Alma Mater Studiorum · Università di Bologna
Campus di Cesena

DIPARTIMENTO DI INFORMATICA – SCIENZA E INGEGNERIA
Corso di Laurea in Ingegneria e Scienze Informatiche

CRITTOGRAFIA QUANTISTICA

Tesi di laurea in
Crittografia

Relatore:
Prof. Luciano Margara

Presentata da:
Antonio Emanuele Pepe

Anno Accademico 2022-2023

"Every piece of software is going to have vulnerabilities which can be exploited and that's a basic rule of life."
– Henry Fisher

"La sicurezza è un'illusione. Non puoi controllare tutto, ma puoi controllare come reagisci."
– ChatGPT 3.5

Sommario

La crittografia rappresenta un elemento cruciale per garantire la sicurezza delle comunicazioni e la protezione dei dati. L'evoluzione dalla crittografia classica a quella moderna ha contribuito alla definizione di standard e protocolli su cui facciamo affidamento quotidianamente. Tuttavia, il sempre crescente livello di sofisticatezza delle minacce informatiche, in particolare attraverso strategie di ingegneria sociale, ha sollevato interrogativi sulla capacità della crittografia convenzionale di affrontare con successo tali nuove sfide.

Per questo motivo, la ricerca di soluzioni innovative ha portato all'interesse per la crittografia quantistica, un settore che sfrutta i principi della meccanica quantistica per offrire soluzioni intrinsecamente sicure.

Questa tesi si propone l'obiettivo di esplorare la crittografia quantistica. Verranno affrontati inizialmente i concetti di base della crittografia e della meccanica quantistica, che manterranno la loro rilevanza anche quando sarà introdotta la crittografia quantistica. Ciò fornirà una base necessaria per comprendere tutti gli argomenti trattati.

La lettura del presente lavoro vuole fornire al lettore una panoramica volta a comprendere le motivazioni, i progressi e le difficoltà che hanno portato a modificare i paradigmi della crittografia moderna. Tale cambiamento si rende necessario per affrontare adeguatamente le minacce alla sicurezza informatica, in modo da proporre soluzioni efficaci che possano mitigare questo rischio.

Indice

Capitolo 1

Introduzione alla Crittografia

La crittografia per come la conosciamo oggi può essere uno strumento efficace ma non per questo esente da vulnerabilità. Uno dei principali indicatori della forza di un sistema crittografico è quello del tempo e delle risorse impiegate per decifrare un testo da esso prodotto. Si può affermare che sia praticamente impossibile decifrare il risultato di un buon sistema crittografico; non basterebbe tutto il tempo dell'Universo e non avremmo la potenza di calcolo necessaria [3]. È ragionevole supporre che tali sistemi possano resistere al più tenace dei crittoanalisti, ma nessuno può dimostrarlo con certezza.

I processi di cifratura e decifrazione fanno largo uso di funzioni matematiche note come algoritmi crittografici, detti anche cifrari. Un qualsiasi testo in chiaro può essere cifrato mediante questi algoritmi a partire da una chiave, che essa sia una parola, un numero o una frase.

Si può intuire che l'esito di un processo di cifratura vari a seconda della chiave utilizzata. Anche il livello di sicurezza che si intende mantenere dipende fortemente dalla scelta dell'algoritmo e dalla confidenzialità nei confronti della chiave.

Con l'aumento della popolarità del commercio elettronico e delle tecnologie di comunicazione su internet, la sicurezza delle informazioni è diventata sempre più rilevante nel mondo moderno. La maggior consapevolezza e le preoccupazioni in materia di sicurezza hanno spinto alla diffusione di sistemi crittografici che risultino essere sempre più efficienti, in modo da non intaccare l'esperienza finale percepita dall'utente.

In questo capitolo si vogliono discutere alcuni concetti introduttivi al mondo della crittografia, che possano facilitare la comprensione dei successivi capitoli. Verranno fornite definizioni di base e resi noti i principi di funzionamento alla base dello scambio di un messaggio attraverso gli approcci più comuni. La scelta del linguaggio è volutamente semplice, in modo da non introdurre sin da subito concetti e notazioni non necessarie.

1.1 Terminologia di base

Si elencano di seguito le definizioni di alcune terminologie di cui è utile conoscere il loro significato.

Testo in chiaro

Il messaggio originale che spesso viene indicato con m. Il contenuto è generalmente di senso compiuto e risulta essere quello interpretato dal livello applicativo (browser, client di posta, ecc.).

Testo cifrato

Il risultato di un processo di cifratura. Un messaggio si considera cifrato quando è necessario decifrarlo per comprendere il contenuto informativo.

Cifratura

La procedura con cui un messaggio viene reso illeggibile a chiunque sia estraneo alla comunicazione.

Decifrazione

La procedura con cui si ripristina il contenuto originale del messaggio.

Chiave

Una stringa di caratteri che viene utilizzata da un algoritmo di cifratura per rendere illeggibile un messaggio. In base all'utilizzo che viene fatto, una chiave può essere:

Privata. La stessa chiave viene impiegata sia per cifrare che per decifrare un messaggio. Risulta essere un metodo più veloce che dà origine a un sistema a chiave simmetrica.

Pubblica. Il messaggio viene cifrato con una chiave nota. Una chiave privata, differente da quella pubblica, è la sola in grado di decifrare il messaggio.

Spazio delle chiavi

L'insieme di tutte le possibili chiavi che possono essere usate da un determinato algoritmo di cifratura.

Attacco

Il tentativo di un malintenzionato di impadronirsi di informazioni o di utilizzare un computer compromesso con lo scopo di lanciare ulteriori attacchi. Questo è possibile grazie alla diffusione di software infetto, tentativi di phishing[1], ingegneria sociale, e altre tecniche. Un attacco può essere:

[1] Attacco che coinvolge l'invio di comunicazioni fraudolente che si presentano come provenienti da entità legittime, al fine di ingannare le persone e ottenere informazioni sensibili come password, dati finanziari o personali.

Passivo. Il malintenzionato cerca di carpire le informazioni a lui necessarie senza alterare il sistema. Questi attacchi sono difficili da individuare proprio perché risultano essere trasparenti.

Attivo. Il malintenzionato agisce con lo scopo di alterare il sistema e modifica le informazioni che in esso vi transitano.

1.2 Finalità della Crittografia

La crittografia è definita come lo studio e l'applicazione di metodi per la comunicazione sicura. In senso più ampio, questo campo comprende la progettazione e lo sviluppo di protocolli che contrastino l'azione di parti esterne. Tali protocolli fanno riferimento a diversi aspetti legati alla sicurezza e, nello specifico, devono permettere tre principi fondamentali: la confidenzialità dei dati, la loro integrità e autenticazione.

1.2.1 Confidenzialità

Per confidenzialità si intende la capacità di inviare un messaggio a un destinatario ed essere sicuri che questo sia il solo a poterlo leggere.

Esistono due classi di crittografia che conferiscono confidenzialità al dato che si vuole trasmettere: simmetrica e asimmetrica; entrambe differiscono nel modo in cui viene utilizzata la chiave. Gli algoritmi di crittografia simmetrica, come Data Encryption Standard (DES) e Advanced Encryption Standard (AES), si basano sul presupposto che ogni partecipante della comunicazione conosca una chiave condivisa precedentemente. Lo stesso livello di confidenzialità si può raggiungere anche attraverso algoritmi asimmetrici come Rivest, Shamir and Adleman (RSA) e l'infrastruttura a chiave pubblica (PKI).

Crittografia simmetrica

Come accennato in precedenza, gli algoritmi simmetrici necessitano di una chiave nota a priori, che i partecipanti della conversazione devono mantenere segreta.

Si consideri lo scenario in cui due soggetti vogliono scambiarsi in segreto una lettera per mezzo postale. Alice è il mittente, Bob è il destinatario. Entrambi hanno una chiave che apre lo stesso lucchetto. Lo scambio avviene come segue:

1. Alice scrive un messaggio, lo pone in una scatola e la chiude con il lucchetto.
2. La lettera viene spedita e viaggia chiusa all'interno della scatola.
3. Bob riceve la scatola e usa la sua chiave per aprire il lucchetto. Un'eventuale risposta può essere mandata con la stessa scatola e lucchetto (Figura 1.1a).

(a) Crittografia simmetrica. (b) Crittografia asimmetrica.

Figura 1.1: Scambio di un messaggio.

Crittografia asimmetrica

Gli algoritmi asimmetrici, chiamati anche a chiave pubblica, sono stati sviluppati in modo da impiegare una chiave di cifratura differente da quella di decifrazione. Inoltre, la chiave di decifrazione non può essere ricavata facilmente da quella di cifratura e viceversa (Figura 1.1b).

Tali algoritmi utilizzano una chiave pubblica e una chiave privata, entrambe capaci di cifrare un messaggio. Una volta cifrato il messaggio con una chiave, può essere decifrato solo con l'altra.

Usando questi principi è possibile raggiungere il livello di confidenzialità e autenticità richiesto. In aggiunta, non vi è alcuna chiave condivisa e la stessa deve essere di lunghezza notevole. Nella pratica, si fa uso di chiavi che vanno dai 512 ai 4096 bit. Un valore di 2048 bit è generalmente ritenuto più che sufficiente.

La natura asimmetrica porta questi algoritmi a essere sostanzialmente più lenti della controparte: i principi alla base del loro funzionamento fanno riferimento alla risoluzione di problemi computazionalmente complessi. Per questo motivo, gli algoritmi asimmetrici trovano il loro utilizzo nello scambio di piccole quantità di dati, aiutati dal fatto che una delle due chiavi può essere resa pubblica durante la comunicazione.

Crittografia asimmetrica - Confidenzialità

Ci si potrebbe domandare come si possa raggiungere un livello di confidenzialità con algoritmi asimmetrici. Nel momento in cui la chiave pubblica viene usata per cifrare un messaggio, solo la chiave privata è in grado di decifrarlo. Nel caso quest'ultima risulti essere compromessa si procederà a generare una nuova coppia di chiavi.

Si consideri questo esempio, dove Alice e Bob vogliono conferire confidenzialità allo scambio di un messaggio:

1. Alice si procura la chiave pubblica di Bob.
2. Alice usa questa chiave per cifrare un messaggio e lo invia a Bob.
3. Bob decifra il messaggio ricevuto con la sua chiave privata.

Dal momento che Bob è l'unico possessore della chiave privata, è anche il solo in grado di decifrare il messaggio. Questo garantisce confidenzialità.

1.2.2 Integrità

Un messaggio che arriva a destinazione è integro quando il contenuto ottenuto non ha subito modifiche durante il tragitto e risulta essere uguale al messaggio originale.

Per assicurare che il dato trasmesso sia integro si ricorre all'utilizzo di *hash*. Un hash è il risultato di una funzione matematica unidirezionale, vale a dire una funzione facile da calcolare ma molto difficile da invertire.

Una funzione di hash riceve in input una stringa arbitraria e produce una sua rappresentazione di lunghezza fissa. Tale risultato viene spesso indicato con il termine *digest*.

Data una funzione di hash, questa non potrà mai produrre lo stesso risultato a partire da due stringhe differenti. Ogni cambiamento dell'input, seppur minimo, produrrà un hash differente. Questo funzionamento è fondamentale per capire se un dato è stato alterato, per errore o intenzionalmente.

Hashing crittografico

In matematica, ci si riferisce alla funzione $h = H(x)$ per descrivere il comportamento di una funzione di hash. Nella notazione, H indica la particolare funzione di hash utilizzata, x rappresenta la stringa di lunghezza arbitraria fornita in input, mentre h è il risultato della funzione di hash.

Una buona funzione di hash deve possedere le seguenti caratteristiche:

1. Il suo input deve poter essere di qualunque lunghezza.
2. Il risultato prodotto ha lunghezza costante.
3. $H(x)$ è una funzione *one-way*, facile da calcolare e difficile da invertire.
4. $H(x)$ non produce collisioni, due input differenti produrranno sempre due output differenti.

Una funzione di hash non può contrastare modifiche deliberate da parte di un malintenzionato. Queste funzioni non forniscono la possibilità di identificare il mittente, consentendo a chiunque di calcolare l'hash di un dato a condizione che conosca la funzione utilizzata.

Il mittente di una conversazione può solo verificare la coerenza dell'hash di un messaggio. Un soggetto esterno potrebbe intercettare il messaggio, modificarlo e ricalcolare l'hash; questa operazione risulterebbe trasparente per il destinatario. Tale tipo di attacco è definito *man-in-the-middle*.

Crittografia asimmetrica - Integrità

Sulla base di quanto descritto finora, è possibile garantire confidenzialità, integrità e autenticazione in un processo crittografico asimmetrico. Si faccia riferimento al seguente esempio:

1. Alice vuole inviare un messaggio a Bob assicurandosi che sia l'unico a poterlo leggere (confidenzialità). Alice utilizza la chiave pubblica di Bob per cifrare il messaggio. Bob è l'unico in grado di decifrarlo.

2. Bob vuole che il messaggio provenga effettivamente da Alice e che sia integro. Alice usa la sua chiave privata per cifrare l'hash del messaggio (autenticazione).

3. Alice manda a Bob il messaggio e l'hash, entrambi cifrati.

4. Bob utilizza la chiave pubblica di Alice per verificare l'hash (integrità).

5. Bob utilizza la sua chiave privata per decifrare il messaggio.

1.2.3 Autenticazione

Il principio di autenticazione garantisce che il messaggio ricevuto non sia un falso e la provenienza sia conforme a quanto esso dichiara.

Esistono vari protocolli che sono adoperati per assicurare l'autenticazione di un messaggio, uno di questi è il cosiddetto *hash message authentication code* (HMAC).

Gli elementi necessari per implementare HMAC sono una chiave condivisa, mantenuta segreta dai partecipanti, e una funzione hash. Questi sono concordati tra mittente e destinatario prima che la trasmissione abbia inizio. Di seguito sono mostrati gli step generali e semplificati del processo, escludendo aspetti tecnici (Figura 1.2):

1. Il mittente genera un hash a partire dalla sua chiave e dal messaggio che vuole inviare. Si genera così un digest HMAC.

2. Il mittente invia il messaggio originale unitamente al digest.

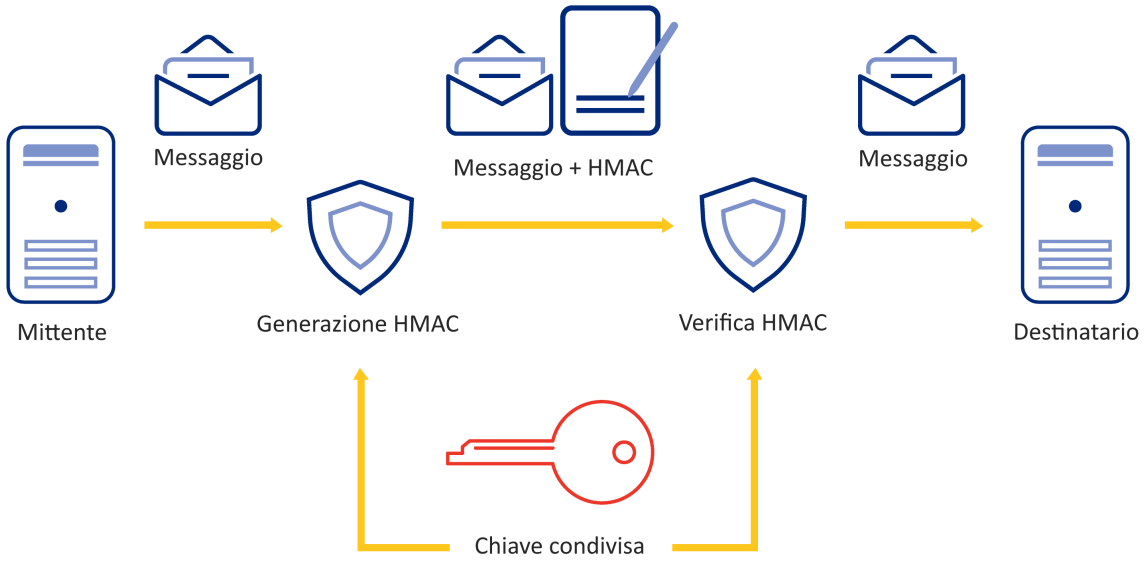


Figura 1.2: Diagramma del funzionamento di HMAC [29].

3. Ricevuto il messaggio, il destinatario lo separa dal digest. Procede poi a calcolare localmente un nuovo digest del messaggio ottenuto, questa volta combinando la sua chiave segreta.

4. Nel caso in cui l'hash calcolato sia uguale a quello ricevuto, il messaggio risulta autentico e non alterato.

Si osserva che questa procedura è in grado di garantire che il messaggio provenga dal mittente sperato e che non sia stato compromesso.

Ricordando che una funzione hash è in grado di generare lo stesso output se e solo se l'input è lo stesso, solo la combinazione del messaggio originale e della chiave segreta possono generare la stessa *impronta*[2]. Questo garantisce integrità.

Si noti che la chiave segreta è coinvolta nel processo; per questo motivo, solo il mittente e il destinatario possono generare lo stesso digest del messaggio da allegare. Questo garantisce autenticazione.

1.3 Crittografia classica

In relazione alla crittografia classica, spesso si fa riferimento a procedure e cifrari utilizzati nel corso della storia che non prevedevano tecniche sofisticate, proprie della crittografia moderna. Tali sistemi non sono quasi più utilizzati, ma forniscono una buona prospettiva sul funzionamento generale della crittografia.

[2] Spesso ci si riferisce al digest di una funzione di hash anche con il termine *impronta digitale* o *impronta*.

1.3.1 Cifrari a sostituzione

Una delle prime implementazioni della crittografia simmetrica è riscontrata nei cifrari a sostituzione. Considerando un semplice testo come messaggio da inviare, esso può essere visto come una successione di simboli, che siano caratteri dell'alfabeto, spazi vuoti o punteggiatura. Nella fase di cifratura, ogni simbolo viene sostituito con un altro simbolo che lo rappresenta, utilizzando lo stesso insieme di simboli originale. La chiave di cifratura viene quindi applicata a ogni simbolo fino a ottenere un testo cifrato. Se consideriamo σ il simbolo da cifrare e π la chiave di cifratura, il simbolo cifrato ρ può essere rappresentato come $\rho = \pi(\sigma)$.

Semplificando ulteriormente, si prenda in esempio quanto segue:

ABCDEFGHIJKLMNOPQRSTUVWXYZ è l'alfabeto del testo in chiaro,
GHIDZXLMBNOPFQERSTCUVWAJYK è l'alfabeto del testo da cifrare.

Si procede sostituendo ogni lettera dell'alfabeto in chiaro (rosso) con il carattere corrispondente dell'alfabeto cifrato (sotto, in blu). Nel caso in cui si voglia cifrare, per esempio, la parola CRITTOGRAFIA, si otterrà il testo cifrato ITBUUELTGXBG.

È intuibile che ogni permutazione dell'alfabeto di 26 lettere sia una chiave valida per questo sistema. Nonostante lo spazio delle chiavi ottenuto è molto vasto, e attacchi di forza bruta[3] risulterebbero inefficienti, strumenti statistici, come la distribuzione e l'analisi delle frequenze, permettono di risalire al testo originale con facilità [34].

Uno dei più antichi algoritmi crittografici di cui si ha traccia è il cifrario di Cesare, utilizzato da Giulio Cesare per conferire segretezza ai suoi messaggi. Questo cifrario segue la stessa logica di quello a sostituzione, con la differenza che non prevede una permutazione dell'alfabeto. Infatti, la chiave consiste nella rotazione di un numero prefissato di posti delle lettere dell'alfabeto. Se si fissa tale numero a 2, si ottengono le corrispondenze A→C, B→D, C→E, ecc.

Altri cifrari a sostituzione sono il Cifrario di Vigenère (1586), il Quadrato di Playfair (1854), il Cifrario di Hill (1929) e il Cifrario di Vernam (1917), noto anche come One-Time Pad.

1.3.2 Cifrari perfetti

Esistono algoritmi crittografici in grado di garantire un'assoluta segretezza dell'informazione. Tuttavia, il loro impiego risulta impraticabile a causa dei costi proibitivi, sollevando dubbi circa la loro reale utilità. L'utilizzo di approssimazioni efficienti di tali cifrari è limitata a comunicazioni sporadiche caratterizzate da estrema riservatezza.

[3] Attacco in cui il malintenzionato cerca di ottenere l'accesso a un sistema, provando ripetitivamente molteplici combinazioni di password o chiavi fino a trovare quella corretta.

D'altra parte, ci sono algoritmi crittografici che, pur non essendo inviolabili, offrono un equilibrio adeguato tra sicurezza ed economicità. Questi sono impiegati quotidianamente per gestire comunicazioni su larga scala.

La definizione formale di cifrario perfetto è stata data da Claude Shannon. Egli affermava che:

Definizione (Cifrario perfetto)**.** Sia M una variabile aleatoria[4] che assume valori nello spazio dei messaggi $\mathcal{M}$. Allo stesso modo, sia C una variabile aleatoria che assume valori nello spazio dei crittogrammi $\mathcal{C}$, ossia i risultati della cifratura. Un cifrario si dice perfetto se, per ogni $m \in \mathcal{M}$ e per ogni $c \in \mathcal{C}$, vale la relazione

$$P(M = m|C = c) = P(M = m).$$

In altre parole, un cifrario è perfetto quando la probabilità che il mittente invii un possibile messaggio è uguale alla probabilità che il messaggio inviato sia proprio m, sapendo che c è stato il crittogramma trasmesso sul canale di comunicazione.

Si supponga che il cifrario non sia perfetto e che la probabilità di inviare un messaggio non sia mai nulla, ossia $P(M = m) = p$ con $0 < p < 1$. In base alla definizione precedente, queste condizioni implicano che $P(M = m|C = c) \neq P(M = m)$. Si possono verificare i tre seguenti casi:

1. Se $P(M = m|C = c) = 0$, osservando c si deduce che il messaggio spedito non è m.
2. Se $P(M = m|C = c) = 1$, il crittogramma è proprio m.
3. Negli altri casi si raffina la conoscenza sul possibile messaggio spedito.

Sulla base di quanto esposto, la conoscenza del crittoanalista rimane invariata solo nel caso in cui si utilizzi un cifrario perfetto. Se si effettuano delle considerazioni sul numero di chiavi di un cifrario perfetto, è possibile dimostrare il seguente enunciato:

Teorema (Enunciato di Shannon)**.** In un cifrario perfetto il numero delle chiavi deve essere maggiore o uguale al numero dei messaggi possibili.

Cifrario di Vernam

Il cifrario di Vernam viene spesso chiamato One-time Pad. Tale denominazione fa riferimento alla natura della chiave annotata su di un blocco per appunti o *pad*, e utilizzata al più una volta. Una chiave riutilizzata esporrebbe il messaggio cifrato ad attacchi simili a quelli visti per i cifrari a sostituzione.

[4] Per la definizione di variabile aleatoria si può fare riferimento all'Appendice A.

Shannon riuscì a formalizzare, mediante un modello matematico, che il cifrario di Vernam possedeva tutte le caratteristiche per considerarlo perfetto, prendendo in esame sequenze binarie arbitrariamente lunghe.

Considerando il messaggio da inviare come una sequenza binaria, il funzionamento generale si descrive come segue:

1. Una chiave segreta è generata come una sequenza di bit completamente casuale nota solo al mittente e al destinatario. Tale chiave è lunga almeno quanto il messaggio da trasmettere.

2. Ogni bit del crittogramma si ottiene come XOR[5] tra i bit del messaggio e quelli della chiave.

3. Per decifrare il messaggio, ogni suo bit si ottiene ponendo a XOR i bit del crittogramma con quelli della chiave.

Se si denota con n la lunghezza del messaggio, $m = m_1 m_2 \cdots m_n$ la sequenza binaria da inviare e $k = k_1 k_2 \cdots k_n$ quella della chiave, otteniamo che il crittogramma risulta essere $c = c_1 c_2 \cdots c_n$, dove

$$c_i = m_i \oplus k_i \quad \text{per} \quad i = 1, 2, \ldots, n.$$

Per dimostrare che effettivamente One-time Pad sia un cifrario perfetto, è necessario fare le seguenti ipotesi:

1. Tutti i messaggi sono di lunghezza n.

2. Tutte le sequenze di n bit sono potenziali messaggi.

Occorre provare quanto già affermato da Shannon, ovvero che $P(M = m|C = c) = P(M = m)$. Applicando la definizione della probabilità condizionale (A.1), possiamo riscrivere l'equivalenza:

$$P(M = m|C = c) = \frac{P(M = m, C = c)}{P(C = c)}.$$

Il termine $P(M = m|C = c)$ corrisponde alla probabilità che il messaggio generato m sia stato cifrato con il crittogramma c. Eseguire un'operazione di XOR tra messaggio e chiave, genera un crittogramma diverso ogni qual volta modifichiamo la chiave. Inoltre, si ricordi che ogni chiave può essere generata con probabilità $(1/2)^n$.

Definita la lunghezza n del messaggio, la probabilità dell'evento $\{C = c\}$ è costante e indipendente da m. Si deduce che gli eventi $\{C = c\}$ e $\{M = m\}$ sono indipendenti tra

[5] L'operazione XOR (chiamata a volte OR esclusivo, da eXclusive OR) è un'operazione logica binaria che restituisce vero quando i suoi operandi sono diversi e falso quando sono uguali. In algebra booleana, viene indicata dal simbolo $\oplus$.

loro. Questo significa che (A.2):

$$P(M = m, C = c) = P(M = m) \cdot P(C = c).$$

In considerazione di quanto detto, possiamo affermare

$$P(M = m|C = c) = P(M = m),$$

ossia che il cifrario è perfetto.

In presenza di questo tipo di cifratura, un crittoanalista che entra in possesso di un crittogramma, non è in grado di decifrarlo senza l'ausilio di una chiave. Per fare un esempio, se il crittoanalista entra in possesso di un crittogramma di 4 lettere corrispondente a m, la probabilità di indovinarlo è pari a quella che sia stato scelto m tra tutti i messaggi di 4 lettere presenti in $\mathcal{M}$.

Il sistema presenta il difetto di non poter essere utilizzato per messaggi particolarmente lunghi. Ne consegue che la chiave debba avere dimensioni notevoli e che debba essere scambiata ogni volta tramite un canale sicuro, oltre a essere generata in modo completamente casuale.

Capitolo 2

Crittografia moderna

La crittografia moderna costituisce il fondamento della sicurezza digitale nell'era contemporanea, offrendo un insieme di algoritmi e protocolli avanzati mirati a preservare la confidenzialità, l'integrità e l'autenticità delle informazioni scambiate attraverso reti e sistemi informatici.

Questo capitolo si propone di evidenziare i risultati emersi dall'analisi dei metodi crittografici impiegati in passato, mettendo in luce le vulnerabilità e le carenze che la crittografia attuale si impegna a superare.

Il concetto di casualità assume un ruolo centrale, in quanto fondamentale nella generazione di chiavi e nell'efficacia degli algoritmi crittografici. L'utilizzo di numeri casuali e funzioni pseudocasuali diventa cruciale per garantire l'imprevedibilità necessaria a proteggere le comunicazioni e le informazioni sensibili.

In questo capitolo, saranno esplorati algoritmi simmetrici avanzati, come l'Advanced Encryption Standard (AES), e algoritmi asimmetrici come il Rivest-Shamir-Adleman (RSA) e le curve ellittiche (ECC).

La crescente potenza computazionale e l'emergere di nuove minacce hanno posto sfide significative alla sicurezza della crittografia moderna. La vulnerabilità di protocolli e algoritmi agli attacchi di tipo quantistico, come l'algoritmo di Shor, sottolinea la necessità di riconsiderare le fondamenta della sicurezza digitale.

2.1 La casualità

L'obiettivo della crittografia è garantire la sicurezza delle comunicazioni, e a tal fine è cruciale che il messaggio trasmesso attraverso il canale di comunicazione non sia prevedibile. A tal proposito, basti pensare alla chiave generata da One-Time Pad che deve essere composta da bit completamente casuali.

Il concetto di casualità, intuitivo e applicabile per un essere senziente, risulta complesso per una macchina. Infatti, una macchina non è in grado di generare autonomamente un bit casuale come se fosse il risultato del lancio di una moneta. In pratica, quando si fa

riferimento a una sequenza casuale di bit, si intende il risultato di un algoritmo o di processo in grado di produrre una serie di bit casuali.

A prima vista, la sequenza di 8 bit `11001011` potrebbe sembrare più "casuale" rispetto a `00000000`. Tuttavia, entrambe hanno la stessa probabilità di essere generate. La differenza tra di loro sta nella presenza di uno schema più evidente nella seconda rispetto alla prima.

Inoltre, si può notare come l'alternanza tra uni e zeri in maniera relativamente uniforme dà l'illusione di casualità; in questo caso la stringa contiene tre zeri e cinque uni. La probabilità di generare una sequenza di 8 bit con la stessa quantità di zeri e uni è comunque molto più alta rispetto a una sequenza di soli zeri o soli uni. Nonostante ciò, si potrebbe argomentare che `11001011` e le sue permutazioni appaiano molto più casuali di `00000000`.

In crittografia, è fondamentale distinguere tra ciò che sembra casuale e ciò che lo è effettivamente. Talvolta, all'assenza di casualità viene attribuita una connotazione di insicurezza.

2.1.1 Entropia

Il risultato di un processo casuale è determinato dalla sua *distribuzione di probabilità*, che elenca i possibili risultati del processo insieme alle relative probabilità. La somma di tutte le probabilità della distribuzione è 1. Ad esempio, considerando il lancio di una moneta equa, la sua distribuzione sarà $1/2$ per il risultato "testa" e $1/2$ per quello "croce".

L'entropia, misura dell'incertezza o del disordine di un sistema, rappresenta la quantità di sorpresa all'interno di un processo: tanto più è elevata, minore è la certezza del risultato. L'entropia H di una distribuzione di probabilità può essere calcolata come la somma negativa delle probabilità moltiplicate per il loro logaritmo in base 2:

$$H = -\sum_i p_i \cdot \log_2(p_i).$$

Applicando questo calcolo al lancio di una moneta regolare, l'entropia risulta essere:

$$-\frac{1}{2} \cdot \log_2\left(\frac{1}{2}\right) - \frac{1}{2} \cdot \log_2\left(\frac{1}{2}\right) = 1.$$

Considerando ora una moneta truccata con probabilità di $1/4$ per testa e $3/4$ per croce, l'entropia è:

$$-\frac{1}{4} \cdot \log_2\left(\frac{1}{4}\right) - \frac{3}{4} \cdot \log_2\left(\frac{3}{4}\right) \approx 0.81.$$

In questo caso, l'entropia risulta inferiore poiché si ha maggiore certezza che il risultato sia croce anziché testa.

2.1.2 Generatori di numeri casuali

I sistemi crittografici richiedono una fonte affidabile di casualità, ovvero un componente in grado di generare un bit casuale quando necessario. Per raggiungere questo obiettivo sono fondamentali due elementi:

1. Una fonte di incertezza o entropia, fornita da un generatore di numeri casuali (RNG, *random number generator*).

2. Un algoritmo crittografico capace di produrre bit casuali di alta qualità a partire dalla sorgente di entropia, utilizzando i cosiddetti generatori di numeri pseudocasuali (PRNG, *pseudo-random number generator*).

In generale, gli algoritmi non sono in grado di generare casualità autonomamente. Per raggiungere questo scopo, fanno affidamento sui RNG, componenti software o hardware influenzati dall'entropia ambientale. Quest'ultima può derivare dalla misurazione di fattori come la temperatura, l'elettricità statica o altri elementi ambientali, ma anche da componenti interni della macchina come l'attività del disco, della rete, eccetera.

D'altra parte, i PRNG generano i loro bit casuali a partire da pochi bit che sono effettivamente frutto del caso. I generatori pseudocasuali riproducono una sequenza che *sembra* casuale, anche se questa è l'esito di un processo deterministico.

Una distinzione importante è che i RNG non garantiscono la produzione di sequenze ad alta entropia, mentre questo è possibile con i PRNG.

Funzionamento di un PRNG

Un PRNG riceve regolarmente un insieme di bit casuali da un RNG, e li utilizza per aggiornare il cosiddetto *pool* di entropia. Proprio come l'ambiente circostante è la fonte di entropia per un generatore di numeri casuali, così lo è il pool di entropia per quella di un PRNG.

Per dare vita a bit pseudocasuali, il generatore procede a eseguire un algoritmo deterministico che espande i bit in ingresso in sequenze molto più lunghe. Il pool di entropia è fondamentale per garantire che non vengano mai forniti gli stessi input alla procedura.

In breve, un generatore di numeri pseudocasuali è capace di eseguire tre operazioni:

- `init()`, la fase con cui viene inizializzato il pool di entropia.

- `refresh(R)`, la procedura che aggiorna il pool con dati `R` provenienti da un RNG.

- `next(N)`, la funzione che restituisce un numero `N` di bit pseudocasuali e aggiorna nuovamente il pool.

Dal punto di vista della sicurezza, un PRNG deve impedire di risalire con facilità ai parametri che hanno portato alla generazione di una specifica sequenza di bit, evitando previsioni sugli elementi futuri. I generatori lineari e polinomiali non riescono a garantire questa caratteristica, motivo per cui si può ricorrere a funzioni one-way.

2.2 Cifrari a blocchi

Durante la Guerra Fredda, Stati Uniti d'America e Unione Sovietica svilupparono rispettivamente il cifrario americano Data Encryption Standard (DES), utilizzato fino al 2005, e il cifrario sovietico GOST. Entrambi, insieme all'Advanced Encryption Standard (AES), successore di DES, sono tutti classificati come cifrari a blocchi (o *block chiper*). Questi cifrari operano attraverso algoritmi che combinano blocchi di dati seguendo una specifica modalità operativa.

Un cifrario a blocchi è costituito da un algoritmo di cifratura e uno di decifrazione. In dettaglio, si ha:

- Un algoritmo di cifratura E che, dati una chiave K e un blocco di testo in chiaro P come input, restituisce un blocco cifrato C. La procedura può essere sintetizzata come $C = E(K, P)$.
- Un algoritmo di decifrazione D, che svolge l'operazione inversa rispetto a E. Viene rappresentato come $P = D(K, C)$.

2.2.1 Caratteristiche

Sicurezza

Come discusso in precedenza nella Sezione 2.1, la sicurezza di una sequenza binaria è misurata come la sua capacità di apparire casuale. Per analogia, ci si aspetta che il risultato di un blocco cifrato sicuro sia una permutazione pseudocasuale del blocco in chiaro.

Un malintenzionato, privo della conoscenza della chiave, non dovrebbe mai essere in grado di dedurre l'output della procedura basandosi sul suo input. In altre parole, fintanto che K rimane segreta, l'effettivo funzionamento di $E(K, P)$ rimarrebbe sconosciuto per ogni possibile P.

Dimensione del blocco

I cifrari a blocchi sono caratterizzati da due parametri: la dimensione del blocco e la dimensione della chiave, entrambi influenti sulla sicurezza del cifrario. Questi valori possono variare, ad esempio, dai 64 bit del DES ai 128 bit del più recente AES.

La scelta di queste dimensioni è dettata dal fatto che gli algoritmi di cifratura a blocchi operano, come suggerisce il nome, su blocchi di dati e non su singoli messaggi. Considerando blocchi di 128 bit, un messaggio di soli 16 bit deve essere prima convertito in un blocco altrettanto grande e poi elaborato. Dimensioni più grandi potrebbero compromettere l'usabilità introducendo un eccessivo overhead[1]. Attualmente, i processori sono ottimizzati per operare efficientemente con blocchi di 128 bit [1].

2.2.2 Funzionamento

L'algoritmo utilizzato per la generazione di cifrari a blocchi non è un'entità monolitica; solitamente, coinvolge una serie di iterazioni che, nel loro insieme, conferiscono complessità e robustezza al risultato finale.

Inoltre, esistono due modalità operative per definire un *round*: le reti a sostituzione e permutazione e quelle di Feistel. Le prime sono adoperate in DES, mentre le seconde in AES.

Il round

Un round non è altro che una trasformazione elementare di facile implementazione, che viene iterata più volte per generare il blocco cifrato.

Se cifrassimo un messaggio in tre round con un cifrario a blocchi, otterremmo che il blocco cifrato risulta essere $C = R_3(R_2(R_1(P)))$. Sono indicati con R_1, R_2 e R_3 i rispettivi round e con P il testo in chiaro. Ogni round è caratterizzato anche dal suo inverso, consentendo il ripristino del messaggio originale. Riprendendo l'esempio, otteniamo $P = R_1^{-1}(R_2^{-1}(R_3^{-1}(C)))$, dove R_1^{-1} rappresenta l'inverso di R_1 e così via.

Le procedure utilizzate in un round sono sostanzialmente equivalenti a livello algoritmico e differiscono per un valore noto come *round key*. Due round si comporteranno diversamente anche se ricevono lo stesso input, a condizione che le rispettive chiavi siano diverse. Ogni chiave utilizzata in un round è derivata dalla chiave K originale, e ogni round key deve sempre differire da quella precedente per evitare la generazione dello stesso blocco cifrato.

Reti a sostituzione e permutazione

Nel campo della crittografia, si fa spesso riferimento ai concetti di *confusione* e *diffusione*. Con il termine confusione ci si riferisce al processo in cui il testo in chiaro e la chiave subiscono complesse trasformazioni per evitarne la separazione a partire dal crittogramma. Per diffusione, invece, si intende l'effetto di distribuire uniformemente le trasformazioni su

[1] In informatica, termine utilizzato per definire le risorse aggiuntive richieste oltre a quelle strettamente necessarie per ottenere un scopo specifico.

tutti i bit dell'input. I cifrari a blocchi implementano questi principi attraverso sostituzioni e permutazioni, che sono messi in pratica dalle reti a sostituzione e permutazione.

Le sostituzioni sono realizzate tramite le cosiddette *S-box*, delle lookup table[2] che mappano blocchi di 4 o 8 bit. Diversi algoritmi fanno uso di queste S-box, ciascuna progettata in modo unico, ma tutte con l'obiettivo di resistere alla crittanalisi.

Le permutazioni possono essere effettuate cambiando l'ordine dei bit oppure mediante l'utilizzo dell'algebra lineare e delle matrici di moltiplicazione. Un semplice scambio di bit, infatti, non garantirebbe una diffusione adeguata.

Reti di Feistel

Nel 1970, un ingegnere della IBM propose un cifrario a blocchi che operava come illustrato di seguito (Figura 2.1):

1. Un blocco di 64 bit viene diviso a metà, ottenendo R e L.
2. Si modifica L come $L = L \oplus F(R)$, tenendo presente che F è un round di sostituzione e permutazione.
3. Si invertono i valori di R e L.
4. Si ripete per 15 volte dal secondo passaggio.
5. Si ricostruisce il blocco di 64 bit unendo R e L.

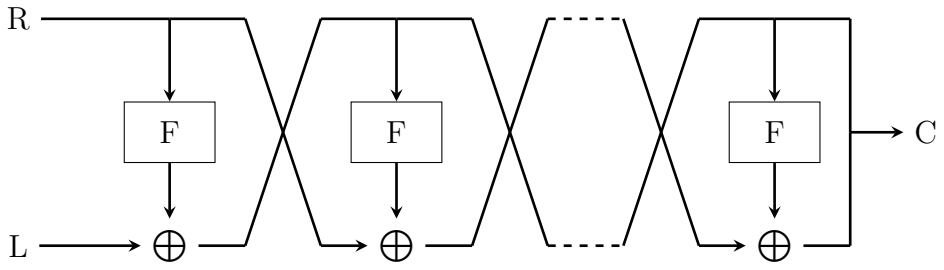


Figura 2.1: Funzionamento di una rete di Feistel.

In una rete di Feistel, la funzione F può essere sia una permutazione pseudocasuale sia una funzione pseudocasuale. Nel primo caso si ottengono due output distinti a partire da due input distinti; nel secondo si ottiene $F(X) = F(Y)$, per qualsiasi valore di X e Y. L'uso di una rispetto all'altra è indifferente, purché entrambe siano crittograficamente robuste.

L'algoritmo DES utilizza 16 round, mentre GOST ne utilizza 32.

[2] Una lookup table (tabella di ricerca) è una struttura dati che associa dei valori di input con altrettanti valori di output. In altre parole, è un meccanismo che consente di recuperare rapidamente informazioni pre-calcolate, migliorando l'efficienza di alcune operazioni.

2.3 Sistemi di cifratura avanzata

Nel vasto panorama della crittografia moderna, emergono tre protagonisti distintivi: AES, RSA e le Curve Ellittiche. Questi algoritmi, ognuno con le sue peculiarità e applicazioni, occupano posizioni di rilievo nella sicurezza informatica contemporanea.

Questa sezione affronta brevemente il funzionamento e gli aspetti alla base di questi protocolli, caratterizzati da tecniche più sofisticate rispetto a quanto avveniva per la crittografia classica.

2.3.1 AES

AES, acronimo di Advanced Encryption Standard, rappresenta uno dei cifrari più diffusi al mondo, divenuto lo standard negli Stati Uniti d'America dal 2000. Si tratta di un algoritmo di cifratura a blocchi e chiave simmetrica particolarmente robusto, che ha gradualmente sostituito l'ormai vulnerabile DES.

AES elabora dati in blocchi di 128 bit e utilizza chiavi di 128, 192 o 256 bit per il processo di cifratura. La dimensione più comune è di 128 bit, poiché consente cifrature veloci senza introdurre differenze significative rispetto all'utilizzo di chiavi di dimensione maggiore.

Funzionamento

Il protocollo gestisce il messaggio o una sua parte attraverso un array di 16 byte, che può essere interpretato logicamente come una matrice bidimensionale chiamata *stato* (Figura 2.2). AES modifica i byte, le colonne e le righe di questa matrice per produrre il testo cifrato.

s_0	s_1	s_2	s_3
s_4	s_5	s_6	s_7
s_8	s_9	s_{10}	s_{11}
s_{12}	s_{13}	s_{14}	s_{15}

Figura 2.2: Stato di 16 byte.

Per apportare modifiche allo stato, AES utilizza una rete a sostituzione e permutazione composta da 10, 12 o 14 round, a seconda della dimensione della chiave (rispettivamente 128, 192 o 256 bit). Ogni round comprende le seguenti fasi:

1. Viene eseguito uno XOR tra la round key e lo stato corrente.
2. Si sostituiscono i valori dello stato con quelli presenti in una S-box.

3. Si sposta l'i-esima riga di i posizioni, per i che va da 0 a 3.

4. Si applica la stessa trasformazione lineare sulle quattro colonne dello stato.

Ciascuna di queste fasi contribuisce in modo essenziale alla sicurezza del risultato prodotto da AES.

Sicurezza

In sintesi, AES è considerato sicuro poiché tutti i bit di output dipendono in modo complesso e pseudocasuale da tutti i bit di input. Per ottenere questo risultato, i progettisti di AES hanno selezionato attentamente ciascun componente, scegliendo la seconda fase per la sua non linearità ottimale e la terza fase per le sue proprietà di massima diffusione. È stato dimostrato che questa scelta è necessaria per garantire protezione contro varie classi di attacco. Non c'è alcuna evidenza che AES sia completamente immune, ma non vi è motivo di dubitare della sua sicurezza [19].

2.3.2 RSA

Il sistema crittografico Rivest-Shamir-Adleman è stato pubblicato per la prima volta nel 1977 e rappresenta la prima implementazione di cifratura a chiave pubblica. Un anno prima, Diffie e Hellman avevano introdotto il concetto di crittografia a chiave pubblica, ma senza proporre un'implementazione concreta.

RSA fa uso di uno stratagemma matematico noto come *trapdoor permutation*, una funzione che trasforma un numero x in un altro y all'interno dello stesso intervallo. Questo consente di ricavare facilmente il valore di y conoscendo x e la chiave pubblica; d'altro canto, risulterebbe impossibile il contrario, a meno che non si conosca la chiave privata, ovvero la trapdoor.

Aspetti matematici

RSA interpreta i messaggi da cifrare come numeri, e il processo di cifratura coinvolge moltiplicazioni tra numeri molto grandi.

Il testo in chiaro è interpretato come un numero intero positivo compreso tra 1 e $n-1$, dove n è un numero molto grande chiamato modulo. In particolare, viene scelto un numero che sia minore di n e diverso da 0, e che non abbia fattori in comune con n; quindi, si desidera che sia *coprimo* con n. I numeri così ottenuti formano l'insieme Z_n^*.

Per esempio, se consideriamo Z_6^*, otteniamo i numeri 1 e 5; Z_9^* sarà composto dai numeri 1, 2, 4, 5, 7, 8; mentre Z_{11}^* conterrà i numeri 1, 2, 3, 4, 5, 6, 7, 8, 9 e 10. Se n è un numero primo, allora $Z_n^* = \{1, \ldots, n-1\}$.

La funzione toziente di Eulero, indicata con $\varphi(n)$, definisce il numero di elementi di Z_n^*, ossia la sua cardinalità.

RSA basa il suo funzionamento sulla scelta di un n che sia il prodotto di due numeri primi abbastanza grandi. Questo prodotto è spesso indicato come $n = pq$. L'insieme corrispondente Z_n^* contiene $\varphi(n) = (p-1)(q-1)$ elementi[3]. Sviluppando questa affermazione, si può arrivare a dire che Z_n^* contiene tutti i numeri da 1 a $n-1$, escludendone esattamente $p+q$, e questi numeri sono potenzialmente validi per RSA.

Trapdoor permutation

Il concetto di trapdoor permutation è fondamentale per gli algoritmi di cifratura basati su RSA.

Definito un modulo n e un numero e chiamato esponente pubblico, si procede a trasformare un numero x di Z_n^* come $y = x^e$. In questo contesto, il numero n e l'esponente e costituiscono la chiave di RSA.

Per ottenere il valore di x da y, si utilizza un altro numero d:

$$y^d \bmod n = (x^e)^d \bmod n = x^{ed} \bmod n = x.$$

Il numero d è la trapdoor, il valore segreto che costituisce la chiave privata di RSA. Viene chiamato, appunto, esponente segreto.

L'esponente d è scelto in modo tale che e moltiplicato per d sia esattamente 1. Più precisamente, deve essere valida l'equivalenza $ed = 1 \bmod \varphi(n)$, per ottenere $x^{ed} = x$ e decifrare il messaggio correttamente.

La sicurezza di RSA si basa sulla difficoltà di trovare $\varphi(n)$ noto n, dal momento che l'esponente segreto può essere ottenuto facilmente ed e si ottiene dalla sua inversa. Per questo motivo, sono tenuti segreti anche i valori di p e q.

Scambio di un messaggio

In RSA, ogni messaggio è interpretato come un numero intero m. Per un uso corretto del cifrario, è necessario che m sia minore di n, e questo è reso possibile dividendo in più parti un messaggio troppo lungo.

Indicando con c il messaggio cifrato e con m il messaggio non cifrato, e considerando gli esponenti descritti precedentemente, si possono sfruttare le seguenti equivalenze per procedere allo scambio di un messaggio:

$$c = m^e \bmod n,$$
$$m = c^d \bmod n.$$

[3] Si applica la moltiplicatività di φ. L'argomento non è approfondito perché non strettamente necessario alla trattazione.

2.3.3 Curve Ellittiche

L'introduzione delle curve ellittiche nella crittografia, avvenuta nel 1985, ha rivoluzionato l'approccio alla cifratura a chiave pubblica. Note come ECC (Elliptic Curve cryptography), costituiscono valide alternative a RSA o Diffie-Hellman, pur mantenendo la loro distintiva complessità. Concettualmente, esistono diverse tipologie di curve ellittiche che variano in efficienza, complessità e sicurezza.

La loro adozione è stata graduale nel corso degli anni, fino a diventare uno standard valido e riconosciuto. Le curve ellittiche permettono di eseguire le più comuni operazioni di crittografia e scambio di chiavi in maniera più veloce rispetto alle controparti a chiave pubblica più "classiche".

Lo scopo di questa sottosezione è fornire un'introspettiva sommaria sulle caratteristiche delle curve ellittiche e sulle loro applicazioni. Per comprendere ogni concetto di questa sottosezione, è consigliabile leggere anche i contenuti riportati in Appendice B.

Definizione di una curva ellittica

Una curva ellittica è una rappresentazione grafica su un piano cartesiano, composta da una serie di punti. La curva specifica può essere individuata mediante la sua equazione, e tutti i punti di coordinate (x, y) che si trovano sulla curva soddisfano l'equazione, indipendentemente dal tipo di curva considerato.

In crittografia, una curva ellittica è tipicamente della forma di Weierstrass:

$$y^2 = x^3 + ax + b,$$

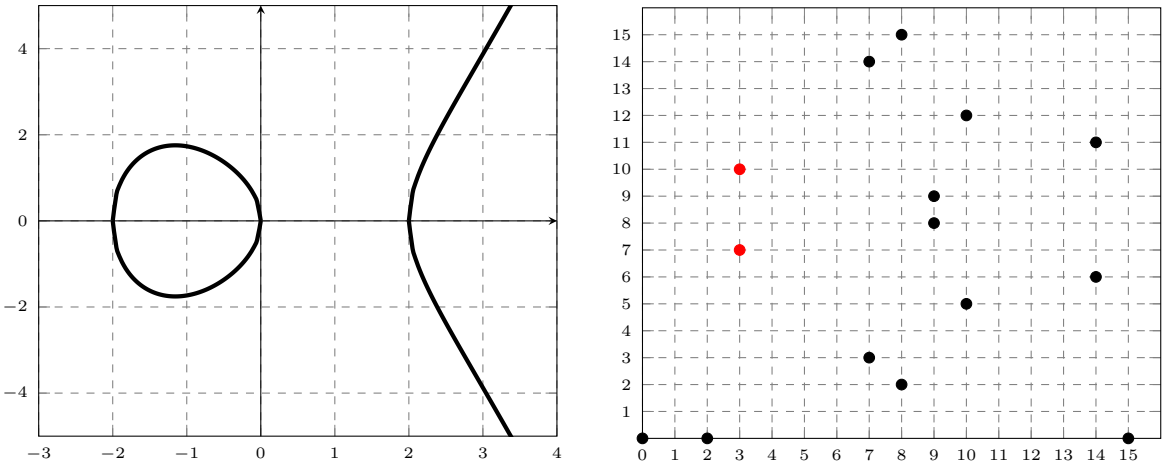
dove i valori di a e b definiscono la forma della curva. Nella Figura 2.3a è illustrato un esempio di curva ellittica.

Campi finiti

Le curve ellittiche utilizzate in crittografia non assomigliano a quelle mostrate in Figura 2.3a, ma piuttosto a quella riportata in Figura 2.3b. Per ottenere questo risultato, è necessario limitare il campo dei numeri utilizzati, in modo simile a quanto avviene per RSA.

Considerando l'esempio in Figura 2.3, entrambe le curve sono frutto della stessa equazione, ma mostrano i loro punti sulla base di due insiemi numerici differenti. La prima, a sinistra, utilizza numeri reali, mentre la seconda utilizza solo numeri interi, nello specifico appartenenti all'insieme dei numeri interi modulo 17, indicato con Z_{17}. Il numero scelto in questo esempio è volutamente piccolo per semplicità di calcolo.

Gli elementi delle coordinate dei punti della curva in Figura 2.3b sono numeri interi presenti in Z_{17}. Inoltre, questi punti soddisfano l'equazione della curva.



(a) Una curva di equazione $y^2 = x^3 - 4x$.

(b) La stessa curva su Z_{17}.

Figura 2.3: Esempio della medesima curva ellittica.

Si noti che tale insieme include tutti i numeri da 0 a 16, essendo 17 un numero primo. Qualora si ottenga un numero al di fuori di Z_{17}, è necessario applicare l'operazione modulo per trovare il valore corretto.

Per esempio, se consideriamo $x = 3$, otteniamo l'equazione $y^2 = 27 - 12 = 15$. Questa equazione ammette due soluzioni in Z_{17}: 7 e 10. Emerge chiaramente che entrambe soddisfano l'equazione in quanto $7^2 \bmod 17 = 15$ e $10^2 \bmod 17 = 15$, risultato presente in Z_{17}. Questi punti appartengono alla curva e sono evidenziati in rosso nella Figura 2.3b.

Scambio di chiavi

È possibile effettuare uno scambio di chiavi tra due soggetti basandosi sull'uso delle curve ellittiche. Di seguito, viene presentato un esempio in cui Alice e Bob desiderano scambiarsi una chiave.

1. Alice e Bob concordano pubblicamente su un campo finito e su una curva ellittica definita su questo campo. Successivamente, scelgono un punto B sulla curva con n molto grande, dove n rappresenta il più piccolo intero positivo tale che soddisfa $nB = O$.

2. Alice seleziona un intero positivo casuale $n_A < n$ come propria chiave privata e genera la chiave pubblica da inviare a Bob come $P_A = n_A B$.

3. Bob procede a generare e inviare la sua chiave pubblica P_B allo stesso modo, ovvero $P_B = n_B B$.

4. Alice calcola $n_A P_B = S$, utilizzando la sua chiave privata. S è un punto sulla curva.

5. Bob, allo stesso modo, calcola $n_B P_A = S$.

6. Alice e Bob condividono lo stesso S, determinato dalle loro scelte casuali. Per trasformarlo in una chiave segreta k, esso viene convertito in un intero, ad esempio ponendo $k = x_S \bmod 2^{256}$ (ovvero considerando gli ultimi 256 bit dell'ascissa di S).

Un crittoanalista, che intercetta i valori di P_A e P_B e conosce sia i parametri della curva scelta sia il punto B, non è in grado di calcolare il valore di S. Tale difficoltà deriva dal fatto che per calcolare una delle due chiavi private è necessario risolvere il problema del logaritmo discreto[4], la cui soluzione risulta complessa data l'entità dei valori coinvolti.

Sicurezza

Nonostante ciò, il protocollo appena visto rimane vulnerabile agli attacchi di tipo man-in-the-middle. Inoltre, è importante sottolineare che la complessità delle curve ellittiche, insieme alla presenza di molteplici parametri di configurazione e alla scelta di valori non propriamente casuali, può creare vulnerabilità nella sicurezza implementativa.

Un esempio noto riguarda Sony, che utilizzò il protocollo ECDSA (Elliptic Curve Digital Signature Algorithm) per firmare i software della console PlayStation 3, scegliendo però un parametro statico anziché casuale. Questa scelta condusse un gruppo di crittoanalisti a individuare la chiave privata di ECDSA, compromettendo la sua sicurezza [6].

[4] Si faccia riferimento alla Sezione 5.2 e all'Appendice B per ulteriori informazioni.

Capitolo 3

Fondamenti di Meccanica Quantistica

La meccanica quantistica rappresenta il ramo della fisica dedicato all'analisi della natura a livelli inferiori o uguali alla scala degli atomi, nell'ordine di grandezza dei 10^{-10} metri. Essa costituisce la base per l'intera fisica quantistica, inclusa la chimica quantistica, la teoria quantistica dei campi, la tecnologia quantistica e la scienza delle informazioni quantistiche.

La fisica classica descrive efficacemente molti aspetti della natura su scala macroscopica, ma mostra limiti nella descrizione di fenomeni su scala atomica e subatomica. Molte teorie della fisica classica possono essere derivate come approssimazioni della meccanica quantistica applicate al mondo macroscopico.

Fondamentalmente, la meccanica quantistica descrive la radiazione e la materia sia come fenomeni ondulatori che come entità particellari. Al contrario, la meccanica classica tratta la luce solo come un'onda e l'elettrone solo come una particella. La relazione tra natura ondulatoria e corpuscolare è enunciata nel principio di complementarità e formalizzata nel principio di indeterminazione di Heisenberg.

Il pioniere nello studio di questi fenomeni fu il fisico tedesco Max Planck, che iniziò le sue ricerche all'inizio del XX secolo. Planck sviluppò la teoria dei quanti, introducendo il concetto fondamentale di quantizzazione dell'energia. Secondo la sua prospettiva, l'energia manifesta una natura quantizzata simile alle cariche elettriche. L'emissione di questa energia avviene in quantità discrete, rappresentate da multipli interi di un quanto di energia universale.

Questi primi tentativi di comprendere i fenomeni microscopici, che poi si evolsero nella cosiddetta *vecchia teoria dei quanti*, portarono al pieno sviluppo della meccanica quantistica nella metà degli anni Venti grazie a Niels Bohr, Erwin Schrödinger, Werner Heisenberg, Max Born, Paul Dirac e altri.

La teoria moderna ha definito vari formalismi matematici appositamente sviluppati. In uno di essi, un'entità matematica chiamata *funzione d'onda* fornisce informazioni, sotto forma di ampiezze di probabilità, su quali misure di energia, momento e altre proprietà fisiche si possono ricavare da una particella.

Lo studio della meccanica quantistica ha condotto allo sviluppo dei computer quantistici, che offrono nuovi paradigmi rispetto ai calcolatori classici. Basandosi sui quantum bit e offrendo una potenza di calcolo eccezionale, i computer quantistici sono impiegati nella crittografia quantistica per implementare nuovi protocolli e modalità di crittografia.

Questo capitolo mira a introdurre aspetti fondamentali e notazioni della meccanica quantistica, determinanti per comprendere i concetti che saranno trattati nel capitolo successivo, cuore di questa tesi.

3.1 Algebra lineare

Il proposito di questa sezione è introdurre e descrivere alcuni formalismi della meccanica quantistica, fornendo definizioni, notazioni e concetti matematici. Per comprendere appieno questa sezione, è necessaria una conoscenza di base di alcuni strumenti matematici come limiti e integrali, unita a una familiarità con il calcolo algebrico. Il linguaggio scelto è volutamente discorsivo, per rendere più agevole la lettura.

3.1.1 Vettori

Uno **spazio vettoriale** è costituito da un insieme di *vettori* $(|\alpha\rangle, |\beta\rangle, |\gamma\rangle, \dots)$[1] e un insieme di scalari[2] con cui è possibile effettuare due operazioni: la somma di vettori e la moltiplicazione per uno scalare.

La **somma** di due vettori $|\alpha\rangle$ e $|\beta\rangle$ qualsiasi è indicata come

$$|\alpha\rangle + |\beta\rangle = |\gamma\rangle.$$

La somma tra vettori è **commutativa**

$$|\alpha\rangle + |\beta\rangle = |\beta\rangle + |\alpha\rangle$$

e **associativa**

$$|\alpha\rangle + (|\beta\rangle + |\gamma\rangle) = (|\alpha\rangle + |\beta\rangle) + |\gamma\rangle.$$

Esiste un vettore **nullo** $|0\rangle$, con la proprietà

$$|\alpha\rangle + |0\rangle = |\alpha\rangle,$$

[1] In questa tesi si farà ampio utilizzo della notazione di Dirac, ossia tramite bra-ket. Il simbolo $\langle\alpha|$ è chiamato "bra", mentre $|\alpha\rangle$ è detto "ket". L'uso combinato di entrambi è noto come "bra-ket", $\langle\alpha\,|\,\beta\rangle$.

[2] Nel contesto della meccanica quantistica, si farà uso esclusivamente dei numeri complessi appartenenti a $\mathbb{C}$. Si ricorda che la notazione z^* fa riferimento al *coniugato* di z. Quindi, se $z = a + ib \in \mathbb{C}$, allora $z^* = a - ib$.

per ogni $|\alpha\rangle$. Allo stesso modo, esiste l'**inverso** $|-\alpha\rangle$:

$$|\alpha\rangle + |-\alpha\rangle = |0\rangle.$$

La **moltiplicazione** per uno scalare a, denominata anche prodotto, è espressa come segue:

$$a|\alpha\rangle = |\gamma\rangle.$$

Analogamente alla somma, la moltiplicazione per uno scalare è **distributiva** e **associativa**:

$$(a+b)(|\alpha\rangle + |\beta\rangle) = a|\alpha\rangle + a|\beta\rangle + b|\alpha\rangle + b|\beta\rangle,$$
$$a(b|\alpha\rangle) = (ab)|\alpha\rangle.$$

Inoltre, la moltiplicazione per 0 e 1 produce:

$$0|\alpha\rangle = |0\rangle; \qquad 1|\alpha\rangle = |\alpha\rangle.$$

Una **combinazione lineare** di vettori $|\alpha\rangle, |\beta\rangle, |\gamma\rangle, \ldots$ è un'espressione della forma

$$a|\alpha\rangle + b|\beta\rangle + c|\gamma\rangle + \cdots .$$

Un vettore $|\lambda\rangle$ si dice **linearmente indipendente** da $|\alpha\rangle, |\beta\rangle, |\gamma\rangle, \ldots$ se non può essere espresso come una loro combinazione lineare. L'insieme di tutte le possibili combinazioni lineari di un insieme di vettori è chiamato **span**. Inoltre, se lo span coincide con lo spazio, i suoi vettori sono detti **generatori**. Un insieme di generatori linearmente indipendenti è denominato **base** di tale spazio. La **dimensione** dello spazio corrisponde al numero di vettori nella sua base.

Fissata una base arbitraria $\{|e_1\rangle, |e_2\rangle, \ldots, |e_n\rangle\}$, un qualsiasi vettore $|\alpha\rangle = a_1|e_1\rangle + a_2|e_2\rangle + \cdots + a_n|e_n\rangle$ può essere identificato univocamente mediante le sue **componenti** ordinate:

$$|\alpha\rangle \leftrightarrow (a_1, a_2, \ldots, a_n).$$

3.1.2 Prodotto interno

Un modo naturale per descrivere uno spazio vettoriale è attraverso il prodotto scalare, detto anche **prodotto interno**. Il prodotto interno tra due vettori $|\alpha\rangle, |\beta\rangle$, indicato come $\langle\alpha \,|\, \beta\rangle$, restituisce un numero complesso e gode delle seguenti proprietà:

- $\langle\alpha \,|\, \beta\rangle = \langle\beta \,|\, \alpha\rangle^*$.
- $\langle\alpha \,|\, \alpha\rangle \geq 0$ e $\langle\alpha \,|\, \alpha\rangle = 0 \Leftrightarrow |\alpha\rangle = |0\rangle$.
- $\langle\alpha|(b|\beta\rangle + c|\gamma\rangle) = b\langle\alpha \,|\, \beta\rangle + c\langle\alpha \,|\, \gamma\rangle$.

Poiché il prodotto interno di qualsiasi vettore $|\alpha\rangle$ con se stesso è sempre una quantità non negativa, possiamo calcolare la sua radice reale. Definiamo questa quantità la **norma** di quel vettore:

$$||\alpha|| \equiv \sqrt{\langle \alpha \,|\, \alpha \rangle}.$$

La norma introduce il concetto di lunghezza: un vettore *unitario*, ossia di norma 1, si dice **normalizzato**. Due vettori il cui loro prodotto interno è 0 sono **ortogonali**. Un insieme di vettori ortogonali e normalizzati, con $\langle \alpha_i \,|\, \alpha_j \rangle = \delta_{ij}$[3], è detto **ortonormale**.

È sempre preferibile scegliere una base ortonormale. In questo modo, quanto detto precedentemente può essere ulteriormente semplificato:

- $\langle \alpha \,|\, \beta \rangle = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$.
- $\langle \alpha \,|\, \alpha \rangle = |a_1|^2 + |a_2|^2 + \cdots + |a_n|^2$.
- $a_i = \langle e_i \,|\, \alpha \rangle$.

3.1.3 Trasformazioni lineari

Consideriamo un vettore nello spazio tridimensionale: la sua moltiplicazione per un numero o una rotazione sull'asse z rappresentano esempi di **trasformazioni lineari**. Una trasformazione lineare $\hat{T}$ agisce su un qualsiasi vettore di uno spazio e lo trasforma in un altro vettore, a condizione che l'operazione sia lineare:

$$\hat{T}(a|\alpha\rangle + b|\beta\rangle) = a(\hat{T}|\alpha\rangle) + b(\hat{T}|\beta\rangle),$$

per ogni vettore $|\alpha\rangle, |\beta\rangle$ e per ogni scalare a, b.

Noto l'effetto di una trasformazione lineare su una base, possiamo dedurne l'effetto su qualsiasi vettore generato da essa. Se $|\alpha\rangle$ è un qualsiasi vettore

$$|\alpha\rangle = a_1|e_1\rangle + a_2|e_2\rangle + \cdots + a_n|e_n\rangle = \sum_{j=1}^{n} a_j|e_j\rangle,$$

allora

$$\hat{T}|\alpha\rangle = \sum_{j=1}^{n} a_j(\hat{T}|e_j\rangle) = \sum_{j=1}^{n}\sum_{i=1}^{n} a_j T_{ij}|e_i\rangle = \sum_{i=1}^{n}\left(\sum_{j=1}^{n} T_{ij}a_j\right)|e_i\rangle.$$

Ciò implica che la trasformazione $\hat{T}$ prende le componenti di un qualsiasi vettore e ne produce altrettante per il vettore risultante:

$$a_i^{'} = \sum_{j=1}^{n} T_{ij}a_j.$$

[3] δ_{ij} è detta delta di Kronecker ed è uguale a 1 se $i = j$ e 0 altrimenti.

Tutti questi T_{ij} identificano la trasformazione $\hat{T}$ allo stesso modo con cui le componenti identificano un vettore. In altre parole, $\hat{T} \leftrightarrow (T_{11}, T_{12}, \ldots, T_{nn})$ e, se si sta usando una base ortonormale, possiamo indicare $T_{ij} = \langle e_i|\hat{T}|e_j\rangle$.

Risulta molto utile riscrivere queste componenti in una **matrice**[4]:

$$\boldsymbol{T} = \begin{pmatrix} T_{11} & T_{12} & \ldots & T_{1n} \\ T_{21} & T_{22} & \ldots & T_{2n} \\ \vdots & \vdots & & \vdots \\ T_{n1} & T_{n2} & \ldots & T_{nn} \end{pmatrix}.$$

La somma di due trasformazioni lineari $\hat{S} + \hat{T}$ si definisce come

$$(\hat{S} + \hat{T})|\alpha\rangle = \hat{S}|\alpha\rangle + \hat{T}|\alpha\rangle,$$

e trova riscontro nella somma di matrici

$$\boldsymbol{U} = \boldsymbol{S} + \boldsymbol{T} \Leftrightarrow U_{ij} = S_{ij} + K_{ij}.$$

Il prodotto $\hat{S}\hat{T}$ si ottiene applicando in successione le trasformazioni al vettore, prima $\hat{T}$ e poi $\hat{S}$. Inoltre,

$$\boldsymbol{U} = \boldsymbol{S}\boldsymbol{T} \Leftrightarrow U_{ik} = \sum_{j=1}^{n} S_{ij} T_{jk}.$$

Questo illustra un modo per eseguire la moltiplicazione tra matrici. Lo stesso procedimento si applica anche a matrici rettangolari, purché il numero di colonne della prima matrice corrisponda al numero di righe della seconda. Se consideriamo le n componenti di un vettore $|\alpha\rangle$ come un vettore colonna $n \times 1$,

$$\boldsymbol{a} \equiv \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix},$$

possiamo indicare la trasformazione come

$$\boldsymbol{a}' = \boldsymbol{T}\boldsymbol{a}.$$

Indichiamo come **trasposta** di $\boldsymbol{T}$ la matrice $\tilde{\boldsymbol{T}}$ con le righe e le colonne scambiate tra di loro. Un vettore colonna trasposto produce un vettore riga.

Una matrice quadrata è **simmetrica** se è uguale alla sua trasposta e **antisimmetrica** se, oltre a essere uguale alla sua trasposta, ha invertito il segno.

[4]Per semplificare la comprensione di questa sezione, le matrici saranno indicate in grassetto.

Per ottenere il coniugato di una matrice, indicato con $\boldsymbol{T}^*$, si prende il coniugato di ogni elemento.

Il coniugato hermitiano di una matrice o **matrice aggiunta**, indicata con $\boldsymbol{T}^\dagger$, è la trasposta del coniugato:

$$\boldsymbol{T}^\dagger \equiv \tilde{\boldsymbol{T}}^* = \begin{pmatrix} T_{11}^* & T_{21}^* & \dots & T_{n1}^* \\ T_{12}^* & T_{22}^* & \dots & T_{n2}^* \\ \vdots & \vdots & & \vdots \\ T_{1n}^* & T_{2n}^* & \dots & T_{nn}^* \end{pmatrix}; \qquad \boldsymbol{a}^\dagger \equiv \tilde{\boldsymbol{a}}^* = \begin{pmatrix} a_1^* & a_2^* & \dots & a_n^* \end{pmatrix}.$$

Una matrice quadrata è detta hermitiana o **autoaggiunta** se è uguale alla sua hermitiana coniugata.

Tramite questa notazione, il prodotto interno di due vettori rispetto a una base ortonormale può essere scritto in forma matriciale come

$$\langle \alpha \,|\, \beta \rangle = \boldsymbol{a}^\dagger \boldsymbol{b},$$

con a e b matrici.

La **matrice identità** è formata da uni sulla diagonale principale e zeri nel resto della stessa:

$$\boldsymbol{I} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix}.$$

L'**inversa** di una matrice, indicata con $\boldsymbol{T}^{-1}$, si definisce classicamente:

$$\boldsymbol{T}^{-1}\boldsymbol{T} = \boldsymbol{T}\boldsymbol{T}^{-1} = \boldsymbol{I}.$$

Una matrice è detta **unitaria** se equivale alla sua hermitiana coniugata.

3.1.4 Autovalori e autovettori

Consideriamo una trasformazione lineare che consiste in una rotazione nello spazio tridimensionale rispetto a un asse. Se il vettore è allineato lungo quell'asse, questo rimarrà invariato. In uno spazio vettoriale complesso, ogni trasformazione lineare può contare su vettori simili a questo, che vengono trasformati in multipli di se stessi:

$$\hat{T}|\alpha\rangle = \lambda|\alpha\rangle.$$

Questi vettori sono chiamati **autovettori** della trasformazione, mentre il numero λ è detto **autovalore**.

La soluzione del polinomio caratteristico[5] associato a una matrice, ottenuto ponendo $\det(\boldsymbol{T} - \lambda \boldsymbol{I}) = 0$, fornisce gli autovalori necessari a riscrivere la matrice $\boldsymbol{T}$ nella base degli autovettori corrispondenti. Ogni autovalore sarà posizionato sulla diagonale della matrice $\boldsymbol{T}$. Una matrice che può essere diagonalizzata attraverso un cambio di base è detta **diagonalizzabile**.

3.1.5 Trasformazioni hermitiane

Ricordando la definizione della matrice aggiunta, si vuole definire il comportamento dell'hermitiana coniugata di una trasformazione lineare. Quando questa è applicata al primo membro del prodotto interno, si deduce che $\hat{T}$ è stata applicata al secondo. In altre parole, per qualsiasi $|\alpha\rangle, |\beta\rangle$:

$$\langle \hat{T}^\dagger \alpha \,|\, \beta \rangle = \langle \alpha \,|\, \hat{T}\beta \rangle.$$

Nel caso si utilizzino basi ortonormali, l'aggiunto di una trasformazione lineare coincide con la rispettiva matrice aggiunta. Pertanto, vale

$$\langle \alpha \,|\, \hat{T}\beta \rangle = \boldsymbol{a}^\dagger \boldsymbol{T} \boldsymbol{b} = (\boldsymbol{T}^\dagger \boldsymbol{a})^\dagger \boldsymbol{b} = \langle \hat{T}^\dagger \alpha \,|\, \beta \rangle.$$

Gli autovalori e gli autovettori di una trasformazione hermitiana godono delle seguenti proprietà:

- Gli autovalori sono reali.
- Gli autovettori corrispondono ad autovalori distinti e sono ortogonali.
- Gli autovettori costituiscono lo span dello spazio.

3.1.6 Operatori

In uno spazio di funzioni[6], gli **operatori** si comportano come trasformazioni lineari, a patto che trasformino funzioni in altre funzioni e siano lineari.

In questo contesto, gli autovettori di un operatore $\hat{T}$ sono chiamati **autofunzioni** e seguono la definizione standard per cui $\hat{T}f(x) = \lambda f(x)$.

Analogamente, un operatore hermitiano si comporta in modo tale che $\langle f \,|\, \hat{T} g \rangle = \langle \hat{T} f \,|\, g \rangle$, per ogni coppia di funzioni $f(x)$ e $g(x)$.

---

[5] Per semplicità, non viene illustrato come trovare il determinante di una matrice. Si tratta di un calcolo sistematico la cui procedura è facilmente reperibile.

[6] Non viene trattato il concetto di spazio di funzioni. Per questa trattazione è sufficiente compararli a un comune spazio vettoriale.

Spazio di Hilbert

Uno spazio è completo se ogni successione convergente di vettori appartenenti a esso ha il limite anch'esso appartenente allo stesso spazio. Inoltre, uno spazio completo su cui è definito un prodotto interno è detto **spazio di Hilbert**.

Siano H_1 e H_2 due spazi di Hilbert, rispettivamente di dimensione m e n. Diciamo che lo spazio di Hilbert H è il **prodotto tensoriale** di H_1 e H_2, e lo indichiamo con $H = H_1 \otimes H_2$, se è uno spazio dotato dell'applicazione bilineare $\otimes : H_1 \times H_2 \to H$ e base $|\alpha\rangle \otimes |\beta\rangle$ con $|\alpha\rangle \in H_1$ e $|\beta\rangle \in H_2$.

3.2 La funzione d'onda

3.2.1 L'equazione di Schrödinger

Immaginiamo una particella di massa m che si sposta esclusivamente lungo l'asse x a causa di una forza $F(x,t)$. Nella meccanica classica, è possibile determinare la posizione della particella in ogni istante $x(t)$. Tramite questa nozione, è possibile calcolare la velocità, l'energia cinetica o altre variabili di nostro interesse, grazie alla seconda legge di Newton, $F = ma$.

La meccanica quantistica affronta lo stesso problema in modo differente. Lo strumento necessario è la cosiddetta funzione d'onda $\Psi(x,t)$ di una particella (o stato), che si ottiene della risoluzione dell'equazione di Schrödinger:

$$i\hbar\frac{\partial\Psi}{\partial t} = \frac{-\hbar^2}{2m}\frac{\partial^2\Psi}{\partial x^2} + V\Psi, \tag{3.1}$$

dove i è la radice quadrata di -1 e $\hbar$ rappresenta la costante di Planck, determinata sperimentalmente a 1.054573×10^{-34} Joule s.

In questo contesto, l'equazione svolge un ruolo simile a quella di Newton, permettendoci di determinare la funzione d'onda in qualsiasi istante. Data una condizione iniziale, per esempio $\Psi(x,0)$, l'equazione di Schrödinger determina i valori di $\Psi(x,t)$ in qualsiasi istante futuro.

Interpretazione statistica

La funzione d'onda ci fornisce quanto descritto da Born, ovvero una sua interpretazione statistica. Secondo lui, $|\Psi(x,t)|^2$ rappresenta la probabilità di trovare una particella nel punto x al tempo t, ossia

$$|\Psi(x,t)|^2\, dx = \left\{ \begin{array}{l} \text{probabilità di trovare la particella} \\ \text{tra } x \text{ e } (x+dx) \text{ al tempo } t. \end{array} \right\}$$

Questo approccio probabilistico ci lascia con un'*incertezza*, una caratteristica molto presente nella meccanica quantistica.

Normalizzazione

Poiché $|\Psi(x,t)|^2$ esprime una probabilità, è logico pensare che il suo integrale sia 1, altrimenti non avrebbe senso parlare di probabilità. È importante ricordare che la somma delle probabilità di tutti gli eventi possibili equivale a 1 (Appendice A). Pertanto, possiamo affermare che

$$\int_{-\infty}^{+\infty} |\Psi(x,t)|^2 \, dx = 1.$$

Per garantire questa condizione, è necessario effettuare alcune verifiche. Possiamo affermare che se $\Psi(x,t)$ è una soluzione dell'equazione (3.1), allora lo è anche $A\Psi(x,t)$, con A una qualsiasi costante complessa. Affinché questa condizione sia rispettata, quanto detto deve essere valido per qualsiasi fattore moltiplicativo. La ricerca di questo fattore viene indicata come *normalizzazione*.

Potremmo chiederci se la funzione d'onda rimanga normalizzata durante l'intera evoluzione di Ψ. Fortunatamente, l'equazione di Schrödinger ha una proprietà che garantisce il mantenimento della normalizzazione della funzione d'onda [17].

Misurazioni

Considerando una particella nello stato Ψ, il valore atteso di x è dato da

$$\langle x \rangle = \int_{-\infty}^{+\infty} x|\Psi(x,t)|^2 \, dx.$$

La prima misurazione, il cui risultato non può essere determinato con esattezza, provoca il *collasso* della funzione d'onda nel valore ottenuto in quella misurazione, e tutte le immediate misurazioni successive avranno lo stesso esito. Alla luce di ciò, $\langle x \rangle$ non va interpretato come la media dei risultati di una stessa misurazione ripetuta, bensì come la media delle misurazioni fatte su più particelle che si trovano nel medesimo stato.

Chiamiamo x *operatore* posizione. Un operatore può essere inteso come un'istruzione che indica come comportarsi nei confronti della funzione che segue, consentendo di ricavare le informazioni necessarie per lo studio di una funzione d'onda.

3.2.2 Il principio di indeterminazione

Supponiamo[7] di tenere in mano una corda e agitarla su e giù: questo genererebbe una serie di onde come mostrato in Figura 3.1a.

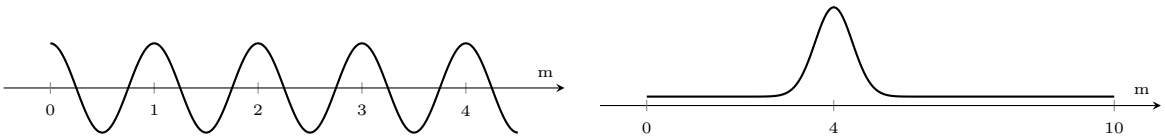
Se qualcuno ci chiedesse *dove* si trovi l'onda, avremmo non poca difficoltà a rispondere. Potremmo dire che le onde della corda si estendono per poco più di 4 metri. Invece, se ci chiedessero quale fosse la lunghezza d'onda[8], potremmo rispondere con abbastanza precisione: 1 metro.

Supponiamo ora di agitare la corda con un colpo deciso, ottenendo un effetto simile alla Figura 3.1b. Questa volta saremo in grado di rispondere alla domanda che ci chiede dove si trovi l'onda, ma non riusciremo a indicare una precisa lunghezza d'onda.

Con questo semplice esempio si possono descrivere anche tutti gli scenari intermedi, che tendono maggiormente verso uno oppure l'altro caso. Quello che emerge è che tanto più cerchiamo di conoscere con precisione la posizione dell'onda, meno siamo in grado di determinare con precisione la sua lunghezza d'onda, e viceversa.

Il medesimo concetto si applica alle funzioni d'onda in meccanica quantistica. In questo caso, la lunghezza d'onda di Ψ corrisponde alle variazioni della velocità. Di conseguenza, tanto più vogliamo sapere la posizione di una particella, tanto meno sarà precisa la determinazione della sua velocità.

Questo concetto è fondamentale per introdurre il principio di indeterminazione di Heisenberg, su cui si basa anche la crittografia quantistica.



(a) L'onda è equamente distribuita. (b) La corda genera un picco in un punto.

Figura 3.1: Esempio della corda.

3.3 Postulati

Questa sezione enuncia una serie di postulati della meccanica quantistica essenziali per comprendere gli argomenti successivi. Si consiglia la lettura preliminare della Sezione 3.1 per ottenere un'interpretazione più chiara.

Lo scopo finale di questa sezione è fornire un'idea generale dei principi alla base della meccanica quantistica che hanno reso possibile lo sviluppo della crittografia quantistica.

[7] Si segue la spiegazione di Griffiths, che risulta essere intuitiva e chiara per illustrare il principio di indeterminazione.

[8] In fisica, la lunghezza d'onda indica la distanza tra due creste di un'onda periodica.

3.3.1 Stato

Lo stato di un sistema quantistico è descritto completamente da una funzione d'onda, o vettore di stato, generalmente indicato con $|\psi\rangle$ e appartenente allo spazio di Hilbert. Questa funzione d'onda contiene tutte le informazioni possibili sullo stato del sistema e permette di calcolare le diverse probabilità delle osservabili.

In un sistema monodimensionale, come nel caso di una particella libera, la funzione d'onda potrebbe essere una funzione che assegna un numero complesso a ogni punto nello spazio. Per un sistema più complesso, come un atomo, questa funzione dipenderà dalle coordinate spaziali di tutte le particelle coinvolte.

Nonostante riassuma tutte le proprietà fisiche di interesse di un sistema, il vettore $|\psi\rangle$ non rappresenta una quantità direttamente misurabile. Tuttavia, il quadrato della norma di una funzione d'onda $||\psi\rangle|^2$ rappresenta la probabilità di trovare il sistema in una particolare configurazione.

L'interpretazione probabilistica della funzione d'onda è cruciale in meccanica quantistica. Durante una misurazione, il sistema collassa in uno degli stati possibili con una probabilità associata alla forma della funzione d'onda. Questo contrasta con la visione deterministica della meccanica classica, dove lo stato di un sistema è completamente definito e prevedibile.

3.3.2 Osservabili e operatori

Nella fisica classica, siamo abituati a esprimere le grandezze osservabili, come ad esempio la posizione e l'energia, come variabili misurabili attraverso la sperimentazione, ovvero grandezze che possono essere quantificate numericamente. In meccanica quantistica è necessario associare a quali quantità algebriche corrispondano quelle osservabili. Per questo motivo, entrano in gioco due nozioni fondamentali:

- Le **osservabili** sono grandezze fisiche misurabili, come la posizione, la quantità di moto, l'energia, eccetera. Ogni osservabile è associata a un operatore matematico.
- Gli **operatori** sono oggetti matematici che agiscono sulla funzione d'onda del sistema. Un operatore è solitamente indicato dal simbolo $\hat{O}$. L'azione di un operatore su una funzione d'onda restituisce un'altra funzione d'onda.

A titolo di esempio, se O rappresenta la posizione di una particella, l'operatore associato sarà proprio $\hat{O} = \hat{x}$, dove $\hat{x}$ è l'operatore posizione. Se $|\psi(x)\rangle$ è la funzione d'onda della particella, allora l'azione dell'operatore $\hat{O}$ su $|\psi(x)\rangle$ restituirà la funzione $x|\psi(x)\rangle$, dove x è la posizione.

Riprendendo i concetti dell'algebra e ricordando che lo stato di un sistema quantistico è descritto da un vettore nello spazio di Hilbert, l'obiettivo è ottenere misurazione numeriche avendo a disposizione questi vettori. Le quantità in questione corrispondono esattamente agli operatori lineari autoaggiunti $O : H \to H$.

Un'osservabile, essendo associata a un operatore autoaggiunto, ammette sempre una base ortonormale dello spazio di Hilbert. Ne consegue che un qualunque $|\psi\rangle$ può essere espresso sulla base degli autostati $|a_i\rangle$ associati a O, con componenti $\langle a_i \,|\, \psi\rangle$. Il quadrato di queste componenti indica la probabilità che lo stato generico $|\psi\rangle$ dia come risultato $|a_i\rangle$, in seguito a una misura dell'osservabile O.

3.3.3 Misurazioni

Come accennato in precedenza, il processo di misurazione è caratterizzato da una natura probabilistica. Quando si effettua una misurazione, si ottengono risultati discreti associati agli autovalori delle grandezze misurate. Ricorrendo all'esempio della misura della posizione, i risultati possibili corrisponderanno agli autovalori dell'operatore posizione.

La probabilità di ottenere un particolare risultato è legata al modulo al quadrato della componente corrispondente della funzione d'onda. Supponendo che $|\psi(x)\rangle$ sia la funzione d'onda associata alla posizione, la probabilità di trovare una particella in un intervallo Δx intorno a x è data da

$$\int_x^{x+\Delta x} ||\psi\rangle|^2 \, dx.$$

Questo postulato sottolinea il forte aspetto probabilistico della meccanica quantistica. Fino a quando una misurazione non viene effettuata, il sistema può esistere in uno stato di *sovrapposizione* di possibili risultati. Solo al momento della misurazione la funzione d'onda collassa su uno degli stati possibili, e la probabilità associata a ciascuno di essi determina quale risultato verrà osservato.

3.3.4 Evoluzione temporale

Se da un lato le possibili misurazioni per un generico stato non possono essere predette, dall'altro è completamente conservato il determinismo della funzione d'onda $|\psi\rangle$. Fissato il suo valore all'istante iniziale, la funzione d'onda segue quanto descritto dall'equazione di Schrödinger, la cui evoluzione temporale è univocamente determinata.

$$i\hbar \frac{d}{dt}|\psi(t)\rangle = \hat{H}|\psi(t)\rangle.$$

In questa formulazione dell'equazione, $\hat{H}$ è un operatore autoaggiunto chiamato Hamiltoniano e rappresenta l'energia totale del sistema.

L'equazione di Schrödinger fornisce una connessione tra lo stato iniziale del sistema e il suo stato futuro, e mostra come la funzione d'onda si evolve nel tempo. È importante notare che l'evoluzione temporale è determinata dall'energia totale del sistema, e l'equazione di Schrödinger è fondamentale per predire il comportamento quantistico nel corso del tempo.

3.4 No cloning quantistico

Il teorema della non clonazione quantistica, enunciato nel 1982 da Wootters, Zurek e Dieks, ha implicazioni significative nel contesto del calcolo quantistico e dei campi a esso correlati.

Questo principio afferma l'impossibilità di duplicare con precisione uno stato quantistico non conosciuto precedentemente. Tuttavia, si ammette la possibilità di replicare senza errori tale stato nel caso in cui faccia parte di un insieme ortogonale di stati già noti.

Teorema (No-cloning quantistico)**.** Un qualsiasi stato quantistico non può essere perfettamente duplicato.

Dimostrazione. Supponiamo di avere due sistemi quantistici A e B con uno spazio di Hilbert comune, $H = H_A = H_B$. Consideriamo una procedura in grado di copiare lo stato $|\phi\rangle_A$ del sistema A nello stato $|e\rangle_B$ del sistema B, a partire da uno stato sconosciuto $|\phi\rangle_A$ qualsiasi.

In altre parole, desideriamo trasformare lo stato $|\phi\rangle_A \otimes |e\rangle_B$ in $|\phi\rangle_A \otimes |\phi\rangle_B$, indipendentemente da $|\phi\rangle_A$, per copiare uno stato di A combinandolo con uno stato sconosciuto $|e\rangle_B$ di B.

Poiché una misura del sistema comprometterebbe lo stato stesso, rimane solo la possibilità di applicare un operatore unitario U dello spazio $H \otimes H$. Tuttavia, non esiste un U tale per cui tutti gli stati normalizzati $|\phi\rangle_A$ e $|e\rangle_B$ in H soddisfino l'equazione

$$U(|\phi\rangle_A|e\rangle_B) = |\phi\rangle_A|\phi\rangle_B.$$

Per dimostrarlo, consideriamo la coppia di stati $|\phi\rangle_A$ e $|\psi\rangle_A$ in H, e costruiamo le corrispondenti coppie in $H \otimes H$ come $|\phi\rangle_A|e\rangle_B$ e $|\psi\rangle_A|e\rangle_B$. Dal momento che U è unitario, conserva il prodotto scalare:

$$\langle\phi\,|\,\psi\rangle\langle e\,|\,e\rangle \equiv \langle\phi|_A\langle e|_B|\psi\rangle_A|e\rangle_B = \langle\phi|_A\langle e|_B U^\dagger U|\psi\rangle_A|e\rangle_B = \langle\phi|_A\langle\phi|_B|\psi\rangle_A|\psi\rangle_B \equiv \langle\phi\,|\,\psi\rangle^2.$$

Dato che gli stati sono normalizzati, segue che

$$\langle\phi\,|\,\psi\rangle = \langle\phi\,|\,\psi\rangle^2.$$

Ciò implica $\phi = \psi$ oppure che ϕ è ortogonale a ψ. Tuttavia, ciò non è valido per due stati qualsiasi, e un operatore U con queste caratteristiche non può esistere. Evidentemente, U non può copiare qualunque stato quantistico. □

In conclusione, il tentativo di clonare uno stato quantistico comporterebbe l'esistenza di un operatore di clonazione valido per qualsiasi stato sconosciuto, ma questa possibilità è limitata agli stati già noti oppure ortogonali.

Capitolo 4

La Crittografia Quantistica

La crittografia si occupa della sicurezza delle comunicazioni o delle operazioni tra due o più parti che potrebbero non avere fiducia reciproca. Il problema crittografico più noto riguarda la trasmissione di messaggi segreti, gestita attraverso protocolli crittografici. La distinzione principale che è stata osservata riguarda i sistemi crittografici simmetrici e quelli asimmetrici.

Come discusso nel capitolo precedente, i sistemi simmetrici coinvolgono un mittente e un destinatario che comunicano attraverso una chiave condivisa e segreta. Il mittente cifra il messaggio con questa chiave e lo invia al destinatario, l'unico in grado di interpretare il contenuto perché possiede la chiave con cui il messaggio è stato originariamente cifrato.

D'altro canto, i sistemi crittografici a chiave privata presentano sfide, in particolare per quanto riguarda la distribuzione delle chiavi, una fase molto sensibile. Un malintenzionato potrebbe intercettare la chiave durante lo scambio, compromettendo la sicurezza dei messaggi trasmessi sul canale di comunicazione.

Una delle prime applicazioni della meccanica quantistica alla crittografia è stata la distribuzione a chiave quantistica (QKD, Quantum Key Distribution). Questa tecnica consente lo scambio di chiavi senza compromettere la sicurezza, grazie alla sensibilità della misura in un sistema quantistico. In caso di intercettazione, la comunicazione risulterebbe perturbata, indicando una possibile compromissione della sicurezza e rendendo necessario un nuovo scambio di chiavi.

Un altro elemento fondamentale della crittografia sono i sistemi asimmetrici, o a chiave pubblica. In questo caso, mittente e destinatario non basano la loro comunicazione sulla conoscenza di una sola chiave condivisa, ma su due chiavi, una pubblica e una privata. Il mittente utilizza la chiave pubblica del destinatario per cifrare il messaggio, mentre il destinatario lo decifra con la sua chiave privata associata alla chiave pubblica utilizzata in precedenza.

L'interesse nella crittografia quantistica è cresciuto notevolmente negli ultimi anni, con risultati sperimentali che dimostrano la sua praticità su lunghe distanze grazie all'utilizzo di fibre ottiche. Il Quantum Key Distribution ha raggiunto successi significativi, anche

grazie a una collaborazione europea nel 2006, riuscendo a far comunicare a cielo aperto due isole delle Canarie a una distanza di 144 km [38].

4.1 Il Qubit

L'informatica più tradizionale affida il suo funzionamento al *bit*, un valore che può assumere solamente due stati: 1 e 0. Tramite l'utilizzo di questi due valori è possibile costruire un vero e proprio sistema numerico, quello binario, con operazioni matematiche e logiche.

La variante quantistica del bit è il quantum bit, o *qubit* per brevità. A differenza dei bit nella computazione classica, i qubit possono esistere contemporaneamente in una sovrapposizione di stati, sfruttando principi della meccanica quantistica come l'entanglement e l'interferenza quantistica.

Il qubit descrive uno spazio di Hilbert complesso bidimensionale, con la base canonica definita come:

$$|0\rangle \equiv \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad |1\rangle \equiv \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

correlata alla rappresentazione binaria classica.

Per il principio di sovrapposizione, lo stato generico di un qubit può essere descritto da

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle,$$

con α e β coefficienti complessi tali che $|\alpha|^2 + |\beta|^2 = 1$.

Un sistema quantistico a due livelli può modellarsi attraverso un qubit se:

- Può essere preparato in uno stato definito, per esempio lo stato $|0\rangle$.
- Ogni stato del qubit può essere trasformato in un altro stato.
- Lo stato del qubit può essere misurato nella base canonica.

Un'interpretazione del qubit può essere trovata anche in natura, come nel modello atomico. Ad esempio, un elettrone può trovarsi nello stato fondamentale o eccitato, con la possibilità che transizioni verso uno di essi con l'apporto di energia. Riducendo il tempo di esposizione, si può posizionare l'elettrone in uno stato "intermedio" [28], rappresentato da:

$$\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle.$$

Tale stato sarà denotato come $|+\rangle$, mentre $|-\rangle$ sarà lo stato $\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$. Una misura pari a $|+\rangle$ corrisponderà a un elettrone nello stato fondamentale o eccitato, con probabilità del 50% in entrambi i casi.

La scelta di $|0\rangle$ e $|1\rangle$ come base ortonormale per esprimere un qubit è solo una di quelle possibili. La scelta di $|+\rangle$ e $|-\rangle$ come base per descrivere lo stesso spazio è altrettanto valida, così come infinite altre opzioni.

Qubit multipli

Finora è stato considerato l'utilizzo di un singolo qubit, ma è possibile estendere quanto appena trattato anche a due qubit. Se con due bit, in modo classico, sono permesse le configurazioni 00, 01, 10, 11, quantisticamente accade lo stesso con altrettanti stati: $|00\rangle, |01\rangle, |10\rangle, |11\rangle$. Questi rappresentano un'altra base ortonormale valida per descrivere lo spazio vettoriale.

Rispettando tale configurazione, due qubit in questa base possono essere indicati come

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle,$$

con $\sum_{ij} |\alpha_{ij}|^2 = 1$, ossia con coefficienti normalizzati.

Quando viene misurato uno dei due qubit, il sistema complessivo collasserà in uno degli stati di base della sovrapposizione. Come previsto, il risultato avrà un esito probabilistico. Nel caso in cui il primo qubit dia come risultato $|0\rangle$, ciò avverrà con probabilità $|\alpha_{00}|^2+|\alpha_{01}|^2$, e il sistema collasserà nello stato

$$|\psi'\rangle = \frac{\alpha_{00}|00\rangle + \alpha_{01}|01\rangle}{\sqrt{|\alpha_{00}|^2 + |\alpha_{01}|^2}};$$

mentre darà $|1\rangle$ con probabilità $|\alpha_{10}|^2 + |\alpha_{11}|^2$, collassando nello stato

$$|\psi'\rangle = \frac{\alpha_{10}|10\rangle + \alpha_{11}|11\rangle}{\sqrt{|\alpha_{10}|^2 + |\alpha_{11}|^2}}.$$

4.1.1 Fotoni e polarità

La meccanica quantistica definisce un fascio di luce come la composizione di quantità discrete di energia, chiamate fotoni. Data la natura ondulatoria della luce [12], queste particelle hanno un angolo θ di polarizzazione rispetto al piano sul quale oscillano, che va da 0° a 180°. L'utilizzo di un filtro polarizzatore, denominato θ-filter, consente di impostare un angolo θ a piacimento.

Un filtro polarizzatore permette il passaggio di un fotone solamente se la sua polarizzazione è parallela all'asse del filtro stesso; in caso contrario, viene bloccato. Al di fuori di questi due casi, non è possibile ottenere altre informazioni se non la probabilità che un fotone attraversi o meno il filtro. Un fotone polarizzato obliquamente rispetto all'asse ottico attraverserà il filtro con una probabilità e con una polarizzazione perpendicolare all'asse stesso. La probabilità che un fotone di angolo φ attraversi un θ-filter risulta essere

$$p_\theta(\varphi) = \cos^2(\varphi - \theta).$$

Una delle metodologie più adottate per realizzare un qubit è l'utilizzo di un fotone polarizzato in due stati differenti: $|0\rangle$ quando è polarizzato lungo l'asse x, $|1\rangle$ quando è polarizzato lungo l'asse y. Tuttavia, è sempre possibile imprimere un angolo θ rispetto all'asse x. Il qubit sarà allora definito dalla seguente funzione d'onda:

$$|\psi\rangle = \cos\theta|0\rangle + \sin\theta|1\rangle.$$

È possibile misurare la polarizzazione di un fotone attraverso un analizzatore di polarizzazione, che può essere realizzato mediante un cristallo di calcite. Per determinare se il fotone è polarizzato lungo l'asse x o y, è sufficiente osservare se attraversa o viene deviato dal cristallo. Indicando con 0 il primo risultato e con 1 il secondo, i rispettivi eventi si verificano secondo quanto descritto nella Sezione 3.3.1:

$$p_0 = |\langle 0 \,|\, \psi\rangle|^2 = \cos^2\theta \quad \text{e} \quad p_1 = |\langle 1 \,|\, \psi\rangle|^2 = \sin^2\theta.$$

Configurazioni

Per scopi descrittivi, il raggio luminoso sarà considerato composto da un singolo fotone polarizzato. Nonostante il fotone possa assumere una polarizzazione arbitraria, verranno considerati solo quelli polarizzati con quattro configurazioni specifiche: 0°, 90°, 45° e 135°. Per semplicità, gli angoli saranno indicati rispettivamente con $\uparrow$, $\rightarrow$, $\nearrow$, $\searrow$.

Va notato che le configurazioni di 45° e 135° corrispondono agli stati $|+\rangle$ e $|-\rangle$ definiti in precedenza. Nel caso di una misurazione attraverso un analizzatore per $|0\rangle$ e $|1\rangle$, non è garantito che un fotone polarizzato con angoli di 45° e 135° lo attraversi. Inoltre, l'analizzatore non sarà in grado di distinguere i fotoni polarizzati in queste configurazioni in quanto non sono ortogonali a $|0\rangle$ e $|1\rangle$. Se il fotone riesce ad attraversare il cristallo, questo subirà una nuova polarizzazione, ovvero quella del cristallo attraversato.

Per una maggiore chiarezza, viene riportata in Tabella 4.1 una sintesi delle probabilità nei vari scenari.

	A: 0°	A: 90°	A: 45°	A: 135°
P: 0° $\uparrow$	1	0	1/2	1/2
P: 90° $\rightarrow$	0	1	1/2	1/2
P: 45° $\nearrow$	1/2	1/2	1	0
P: 135° $\searrow$	1/2	1/2	0	1

Tabella 4.1: Probabilità che un fotone attraversi un analizzatore. Gli angoli riportati in prima riga sono quelli rilevati dall'analizzatore, quelli in prima colonna sono le polarizzazioni dei fotoni.

4.2 Computazione quantistica

4.2.1 Quantum gates

Analogamente a un computer classico che fa affidamento sulle porte logiche[1], la realizzazione di un computer quantistico è resa possibile grazie a circuiti simili che permettono la manipolazione dell'informazione quantistica: le porte quantistiche o *quantum gates*.

Mentre per una porta logica tradizionale basta conoscere la sua tabella di verità per ottenere il valore del bit dell'operazione risultante, un quantum gate fornisce informazioni utili solo rispetto a una base ortonormale. Gli stati generici $|\psi\rangle$, come visto, sono espressi come combinazione lineare della base canonica, cioè $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$.

Dato che un'eventuale porta quantistica deve restituire uno stato normalizzato, gli operatori per realizzare una corrispondente porta logica devono essere unitari, e qualsiasi operatore unitario può essere una potenziale porta. Classicamente, l'unica porta logica valida per un bit, oltre all'identità (1→1, 0→0), è il NOT (1→0, 0→1). Tuttavia, in ambito quantistico, non esiste più una sola porta per un singolo qubit.

Esistono diverse porte quantistiche. Per esempio, la porta NOT può essere rappresentata come

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

e agisce su $|\psi\rangle$ nel seguente modo:

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix},$$

invertendo le componenti dello stato.

Ulteriori porte sono lo Z gate

$$Z \equiv \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},$$

che lascia invariato lo stato di $|0\rangle$ e inverte quello di $|1\rangle$, e l'Hadamard gate

$$H \equiv \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

[1] In elettronica digitale e informatica, è un circuito digitale in grado di implementare una particolare operazione logica di una o più variabili booleane.

Questo gate ha il particolare effetto di applicare le seguenti trasformazioni:

$$H|1\rangle = |-\rangle, \quad H|0\rangle = |+\rangle,$$
$$H|+\rangle = |0\rangle, \quad H|-\rangle = |1\rangle.$$

Si noti che applicare due volte H a qualsiasi stato lo lascia invariato, infatti

$$H^2 = \frac{1}{2}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} = \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix} = I.$$

4.2.2 Canale quantistico

Nella teoria dell'informazione quantistica, un canale quantistico rappresenta il mezzo attraverso il quale l'informazione quantistica viene trasmessa da un sistema all'altro. A differenza dei canali classici, i canali quantistici devono tener conto dei principi della meccanica quantistica, inclusi la sovrapposizione e l'entanglement. Un canale quantistico si compone di tre elementi:

- Un dispositivo ottico in grado di polarizzare i fotoni in entrata secondo una delle quattro configurazioni precedentemente citate.
- Un supporto che permette il transito dei fotoni (per esempio, un cavo di fibra ottica).
- Un dispositivo che consente al destinatario della comunicazione di misurare la polarizzazione dei fotoni.

4.3 Entanglement

La natura controintuitiva della correlazione tra sistemi quantistici fu inizialmente discussa da Albert Einstein nel 1935, in un articolo in collaborazione con a Boris Podolsky e Nathan Rosen [13]. In questo articolo, sì cercò di dimostrare sperimentalmente l'incompletezza della meccanica quantistica (paradosso EPR).

Successivamente, Erwin Schrödinger coniò il termine *entanglement* in una lettera indirizzata a Einstein per descrivere la correlazione tra due particelle che interagiscono e poi si separano, come evidenziato nell'esperimento EPR.

L'entanglement si manifesta solitamente attraverso l'interazione tra due particelle subatomiche, con queste interazioni che possono assumere diverse forme. Un sistema con tali caratteristiche permetterebbe a due particelle di diventare "intrecciate", in modo che la misura di una particella implichi necessariamente la conoscenza della misura dell'altra, indipendentemente da qualsiasi osservazione.

Se consideriamo due sistemi A e B, basati sui rispettivi spazi di Hilbert H_A e H_B, possiamo indicare lo spazio del sistema composto come $H_A \otimes H_B$. Se il primo sistema si

trova nello stato $|\psi\rangle_A$ e il secondo nello stato $|\psi\rangle_B$, lo stato del sistema composto sarà $|\psi\rangle_A \otimes |\psi\rangle_B$. Gli stati rappresentabili in questa forma sono detti stati *separabili*. Tuttavia, non tutti gli stati sono separabili. Fissata una base $|i\rangle_A$ per H_A e $|j\rangle_B$ per H_B, uno stato di $H_A \otimes H_B$ può essere espresso come

$$|\psi\rangle_{AB} = \sum_{i,j} c_{ij} |i\rangle_A \otimes |j\rangle_B.$$

Se $c_{ij} = c_i^A c_j^B$, con

$$|\psi\rangle_A = \sum_i c_i^A |i\rangle_A \quad \text{e} \quad |\psi\rangle_B = \sum_j c_j^B |j\rangle_B,$$

allora gli stati sono separabili. Tuttavia, se $c_{ij} \neq c_i^A c_j^B$, gli stati sono inseparabili e sono detti **entangled**.

Se supponiamo che $\{|0\rangle_A, |1\rangle_A\}$ sia la base di H_A e $\{|0\rangle_B, |1\rangle_B\}$ quella di H_B, lo stato

$$\frac{1}{\sqrt{2}}\big(|0\rangle_A|0\rangle_B + |1\rangle_A|1\rangle_B\big)$$

sarebbe entangled. Infatti, se il sistema composto si trova in questo stato, è impossibile attribuire al sistema A o a quello B uno stato *puro* (come le funzioni d'onda viste finora).

Un sistema con queste caratteristiche assume un ruolo significativo in crittografia, specialmente nella realizzazione di un canale di comunicazione sicuro. Il forte legame tra le parti impone limiti significativi alla possibilità di intercettazione, poiché qualsiasi misura influirebbe sul legame, generando gli effetti di una misura come discusso in precedenza.

4.4 Quantum Key Distribution

Il primo coinvolgimento della meccanica quantistica per affrontare le sfide non superate dalla crittografia classica avvenne nei primi anni Settanta. Il fisico Stephen Wiesner propose due modalità di comunicazione che furono inizialmente respinte. Solo dieci anni dopo, Charles H. Bennett e Gilles Brassard approfondirono l'idea di Wiesner, arrivando a sviluppare uno scambio di chiavi simile a quello della crittografia classica. Nel 1984, venne pubblicato il protocollo di distribuzione a chiave quantistica BB84 [7].

Un protocollo quantum key distribution consente lo scambio di chiavi attraverso un canale che non è sicuro, generando un risultato del tutto indipendente dai valori in ingresso. Ciò non implica che i principi descritti nella Sezione 1.2 debbano venir meno, ma permette di realizzare sistemi più sicuri. L'idea alla base di questa rinnovata sicurezza risiede nell'impossibilità per il crittoanalista di estrarre informazioni dalla comunicazione senza disturbare lo stato delle informazioni trasmesse attraverso i qubit.

Per il teorema di non clonazione, si ricorda che non è possibile copiare un qubit qualsiasi. Consideriamo due stati non ortogonali $|\psi\rangle$ e $|\phi\rangle$ desiderati dal crittoanalista. Quest'ultimo cerca di far interagire gli stati con un sistema nello stato $|u\rangle$ attraverso una trasformazione

unitaria. Se assumiamo che questo processo non modifichi gli stati $|\psi\rangle$ e $|\phi\rangle$, otteniamo quanto segue:

$$|\psi\rangle|u\rangle \mapsto |\psi\rangle|v\rangle,$$
$$|\phi\rangle|u\rangle \mapsto |\phi\rangle|v'\rangle,$$

dove $|v\rangle \neq |v'\rangle$ affinché il crittoanalista possa determinare correttamente gli stati. Le trasformazioni unitarie conservano il prodotto interno, ossia

$$\langle v \,|\, v'\rangle\langle \psi \,|\, \phi\rangle = \langle u \,|\, u\rangle\langle \psi \,|\, \phi\rangle,$$
$$\langle v \,|\, v'\rangle = \langle u \,|\, u\rangle = 1.$$

Ciò comporta che $|v\rangle = |v'\rangle$ e per distinguere $|\psi\rangle$ e $|\phi\rangle$ è necessario disturbare almeno uno dei due stati.

Nel QKD, un mittente e un destinatario, supponiamo Alice e Bob, ricevono dei qubit e li misurano. Successivamente, comunicano in modo classico i risultati di queste misurazioni per generare dei bit utili per la chiave. Un'eventuale intercettazione del crittoanalista sarà rivelata da Alice e Bob durante una fase di verifica. Ciò produce una stima di quanto il crittoanalista sia a conoscenza circa le informazioni scambiate e, nel caso superi una certa soglia, la procedura viene ripetuta.

4.4.1 Il protocollo BB84

Il protocollo di distribuzione a chiave quantistica Bennett-Brassard è probabilmente uno dei più conosciuti della sua categoria. L'informazione viene trasmessa attraverso dei fotoni polarizzati in un canale quantistico e, qualora non sia necessario trasmettere informazioni quantistiche, si fa ricorso a un canale classico.

La sicurezza del protocollo risiede nella codifica delle informazioni attraverso stati non ortogonali tra di loro. BB84 utilizza due coppie di stati, ognuna coniugata con l'altra. Le coppie ortogonali sono, appunto, le basi utilizzate per la codifica.

Supponiamo che Alice voglia condividere una chiave con Bob senza mai incontrarlo. A tal scopo, genera una sequenza casuale di $4n$ bit e una sequenza casuale di altrettante basi, da associare alla stringa di bit. Per semplicità, indicheremo la base $\{|0\rangle, |1\rangle\}$ come rettilinea $(+)$; mentre $\{|+\rangle, |-\rangle\}$ come base diagonale $(\times)$. La Tabella 4.2 fornisce una sintesi utile per la comprensione.

Ai fini descrittivi, viene impiegata una quantità ridotta di bit; nella pratica si punta a creare chiavi sufficientemente lunghe per garantire una buona sicurezza. Alice ottiene la stringa di bit S_A a cui sono associate le basi B_A. Per esempio:

S_A	0	1	1	0	1	0	0	1
B_A	$+$	$\times$	$+$	$+$	$\times$	$\times$	$\times$	$+$

Facendo riferimento alla Tabella 4.2, Alice procede a polarizzare i fotoni P_A secondo le rispettive basi per rappresentare ogni bit della sua stringa casuale. Il risultato ottenuto è il seguente:

S_A	0	1	1	0	1	0	0	1
B_A	$+$	$\times$	$+$	$+$	$\times$	$\times$	$\times$	$+$
P_A	$\uparrow$	$\searrow$	$\rightarrow$	$\uparrow$	$\searrow$	$\nearrow$	$\nearrow$	$\rightarrow$

Bob prepara un analizzatore per determinare la polarizzazione dei fotoni in arrivo. Poiché le basi utilizzate per la codifica sono due, qualsiasi analizzatore scelga non gli permetterà mai di determinare con esattezza ogni stato dei fotoni. Se, per esempio, Bob scegliesse di misurare un fotone con la base rettilinea, tutti i fotoni polarizzati con la base diagonale sarebbero "persi". Più precisamente, la misura ritornerebbe uno stato nella base rettilinea, la metà delle volte 0 e l'altra metà 1.

Una volta effettuate tutte le misure, Alice e Bob condividono pubblicamente le basi che hanno utilizzato, senza divulgare i valori ottenuti. Ognuno procede a scartare i bit in cui Bob ha usato una base differente. Grazie a questo procedimento, Alice e Bob sono riusciti a ottenere una chiave condivisa, che mediamente risulta avere una dimensione dimezzata, ossia di $2n$ bit. Un esempio delle scelte di Bob è riportato di seguito:

S_A	0	1	1	0	1	0	0	1
B_A	$+$	$\times$	$+$	$+$	$\times$	$\times$	$\times$	$+$
P_A	$\uparrow$	$\searrow$	$\rightarrow$	$\uparrow$	$\searrow$	$\nearrow$	$\nearrow$	$\rightarrow$
B_B	$\bar{\times}$	$\bar{+}$	$+$	$\bar{\times}$	$\times$	$\bar{+}$	$\times$	$+$
P_B	$\searrow$	$\uparrow$	$\rightarrow$	$\nearrow$	$\searrow$	$\rightarrow$	$\nearrow$	$\rightarrow$
Chiave			1		1		0	1

Nella tabella, le basi che Bob ha scelto differentemente sono indicate con $\bar{+}$ e $\bar{\times}$.

Alice annuncia pubblicamente di voler scegliere casualmente n bit dei $2n$ rimanenti. A tal fine, Alice e Bob scelgono separatamente questi valori e successivamente li discutono: nel caso in cui più di p bit risultano essere differenti, il protocollo viene inizializzato nuovamente perché si presuppone una compromissione. Il valore di p è scelto in modo da permettere le procedure di *privacy amplification* e *information reconciliation*, al fine di ottenere una chiave sicura di m bit a fronte degli n rimasti.

Base	0	1
$+$	$\uparrow$	$\rightarrow$
$\times$	$\nearrow$	$\searrow$

Tabella 4.2: Basi e rispettive polarizzazioni.

4.4.2 Intercept resend

Supponiamo che Eve stesse in ascolto durante l'invio dei fotoni sul canale da parte di Alice. Nel caso in cui Eve avesse voluto intercettare le informazioni in transito, avrebbe dovuto disporre di un analizzatore, simile a quello utilizzato da Bob.

Il più semplice tipo di attacco consiste nel cosiddetto *intercept-resend*. In questo attacco, Eve esegue la misura di ogni singolo qubit inviato da Alice, per poi instradare a Bob un qubit con lo stesso stato. È importante notare che eventuali errori introdotti dalle condizioni ambientali del sistema sono sempre possibili.

Se la scelta della base di misura da parte di Eve coincide con quella utilizzata da Alice per la codifica, potrà inviare lo stato a Bob senza perturbare la comunicazione, risultando del tutto trasparente. Nel caso contrario, Bob riceverà una serie di dati alterati e differenti da quelli inviati inizialmente da Alice. A questo si deve aggiungere il fatto che anche Bob potrebbe ottenere un risultato errato con il 50% di probabilità.

Facendo una semplice considerazione, la probabilità che Eve scelga la base scorretta è del 50% (tenendo conto che Alice abbia scelto casualmente la sua). Se Bob misura il qubit intercettato da Eve con la stessa base utilizzata da Alice, otterrebbe un errore nel 50% dei casi. In conclusione, un qubit intercettato genera un errore con una probabilità del $50\% \times 50\% = 25\%$.

Formalmente, se Alice e Bob volessero confrontare a campione n bit della chiave, la probabilità che questi siano corretti risulta pari a

$$P(\text{correttezza}) = \lim_{n\to\infty} \left(\frac{3}{4}\right)^n = 0.$$

Considerando il suo complementare, la probabilità di rilevare la presenza di Eve risulta essere:

$$P(\text{Eve}) = \lim_{n\to\infty} 1 - \left(\frac{3}{4}\right)^n = 1.$$

Se si volesse individuare la presenza di Eve con una probabilità Pr= 0.999999999, Alice e Bob avrebbero bisogno di confrontare 72 bit della chiave, risultando un numero particolarmente ragionevole [23].

4.4.3 Mitigazione delle intercettazioni

Come accennato precedentemente, i bit finali ottenuti da mittente e destinatario sono il risultano di scelte casuali, rumori nella trasmissione ed eventuali intercettazioni. Non è possibile determinare la causa delle discrepanze e si assume che siano dovute a compromissioni della comunicazione.

Una volta stimata la percentuale di errore e se questa non risulta essere soddisfacente, è possibile avviare le procedure di *Privacy amplification* e *Information reconciliation* per migliorare il risultato e ridurre l'informazione in possesso dell'intercettatore [10].

Information reconciliation

Per information reconciliation si intende una tecnica di correzione dell'errore che si attua tra le chiavi del mittente e del destinatario. Riprendendo l'esempio di Alice e Bob riportato in precedenza, essi confrontano pubblicamente le informazioni sulle chiavi, che potrebbero contenere alcune discrepanze a causa di disturbi o errori durante il trasporto. Le parti identificano le discrepanze attraverso un canale pubblico, minimizzando le informazioni scambiate e senza rivelare la chiave segreta.

Uno dei modi più semplici per eseguire questo compito è il metodo a cascata proposto nel 1994. La chiave viene divisa in blocchi, e su ciascun blocco viene eseguito un controllo di parità per determinare se il numero di bit a 1 sia pari o dispari. Solo quando si individua un blocco con una discrepanza si procede con una correzione locale per risolvere il problema. La procedura continua sui blocchi rimanenti fino a una verifica globale che coinvolge l'intera chiave. È da notare che questa procedura fornisce al crittoanalista una quantità minima di informazioni aggiuntive, ovvero la parità dei bit.

Privacy amplification

Con privacy amplification si intende una procedura volta a ridurre, fino a eliminare, l'informazione che un crittoanalista ha ottenuto durante le sue intercettazioni. L'obiettivo è lasciare il crittoanalista con informazioni di scarsa utilità in modo che la chiave possa essere considerata sicura.

Il processo di privacy amplification riduce la possibilità che informazioni indesiderate o compromesse siano presenti nella chiave finale. Per farlo, si utilizza una funzione hash unidirezionale che prende in input la chiave di n bit e restituisce una chiave più corta di lunghezza m.

4.4.4 Il protocollo E91

Un protocollo simile a BB84, che utilizza le correlazioni Einstein-Podolsky-Rosen (EPR), è stato sviluppato da Artur K. Ekert [14] e Bennett, Brassard e Mermin [8] a partire dal 1991.

Il principio di funzionamento si basa sull'utilizzo di particelle entangled. Come accennato in precedenza, questo permette di avere un comportamento *dipendente* tra le particelle nonostante ci sia una notevole distanza a separarle. Per esempio, se viene scelta una base rettilinea, la misura di un fotone avrà una polarizzazione $\uparrow$; inoltre, l'altro fotone entangled dovrà necessariamente avere la stessa polarizzazione, a condizione che venga utilizzata la stessa base per la misura.

Ekert suggerisce di porre la sicurezza dei protocolli che fanno uso di qubit sulle disuguaglianze di Bell. Tali disuguaglianze sono state progettate per dimostrare che alcune

correlazioni tra particelle quantistiche non possono essere spiegate da teorie locali, seguendo il principio di località[2].

Supponiamo che i nostri Alice e Bob utilizzino una terza base. Ciò è necessario per raccogliere informazioni sufficienti a verificare le disuguaglianze di Bell. Questo significa che, oltre a stabilire una chiave segreta, stanno anche raccogliendo informazioni che consentiranno loro di verificare se la sorgente sta realmente emettendo uno stato entangled anziché stati separabili.

Il protocollo di Ekert rispecchia in maniera più accurata quella che potrebbe essere una potenziale implementazione realmente utilizzabile. Infatti, gli scambi di chiavi avvengono a distanze che sarebbero di forte limitazione per la trasmissione affidabile dei qubit. L'utilizzo di una sorgente interposta tra mittente e destinatario, come un satellite, riuscirebbe a mitigare questo inconveniente.

Anziché fare affidamento sul qubit inviato dal mittente, il protocollo è predisposto affinché la sorgente emetta una coppia di fotoni in stato entangled, ad esempio della forma

$$|\psi\rangle = |+\rangle = \frac{|01\rangle + |10\rangle}{\sqrt{2}}. \tag{4.1}$$

Con questa equazione si denota lo stato I (primo ket), con i qubit che hanno rispettivamente valori 0 e 1; similmente, lo stato II (secondo ket) presenta i qubit scambiati.

Sia Alice che Bob scelgono casualmente una delle tre basi per misurare le particelle in arrivo dalla sorgente. Si deduce facilmente che la probabilità che entrambi scelgano una stessa base si riduce da 1/2 a 1/3. Nel caso in cui entrambi scelgano una base uguale e Alice misuri uno 0, il sistema collasserebbe nello stato I, e la misura di Bob sarà sicuramente 1. Similmente, nel caso in cui Alice misuri 1, Bob sicuramente misurerà un valore pari a 0.

La situazione diventa assai più interessante nel caso in cui essi non scelgano la stessa base di misurazione: l'esito della misura di Bob sarà del tutto casuale. Per esempio, se Alice sceglie una base diagonale e misura 1, Bob avrà un'equa probabilità di ottenere come risultato 0 o 1 se sceglie una base rettilinea. Ciò suggerisce che Bob "conosce" il modo in cui Alice ha effettuato la misurazione e può, a tal proposito, prendere delle contromisure. Affinché Alice e Bob possano scartare le misure ottenute con basi incompatibili, devono annunciare pubblicamente le loro scelte, come accadeva per BB84.

Il funzionamento di questo protocollo utilizzando delle coppie EPR di fotoni è illustrato di seguito:

- Una volta preparata una coppia di fotoni entangled, il primo viene inviato ad Alice mentre il secondo lo riceve Bob.
- Alice misura la polarizzazione del fotone con una base casuale.

[2] Il principio di località afferma che le informazioni non possono viaggiare istantaneamente al di là di una certa soglia di velocità, limitando la connessione diretta e immediata tra eventi distanti nello spazio.

- Bob esegue la misura allo stesso modo sul fotone che ha ricevuto.
- Alice e Bob condividono pubblicamente le basi utilizzate per le misure e scartano tutte le coppie che hanno basi differenti.
- Alice e Bob mappano i fotoni rimanenti in bit classici, come mostrato precedentemente in Tabella 4.2.

Una differenza importante di questo protocollo risiede nel fatto che la chiave ottenuta tramite BB84 deve essere conservata in memorie tradizionali fino all'utilizzo. Utilizzando un "approccio entangled", Alice e Bob possono conservare i loro fotoni fino al momento in cui non decidono di creare una chiave, eliminando la questione di un'archiviazione non sicura. Inoltre, per evitare che Eve possa essere l'ente che distribuisce i fotoni, E91 utilizza solo qubit in stato massimamente entangled[3], come quello indicato nell'equazione (4.1).

Nel caso in cui Eve volesse impersonare la fonte delle particelle entangled, dovrebbe scegliere una base di misurazione; la misura sarebbe casuale e distruggerebbe la particella. Anche se decidesse di inoltrare a Bob una particella coerente alla sua misura, al momento della ricezione questa verrebbe rigettata qualora non ci sia una correlazione con quella di Alice. Da ciò si evince la notevole utilità delle particelle entangled nel fornire una sicurezza aggiuntiva.

[3] Uno stato massimamente entangled è uno dei quattro stati di Bell, utilizzati in questi contesti proprio per le loro importanti proprietà. Questi sono $\Phi^+ = |00\rangle+|11\rangle/\sqrt{2}$, $\Phi^- = |00\rangle-|11\rangle/\sqrt{2}$, $\Psi^+ = |01\rangle+|10\rangle/\sqrt{2}$ e $\Psi^- = |01\rangle-|10\rangle/\sqrt{2}$.

Capitolo 5

Sfide e prospettive

Il quinto e ultimo capitolo di questa tesi si propone di esaminare il panorama attuale e futuro della crittografia quantistica. In questo contesto, verranno esaminate le sfide che ostacolano la sua adozione su scala estesa, oltre alle motivazioni alla base della sua emergenza come risposta alle minacce computazionali avanzate, come evidenziato dall'algoritmo di Shor. Contestualmente, il capitolo esplorerà le implementazioni pratiche esistenti, offrendo uno sguardo attento allo stato corrente della crittografia quantistica e alle prospettive future.

5.1 Sfide attuali

5.1.1 Comunicazione satellitare

L'opportunità di utilizzare protocolli di distribuzione a chiave quantistica su lunghe distanze è stata oggetto di approfonditi studi a partire dal decennio appena trascorso. Inizialmente, si è considerata l'estensione della comunicazione quantistica anche nello spazio, basandosi sull'applicazione dei principi della computazione quantistica in questo specifico contesto[1].

L'invio di payload[2] verso la Stazione Spaziale Internazionale (ISS) e lo sviluppo di satelliti autonomi hanno ricevuto sostegno grazie ai primi risultati positivi, avvenuti su grandi distanze e nello spazio. La sfida, infatti, non è affatto scontata, poiché la trasmissione di fotoni con specifiche sulle lunghezze d'onda risulta decisamente più complessa rispetto alla comunicazione ottica tradizionale.

[1] Questa sezione fa riferimento agli sviluppi di [30].

[2] In informatica, indica la parte di dati effettivamente trasmessi che è destinata all'utilizzatore, in contrasto con i metadati e con gli header che servono esclusivamente a far funzionare il protocollo di comunicazione.

A partire dal 2003, con una campagna sperimentale presso l'Osservatorio Laser di Matera in Italia, è stata dimostrata la fattibilità dello scambio di singoli fotoni tra un satellite in orbita bassa (LEO, Low Earth Orbit) e il suolo, anche senza una sorgente attiva in orbita. Questa dimostrazione è stata possibile grazie all'utilizzo di satelliti dotati di retro-riflettori ottici, ai quali veniva inviato un treno di impulsi opportunamente calibrato. L'energia raccolta veniva poi trasmessa sulla Terra in modo tale da ottenere uno stato coerente descritto da un singolo fotone. Tale tecnica è stata successivamente estesa alla comunicazione quantistica utilizzando diversi gradi di libertà.

Lo sviluppo di una QKD globale è stato considerato fin dai suoi albori come una soluzione per collegare efficacemente reti già unite attraverso una fibra ottica via terra. Nonostante l'interesse per questo scenario, la realizzazione di simili satelliti è stata sospesa in Europa e negli USA. D'altro canto, in stati come Cina e Giappone, si è manifestato un interesse che si è concretizzato con il lancio dei satelliti giapponesi SOTA (2014) e quelli cinesi Micius (2015). Questo ha promosso ulteriori iniziative europee focalizzate su nanosatelliti o cubesat[3], con l'obiettivo di sviluppare componenti spaziali efficienti e di dimensioni ridotte.

5.1.2 Possibilità della comunicazione quantistica

La ricerca ha aperto la strada a nuovi protocolli basati sulla comunicazione quantistica. Tra quelli progettati per operare su lunghe distanze, spiccano la firma digitale quantistica (QDS, Quantum Digital Signature) e il calcolo quantistico cieco (BQC, Blind Quantum Computation).

La QDS rappresenta l'equivalente quantomeccanica di una firma digitale, volta a garantire principi come la non ripudiabilità, inalterabilità e trasferibilità di una firma. Recentemente, è stato dimostrato un protocollo di firma digitale quantistica a lunga distanza senza l'uso di canali sicuri, riuscendo a firmare con successo un messaggio di un bit attraverso una fibra ottica di 102 km.

Un secondo esempio è la BQC, in cui un client invia al server uno stato quantico $|\psi\rangle$ che codifica sia l'algoritmo scelto sia l'input [15]. Questo protocollo offre privacy agli utilizzatori di un computer quantistico in cloud, consentendo loro di mantenere nascosti i risultati della computazione. Le modalità di comunicazione satellitare descritte precedentemente possono essere impiegate per l'implementazione di questo protocollo.

La comunicazione quantistica nello spazio a lunga distanza costituisce l'elemento chiave per effettuare test fondamentali della meccanica quantistica in condizioni sconosciute. Esperimenti significativi, come la violazione delle disuguaglianze di Bell e l'esperimento di

[3] Un cubesat è un tipo di satellite miniaturizzato avente forma cubica, sviluppato a partire dal 1999 dall'Università statale politecnica della California e dall'Università di Stanford.

scelta ritardata di Wheeler, rappresentano esempi paradigmatici delle possibilità offerte dalla comunicazione quantistica nello spazio.

Gli esperimenti sulla violazione delle disuguaglianze di Bell dimostrano che un modello di variabile nascosta locale non può riprodurre i risultati sperimentali ottenibili con stati entangled. Attualmente, queste disuguaglianze sono utilizzate per verificare la persistenza dell'entanglement tra due particelle. Nel 2017, il satellite Micius ha permesso di dimostrare che due particelle risultassero ancora entangled a una distanza record di 1200 km.

La meccanica quantistica prevede che l'entanglement quantistico possa essere misurato a qualsiasi distanza. La disponibilità della comunicazione quantistica nello spazio consente ora di estendere questi limiti a distanze sempre maggiori. Ad esempio, utilizzando una sorgente entangled su un satellite geostazionario, che invia due fotoni alla Terra, potrebbe essere possibile aumentare la distanza tra i due fotoni entangled di un ordine di grandezza.

Un ulteriore esempio è l'esperimento di scelta ritardata di Wheeler, che considera la dualità onda-particella. In questo esperimento, viene evidenziato che non è possibile rivelare contemporaneamente le proprietà di onda e particella di un oggetto quantistico. L'esperimento di scelta ritardata è stato esteso allo spazio, sfruttando il grado temporale di libertà dei fotoni riflessi da un satellite in orbita. L'esperimento ha confermato la validità del modello onda-particella su distanze fino a 3500 km, molto superiori a tutti gli esperimenti precedenti.

5.2 L'algoritmo di Shor

All'interno del panorama della crittografia quantistica, una delle pietre miliari concettuali è rappresentata dall'algoritmo di Shor [33], capace di rivelare la vulnerabilità degli algoritmi di crittografia classica di fronte alla potenza computazionale dei computer quantistici.

Sviluppato da Peter Shor nel 1994, l'algoritmo di Shor assume un'importanza rilevante poiché è in grado di fattorizzare grandi numeri in tempi notevolmente inferiori rispetto agli algoritmi classici. Tale capacità, se implementata su un computer quantistico, rappresenterebbe una minaccia diretta alla sicurezza di algoritmi di crittografia a chiave pubblica, come RSA.

5.2.1 Problema del logaritmo discreto

Il problema del logaritmo discreto si pone come segue: dato un numero primo p, un generatore g del gruppo moltiplicativo[4] $\mathbb{Z}_p^*$ e un elemento h appartenente al gruppo, trovare

[4] Si faccia riferimento a quanto già riportato in Sezione 2.3.2.

l'esponente x tale che $g^x \equiv h \bmod p$[5].

In termini più semplici, il problema chiede di trovare l'esponente x che, elevando g alla sua potenza, produce h considerato in modulo p. Questo problema è notoriamente difficile da risolvere in modo efficiente su un computer classico quando p è un numero primo grande e g è scelto correttamente.

Come si è potuto osservare, questo problema è utilizzato proprio per implementare gli algoritmi crittografici moderni. Il tempo non polinomiale richiesto per la sua soluzione diventa garanzia della sicurezza di questi algoritmi. Shor suggerisce un nuovo approccio, che permetterebbe di risolvere il problema del logaritmo discreto in tempo polinomiale, compromettendo potenzialmente i sistemi crittografici moderni.

5.2.2 Idea di funzionamento

Si fornisce di seguito una panoramica semplificata che illustra il funzionamento dell'algoritmo di Shor.

1. Selezione di un numero da fattorizzare

L'algoritmo inizia con la scelta di un numero intero n che si desidera fattorizzare. Tale numero è il prodotto di due numeri primi distinti.

2. Selezione di un numero casuale

Si sceglie un numero casuale a compreso tra 2 e $n - 1$.

3. Calcolo del MCD

Si calcola il massimo comune divisore tra a e n utilizzando un algoritmo classico. Se tale numero è diverso da 1, allora significa che questo è un fattore di n. Per questo motivo l'algoritmo si conclude.

4. Ricerca del periodo

Si cerca la lunghezza r del periodo della funzione $f(x) = a^x \bmod n$. La trasformata di Fourier quantistica [20] è in grado di trovare r in modo significativamente più veloce rispetto agli algoritmi classici.

5. Verifica della parità

Se r è dispari o se $a^{r/2} \equiv -1 \bmod n$, si ritorna al secondo passo scegliendo un nuovo a.

6. Fattorizzazione

Se tutte le condizioni sono soddisfatte, è possibile utilizzare r per calcolare i fattori di n in maniera opportuna.

---

[5] Si utilizza il concetto di congruenza dell'aritmetica modulare. Diciamo che $a \equiv b \bmod n$, letto "a congruo a b modulo n", se $a - b$ è un multiplo di n.

Esempio

Consideriamo il numero $n = 15$, che soddisfa i criteri di scelta dell'algoritmo di Shor in quanto non è primo né pari. Supponendo di scegliere $a = 7$ (la condizione $1 < a < n$ è rispettata), il massimo comune divisore tra 7 e 15 è 1. Una volta verificato ciò, occorre trovare il più piccolo intero positivo r tale che se $f(x) = a^x \bmod n$, allora $f(x) = f(x + r)$; in questo caso, r identifica il periodo.

Procediamo iterativamente finché $(q \cdot a) \bmod n = 1$, iniziando con $q = 1$. Nel caso in cui non si ottenga 1, continuiamo utilizzando il resto appena ricavato come nuovo valore di q. Nell'esempio, otteniamo:

1) $1 \times 7 \bmod 15 = 7$.

2) $7 \times 7 \bmod 15 = 4$.

3) $4 \times 7 \bmod 15 = 13$.

4) $13 \times 7 \bmod 15 = 1$.

Terminata questa procedura, assegniamo a r il numero di passaggi eseguiti, in questo caso $r = 4$. Dato che r non è dispari e $7^{4/2} \not\equiv -1 \bmod 15$, possiamo trovare i fattori f_1 e f_2 di n. Consideriamo il resto p al passaggio $r/2$, ossia $p = 4$. Sapendo che $\gcd(a, b)$ definisce il massimo comune divisore tra a e b, otteniamo:

- $f_1 = \gcd(p + 1, n) = \gcd(5, 15) = 5$.
- $f_2 = \gcd(p - 1, n) = \gcd(3, 15) = 3$.

5.3 Crittografia Post-Quantistica

I recenti avanzamenti[6] nell'ambito della tecnologia quantistica hanno dato vita alla crittografia post-quantistica (PQC, Post-Quantum Cryptography). Questa nuova categoria di sistemi crittografici è concepita per resistere matematicamente agli attacchi eseguiti tramite computer quantistici, garantendone al contempo la realizzazione tramite l'utilizzo della tecnologia a semiconduttori convenzionale [9].

Data la rapida evoluzione del settore, i sistemi a crittografia asimmetrica potrebbero diventare vulnerabili a una realizzazione dell'algoritmo di Shor in tempo polinomiale. Di conseguenza, è imperativo esplorare alternative capaci di superare questa sfida.

Il National Institute of Standards and Technology ha avviato il processo di standardizzazione per la crittografia post-quantistica [36]. Questo procedimento ha seguito

[6] In questa sezione si fa riferimento agli sviluppi di [39].

uno sviluppo graduale: nella fase iniziale di valutazione, è stata dedicata particolare attenzione alla sicurezza, con indicazioni sul software consigliato. Solo successivamente sono stati sviluppati sistemi ottimizzati per sfruttare appieno le potenzialità dei moderni microprocessori, culminando infine in soluzioni hardware.

L'implementazione degli algoritmi post-quantistici si presenta come una sfida complessa. Una delle principali difficoltà risiede nella notevole complessità matematica che questi algoritmi devono affrontare. È rilevante sottolineare che queste specifiche sono definite dai crittografi stessi, i quali spesso prioritizzano la sicurezza rispetto alla facilità di implementazione. Allo stesso modo, coloro che si occupano della realizzazione hardware potrebbe non possedere sempre le competenze necessarie per comprendere appieno tali indicazioni.

In linea generale, le operazioni richieste dalla PQC differiscono da quelle di un sistema a chiave pubblica come RSA o ECC. Come già illustrato, sono le operazioni su numeri interi di considerevole dimensione a determinare la complessità di questi sistemi. Al contrario, gli schemi post-quantistici possono beneficiare di una maggiore parallelizzazione e trarre vantaggio delle ottimizzazioni sviluppate nei vari ambiti scientifici e ingegneristici.

Conclusioni

La presente tesi si propone come un documento informativo mirato a esplorare la crittografia quantistica, un elemento sempre più centrale nel contesto degli studi e della ricerca crittografica. Per raggiungere questo obiettivo, la struttura del lavoro è stata concepita fornire al lettore gli strumenti necessari a comprendere i molteplici concetti alla base di questo campo.

Inizialmente, è stata fornita un'ampia panoramica sulla crittografia classica, introducendo i principi essenziali che i sistemi crittografici devono garantire: confidenzialità, integrità e autenticazione dell'informazione trasmessa. Associati ai concetti di cifratura simmetrica e asimmetrica, questi costituiscono un punto di partenza cruciale per affrontare l'evoluzione del panorama crittografico.

Sono stati presentati concetti avanzati, definendo strumenti rilevanti per l'applicazione degli attuali algoritmi crittografici. La casualità e l'analisi di come una sequenza binaria possa acquisire caratteristiche casuali, insieme a sofisticati strumenti matematici, rappresentano il nucleo della crittografia contemporanea. Ciò ha permesso la definizione di protocolli significativi, come RSA e quelli basati sulle curve ellittiche, che garantiscono la sicurezza delle comunicazioni.

Prima di affrontare il cuore di questa tesi, è stato doveroso esaminare ciò che l'ha resa possibile: la meccanica quantistica. Questo ramo della fisica, di natura intrinsecamente controintuitiva, è stato introdotto attraverso una sezione sui fondamenti dell'algebra lineare, per agevolare la comprensione di ciò che la meccanica quantistica cerca di spiegare, come il concetto di funzioni d'onda e le misure a essa associate.

Il fulcro di questo lavoro è stato il quarto capitolo, che ha delineato il funzionamento dei qubit, elementi imprescindibili per l'applicazione corretta dei protocolli quantistici come BB84 e E91. Attraverso tale capitolo, il lettore ha potuto apprezzare le distinzioni sostanziali rispetto alla crittografia tradizionale.

Infine, si è analizzato l'avanzamento della crittografia quantistica e le tecnologie che offrono potenziale per implementazioni future. I progressi ottenuti, unitamente a quelli della crittografia post-quantistica, si confrontano con le sfide delineate da Shor, il quale ha illustrato una minaccia significativa per la crittografia moderna. Affrontare tali sfide con successo diventa determinante per garantire la sicurezza delle comunicazioni future.

Appendice A

Elementi di probabilità

Spazi di probabilità

Di seguito sono definite alcune nozioni fondamentali.

Definizione (Fenomeno)**.** Un fenomeno è qualcosa che accade e alla fine produce un risultato.

Definizione (Evento)**.** Un evento è un insieme di risultati.

Definizione (Valutazione di probabilità)**.** Una valutazione di probabilità è una funzione che associa a ogni evento un numero, tanto maggiore quanto consideriamo probabile che l'evento si verifichi.

Per quanto riguarda gli eventi, desideriamo che (c_1):

1. $\emptyset, \Omega = \{\text{tutti i possibili eventi}\}$ siano eventi.
2. Unioni e intersezioni di eventi (anche numerabili) siano ancora eventi.
3. Il complementare di un evento sia un evento.

La valutazione di probabilità deve soddisfare (c_2):

1. $P(\Omega) = 1$.
2. Se $A_1, A_2, \dots$ sono eventi disgiunti, allora $P(A_1 \cup A_2 \cup \dots) = P(A_1) + P(A_2) + \dots$.

Definizione (Spazio di probabilità)**.** Uno spazio di probabilità è:

- Un insieme Ω.
- Un insieme di sottoinsiemi di Ω che soddisfa (c_1).
- Una valutazione di probabilità che soddisfa (c_2).

Probabilità condizionale

Supponiamo di avere uno spazio di probabilità Ω con i suoi eventi e la valutazione P. Sia A un evento con $P(A) \neq 0$. Costruiamo un nuovo spazio di probabilità con lo stesso Ω e gli stessi eventi ma con probabilità:

$$P'(E) = \frac{P(E \cap A)}{P(A)}. \tag{A.1}$$

Interpretiamo $P'(E)$ come la probabilità che accada E sapendo che il risultato si trova in A. Tale probabilità la indicheremo con:

Definizione (Probabilità condizionata)**.**

$$P'(E) = P(E|A).$$

Eventi indipendenti

Definizione (Indipendenza)**.** Siano A e B due eventi. Questi si dicono indipendenti se

$$P(A \cap B) = P(A) \cdot P(B). \tag{A.2}$$

Variabili aleatorie

Definizione (Variabile aleatoria)**.** Una variabile aleatoria (Ω, A, P) è una funzione

$$X : \Omega \to \mathbb{R},$$

tale che $\{\omega \in \Omega | X \leq a\}$ $\forall a \in \mathbb{R}$, vale a dire $\{\omega \in \Omega | X \leq a\} \in A$.

Appendice B

Aritmetica delle curve ellittiche

Questa appendice ha lo scopo di fornire alcune notazioni e concetti utili per comprendere meglio le operazioni che possono essere eseguite con le curve ellittiche.

Definizione (Gruppo)**.** $(G, *)$ è un gruppo se e solo se valgono le seguenti:

- Proprietà associativa, $\forall a, b, c \in G : (ab)c = a(bc)$.
- Elemento neutro, $\exists e \in G, \forall a \in G : ae = ea = a$.
- Inverso, $\forall a \in G, \exists a' \in G : aa' = a'a = e$.

Definizione (Gruppo abeliano)**.** $(G, *)$ è un gruppo abeliano se e solo se $(G, *)$ è un gruppo e vale la commutatività, ossia che $\forall a, b \in G : ab = ba$.

Definizione (Campo)**.** Un campo K è un insieme non vuoto dotato di due operazioni binarie, chiamate somma e prodotto, che soddisfano la proprietà associativa e commutativa e ammettono l'esistenza di un elemento neutro e dell'inverso di ciascun elemento, fatta eccezione per lo zero.

$(K, +, *)$ è un campo se e solo se:

- $(K, +)$ è un gruppo abeliano con elemento neutro 0.
- $(K \setminus \{0\}, *)$ è un gruppo abeliano con elemento neutro 1.
- La moltiplicazione è distributiva rispetto all'addizione, vale a dire $a(b + c) = (ab) + (ac) \quad \forall a, b, c \in G$.

Punto all'infinito

Una curva ellittica definita su un campo K è l'insieme dei punti (x, y) che soddisfano l'equazione $y^2 = x^3 + ax + b$, oltre al punto neutro.

Assumiamo che la curva sia definita sul campo $\mathbb{R}$ dei numeri reali e che sia dunque rappresentabile sul piano cartesiano.

L'elemento neutro O è il punto all'infinito sull'asse delle ordinate. La curva sarà costituita dall'insieme $E(a, b)$ secondo questa forma:

$$E(a, b) = \{(x, y) \in \mathbb{R}^2 : y^2 = x^3 + ax + b\}.$$

Somma su Curve Ellittiche

Dati tre punti P, Q, R di una curva ellittica $E(a, b)$, se questi sono allineati allora la loro somma è al punto all'infinito, ossia:

$$P + Q + R = O.$$

Da questo si può ricavare che la somma tra P e Q si può effettuare come segue (Figura B.1):

- Si considera la retta passante per P e Q oppure la tangente nel caso $P = Q$.
- Si determina l'intersezione tra la retta passante per P e Q e la curva, oppure tra la tangente in P e la curva nel caso $P = Q$.
- Si definisce la somma di P e Q come il punto simmetrico a R rispetto all'asse delle ascisse, ovvero: $P + Q = -R$.

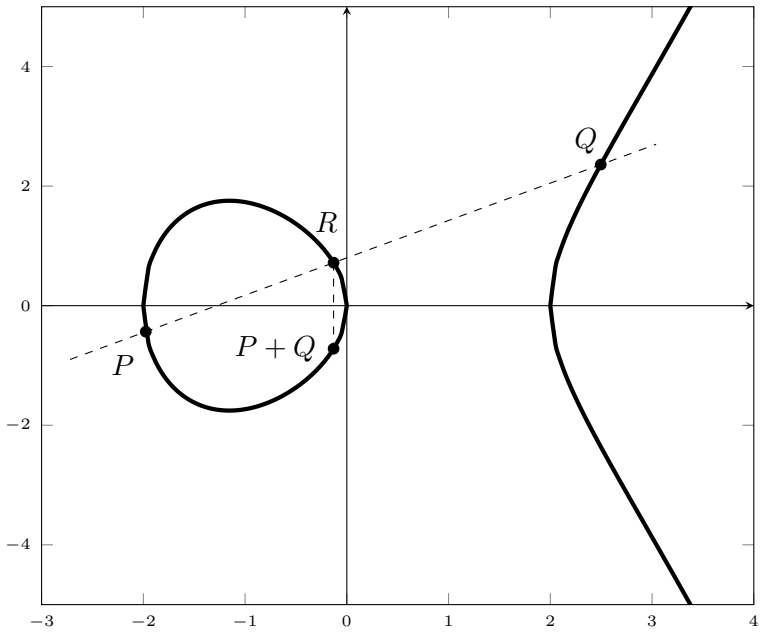


Figura B.1: Somma di due punti.

Una volta definita la somma di due punti su una curva, è possibile definire anche la moltiplicazione di un punto P per uno scalare k. Si può interpretare la moltiplicazione come la somma di k volte il punto P. Per esempio, con $k = 2$, possiamo dire che $2P = P + P$.

Logaritmo discreto

Per definire un sistema crittografico a chiave pubblica è necessario individuare una buona funzione one-way con trapdoor che ne garantisca la sicurezza.

Tale funzione la si può ottenere dall'analogia del logaritmo discreto nell'algebra modulare. In effetti, l'operazione di addizione di punti di una curva ellittica su un campo finito presenta delle analogie con l'operazione di prodotto modulare.

Dato un intero positivo k, esiste un'analogia tra l'elevamento alla potenza k di un intero in modulo con la moltiplicazione scalare per k di un punto P di una curva ellittica, operazione che consiste nel sommare P con se stesso k volte. Entrambe le operazioni si possono eseguire in tempo polinomiale.

Si consideri l'operazione inversa, che costituisce il problema del logaritmo discreto per le curve ellittiche. Dati due punti P e Q, si vuole trovare il più piccolo intero k tale che $Q = kP$. Il problema del logaritmo discreto per le curve ellittiche risulta computazionalmente difficile perché tutti gli algoritmi noti per risolverlo hanno complessità esponenziale [25].

Bibliografia

[1] Jean-Philippe Aumasson. *Serious cryptography: a practical introduction to modern encryption*. No Starch Press, 2017. ISBN: 978-1-59327-826-7.

[2] Alessandro Bagagli. "L'arte della segretezza, tra mondo classico e quantistico". Tesi di laurea. Alma Mater Studiorum - Università di Bologna, 2016.

[3] Rajkumar Banoth e Rekha Regar. "Classical and Modern Cryptography for Beginners". In: Springer Nature Switzerland, 2023, pp. 1–46. ISBN: 978-3-031-32959-3. DOI: 10.1007/978-3-031-32959-3_1.

[4] Craig P. Bauer. *Secret History: The Story of Cryptology (2nd edition)*. Chapman e Hall/CRC, 2021. ISBN: 978-1-315-16253-9. DOI: 10.1201/9781315162539.

[5] Mihir Bellare e Phillip Rogaway. *Introduction to modern cryptography*. 2005.

[6] Mike Bendel. *Hackers Describe PS3 Security As Epic Fail, Gain Unrestricted Access*. 2010. URL: https://www.exophase.com/20540/hackers-describe-ps3-security-as-epic-fail-gain-unrestricted-access/ (visitato il 20/02/2024).

[7] Charles H. Bennett e Gilles Brassard. "Quantum cryptography: Public key distribution and coin tossing". In: *Theoretical computer science* 560 (2014), pp. 7–11. DOI: 10.1016/j.tcs.2014.05.025.

[8] Charles H. Bennett, Gilles Brassard e N. David Mermin. "Quantum cryptography without Bell's theorem". In: *Physical review letters* 68.5 (1992), p. 557. DOI: 10.1103/PhysRevLett.68.557.

[9] Daniel J. Bernstein. *Introduction to post-quantum cryptography*. A cura di Daniel J. Bernstein, Johannes Buchmann e Erik Dahmen. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. ISBN: 978-3-540-88702-7. DOI: 10.1007/978-3-540-88702-7.

[10] Christian Cachin e Ueli M. Maurer. "Linking information reconciliation and privacy amplification". In: *journal of Cryptology* 10.2 (1997), pp. 97–110. DOI: 10.1007/s001459900023.

[11] Paul Adrien Maurice Dirac. *The principles of quantum mechanics*. 27. Oxford university press, 1981.

[12] U. Eichmann et al. “Young’s interference experiment with light scattered from two atoms”. In: *Phys. Rev. Lett.* 70 (16 1993), pp. 2359–2362. DOI: 10.1103/PhysRevLett.70.2359.

[13] A. Einstein, B. Podolsky e N. Rosen. “Can Quantum-Mechanical Description of Physical Reality Be Considered Complete?” In: *Phys. Rev.* 47 (10 1935), pp. 777–780. DOI: 10.1103/PhysRev.47.777.

[14] Artur K. Ekert. “Quantum cryptography based on Bell’s theorem”. In: *Physical review letters* 67.6 (1991), p. 661. DOI: 10.1103/PhysRevLett.67.661.

[15] Joseph F. Fitzsimons. *Private quantum computation: An introduction to blind quantum computing and related protocols*. 2016. DOI: 10.48550/arXiv.1611.10107.

[16] Fabiola Genevois, Alessio Nunzi e Federico Russo. *I cifrari perfetti*. Dispense del corso di Crittografia (Università Roma Tre). 2006.

[17] David J. Griffiths e Darrell F. Schroeter. *Introduction to quantum mechanics*. Cambridge university press, 2018. ISBN: 978-1-31699-543-3. DOI: 10.1017/9781316995433.

[18] Nikolina Ilic. “The ekert protocol”. In: *Journal of Phy334* 1 (2007), p. 22.

[19] Darshana Jayasinghe et al. “Advanced modes in AES: Are they safe from power analysis based side channel attacks?” In: *2014 IEEE 32nd International Conference on Computer Design (ICCD)*. 2014, pp. 173–180. DOI: 10.1109/ICCD.2014.6974678.

[20] Marc Kaplan et al. “Breaking Symmetric Cryptosystems Using Quantum Period Finding”. In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2016, pp. 207–237. ISBN: 978-3-66253-008-5. DOI: 10.1007/978-3-662-53008-5_8.

[21] Vivek Kapoor, Vivek Sonny Abraham e Ramesh Singh. “Elliptic curve cryptography”. In: *Ubiquity* 2008.May (2008), pp. 1–8. DOI: 10.1145/1386853.1378356.

[22] Seema S. Kute e Chitra G. Desai. “Quantum cryptography: a review”. In: *Indian Journal of Science and Technology* 10.3 (2017), pp. 1–5.

[23] Veronica Malizia. “Il protocollo BB84 per la distribuzione quantistica delle chiavi”. Tesi di laurea. Alma Mater Studiorum - Università di Bologna, 2016.

[24] Luciano Margara. *Cifrari Perfetti: One-Time Pad*. Dispense del corso di Crittografia. 2022.

[25] Luciano Margara. *Crittografia su Curve Ellittiche*. Dispense del corso di Crittografia. 2022.

[26] Alfred J Menezes, Paul C. Van Oorschot e Scott A. Vanstone. *Handbook of applied cryptography*. CRC press, 1997. ISBN: 978-0-8493-8523-0. DOI: 10.1201/9781439821916.

[27] Saptarshi Mitra et al. “Quantum cryptography: Overview, security issues and future challenges”. In: *2017 4th International Conference on Opto-Electronics and Applied Optics (Optronix)*. 2017, pp. 1–7. DOI: 10.1109/OPTRONIX.2017.8350006.

[28] Michael A. Nielsen e Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010. ISBN: 978-1-107-00217-3. DOI: 10.1017/CBO9780511976667.

[29] Okta. *HMAC (Hash-Based Message Authentication Codes) Definition*. 2023. URL: https://www.okta.com/identity-101/hmac/ (visitato il 09/01/2024).

[30] S. Pirandola et al. “Advances in quantum cryptography”. In: *Adv. Opt. Photon.* 12.4 (2020), pp. 1012–1236. DOI: 10.1364/AOP.361502.

[31] Lorenza Principe. “Crittografia Quantistica: Il Protocollo BB84”. Tesi di laurea. Alma Mater Studiorum - Università di Bologna, 2019.

[32] Rosaria Rota. *Crittografia quantistica*. Dispense del corso di Crittografia (Università Roma Tre). 2004.

[33] P.W. Shor. “Algorithms for quantum computation: discrete logarithms and factoring”. In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. 1994, pp. 124–134. DOI: 10.1109/SFCS.1994.365700.

[34] Nigel P. Smart. *Cryptography made simple*. Springer Nature Switzerland, 2016. ISBN: 978-3-319-21935-6. DOI: 10.1007/978-3-319-21936-3.

[35] Information Technology Laboratory (National Institute of Standards e Technology). “Announcing the Advanced Encryption Standard (AES)”. In: *Federal Information Processing Standards Publication* 197.1 (2001).

[36] Information Technology Laboratory (National Institute of Standards e Technology). *Post-Quantum Cryptography Standardization*. 2017. URL: https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization (visitato il 30/01/2024).

[37] James M. Turner. “The keyed-hash message authentication code (hmac)”. In: *Federal Information Processing Standards Publication* 198.1 (2008), pp. 1–13. DOI: 10.6028/nist.fips.198-1.

[38] Tan Xiaoqing. “Introduction to Quantum Cryptography”. In: a cura di Jaydip Sen. Vol. Theory and Practice of Cryptography and Network Security Protocols and Technologies. InTech, 2013. Cap. 5, pp. 111–146. DOI: 10.5772/56092.

[39] Jiafeng Xie et al. “Special Session: The Recent Advance in Hardware Implementation of Post-Quantum Cryptography”. In: *2020 IEEE 38th VLSI Test Symposium (VTS)*. 2020, pp. 1–10. DOI: 10.1109/VTS48691.2020.9107585.

Ringraziamenti

Questa tesi rappresenta la conclusione tangibile del mio percorso di studi. Mi sono iscritto a questa facoltà con l'obiettivo di ottenere qualcosa di duraturo nel tempo, e oggi posso dire di averlo raggiunto. Le diverse esperienze quotidiane, con nuovi volti da conoscere, sia tra gli studenti come me che tra i docenti sempre disponibili, hanno arricchito il mio percorso di apprendimento.

Innanzitutto, desidero esprimere la mia riconoscenza al professor Margara per avermi dato l'opportunità di sviluppare una tesi su un argomento tanto complesso quanto affascinante, come quello della crittografia quantistica. Lo ringrazio soprattutto per la sua disponibilità costante: sin dall'inizio, ha fornito preziose indicazioni e mi ha guidato in questo importante passo finale.

Un sentito ringraziamento va al mio amico Alex, con cui ho condiviso gli stessi corsi ogni giorno. Abbiamo affrontato insieme lo studio e la preparazione degli esami, dedicando interi pomeriggi a sbobinare le diapositive. Il suo sostegno è stato prezioso, specialmente nei momenti più difficili, riuscendo a rendere più leggeri anche i viaggi in treno più noiosi.

Un profondo senso di gratitudine è rivolto ai miei genitori. In ogni momento del mio percorso hanno sostenuto le mie scelte senza mai dubitare di me. Ogni volta che dovevo affrontare un esame ci sono stati, sia durante la preoccupazione iniziale che dopo, per festeggiare questi piccoli successi. E ci sono ora, pronti ad accompagnarmi alla mia laurea.

L'ultimo, ma non meno importante, pensiero va a Matteo, il mio ragazzo. Ci siamo conosciuti al terzo anno, quando era uno studente di fisica alla triennale di Bologna, e posso affermare con sicurezza che questa tesi non sarebbe stata possibile senza di lui. La sua pazienza nello spiegarmi i vari concetti di meccanica quantistica è stata infinita. Mi ha sostenuto ogni giorno, ascoltandomi nonostante la mia profonda testardaggine, e ha costantemente cercato di spronarmi a dare il massimo, portando alla luce il mio lato migliore.