Crittografia Post-Quantum

Olivieri Michele

January 30, 2026

Contents

1 Introduzione

1.1 Contesto e motivazioni

Per comprendere l'importanza della crittografia post-quantistica, è fondamentale analizzare il contesto in cui essa si inserisce e le motivazioni che ne hanno guidato lo sviluppo.

La crittografia classica, come visto a lezione, pone le sue fondamenta su problemi computazionalmente difficili per i quali non esistono algoritmi efficienti in grado di risolverli in tempi polinomiali.

Tuttavia, l'emergere dei computer quantistici ha introdotto una nuova dimensione nel panorama della sicurezza informatica. Questi dispositivi sfruttando i principi della meccanica quantistica sono in grado di eseguire calcoli in modo radicalmente diverso rispetto ai computer classici, rendendoli potenzialmente capaci di risolvere in tempi polinomiali quei problemi matematici che risultano intrattabili per le macchine convenzionali.

1.2 Minaccia dei computer quantistici: Perché è necessaria?

L'ipotesi di una minaccia quantistica emerge già nel ventesimo secolo, quando nel 1994 l'algoritmo di Shor dimostra che un computer quantistico sufficientemente potente potrebbe fattorizzare grandi numeri interi e calcolare logaritmi discreti in tempo polinomiale. Questo rende vulnerabili algoritmi come RSA, ECC e DSA, che costituiscono la base della sicurezza informatica moderna.

Per molti anni il problema è rimasto solo teorico, perché il calcolo quantistico non aveva applicazioni pratiche. Lo scenario è cambiato con i recenti progressi nel settore e con lo sviluppo dei primi prototipi di calcolatori quantistici da parte di aziende come Google e Microsoft, che hanno riportato risultati significativi con i loro progetti: Myorana 1 e Willow. Questi progetti hanno quindi portato alla luce dei veri calcolatori quantistici funzionanti e nonostante il limitato numero di qbit stabili sia insufficiente per applicazioni pratiche su larga scala, questo segna una svolta nel settore.

Sebbene le macchine quantistiche siano ancora in fase sperimentale, diversi scienziati ritengono che la loro costruzione su larga scala sia ormai una sfida principalmente ingegneristica, e alcuni prevedono la loro maturazione entro i prossimi vent'anni.

Considerando che l'attuale infrastruttura crittografica ha richiesto quasi due decenni per essere implementata, risulta necessario iniziare da subito la transizione verso sistemi progettati per resistere al calcolo quantistico.

1.3 Obiettivi della crittografia post-quantistica

L'obiettivo della crittografia post-quantistica è quindi sviluppare algoritmi crittografici sicuri sia contro i computer quantistici che classici, garantendo così un futuro sicuro anche in un'era dominata dai computer quantistici.

La crittografia post-quantistica non si limita a sostituire gli algoritmi vulnerabili, ma mira a costruire un'infrastruttura di sicurezza robusta e duratura, capace di adattarsi alle sfide tecnologiche future mantenendo la compatibilità con i protocolli e le reti esistenti.

2 Fondamenti della crittografia classica

Per comprendere le sfide poste dai computer quantistici alla crittografia moderna, è fondamentale conoscerne i principi di base dietro i principali algoritmi crittografici attualmente in uso. Una volta capiti tali principi, procederemo ad analizzare Shor, il suo impatto sugli algoritmi ed infine le soluzioni proposte dalla crittografia post-quantistica.

2.1 Crittografia a chiave pubblica: RSA, ECC

La crittografia a chiave pubblica, introdotta nel 1976 da Diffie, Hellman e Merkle, costituisce una svolta fondamentale nel panorama della sicurezza informatica moderna. A differenza dei sistemi simmetrici, che vincolano mittente e destinatario alla condivisione di un unico segreto, i protocolli asimmetrici impiegano una coppia di chiavi: una pubblica k_{pub}, liberamente distribuibile, e una privata k_{prv}, mantenuta segreta dal proprietario. La sicurezza di questo sistema si basa sulla difficoltà di risalire alla chiave privata a partire da quella pubblica. Le funzioni di cifratura C e decifratura D sono note a tutti, e per ogni messaggio m deve valere:

$$D(C(m; k_{\text{pub}}); k_{\text{prv}}) = m.$$

Il funzionamento di questo sistema si basa sulle funzioni one-way trapdoor: operazioni matematiche semplici da eseguire in una direzione, ma computazionalmente intrattabili da invertire senza la conoscenza di una informazione specifica (la "trappola").

La teoria dei numeri e l'algebra modulare forniscono il substrato matematico necessario per generare tali funzioni; a seconda del problema matematico sottostante, si distinguono i vari algoritmi di crittografia asimmetrica oggi in uso.

Richiami di algebra modulare L'aritmetica modulare è un sistema in cui i numeri si riavvolgono entro un intervallo fissato da un modulo n. Quando un valore supera (o scende sotto) questo intervallo, viene riportato all'interno prendendo il resto della divisione per n.

Un esempio quotidiano è l'orologio: in un sistema a 12 ore il modulo è 12. Se sono le 10 e aggiungo 4 ore, il risultato non è 14, ma 2, perché:

$$14 \equiv 2 \pmod{12}.$$

Per calcolare $c = a \bmod b$ si considera il resto della divisione intera tra a e b, ottenendo un valore sempre compreso tra 0 e $b-1$, esempio: $6 \bmod 4 = 2$.

Problemi difficili

- **Fattorizzazione:** dati p, q è facile calcolare $n = pq$; dato n è difficile trovare p e q.
- **Radice modulare:** dato $y = x^z \bmod s$ invertire la potenza è difficile senza conoscere $\varphi(s)$.
- **Logaritmo discreto:** data $y = x^z \bmod s$ trovare z è computazionalmente difficile.

2.1.1 RSA

Il cifrario RSA, proposto da Rivest, Shamir e Adleman nel 1978, è il sistema crittografico a chiave pubblica più diffuso e studiato. La sua sicurezza si fonda sulla difficoltà computazionale della fattorizzazione di numeri interi molto grandi.

Generazione delle chiavi Ogni utente genera la propria coppia di chiavi attraverso i seguenti passaggi:

1. Scelta di due numeri primi p e q molto grandi
2. Calcolo di $n = pq$ e della funzione di Eulero $\phi(n) = (p-1)(q-1)$
3. Scelta di un intero e minore di $\phi(n)$ e coprimo con esso
4. Calcolo dell'intero d, inverso moltiplicativo di e modulo $\phi(n)$

La chiave pubblica è la coppia (e, n), mentre la chiave privata è d. La cifratura di un messaggio m avviene calcolando $c = m^e \bmod n$, mentre la decifratura richiede il calcolo di $m = c^d \bmod n$.

La correttezza dell'algoritmo è garantita dal teorema di Eulero: poiché $ed \equiv 1 \pmod{\phi(n)}$, si ha $ed = 1 + k\phi(n)$ per qualche intero k, e quindi:

$$m^{ed} \bmod n = m^{1+k\phi(n)} \bmod n = m \cdot (m^{\phi(n)})^k \bmod n = m \bmod n$$

Sicurezza e dimensioni delle chiavi La sicurezza di RSA dipende da l'impossibilità pratica di fattorizzare n quando questo è sufficientemente grande. Conoscendo la fattorizzazione $n = pq$, un attaccante potrebbe infatti calcolare $\phi(n)$ e di conseguenza la chiave privata d.

Attualmente, le dimensioni delle chiavi considerate sicure sono di almeno 2048 bit, con raccomandazioni crescenti verso 4096 bit per applicazioni che richiedono sicurezza a lungo termine. Chiavi di 1024 bit sono considerate obsolete e vulnerabili ad attacchi con risorse computazionali moderne.

2.1.2 Crittografia su Curve Ellittiche (ECC)

La crittografia su curve ellittiche, sviluppata indipendentemente da Neal Koblitz e Victor Miller nel 1985, offre un'alternativa matematicamente elegante e computazionalmente efficiente a RSA.

Fondamenti matematici Una curva ellittica su un campo finito $\mathbb{Z}_p$ (con p primo e $p > 3$) è definita dall'equazione di Weierstrass in forma normale:

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

dove $a, b \in \mathbb{Z}_p$ soddisfano la condizione $4a^3 + 27b^2 \bmod p \neq 0$, che garantisce l'assenza di punti singolari sulla curva.

L'insieme dei punti (x, y) che soddisfano questa equazione, insieme al punto all'infinito $\mathcal{O}$, forma un gruppo abeliano additivo. È possibile definire un'operazione di addizione tra punti della curva tale che, dati due punti P e Q, la loro somma $P + Q$ sia ancora un punto della curva.

Per punti distinti $P = (x_P, y_P)$ e $Q = (x_Q, y_Q)$, con $P \neq -Q$, si ha:

$$\lambda = \frac{y_Q - y_P}{x_Q - x_P}, \quad x_S = \lambda^2 - x_P - x_Q, \quad y_S = -y_P + \lambda(x_P - x_S)$$

dove $S = P + Q = (x_S, y_S)$. Nel caso di raddoppio di un punto ($P = Q$), il coefficiente angolare diventa $\lambda = \frac{3x_P^2 + a}{2y_P}$.

Il problema del logaritmo discreto La sicurezza di ECC si basa sulla difficoltà del problema del logaritmo discreto su curve ellittiche (ECDLP): dati due punti P e Q sulla curva, trovare l'intero k tale che $Q = kP$, dove kP denota l'addizione di P con se stesso k volte.

La moltiplicazione scalare $Q = kP$ è computazionalmente efficiente (tempo polinomiale), mentre il problema inverso è considerato intrattabile: tutti gli algoritmi classici noti hanno complessità esponenziale nella dimensione della chiave.

Vantaggi rispetto a RSA ECC offre lo stesso livello di sicurezza di RSA con chiavi significativamente più corte. Una chiave ECC di 256 bit fornisce una sicurezza paragonabile a una chiave RSA di 3072 bit. Questo si traduce in:

- Minore occupazione di memoria e larghezza di banda
- Operazioni crittografiche più veloci
- Minore consumo energetico, cruciale per dispositivi mobili e IoT

2.1.3 Sicurezza classica

Entrambi gli algoritmi sono considerati sicuri nell'ambito del calcolo classico per dimensioni di chiave appropriate. La loro robustezza deriva dalla complessità computazionale dei problemi matematici sottostanti:

- **Fattorizzazione per RSA**: il miglior algoritmo classico noto è il General Number Field Sieve (GNFS), con complessità sub-esponenziale $O(e^{(64/9)^{1/3}(\ln n)^{1/3}(\ln \ln n)^{2/3}})$
- **ECDLP per ECC**: gli algoritmi più efficienti, come il metodo rho di Pollard, hanno complessità completamente esponenziale $O(\sqrt{n})$, dove n è l'ordine del gruppo

Questa differenza nella complessità degli attacchi spiega perché ECC richiede chiavi più corte per garantire lo stesso livello di sicurezza.

RSA e ECC costituiscono oggi la base dell'infrastruttura di sicurezza digitale globale, utilizzati in TLS/SSL per la sicurezza web, in SSH per l'accesso remoto sicuro, nella firma digitale di documenti e software, e in numerose altre applicazioni critiche.

2.2 Limiti rispetto ai computer quantistici

Avendo introdotto i fondamenti della crittografia classica possiamo provare ora ad analizzare le vulnerabilità di questi algoritmi in particolar modo rispetto al calcolo quantistico. Infatti nei prossimi capitoli capiremo quali sono i vantaggi che i computer quantistici offrono rispetto a quelli classici e come questi possono compromettere la sicurezza degli algoritmi classici. in particolare esamineremo l'algoritmo di Shor e il suo impatto su RSA.

3 Computazione quantistica e impatto sulla crittografia classica

Entriamo ora nel vivo della relazione, sviscerare i segreti dietro l'algoritmo di Shor per capire in che modo rompere il protocllo RSA appena descritto. Per farlo avremo prima bisogno di introdurre alcuni concetti di base della computazione quantistica che shor utilizza per ottenere i suoi risultati. Una volta compresi questi concetti potremo procedere alla spiegazione dell'algoritmo vero e proprio.

3.1 Nozioni base di computazione quantistica

La computazione quantistica rappresenta un paradigma di calcolo che sfrutta i principi della meccanica quantistica per elaborare l'informazione in modi che non sono possibili con i computer classici. Di questi principi di meccanica quantistica non esiste un corrispettivo diretto nel modelli di calcolo classico, e proprio per questo motivo che ne analizziamo gli effetti e le conseguenze.

3.1.1 Richiami essenziali di meccanica quantistica

In questa sezione non si fornisce una trattazione formale della teoria, ma si introducono i concetti essenziali necessari a comprendere il funzionamento dei computer quantistici e dei meccanismi che consentono loro di superare i limiti della computazione classica.

Il primo concetto fondamentale è quello di ***stato quantistico***. Mentre un bit classico può assumere esclusivamente i valori 0 o 1, un sistema quantistico può trovarsi in una *sovrapposizione* di stati. Questo non significa che sia 0 e 1 "contemporaneamente" in senso magico ma semplicemente che lo stato di un qubit può essere descritto come una combinazione lineare degli stati base $|0\rangle$ e $|1\rangle$:

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

con coefficienti complessi che ne determinano le probabilità di osservazione.

Un secondo concetto chiave è l'***entanglement***, una forma di correlazione quantistica tra più qubit per cui lo stato complessivo del sistema non può essere descritto come il prodotto degli stati dei singoli qubit. In presenza di entanglement, la misura di un qubit influisce istantaneamente sullo stato degli altri, indipendentemente dalla distanza che li separa. Questa proprietà consente di rappresentare e manipolare informazioni in modo non separabile, ciò significa che le operazioni su un qubit possono avere effetti immediati sugli altri qubit entangled con esso.

Il terzo elemento fondamentale è la ***misura***. Quando uno stato quantistico viene misurato, la sovrapposizione collassa e il risultato ottenuto è un valore classico. L'esito della misura è probabilistico e dipende dallo stato del sistema immediatamente prima dell'osservazione.

Sovrapposizione, entanglement e misura costituiscono i principi alla base della differenza concettuale tra computazione classica e computazione quantistica, e determinano il modo in cui l'informazione viene elaborata in un sistema quantistico.

Qubit e operazioni quantistiche

L'unità fondamentale di informazione in un computer quantistico è quindi il qubit. Dal punto di vista matematico, un qubit è rappresentato come un vettore unitario in uno spazio di Hilbert bidimensionale, mentre un registro composto da n qubit è descritto in uno spazio di dimensione 2^n.

Le operazioni sui qubit sono realizzate mediante *porte quantistiche*, che corrispondono a trasformazioni lineari e reversibili, descritte da operatori unitari. A differenza delle porte logiche classiche, le porte quantistiche agiscono su stati in sovrapposizione, modificando simultaneamente tutte le componenti dello stato quantistico.

Matematicamente, queste porte sono rappresentate come matrici unitarie, il che conferisce loro due proprietà cruciali: la somma delle probabilità dei possibili stati deve essere sempre pari a 1 e ogni operazione deve essere reversibile (ovvero, ogni porta ha una sua inversa che può annullare l'operazione). Le principali porte quantistiche:

- porte a bit singolo come: la porta Hadamard (H), che viene usata per mettere i qubit in superposizioni, la porta Pauli-X (equivalente a una NOT classica), e le porte di Pauli(X, Y, Z) che ruotano il vettore.
- porte a più bit che realizzano operazioni condizionate tra due qubit, fondamentali per creare entanglement.
- porta SWAP che scambia lo stato di due qubit, la porta di Fase Controllata (Rk) utile per la QFT
- porte specializzate costruite per specifici algoritmi a partire da queste.

L'uso combinato di queste porte consente di elaborare, in un singolo passo computazionale, un insieme di stati che cresce esponenzialmente con il numero di qubit. Tale fenomeno è noto come *parallelismo quantistico* e rappresenta una delle principali fonti di vantaggio rispetto al calcolo classico, pur non traducendosi automaticamente in un'accelerazione per qualsiasi problema.

Modello di calcolo quantistico

Il funzionamento di un computer quantistico segue un modello di calcolo specifico, distinto da quello deterministico utilizzato nei computer classici. Un algoritmo quantistico è generalmente articolato in tre fasi principali:

- preparazione dello stato iniziale;
- applicazione di una sequenza di porte quantistiche;
- misura finale del sistema.

Durante la fase di evoluzione, lo stato quantistico del sistema viene trasformato in modo deterministico secondo le leggi della meccanica quantistica. La natura probabilistica emerge esclusivamente al momento della misura, quando lo stato viene proiettato su uno dei possibili risultati osservabili.

La potenza del modello di calcolo quantistico non risiede nella possibilità di valutare esplicitamente tutte le soluzioni di un problema, bensì nella capacità di sfruttare il fenomeno dell'*interferenza quantistica*. Attraverso opportune sequenze di operazioni, le ampiezze associate alle soluzioni corrette possono essere amplificate, mentre quelle delle soluzioni errate vengono attenuate.

3.1.2 Collegamento con la crittografia

I concetti introdotti in questa sezione costituiscono il fondamento teorico per analizzare l'impatto della computazione quantistica sulla crittografia moderna. In particolare, l'uso combinato di sovrapposizione, entanglement e interferenza consente di affrontare in modo efficiente problemi come la fattorizzazione di interi e il calcolo del logaritmo discreto.

3.2 Algoritmi quantistici rilevanti: Shor

L'algoritmo di Shor rappresenta uno dei risultati più significativi nel campo della computazione quantistica, non solo per le sue implicazioni sulla crittografia, ma anche per la profondità delle idee matematiche che lo compongono. Infatti per comprendere appieno il funzionamento di questo algoritmo, è necessario procedere con un'analisi rigorosa che parta dai fondamenti matematici, prescindendo inizialmente dagli aspetti quantistici.

L'obiettivo quindi di questa sezione è chiarire perché l'algoritmo funziona dal punto di vista matematico, e solo in una fase successiva dell'analisi vedremo come viene accelerato dal computer quantistico.

Il problema della fattorizzazione

Il punto di partenza è il problema della fattorizzazione di interi. Dato un intero composto $N = p \cdot q$, dove p e q sono numeri primi di grandi dimensioni, il problema consiste nel determinare p e q conoscendo solamente N. La difficoltà computazionale della fattorizzazione deriva da due caratteristiche fondamentali: in primo luogo, non è noto alcun algoritmo classico in grado di fattorizzare un numero in tempo polinomiale rispetto alla dimensione dell'input; in secondo luogo, i tentativi diretti di fattorizzazione crescono rapidamente con l'aumentare della dimensione di N, rendendo il problema intrattabile per valori sufficientemente grandi.

L'intuizione fondamentale di Shor consiste nel trasformare il problema della fattorizzazione in un problema di natura diversa, caratterizzato da una struttura matematica più favorevole. Questa trasformazione permette di ricondurre la ricerca dei fattori primi a un problema di teoria dei numeri che, come vedremo, può essere risolto in modo efficiente sfruttando le peculiarità della computazione quantistica.

3.2.1 Dalla fattorizzazione alla teoria dei numeri

La strategia di Shor si basa sull'osservazione che la fattorizzazione può essere ricondotta al problema della determinazione del periodo di una funzione. Questa connessione non è immediata e richiede una serie di passaggi matematici che è necessario esplicitare con precisione.

Il primo passo consiste nella scelta di un numero intero a tale che $1 < a < N$ e $\gcd(a, N) = 1$. La condizione sulla coprimalità è essenziale: se infatti $\gcd(a, N) \neq 1$, avremmo già trovato un fattore non banale di N semplicemente calcolando il massimo comun divisore. Una volta scelto a,

si considera la funzione esponenziale modulare definita come:

$$f(x) = a^x \bmod N$$

Questa funzione possiede una proprietà fondamentale: è periodica. Più precisamente, esiste un minimo intero $r > 0$ tale che:

$$a^r \equiv 1 \pmod{N}$$

Questo valore r è chiamato ordine di a modulo N. L'esistenza di tale periodo è garantita dal teorema di Eulero, secondo cui $a^{\phi(N)} \equiv 1 \pmod{N}$, dove $\phi(N)$ è la funzione di Eulero. Il periodo cercato è dunque un divisore di $\phi(N)$.

3.2.2 L'utilità del periodo per la fattorizzazione

A questo punto sorge naturale una domanda: perché la conoscenza del periodo r dovrebbe aiutarci a fattorizzare N? La risposta risiede in una proprietà algebrica profonda che collega l'ordine di un elemento alla struttura moltiplicativa del gruppo $(\mathbb{Z}/N\mathbb{Z})^*$.

Se il periodo r soddisfa due condizioni specifiche, ovvero se r è pari e se $a^{r/2} \not\equiv -1 \pmod{N}$, allora è possibile estrarre fattori non banali di N calcolando:

$$\gcd(a^{r/2} - 1, N) \quad \text{e} \quad \gcd(a^{r/2} + 1, N)$$

Questa è la chiave matematica dell'intero algoritmo. Per comprendere perché questa procedura funziona, è necessario esaminare più da vicino la struttura algebrica sottostante. Dalla relazione $a^r \equiv 1 \pmod{N}$ segue immediatamente che:

$$a^r - 1 \equiv 0 \pmod{N}$$

Poiché r è pari per ipotesi, possiamo fattorizzare il termine $a^r - 1$ utilizzando la differenza di quadrati:

$$a^r - 1 = (a^{r/2})^2 - 1 = (a^{r/2} - 1)(a^{r/2} + 1)$$

Ne consegue che:

$$(a^{r/2} - 1)(a^{r/2} + 1) \equiv 0 \pmod{N}$$

Questa congruenza ci dice che N divide il prodotto $(a^{r/2} - 1)(a^{r/2} + 1)$. Tuttavia, per ipotesi, $a^{r/2} \not\equiv -1 \pmod{N}$, il che significa che $a^{r/2} + 1$ non è divisibile per N. Analogamente, poiché r è il periodo minimo, anche $a^{r/2} - 1$ non può essere divisibile per N (altrimenti $r/2$ sarebbe il periodo).

Di conseguenza, N divide il prodotto di due fattori senza dividere ciascun fattore singolarmente. Questo implica necessariamente che ciascuno dei due fattori condivide almeno un divisore primo con N, ma non tutti. Il calcolo del massimo comun divisore permette quindi di estrarre questi divisori non banali.

3.2.3 Esempio numerico completo

Per rendere concreti i concetti esposti, consideriamo un esempio numerico completo. Supponiamo di voler fattorizzare $N = 15$. Scegliamo $a = 2$ e verifichiamo che $\gcd(2, 15) = 1$, come richiesto. Procediamo quindi al calcolo della sequenza di potenze di a modulo N:

x	$2^x \bmod 15$
1	2
2	4
3	8
4	1

Osserviamo che il valore si ripete dopo quattro iterazioni, pertanto il periodo è $r = 4$. Verifichiamo ora le condizioni necessarie: il periodo è effettivamente pari, e calcoliamo $a^{r/2} = 2^2 = 4$. Poiché $4 \not\equiv -1 \pmod{15}$ (infatti $4 \equiv 4 \pmod{15}$), entrambe le condizioni sono soddisfatte. Possiamo quindi applicare la formula per estrarre i fattori:

$$\gcd(4 - 1, 15) = \gcd(3, 15) = 3$$
$$\gcd(4 + 1, 15) = \gcd(5, 15) = 5$$

Abbiamo così ottenuto la fattorizzazione $15 = 3 \cdot 5$.

La complessità computazionale del metodo

A questo punto è importante sottolineare quali passaggi del metodo siano effettivamente computazionalmente trattabili e quale rappresenti invece il collo di bottiglia. I passaggi che non presentano difficoltà computazionali significative includono il calcolo del massimo comun divisore, che può essere eseguito efficientemente tramite l'algoritmo di Euclide, e l'esecuzione di esponenziazioni modulari, che può essere realizzata in tempo polinomiale mediante l'algoritmo di esponenziazione veloce.

Il passaggio critico dell'intero procedimento è la determinazione del periodo r. Nel contesto della computazione classica, il calcolo del periodo della

funzione $a^x \bmod N$ richiede essenzialmente di valutare la funzione ripetutamente fino a quando non si osserva la ripetizione del valore iniziale. Il numero di valutazioni necessarie nel caso peggiore può essere dell'ordine di N, e anche nel caso medio il numero di operazioni cresce in modo tale da rendere il problema intrattabile per valori di N sufficientemente grandi. È precisamente in questo passaggio che risiede il collo di bottiglia computazionale che impedisce alla fattorizzazione classica di essere efficiente.

Casi di fallimento e strategie di gestione

È importante osservare che il metodo descritto non ha sempre successo. Possono verificarsi due situazioni in cui la procedura non produce una fattorizzazione: il periodo r potrebbe risultare dispari, oppure potrebbe accadere che $a^{r/2} \equiv -1 \pmod{N}$. In entrambi questi casi, non è possibile applicare la fattorizzazione attraverso il massimo comun divisore nel modo descritto. Tuttavia, è possibile dimostrare che la probabilità di incorrere in uno di questi casi sfavorevoli è relativamente bassa. Inoltre, qualora si verifichi uno di questi casi, è sufficiente scegliere un valore diverso di a e ripetere l'intero procedimento. Con alta probabilità, dopo pochi tentativi si otterrà un periodo che soddisfa le condizioni necessarie.

Sintesi del contributo matematico

Ricapitolando quanto esposto, dal punto di vista strettamente matematico, l'algoritmo di Shor opera attraverso una sequenza logica di trasformazioni: in primo luogo, trasforma il problema della fattorizzazione in un problema di determinazione della periodicità di una funzione; in secondo luogo, sfrutta proprietà fondamentali dell'aritmetica modulare per collegare il periodo alla struttura dei fattori di N; in terzo luogo, riduce il problema al calcolo dell'ordine di un elemento modulo N; infine, estrae i fattori cercati attraverso il calcolo del massimo comun divisore.

È fondamentale sottolineare che fino a questo punto non è stato introdotto alcun concetto di natura quantistica. Tutto quanto discusso appartiene al dominio della matematica classica e della teoria dei numeri. Il contributo essenziale della computazione quantistica risiede nella capacità di rendere efficiente il passaggio centrale, ovvero la determinazione del periodo. Mentre su un computer classico questo passaggio è computazionalmente intrattabile per numeri di grandi dimensioni, vedremo che un computer quantistico è in grado di eseguirlo in tempo polinomiale, rendendo così l'intero algoritmo efficiente e ponendo una seria minaccia alla sicurezza dei sistemi crittografici basati sulla difficoltà della fattorizzazione.

3.2.4 La trasformata di Fourier quantistica

Per comprendere come un computer quantistico risolva efficientemente il problema della determinazione del periodo, è necessario introdurre uno strumento matematico fondamentale: la trasformata di Fourier. L'obiettivo di questa parte dell'analisi è chiarire perché la trasformata di Fourier rappresenti lo strumento naturale per affrontare problemi di periodicità e come la sua versione quantistica costituisca l'elemento centrale dell'algoritmo di Shor. Procederemo costruendo il ragionamento a partire da concetti elementari, senza assumere familiarità con gli aspetti tecnici della trasformata.

Il ruolo della trasformata di Fourier nell'analisi della periodicità Dall'analisi matematica precedente è emerso che il problema centrale consiste nel determinare il periodo di una funzione della forma $f(x) = a^x \bmod N$. È importante chiarire che il calcolo dei singoli valori di $f(x)$ non presenta difficoltà computazionali: dato un valore specifico di x, è possibile calcolare $f(x)$ in modo efficiente. La difficoltà risiede nella determinazione della frequenza con cui i valori della funzione si ripetono.

La trasformata di Fourier è uno strumento matematico che permette di operare un cambio di prospettiva fondamentale. Concettualmente, essa consente di passare da una descrizione di un segnale o di una funzione nel cosiddetto "dominio del tempo" a una descrizione nel "dominio delle frequenze". Il periodo di una funzione è intrinsecamente una proprietà legata alle frequenze: una funzione periodica con periodo r può essere interpretata come un segnale che oscilla con frequenza fondamentale $1/r$. Questa osservazione costituisce la chiave per comprendere perché la trasformata di Fourier sia lo strumento appropriato per estrarre informazioni sulla periodicità.

Intuizione alla base della trasformata di Fourier Prima di procedere con la formulazione matematica precisa, è utile sviluppare un'intuizione del funzionamento della trasformata di Fourier attraverso un esempio semplice. Si consideri un segnale periodico, che potrebbe rappresentare un'onda sonora, una vibrazione meccanica o qualsiasi altra grandezza fisica che varia nel tempo secondo un pattern ripetitivo. Nel dominio temporale, osserviamo come il segnale evolve istante per istante, registrando il valore della grandezza in funzione del tempo.

La trasformata di Fourier effettua un'operazione concettualmente diversa: prende il segnale nel dominio temporale e lo scompone in una somma di oscillazioni elementari, ciascuna caratterizzata da una frequenza ben definita. In altre parole, invece di descrivere quando il segnale assume determinati valori, la trasformata descrive quali frequenze compongono il segnale e con

quale intensità ciascuna frequenza contribuisce. Il periodo del segnale originale emerge naturalmente da questa rappresentazione in termini di frequenze.

Per rendere questa idea più concreta, si consideri un segnale definito come la somma di due sinusoidi:

$$s(t) = \sin(2\pi t) + \sin(4\pi t)$$

Nel dominio temporale, questo segnale appare come un'onda dalla forma piuttosto irregolare, risultante dalla sovrapposizione delle due componenti. Tuttavia, nel dominio delle frequenze, la trasformata di Fourier rivela immediatamente la struttura sottostante: il segnale è composto da esattamente due componenti, una con frequenza 1 e una con frequenza 2. La trasformata di Fourier serve precisamente a estrarre questa decomposizione in componenti frequenziali, permettendo di identificare le frequenze fondamentali che caratterizzano il segnale.

La trasformata di Fourier discreta Nel contesto dell'algoritmo di Shor, non abbiamo a che fare con segnali continui nel tempo, ma piuttosto con sequenze discrete di valori. La funzione $f(x) = a^x \bmod N$ produce una sequenza di valori interi $f(0), f(1), f(2), \ldots$ che si ripete con periodo r. Per analizzare questo tipo di segnali discreti è necessario utilizzare la trasformata di Fourier discreta, comunemente indicata con l'acronimo DFT (Discrete Fourier Transform).

La trasformata di Fourier discreta opera su una sequenza finita di valori $x_0, x_1, \ldots, x_{N-1}$ e la trasforma in un'altra sequenza $X_0, X_1, \ldots, X_{N-1}$, dove ciascun coefficiente X_k misura l'intensità con cui la frequenza k è presente nel segnale originale. La definizione formale della trasformata di Fourier discreta è data da:

$$X_k = \sum_{n=0}^{N-1} x_n \, e^{-2\pi i k n / N}$$

Sebbene non sia necessario memorizzare questa formula nei dettagli, è importante comprenderne il significato concettuale. La sommatoria confronta il segnale originale con tutte le possibili oscillazioni discrete di frequenza k, e il valore di X_k misura quanto bene l'oscillazione di frequenza k si "accorda" con il segnale. Le oscillazioni che sono in fase con il segnale contribuiscono costruttivamente, mentre quelle fuori fase tendono a cancellarsi.

Periodicità e concentrazione spettrale Il collegamento diretto tra periodicità e trasformata di Fourier emerge quando si considera il caso di una funzione perfettamente periodica. Se una funzione possiede periodo r, la sua

trasformata di Fourier non è distribuita uniformemente su tutte le frequenze, ma risulta concentrata su valori specifici. In particolare, l'energia spettrale è localizzata in corrispondenza di frequenze che sono multipli interi di $1/r$.

Questa proprietà implica che, osservando il risultato della trasformata di Fourier, è possibile ricavare informazioni precise sul periodo r. I picchi nella trasformata identificano le frequenze caratteristiche, e da queste frequenze si può risalire al periodo. Tuttavia, nel contesto della computazione classica, il calcolo della trasformata di Fourier discreta richiede un numero di operazioni che, anche utilizzando algoritmi efficienti come la FFT (Fast Fourier Transform), cresce almeno proporzionalmente al numero di campioni. Per sequenze di lunghezza comparabile a N, questo rappresenta comunque un costo computazionale significativo che non risolve il problema della complessità della fattorizzazione.

Dalla DFT alla trasformata di Fourier quantistica La trasformata di Fourier quantistica, comunemente indicata con l'acronimo QFT (Quantum Fourier Transform), rappresenta l'adattamento della trasformata di Fourier discreta al contesto della computazione quantistica. È fondamentale comprendere che la QFT non introduce nuove idee matematiche: si tratta essenzialmente della stessa trasformazione matematica definita dalla DFT, ma applicata a stati quantistici invece che a sequenze di numeri classici.

In un computer quantistico, una sequenza di valori viene rappresentata attraverso una sovrapposizione quantistica, ovvero uno stato della forma:

$$\sum_x \alpha_x \left|x\right\rangle$$

dove i coefficienti complessi α_x rappresentano le ampiezze di probabilità associate a ciascun valore di base $|x\rangle$. La QFT trasforma questo stato in un nuovo stato:

$$\sum_k \beta_k \left|k\right\rangle$$

dove i coefficienti β_k sono esattamente la trasformata di Fourier dei coefficienti originali α_x. In altre parole, la QFT realizza fisicamente, a livello di stato quantistico, la stessa operazione matematica che la DFT realizza a livello di array di numeri.

Il vantaggio computazionale della QFT La ragione per cui la QFT rappresenta un avanzamento rivoluzionario rispetto alla DFT classica risiede in una proprietà fondamentale della computazione quantistica: la capacità di operare simultaneamente su una sovrapposizione di stati. Nel computer

quantistico, tutti i coefficienti α_x esistono contemporaneamente nella sovrapposizione, e la QFT agisce sull'intera sovrapposizione in un'unica operazione coerente.

Il risultato di questa applicazione coerente della trasformata è che le ampiezze dei diversi stati interferiscono tra loro secondo le leggi della meccanica quantistica. Le frequenze che sono compatibili con la periodicità della funzione originale vengono amplificate attraverso interferenza costruttiva, mentre le frequenze incompatibili vengono soppresse attraverso interferenza distruttiva. Quando si effettua una misura sullo stato risultante, si ottiene con alta probabilità un valore che contiene informazione sul periodo cercato.

È importante sottolineare che la misura non fornisce direttamente il periodo r, ma piuttosto un valore dal quale è possibile estrarre r utilizzando tecniche matematiche classiche. In particolare, si utilizza l'algoritmo delle frazioni continue per approssimare il rapporto misurato con una frazione che ha il periodo come denominatore. Questa fase di post-elaborazione classica è essenziale per completare l'algoritmo.

Esempio concettuale del meccanismo Per rendere più concreto il meccanismo di estrazione del periodo, consideriamo un esempio semplificato. Supponiamo che la funzione analizzata abbia periodo $r = 4$ e che il registro quantistico abbia dimensione Q. Dopo l'applicazione della QFT, lo stato quantistico non è distribuito uniformemente, ma presenta concentrazioni di ampiezza in corrispondenza di valori approssimabili come:

$$k \approx \frac{m}{4} \cdot Q$$

dove m è un intero. Quando si effettua la misura, si ottiene con alta probabilità uno di questi valori di k. Dal rapporto:

$$\frac{k}{Q} \approx \frac{m}{r}$$

è possibile ricostruire il periodo r attraverso l'approssimazione con frazioni continue. Questo esempio illustra il collegamento diretto tra l'applicazione della QFT, l'estrazione di informazione sulla periodicità attraverso la misura quantistica, e il recupero finale del periodo che permette la fattorizzazione.

La natura del vantaggio quantistico È essenziale chiarire un aspetto concettuale che spesso genera confusione. Il computer quantistico non ottiene il suo vantaggio semplicemente "provando tutte le frequenze in parallelo" né "calcolando la trasformata esplicitamente come farebbe una CPU classica".

Il meccanismo è qualitativamente diverso: il computer quantistico prepara uno stato quantistico con una struttura specifica, sfrutta le proprietà di interferenza e sovrapposizione per far emergere le frequenze rilevanti, e produce il periodo come risultato di un processo probabilistico governato dalle leggi della meccanica quantistica.

La QFT è il meccanismo matematico che rende possibile questa interferenza controllata. Essa permette di organizzare le ampiezze quantistiche in modo tale che, al momento della misura, i valori che contengono informazione sul periodo abbiano probabilità di osservazione significativamente maggiore rispetto agli altri. Questa capacità di manipolare coerentemente le ampiezze di probabilità attraverso interfer

4 Crittografia post quantistica

La crittografia post-quantistica[1], anche nota come crittografia quantum-safe o quantum-resistant, è quell'ambito della ricerca e dello sviluppo di algoritmi crittografici (solitamente a chiave pubblica) progettati per essere sicuri contro attacchi crittanalitici effettuati da computer quantistici. Ecco i pilastri fondamentali che compongono questa definizione:

1. Indipendenza dall'hardware quantistico. A differenza della "crittografia quantistica"[2] (che sfrutta i principi della fisica quantistica per proteggere le comunicazioni, come la distribuzione quantistica delle chiavi), la crittografia post-quantistica si basa su problemi matematici eseguiti su computer classici, ma strutturati in modo da essere resistenti anche ai futuri computer quantistici.

2. Superamento della vulnerabilità agli algoritmi quantistici

   La crittografia post-quantistica deve essere immune alle due minacce principali rappresentate dai computer quantistici:

   - Algoritmo di Shor: di cui abbiamo ampiamente discusso nel capitolo precedente.
   - Algoritmo di Grover: in grado di fornire un'accelerazione quadratica per le ricerche brute-force non strutturate. Per contrastare Grover, la PQC simmetrica richiede semplicemente il raddoppio della lunghezza delle chiavi (ad esempio, passare da AES-128 a AES-256) per mantenere lo stesso livello di sicurezza.

3. Fondazione su nuovi problemi matematici: La PQC si basa su problemi matematici che, allo stato attuale della ricerca, non presentano vulnerabilità esponenziali quantistiche. Nel prossimo capitolo: "Algoritmi post-quantistici" esamineremo nel dettaglio le principali famiglie di problemi matematici su cui si basano gli algoritmi post-quantistici.

4. "Forward Secrecy" Una definizione completa di PQC include la necessità di proteggere i dati non solo in futuro, ma anche oggi.

   Il modello di minaccia "Harvest Now, Decrypt Later" (raccogli ora, decifra dopo) suggerisce che attori malintenzionati possano archiviare dati criptati oggi per decifrarli quando saranno disponibili computer quantistici sufficientemente potenti.

[1] Wikipedia: Post-quantum cryptography
[2] Quantum Computing and Cryptography

Per questo motivo è considerata una priorità di sicurezza nazionale e infrastrutturale immediatarisulta e non così lontana. Queste motivazioni infatti stanno guidando la comunità scentifica a standardizzare quanto prima possibile questi algoritmi in modo da iniziare una migrazione verso questi protocolli già oggi.

4.1 Requisiti e obiettivi

Requisiti pratici e tempistiche Sebbene la minaccia sia teoricamente dimostrata, la realizzazione pratica di computer quantistici capaci di violare RSA ed ECC richiede risorse considerevoli. Secondo stime del NIST, per compromettere una chiave RSA-2048 sarebbero necessari diversi milioni di qubit logici affidabili, mentre le implementazioni attuali (2024) operano con centinaia di qubit fisici caratterizzati da elevati tassi di errore.

La transizione da qubit fisici a qubit logici richiede tecniche di correzione degli errori quantistici che impongono un overhead significativo: potrebbero essere necessari da centinaia a migliaia di qubit fisici per realizzare un singolo qubit logico stabile.

4.2 Ruolo del Nist e processo di standardizzazione

La risposta: crittografia post-quantistica Di fronte a questa minaccia emergente, il National Institute of Standards and Technology (NIST) ha avviato nel 2016 un processo di standardizzazione per identificare algoritmi crittografici resistenti agli attacchi quantistici. Nel luglio 2022, il NIST ha annunciato i primi algoritmi selezionati per la standardizzazione, basati su problemi matematici ritenuti difficili anche per computer quantistici, come i reticoli algebrici e i codici correttori di errori.

La migrazione verso la crittografia post-quantistica rappresenta una delle sfide più urgenti per la sicurezza informatica moderna, richiedendo un'attenta pianificazione per sostituire l'infrastruttura crittografica esistente mantenendo retrocompatibilità e garantendo una transizione graduale e sicura.

5 Algoritmi post-quantistici

A questo punto della relazione, abbiamo compreso come funziona la crittografia classica e come i computer quantisci sono in grado di violarla, ora possiamo quindi entrare nel secondo punto fondamentale di questo elaborato : la crittografia post-quantistica (PQC). Prima però servirà un ultimo preambolo, visto che abbiamo compreso che l'interra crittografia si basa su problemi matematici, analizziamo in breve quali sono le classi di complessità computazionali dei problemi matematici che andremo poi ad analizzare per capirne la sicurezza.

Introduzione

Classi di complessità e problemi matematici Per comprendere perché la crittografia post-quantistica rappresenta una soluzione efficace, introduciamo la gerarchia delle classi di complessità computazionale per capire dove si collocano i diversi problemi che ci saranno utili in seguito.

Il problema P vs NP Uno dei problemmi ancora irrisolti nella matematica e informatica teorica è appunto il problema P vs NP, difatti è inserito tra i sette problemi del millennio dal [3]. La questione riguarda la differenza tra il "risolvere" un problema e il "verificare" una soluzione giò data.

- **P (Polynomial time)**: Problemi che possono essere risolti in tempo polinomiale da un algoritmo deterministico. Questi problemi sono considerati "facili" da risolvere. Esempio: ordinamento di una lista.
- **NP (Nondeterministic Polynomial time)**: Problemi per i quali, data una possibile soluzione, è possibile verificare la correttezza in tempo polinomiale. Esempio: il problema del cammino hamiltoniano.
- **Np-hard**: Problemi almeno difficili quanto i problemi più difficili in NP. Formalmente, un problema NP-hard è tale se può essere ridotto in tempo polinomiale a un qualisiasi problema in NP, percui risolvere un problema NP-hard significa dimmostrare che $P = NP$. Esempio: il problema del commesso viaggiatore (TSP).

Da qui emerge una domanda cruciale: per ogni problema la cui soluzione è facile da verficare (NP) è anche facile trovare una solzuione (P)? Se $P = NP$, allora ogni volta che possiamo controllare rapidamente una soluzione deve esistere anche un modo veloce per trovarla. La comunità scientifica concorda

[3] Clay Mathematic Institute

che $P \neq NP$, il che implica che esistono problemi in intrinecamente difficili per i quali trovare la soluzione richiede tempi esponenziali, anche se verificarli è immediato, e questo rappresenta il fondamento della sicurezza crittografica.

BQP Nel caso della nostra analisi, è importante introdurre in questa gerarchia anche la classe BQP (Bounded-error Quantum Polynomial time), che rappresenta l'insieme dei problemi risolvibili efficientemente da un computer quantistico. Attualmmente, sebbene non sia dimostrato, la comunità scientifica ritiene che BQP non contenga NP-hard. In altre parole, si ipotizza che nemmeno un computer quantistico possa risolvere in modo efficiente problemi come il TSP. Questo è fondamentale perché implica che esistono problemi matematici che rimangono difficili da risolvere anche rispetto al calcolo quantistico e che quindi possono essere utilizzati come base per la crittografia post-quantistica. In questa classe di problemi rientrano infatti RSA e ECC, che come abbiamo visto sono vulnerabili per via della loro struttura matematica basata sulla periodicità.

PQC La crittografia post-quantistica sposta quindi la sua sicurezza dai problemmi di classe NP-Intermediate (di cui fanno parte RSA ed ECC) a nuove famiglie matematiche che, allo stato attuale della ricerca, non presentano vulnerabilità esponenziali quantistiche, in particolare motli di questi sono legati a problemi NP-hard.

5.1 Lattice-baces

La crittografia basta sui reticoli (Lattice-based)[4] è una delle famiglie più promettenti della crittografia post-quantistica. La sua sicurezza si basa sulle proprietà geometriche dei reticoli. Un reticolo è un insieme di punti nello spazio n-dimensionale che possono essere rappresentati come combinazioni lineari di vettori base con coeficenti interi.

Il Problema matematico fondamentale La sicurezza di questa famiglia di algoritmi si basa sulla difficoltà di risolvere problemi specifici all'interno di questi reticoli. Questo è il campo da gioco, i principali problemi che ne derivaono sono:

Learning With Errors (LWE) Introdotto da Oded Regev nel 2005, per il quale ha ricevuto il premio Gödel nel 2018, LWE[5] consiste nel risalire ad

[4]NIST FIPS 203: Lattice-Based Cryptography
[5]LWE

un vettore segreto $\mathbf{s} \in \mathbb{Z}_q^n$ dato un insieme di equazioni lineari rumorose.

Formalmente:

- si sceglie una matrice pubblica $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ con elementi scelti uniformemente a caso
- ognuno sceglie un vettore segreto $\mathbf{s} \in \mathbb{Z}_q^n$ (la chiave privata)
- e un vettore di errore $\mathbf{e} \in \mathbb{Z}_q^m$ con componenti piccole

Il problema fornisce coppie $(\mathbf{A}, \mathbf{b})$ dove:

$$\mathbf{b} = \mathbf{A} \cdot \mathbf{s} + \mathbf{e} \pmod{q} \tag{1}$$

L'obiettivo, come abbiamo anticipato, è recuperare il vettore segreto conoscendo solo $\mathbf{A}$ e $\mathbf{b}$. Non conoscere l'errore $\mathbf{e}$ rende il problema computazionalmente intrattabile. La difficoltà del problema è stata dimostrato che è correlata alla risoluzione di problemi nel caso pessimo sui reticoli, come come il Shortest Vector Problem (SVP) e il Shortest Independent Vectors Problem (SIVP). Esisitono due versioni del problema:

- **Search-LWE**: Trovare il vettore segreto $\mathbf{s}$ dato un insieme di campioni
- **Decision-LWE**: Distinguere campioni LWE $(A, \mathbf{As} + \mathbf{e})$ da campioni completamente casuali $(A, \mathbf{u})$ dove $\mathbf{u}$ è uniforme

È stato dimostrato che le due versioni sono equivalenti, risolvere il problema decisionale consente di risolvere anche quello di ricerca.

Module Learning with Errors (MLWE) È una generalizzazione di LWE che opera su strutture chiamate "moduli" su anelli polinomiali, tipicamente $R_q = \mathbb{Z}_q[x]/(x^n + 1)$ dove n è una potenza di 2. Questa variante è utilizzata negli standard moderni perché permette una maggiore efficienza computazionale e chiavi di dimensioni significativamente ridotte rispetto al LWE standard.

La formulazione MLWE sostituisce i vettori con vettori di polinomi e le matricrici con matrici di polinomi, mantenendo la stessa struttura generale ma sfruttando la struttura algebrica degli anelli per migliorare le prestazioni. L'equazione diventa:

$$\mathbf{t} = \mathbf{A} \cdot \mathbf{s} + \mathbf{e} \pmod{q} \tag{2}$$

dove ora $\mathbf{A}$, $\mathbf{s}$ e $\mathbf{e}$ sono elementi del modulo su R_q.

Short Integer Solution (SIS/MSIS) Consiste nel trovare una soluzione "piccola" (con coefficienti bassi) per un sistema lineare. Data una matrice $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, trovare un vettore non nullo $\mathbf{x}$ con norma piccola tale che $\mathbf{Ax} = \mathbf{0} \pmod{q}$. Questo problema è utilizzato principalmente per schemi di firma digitale.

Resistenza agli attacchi quantistici

I problemi basati sui reticoli (come MLWE e SIS) sono ritenuti difficili da risolvere anche per un avversario dotato di un computer quantistico a tolleranza d'errore. Al momento non sono noti algoritmi quantistici in grado di rompere efficientemente questi schemi. L'algoritmo quantistico di Grover può fornire solo un'accelerazione quadratica nella ricerca, che è significativa ma non sufficiente a rendere il problema trattabile.

La resistenza quantistica deriva dalla natura "disordinata" e non strutturata dei problemi sui reticoli, in contrasto con la struttura periodica che caratterizza i problemi di fattorizzazione e logaritmo discreto.

Key-Encapsulation Mechanism (KEM)

Un Key-Encapsulation Mechanism è un insieme di algoritmi che, sotto determinate condizioni, può essere utilizzato da due parti per stabilire una chiave segreta condivisa su un canale pubblico[6]. A differenza della crittografia a chiave pubblica tradizionale come RSA, dove un messaggio può essere cifrato direttamente con la chiave pubblica, un KEM è progettato specificamente per lo scambio sicuro di chiavi simmetriche.

Il protocollo KEM basato su MLWE funziona secondo il seguente schema:

Generazione delle chiavi (KeyGen) Alice genera una coppia di chiavi:

1. Sceglie un vettore segreto $\mathbf{s}$ (piccolo, campionato da una distribuzione di errore)
2. Genera una matrice pubblica $\mathbf{A}$ (uniforme casuale)
3. Campiona un vettore di errore $\mathbf{e}$ (piccolo)
4. Calcola la chiave pubblica: $\mathbf{t} = \mathbf{As} + \mathbf{e} \pmod{q}$

La **chiave di incapsulamento** (pubblica) è la coppia $(\mathbf{A}, \mathbf{t})$, mentre la **chiave di decapsulamento** (privata) è $\mathbf{s}$.

[6]NIST FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard

Incapsulamento (Encaps) Bob, ricevuta la chiave pubblica di Alice, vuole stabilire una chiave condivisa:

1. Campiona un nuovo vettore segreto temporaneo $\mathbf{r}$ e vettori di errore $\mathbf{e}_1$, $\mathbf{e}_2$
2. Calcola il ciphertext:

$$\mathbf{u} = \mathbf{A}^T\mathbf{r} + \mathbf{e}_1 \pmod{q} \tag{3}$$
$$v = \mathbf{t}^T\mathbf{r} + \mathbf{e}_2 + \text{Encode}(K) \pmod{q} \tag{4}$$

   dove K è la chiave segreta condivisa (tipicamente 256 bit) codificata opportunamente
3. Invia il ciphertext $(\mathbf{u}, v)$ ad Alice

Decapsulamento (Decaps) Alice, ricevuto il ciphertext da Bob, recupera la chiave condivisa:

1. Calcola: $w = v - \mathbf{s}^T\mathbf{u} \pmod{q}$
2. Decodifica w per ottenere K

Correttezza del protocollo La correttezza si basa sul fatto che:

$$w = v - \mathbf{s}^T\mathbf{u} \tag{5}$$
$$= (\mathbf{t}^T\mathbf{r} + \mathbf{e}_2 + \text{Encode}(K)) - \mathbf{s}^T(\mathbf{A}^T\mathbf{r} + \mathbf{e}_1) \tag{6}$$
$$= ((\mathbf{A}\mathbf{s} + \mathbf{e})^T\mathbf{r} + \mathbf{e}_2 + \text{Encode}(K)) - \mathbf{s}^T\mathbf{A}^T\mathbf{r} - \mathbf{s}^T\mathbf{e}_1 \tag{7}$$
$$= \mathbf{e}^T\mathbf{r} + \mathbf{e}_2 - \mathbf{s}^T\mathbf{e}_1 + \text{Encode}(K) \tag{8}$$

Il termine di errore totale $\mathbf{e}^T\mathbf{r} + \mathbf{e}_2 - \mathbf{s}^T\mathbf{e}_1$ rimane sufficientemente piccolo (perché tutti i vettori coinvolti hanno componenti piccole) da permettere la decodifica corretta di K.

Sicurezza Un attaccante che intercetta $(\mathbf{A}, \mathbf{t})$ e $(\mathbf{u}, v)$ deve risolvere il problema MLWE per recuperare K, il che è computazionalmente intrattabile. La chiave segreta condivisa può poi essere utilizzata con algoritmi crittografici simmetrici (come AES) per cifrare e autenticare le comunicazioni.

Per ottimizzare le prestazioni, questi algoritmi utilizzano la Trasformata Teoretica dei Numeri (NTT), che permette di eseguire moltiplicazioni polinomiali in $O(n \log n)$ anziché $O(n^2)$, rendendo il sistema molto più veloce rispetto ai metodi tradizionali.

Firme Digitali basate su reticoli

Le firme digitali sono utilizzate per autenticare l'identità e l'integrità dei dati. Inoltre, il destinatario di dati firmati può utilizzare una firma digitale come prova per dimostrare a terzi che la firma è stata effettivamente generata dal firmatario dichiarato (proprietà di non ripudio)[7].

Gli schemi di firma basati su reticoli utilizzano tipicamente il problema SIS/MSIS come fondamento della loro sicurezza.

Stato della standardizzazione

Il NIST (National Institute of Standards and Technology) ha avviato un processo di standardizzazione della crittografia post-quantistica nel 2016[8]. Il 13 agosto 2024 sono stati pubblicati i primi standard finali basati sui reticoli[9]:

- **FIPS 203 (ML-KEM)**: Basato sull'algoritmo CRYSTALS-Kyber, è lo standard primario per lo scambio di chiavi. ML-KEM è l'acronimo di Module-Lattice-Based Key-Encapsulation Mechanism. La sicurezza di ML-KEM è correlata alla difficoltà computazionale del problema Module Learning with Errors. Attualmente si ritiene che ML-KEM sia sicuro anche contro avversari in possesso di un computer quantistico.
- **FIPS 204 (ML-DSA)**: Basato su CRYSTALS-Dilithium, è lo standard primario per le firme digitali. ML-DSA è l'acronimo di Module-Lattice-Based Digital Signature Algorithm. Si ritiene che ML-DSA sia sicuro anche contro avversari in possesso di un computer quantistico su larga scala.
- **FIPS 206 (FN-DSA)**: Basato sull'algoritmo FALCON, è un ulteriore standard per firme digitali attualmente in fase di standardizzazione finale. FN-DSA è l'acronimo di FFT (Fast-Fourier Transform) over NTRU-Lattice-Based Digital Signature Algorithm[10]. FALCON utilizza reticoli NTRU e, a differenza degli altri algoritmi selezionati, si basa sull'aritmetica in virgola mobile. Offre firme molto compatte e prestazioni elevate, rendendolo particolarmente adatto a scenari in cui la larghezza di banda è limitata o la velocità è critica.

Gli standard possono e devono essere messi in uso ora. Le organizzazioni sono incoraggiate a iniziare la migrazione verso questi sistemi per proteggersi dalla futura minaccia quantistica.

[7]NIST FIPS 204: Module-Lattice-Based Digital Signature Standard
[8]NIST: Post-Quantum Cryptography Standardization
[9]NIST: NIST Releases First 3 Finalized Post-Quantum Encryption Standards
[10]NIST: FIPS 206 FN-DSA (FALCON)

Parametri e livelli di sicurezza

All'interno degli standard esistono diversi set di parametri che offrono compromessi tra sicurezza e prestazioni:

ML-KEM (ex CRYSTALS-Kyber) Questo standard specifica tre set di parametri per ML-KEM. In ordine di crescente forza di sicurezza e decrescente prestazione, questi sono ML-KEM-512, ML-KEM-768 e ML-KEM-1024:

- **ML-KEM-512**: Livello di sicurezza Categoria 1 (equivalente a AES-128), chiave di incapsulamento di 800 bytes, chiave di decapsulamento di 1632 bytes, ciphertext di 768 bytes.
- **ML-KEM-768**: Livello di sicurezza Categoria 3 (equivalente a AES-192), raccomandato come default dal NIST. Chiave di incapsulamento di 1184 bytes, chiave di decapsulamento di 2400 bytes, ciphertext di 1088 bytes. Offre un ottimo equilibrio tra sicurezza e velocità.
- **ML-KEM-1024**: Livello di sicurezza Categoria 5 (equivalente a AES-256), massima sicurezza. Chiave di incapsulamento di 1568 bytes, chiave di decapsulamento di 3168 bytes, ciphertext di 1568 bytes. Prestazioni ridotte e chiavi più grandi.

Tutti e tre i set di parametri producono una chiave segreta condivisa di 32 bytes (256 bit).

ML-DSA (ex CRYSTALS-Dilithium) Esistono tre versioni: ML-DSA-44, ML-DSA-65 e ML-DSA-87, dove i numeri si riferiscono alle dimensioni della matrice utilizzata nell'algoritmo (rispettivamente matrici 4×4, 6×5 e 8×7), corrispondenti a diversi livelli di sicurezza (rispettivamente Categorie 2, 3 e 5).

Algoritmi alternativi Esistono anche altri algoritmi basati sui reticoli che non sono stati selezionati come standard primari ma che sono serviti come candidati o alternative, come NTRU, SABER e FrodoKEM. Ad esempio, NTRU è basato su problemi di reticoli ma con una struttura matematica differente che lo rende una potenziale alternativa in caso di vulnerabilità scoperte in ML-KEM.

5.2 Code-based

Crittografia basata sui codici (Code-based)[11]

Si basa sulla difficoltà di decodificare un codice lineare casuale.

- Classic McEliece: È l'algoritmo più antico (proposto nel 1978) e ha resistito alla crittanalisi classica e quantistica per oltre 40 anni. Lo svantaggio principale risiede nelle dimensioni delle chiavi pubbliche estremamente grandi (spesso nell'ordine dei megabyte).
- HQC: Recentemente selezionato dal NIST per la futura standardizzazione.

[11] Wikipedia: Code-based cryptography

5.3 Hash-based

Crittografia basata su Hash (Hash-based)[12]

Questi algoritmi creano firme digitali basandosi esclusivamente sulla sicurezza delle funzioni hash crittografiche.

- SPHINCS+ (SLH-DSA): È una firma digitale standardizzata che non richiede assunzioni matematiche complesse se non la resistenza alle collisioni dell'hash scelto. Le firme sono più grandi rispetto ad altri schemi (˜40 KB), ma la sicurezza è ritenuta molto solida.
- XMSS / LMS: Schemi di firma "stateful", adatti per scenari specifici come gli aggiornamenti del firmware.

[12]NIST FIPS 205: Stateless Hash-Based Digital Signature Standard

5.4 Multivariate

Crittografia multivariata (Multivariate)[13]

Si basa sulla difficoltà di risolvere sistemi di equazioni polinomiali multivariate, che è un problema NP-difficile.

- Rainbow: Uno schema di firma che offre firme molto piccole e processi di firma rapidi. Tuttavia, la sua sicurezza è stata messa in discussione da recenti attacchi algebrici.

[13]Wikipedia: Multivariate and Isogeny-based cryptography

5.5 Isogeny-based

Crittografia basata sulle isogenie (Isogeny-based)

Utilizza le proprietà delle mappe (isogenie) tra curve ellittiche supersingolari.

- CSIDH / SIDH: Offrono le chiavi più piccole tra tutti i candidati PQC, ma i tempi di calcolo sono generalmente più lenti. Nota: lo schema SIDH/SIKE è stato violato nel 2022 da un attacco classico, sebbene altre costruzioni basate su isogenie rimangano valide.

5.6 Crittografia simmetrica

Crittografia simmetrica e resistenza quantistica[14]

Per quanto riguarda i sistemi a chiave simmetrica (come l'AES) e le funzioni hash, come abbiamo accennato nel capitolo 4, sono già intrinsecamente resistenti agli attacchi quantistici. Infatti abbiamo brevemente introddo l'algoritmo di Grover che fornisce un'accelerazione quadratica per le ricerche brute-force ma non rappresenta veramente una grossa minaccia come shor. Pertanto per mantenere un livello di sicurezza di 128 bit, è sufficiente raddoppiare la lunghezza della chiave (utilizzando AES-256 anziché AES-128).

Considerazioni sulla migrazione Per mitigare i rischi, molte aziende (come Apple con PQ3 o Google) stanno adottando un approccio di crittografia ibrida, combinando un algoritmo classico con uno post-quantistico per garantire sicurezza anche nel caso in cui uno dei due si rivelasse vulnerabile in futuro.

[14]Grover's Algorithm and Symmetric Key Lengths

6 Considerazioni finali

6.1 Stato attuale della tecnologia dei computer quantistici

Un computer quantistico reale è un sistema fisico composto da qubit che devono mantenere le loro proprietà quantistiche per tutta la durata del calcolo. Le tecnologie attualmente più diffuse si basano su qubit superconduttori, che operano a temperature estremamente basse, prossime allo zero assoluto, tipicamente dell'ordine di alcune decine di millikelvin, al fine di ridurre il rumore termico e preservare la coerenza quantistica.

Il raggiungimento di tali condizioni richiede l'utilizzo di refrigeratori a diluizione, dispositivi complessi e costosi che rappresentano un primo limite alla scalabilità dei sistemi. Anche minime interferenze ambientali possono causare la *decoerenza*, ovvero la perdita delle proprietà quantistiche dei qubit, compromettendo la correttezza del calcolo.

Un'ulteriore difficoltà riguarda il controllo delle operazioni quantistiche. Le porte devono essere applicate con estrema precisione, poiché errori anche molto piccoli tendono ad accumularsi rapidamente con l'aumentare del numero di qubit e della profondità del circuito.

Per mitigare questo problema si ricorre a tecniche di *correzione d'errore quantistica*. Tuttavia, gli schemi attualmente noti richiedono l'impiego di un numero elevato di qubit fisici per realizzare un singolo qubit logico affidabile, tipicamente dell'ordine di centinaia o migliaia.

A causa dei limiti tecnologici attuali, in particolare dei tempi di coerenza ridotti e dell'elevato tasso di errore, l'esecuzione di algoritmi quantistici complessi su larga scala, come quelli necessari per rompere RSA a 2048 bit, non è ancora praticabile.

Per queste ragioni si ritiene che saranno necessari ancora diversi anni, se non decenni, prima che computer quantistici in grado di compromettere concretamente i sistemi crittografici attuali diventino disponibili. Nonostante ciò, i progressi tecnologici e gli investimenti in corso rendono questa prospettiva rilevante dal punto di vista della sicurezza a lungo termine.

6.2 Stato attuale dei protocolli post-quantistici