Liceo Statale “Enrico Medi” - Indirizzo Scientifico delle Scienze Applicate
Esame di Stato 2020-2021
d’Atri Sofia 5G

Crittografia classica e quantistica

Indice

Introduzione

Da sempre l’uomo ha avuto la necessità di nascondere dei messaggi per mandare dati segreti senza che nessuno al di fuori del destinatario potesse leggerli.

In Egitto sono stati rinvenuti tentativi di scrittura crittografica risalenti a più di **4500 anni fa**, così come in antica Grecia, a Roma e in India, dove nacquero tecniche crittografiche di rilevante importanza, come il **cifrario di Cesare**. Anche gli arabi, gli ebrei e le prime comunità cristiane ne fecero largo uso, per proteggere le informazioni o attaccare la cultura dominante di nascosto. Spesso, però, si trattava di metodi riconducibili più alla **steganografia** (nascondere l’esistenza di un messaggio) che alla **crittografia** (nascondere il contenuto di un messaggio).

Il contributo degli arabi alla crittologia fu fondamentale: il documento più antico sulla crittologia risale intorno all’**800 d.C.**, scritto dallo scienziato arabo **al-Kindī**. Si dice che il suo predecessore, al-Khalīl, abbia scritto un libro sull’argomento un secolo prima, il quale però è andato perduto.

Nei secoli la crittografia si è raffinata, da Atbash a Enigma, fino ad arrivare ai protocolli di sicurezza usati al giorno d’oggi per tenere i nostri dati sensibili al sicuro. Ma con il veloce sviluppo informatico e l’avvento dei computer quantistici, per quanto ancora basterà la crittografia odierna per proteggerli?

Crittografia classica – RSA

Nel 1976, due crittologi americani, **Whitfield Diffie** e **Martin Hellman**, pubblicarono il primo sistema di crittografia a **chiave asimmetrica**. Tale sistema presupponeva uno scambio di chiavi tra mittente e destinatario, ma non fu definito un modo pratico per applicarlo.

Nel 1977, **Ronald Rivest**, **Adi Shamir** e **Leonard Adleman**, matematici del MIT, definirono un metodo di cifratura basato sulla pubblicazione di Diffie e Hellman, che chiamarono **RSA**. Oggi RSA si usa per i certificati del protocollo https e per generare firme digitali, come PGP.

La crittografia RSA si basa sull'esistenza di due chiavi distinte e dipendenti, dette pubblica e privata, generate da un ente che ne garantisce l'**autenticità**. La **chiave pubblica** è chiamata così in quanto verrà resa pubblica dal suo proprietario, mentre quella **segreta** verrà custodita. Quando un messaggio viene cifrato con una chiave, potrà essere decifrato solo con l'altra e viceversa. Inoltre, nonostante le chiavi siano dipendenti l'una dall'altra, è impossibile risalire dall'una all'altra. In questo modo viene garantita l'**integrità** della crittografia.

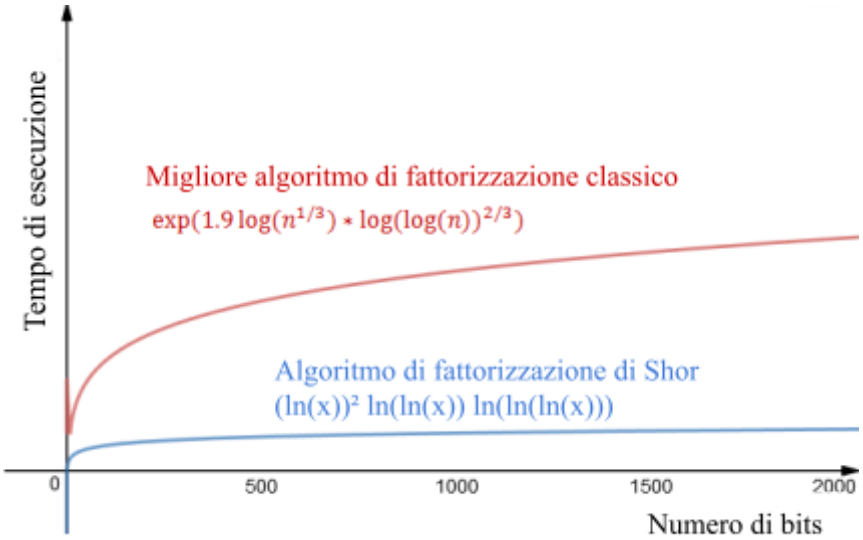
La crittografia RSA è sicura poiché le chiavi, di 2048 bit, sono generate moltiplicando tra di loro due **numeri primi** molto grandi, con centinaia di cifre decimali. La sicurezza è basata sulla **difficoltà di fattorizzare** un numero tanto grande nei suoi fattori primi in un tempo ragionevole. Il migliore algoritmo di fattorizzazione di cui disponiamo adesso ha una complessità computazionale **esponenziale** $O(exp[c\,N^{1/3}(log\,N)^{2/3}])$, dove N è il numero di bit necessari a descrivere il numero da fattorizzare. Le chiavi a 512 bit si possono ricavare in poche ore, mentre quelle a 1024 bit possono essere fattorizzate in un anno.

Ma adesso che è entrata in gioco la computazione quantistica, e con essa infinite nuove possibilità da esplorare, RSA non è più al sicuro.

Algoritmo di fattorizzazione di Shor

Peter Williston Shor, nato a New York il 14 agosto 1959, è un professore di matematica applicata al MIT. Noto per il suo lavoro nell'informatica quantistica, nel 1994 inventò l'algoritmo che oggi prende il suo nome.

L'**algoritmo di fattorizzazione di Shor** è un algoritmo quantistico per fattorizzare i numeri, a una velocità **esponenzialmente maggiore** rispetto al miglior algoritmo attualmente in uso sui computer classici. Con la risoluzione classica, infatti, si ha una complessità computazionale **esponenziale**, mentre quella dell'algoritmo di Shor è **logaritmica** $O((log\,N)^2(log\,log\,N)(log\,log\,log\,N))$.



Il problema di fattorizzare il prodotto di due numeri primi è alla base della sicurezza dei nostri computer. Questo algoritmo ha quindi serie implicazioni per la sicurezza informatica. Prima di addentrarsi nel funzionamento dell'algoritmo di Shor, è necessaria un'introduzione ai computer quantistici, per comprendere cosa rende la loro computazione così potente.

Da bits a qubits

Gli stati classici per la computazione sono "0" **o** "1" e si possono rappresentare con i bits.

In meccanica quantistica, lo stato di un quantum bit (comunemente detto **qubit**) può essere in una **sovrapposizione**, ovvero può essere "0" **e** "1" simultaneamente.

Le sovrapposizioni permettono di effettuare **calcoli in più stati allo stesso tempo** e di creare algoritmi quantici con un **aumento di velocità esponenziale**. Al contrario delle macchine classiche, che possono effettuare un solo calcolo alla volta e dare un singolo risultato per ogni input, i computer quantistici possono calcolare più risposte per un singolo input usando le sovrapposizioni. In altre parole, grazie alla sovrapposizione di stati, il computer quantistico è in grado di processare più soluzioni ad un singolo problema tramite **calcoli paralleli**, al contrario dei calcoli sequenziali delle macchine classiche.

Tuttavia, quando si misura uno stato in sovrapposizione, esso **collasserà** su uno dei suoi stati, con diversa probabilità per ognuno, dando come risultato **una sola delle risposte possibili**. Ritengo sia importante chiarire un punto spesso frainteso: lo stato di un qubit non è sia "0" che "1" **allo stesso tempo**, ma ha una probabilità p(0) di collassare su "0" e una probabilità p(1) di collassare su "1" alla misurazione. La probabilità di collassare su uno qualunque dei suoi stati è quello che si intende quando si dice che "*un qubit è 0 e 1 simultaneamente*".

Nell'algoritmo quantistico, si creano sovrapposizioni con cui calcolare tutte le risposte possibili **simultaneamente**, impostandole in modo da poter sfruttare gli **effetti di interferenza**: gli elementi non necessari formeranno un'**interferenza distruttiva** e si cancelleranno tra di loro, mentre quelli richiesti creeranno un'**interferenza costruttiva** e si avranno come risultato alla misurazione.

Sfera di Bloch

Lo stato di un qubit si può rappresentare usando una sfera di raggio 1, chiamata **Sfera di Bloch**. Tutti gli stati definiti **puri** (corrispondenti cioè a vettori di base) possono essere illustrati sulla sua **superficie**, mentre esistono degli stati chiamati **misti** che si trovano all'interno. In questo caso ci concentreremo solo sugli stati puri.

Un qualunque stato quantico puro viene scritto come

$$|\psi\rangle = cos\frac{\theta}{2}|0\rangle + e^{i\varphi} sin\frac{\theta}{2}|1\rangle$$

dove $\varphi \in [0, 2\pi)$ descrive la **fase relativa** e $\theta \in [0, \pi]$ determina la **probabilità** di misurare zero $|0\rangle$ o uno $|1\rangle$:

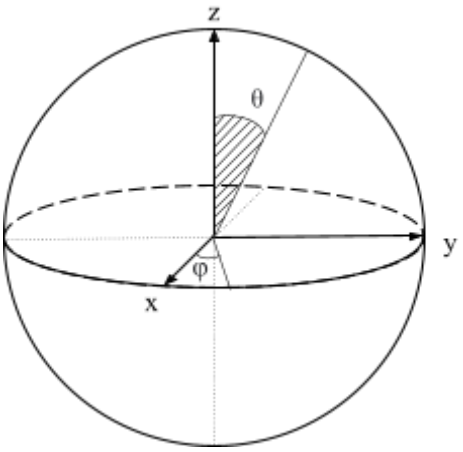
$$p(0) = \left(cos\frac{\theta}{2}\right)^2,$$

$$p(1) = \left(sin\frac{\theta}{2}\right)^2.$$

La fase relativa è l'angolo tra gli stati zero $|0\rangle$ e uno $|1\rangle$. Grazie ad esso si ottengono gli **effetti di interferenza**.

Le coordinate di tali stati sono date da un vettore $\vec{r}$:

$$\begin{cases} x = sin\theta cos\theta \\ y = sin\theta sin\varphi \\ z = cos\theta \end{cases}$$



Usiamo queste formule per calcolare le coordinate dei principali stati $|0\rangle$ e $|1\rangle$.

Il primo stato è $|0\rangle$. La probabilità di misurare $|0\rangle$ deve essere uguale a 1, mentre la probabilità di misurare $|1\rangle$ deve essere uguale a 0: θ dovrà essere uguale a 0°.

$$p(0) = \left(cos\frac{0}{2}\right)^2 = 1,$$

$$p(1) = \left(sin\frac{0}{2}\right)^2 = 0.$$

Sostituendo $\theta = 0$ nella formula del generico stato, si ha come risultato proprio lo stato $|0\rangle$:

$$|\psi\rangle = cos\frac{0}{2}|0\rangle + e^{i\varphi}sin\frac{0}{2}|1\rangle = 1|0\rangle + 0|1\rangle = |0\rangle$$

Di conseguenza, le coordinate saranno:

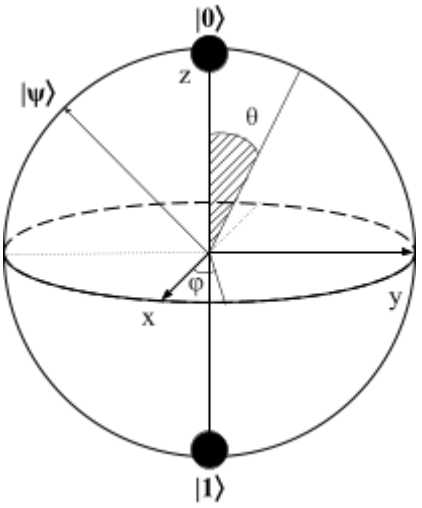
x = $sin\theta cos\theta = 0$

y = $sin\theta sin\varphi = 0$

z = $cos\theta = 1$

Con calcoli analoghi si ricava che le coordinate dello stato $|1\rangle$ sono (0, 0, -1).

Gli stati $|0\rangle$ e $|1\rangle$ si dicono **ortogonali**. Due stati ortogonali costituiscono una **base**. Le basi ortogonali sono usate per descrivere e misurare gli stati quantici. Ad esempio, se si misura sulla base $\{|0\rangle; |1\rangle\}$, lo stato collasserà su $|0\rangle$ o $|1\rangle$. In particolare, nell'immagine a destra, il generico stato $|\psi\rangle$ collasserà con **maggiore probabilità** su $|0\rangle$.



Guardando l'immagine si nota anche come non sia necessario calcolare l'angolo φ per questi due stati: compierebbero una rotazione sul proprio asse, senza modificare la loro posizione.

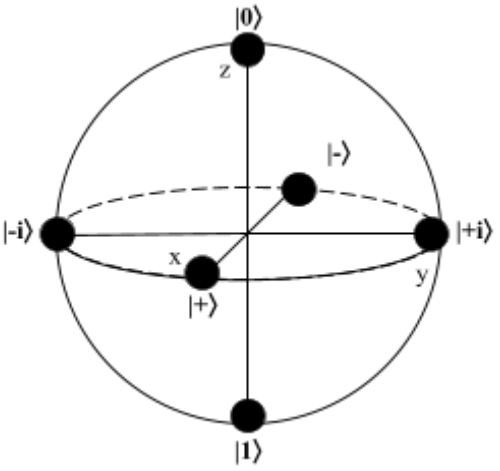
Esistono infinite basi diverse, ma le principali, oltre a $\{|0\rangle; |1\rangle\}$, sono:

$|+\rangle: \theta = \frac{\pi}{2}, \varphi = 0, \quad \vec{r} = (1, 0, 0)$

$|-\rangle: \theta = \frac{\pi}{2}, \varphi = \pi, \quad \vec{r} = (-1, 0, 0)$

$|+i\rangle: \theta = \frac{\pi}{2}, \varphi = \frac{\pi}{2}, \quad \vec{r} = (0, 1, 0)$

$|-i\rangle: \theta = \frac{\pi}{2}, \varphi = \frac{3\pi}{2}, \quad \vec{r} = (0, -1, 0)$



Una **misurazione** nella base $\{|0\rangle; |1\rangle\}$ è anche detta **misurazione z**, poiché viene effettuata sull'asse z. Analogamente, se avviene nella base $\{|+\rangle; |-\rangle\}$ viene detta **misurazione x**, mentre nella base $\{|+i\rangle; |-i\rangle\}$ si ha una **misurazione y**.

Protocollo per l'algoritmo di Shor

Il protocollo per l'algoritmo di Shor richiede una parte di esecuzione su un **sistema classico** e una parte su un **sistema quantistico**. In generale, svolge i seguenti passaggi.

1. Si sceglie un numero casuale *a* che sia **coprimo**, ovvero che non abbia divisori in comune, con *N*. Se così non fosse, sarebbe facile fattorizzare *N*, ma è **estremamente improbabile** che un numero casuale sia coprimo con *N*.
2. Per ogni coppia di numeri coprimi, se si moltiplica uno dei due per se stesso abbastanza volte, si otterrà un multiplo dell'altro numero sommato a uno, $a^r = mN + 1$, dove *r* e *m* sono numeri interi.

   Questa equazione può essere riscritta come $a^r \equiv 1\ (mod\ N)$.
3. Riordinando l'equazione, può essere letta come *"un numero moltiplicato per un numero è uguale a un multiplo di N"*, ovvero i **fattori di N**.

   $a^r - 1 = mN$

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

   Calcolando il massimo comun divisore, si troverà almeno uno dei fattori di N in $\{MCD(x + 1,\ N);\ MCD(x - 1,\ N)\}$, dove $x \equiv a^{r/2}(mod\ N)$.

Per poter calcolare $a^{r/2}$, è necessario che ***r* sia pari**. In caso non lo fosse, è necessario cercare un altro valore *a* e ricominciare dall'inizio. Fortunatamente, per ogni numero di partenza, almeno il **37,5%** delle volte $a^{r/2} \pm 1$ è un fattore di N ed è possibile trovare entrambi i fattori con **meno di 10 tentativi**. Si tratta di calcoli che possono essere svolti anche su un **sistema classico**, ma, se N ha migliaia di cifre e *a* ne ha 500, cercare di capire a che potenza *r* elevare *a* affinché $a^r \equiv 1\ (mod\ N)$ richiederebbe **tantissimo tempo**, più di quanto ne servirebbe per fattorizzare N a forza bruta.

Ricerca del periodo di una funzione - Protocollo quantistico

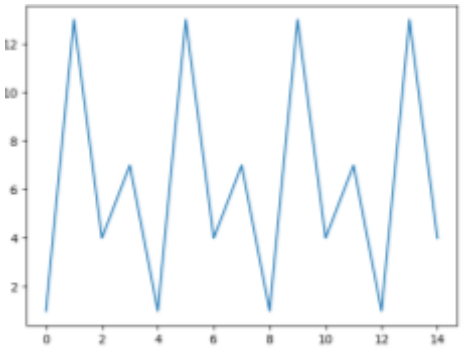
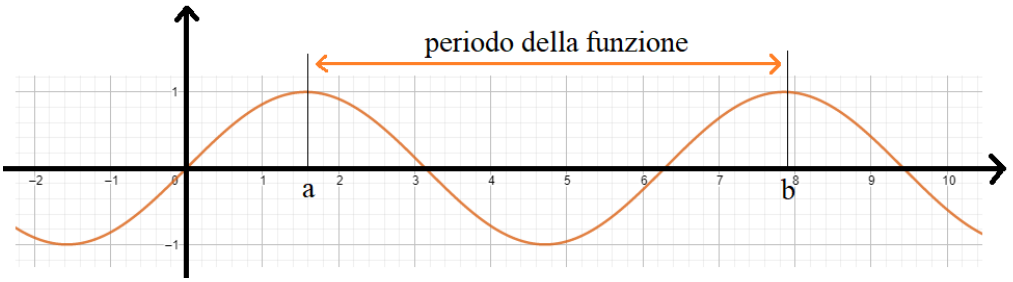


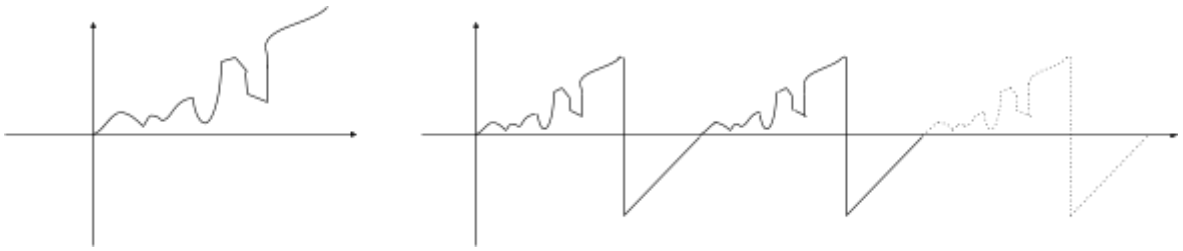
Grafico della funzione $y=a^x(mod\ N)$.

La potenza *r* è il **periodo della funzione** $y = a^x(mod\ N)$. Il periodo di una funzione è il più piccolo numero reale positivo che realizza l'uguaglianza *f(x+r) = f(x)* al variare di *x*.

Per alcune funzioni, trovare il periodo è immediato. Il periodo della funzione $y = sin(x)$ è facilmente intuibile dal grafico e corrisponde a 2π.



Tuttavia, non è sempre così semplice trovare il periodo di una funzione: considerando un numero insufficiente di valori, la periodicità di una funzione può non apparire evidente.



La **ricerca del periodo di una funzione** non può essere svolta in maniera efficiente da un computer classico: è grazie al contributo quantistico nella ricerca del periodo che si ha un aumento di velocità così elevato nell'algoritmo di fattorizzazione.

Per comprendere meglio il protocollo quantistico, ogni passaggio sarà accompagnato dall'esempio di applicazione dell'algoritmo alla ***ricerca dei fattori di 15****.*

Per iniziare il calcolo, si imposta un computer in modo che accetti un numero *x* come input e calcoli $w = a^{x}(mod\ N)$. Dal momento che si tratta di un computer quantistico, può ricevere in input una **sovrapposizione di diversi numeri** ed effettuare il calcolo **simultaneamente**, ottenendo una sovrapposizione di **tutte le possibili potenze *x*** a cui *a* può essere elevata e una sovrapposizione di **tutti i risultati** di $w = a^{x}(mod\ N)$.

$N = 15 = 5 \cdot 3$, $a = 13$, x = *sovrapposizione di tutti i valori [0;15],*

$w = a^{x}(mod\ N)$

x	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*
w	*1*	*13*	*4*	*7*	*1*	*13*	*4*	*7*	*1*	*13*	*4*	*7*	*1*	*13*	*4*	*7*

Per trovare il periodo *r* si sfrutta una semplice **osservazione**: elevando *a* a una potenza casuale *p*, il risultato sarà *"un numero w in più di un multiplo di N"*, ovvero $a^{p} = m\ N\ +\ w$, che può essere letto come il resto $w = a^{p} (mod\ N)$. Elevando *a* a *p+r*, il risultato sarà ancora *"un numero w in più di un multiplo di N"*: nonostante si tratti di un multiplo diverso, il resto *w* non cambia. Lo stesso si verifica elevando *a* a *p+2r*. E così via.

In questo caso, il periodo è $r = 4$. *Scegliamo come potenza di partenza* $p = 3$*: notiamo che sommando* ***r*** *a* ***p*** *il resto* ***w*** *è sempre* $w = 7$.

x	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*
w	*1*	*13*	*4*	*7*	*1*	*13*	*4*	*7*	*1*	*13*	*4*	*7*	*1*	*13*	*4*	*7*

La potenza *r* ha una **proprietà ripetitiva** tale per cui, se si aggiunge o sottrae *r* a un'altra potenza, $w = a^{x} (mod\ N)$ rimane lo stesso. Questa proprietà è una **relazione strutturale** tra diverse potenze e il computer quantistico può trarne vantaggio proprio perché i calcoli quantistici possono essere eseguiti **simultaneamente** su sovrapposizioni di diversi valori possibili. Misurando solo la parte corrispondente a *w*, si otterrà casualmente **uno dei resti possibili** come output. Il numero specifico non è rilevante: l'importante è che è rimasta una sovrapposizione di potenze per cui $w = a^{x} (mod\ N)$ risulta **uguale**.

Alla misurazione otteniamo un valore w casuale tra 1, 13, 4 o 7 con uguale probabilità. Per questo esempio, misuriamo w = 7, quindi il valore di x diventa la sovrapposizione di 3, 7, 11, 15.

Questa è una delle proprietà principali del calcolo quantistico: data una sovrapposizione e ottenuta una risposta che potrebbe provenire da più di un elemento della sovrapposizione, si rimarrà con una **sovrapposizione di quegli elementi**. A causa della proprietà ripetitiva, quelle potenze sono tutti numeri che sono separati l'uno dall'altro da una **distanza *r***. Alla fine di questi calcoli risulta una sovrapposizione di numeri che si ripetono periodicamente con un periodo di *r*, o equivalentemente, si ripetono con una frequenza di *1/r*. Trovando la frequenza, si possono trovare i fattori per rompere la crittografia.

Trasformata di Fourier Quantistica (QFT)

Qui entra in gioco la Trasformata di Fourier Quantistica (**Quantum Fourier Transform, QFT**), il migliore strumento per trovare le frequenze. La Trasformata di Fourier Quantistica può essere applicata alla sovrapposizione che si ripete con una frequenza di *1/r* per causare tutte le possibili

frequenze sbagliate di interferire distruttivamente, lasciando come unico stato quantico il numero *1/r*. Quando si fa la QFT di una sovrapposizione di valori, si otterrà una **sovrapposizione di sovrapposizioni** e le onde sinusoidali si sommano o si sottraggono, cancellandosi a vicenda.

Effettuando la QFT della sovrapposizione x, ricaviamo che r = 4.
Di tutti i risultati possibili, ne è rimasto solo uno perché vi è stato un uguale contributo di valori positivi e negativi. La risposta converge verso il valore che interessa a noi.

Se i numeri in sovrapposizione sono separati da una quantità *r*, tutte le onde sinusoidali interferiranno e daranno come risultato della misurazione il singolo stato quantico che rappresenta *1/r*. Una volta trovato *r*, si può calcolare $x \equiv a^{r/2}(mod\ N)$ e trovare in $\{MCD(x+1,\ N);\ MCD(x-1,\ N)\}$ almeno uno dei fattori di *N*.

Notiamo che r è pari, quindi possiamo trovare $x \equiv a^{r/2}(mod\ N) = 13^{4/2}(mod\ 15) = 4$.
Ora possiamo calcolare il massimo comun divisore:
$\{MCD(x+1,\ N);\ MCD(x-1,\ N)\} = \{MCD(5,\ 15);\ MCD(3,\ 15)\} = \{5;\ 3\}$,
trovando così entrambi i fattori di N.

Conclusioni

Un computer classico dovrebbe provare una a una tutte le potenze possibili e richiederebbe un'enorme quantità di tempo, soprattutto per numeri tanto grandi quanto quelli usati per la crittografia. La versione quantistica è estremamente più veloce e, se un computer quantistico abbastanza potente dovesse essere mai costruito, con l'**algoritmo di Shor** chiunque potrebbe decifrare dati criptati con un sistema basato sulla fattorizzazione di grandi numeri primi - un'incombente minaccia per tutta la rete Internet.

Tuttavia, per il funzionamento di questo algoritmo servono **due volte** il numero di qubits necessari per descrivere il numero da fattorizzare. Per un numero a 2048 cifre sarebbero necessari 4096 qubits e i computer quantistici più potenti in uso al momento non ne hanno ancora così tanti: il **Quantum Hummingbird processor** di **IBM** funziona con 65 qubit, mentre il **Bristlecone processor** di **Google** ne usa 72. Quindi, l'algoritmo di Shor **non è implementabile** sui computer quantistici di cui disponiamo oggi, se non per numeri piccoli come 15: per adesso non c'è bisogno di preoccuparsi di questo potente algoritmo.

In conclusione, l'avvento e lo sviluppo della meccanica quantistica porteranno al declino di molti dei sistemi che si usano oggi, ma porteranno anche alla nascita di nuovi sistemi, capaci di resistere agli attacchi quantistici o costituiti essi stessi da algoritmi quantici.

Peter Shor ebbe il merito di dimostrare nel 1994 le capacità dei computer quantistici ancor prima della loro nascita: grazie a lui, l'interesse del mondo scientifico per questo settore è salito alle stelle. Oggi la ricerca è un fiume in piena e lo sviluppo è inarrestabile.

Il futuro è quantistico.

Bibliografia

IBM Quantum team, Introduction to Quantum Computing and Quantum Hardware (2020).
http://qiskit.org/learn/intro-qc-qh

IBM's Roadmap For Scaling Quantum Technology (2020), IBM, Jay Gambetta.
https://www.ibm.com/blogs/research/2020/09/ibm-quantum-roadmap/

MIT Mathematics, Peter Shor.
https://math.mit.edu/directory/profile.php?pid=247

"Storia della crittografia." Wikipedia, L'enciclopedia libera.
https://it.wikipedia.org/wiki/Storia_della_crittografia

Il cifrario RSA, Paolo Bonavoglia.
http://www.crittologia.eu/critto/rsa/rsa.html

Ibrahim A. Al-Kadit (1992): ORIGINS OF CRYPTOLOGY: THE ARAB CONTRIBUTIONS, Cryptologia, 16:2, 97-126
https://www-ljk.imag.fr/membres/Bernard.Ycart/mel/hm/AlKadi_cryptology.pdf

Quantum Algorithm Zoo (2011-2021), Stephen Jordan.
https://quantumalgorithmzoo.org/

A Preview of Bristlecone, Google's New Quantum Processor (2018), Julian Kelly.
https://ai.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html

How Quantum Computers Break Encryption | Shor's Algorithm Explained (2019), Henry Reich.
https://youtu.be/lvTqbM5Dq4Q