Capitolo 1

Panoramica sulla crittografia e le sue applicazioni

La gente è sempre stata affascinata dalla possibilità di nascondere informazioni agli altri. Da bambini, molti di noi avevano decodificatori magici per scambiare messaggi in codice con gli amici e tenere i nostri segreti nascosti a genitori, fratelli o insegnanti. La Storia è piena di esempi di persone che cercarono di nascondere informazioni ai loro avversari. Re e generali comunicavano con le truppe usando semplici metodi crittografici per impedire ai nemici di venire a conoscenza di informazioni militari riservate. Si narra che Giulio Cesare usasse un semplice cifrario noto oggi con il suo nome.

Con l'evoluzione della società, è cresciuta la necessità di metodi più sofisticati per la protezione dei dati. Ora, nell'era dell'informazione, questa necessità è più pronunciata che mai. Con il mondo che diventa sempre più connesso cresce la richiesta di informazione e di servizi elettronici e, con essa, si crea una maggior dipendenza dai sistemi elettronici. Lo scambio su Internet di informazioni riservate, come i numeri di carta di credito, è già una pratica comune. Proteggere i dati e i sistemi elettronici è cruciale per il nostro stile di vita.

Le tecniche necessarie per proteggere i dati appartengono al campo della crittografia. In realtà la materia ha tre nomi: **crittografia**, **crittologia** e **crittanalisi**, che sono spesso usati in modo intercambiabile. Tuttavia, più precisamente, la crittologia studia in generale la comunicazione su canali non sicuri e i relativi problemi. Il processo di progettazione di sistemi in grado di farlo è chiamato crittografia. La crittanalisi tratta le tecniche per rompere questi sistemi. Certamente non è possibile fare crittografia o crittanalisi senza avere una buona comprensione dei metodi di entrambe le aree.

Spesso il termine **teoria dei codici** è usato per descrivere la crittografia, ma ciò può ingenerare confusione. La teoria dei codici tratta come rappresentare i simboli informativi in ingresso usando simboli in uscita chiamati simboli di codice. La teoria dei codici copre tre applicazioni fondamentali: la compressione, la segretezza e la correzione di errore. Negli ultimi decenni il termine teoria dei codici è stato prevalentemente

associato ai codici correttori di errori. Quindi la teoria dei codici studia la comunicazione su canali rumorosi e come garantire che il messaggio ricevuto sia corretto, al contrario della crittografia che protegge la comunicazione su canali non sicuri.
Benché i codici correttori di errori siano un argomento secondario di questo libro, bisogna sottolineare che nei sistemi reali i codici correttori di errori sono usati congiuntamente alla cifratura perché, se il crittosistema è progettato bene, un errore su un singolo bit è sufficiente per distruggere completamente il messaggio.
La crittografia moderna è un campo che attinge pesantemente alla matematica, all'informatica e all'ingegnosità. Questo libro fornisce un'introduzione alla matematica e ai protocolli necessari per rendere sicuri la trasmissione dei dati e i sistemi elettronici, così come tecniche quali le firme digitali e la ripartizione dei segreti.

1.1 Comunicazioni sicure

Nello scenario fondamentale di comunicazione, raffigurato nella Figura 1.1, ci sono due parti, che chiameremo Alice e Bob, che vogliono comunicare tra di loro. Una terza parte, Eva, vuole intercettare la comunicazione.
Quando Alice vuole mandare a Bob un messaggio, chiamato **testo in chiaro**, lo cifra usando un metodo preventivamente concordato. Di solito si suppone che il metodo di cifratura sia noto a Eva; ciò che garantisce la segretezza del messaggio è la **chiave**. Quando Bob riceve il messaggio cifrato, chiamato **testo cifrato**, lo ritrasforma nel messaggio in chiaro usando una chiave di decifrazione.
Eva può avere uno di questi obiettivi.

1. Leggere il messaggio.
2. Trovare la chiave e quindi leggere tutti i messaggi cifrati con quella chiave.
3. Modificare il messaggio di Alice in un altro messaggio in modo che Bob pensi che Alice abbia mandato il messaggio alterato.
4. Fingersi Alice e quindi comunicare con Bob mentre Bob pensa di comunicare con Alice.

In quale caso ci troviamo dipende dalle intenzioni di Eva. I casi (3) e (4) sono legati, rispettivamente, ai problemi della integrità e della autenticazione, di cui discuteremo a breve. Un avversario attivo, corrispondente ai casi (3) e (4), in letteratura è talvolta chiamato Mallory. Gli osservatori passivi, come nei casi (1) e (2), sono talvolta chiamati Oscar. Noi useremo generalmente solo Eva, che supporremo tanto malintenzionata quanto permesso dalla specifica situazione.

1.1.1 Possibili attacchi

I principali tipi di attacco che Eva potrebbe essere in grado di fare sono quattro. La differenza tra questi tipi di attacco è la quantità di informazioni che Eva ha a disposizione quando tenta di determinare la chiave. I quattro attacchi sono i seguenti.

1. **Solo testo cifrato:** Eva ha a disposizione solo una copia del testo cifrato.

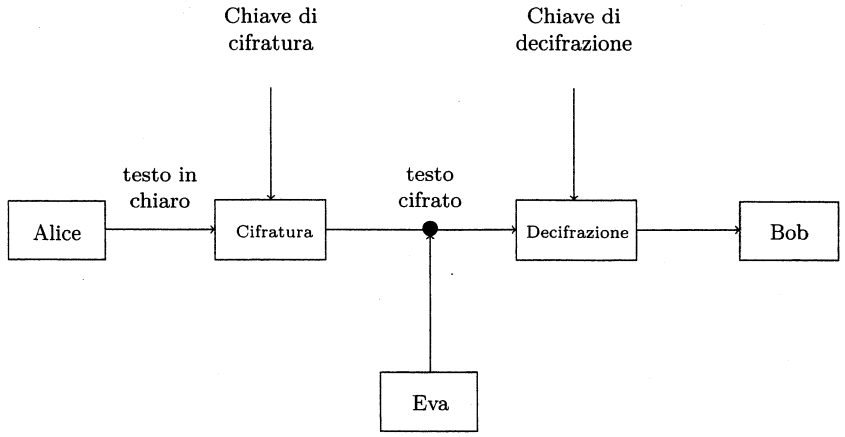

Figura 1.1 Lo scenario fondamentale di comunicazione per la crittografia.

2. **Testo in chiaro noto:** Eva ha una copia del testo cifrato e del corrispondente testo in chiaro. Supponiamo per esempio che Eva intercetti un comunicato stampa cifrato e poi veda il messaggio decifrato pubblicato il giorno seguente. Se Eva può dedurre la chiave di decifrazione e se Alice non cambia chiave, Eva può leggere tutti i messaggi futuri. Oppure, se Alice comincia sempre i suoi messaggi con "Caro Bob", allora Eva ha un frammento di testo cifrato e del corrispondente testo in chiaro. Questa informazione può essere sufficiente per trovare la chiave di molti crittosistemi deboli. In ogni caso, questa informazione è stata utile anche per crittositemi più robusti come la macchina tedesca Enigma, usata durante la Seconda Guerra Mondiale.
3. **Testo in chiaro scelto:** Eva ottiene temporaneamente accesso alla macchina cifrante. Anche se non può aprirla per trovare la chiave, può comunque cifrare un gran numero di messaggi in chiaro scelti opportunamente e provare a usare i corrispondenti messaggi cifrati per dedurre la chiave.
4. **Testo cifrato scelto:** Eva ottiene temporaneamente accesso alla macchina decifratrice, la usa per "decifrare" parecchie stringhe di simboli e cerca di usare i risultati per dedurre la chiave.

Il seguente è un possibile scenario di attacco di testo in chiaro scelto. Supponiamo di voler stabilire se un aereo è amico o nemico. Inviamo un messaggio casuale all'aereo che, automaticamente, lo cifra e lo rispedisce al mittente. Supponiamo anche che solo un aereo amico possieda la chiave corretta. Confrontiamo il messaggio proveniente dall'aereo con il messaggio cifrato correttamente: se sono uguali l'aereo è amico, altrimenti è nemico. Tuttavia il nemico potrebbe inviare a un nostro aereo un gran numero di messaggi scelti opportunamente e osservare i messaggi cifrati risultanti. Se

riesce a dedurre la chiave, il nemico può equipaggiare i suoi aerei in modo che possano fingersi amici.
Un esempio di attacco di testo in chiaro noto si racconta sia avvenuto durante la Seconda Guerra Mondiale nel deserto del Sahara. Un avamposto tedesco isolato inviava tutti i giorni lo stesso messaggio per comunicare che non c'era nulla di nuovo da segnalare, ma cifrato con la chiave da usare quel giorno. In questo modo, ogni giorno gli Alleati avevano una coppia di testo in chiaro e testo cifrato estremamente utile per determinare la chiave. Tanto che, durante la campagna del Sahara, al generale Montgomery fu ordinato di aggirare l'avamposto in modo da non interrompere le trasmissioni.
Una delle più importanti assunzioni della crittografia moderna è il **principio di Kerckhoffs**: nel valutare la sicurezza di un crittosistema si deve sempre assumere che il nemico conosca il metodo usato. Questo principio fu enunciato da Auguste Kerckhoffs nel 1883 nel trattato classico *La Cryptographie Militaire*. Il nemico può ottenere questa informazione in molti modi. Per esempio, può catturare e analizzare le macchine cifratrici e decifratrici, oppure le persone che lo conoscono possono disertare o essere catturate. Pertanto la sicurezza del sistema dovrebbe essere basata sulla chiave e non sulla segretezza dell'algoritmo. Di conseguenza noi assumeremo sempre che Eva conosca l'algoritmo usato per cifrare i messaggi.

1.1.2 Algoritmi a chiave simmetrica e a chiave pubblica

I metodi di cifratura e decifrazione ricadono in due categorie: **a chiave simmetrica** e **a chiave pubblica**. Negli algoritmi a chiave simmetrica, le chiavi di cifratura e decifrazione sono note sia ad Alice sia a Bob. Per esempio la chiave di cifratura è condivisa e la chiave di decifrazione è facilmente calcolabile da questa. In molti casi la chiave di cifratura e la chiave di decifrazione coincidono. Tutti i sistemi crittografici classici (antecedenti il 1970) sono simmetrici, così come lo sono i più recenti Data Encryption Standard (DES) e Advanced Encryption Standard (AES).
Gli algoritmi a chiave pubblica furono introdotti negli anni '70, rivoluzionando la crittografia. Supponiamo che Alice voglia comunicare con Bob in modo sicuro, ma si trova a centinaia di chilometri di distanza e non ha concordato con Bob una chiave di cifratura. Sembra quasi impossibile che ci riesca senza prima incontrare Bob per concordare una chiave oppure usando un corriere fidato per portare la chiave. Certamente Alice non può inviare un messaggio su un canale aperto per comunicare la chiave a Bob e poi inviare un messaggio cifrato con questa stessa chiave. Il fatto strabiliante è che il problema ha una soluzione, che si chiama crittografia a chiave pubblica. La chiave di cifratura è resa pubblica, ma è computazionalmente impossibile trovare la chiave di decifrazione senza conoscere informazioni note solo a Bob. L'implementazione più nota è RSA (Capitolo 6), che si basa sulla difficoltà di fattorizzare gli interi grandi. Altre versioni (Capitoli 7, 17 e 18) sono il sistema di ElGamal (basato sul problema del logaritmo discreto), NTRU (basato su reticoli) e il sistema McEliece (basato sui codici correttori di errori).
Una tecnica non matematica per effettuare la comunicazione a chiave pubblica può essere questa. Bob invia ad Alice una scatola e un lucchetto aperto. Alice mette il proprio messaggio nella scatola, la chiude con il lucchetto di Bob e la manda indietro

a Bob. Sicuramente solo Bob può aprire la scatola e leggere il messaggio. I metodi appena menzionati sono realizzazioni matematiche di questa idea. Chiaramente ci sono problemi di autenticazione che devono essere gestiti. Per esempio Eva potrebbe intercettare la prima trasmissione e sostituire il lucchetto con un altro di cui lei stessa ha la chiave. Se, successivamente, intercetta la scatola chiusa che Alice rimanda a Bob, Eva può aprirne il lucchetto e leggere il messaggio di Alice. Questo è un problema comune che deve essere affrontato in tutti questi sistemi.
La crittografia a chiave pubblica rappresenta forse il passo finale di un'interessante progressione storica. Nei primi anni della crittografia, la sicurezza dipendeva dalla segretezza del metodo di cifratura. Successivamente, si è fatta l'assunzione che il metodo fosse noto pubblicamente e la sicurezza dipendeva dal mantenere la chiave (simmetrica) privata o sconosciuta agli avversari. Nella crittografia a chiave pubblica, il metodo e la chiave sono pubblici e tutti sanno come ottenere la chiave di decifrazione. La sicurezza risiede nel fatto (o nella speranza) che questa operazione sia computazionalmente impossibile. È piuttosto paradossale che, all'aumento nel tempo della potenza degli algoritmi crittografici, abbia corrisposto un aumento nella quantità di informazioni rivelate agli avversari rispetto agli algoritmi stessi.
I metodi a chiave pubblica sono molto potenti e potrebbe sembrare che abbiano reso obsoleto l'uso della crittografia a chiave simmetrica. Tuttavia questa maggior flessibilità non è gratuita e ha un costo computazionale. La quantità di calcoli necessari negli algoritmi a chiave pubblica è tipicamente di parecchi ordini di grandezza superiore rispetto a quella di algoritmi come DES o Rijndael. L'esperienza insegna che i metodi a chiave pubblica non si dovrebbero usare per cifrare grandi quantità di dati. Per questa ragione, i metodi a chiave pubblica si usano solo in applicazioni che richiedono l'elaborazione di limitate quantità di dati (per esempio le firme digitali e l'invio di chiavi per l'uso in algoritmi a chiave simmetrica).
Ci sono due tipi di cifrari a chiave simmetrica: i cifrari a flusso (*stream ciphers*) e i cifrari a blocchi (*block ciphers*). Nei cifrari a flusso, i dati sono divisi in frammenti (composti da un singolo bit o da un singolo carattere) ed elaborati uno alla volta, fornendo in uscita i frammenti corrispondenti. Nei cifrari a blocchi, invece, i bit in ingresso sono raccolti in un blocco ed elaborati tutti insieme dall'algoritmo, fornendo in uscita un blocco di bit. Nel Paragrafo 2.11 discuteremo un esempio di cifrario a flusso basato su un registro a scorrimento con retroazione lineare. Ci occuperemo più diffusamente dei cifrari a blocchi, in particolare vedremo due esempi notevoli. Il primo è DES, il secondo è Rijndael, che fu scelto nel 2000 dal National Institute for Standards and Technology per sostituire DES. Anche i metodi a chiave pubblica, come RSA, possono essere considerati cifrari a blocchi.
Citiamo infine la distinzione storica tra diversi tipi di strumenti crittografici, ovvero i **codici** e i **cifrari**. In un codice, le parole o certe combinazioni di simboli sono sostituiti con *parole di codice* (che possono essere sequenze di simboli). Per esempio la Marina Militare inglese, durante la Prima Guerra Mondiale, usava 03680C, 36276C e 50302C per rappresentare, rispettivamente, *consegnato a, consegnato da* e *proveniente da*. I codici hanno lo svantaggio di non consentire l'uso di parole non previste. Un cifrario, invece, non prende in considerazione le strutture linguistiche del messaggio, ma cifra ogni stringa di caratteri, di senso compiuto o meno, con qualche algoritmo. Pertanto un cifrario è più versatile di un codice. Agli albori della crittografia si usavano

comunemente i codici, talvolta unitamente ai cifrari. I codici sono in uso ancora oggi; spesso le operazioni segrete sono indicate da un nome in codice. Tuttavia un segreto che debba rimanere sicuro deve essere protetto con un cifrario. In questo libro tratteremo esclusivamente cifrari.

1.1.3 Lunghezza della chiave

La sicurezza di un algoritmo crittografico è una proprietà difficile da misurare. Per la maggior parte degli algoritmi la sicurezza è legata alla difficoltà per un avversario di determinare la chiave. L'approccio più ovvio è provare tutte le chiavi possibili e vedere quali danno un testo decifrato di senso compiuto. In questo attacco, chiamato **attacco a forza bruta**, la lunghezza della chiave determina direttamente il tempo richiesto per esplorare l'intero spazio delle chiavi. Per esempio, se la chiave è lunga 16 bit, ci sono $2^{16} = 65536$ chiavi possibili. L'algoritmo DES ha chiavi di 56 bit e quindi ci sono $2^{56} \approx 7{,}2 \cdot 10^{16}$ chiavi possibili.

Potrebbe sembrare che molti dei sistemi che vedremo in questo libro si possano violare provando tutte le possibili chiavi. Tuttavia è molto più facile a dirsi che a farsi. Supponiamo di provare 10^{30} possibilità e di avere a disposizione un calcolatore in grado di svolgere 10^9 calcoli al secondo. Considerando che i secondi in un anno sono circa $3 \cdot 10^7$, occorrono poco più di $3 \cdot 10^{13}$ anni per completare la ricerca, più della vita stimata dell'universo.

Chiavi lunghe sono vantaggiose, ma non garantiscono che il lavoro dell'avversario sia difficile; anche l'algoritmo gioca la sua parte. Alcuni algoritmi possono essere attaccati con tecniche diverse dalla forza bruta oppure non usano efficientemente i bit della chiave. È importante ricordare che non tutti gli algoritmi con chiavi di 128 bit sono uguali!

Per esempio, uno dei crittosistemi più facili da rompere è il cifrario a sostituzione, discusso nel Paragrafo 2.4, il cui numero di chiavi possibili è $26! \approx 4 \cdot 10^{26}$. Per contro, l'algoritmo DES (vedi Capitolo 4) ha solo $2^{56} \approx 7{,}2 \cdot 10^{16}$ chiavi, ciò nonostante trovare una chiave DES richiede oltre un giorno di calcolo su un computer progettato appositamente. La differenza è che l'attacco al cifrario a sostituzione sfrutta la struttura della lingua usata, mentre l'attacco a DES è a forza bruta, ovvero si provano tutte le chiavi.

Un attacco a forza bruta dovrebbe essere l'ultima risorsa. Un crittanalista spera sempre di trovare un attacco più rapido. Vedremo, per esempio, l'analisi di frequenza (usata per il cifrario a sostituzione e di Vigenère) e gli attacchi del compleanno (usati per i cifrari basati sul problema del logaritmo discreto).

Vale la pena segnalare al lettore che, solo perché un algoritmo sembra sicuro oggi, non significa che lo sarà in futuro. Gli uomini hanno inventato metodi molto ingegnosi per attaccare i protocolli crittografici. Nella crittografia moderna ci sono molti esempi di algoritmi o protocolli attaccati con successo grazie a debolezze causate da implementazioni scadenti o, semplicemente, grazie a progressi tecnologici. L'algoritmo DES ha resistito 20 anni di esami da parte della comunità dei crittografi, prima di soccombere agli attacchi di un computer parallelo ben progettato. La continua ricerca nel campo dell'elaborazione quantistica potrebbe modificare radicalmente le basi dei futuri algoritmi crittografici.

Per esempio, la sicurezza di molti sistemi che studieremo dipende dalla difficoltà di fattorizzare un numero intero n grande, approssimativamente di 200 cifre. Il metodo usato fino dalle scuole elementari è di dividere n per tutti i numeri primi minori della radice quadrata di n. Ci sono approssimativamente $4 \cdot 10^{97}$ numeri primi minori di 10^{100} e provarli tutti è impossibile, se pensiamo che il numero stimato di elettroni nell'universo è minore di 10^{90}. Molto prima di avere finito i calcoli, riceverete una chiamata dalla compagnia elettrica che vi chiede di interrompere il lavoro. Chiaramente si devono usare algoritmi di fattorizzazione più sofisticati di questo attacco a forza bruta. Quando è stato inventato l'algoritmo RSA esistevano già alcuni buoni algoritmi di fattorizzazione, ma si prevedeva che, nell'immediato futuro, non sarebbe stato possibile fattorizzare un numero di 129 cifre come proposto nella sfida RSA (vedi il Paragrafo 6.5). Tuttavia, i progressi nei campi degli algoritmi e delle architetture dei calcolatori hanno reso abbastanza normali fattorizzazioni di questo ordine di grandezza (anche se richiedono ugualmente considerevoli risorse di calcolo), motivo per cui, oggi, si raccomanda di usare parecchie centinaia di cifre per avere sicurezza. Se mai si costruissero computer quantistici su larga scala, anche la fattorizzazione di questi numeri sarebbe facile e si dovrebbe ripensare tutto lo schema RSA (e anche molti altri).

Naturalmente viene da chiedersi se esistono crittosistemi indecifrabili e perché non siano usati sempre.

La risposta è affermativa; esiste un sistema, chiamato stringa monouso, che è indecifrabile. Anche un attacco a forza bruta non è in grado di rivelare la chiave. Sfortunatamente il costo di una stringa monouso è enorme. Infatti richiede lo scambio di una chiave lunga come il testo in chiaro che può essere usata una volta sola. Pertanto si scelgono algoritmi che, implementati correttamente e con un'appropriata lunghezza della chiave, sono indecifrabili in un tempo ragionevole.

Sempre con riferimento alla lunghezza della chiave, in molti casi bisogna tenere presente che, matematicamente, è possibile aumentare la sicurezza allungando leggermente la chiave, ma, in pratica, non è sempre possibile. Quando si lavora con microchip che gestiscono parole di 64 bit, per aumentare la lunghezza della chiave da 64 a 65 bit può essere necessario riprogettare l'hardware, che è un'operazione costosa. Il progetto di un buon crittosistema deve basarsi su considerazioni sia matematiche sia ingegneristiche.

Con riferimento alla lunghezza dei numeri, l'intuito potrebbe far pensare che lavorare con numeri di 20 cifre richieda un tempo doppio rispetto a lavorare con numeri di 10 cifre. In alcuni algoritmi questo è vero. Tuttavia, se contate fino a 10^{10}, non siete a metà strada rispetto a contare fino a 10^{20}; siete a un diecimiliardesimo del percorso. Allo stesso modo, un attacco a forza bruta contro una chiave di 60 bit richiede circa un miliardo di tentativi in più rispetto a un attacco a una chiave di 30 bit.

Ci sono due modi per misurare la grandezza di un numero: il suo valore, n, oppure il numero di cifre nella sua rappresentazione decimale (o, alternativamente, binaria), approssimativamente $\log_{10}(n)$. Per elevare al quadrato un numero n di k cifre usando l'algoritmo comunemente insegnato nelle scuole elementari, occorrono k^2 moltiplicazioni di numeri composti da una sola cifra, ovvero approssimativamente $(\log_{10} n)^2$ operazioni. Il numero di divisioni necessarie per fattorizzare un numero n dividendolo per tutti i numeri primi fino alla radice quadrata di n è circa $n^{1/2}$. In generale un algoritmo che viene eseguito in un tempo che cresce con $\log n$ è più desiderabile di un algoritmo

che richiede un tempo che cresce con una potenza di n. Proseguendo l'esempio, se raddoppiamo il numero di cifre che compongono il numero n, il tempo richiesto dall'elevamento al quadrato cresce di un fattore 4, mentre il tempo richiesto dalla fattorizzazione cresce enormemente. Esistono algoritmi più efficienti per svolgere entrambe queste operazioni, ma, a oggi, la fattorizzazione richiede significativamente più operazioni della moltiplicazione.
Vedremo in seguito algoritmi che impiegano un tempo proporzionale a $\log n$ per svolgere certe operazioni, per esempio trovare il Massimo Comun Divisore oppure elevare a potenza in aritmetica modulare. Per altri calcoli, gli algoritmi migliori conosciuti richiedono un tempo solo leggermente inferiore a una potenza di n, per esempio la fattorizzazione e il logaritmo discreto. L'interazione tra algoritmi veloci e algoritmi lenti è la base di parecchi algoritmi crittografici descritti in questo libro.[1]

1.2 Applicazioni della crittografia

La crittografia non serve solo a cifrare e decifrare messaggi, ma anche a risolvere problemi reali che richiedono la sicurezza delle informazioni. I principali obiettivi sono quattro.

1. **Riservatezza:**[2] Eva non dovrebbe essere in grado di leggere i messaggi di Alice diretti a Bob. Gli strumenti principali sono gli algoritmi di cifratura e decifrazione.

2. **Integrità dei dati:** Bob vuole essere sicuro che il messaggio di Alice non sia stato alterato. Una causa potrebbero essere, per esempio, gli errori di trasmissione. Oppure un antagonista potrebbe intercettare la trasmissione e alterarla prima che raggiunga il destinatario. Molte primitive crittografiche, come le funzioni hash, forniscono metodi per identificare la modifica dei dati, intenzionale o accidentale.

3. **Autenticazione:** Bob vuole essere sicuro che soltanto Alice possa avere inviato il messaggio appena ricevuto. In questo ambito sono inclusi anche gli schemi di identificazione e i protocolli con password (nel qual caso Bob è il computer). Ci sono, in realtà, due tipi di autenticazione che interessano la crittografia: l'autenticazione dell'entità e l'autenticazione dell'origine dei dati. Lo scopo dell'autenticazione dell'entità è provare l'identità dei soggetti coinvolti nella comunicazione, pertanto si usa spesso il termine *identificazione*. L'autenticazione dell'origine dei dati ha il compito di collegare i dati alle informazioni relative alla loro origine, quali l'identità del creatore e la data di creazione.

4. **Non ripudiabilità:** Alice non può sostenere di non aver inviato il messaggio. La proprietà di non ripudiabilità è particolarmente importante nelle applicazioni di commercio elettronico, in cui il consumatore non può negare di aver concesso l'autorizzazione a un acquisto.

[1] Nel resto del libro si farà più volte riferimento a problemi facili e problemi difficili. Con il termine "problema difficile" si indicherà un problema computazionalmente intrattabile, ovvero un problema il cui algoritmo risolutivo impiega un numero di operazioni talmente grande da non poter essere eseguito in pratica. Viceversa con il termine "problema facile" indicheremo un problema il cui algoritmo risolutivo impiega un numero di operazioni abbastanza piccolo da poter essere eseguito. (*N.d.Rev.*)

[2] Per indicare la riservatezza è molto diffuso anche il termine "confidenzialità". (*N.d.Rev.*)

Autenticazione e non ripudiabilità sono concetti strettamente connessi, ma diversi. Nei crittosistemi a chiave simmetrica Bob può essere sicuro che un messaggio venga da Alice (o da qualcuno che conosce la chiave di Alice) perché nessun altro potrebbe avere cifrato il messaggio che Bob ha decifrato correttamente. Pertanto l'autenticazione è automatica. Tuttavia, Bob non può provare che Alice abbia mandato il messaggio, perché potrebbe essersi mandato il messaggio lui stesso. Pertanto la non ripudiabilità è fondamentalmente impossibile. Nei crittosistemi a chiave pubblica si possono ottenere sia autenticazione che non ripudiabilità (Paragrafo 6.7 e Capitolo 9).
Gran parte di questo libro presenta specifiche applicazioni crittografiche, sia nel testo sia negli esercizi. Quella che segue ne è una panoramica.

Firma digitale: una delle caratteristiche più importanti di una lettera scritta a mano è la firma. Quando un documento è firmato, l'identità dell'autore è legata al messaggio. Si può supporre che sia difficile per un'altra persona falsificare la firma copiandola da un altro documento. Per contro, i messaggi elettronici sono molto facili da copiare esattamente. Come possiamo evitare che un antagonista ritagli la firma da un documento e la incolli su un diverso documento elettronico? Studieremo protocolli crittografici che permettono ai messaggi elettronici di essere firmati in modo che tutti possano ritenere che il firmatario sia veramente colui che ha firmato il documento, e che questi non possa negare di averlo fatto.

Identificazione: quando ci si collega a una macchina oppure si attiva un collegamento, un utente deve identificarsi. Limitarsi a digitare il proprio nome non è sufficiente, perché non prova che l'utente sia davvero chi dice di essere. Tipicamente si usa una password. Vedremo alcuni metodi di identificazione. Nel capitolo sull'algoritmo DES discuteremo i file che contengono le password. Successivamente presenteremo lo schema di identificazione di Feige-Fiat-Shamir, che è un metodo a conoscenza zero per provare l'identità senza rivelare la password.

Instaurazione della chiave: quando si devono cifrare grosse quantità di dati, è opportuno usare algoritmi di cifratura a chiave simmetrica. Ma come può Alice dare la chiave segreta a Bob se non può incontrarlo di persona? Esistono diverse tecniche per risolvere questo problema. Una possibilità è usare la crittografia a chiave pubblica. Un altro metodo è l'algoritmo di instaurazione della chiave di Diffie-Hellman. Un diverso approccio al problema consiste nell'avere una terza parte fidata che dia le chiavi ad Alice e Bob. Due esempi sono lo schema di Blum per la generazione della chiave e il sistema Kerberos, che è un protocollo crittografico molto diffuso che fornisce autenticazione e sicurezza per lo scambio di chiavi in una rete.

Condivisione di segreti: nel Capitolo 12 introdurremo gli schemi di condivisione dei segreti. Supponiamo di avere la combinazione della cassaforte di una banca, ma non ci fidiamo a comunicare l'intera combinazione a una sola persona. Piuttosto vogliamo suddividere la combinazione tra le persone di un gruppo, in modo che almeno due di questi debbano essere presenti per aprire la cassaforte. La condivisione di segreti risolve questo problema.

Protocolli per la sicurezza: come possiamo effettuare transazioni sicure su canali non sicuri come Internet e come possiamo proteggere le informazioni sulla carta di credito da commercianti fraudolenti? Discuteremo vari protocolli, tra cui SSL e SET.

Moneta elettronica: le carte di credito e altri strumenti simili sono comodi, ma non garantiscono l'anonimato. Sicuramente una forma elettronica della moneta sarebbe utile, almeno per alcune persone. Tuttavia tutto ciò che è elettronico può essere copiato. Vedremo un esempio di un sistema di moneta elettronica che garantisce l'anonimato, ma identifica i falsari.

Giochi: com'è possibile lanciare una moneta o giocare a poker con persone che non sono nella stessa stanza? Distribuire le carte, per esempio, presenta problemi. Vedremo come alcune idee crittografiche possano risolvere questi problemi.

Capitolo 2

Crittosistemi classici

Nel corso della storia, per l'uomo è sempre stato importante avere dei metodi che permettessero di camuffare i suoi messaggi rendendoli incomprensibili agli occhi degli avversari. In questo capitolo verranno presentati alcuni dei più vecchi crittosistemi in uso prima dell'avvento del computer. Questi crittosistemi sono troppo deboli per poter essere ancora usati oggi, soprattutto avendo a disposizione il calcolatore, ma illustrano bene molte idee importanti della crittologia.
Per trattare questi semplici crittosistemi si adotteranno le seguenti convenzioni.

- Il *testo in chiaro* sarà scritto in lettere minuscole, mentre il *TESTO CIFRATO* sarà scritto in lettere maiuscole (tranne nei problemi al calcolatore).
- A ogni lettera dell'alfabeto sarà assegnato un numero:

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

q	r	s	t	u	v	w	x	y	z
16	17	18	19	20	21	22	23	24	25

  Come si vede, alla lettera a viene assegnato il numero 0, mentre alla lettera z viene assegnato il numero 25. Questa convenzione (diversa da quella abituale in cui a corrisponde a 1 e z a 26) è standard nei crittosistemi elementari che verranno considerati.

- Gli spazi e la punteggiatura sono omessi. Anche se questa convenzione può sembrare fastidiosa, è quasi sempre possibile ricollocare gli spazi nel testo in chiaro dopo la decifrazione. Se si lasciassero gli spazi, si avrebbero ovviamente due possibilità. Essi potrebbero essere lasciati come spazi, e allora si avrebbero talmente tante informazioni sulla struttura del messaggio che la decifrazione