ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA

Corso di Laurea in Ingegneria e Scienze Informatiche

Post-Quantum Cryptography

Tesi di laurea in:
Crittografia

Relatore
Prof. Margara Luciano

Candidato
Olivieri Michele

III Sessione di Laurea
Anno Accademico 2024-2025

Abstract

Max 2000 characters, strict.

The enemy knows the system.
Claude Shannon

Contents

List of Figures

List of Listings

Chapter 1

Introduction

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 [NC10].

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 [Sho94] 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 tecnologici e lo sviluppo dei primi prototipi avanzati da parte di leader industriali

come Google e Microsoft, che hanno riportato risultati significativi con i loro progetti: Willow[1] di Google e il processore Maiorana 1[2] di Microsoft. 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 [Ber09].

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[3].

Structure of the Thesis Lo scopo della relazione invece è quindi quello di fornire una panoramica completa, introducendo in primo luogo i fondamenti della crittografia classica, per capire dove risiedono le vulnerabilità che i computer quantistici sono in grado di sfruttare. Una volta capiti tali principi, procederemo ad analizzare Shor, il suo impatto su tali algoritmi ed infine le soluzioni proposte dalla crittografia post-quantistica.

[1] Google Quantum AI, "Willow: A quantum computing milestone", Dicembre 2024. Disponibile su: blog.google

[2] Microsoft Quantum, "Maiorana 1: The first milestone on the path to a quantum supercomputer", Ottobre 2024. Disponibile su: www.microsoft.com

[3] NIST, "Post-Quantum Cryptography Standardization", `https://csrc.nist.gov/projects/post-quantum-cryptography`

Chapter 2

Fondamenti di crittografia classica

Come anticipato, 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, in particolar modo ci concentreremo sui protocolli a chiave pubblica.

2.1 Crittografia a chiave pubblica: RSA, ECC

La crittografia a chiave pubblica, introdotta nel 1976 da Diffie, Hellman e Merkle [DH76], 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 [RSA78], 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[1].

[1]Materiale didattico del corso di Crittografia, Università di Bologna, a.a. 2024/2025

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 [Nat20]. 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 [Kob87, Mil85], 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[2].

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.

[2]Materiale didattico del corso di Crittografia, Università di Bologna, a.a. 2024/2025

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 [Nat20]. 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 [LL93]
- **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 [Pol78]

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.

Bibliography

[AAC+22] Gorjan Alagic, Daniel Apon, David Cooper, Quynh Dang, Thinh Dang, John Kelsey, Jacob Lichtinger, Yi-Kai Liu, Carl Miller, Dustin Moody, et al. Status report on the third round of the nist post-quantum cryptography standardization process. NIST Interagency Report NISTIR 8413, National Institute of Standards and Technology, 2022.

[BBF+19] Nina Bindel, Jacqueline Brendel, Marc Fischlin, Brian Goncalves, and Douglas Stebila. Transitioning to a quantum-resistant public key infrastructure. In *International Conference on Post-Quantum Cryptography*, pages 384–405. Springer, 2019.

[Ber09] Daniel J Bernstein. Introduction to post-quantum cryptography. In *Post-Quantum Cryptography*, pages 1–14. Springer, 2009.

[BL17] Daniel J Bernstein and Tanja Lange. Post-quantum cryptography. *Nature*, 549(7671):188–194, 2017.

[DH76] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[KKY+19] Philip Krantz, Morten Kjaergaard, Fei Yan, Terry P Orlando, Simon Gustavsson, and William D Oliver. A quantum engineer's guide to superconducting qubits. *Applied Physics Reviews*, 6(2):021318, 2019.

[Kob87] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, 48(177):203–209, 1987.

[LL93] Arjen K Lenstra and Hendrik W Lenstra. *The Development of the Number Field Sieve*, volume 1554 of *Lecture Notes in Mathematics*. Springer, 1993.

[Mil85] Victor S Miller. Use of elliptic curves in cryptography. In *Conference on the Theory and Application of Cryptographic Techniques*, pages 417–426. Springer, 1985.

[Mos18] Michele Mosca. Cybersecurity in an era with quantum computers: Will we be ready? *IEEE Security & Privacy*, 16(5):38–41, 2018.

[Nat20] National Institute of Standards and Technology. Recommendation for key management: Part 1 – general. Special Publication 800-57 Part 1 Rev. 5, NIST, 2020.

[NC10] Michael A Nielsen and Isaac L Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 10th anniversary edition, 2010.

[Pol78] John M Pollard. Monte carlo methods for index computation (mod p). *Mathematics of Computation*, 32(143):918–924, 1978.

[Pre18] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM*, 56(6):1–40, 2009.

[RSA78] Ronald L Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[Sho94] Peter W Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 124–134. IEEE, 1994.

Acknowledgements

Optional. Max 1 page.