono congrui ai numeri $1, 2, \ldots, p-1$, considerati in un ordine opportuno- Ma allora moltiplicando si ha

$$m_1 m_2 \ldots m_{p-1} = 1 \cdot 2 \cdot 3 \cdots (p-1)a^{p-1} \equiv 1 \cdot 2 \cdot 3 \cdots (p-1) \mod p,$$

e se per brevità poniamo $K = 1 \cdot 2 \cdot 3 \cdots (p-1)$ possiamo scrivere

$$K(a^{p-1} - 1) \equiv 0 \mod p.$$

Ma K non è divisibile per p, perché nessuno dei suoi fattori lo è, e dunque deve essere $(a^{p-1} - 1)$ divisibile per p, cioè

$$a^{p-1} \equiv 1 \mod p. \quad \square$$

Notiamo che se n non è un numero primo, allora non si può dire che $a^{n-1} \equiv 1 \mod n$, in generale. Per esempio, se $n = 4$ e $a = 3$, si ha $3^3 = 27 \equiv 3 \mod 4$. Ci si può dunque chiedere se ci sia una potenza di a che sia congrua ad 1 modulo n. La risposta è sì, e l'esponente corretto è stato trovato da Euler.

Definizione 5.1. Per un numero intero positivo n, indichiamo con $\varphi(n)$ il numero degli interi compresi fra 1 ed n coprimi con n.

Per esempio,

$$\begin{aligned}
\varphi(1) &= 1 && \text{perché } 1 \text{ è coprimo con } 1,\\
\varphi(2) &= 1 && 1 \text{ è coprimo con } 2,\\
\varphi(3) &= 2 && 1 \text{ e } 2 \text{ sono coprimi con } 3,\\
\varphi(4) &= 2 && 1 \text{ e } 3 \text{ sono coprimi con } 4,\\
\varphi(5) &= 4 && 1, 2, 3 \text{ e } 4 \text{ sono coprimi con } 5,\\
\varphi(6) &= 2 && 1 \text{ e } 5 \text{ sono coprimi con } 6,\\
\varphi(7) &= 6 && 1, 2, 3, 4, 5 \text{ e } 6 \text{ sono coprimi con } 7,\\
\varphi(8) &= 4 && 1, 3, 5 \text{ e } 7 \text{ sono coprimi con } 8,\\
\varphi(9) &= 6 && 1, 2, 4, 5, 7 \text{ e } 8 \text{ sono coprimi con } 9,\\
\varphi(10) &= 4 && 1, 3, 7 \text{ e } 9 \text{ sono coprimi con } 10,\\
&\dots && \dots
\end{aligned}$$

Una proprietà della funzione φ è evidente:

$$\text{se } p \text{ è un numero primo, allora } \varphi(p) = p - 1,$$

in quanto *tutti* i numeri $1, 2, \dots, p-1$ sono coprimi con p. Altrettanto semplice è calcolare $\varphi(n)$ per un numero che sia potenza di un primo. La formula è la seguente:

$$\text{se } n = p^a, \text{ con } p \text{ numero primo, allora } \varphi(n) = p^a - p^{a-1}.$$

Infatti, i numeri non coprimi con p^a sono tutti i multipli di p, e ne ne sono p^{a-1} minori di p^a. Notiamo che se $a = 1$ (cioè $n = p$ è primo) ritroviamo la stessa formula di prima. Per esempio, $\varphi(125) = \varphi(5^3) = 5^3 - 5^2 = 100$.

L'ultima proprietà della funzione φ è quella che consente di calcolare $\varphi(n)$ per un numero qualsiasi. La formula è

$$\text{se } m \text{ e } n \text{ sono coprimi, cioè } (m, n) = 1, \text{ allora } \varphi(m \cdot n) = \varphi(m) \cdot \varphi(n)$$

Per esempio, usando la tabella precedente, si ha $\varphi(21) = \varphi(3 \cdot 7) = 12$, $\varphi(30) = \varphi(5 \cdot 6) = 8$, e così via. Come vediamo, è molto semplice calcolare $\varphi(n)$ se si conosce la fattorizzazione di n. Questa è la proprietà che verrà sfruttata nell'algoritmo RSA.

Vediamo ora il Teorema di Euler-Fermat.

Teorema 5.2 (di Euler-Fermat)**.** *Sia n un intero positivo e a un intero coprimo con n. Allora*

$$a^{\varphi(n)} \equiv 1 \mod n.$$

Osserviamo che, se n è primo, allora $\varphi(n) = n - 1$ e quindi ritroviamo il Teorema di Fermat.

Ci si può anche chiedere cosa capiti se $(a, n) \neq 1$. Per esempio, se $n = 6$,$a = 2$, si ha

$$\begin{aligned} 2^2 &\equiv 4 \mod 6 \\ 2^3 &\equiv 2 \mod 6 \\ 2^4 &\equiv 2 \mod 6 \end{aligned}$$

e così via, e vediamo che le potenze di 2 modulo 6 alternano fra 2 e 4 e non si ottiene mai 1. Osserviamo però che il teorema di Euler-Fermat si può enunciare in modo equivalente come

Teorema 5.3 (di Euler-Fermat)**.** *Sia n un intero positivo ed a un intero coprimo con n. Allora*

$$a^{\varphi(n)+1} \equiv a \mod n.$$

In questa forma la conclusione del teorema vale anche per $n = 6$, $a = 2$. Non è però possibile eliminare del tutto l'ipotesi che a ed n siano coprimi come mostra l'esempio $n = 12$, $a = 2$: si ha

$$\begin{aligned} 2^2 &\equiv 4 \mod 12 \\ 2^3 &\equiv 8 \mod 12 \\ 2^4 &\equiv 4 \mod 12 \\ 2^5 &\equiv 8 \mod 12 \end{aligned}$$

e tutte le potenze successive alternano fra 4 e 8, e dunque non si riottiene più 2. Se però facciamo un'ipotesi su n, è possibile eliminare la condizione che a ed n siano coprimi.

Definizione 5.2. Un intero n si dice **libero da quadrati** se è il prodotto di numeri primi distinti, cioè se non è divisibile per nessun quadrato > 1.

Per esempio, $6 = 2 \cdot 3$ è libero da quadrati, mentre $12 = 2 \cdot 2 \cdot 3$ non lo è. Con questa definizione si può enunciare il seguente teorema:

Teorema 5.4. *Se $n = p_1 \cdot p_2, \cdots p_k$ è libero da quadrati, allora*

$$a^{\varphi(n)+1} \equiv a \mod n$$

per **ogni** *intero a.*

Osserviamo che la conclusione è la stessa del Teorema di Euler-Fermat, ma l'ipotesi $(a, n) = 1$ è stata sostituita dall'ipotesi n libero da quadrati. Usando il Teorema 5.4 si può dimostrare facilmente per induzione che vale il seguente risultato più generale:

Definizione 5.3. Se $n = p_1 \cdot p_2, \cdots p_k$ è libero da quadrati allora

$$a^{h\varphi(n)+1} \equiv a \mod n$$

per ***ogni*** intero a e per ***ogni*** intero $h \geq 1$.

6 I sistemi di crittografia

Il problema della crittografia è semplice da enunciare: ci sono due persone, il *mittente* ed il *destinatario*[1], che chiameremo $\mathcal{A}$ e $\mathcal{B}$, che vogliono comunicare fra loro senza che nessun altro possa leggere i loro messaggi. Dato un messaggio M, che possiamo immaginare essere una sequenza di simboli o numeri, $\mathcal{A}$ usa una funzione f (ovvero un algoritmo) che trasforma M in un'altra sequenza C e invia questa sequenza. $\mathcal{B}$ usa un'altra funzione g che trasforma C di nuovo nel messaggio originale M. Naturalmente, se qualcuno intercetta C non deve essere in grado di usare la funzione g per decodificare il messaggio.

L'idea di *sistema crittografico* può dunque essere resa mediante uno schema del tipo:

$$M \xrightarrow{f} C \xrightarrow{g} M$$

dove M è l'insieme dei *messaggi*, C quello dei messaggi *cifrati* ed f e g sono funzioni tali che

$$f \circ g = \iota_M$$

(quindi è necessario che f sia iniettiva). f è la procedura di *cifratura* e g quella di *decifratura*.

Per ottenere ciò, di solito si usa una **chiave** (per esempio una password). Dunque la funzione che codifica f non ha come argomento solo il messaggio M, ma anche un altro parametro, la chiave k. In simboli

$$f(M, k) = C$$

e la funzione g tale che

$$g(C, k) = M$$

[1] In letteratura spesso chiamati Alice e Bob.

cosicché, conoscendo la chiave, si possa decrittare il messaggio. Questo schema, che usa la stessa chiave k sia per la codifica che per la decodifica, si chiama **simmetrico**.

In crittografia si suppone che tutti siano a conoscenza dell'algoritmo che usano f e g, perché è sempre possibile avere questa informazione mediante attività di spionaggio o simili. Basare la segretezza delle comunicazioni sulla segretezza dell'algoritmo non è una buona idea. Invece la segretezza è basata sul fatto che solo il $\mathcal{A}$ e $\mathcal{B}$ conoscono la chiave. Infatti, l'algoritmo viene usato sempre, mentre la chiave può essere cambiata ogni volta. In questo modo, anche se qualcuno "ruba" una chiave, la può usare solo poche volte.

Questa situazione (cambiare spesso la chiave) richiede però un modo di comunicare sicuro fra $\mathcal{A}$ e $\mathcal{B}$. Un altro problema è il seguente: se vi è un solo ricevente, ma molti mittenti, tutti i mittenti conoscono la chiave, e possono leggere i messaggi l'uno dell'altro, cosa che un mittente certamente non vuole. Questo è il problema che il metodo a chiave pubblica vuole risolvere. Lo schema è il seguente:

> $\mathcal{B}$ comunica pubblicamente una chiave e. $\mathcal{A}$ codifica il messaggio usando la funzione $f = (M, e) = C$ e trasmette il messaggio. $\mathcal{B}$ possiede un'altra chiave d privata. $\mathcal{B}$ decodifica il messaggio usando $g = (C, d) = M$.

Questo elimina il problema della distribuzione delle chiavi, e anche la chiave che serve per codificare non è quella che decodifica, e quindi solo $\mathcal{B}$ può leggere i messaggi. Questo schema è stato proposto per la prima volta da Whitfield Diffie e Martin Hellman nel 1976. Uno schema differente di crittografia a chiave pubblica è stato proposto nello stesso anno da Ralph Merkle.

Per fare un'analogia, un metodo simmetrico è come una cassaforte, e la chiave è la combinazione per aprirla. Una persona che ha la chiave può aprire la cassaforte e depositare un documento (codificare un messaggio). Un'altra persona con la chiave può aprire la cassaforte e prendere il documento (decodificare il messaggio). Chi non ha la chiave (l'avversario, la spia, ecc.) deve scassinare la cassaforte. Un metodo a chiave pubblica invece è come una buca delle lettere. Chiunque può infilare una lettera nella buca (codificare il messaggio con la chiave pubblica), ma solo il proprietario della buca ha la chiave per aprirla facilmente (decodificare il messaggio). Tutti gli altri devono scassinare la buca delle lettere, e questo è possibile, e anche più facile che scassinare una cassaforte, ma richiede tempo. Se il padrone della buca ritira spesso la sua posta oppure se cambia spesso la serratura, la sua posta è al sicuro.

Naturalmente, la domanda è: funzioni f e g con le proprietà richieste da questo schema esistono? Si può pensare che se si conosce la chiave e usata per codificare il messaggio, e la funzione f che effettua la codifica, deve essere possibile in qualche modo decodificare il messaggio, e cioè un simile schema non può funzionare. L'intuizione di Rivest, Shamir e Adleman (1978) è stata che è possibile costruire funzioni f e g tali che:

1. mediante la chiave privata d la decodifica è semplice e rapida, e
2. usando la conoscenza della funzione f e della chiave pubblica e, la decodifica è possibile ma richiede troppo tempo per essere utile.

Per esempio, troppo tempo può voler dire che eseguire tutti i calcoli necessari, anche usando i computer più veloci a disposizione, richiede alcuni anni. È chiaro che leggere un messaggio segreto alcuni anni dopo che è stato scritto non è di grande utilità.

7 L'algoritmo RSA

Descriviamo ora l'algoritmo RSA che, fra tutti gli algoritmi proposti per la crittografia a chiave pubblica che si sono rivelati sicuri, è il più semplice sia da capire che da implementare.

$\mathcal{B}$ sceglie due numeri primi p e q, molto grandi, li moltiplica e forma il numero $n = pq$, che è libero da quadrati. Sappiamo che $\varphi(n) = (p-1)(q-1)$. Ora sceglie un numero e che sia coprimo con $\varphi(n)$. La chiave pubblica per codificare i messaggi è la coppia (n, e). La funzione f per la codifica è la seguente:

- si trasforma il messaggio in un numero M, minore di n, Se il messaggio è molto lungo, lo si spezza in blocchi più piccoli $M_1, \ldots, M_k$;
- si calcola $C_i = M_i^e \mod n$ per ogni i;
- si trasmette $C_1, \ldots, C_k$.

Per decodificare, si procede nel modo seguente: consideriamo l'equazione

$$ex \equiv 1 \mod \varphi(n).$$

Poiché e è coprimo con $\varphi(n)$, questa equazione ha una soluzione d, per cui vale $ed = 1 + h\varphi(n)$. Allora la funzione g di decodifica è:

- si calcola $M_i = C_i^d \mod n$.

Infatti

$$C_i^d = (M_i^e)^d = M_i^{ed} = M_i^{1+h\varphi(n)} \equiv M_i \mod n = M_i$$

dove la congruenza è vera in virtù della definizione 5.3 e l'ultima eguaglianza vale in quanto $M_i < n$.

Facciamo qualche esempio semplice, tralasciando i calcoli. Se prendiamo $p = 47$ e $q = 71$, allora

$$n = 47 \cdot 71 = 3337$$

La chiave pubblica e deve essere coprima con

$$\varphi(3337) = (p-1)(q-1) = 46 \cdot 79 = 3220$$

Scegliamo a caso un numero e coprimo con 3220, ad esempio $e = 79$, che essendo primo è sicuramente coprimo con 3220. In questo caso, calcolando con l'algoritmo di Euclide si ottiene

$$\begin{aligned} \mathbf{3220} &= 40 \cdot \mathbf{79} + \mathbf{60} \\ \mathbf{79} &= 1 \cdot \mathbf{60} + \mathbf{19} \\ \mathbf{60} &= 3 \cdot \mathbf{19} + \mathbf{3} \\ \mathbf{19} &= 6 \cdot \mathbf{3} + \mathbf{1} \end{aligned}$$

Calcolando con i resti si ottiene

$$(79, 3220) = 1 = 1019 \cdot 79 - 25 \cdot 3220$$

e dunque abbiamo la chiave privata $d = 1019$. Rendiamo pubblici n ed e, teniamo segreto d ed eliminiamo p e q.

Per codificare il messaggio

$$M = 6882326879666683$$

per prima cosa lo spezziamo in blocchi più piccoli. Poiché $n = 3337$ ha quattro cifre, blocchi di tre cifre vanno bene. Dunque il messaggio si spezza in

$$\begin{aligned} M_1 &= 688 \\ M_2 &= 232 \\ M_3 &= 687 \\ M_4 &= 966 \\ M_5 &= 668 \\ M_6 &= 003 \end{aligned}$$

Il primo blocco è codificato da

$$688^{79} \mod 3337 = 1570 = C_1$$

Notiamo che non bisogna elevare 688 alla 79-esima potenza (numero molto grande) e poi ridurre modulo 3337. Invece, si calcola 688^2 e si riduce modulo 3337, poi si moltiplica di nuovo per 688 e si riduce nuovamente modulo 3337 e così via, e in ogni passo i numeri sono piccoli. Ci sono metodi per calcolare queste potenze in modo ancora più veloce di quello descritto, rendendo così la codifica del messaggio veloce.

Ripetendo gli stessi passi su tutti gli altri blocchi otteniamo il messaggio codificato

$$C = 1570\,2756\,2091\,2276\,2423\,158$$

Decodificare il messaggio richiede le stesse operazioni usando l'esponente $d = 1019$ sui blocchi di C. Si ha

$$1570^{1019} \mod 3337 = 688 = M_1$$

e in modo simile per i blocchi successivi.

Perché questo metodo è sicuro? In fondo, per ottenere la chiave privata d basta conoscere $\varphi(n)$ e tutti conoscono n. Il modo più rapido di calcolare $\varphi(n)$ è avere la decomposizione $n = pq$, cioè dobbiamo fattorizzare n. Negli anni recenti sono stati sviluppati molti metodi di fattorizzazione, ed il migliore di quelli attualmente noti si chiama "General Number Field Sieve", abbreviato in GFNS. La matematica che sta dietro questi algoritmi è molto complicata, e non possiamo in queste note dire niente al riguardo.

Piuttosto, è interessante vedere alcuni risultati concreti. I dati che citeremo sono in continua evoluzione, perciò possono diventare obsoleti anche molto presto. Un aspetto interessante del problema della fattorizzazione è costituito dai cosiddetti *RSA Challanges*, le "sfide RSA", che consistono in una serie di numeri da fattorizzare. Iniziata negli anni '90, questa sfida è ufficialmente terminata nel 2007.[2] L'ultimo nuemero fattorizzato (12 dicembre 2009) è noto come RSA 768, cioè è un numero che in rappresentazione

[2] Un archivio è disponibile presso `http://www.emc.com/emc-plus/rsa-labs/historical/the-rsa-challenge-numbers.htm`

binaria ha 768 cifre, in notazione decimale ha 232 cifre. Il numero è:

123018668453011775513049495838496272077285
356959533479219732245215172640050726365751
874520219978646938995647494277406384592519
255732630345373154826850791702612214291346
167042921431160222124047927473779408066535
141959745985690214341 3

e i suoi due fattori primi sono

334780716989568987860441698482126908177047
949837137685689124313889828837938780022876
147116525317430877378144679994 89

e

367460436667995904282446337996279526322791 5
816434308764267603228381573966651127923337 3
417143396810270092798736308917

Il tempo necessario per fattorizzare questi numeri è notevole. L'algoritmo GNFS è composto da due parti, una preliminare per cercare una serie di equazioni lineari da risolvere (*sieving step*) e una seconda in cui si risolve un sistema di equazioni (*matrix step*). La prima può essere eseguita in parallelo su più calcolatori, mentre la seconda deve essere fatta tutta insieme.

La fattorizzazione di RSA 768 ha richiesto l'equivalente di circa 1500 anni di lavoro di un processore AMD Opteron a 2.2 Ghz con 2GB di RAM per il primo passo, mentre la risoluzione del sistema ha richiesto sei mesi di lavoro ad un cluster di 80 processori AMD Opteron a 2.2 Ghz. In termini di tempo reale, i calcoli per la fattorizzazione sono durati un paio di anni.[3] In confronto RSA 663, un po' più piccolo, ha richiesto circa la metà del tempo. Si stima che fattorizzare RSA 1024, con 1024 cifre binarie (309 cifre decimali) richieda circa 1000 volte il tempo occorso per fattorizzare RSA 768.

Cosa possiamo concludere: chiavi a 1024 bit sono abbastanza sicure oggi e per ancora (forse) molti anni, mentre chiavi a 640 bits sono già insicure.

[3] `http://eprint.iacr.org/2010/006.pdf`

Si potrebbe osservare: non è necessario trovare i fattori p e q, ma solo calcolare $\varphi(n)$. In questo caso però, se conosciamo p e q sappiamo calcolare $\varphi(n)$ semplicemente, bastano due sottrazioni e una moltiplicazione:

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

Viceversa, conoscendo n e $\varphi(n)$ è facile trovare p e q, basta risolvere una equazione di secondo grado. Ponendo $B = pq$ e $A = p + q$ si ha che l'equazione

$$x^2 - Ax + B = 0$$

ha come soluzioni esattamente p e q. Quindi, conoscendo $\varphi(n)$ si ha un algoritmo che in pochi passi, una somma e la risoluzione di un'equazione di secondo grado, consente di fattorizzare il numero n. Dunque la complessità di calcolo per determinare $\varphi(n)$ è equivalente alla complessità di fattorizzare il numero n che, come abbiamo visto, è piuttosto grande.

Questo spiega perché il metodo è sicuro. È però importante osservare che nessuno ha mai dimostrato che fattorizzare il numero n è veramente così difficile, ma solo che i migliori algoritmi conosciuti sono troppo lenti. Potrebbe capitare che un giorno qualcuno scopra un metodo più efficiente per fattorizzare i numeri, ed allora il metodo RSA non sarebbe più sicuro.

La crittografia è un argomento affascinante, ed è un campo in cui l'avvento dei computer ha portato grandi novità e possibilità di utilizzo. È bene osservare che non sempre il mittente ed il ricevente sono esseri umani, ma potrebbero essere entrambi computer che devono comunicare fra loro, per esempio un telefono cellulare che comunica con la rete di telefonia (non vogliamo che un telefono non autorizzato possa mettersi in contatto ed usare le risorse della rete oppure che qualcuno possa intercettare la trasmissione e ascoltare le nostre telefonate), oppure un computer che manda una richiesta ad un server, per esempio per fare una stampa (non vogliamo che tutti possano usare le nostre stampanti) e così via. Quindi è importante avere dei metodi semplici per trasmettere in sicurezza.

La sicurezza di un sistema però non è solo nella matematica che si usa, ma anche nel protocollo. Possiamo avere il sistema più sicuro del mondo, ma se riveliamo la chiave privata ad un estraneo la nostra sicurezza è irrimediabilmente compromessa.