Matematica e crittografia

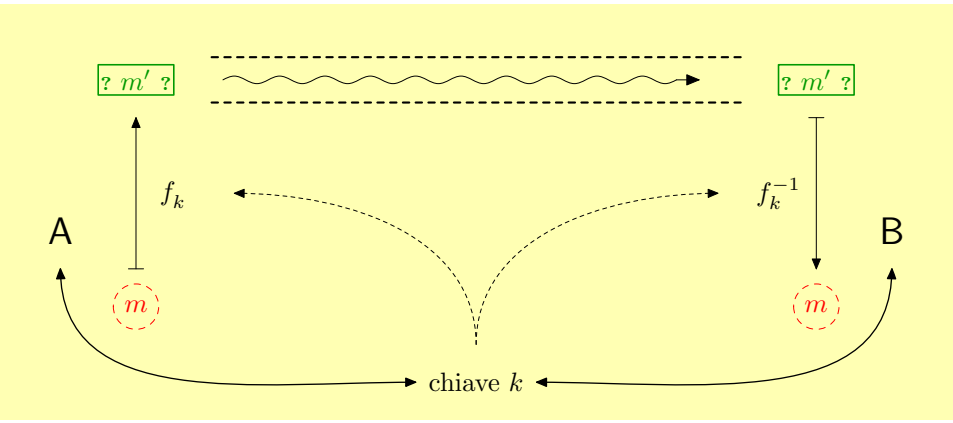
Giovanni Cutolo

L'esigenza di scambiare messaggi privati, incomprensibili per un estraneo non autorizzato che in un modo o nell'altro ne venga in possesso, sembra essere antica quanto la scrittura. Esempi di testi volutamente *cifrati*, cioè trasformati in modo da non essere intellegibili se non dopo un'operazione di *decifrazione* (la trasformazione inversa, dal testo cifrato al testo, come si dice, 'in chiaro'), sono giunti a noi dalle più antiche civiltà dotate di lingua scritta. La crittografia può essere definita come la disciplina che studia le diverse possibili tecniche di cifratura, allo scopo di valutarne l'effettiva sicurezza e di introdurne di più efficaci.

Se per secoli, e forse tuttora nelle percezione di molti, la pratica della crittografia è stata strettamente associata ad aspetti lontani dalla vita ordinaria, come lo spionaggio (militare, diplomatico, industriale), al giorno d'oggi, nell'era della telematica, ciascuno di noi ne fa uso quotidianamente e probabilmente inconsapevolmente ogni volta che utilizza servizi come il bancomat, la posta elettronica o il commercio on-line, o magari quando assiste a programmi televisivi a pagamento.

Diversi matematici di diverse epoche si sono cimentati sul versante della crittoanalisi (cioè nel trovare metodi per decifrare messaggi pur senza conoscere, almeno in partenza, le chiavi di decifrazione): è famoso il caso della violazione della cifratura Enigma, il sistema crittografico usato dall'esercito nazista durante la II guerra mondiale, dovuta, in una prima fase, al matematico polacco Marian Rejewski e completata poi da un gruppo di scienziati inglesi, tra i quali ruolo prominente ebbe il celeberrimo logico Alan Turing. È però a partire dagli anni '70 del novecento che l'utilizzo sistematico di idee matematiche, spesso nate in ambiti del tutto indipendenti, ha rivoluzionato teoria e tecnica della crittografia, con l'introduzione della crittografia a chiave pubblica e di altri simili protocolli.

I metodi crittografici più tradizionali, quelli a chiave privata, gli unici in uso sino a tempi molto recenti, (vedi schema) si possono descrivere come il proteggere un messaggio chiudendolo in un cofanetto munito di una serratura; il mittente ed il destinatario (e nessun altro) sono in possesso di



Crittografia a chiave privata: A e B concordano una chiave segreta comune, k. Questa chiave serve per descrivere una funzione di cifratura f_k e la sua inversa, la funzione di decifrazione f_k^{-1}. Per inviare un messaggio m a B, A provvede innanzitutto a trasformare m nel messaggio cifrato m', tramite la funzione f_k, e trasmette poi m' a B, anche per un canale non sicuro (quindi si ammette la possibilità che m' venga letto da terzi). B provvederà infine a decifrare m', ritrasformandolo in m grazie alla funzione f_k^{-1}.

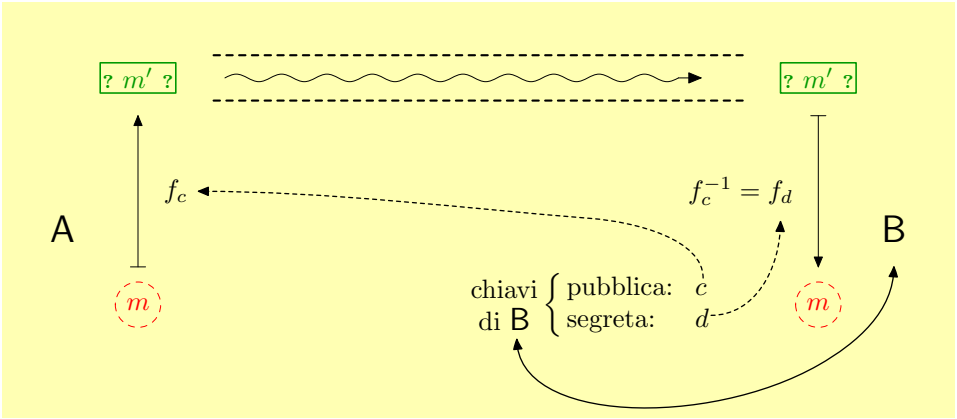
una chiave che permette di chiudere (cifrare) ed aprire (decifrare) il cofanetto. Uno dei punti deboli di questo tipo di crittografia è che in una certa fase iniziale, mittente e destinatario hanno dovuto scambiarsi la chiave (segreta) di cifratura, cosa che non è sempre possibile realizzare in modo del tutto sicuro. Inoltre, in un sistema con molti utenti, ciascuno dei quali voglia poter comunicare in modo riservato con ciascun altro (cosa spesso necessaria in tante moderne applicazioni), è necessario un altissimo numero di chiavi: il calcolo combinatorio mostra che servono $t(t-1)/2$ chiavi distinte per un sistema con t utenti (ad esempio, quasi due milioni di chiavi per 2000 utenti). Produrre e gestire tante chiavi può creare grandi, talvolta insuperabili, difficoltà pratiche.

Per risolvere questo genere di problemi si è fatto ricorso a due strategie. La prima consiste nel sostituire lo scambio della chiave con una procedura in cui una chiave, magari di uso temporaneo, possa essere concordata pur senza mai essere trasmessa. La seconda strategia, quella della crittografia a chiave pubblica, permette, come vedremo, di eliminare il problema alla radice, facendo del tutto a meno di chiavi segrete condivise. La maggior parte degli attuali protocolli crittografici fa infine ricorso a una strategia ibrida: viene usata la crittografia a chiave pubblica per effettuare la trasmissione di una chiave 'usa e getta', che viene poi utilizzata per trasmettere il messaggio con un metodo di crittografia a chiave privata. Questo perché, generalmente, i protocolli crittografici a chiave privata sono più veloci e richiedono meno risorse di calcolo dei protocolli a chiave pubblica.

1 La crittografia a chiave pubblica

Se abbiamo paragonato la crittografia a chiave privata ad una comune serratura, la crittografia a chiave pubblica funziona invece come un lucchetto a scatto: chiunque lo può chiudere, ma solo il proprietario, in possesso della chiave, lo può aprire. Il metodo si può schematizzare come nella figura in alto: ogni utente del sistema crittografico ha una personale coppia di chiavi, una pubblica ed una segreta; un mittente A che voglia inviare un messaggio riservato a B provvederà innanzitutto a cifrarlo, utilizzando a questo scopo la chiave pubblica di B, il quale B potrà poi usare la sua chiave segreta per decifrarlo. Affinché il sistema possa funzionare, è necessario che chiunque sia in grado di effettuare la cifratura, ma che la decifrazione sia praticamente impossibile da eseguirsi a meno di non conoscere la chiave segreta di B.

Questa frase nasconde un problema matematico tutt'altro che banale. In forma un po' semplificata si tratta di questo: per realizzare un sistema del genere bisogna trovare una funzione invertibile f (da usare per la cifratura e descritta dalla chiave pubblica, f_c nello schema) tale che, dato un qualunque messaggio m, sia semplice il calcolo di $m' = f(m)$ (il messaggio cifrato) ma, viceversa, qualunque algoritmo che permetta di ricavare m da m' senza disporre di una informazione extra (la chiave segreta) richieda troppe risorse e troppo tempo di esecuzione per essere effettivamente utilizzabile. È proprio in questo che consiste la sottigliezza, infatti, poiché f è nota è certamente nota anche la funzione inversa f^{-1} (f_d nello schema), quindi un ficcanaso che volesse



Crittografia a chiave pubblica

decifrare il messaggio senza autorizzazione si troverebbe nella situazione di conoscere (in astratto) la funzione necessaria, ma ciononostante non saperne calcolare i valori!

Il primo sistema di crittografia ad essere apparso, e tuttora uno dei più utilizzati, è il crittosistema RSA (dai nomi, Rivest, Shamir e Adleman, di coloro che lo proposero nel 1977), descritto in dettaglio più avanti. Esso è basato sul fatto che non sono noti efficienti metodi che permettano di calcolare i fattori primi di un numero molto grande, quindi mentre, assegnati due numeri primi p e q, non presenta alcuna difficoltà il calcolo del prodotto $n = pq$, il calcolo inverso—ricavare i fattori p e q da n—può essere di estrema difficoltà, praticamente impossibile anche per i più veloci computer attualmente a disposizione se i numeri coinvolti sono molto grandi (e accortamente scelti).

Esistono diversi crittosistemi a chiave pubblica, quasi tutti basati sulla matematica del mondo discreto: strutture algebriche (cioè 'ambienti di calcolo') o geometriche costruite su insiemi finiti. Ad esempio, l'ambiente di calcolo per RSA è quello dell'aritmetica modulare, di cui si parlerà più avanti, ma sono molto usati anche sistemi in cui l'ambiente è costituito da particolari curve, chiamate curve ellittiche, costruite in piani che, a differenza di quelli della tradizionale geometria euclidea, hanno solo un numero finito di punti. Le curve ellittiche, in queste ed in altre geometrie, sono tra gli oggetti più ubiqui della matematica contemporanea; tra l'altro, hanno avuto un importantissimo ruolo nella recente (1994) dimostrazione dell'ultimo teorema di Fermat.

P vs. NP

Il tipo di difficoltà incontrata per la definizione di un crittosistema a chiave pubblica è connesso con uno dei più famosi problemi irrisolti della matematica contemporanea, il cosiddetto problema 'P vs. NP'. Per dirla nel modo più semplice possibile, il problema è: *esiste una domanda alla quale, in generale, una risposta esista ma sia estremamente difficile da calcolare, mentre sia invece sempre facile verificare se una risposta proposta è o meno corretta?* Un possibile esempio potrebbe essere quello della fattorizzazione in primi dei numeri interi: verificare se un numero intero n sia o non sia il prodotto di alcuni, assegnati, numeri primi è compito agevole (basta eseguire il prodotto di questi primi e confrontare il risultato con n), trovare la fattorizzazione in primi di n invece non sembra affatto essere un problema banale, nel senso che i metodi noti richiedono calcoli la cui lunghezza può crescere enormemente al crescere di n (il che non esclude che esistano dei metodi di fattorizzazione molto più efficienti, ancora in attesa di essere scoperti). Un importante risultato degli ultimi anni (2002) è stata la dimostrazione del fatto che un caso particolare del problema della fattorizzazione, quello di decidere se un assegnato numero intero n sia o non sia primo, rientra tra quelli definiti come 'facili' nella formulazione precisa del problema *P vs. NP*.
Un'altra domanda che potrebbe essere del tipo richiesto nella formulazione del problema *P vs. NP* è quella nota come *problema della zaino*: dati un numero intero n ed una lista finita di interi, è possibile selezionare dalla lista alcuni termini che abbiano come somma n? (una versione tridimensionale del problema rende ragione del nome: avendo un certo numero di oggetti a disposizione, è possibile riempire completamente un assegnato contenitore con alcuni di essi?). È stato dimostrato che chi riuscisse a calcolare il livello di difficoltà del problema dello zaino risolverebbe anche il problema *P vs. NP*, nel senso che se una qualche domanda del tipo richiesto esiste, allora il problema dello zaino è una di esse. Dunque, se il problema dello zaino è 'difficile' allora *P vs. NP* ha risposta positiva, se esso è 'facile' *P vs. NP* ha risposta certamente negativa!
P vs. NP fa parte di una lista di sette problemi ritenuti di grande importanza per lo sviluppo della matematica nel XXI secolo, per la risoluzione di ciascuno dei quali il Clay Institute (`http://www.claymath.org/millennium/`) ha messo in palio la bella cifra di un milione di dollari.
Va infine detto che l'analogia tra il problema *P vs. NP* e quelli legati alla crittografia a chiave pubblica ha precisi limiti. In particolare, non coincidono nei due contesti le nozioni che qui abbiamo semplificato con gli aggettivi 'facile' e 'difficile'.

Molta ricerca matematica è attualmente impegnata nel cercare di dimostrare (o smentire) la sicurezza di crittosistemi a chiave pubblica (non è dimostrato che, ad esempio, RSA sia inviolabile),

nel renderli più efficienti, nell'inventarne di totalmente nuovi. Non è solo la matematica ad intervenire in questi sviluppi: sono stati ad esempio proposti e sperimentati con successo crittosistemi basati sulla meccanica quantistica, la cui sicurezza dipende in ultima analisi dal principio di indeterminazione di Heisenberg, quindi dall'intrinseca imprevedibilità dell'esito di singoli esperimenti nella fisica delle particelle.

2 Aritmetica modulare

L'aritmetica modulare (chiamata talvolta aritmetica dell'orologio) si può descrivere come un'aritmetica in cui i calcoli (addizioni, sottrazioni, moltiplicazioni) si eseguono 'a meno di' multipli di un prefissato intero positivo n. Ad esempio l'aritmetica modulo 2 è quella in cui risultano identificati tra loro tutti i numeri che differiscano per un numero pari (quindi tutti i pari da una parte, tutti i dispari dall'altra), dunque una aritmetica in cui l'unica distinzione possibile è quella tra 'numero pari' e 'numero dispari', ma non, per esempio, quella tra due particolari numeri dispari, come 5 e 41.

La nozione chiave alla base dell'aritmetica modulare è quella di congruenza modulo un assegnato intero positivo n. Due interi a e b si dicono congrui modulo n (in simboli, $a \equiv_n b$) se e solo se $a - b$ è multiplo di n. Si ha che ogni intero è congruo (modulo n) ad esattamente un intero compreso tra 0 e $n - 1$, precisamente al suo resto nella divisione per n. Ciò che rende particolarmente utili le congruenze è che esse 'rispettano' le consuete operazioni tra i numeri interi; infatti se $a \equiv_n b$ e $c \equiv_n d$ allora $a + c \equiv_n b + d$ e $ac \equiv_n bd$, qualsiasi siano gli interi a, b, c e d coinvolti.

Supponiamo, dunque, di dover eseguire dei calcoli tra numeri interi, ma di essere interessati a conoscere solo il resto del risultato nella divisione per n; allora potremo metodicamente sostituire gli operandi con numeri che siano congrui ad essi modulo n senza che questo resto cambi. È proprio su questo trucco che sono basati i criteri di divisibilità che vengono insegnati sin dalla scuola elementare, o anche la cosiddetta 'prova del nove'.

Ad un livello appena maggiore di astrazione (e di chiarezza), tutto ciò si formalizza in questi termini: le proprietà menzionate sopra permettono di definire correttamente operazioni tra *classi di resto* (la classe di resto di a modulo n è l'insieme di tutti gli interi congrui ad a modulo n): la somma e il prodotto tra la classe di a e quella di b sono le classi di resto di $a+b$ e di ab. In questo modo definiamo una aritmetica tra le classi di resto modulo n (che sono in tutto n), è questa l'aritmetica modulo n. Ad esempio, se $n = 2$, come accennato sopra le classi di resto sono due: l'insieme P dei numeri pari, e l'insieme D dei numeri dispari, e valgono (ovvie) regole di calcolo come $P + P = P = D + D = PD$ o $P + D = DD = D$.

Alcuni calcoli sono notevolmente semplificati in aritmetica modulare rispetto all'aritmetica ordinaria, ciò è particolamente vero per il calcolo delle potenze; ad esempio, poiché $4^2 \equiv_{13} 3$ e quindi $4^3 \equiv_{13} 4 \cdot 4^2 \equiv_{13} 4 \cdot 3 \equiv_{13} 12 \equiv_{13} -1$, si ha $4^{10000} \equiv_{13} 4(4^3)^{3333} \equiv_{13} 4(-1)^{3333} \equiv_{13} -4 \equiv_{13} 9$. Per contro, i logaritmi in aritmetica modulare sono spesso molto difficili da calcolare (ad esempio, per arrivare a stabilire che, modulo 4567 il logaritmo in base 10 di 2 è 2916, nel senso che $10^{2916} \equiv_{4567} 2$, non si conoscono metodi essenzialmente migliori che quello di procedere per tentativi, calcolando varie potenze di 10 sino a trovare quella giusta). Per questo motivo le funzioni esponenziali in aritmetica modulare (facili da calcolare, ma con inversa 'difficile') sono utilizzate spessissimo nei protocolli crittografici.

3 Descrizione del protocollo RSA

Come per ogni crittosistema a chiave pubblica, descrivere RSA significa spiegare in che modo ogni utente effettua la scelta delle sue chiavi (pubblica e segreta) e in che modo queste chiavi permettano la cifratura e la decifrazione.

Per definire le sue chiavi, l'utente B sceglie due primi distinti (molto grandi) p e q e ne calcola il prodotto n; sceglie poi un intero positivo c che sia minore di n e coprimo con $\varphi := (p-1)(q-1)$ (due numeri interi sono *coprimi* se non hanno divisori comuni maggiori di 1). La coppia (n, c) costituisce la chiave pubblica di B. Utilizzando un vecchio e notissimo algoritmo che nella sue

essenza risale ad Euclide, B può infine calcolare un intero d tale che $cd \equiv_\varphi 1$. A questo punto egli può cancellare ogni traccia di p, q e φ e conservare (gelosamente) d come sua chiave segreta.

La cifratura di un messaggio per B avviene come segue: se, come quasi sempre accade, il messaggio è in forma digitale, allora esso è già in qualche modo rappresentato come un numero, altrimenti esso viene trasformato in un numero intero positivo con una qualsiasi procedura di codifica (del tutto trasparente e magari anche universalmente nota, la cosa non ha alcuna importanza). Serve anche che il numero m che rappresenta il messaggio sia minore di n; se ciò non è possibile perché il messaggio è troppo lungo, quest'ultimo andrà preliminarmente suddiviso in blocchi da trasmettere uno per volta. Il mittente A calcola il resto m' di m^c nella divisione per n (come abbiamo visto, questi calcoli sono facili in aritmetica modulare; ricordiamo anche che A, come chiunque, ha a disposizione la chiave pubblica (n, c) di B); questo sarà il messaggio cifrato.

Il teorema di Fermat-Eulero

Il protocollo RSA è basato sul seguente teorema di aritmetica modulare, dovuto in un caso particolare a Pierre de Fermat ed esteso poi da Leonhard Euler (due grandissimi matematici dei secoli XVII e XVIII, francese il primo, svizzero il secondo). Per ogni intero positivo n si indica con $\varphi(n)$ il numero degli interi compresi tra 1 e n e coprimi con n. Il teorema di Eulero si enuncia abitualmente dicendo che, *per ogni intero a, se a e n sono coprimi allora $a^{\varphi(n)} \equiv_n 1$.* RSA utilizza una variante di questo teorema: *se n non è divisibile per il quadrato di nessun numero intero maggiore di 1, allora $a^{1+k\varphi(n)} \equiv_n a$ per ogni intero a ed ogni intero non negativo k.*

La decifrazione avviene in modo simile: B calcola il resto di $(m')^d$ modulo n, questo resto sarà proprio m. Il motivo risiede nel teorema di Fermat e Eulero discusso nel box e nel fatto che risulta $\varphi = \varphi(n)$ (cosa che qui non giustifichiamo): si ha $(m')^d \equiv_n m^{cd}$ e $cd \equiv_\varphi 1$, dunque $(m')^d \equiv_n m$. Poiché $0 \leq m < n$ ciò basta per provare la nostra affermazione.

Su cosa si basa la sicurezza di questo sistema? Una persona, diversa da B, che sapesse fattorizzare n saprebbe anche calcolare $\varphi = (p-1)(q-1)$, quindi ricavare d da (n, c) e decifrare il messaggio. Il punto è che, se i primi p e q sono scelti bene, fattorizzare n è (meglio: si ritiene che sia) estremamente complesso. Si può anche (viceversa) dimostrare che il problema di ricavare d da (n, c) ha lo stesso grado di complessità di quello di fattorizzare n. Dunque ciò che salvaguarda la segretezza della chiave d è proprio l'incapacità, da parte di un eventuale spione, di decomporre n in fattori primi. Questo mostra che la sicurezza di RSA verrebbe messa in crisi se fossero scoperti

Un esempio di cifratura RSA

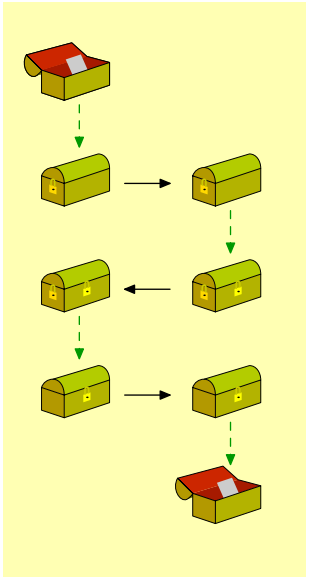
Supponiamo che B abbia deciso di vendere libri in rete, usando RSA per ricevere informazioni bancarie cifrate dai clienti. Il primo passo sarà quello di definire le proprie chiavi. B sceglie due numeri primi, nel nostro esempio $p = 137562311$ e $q = 289877297$, calcola $n = pq = 39876190881753367$ e $\varphi = (p-1)(q-1) = 39876190454313760$; completa poi la sua chiave pubblica con il numero $c = 87644617$, che è coprimo con φ, e rende questa chiave (n, c) nota. Infine B calcola la sua chiave segreta trovando un intero d tale che $cd \equiv_\varphi 1$, calcolo non difficile che non mostriamo qui e fornisce $d = 330767495513$.
Il giorno in cui A voglia acquistare un libro da B, dovrà trasmettergli il numero della sua carta di credito: 2876 5780 3204 2395. Allora A calcolerà $2876578032042395^{87644617}$ modulo n, ottenendo 35510332034374035, il numero (cifrato) che invierà a B. Ricevuto questo numero, B calcolerà $35510332034374035^{330767495513}$ modulo n, ricavando così 2876578032042395, il numero della carta di credito di A!
Va da sé che, nella realtà, il commercio elettronico (che costituisce una delle più delicate applicazioni della crittografia a chiave pubblica) utilizza procedure più sofisticate di questa appena schematizzata. Inoltre, i numeri qui utilizzati per la definizione delle chiavi sono troppo piccoli per garantire alcuna sicurezza: n è ancora molto facile da fattorizzare per un computer. I sistemi per le comunicazioni bancarie utilizzano numeri ben più grossi, di oltre trecento cifre decimali.

nuovi, veloci, metodi di fattorizzazione. Inoltre, tutto ciò non esclude che esista qualche metodo per violare la sicurezza di RSA che prescinda dalla fattorizzazione di n; al momento, per quanto se ne sa, non sono noti metodi del genere.

4 Come accordarsi su una chiave ed altri giochetti

Quasi in contemporanea al protocollo RSA venne proposto un semplice protocollo (il protocollo Diffie-Hellman) che permette ai nostri due soliti interlocutori di accordarsi su una chiave senza mai avere per questo necessità di trasmettersela. Il protocollo funziona in questo modo: A e B fissano un numero primo p ed un intero t tale che $0 < t < p$. A sceglie un intero (segreto) α e trasmette a B il resto a di t^α modulo p; B sceglie un intero segreto β e trasmette ad A il resto b di t^β modulo p. A e B possono ora, ciascuno per conto suo, calcolarsi la chiave: A calcola il resto di b^α e B il resto di a^β (sempre modulo p), entrambi otterranno il resto di $t^{\alpha\beta}$ modulo p, che potranno utilizzare come chiave privata comune. L'eventuale spione che avesse intercettato le trasmissioni tra A e B conoscerà sia t che p, e si sarà visto passare sotto il naso delle potenze di t modulo p, ma, se p è sufficientemente grande, non saprà calcolare la chiave perché, come abbiamo accennato sopra, egli sarà incapace di calcolare i logaritmi (α e β) in aritmetica modulo p.

Come si vede, ciò che fa funzionare questa procedura è il fatto che le funzioni di elevazione a potenza (per α e β) commutano tra loro (il risultato finale non dipende dall'ordine in cui sono eseguite: $(t^\alpha)^\beta = (t^\beta)^\alpha$). Il protocollo di Diffie-Hellman viene applicato anche in contesti diversi dall'aritmetica modulare e con altre funzioni 'facili con inversa difficile' che commutino tra loro.



Esiste anche un modo per inviare in modo sicuro un messaggio senza bisogno di alcuna chiave, è quello schematizzato a lato. A invia a B un messaggio in un cofanetto chiuso da un lucchetto. B non ha la chiave di questo lucchetto e quindi non può apre il cofanetto, piuttosto gli applica un altro lucchetto lo restituisce ad A. Ora A può asportare il suo lucchetto (ne ha la chiave!) e spedire ancora il cofanetto a B, che non avrà difficoltà ad aprirlo con la sua chiave. L'aspetto matematicamente interessante di questa piccola storia è che, di nuovo, ciò che la fa funzionare è la proprietà commutativa: l'aspetto del cofanetto dopo che siano stati applicati i due lucchetti è lo stesso indipendentemente dall'ordine in cui essi siano stati applicati. Sarebbe ben diversa la storia se B avesse invece pensato di infilare il cofanetto ricevuto in un cofanetto più grande e assicurare quest'ultimo col suo lucchetto! Vediamo direttamente con un esempio numerico come si può implementare questa procedura usando le potenze in aritmetica modulare. A e B possono concordare il numero primo 17, ciascuno decide per sé un numero segreto che sia coprimo con $\varphi(17) = 17 - 1 = 16$ (diciamo 5 per A e 3 per B) ed usa come 'lucchetto' l'elevazione a potenza di esponente questo numero modulo 17. Supponiamo che A voglia trasmettere il messaggio "10". Allora A calcola $10^5 \equiv_{17} 6$ e invia 6 a B, il quale calcola $6^3 \equiv_{17} 12$ e restituisce 12 ad A. Ora A deve 'togliere il suo lucchetto'. In che modo? Applicando l'inversa della funzione di elevazione alla quinta potenza modulo 17. Esattamente come per RSA utilizziamo il teorema di Fermat-Eulero: poiché $\varphi(17) = 16$ e $5 \cdot 13 \equiv_{16} 1$, si ha $(a^5)^{13} \equiv_{17} a$ per ogni intero a. Dunque, per 'togliere il lucchetto' A eleva alla tredicesima potenza: calcola $12^{13} \equiv_{17} 14$ e manda 14 a B. Similmente, infine, poiché $3 \cdot 11 \equiv_{16} 1$, B calcola 14^{11} modulo 17, ottenendo così 10: il messaggio è arrivato a destinazione.

5 Conclusione

Queste note hanno solo provato a dare un assaggio della matematica che sta dietro alla moderna crittografia e non hanno, ovviamente, alcuna pretesa di completezza. Molte altre sono le tecniche matematiche utilizzabili e molti i protocolli crittografici che le impiegano. Gli stessi obiettivi di

base della crittografia sono stati solo superficialmente descritti—ad esempio, non si è discusso dei problemi legati all'autenticazione dei dati: come fa il destinatario di un messaggio ad esser sicuro che il mittente sia effettivamente chi dichiara di essere? E come accertarsi che il messaggio non sia stato alterato, per errore o di proposito, prima di giungere a destinazione?

Si è comunque cercato di dare qualche informazione su un tipo di applicazione della matematica (e anche di vecchissima matematica) alla quale nessuno aveva pensato sino a pochi decenni fa. Sembra molto opportuno chiudere con le parole di Godfrey H. Hardy, un importantissimo studioso inglese di teoria dei numeri. Durante il primo anno della seconda guerra mondiale egli scrisse un breve libro, 'Apologia di un matematico', nel quale cercava di trarre un bilancio della sua vita e della sua attività. La teoria dei numeri era allora considerata (con ragione!) un settore della matematica di grandissima bellezza e valore intrinseco, ma di nessuna utilità esterna alla matematica stessa. Hardy ne rivendica, per lo meno, la non nocività: *"C'è una conclusione facile e confortante per un vero matematico. La vera matematica non ha alcun effetto sulla guerra. Nessuno ha ancora scoperto un uso bellico della teoria dei numeri . . . e sembra molto improbabile che se ne scopra uno ancora per molti anni."* Eccellente matematico, Hardy, ma pessimo profeta!