Serviço Público Federal
Ministério da Educação
Fundação Universidade Federal de Mato Grosso do Sul

Curso de Física - Bacharelado

Uma Análise Física e Computacional do Impacto da Computação Quântica na Criptografia

Bruno Marinho Maciel

Orientação: Prof. Dr. João Vitor Batista Ferreira

Instituto de Física
Universidade Federal de Mato Grosso do Sul
23 de março de 2026

Serviço Público Federal
Ministério da Educação
Fundação Universidade Federal de Mato Grosso do Sul

Curso de Física - Bacharelado

Uma Análise Física e Computacional do Impacto da Computação Quântica na Criptografia

Bruno Marinho Maciel

Orientação: Prof. Dr. João Vitor Batista Ferreira

Monografia apresentada ao Instituto de Física da Universidade Federal de Mato Grosso do Sul como requisito da qualificação para a obtenção do título de bacharelado em Física.

Instituto de Física
Universidade Federal de Mato Grosso do Sul
23 de março de 2026

"Aguardo, equânime, o que não conheço —
Meu futuro e o de tudo.
No fim tudo será silêncio, salvo
Onde o mar banhar nada."

Fernando Pessoa

Agradecimentos

É com imensa gratidão que finalizo essa etapa da minha trajetória acadêmica. Gratidão a todos que vieram antes de mim e tornaram tudo isso possível.

Apesar das dificuldades que foram enfrentadas e, quiçá, por sorte, superadas, chego ao fim ciente de que muito aprendi e, por isso, tenho muito a aprender.

Agradeço a todos os meus professores da graduação que fizeram parte dessa jornada, especialmente à professora Dra. Isabela Porto Cavalcante por ser uma pessoa sempre atenciosa, transparente e me apoiar nos momentos em que precisei de ajuda. Ao professor Dr. Anderson Rodrigues Lima Caires por ter proporcionado 3 anos de iniciação científica e, com eles, incontáveis conversas e conexões.

Agradeço, de forma especial, à professora Dra. Ana Karina Salina por ter me orientado ao longo desses últimos anos de graduação e me apoiado nessa jornada. Agradeço também ao meu orientador Dr. João Vitor Batista Ferreira pelas disciplinas que pude ter com o Sr. e pelo apoio que me deu nesse fim de jornada, assumindo a orientação do meu trabalho.

Agradeço a todas as circunstâncias que me proporcionaram amizades tão especiais ao longo dessa jornada, especialmente a Raphael de Souza Flores e Gabriel Estrella Medeiros que me acompanharam durante períodos de escuridão e esclarecimento; de lágrimas e risadas. Não poderia deixar de falar, também, dessas pessoas por quem carrego muito carinho e apreço: Dr. Ivo Leite Filho, Thailenny Dantas Rezende, Cristian Haas Fretes, Gustavo Alfredo Zamboni e Carlos Henrique Duarte Batista.

Agradeço à minha amada namorada e melhor amiga Júlia Lelis Soares por ter me apoiado e ter sido esse farol na minha vida, esse presente do tempo, e por estar junto comigo em todos os momentos. Ter estado presente mesmo quando estamos longe. Por termos nos guiado mesmo quando estávamos perdidos.

Por fim, mas não menos importante, agradeço ao meu irmão Lucas Henrique Marinho Maciel e aos meus pais Ivon Moraes Maciel e Maria Aparecida Marinho por terem me proporcionado todo suporte ao longo desses anos.

Resumo

Este trabalho consiste em uma revisão bibliográfica introdutória da criptografia clássica e computação quântica. No texto exploramos a formulação dos métodos de cifragem simétrica e assimétrica, enfatizando o algoritmo RSA e sua base limitante no problema matemático da fatoração de inteiros. Introduzimos a teoria quântica através dos postulados da mecânica quântica com notação de Dirac, destacando propriedades operacionais essenciais, como a superposição e a interferência. No âmbito computacional, o estudo descreve as portas lógicas reversíveis e o conceito de paralelismo quântico a partir do algoritmo de Deutsch-Jozsa. Em seguida, detalha-se a estrutura teórica do algoritmo de Shor e o emprego da Transformada de Fourier Quântica na drástica redução da complexidade temporal da fatoração. Conclui-se o texto com uma breve compilação dos obstáculos empíricos para a construção de hardwares viáveis, pautada no desafio imposto pelas rápidas taxas de decoerência frente ao tempo necessário para a execução das operações lógicas nos qubits, dentre outros desafios.

Palavras-Chave: *Computação Quântica, Criptografia*

Abstract

This work consists of a bibliographic and introductory review on the intersection between classical cryptography and quantum computing. The text is structured from the formulation of symmetric and asymmetric encryption methods, emphasizing the RSA algorithm and its limiting basis on the mathematical problem of integer factorization. To substantiate the quantum paradigm, the postulates of quantum mechanics are reviewed through Dirac notation, highlighting essential operational properties, such as superposition and interference. In the computational scope, the study describes reversible logic gates and the concept of quantum parallelism based on the Deutsch-Jozsa algorithm. Next, the theoretical structure of Shor's algorithm and the use of the Quantum Fourier Transform in the drastic reduction of the time complexity of factorization are detailed. The text concludes with a brief compilation of the empirical obstacles for the construction of viable hardware, based on the challenge imposed by the fast decoherence rates against the time necessary for the execution of logical operations on the qubits.
Keywords: *Quantum Computing*, *Cryptography*

Lista de Figuras

Sumário

Capítulo 1

Introdução

Atualmente, utilizamos redes de computadores e, predominantemente, a internet na execução de diversas tarefas que são, fundamentalmente, privadas. Dentre essas atividades, podemos citar: troca de mensagens privadas, armazenamento em nuvem e produção de documentos (oficiais e não oficiais), transações bancárias, dentre muitas outras. Essa pequena lista deixa claro que tanto o cidadão comum, quanto empresas e mesmo entidades governamentais dependem fundamentalmente desse meio interconectado de comunicação para trafegar e armazenar informações sensíveis e privadas [1]. A confiabilidade que temos desse meio hoje é evidente com aspectos como a lei que permite a utilização de assinaturas digitais com todos os fins legais [2], graças ao advento das técnicas modernas de criptografia desenvolvidas no fim do século XX.

O sufixo *cripto* vem do grego *kruptos*, que pode ser traduzido como *esconder*, e há registros de *textos escondidos*, mesmo que de forma mais rudimentar, datados de 4000 anos atrás [1]. O tipo mais antigo de criptografia é a criptografia simétrica, ou seja, que utiliza uma dada chave A para encriptar um texto, e a mesma chave A é capaz de decriptar a cifra. Esse termo é bastante geral e abrangente; afinal, vai desde métodos como a criptografia de César, onde a chave pode ser simplesmente um número inteiro entre 0 e 24 representando o *offset* no seu alfabeto, até técnicas mais avançadas como o *Advanced Encryption Standard (AES)*. Outro tipo de técnica de criptografia bastante utilizada é a criptografia assimétrica, onde as partes que estão se comunicando utilizam chaves distintas para obter um segredo em comum.

A criptografia assimétrica surgiu no último século e trouxe uma forma distinta de garantir a segurança que vai além da aplicação de muitas transformações simples, como o *Exclusive-Or (XOR)* e *bitshifts* consecutivos, que dificultam a obtenção do texto original. A criptografia assimétrica baseia-se na dificuldade matemática de resolver alguma classe de problemas matemáticos. Por exemplo, o algoritmo de *Rivest–Shamir–Adleman (RSA)* tem sua segurança garantida pela dificuldade de fatorar um número inteiro em seus fatores primos. Nesse sentido, decriptar de forma eficiente mensagens trocadas utilizando o RSA é, essencialmente, resolver o problema de fatoração de um número inteiro de forma eficiente e, até onde sabemos, não há algoritmos clássicos computacionalmente eficientes para executar a fatoração.

Com o advento da mecânica quântica no início do século XX, surge, também, uma nova forma de armazenar e processar informações. Em 1982, Richard Feynman introduziu a ideia de utilizar as ferramentas desenvolvidas na mecânica quântica para simular sistemas físicos complexos já sob o nome de computador quântico [3]. Em 1984, um algoritmo foi desenvolvido e o conceito de computador quântico universal foi introduzido [4]. Em 1994, Shor apresentou o seu algoritmo que, se implementado em um computador quântico com capacidade computacional suficiente, pode tornar a implementação moderna do algoritmo RSA insegura [5]. Hoje, a computação quântica é uma área ativa de pesquisa com estudo de algoritmos quânticos, distribuição de chaves quânticas, dentre outros, além de um potencial de aplicações mais amplo que originalmente pensado [6] [7].

Este trabalho tem como objetivo oferecer uma breve introdução dos tópicos de criptografia simétrica e assimétrica, oferecendo exemplos fundamentais no seu respectivo contexto; apresentar a computação quântica enquanto ferramenta para executar a redução da complexidade temporal de operações que garantem a segurança do RSA; e, por fim, apresentar brevemente alguns dos desafios na produção de um computador quântico funcional.

Capítulo 2

Criptografia Clássica

A Criptografia é a técnica usada para assegurar que uma mensagem transmitida de um ponto para outro não possa ser lida ou alterada (de forma imperceptível) por qualquer interceptador daquela mensagem [8].

Ao desenvolver técnicas de criptografia pode-se assumir que haverá mais de um interceptador da mensagem: o intencional e/ou qualquer outro [9]. Desenvolveram-se ao longo dos anos diversas técnicas de criptografia diferentes. Uma cifra historicamente relevante para se encriptar uma mensagem é a cifra de César, descrita a seguir.

Como motivação para este tópico, pode-se definir uma função que esconde um texto/mensagem como: a função $E : M \to C$ que encripta uma mensagem $m \in M$ (onde M é o conjunto de todas as possíveis mensagens) para a cifra $c \in C$ (onde C é o conjunto de todas as cifras C possíveis) é uma função bijetiva nos conjuntos M e C [8], ou seja,

$$\begin{aligned} \forall\ c_1 \in C\ \ \exists\ \ m_1 \in M :\ E(m_1) = c_1 \\ E(m_1) = E(m_2) \iff m_1 = m_2\ . \end{aligned} \tag{2.1}$$

Isso garante que, para uma função E e uma mensagem m, haverá somente uma cifra c que satisfaça $E(m) = c$ e, portanto, garante a existência da inversa $E^{-1}(c) = m | c \in C, m \in M$. Um problema dessa definição é que a função E deve ser de caráter secreto no esquema de proteção dos dados, além da própria mensagem que está sendo criptografada, ou seja, uma vez que se tem acesso à função E, todas as mensagens encriptadas usando E podem ser obtidas trivialmente. Uma das formas de resolver este problema é inserindo uma chave K na função E, de forma que

$$\forall k \in K\ \ \exists\ \ E_k \in E\ , \tag{2.2}$$

Note que, k representa um membro do conjunto K e, portanto, para cada chave $k \in K$ existe um espaço de cifras C_k distinto [8]. A prática de não colocar garantias de segurança no algoritmo é conhecida como *princípio de Kerckhoffs* e pode ser resumido como: um método de criptografia seguro não deve exigir sigilo e deve poder ser de conhecimento público [9].

2.1 Criptografia Simétrica

Dadas as definições sobre a função E, para garantir a comunicação entre o emissor e o receptor intencional da mensagem, deve haver o conhecimento, primeiro, do mecanismo de encriptação E usado para cifrar a mensagem original m e, para o caso da criptografia simétrica, uma chave *privada* que permita a associação da cifra c com uma mensagem m, pois, quando a chave k foi introduzida nota-se que a função agora depende da chave secreta tanto para cifrar uma mensagem quanto para decifrá-la como mostra a Figura 2.1.

Figura 2.1: Exemplo da cifra simétrica

Fonte: Autor

A criptografia simétrica foi a principal forma de criptografia por centenas de anos, por exemplo a cifra de César: consiste da substituição individual de cada uma das letras x_i em um texto legível m pela letra que estivesse $i + k$ posições à frente no alfabeto [1]

$$\begin{aligned} m &= x_1x_2x_3...x_n \ \in \mathbb{Z}; \\ k &\in \mathbb{Z} \mid 1 \leq k \leq 26; \\ c = y_1y_2y_3...y_n &\mid y_i = LET(POS(x_i) + k \mod 26). \end{aligned} \tag{2.3}$$

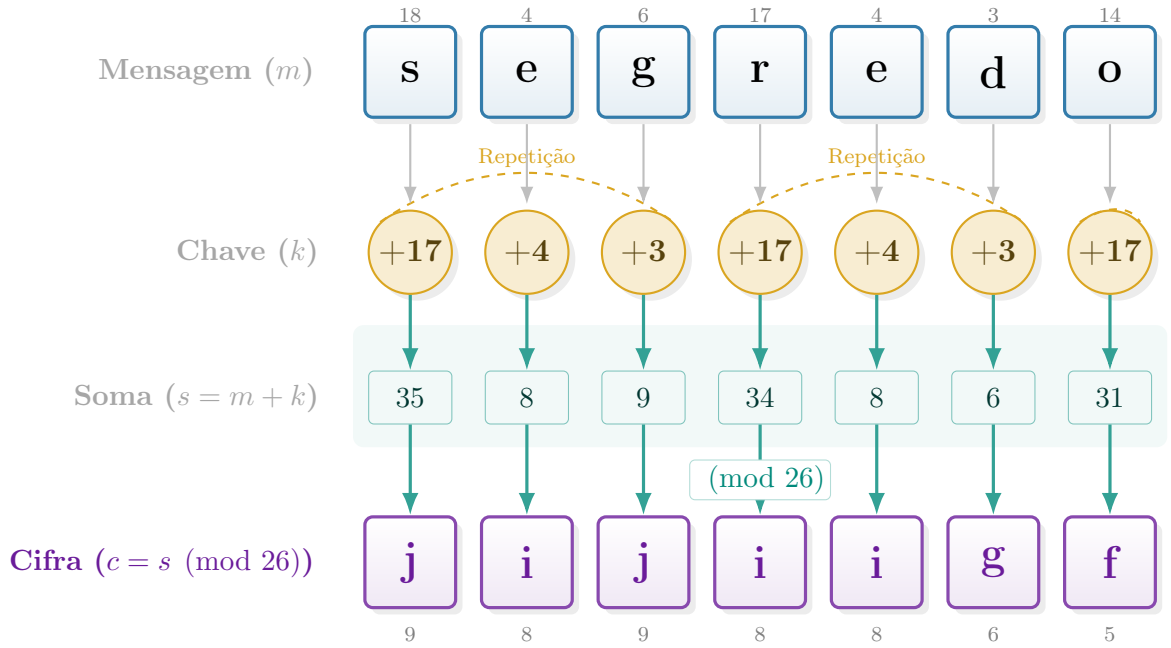
A função $POS(x_i)$ calcula a posição da letra x_i no alfabeto e $LET(m)$ calcula a letra do alfabeto na posição m. O principal problema com a Cifra de César é que a chave possui apenas 25 possibilidades, tornando suas cifras muito vulneráveis à exploração por tentativa e erro. Além disso, quando é analisado a particularidade de uma língua a partir de um conjunto suficientemente grande de dados, pode-se inferir a frequência com a qual certos símbolos aparecem em mensagens/comunicações, tanto de forma geral, quanto em um dado contexto.

Assim, há duas grandes vulnerabilidades que comprometem a segurança na cifra de César: frequência e força bruta. Essas vulnerabilidades estão presentes nessa cifra quanto em qualquer outro algoritmo criptográfico que utilize das técnicas de **substituição** e/temos ou **permutação** das letras com chaves pequenas. Essas preocupações, apesar de parecerem consistentes somente com as comunicações entre duas pessoas, elas são generalizáveis ao contexto digital, observando-se que todos os protocolos possuem cabeçalhos padronizados; além de outras possíveis estruturas no texto que esse tipo de função (apenas substituição) não é capaz de esconder, expondo outro ponto de vulnerabilidade na cifra [8].

Outro exemplo clássico é a Cifra de Vigenère-Vernam: para resolver o problema das cifras que mantêm a estrutura do texto legível, Vernam propôs que, em vez de aplicar chaves reduzidas, utiliza-se uma chave, tão grande quanto o texto legível e que fosse executado operações sobre os componentes da mensagem [8]. A cifra de Vigenère consiste basicamente na execução da cifra de César para cada letra individual da mensagem. Nesse

caso, a chave deve ser um conjunto de n subchaves, cada uma responsável por uma cifra de César diferente. Por exemplo, a chave $n = (17, 4, 3)$ pode ser utilizada na mensagem $m =$ 'segredo' para obtermos $x =$ 'jijiigf', como mostra a Figura 2.2 a seguir [8].

Figura 2.2: Exemplo da cifra de Vigenère-Vernam.

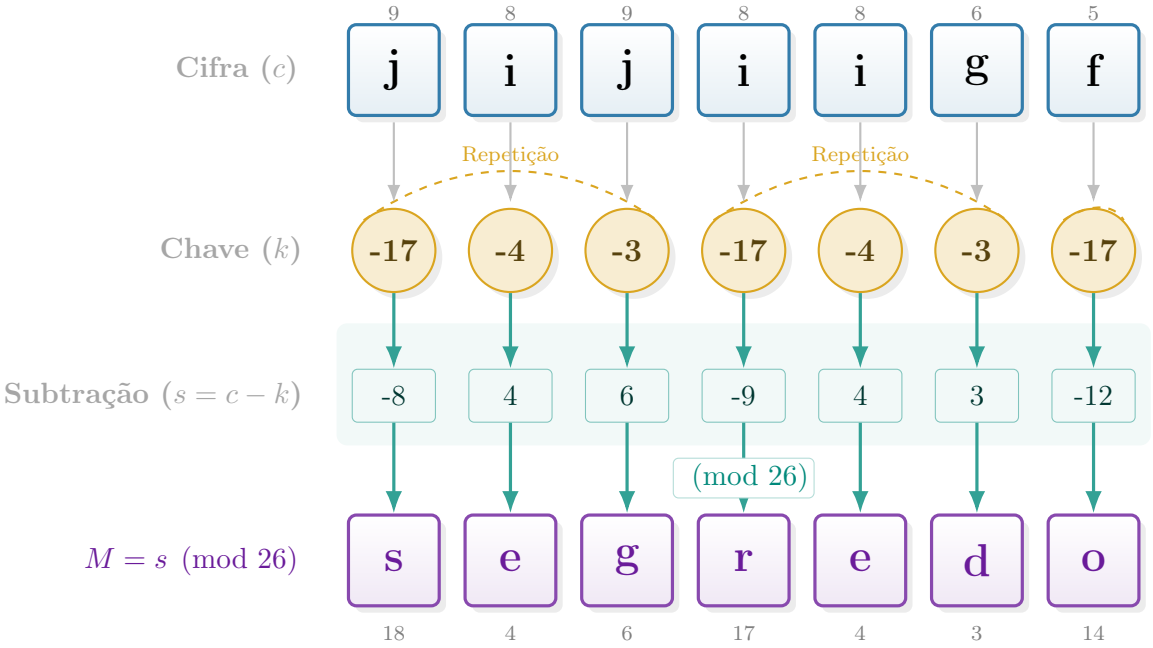


Fonte: Autor (Utilizando recursos de IA).

Nesse caso, pode-se representar chave com a palavra *red*, já que são as respectivas posições das letras do alfabeto representando a chave. Esse exemplo, tratou a mensagem e o processo de cifra, essencialmente como uma soma na base 26 e os menores elementos lógicos como as letras.

Como apresentado na Figura 2.3, para o inverso dessa operação faz-se a subtração modulo 26 para cada elemento da chave.

Figura 2.3: Exemplo da decifração de Vigenère-Vernam

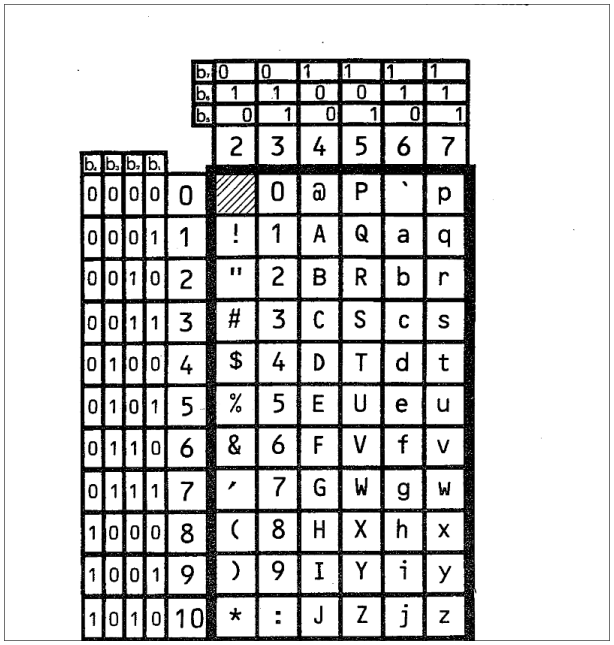


Fonte: Autor (Utilizando recursos de IA)

Para continuidade na tratativa dos algoritmos de criptografia, faz-se introduzir a ideia de conversão de mensagens em cadeias numéricas, ou ainda, no contexto da computação binária, *bitstrings*. As técnicas de conversão de mensagens em cadeias numéricas são chamadas de *encode*. Esse processo pode ser executado de diversas formas diferentes, por exemplo, dentre os algoritmos de *encoding* mais usados, temos o *American National Standard Code for Information Interchange* (ASCII) [10]

No caso do ASCII, então, pode-se mapear para cada número inteiro entre 1 e 127, um caractere ou função única como mostra o segmento da Tabela ASCII na Figura 2.4.

Figura 2.4: Segmento Tabela ASCII



b_4	b_3	b_2	b_1		0 0 1 1 1 1						
					1 1 0 0 1 1						
					0 1 0 1 0 1						
					2	3	4	5	6	7
0	0	0	0	0		0	@	P	`	p
0	0	0	1	1	!	1	A	Q	a	q
0	0	1	0	2	"	2	B	R	b	r
0	0	1	1	3	#	3	C	S	c	s
0	1	0	0	4	$	4	D	T	d	t
0	1	0	1	5	%	5	E	U	e	u
0	1	1	0	6	&	6	F	V	f	v
0	1	1	1	7	'	7	G	W	g	w
1	0	0	0	8	(	8	H	X	h	x
1	0	0	1	9	)	9	I	Y	i	y
1	0	1	0	10	*	:	J	Z	j	z

Fonte: [11]

A letra **B** pode ser representada pela *bitstring* 1000010 em ASCII (binário), ou pelo número inteiro 66 (em base 10). Esse conceito é fundamental para o desenvolvimento das técnicas de criptografia moderna pois permite dar uma tratativa fundamentalmente matemática aos algoritmos.

De tal maneira, pode-se, também ser tratado os menores elementos lógicos como bits no processo da cifra de Vigenère-Vernam, e a operação passaria a ser a soma módulo 2, ou também, *Exclusive Or (XOR)* (Ou Exclusivo) sobre cada um dos bits da mensagem. Se a mensagem é definida $M = x_0, x_1...x_{n-1}, x_n$ e a chave $K = k_0,\ k_1...k_{n-1}, k_n$ então a cifra E pode ser definida como:

$$E(M, K) = x_i \oplus k_i. \tag{2.4}$$

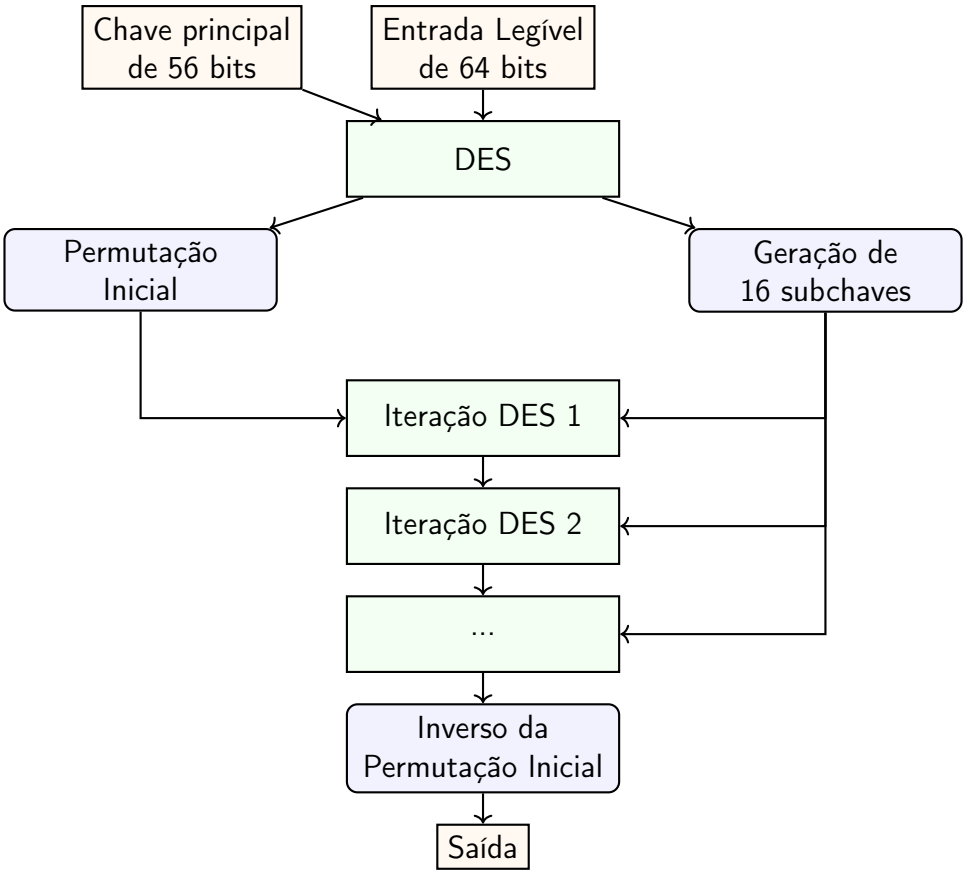
Ou seja, é feita uma operação com cada um dos dígitos da chave. Como transmitir uma chave K do mesmo tamanho do legível é uma tarefa computacionalmente custosa e, muitas vezes, ineficiente, se a chave K tem comprimento n, então é necessário recomeçar a chave sempre que ela chegar no final.

Dessa forma, a cifra torna-se substancialmente mais complexa para ser decifrada, principalmente se escolhermos uma chave K com pouca ou nenhuma relação estatística com M [8]. Estes exemplos já não são recomendados [12], mas ambos compartilham da técnica de permutação que é utilizada nos métodos seguros utilizados. Essencialmente, muitos modelos modernos utilizam-se da composição de permutações e operações como o $\oplus$ nas bitstrings para garantir o **sigilo** e a **integridade** da informação [8].

Por hipótese, toda cifra obtida via técnicas de criptografia simétrica deve ter seu texto original definido somente pela chave que gerou a cifra. Tendo em vista isso, os algoritmos

de criptografia utilizados pelos órgãos governamentais para fins críticos foram, historicamente, secretos [8]. Foi na década de 1970 que a *International Business Machines Corporation* (IBM) publicou junto do *National Bureau of Standards (NBS)* o *Data Encryption Standard (DES)*. O DES foi utilizado de forma ampla internacionalmente como o mais alto padrão de segurança e consiste, essencialmente, de diversas iterações de permutações da chave com o texto original como mostra a Figura 2.5 [8].

Figura 2.5: Esquema simplificado DES



Fonte: Autor

Com isso, a comunidade adotou uma oposição à "Segurança por Obscuridade"(*Security by obscurity*), ou seja, as cifras geradas pelos algoritmos devem ter seu sigilo garantido exclusivamente pela escolha da sua chave e não pelas peculiaridades do algoritmo.

No fim da década de 1990 foi definida uma competição para determinar o algoritmo para substituir o DES, o Advanced Encryption Standard (AES). O AES também consiste, em partes, de técnicas de substituição, permutação e composição. Claro que há sofisticações, por exemplo ele utiliza-se da manipulação de blocos matriciais para executar algumas etapas do seu processamento, por exemplo a *SubBytes* que faz transformações não-lineares nas matrizes/blocos para garantir segurança contra criptoanálise diferencial [8].

A Criptografia Simétrica é, ainda, amplamente utilizada em diversos âmbitos modernos

que necessitem de eficiência espacial (memória) e temporal (rápidos). Por exemplo o *Hyper Text Transfer Protocol Secure* (HTTPS) utiliza amplamente do AES, devido a sua grande eficiência, em parte por conta de hardwares modernos otimizados para efetuar tais operações.

2.2 Criptografia Assimétrica

A Criptografia assimétrica ou de chave pública consiste na forma de encriptar dados e/ou trocar/transportar chaves de criptografia secretas K entre dois ou mais pontos de comunicação através de um meio inseguro. Isso deve ser feito sem que haja riscos de que a chave de criptografia simétrica K seja comprometida/descoberta. Os algoritmos de criptografia assimétrica baseiam-se em ferramentas/problemas matemáticos, diferentemente dos algoritmos citados anteriormente (simétricos) que, em sua maioria, utilizam de manipulações relativamente simples, utilizando a vantagem de sistemas digitais em fazer tarefas simples de forma repetitiva para torná-los criptograficamente seguros. O *RSA (Rivest-Shamir-Adleman)*, é baseado no problema da fatoração de um número inteiro n em números primos, i.e.:

$$\text{Dado } n \in \mathbb{Z} \text{ encontrar os primos } p_1, p_2, \ldots, p_m \text{ que satisfaçam } n = \prod_{i=1}^{m} p_i$$

Para discutir sobre o algoritmo RSA precisamos, é necessário definir alguns aspectos de teoria dos números:

- Para os números $n_1, n_2 \in \mathbb{Z}$, o

$$\begin{aligned} &\text{maior divisor comum entre } n_1,\ n_2 \equiv mdc(n_1, n_2) = m \\ \iff &\begin{cases} m \in \mathbb{Z} \\ m|n_1 \text{ e } m|n_2. \\ \nexists\ (m_1 \in \mathbb{Z}\ : m_1|n_1 \text{ e } m_1|n_2) \text{ e } m_1 > m \end{cases} \end{aligned} \tag{2.5}$$

  Além disso, $\forall m_i\ m_i|n_1$ e $m_i|n_2|$ tem-se $m_i|m$ ou seja, $mdc(n_1, n_2)$ representa o maior número que divide n_1 e n_2, portanto, todos os outros divisores comuns de n_1 *e* n_2 também dividem $mdc(n_1, n_2)$.

- Um número primo p é qualquer número que possui como únicos divisores os elementos do conjunto $\{1, p\}$.

- Um número é primo p^* relativo a n quando satisfaz $mdc(n, p^*) = 1$. Ou seja, o maior divisor comum de n e p^* é 1.

A função $\phi(n) \equiv \#\{p_i^*\} : 0 \leq p_i^* < n \ \wedge \ mdc(p_i^*, n) = 1$ em que p_i^* é relativamente primo a n. Dessa forma, $\phi(n)$, em palavras, é a função que "conta"a quantidade de números menores que n que são relativamente primos a n". É trivial notar que

$$\phi(p) = (p - 1)\ \forall p \in Primos \tag{2.6}$$

Visto que, quando p é primo, todos os números inteiros menores que p, ou seja, $p-1$ são relativamente primos a p por definição de um número primo. Pode-se notar, ainda, que ϕ é multiplicativa, ou seja:

$$\begin{aligned} \phi(1) &= 1 \\ \phi(n \cdot m) &= \phi(n) \cdot \phi(m). \end{aligned} \tag{2.7}$$

O algoritmo de RSA necessita da existência de 2 chaves distintas: Uma chave pública K_p (pode ser trocada no meio inseguro) e uma chave privada K_s (Não pode ser revelada). Para o cálculo de chaves é o suficiente apenas encontrar um par de números **primos** p_1 e p_2 para obter o número composto $n = p_1 \cdot p_2$ e, com isso, calculamos o número K_s relativamente primo a $\phi(n)$. Dados K_s e n, calcula-se um terceiro número K_p que satisfaz

$$K_p \cdot K_s = 1 \mod \phi(n). \tag{2.8}$$

Com isso, obtém-se *chave secreta (K_s, n)* e a *chave pública (K_p, n)*. Podendo, por meio de uma rede de computadores ser disponibilizada publicamente.

Nota-se, a seguir, que não importa qual das duas chaves sejam usadas como pública ou como privada, desde que sejam usadas separadamente e, em momento algum, ambas sejam disponibilizadas publicamente.

Então, para criptografar uma mensagem qualquer m, é necessário efetuar uma operação:

$$E(m) = m^{K_p} \mod n, \tag{2.9}$$

onde $(K_p,\ n)$ é uma chave pública. Dado o teorema de Euler:

$$mdc(n, m) = 1 \Rightarrow m^{\phi(n)} = 1 \mod n. \tag{2.10}$$

Tem-se:

$$E(m)^{K_p} \mod n = (m^{K_s} \mod n)^{K_p} \mod n = m^{K_p \cdot K_s} \mod n, \tag{2.11}$$

mas, por construção $K_p \cdot K_s = 1 \mod \phi(n) = 1 + x \cdot \phi(n)$, então segue:

$$E(m)^{K_p} \mod n = m^{1+x\phi(n)} \mod n = m \cdot m^{x \cdot \phi(n)} \mod n, \tag{2.12}$$

e, finalmente, obtém-se

$$E(m)^{K_p} = \big([m \mod n] \cdot [m^{x\phi(n)} \mod n]\big) \mod n = m \mod n \tag{2.13}$$

Dessa forma, tem-se também a limitação $0 \leq m < n$.

Se ambas as partes possuírem um par de chaves Pública-Privada, quem faz o envio (por exemplo Alice), pode criptografar m com sua chave secreta $m^{K_p^a} = m_{autenticado}$ e, após isso, criptografar $m_{autenticado}$ com o mesmo processo descrito anteriormente com a chave pública do recebedor, (por exemplo Beto) como mostra a Figura 2.6.

Figura 2.6: Exemplo Cifra Assimétrica

Fonte: Autor

Nesse processo, Beto obteria após descriptografar o ilegível $m_{autenticado}$ que, como mencionado anteriormente, pode ser descriptografado com a chave pública de Alice e somente com a chave pública de Alice. Dessa forma, a mensagem estaria assinada por Alice, pois a chave pública de Alice seria única naquela rede e, portanto, somente a sua chave secreta seria capaz de assinar m daquela forma.

Observa-se que, se for possível fatorar n de forma eficiente, um par de chaves pode ser obtido $(K_s,\ K_p)$. Dessa forma, esse é um sistema com um método bem definido para quebrar *qualquer comunicação*, mas fatorar um número inteiro qualquer n *é computacionalmente inviável utilizando computadores* **clássicos** para n grande o suficiente. Então, deve-se sempre escolher um n consideravelmente grande e, além disso, durante a comunicação o par de chaves podem ser recalculados e substituídos de forma a garantir que, mesmo que uma parte da comunicação seja quebrada, isso não compromete toda a comunicação feita anteriormente.

O algoritmo RSA é amplamente utilizado, por exemplo, em assinaturas digitais. Esse algoritmo tem sua vulnerabilidade baseada no cálculo dos fatores de n, todavia esse algoritmo também pode ser quebrado se houver solução ao logaritmo discreto e acesso a um texto legível e sua cifra, já que $C = M^{K_p} \mod n$, portanto se o acesso ao par é conhecido$(M,\ C)$ e tem-se capacidade de resolver o problema do logaritmo discreto, pode ser obtido K_p.

Existem diversos outros algoritmos que baseiam-se diretamente no problema do logaritmo discreto, como o El-Gamal, o Diffie-Helman, dentre outros. Todos estes representam casos específicos do Problema do Sub-Grupo Escondido (HSP - *Hidden Subgroup Problem*) como garantia de segurança [5].

Capítulo 3

Mecânica Quântica

3.1 Conceitos básicos

Com os avanços da física durante os séculos XVIII e XIX, principalmente o eletromagnetismo, a mecânica analítica e a mecânica estatística, a física, para alguns, chegou a ser considerada estar na iminência de uma teoria final [13]. É o trabalho de doutorado de Max Planck no fenômeno do espectro de emissão de um corpo negro que dá início à uma revolução na física, catapultando uma abordagem completamente diferente aos problemas em escala atômica. O formalismo e o desenvolvimento de um tratamento mais completo são feitos, por outro lado, nos anos seguintes por Werner Heisenberg, Erwin Schrödinger e Paul Dirac.

A descrição clássica da física pode ser apresentada na forma de postulados, para facilitar seu desenvolvimento e fortalecer o rigor matemático, da seguinte forma [14]:

- O estado de um sistema físico em algum momento t_0 é dado pelas n coordenadas generalizadas $q(t_0)$ e seus n momentos generalizados.
- O valor de qualquer variável física (e, portanto, qualquer medida) é determinado pelo estado do sistema.
- A evolução do sistema ao longo do tempo é determinada pela equação de Hamilton-Jacobi e, se o estado do sistema é conhecido em um momento, qualquer estado futuro ou passado pode ser obtido matematicamente.

Da mesma forma, podemos determinar um conjunto de postulados que descrevem os sistemas quânticos (adaptado diretamente [14]):

- Em um dado momento t_0 o estado de um sistema fisicamente isolado é definido pelo vetor $|\varphi(t_0)\rangle$ parte do espaço de estados ξ.
- Toda quantidade física mensurável $\mathcal{A}$ é descrita por um operador A que age sobre ξ. Este operador é um observável.

- O único resultado possível de uma medição de uma quantidade associada ao observável $\mathcal{A}$ é um dos autovalores de A.

Todo o estado de um sistema quântico deve ser representado por um vetor contido em um espaço vetorial complexo, munido de um produto escalar que define de forma particular seu produto interno [13]. Os vetores nesse espaço são representados na notação de Dirac como $|\psi\rangle$ e são chamados de *kets*. Enquanto que o vetor dual correspondente à $|\psi\rangle$ é chamado de *bra* e escrito $\langle\psi|$. O produto interno entre um vetor $|\psi\rangle$ e o vetor $|\phi\rangle$ é representado por $\langle\phi|\psi\rangle$ em que $\langle\phi|$ é o vetor dual correspondente à $|\phi\rangle$. A norma do vetor $|\psi\rangle$ é dada por $(\langle\psi|\psi\rangle)^{\frac{1}{2}}$. Essas condições são algumas das definições são algumas partes da definição de um *Espaço de Hilbert*. Todo estado quântico é definido dentro de um espaço de Hilbert. A notação utilizando Bras e Kets é chamada de notação de Dirac.

- Uma quantidade física associada ao observável A, quando mensurada em um sistema em um estado normalizado $|\varphi\rangle$, a probabilidade $P(a_n)$ de obter um autovalor (não degenerado) a_n do observável A é

$$P(a_n) = |\langle u_n|\varphi\rangle|^2 \tag{3.1}$$

  onde $\langle u_n| \equiv$ autovetor dual associado com A.

- Caso Discreto: Uma quantidade física associada ao observável A, quando mensurada em um sistema em um estado normalizado $|\varphi\rangle$, a probabilidade $P(a_n)$ de obter um autovalor (degenerado) a_n do observável A é

$$P(a_n) = \sum_{j=1}^{g_n} |\langle u_n^j|\varphi\rangle|^2 \tag{3.2}$$

  onde g_n é o grau de degenerescência de a_n e $|u_n^j\rangle$ $(j \in \mathbb{Z}^+)$ é uma base no subespaço associado ao auto-valor a_n de A.

- Caso Contínuo: Quando uma quantidade A é medida em um sistema no estado normalizado $|\varphi\rangle$, a probabilidade $dP(\alpha)$ de encontrar um resultado r tal que $\alpha < r < \alpha + d\alpha$ é

$$dP(\alpha) = |\langle v_\alpha|\varphi\rangle|^2 d\alpha \tag{3.3}$$

  onde $|v_\alpha\rangle$ é o autovetor correspondente ao autovalor α do observável.

- Dado o vetor $|\psi_n\rangle$ definido como a projeção de $|\psi\rangle$ no subespaço associado ao autovalor a_n, então definimos o operador projeção como $P_n|\psi\rangle = |\psi_n\rangle$. Se a medida física $\mathcal{A}$ de um sistema no estado $|\psi\rangle$ tem valor igual a a_n, então o estado do sistema imediatamente após a medição é a projeção normalizada

$$\frac{P_n|\psi\rangle}{\sqrt{\langle\psi|P_n|\psi\rangle}} \tag{3.4}$$

  de $|\psi\rangle$ no subespaço de autovetores associados com a_n.

- A evolução temporal de $|\psi(t)\rangle$ é dada pela equação de Schrödinger

$$i\hbar\frac{d}{dt}|\psi(t)\rangle = H(t)\,|\psi(t)\rangle \ , \tag{3.5}$$

onde H é o operador Hamiltoniano, i é a unidade imaginária ($i : \ i^2 = -1$) e a constante de Planck reduzida $\hbar = \frac{h}{2\pi} \approx 1,054\ 10^{-34}\ J \cdot s$.

Estes postulados são suficientes para construir uma *Mecânica* (ou seja, uma descrição física completa) completamente distinta da clássica, todavia, como foi validado diversas vezes por experimentos, essa teoria é capaz de descrever com grande precisão fenômenos que ocorrem em pequenas escalas. Apesar de fenômenos quânticos ocorrerem também em escala macroscópica, como o tunelamento demonstrado pelo trabalho laureado com o Prêmio Nobel de Física de 2025 [15], esses fenômenos só ocorrem em situações extremas, como temperaturas extremamente baixas, materiais altamente puros, entre outras configurações experimentais sensíveis. De maneira análoga, espera-se que a mecânica quântica tende a exibir comportamentos não previstos pela mecânica clássica em escala da ordem de poucos átomos e/ou, como comentado, em situações extremas. Portanto, para utilizar esses fenômenos como os principais responsáveis na produção, transmissão e armazenamento de informação, deve-se ter em mente que é preciso inovar na maneira de produzir tanto hardware quanto software para tornar essa ferramenta viável.

Uma das consequências destes postulados é de que *um sistema físico pode existir em um estado correspondente à superposição de outros estados* (linearidade do espaço vetorial). Podemos imaginar que um sistema físico e um parâmetro limite de diferença de potencial elétrico (por exemplo) forma um sistema com 2 estados possíveis:

- Estado alto (1): diferença de potencial maior que um limite pré-determinado.
- Estado baixo (0): diferença de potencial menor que um limite pré-determinado.

Essa é uma forma válida de representar a menor unidade lógica da computação: o *bit*. De forma bastante intuitiva, o bit $b \in \{0, 1\}$ é a abstração de um sistema físico com uma propriedade que pode assumir dois valores facilmente distinguíveis experimentalmente. De forma análoga, em um computador quântico, precisa-se de dois estados característicos $|0\rangle \ e \ |1\rangle$ que definem um *qubit* e são a menor unidade lógica. Eles são abstrações de qualquer sistema físico descrito pela mecânica quântica que possuem dois valores característicos, como o spin de um elétron, as energias do estado fundamental e do primeiro estado excitado de um átomo, a direção de polarização de um fóton, etc [5].

A diferença em relação ao bit clássico é que um sistema quântico pode assumir uma superposição destes dois estados, $|s\rangle = \alpha_0\,|0\rangle + \alpha_1\,|1\rangle$, tal que $|\alpha_i|^2$ representa a probabilidade da medição do sistema no estado $|s\rangle$ resultar em $|s_i\rangle$. Esta propriedade é a superposição de estados e é chave no processo de desenvolvimento de algoritmos aplicados em computação quântica.

Naturalmente, não deve ser limitado ao tratamento de um único qubit, mas sim de um conjunto de diversos qubits em cadeia e, por isso, intimamente relacionados. Um registrador que contenha, por exemplo, 3-qubits será representado de forma geral como

$$|\psi\rangle = a_0 |000\rangle + a_1 |001\rangle + a_2 |010\rangle + a_3 |011\rangle + a_4 |100\rangle + a_5 |101\rangle + a_6 |110\rangle + a_7 |111\rangle \tag{3.6}$$

Em geral, um registrador que contenha n-qubits terá um estado representado pela superposição linear de todas as permutações de seus 2^n possíveis qubits.

Esse conjunto de n qubits, juntamente com o penúltimo postulado garantem que podemos manipular as probabilidades, por exemplo, fazendo medidas parciais de um bit em uma posição antes de fazer a medição do sistema como um todo. Por exemplo, o qubit for medido na primeira posição e obtivermos o valor 1, nós colapsamos a função de onda parcialmente e, por isso, os auto-estados $|000\rangle, |001\rangle, |010\rangle, |011\rangle$ passam a ter probabilidade igual a zero de serem medidos desde que a leitura do restante do sistema seja feita pouco tempo após essa leitura parcial.

3.1.1 Interferência

Os observáveis são representados como operadores Hermitianos. Suponha um computador clássico com registradores probabilísticos, ou seja, em que o resultado da medição já está pré-determinado e a medição que revela este valor. Por outro lado, considere um computador quântico que suporta um registrador em estado de superposição.

Agora, então, suponha um observável A que age no registrador e, ainda, **a** é um dos autovalores de A com o autovetor $|\phi_a\rangle$. Então, por definição, $A|\phi_a\rangle = a|\phi_a\rangle$. Então, faz-se necessário a seguinte pergunta: *Qual a probabilidade de obtermos o autovalor 'a' no momento que medir o registrador em uma superposição* $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$*?*

Em um contexto clássico, o registrador só poderia estar, efetivamente, em um único estado: ou $|0\rangle$ ou $|1\rangle$ carregados com probabilidade $|\alpha|^2$ e $|\beta|^2$ respectivamente. Não é conhecido qual o estado de antemão. Nesse cenário, é simples para obter a probabilidade, basta apenas tomar a soma das projeções/produtos internos dos estados possíveis do registrador com o autoestado $|\phi_a\rangle$, ou seja,

$$P^*(a \ : \ |\psi\rangle = \alpha|0\rangle + \beta|1\rangle) = |\alpha|^2 |\langle\phi_a|0\rangle|^2 + |\beta|^2 |\langle\phi_a|1\rangle|^2 \tag{3.7}$$

Ou seja, o valor final é dado pela soma da probabilidade de medir a tal que o sistema está no estado $|0\rangle$ vezes a probabilidade do sistema estar no estado $|0\rangle$ (α) ***ou*** a probabilidade de medir a tal que o sistema está no estado $|1\rangle$ vezes a probabilidade do sistema estar no estado $|1\rangle$ (β). O detalhe fundamental dessa abordagem é o termo ou que torna explícito que o sistema em superposição não existe nessa abordagem, o que existe é somente a incerteza clássica do seu estado verdadeiro. Não há interação entre os estados fundamentais $|0\rangle$, $|1\rangle$ [16].

Por outro lado, no caso quântico, o sistema genuinamente se encontra no estado de superposição $\alpha|0\rangle + \beta|1\rangle$. Dessa forma, a parte da projeção $\langle\phi_a|0\rangle$ *interfere* com a projeção sobre o estado $\langle\phi_a|1\rangle$ dando origem a um termo extra na probabilidade real (quântica):

$$
\begin{aligned}
&P(a \mid |\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle) = \mid \langle\phi_a|\psi\rangle \mid^2 \\
&= \mid \langle\phi_a| \left(\alpha\,|0\rangle + \beta\,|1\rangle\right) \mid^2 \\
&= \mid \alpha\,\langle\phi_a|0\rangle + \beta\,\langle\phi_a|1\rangle \mid^2 \\
&= \left(\alpha\,\langle\phi_a|0\rangle + \beta\,\langle\phi_a|1\rangle\right)^* \left(\alpha\,\langle\phi_a|0\rangle + \beta\,\langle\phi_a|1\rangle\right)
\end{aligned}
$$

para encontrar esse resultado pode-se utilizar a propriedade módulo ao quadrado dos números imaginários quaisquer $l = (a + ib)$ e $k = (c + id)$:

$$
\begin{aligned}
\mid (a+ib) + (c+id) \mid^2 &= \big((a+ib) + (c+id)\big)^* \big((a+ib) + (c+id)\big) \\
&= ((a-ib) + (c-id))((a+ib) + (c+id)) \\
&= (a^2+b^2) + (c^2+d^2) + (ac + i\cdot ad) + (bd - i\cdot bc) \\
&+ (ca + i\cdot bc) + (bd - i\cdot ad) \\
&= (a^2+b^2) + (c^2+d^2) + 2(ac+db) \\
&= \mid l \mid^2 + \mid k \mid^2 + 2\cdot Re(l\cdot k^*)
\end{aligned}
\tag{3.8}
$$

portanto

$$
\begin{aligned}
P(a \mid |\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle) = &\mid \alpha \mid^2 \mid \langle\phi_a|0\rangle \mid^2 + \mid \beta \mid^2 \mid \langle\phi_a|1\rangle \mid^2 \\
&+ 2\cdot Re(\alpha\,\langle\phi_a|0\rangle \cdot \langle\phi_a|1\rangle^*\,\beta^*)
\end{aligned}
\tag{3.9}
$$

Ou seja, como esperado, obteve-se o termo 'cruzado' $2\cdot Re(\alpha\,\langle\phi_a|0\rangle \cdot \langle\phi_a|1\rangle^*\,\beta^*)$ relacionado justamente com a interferência dos dois estados quando em superposição. Esse fenômeno também diferencia fundamentalmente a computação clássica da computação quântica e representa um fator importante no desenvolvimento de algoritmos quânticos [16].

Capítulo 4

Computação Quântica

4.1 Operadores Lógicos na Computação Quântica

Primeiramente, é importante notar que, quando o sistema está isolado e submetido somente a potenciais independentes do tempo $V(\mathbf{r}, t_1) = V(\mathbf{r}, t_2)\ \forall\ t_1, t_2$, a equação de Schrödinger tem soluções separáveis e, ainda, seu operador de evolução temporal $\hat{U}$ (operador que descreve a translação no tempo de um estado $|\psi\rangle$) é um operador unitário [17] [14]. Ou seja, trata-se de um operador que satisfaz $U : U^{-1} = U^{\dagger}$. Como, por definição, se o operador U existe, segue $U^{\dagger}$ também existe, então o inverso do operador U também existe. Consequentemente, toda operação quântica implementada nos algoritmos quânticos deve, também ser reversível, já que essa terá translação no tempo. Quando tratamos da computação clássica, é de praxe o uso de diferentes operadores lógicos como parte da grande cadeia de operações (envolvendo *bitshifts*, escritas e leituras nos registradores do computador) concatenados formando grandes cadeias de operações para formar um algoritmo mais complexo. Esses operadores, de forma geral, são bastante simples e podem ser representados e explicados em poucas linhas. Por exemplo: o operador de negação ($\neg$) pode ser representado como:

$$f : \{0,1\} \to \{0,1\}\ :\ f(b) = b + 1 \mod 2 \tag{4.1}$$

Ou seja, ele transforma um **estado de um registrador em outro estado**. Um operador lógico em um computador quântico faria o mesmo, transformaria um estado do registrador em outro. A diferença é que *o estado da memória/sistema e o estado que é medido/lido são coisas distintas em sistemas quânticos, enquanto que em sistemas clássicos não* [16].

Se, em um computador clássico, a memória está num estado 0100 e é feita a medição, o valor 0100 dessa medida é obtido. Além disso, em um dado momento, um registrador clássico só pode conter um valor e, portanto, qualquer operação lógica sobre este estado também vai resultar em outro estado definido e mensurável. Por outro lado, em um computador quântico, se sua memória está em um estado $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$, *não pode-se* ***medir*** *este estado (de superposição)*; se uma medição é efetuada, vamos obtém-se ou $|1\rangle$

ou $|0\rangle$ e, tão importante quanto, caso alguma operação for executada no registrador em estado $|\psi\rangle$, o computador estará operando justamente sobre as probabilidades de obter-se uma medição específica, não sobre os valores em si.

Dessa forma, pode-se iniciar os operadores com uma notação matricial e analisar o que seria o NOT clássico: (uma das) matrizes de spin de Pauli.

$$\sigma_1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{4.2}$$

Que é capaz de negar os elementos da base computacional $\{|0\rangle, |1\rangle\}$:

$$\sigma_1 |0\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle \tag{4.3}$$

E realmente é tentador assumir que a matriz de Pauli σ_1 deve também ser definida como o operador lógico NOT quântico. Todavia, é necessário lembrar que, quando o estado é representado como $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$, $\alpha, \beta \in \mathbb{C}$, ou seja, opera-se σ_1 em um estado $|\psi\rangle$ qualquer, obtém-se:

$$\begin{aligned} \sigma_1 |\psi\rangle = \sigma_1 \cdot (\alpha |0\rangle + \beta |1\rangle) = \\ = \sigma_1 \cdot [(a + bi) |0\rangle + (c + di) |1\rangle] = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \\ = \begin{pmatrix} \beta \\ \alpha \end{pmatrix} = (c + di) |0\rangle + (a + bi) |1\rangle \end{aligned} \tag{4.4}$$

Esse resultado mostra que a matriz σ_1 não parece ser a melhor definição de um NOT quântico. Com esta contextualização, pode-se introduzir um operador bastante útil na computação quântica: o operador de Hadamard.

4.1.1 Operador de Hadamard

O operador de Hadamard é um operador útil pois ele transforma um bit (ou seja, como o NOT clássico, esse operador transforma um bit por operação) em uma superposição da sua base de estados. Ele é representado matricialmente pela forma:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{4.5}$$

Aplicando à base computacional $(|0\rangle, |1\rangle)$ o operador H transforma:

$$\begin{aligned} H|1\rangle &= \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\cdot\begin{pmatrix}0\\1\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\cdot 0+1\cdot 1\\ 1\cdot 0-1\cdot 1\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix} \\ &= \frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\end{pmatrix} - \frac{1}{\sqrt{2}}\begin{pmatrix}0\\1\end{pmatrix} = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{aligned} \tag{4.6}$$

Ou seja, o operador H transformou o bit que estava em um estado determinado $|1\rangle$ em uma superposição dos estados $(|0\rangle, |1\rangle)$ igualmente prováveis. Essa operação é bastante útil pois, por exemplo, em um registrador contendo 3 qubits: $|abc\rangle$ (abc representa a *bitstring* formada pelo seu registrador), pode-se colapsar todos os qubits para zero e, então, aplicar o operador de Hadamard em paralelo em todos os qubits para obter uma superposição de todas as permutações possíveis desse registrador: basta notar que

$$\begin{aligned} |000\rangle &= |0\rangle \ \otimes |0\rangle \otimes |0\rangle \\ &= \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} \\ &= \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}1\cdot\begin{pmatrix}1\\0\end{pmatrix}\\ 0\cdot\begin{pmatrix}1\\0\end{pmatrix}\end{pmatrix} \\ &= \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}1\\0\\0\\0\end{pmatrix} \\ &= \begin{pmatrix}1\cdot\begin{pmatrix}1\\0\end{pmatrix}\\ 0\cdot\begin{pmatrix}1\\0\end{pmatrix}\\ 0\cdot\begin{pmatrix}1\\0\end{pmatrix}\\ 0\cdot\begin{pmatrix}1\\0\end{pmatrix}\end{pmatrix} = \begin{pmatrix}1\\0\\0\\0\\0\\0\\0\\0\end{pmatrix} \end{aligned} \tag{4.7}$$

Então

$$H|0\rangle \otimes H|0\rangle \otimes H|0\rangle = (\frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}) \otimes (\frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix} \otimes (\frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}))$$

$$\vdots$$

$$= \frac{1}{2\sqrt{2}}\left[\begin{pmatrix}1\\1\\1\\1\\1\\1\\1\\1\end{pmatrix} = \begin{pmatrix}1\\0\\0\\0\\0\\0\\0\\0\end{pmatrix} + \begin{pmatrix}0\\1\\0\\0\\0\\0\\0\\0\end{pmatrix} + \begin{pmatrix}0\\0\\1\\0\\0\\0\\0\\0\end{pmatrix} + \ldots + \begin{pmatrix}0\\0\\0\\0\\0\\0\\0\\1\end{pmatrix}\right] \tag{4.8}$$

$$= \frac{1}{2\sqrt{2}}\Big[|000\rangle + |001\rangle + |010\rangle$$

$$+ |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle\Big]$$

Ou seja, consegue-se aplicar em paralelo uma única operação e obtivemos um registrador em um estado com igual probabilidade para todas as 2^3 permutações do registrador.

4.2 Algoritmo de Deutsch-Jozsa

O problema de Deutsch pode ser resumido como: Dada uma função $f : D \to \{0,1\} \mid D = \{0,1\}^n$, ou seja que recebe uma bitstring de n-bits e retorna um único bit. Essa função, obrigatoriamente, apresenta um dos seguintes comportamentos:

- Ou f é constante:
$$f := (f(b) = 0 \vee f(b) = 1) \ \forall \ b \tag{4.9}$$
- Ou f é balanceada: $f(x) = 0$ para metade dos valores possíveis de x e $f(x') = 1$ para a outra metade. Ou, ainda,
$$f := f(b) = 0 \ \forall \ b \in K \subsetneq D \wedge f(b') = 1 \ \forall \ b' \in K^c \subsetneq D \ : \ \#K = \frac{\#D}{2} \tag{4.10}$$

O problema, então, consiste em determinar qual dos comportamentos a função apresenta utilizando o menor número possível de chamadas à função.

Em um contexto clássico, só consegue-se ter certeza de qual o comportamento dessa função com, ao menos, $2^{n-1}+1$ chamadas de f. Por outro lado, para abordar esse problema utilizando a computação quântica, precisa-se primeiro estabelecer essa função como uma operação reversível, como é pré-requisito das transformações quânticas. Pode-se, então, dispensar a transformação $|x\rangle \to |f(x)\rangle$, já que, se f for constante, todas as suas saídas

serão iguais e, portanto, a relação anterior não permite determinar a entrada. Por outro lado, é possível inserir um qubit inicializado em zero na entrada e escrever a saída da função sobre esse qubit e preservar o bit de entrada: $(|x\rangle, |0\rangle) \rightarrow (|x\rangle, |f(x)\rangle)$. Embora essa abordagem seja promissora o pré-requisito de reversibilidade exige que todos os inputs *possíveis* sejam mapeados para uma única saída e, na última representação, inserimos 2 qubits e só é considerado o caso em que o segundo qubit é zero. Se considerar o caso de f constante, que $(|x\rangle, |0\rangle)$ é mapeado para o mesmo resultado que o input $(|x\rangle, |1\rangle)$ $[(|x\rangle, |f(x)\rangle)]$. Dessa forma, o segundo método também não é reversível. Para manter a estrutura e obter uma transformação reversível, utiliza-se o ou-exclusivo:

$$XOR(\oplus) : \{0,1\}^2 \rightarrow \{0,1\} \mid a \oplus b = 1 \iff (a = 1 \wedge b = 0) \vee (a = 0 \wedge b = 1) \quad (4.11)$$

A operação $U_f : (|x\rangle, |y\rangle) \rightarrow (|x\rangle, y \oplus |f(x)\rangle)$ que é uma transormação reversível e que preserva toda a informação de f. Por exemplo, tem-se (para o caso de um único qubit) os seguintes valores nas Tabela 4.1 e Tabela 4.2 abaixo:

Tabela 4.1: Valores de entrada e saída possíveis de f (constante $f(x) = 1$)

x	y	f	$f \oplus y$
0	0	1	1
1	0	1	1
0	1	1	0
1	1	1	0

Tabela 4.2: Valores de entrada e saída possíveis de f (constante $f(x) = 0$)

x	y	f	$f \oplus y$
0	0	0	0
1	0	0	0
0	1	0	1
1	1	0	1

Para estados preparados nos estados na base computacional $|0\rangle$, $|1\rangle$, toda a computação será idêntica à clássica (com um grau maior de complicação e cuidados/formalismos). Por outro lado, se, por exemplo, para o caso de um único qubit, for inserido na operação a entrada $[\frac{|0\rangle+|1\rangle}{\sqrt{2}}, |0\rangle]$, obtém-se (assumindo a linearidade de U_f sobre o estado de superposição):

$$\begin{aligned} \frac{|0\rangle + |1\rangle}{\sqrt{2}}, |0\rangle \xrightarrow{U_f} \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \left(\frac{|f(0)\rangle}{\sqrt{2}} \oplus |0\rangle\right) + \left(\frac{|f(1)\rangle}{\sqrt{2}} \oplus |0\rangle\right) \\ \rightarrow \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \frac{|f(0)\rangle + |f(1)\rangle)}{\sqrt{2}}, \end{aligned} \quad (4.12)$$

Ou seja, o estado final é uma superposição do valor de $f(|0\rangle)$ e $|f(1)\rangle$; então, de certa forma, com uma única operação tanto o valor de $f(|0\rangle)$ quanto $f(|1\rangle)$ são utilizados simultaneamente. Essa característica é o *paralelismo* quântico, que, se usado de forma engenhosa, pode diminuir drasticamente o tempo computacional para processar a informação. Nesse caso, esta propriedade não é o suficiente para resolver o problema, pois não é conhecido $|f(0)\rangle$ e nem $|f(1)\rangle$. Todavia, ainda no mesmo exemplo de um único qubit, pode ser desenvolvido uma forma mais eficiente de resolver o problema do que avaliando $f(0)$ e $f(1)$.

Primeiro, considera-se a transformação U_f para o input $|x\rangle\,,\ \frac{|0\rangle-|1\rangle}{\sqrt{2}}$ que pode ser obtido aplicando o operador de Hadamard no segundo qubit do estado preparado $(|x\rangle\,,\ |1\rangle)$, ou seja, $(|x\rangle\,,\ H\,|1\rangle)$ nesse caso, é obtido:

$$\begin{aligned}
|x\rangle\,,\ \frac{|0\rangle-|1\rangle}{\sqrt{2}} &\xrightarrow{U_f} |x\rangle\,,\ \frac{|0\rangle-|1\rangle}{\sqrt{2}} \oplus |f(x)\rangle \\
&\to |x\rangle\,,\ \frac{1}{\sqrt{2}}[(|0\rangle \oplus |f(x)\rangle) - (|1\rangle \oplus |f(x)\rangle)] \\
&\to |x\rangle\,,\ \frac{1}{\sqrt{2}}(|f(x)\rangle - |f(x)+1 \mod 2\rangle) \\
&\to \begin{cases} f(x)=0 & |x\rangle\,,\ \frac{1}{\sqrt{2}}(|0\rangle-|1\rangle) \\ f(x)=1 & |x\rangle\,,\ \frac{1}{\sqrt{2}}(|1\rangle-|0\rangle) \end{cases} \\
\therefore |x\rangle\,,\ \frac{|0\rangle-|1\rangle}{\sqrt{2}} &\xrightarrow{U_f} |x\rangle\,,\ \frac{(-1)^{f(x)}}{\sqrt{2}}(|0\rangle-|1\rangle) \\
&= (-1)^{f(x)}\,|x\rangle\,,\ \frac{1}{\sqrt{2}}(|0\rangle-|1\rangle)
\end{aligned} \tag{4.13}$$

Então, para explorar as possibilidades, pode-se inserir como input $H\,|0\rangle\,,\ H\,|1\rangle$. Com isso obtém-se:

$$\begin{aligned}
\frac{|0\rangle+|1\rangle}{\sqrt{2}},\ \frac{|0\rangle-|1\rangle}{\sqrt{2}} &\xrightarrow{U_f} \left[(-1)^{f(0)}\frac{1}{\sqrt{2}}\,|0\rangle\,,\ \left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)\right] \\
&+ \left[(-1)^{f(1)}\frac{1}{\sqrt{2}}\,|1\rangle\,,\ \left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)\right] \\
&= \left((-1)^{f(0)}\frac{1}{\sqrt{2}}\,|0\rangle + (-1)^{f(1)}\frac{1}{\sqrt{2}}\,|1\rangle\right), \frac{|0\rangle-|1\rangle}{\sqrt{2}}
\end{aligned} \tag{4.14}$$

Por fim, é aplicado o operador de Hadamard no primeiro bit e o resultado:

$$\left((-1)^{f(0)}\frac{1}{\sqrt{2}}\,|0\rangle + (-1)^{f(1)}\frac{1}{\sqrt{2}}\,|1\rangle\right), \frac{|0\rangle-|1\rangle}{\sqrt{2}} \xrightarrow{H} \begin{cases} f(0)=f(1) & \pm\,|0\rangle\,,\ \frac{|0\rangle-|1\rangle}{\sqrt{2}} \\ f(0)\neq f(1) & \pm\,|1\rangle\,,\ \frac{|0\rangle-|1\rangle}{\sqrt{2}} \end{cases} \tag{4.15}$$

Ou seja, com uma única medida do qubit da esquerda, foi capaz de determinar se a função f é balanceada ($f(0) \neq f(1)$) ou constante ($f(0) = f(1)$). De forma similar, pode-se estender essas ideias para uma função que recebe uma bitstring de tamanho arbitrário N com algumas adaptações para garantir o ganho de operações (vide [16]). Ou seja, com adaptações relativamente simples, é diminuído para uma única chamada da função para obter um comportamento global dessa mesma função utilizando as ferramentas da computação Quântica.

Esse algoritmo, apesar da pouca utilidade real [16], representa uma tendência e uma das grandes motivações para o estudo da computação quântica: a redução em ordens de grandeza da complexidade temporal para resolução de um problema conhecido.

Um dos exemplos mais proeminentes do uso da computação Quântica para resolução de problemas de forma mais eficiente é o algoritmo de Shor apresentado no capítulo a seguir.

4.3 Algoritmo de Shor

O algoritmo de Shor é outro exemplo de algoritmo baseado na transformada de Fourier Quântica[16]. Diferente do algoritmo de Deutsch-Jozsa (para um bit), este algoritmo requer mais etapas na sua execução e cada uma delas desempenha um papel mais sutil na resolução do problema que o algoritmo resolve: *dado um número $N \in \mathbb{Z}$, obter o conjunto $L = \{n_i\} \ : \ n_i \in \mathbb{Z}$ em que n_i é um número primo e $N = \prod_{i=0}^{\#L} n_i$.* A partir dessa definição, constata-se que o algoritmo de Shor tem o potencial de tornar a criptografia RSA obsoleta, que se baseia justamente na dificuldade dessa fatoração.

Para motivar o desenvolvimento do algoritmo de Shor, note que a função responsável por garantir a segurança do algoritmo (exponenciação modular) é uma função periódica. Uma ferramenta versátil para a análise de funções periódicas é a *Transformada de Fourier (TF)*. De maneira mais específica, a transformada de Fourier Discreta (TFD):

$$F : X \subset \mathbb{C}^N \to Y \subset \mathbb{C}^N : \forall\ y_k \in Y; x_j \in X : y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{\frac{2\pi i j k}{N}} \tag{4.16}$$

É natural, então, tomar a TF Quântica (TFQ) como a transformada que, dada a base $(|0\rangle, \ldots, |N-1\rangle)$ e o estado $|j\rangle$:

$$QFT : |j\rangle \to \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\frac{2\pi i j k}{N}} |k\rangle \tag{4.17}$$

Essa transformação é unitária e, portanto, pode ser implementada no contexto da computação quântica. Além disso, por ser unitária, tem-se que $QFT^{-1} = QFT^{\dagger}$.

O algoritmo de Shor, na verdade, não fatora diretamente o valor n de sua entrada. Em vez disso, o algoritmo é eficiente em encontrar a ordem r, ou seja, encontrar o menor $r \in \mathbb{Z}^+$ tal que $x^r \equiv 1 \mod n$, de um dado $x : mdc(x, n) = 1$. Isso permite encontrar o divisor não trivial de n $mdc(x^{r/2} - 1, n)$ desde que $r \equiv 0 \mod 2$ (seja par) e $x^{r/2} \not\equiv -1 \mod n$, já que nesse caso $(x^{r/2} - 1)(x^{r/2} + 1) = x^r - 1$ e, por definição $x^r \equiv 1 \mod n$, então:

$$(x^{r/2} - 1)(x^{r/2} + 1) \equiv 0 \mod n \tag{4.18}$$

Ou seja, encontrar r permite encontrar fatores que dividem n [18].

Por exemplo, para a fatorar o numero 21 podemos:

- Selecionar o número aleatório tal que $mdc(21, x) = 1$. Para esse exemplo $x = 11$
- Encontrar a ordem de 11 módulo 20, ou seja $r = 6$, já que $11^6 = 66 = 1771561 = 84360 \cdot 21 + 1 \equiv 1 \mod 21$. Essa é a etapa mais fundamental do processo e é justamente onde o algoritmo de Shor utiliza as ferramentas da computação quântica para obter vantagem sobre a computação clássica.
- Com $r = 6$, utiliza-se a relação $x^r - 1 = (x^{r/2} + 1)(x^{r/2} - 1)$ e obtem-se

$$(11^3 - 1)(11^3 + 1) \equiv 0 \mod 21 \tag{4.19}$$

- Calcula-se $mdc(11^3 - 1, 21) = mdc(1331, 21) = 7$. Então é encontrado um fator não trivial de 21.

O seguinte algoritmo em python implementa uma fatoração de número inteiro baseado nesse método

```
import math
from os import urandom as random


def find_order(val, n):
    r = 1

    if math.gcd(val, n) != 1:
        raise Exception("Ordem indefinida.")

    while (val**r % n) != 1:
        r += 1

    return r


def is_prime(n):
    if n == 2:
```

```
        return True

    test_val = 2

    while test_val <= int(math.sqrt(n)):
        if n % test_val == 0:
            return False
        test_val += 1

    return True


def find_factor(n: int):
    rand = int.from_bytes(random(n.bit_length() + 1)) % n

    if is_prime(n):
        return n

    if rand == 0:
        return find_factor(n)

    if math.gcd(rand, n) != 1:
        return find_factor(n)

    order = find_order(rand, n)

    if order % 2 == 1:
        return find_factor(n)

    result = math.gcd((rand ** int(order / 2)) - 1, n)
    return result


val = input("Valor a ser fatorado: ")

if not val:
    print("Valor invalido", val)
    exit(1)

val = int(val, base=10)

trial = 1

factors = {}

while trial < 10000:
```

```
    factor = find_factor(val)

    if factors.get(factor) is None:
        factors[factor] = 1
    else:
        factors[factor] += 1

    trial += 1

print(factors)
```

Foi obtido distribuições para a fatoração do número 21 (10.000 amostras), por exemplo, como

```
    {1: 3280, 3: 3414, 7: 3305}
```

Ou seja, da ordem de 2/3 dos fatores que foram encontrados foi um fator não trivial ($\neq 1$). Claro que esse método é apenas um exemplo, já que é extremamente ineficiente, principalmente na função *find_ order* e na função *is_ prime*. Todavia, mostra que a técnica é, de fato, capaz de fatorar um número inteiro qualquer.

Outro resultado importante é:

$$mdc(x, N) = 1;\ N > 0;\ x^i \equiv x^j \mod N \Rightarrow i \mod r \equiv j \mod r \tag{4.20}$$

Onde r é a ordem de x módulo N [19]. Esse resultado mostra que, caso tenha diversas potencias $(i_1,\ i_2,\ i_3, \dots)$ e $x^{i_n} \mod N \equiv x^{i_m} \mod N$, então $i_n - i_m = Lr$ para algum $L \in \mathbb{Z}$ e r é a ordem de x módulo N.

Com isso, pode-se formular o algoritmo de Shor para fatoração de um número inteiro N:

1. Gera-se um inteiro $q = 2^{t_0} : N^2 \leq q \leq 2N^2$
2. Gera-se um inteiro $x : mdc(x, N) = 1$
3. Executa-se a Subrotina 1 m vezes e analisa os retornos na forma $(\frac{2^{t_0} k_1}{r}, \dots, \frac{2^{t_0} k_m}{r})$ em conjunto para extrair r
4. Calcula-se o $mdc(x^{r/2} - 1)$ que, provavelmente, será um divisor não trivial de N

Subrotina 1:

1. Inicializa o registrador R_0 e R_1 no estado $|000...00\rangle$, onde R_0 tem t_0 qubits e R_1 tem t_1 qubits, obtendo o estado inicial $|\psi_0\rangle = |0\rangle_{R_0}, |0\rangle_{R_1}$

2. Aplica-se o operador de Hadamard no registrador R_0, obtendo:

$$|\psi_1\rangle = H\,|0\rangle_{R_0}\,,|0\rangle_{R_1} = \frac{1}{\sqrt{2^{t_0}}}\sum_{j=0}^{2^{t_0}-1}|j\rangle_{R_0}\,,|0\rangle_{R_1}$$

3. Aplica-sea transformação $U_x : |j\rangle_{R_0}\,,|0\rangle_{R_1} \to |j\rangle_{R_0}\,,|x^j \mod N\rangle_{R_1}$. Note que, neste ponto do algoritmo, é inserido um fator periódico $\mod N$. Então obtem-se:

$$|\psi_2\rangle = U_x\psi_1 = \frac{1}{\sqrt{2^{t_0}}}\sum_{j=0}^{2^{t_0}-1}|j\rangle_{R_0}\,,\left|x^j \mod N\right\rangle_{R_1}$$

4. Então, estrategicamente observa-se o registrador R_1, de forma que, efetivamente, esta medindo um certo b_0 tal que $x^{b_0} \mod N \equiv x^j \mod N$. Mais importante, o estado do registrador R_0 irá colapsar para uma superposição de j's tal que $x^{b_0} \mod N \equiv x^j \mod N$. Mas, como mostra o resultado 4.20, esses j's estão separados pelo valor r, ou seja, a ordem de x módulo N. Portanto, a nova 'frequência' da sequência de valores em R_0 é proporcional à ordem de x e, **caso r seja uma potência de** 2, pode-se escrever o estado dos registradores na forma [16]:

$$|\psi_3\rangle = \sqrt{\frac{r}{2^{t_0}}}\sum_{a=0}^{\frac{2^{t_0}}{r}-1}|ar+b_0\rangle_{R_0}\,,\left|x^{b_0} \mod N\right\rangle_{R_1}$$

Como é necessário que o r calculado seja par, essa não é uma simplificação tão distante. Mas, mesmo que r não fosse uma potência de 2, o estado de R_0 ainda seria periódico em r e faz-se presente a necessidade qde adaptar algumas partes do algoritmo; então, segu-se com essa simplificação sem grandes perdas na generalidade [16].

5. Aplica-se a inversa da transformada de Fourier Quântica ao primeiro registrador, resultando em:

$$|\psi_3\rangle = |\psi_2\rangle \overset{QFT^{-1}}{\to} \sqrt{\frac{r}{2^{t_0}}}\sum_{a=0}^{\frac{2^{t_0}}{r}-1}\left(\frac{1}{\sqrt{2^{t_0}}}\sum_{j=0}^{2^{t_0}-1}e^{\frac{-2\pi i(ar+b_0)j}{2^{t_0}}}|j\rangle\right)_{R_0}, \left|x^{b_0} \mod N\right\rangle$$

$$= \frac{1}{\sqrt{r}}\left(\sum_{j=0}^{2^{t_0}-1}\left(\frac{1}{\frac{2^{t_0}}{r}}\sum_{a=0}^{\frac{2^{t_0}}{r}-1}e^{-2\pi i\frac{j\cdot a}{\frac{2^{t_0}}{r}}}\right)e^{\frac{-2\pi ijb_0}{2^{t_0}}}|j\rangle\right)_{R_0}, \left|x^{b_0} \mod N\right\rangle$$

O termo nos parênteses mais internos, então, pode ser simplificado de acordo com:

$$\frac{1}{L}\sum_{a=0}^{L-1}e^{-2\pi i\frac{j\cdot a}{L}} = \begin{cases}1 \Leftarrow \exists\ a \in \mathbb{Z} : j = aL\\ 0 \Leftarrow \nexists\ a \in \mathbb{Z} : j = aL\end{cases}$$

Para obter:

$$= \frac{1}{\sqrt{r}}\left(\sum_{j=0}^{2^{t_0}-1}e^{\frac{-2\pi ijb_0}{2^{t_0}}}|j\rangle\right)_{R_0}, \left|x^{b_0} \mod N\right\rangle_{R_1}$$

Então é feita a substituição $k = \frac{2r}{2^{t_0}}$ e obtém-se:

$$= \frac{1}{\sqrt{r}} \left(-e^{\frac{t\pi i 2^{t_0} b_0}{2^{t_0}}} + \sum_{j=0}^{2^{t_0}} e^{\frac{-2\pi i j b_0}{2^{t_0}}} |j\rangle \right)_{R_0} , \left| x^{b_0} \mod N \right\rangle_{R_1}$$

$$\overset{j \to k}{\to} \frac{1}{\sqrt{r}} \left(-e^{\frac{t\pi i 2^{t_0} b_0}{2^{t_0}}} + \sum_{k=0}^{r} e^{\frac{-2\pi i k b_0}{r}} \left| \frac{2^{t_0} k}{r} \right\rangle \right)_{R_0} , \left| x^{b_0} \mod N \right\rangle_{R_1}$$

$$= \frac{1}{\sqrt{r}} \left(\sum_{k=0}^{r-1} e^{\frac{-2\pi i k b_0}{r}} \left| \frac{2^{t_0} k}{r} \right\rangle \right)_{R_0} , \left| x^{b_0} \mod N \right\rangle_{R_1}$$

6. Mede-se o registrador R_0 para obter um resultado na forma $\left| \frac{2^{t_0} k}{r} \right\rangle$ onde não é conhecido o k. A partir desse resultado, aplicamos o método de frações continuadas a uma amostra de resultados obtidos para extrairmos r.

Dessa forma, estabelece-se um método capaz de obter(com uma quantidade polinomial de operações em relação ao input) a fatoração de um número composto em números primos. Em outras palavras, o algoritmo fornece um procedimento eficiente para resolver o problema da fatoração inteira, que é justamente a base da segurança de sistemas criptográficos como o RSA, bem como de outros esquemas de criptografia assimétrica. [16].

Capítulo 5

Impactos na Criptografia e Desafios Físicos

O algoritmo de Shor apresenta uma mudança fundamental de paradigma. O estudo da eficiência de algoritmos, seja por meio do modelo da Máquina de Turing, ou de outros métodos é fundamentalmente associado com a natureza da implementação desses algoritmos. Nesse contexto, algoritmos quânticos como o de Shor, são capazes de oferecer uma melhora exponencial na complexidade computacional da execução de tarefas especificas [18].

Como demonstrado anteriormente, essa capacidade de fatorar números inteiros de forma exponencialmente mais eficiente, coloca em risco a segurança de, essencialmente, todo o legível criptografado com o algoritmo RSA. Além disso, no mesmo trabalho, Shor [18] também demonstra um algoritmo capaz de resolver o problema do logaritmo discreto. Isso compromete a segurança de diversos outros protocolos criptográficos como o Diffie-Helman e as técnicas de Assinatura Digital sobre Curvas Elípticas (*Eliptic Curve Digital Signature Algorithm* - ECDSA). Como mostra Thorsten Kleinjung et al o tempo para quebrar o RSA com uma chave de 768 bits para um unico *core* do processador 2.2GHz AMD Opteron é da ordem de 2000 anos; todavia mesmo utilizando diversos processadores o processo levou meses para ser concluido [20]. Por outro lado, estimativas indicam que um computador quântico será capaz de executar a fatoração de números da ordem de 2048 bits em algumas horas [21].

Diante desde cenário, agências como o NIST, têm conduzido processos de padronização e escolhas de algoritmos com segurança pós quântica [22]. Esses algoritmos baseiam-se em problemas computacionais para os quais não se conhecem soluções eficientes seja por algoritmos clássicos ou quânticos. Agências governamentais também vem colocando metas e prazos para a adaptação das aplicações críticas com os métodos de criptografia pós-quântica.

Mediante esse potencial de avanço e aplicações dessas máquinas, empresas como Google, Microsoft, IBM, dentre outras fazem investimento contínuo no desenvolvimento de técnicas mais confiáveis para a produção de qubits, pesquisa em correção de erros, algoritmos, dentre diversos outros [23] [24] . Estes investimentos são reflexos do desafio de

concretizar um computador quântico funcional [25] [26] [27].

Um computador quântico precisa satisfazer uma serie de requisitos para oferecer capacidade de computação útil. Por exemplo, nos algoritmos discutidos anteriormente, há necessidade de *preparação de um estado inicial especifico*, *aplicação de uma operação reversível*, *evolução de um estado quântico* e *leitura consistente dos registradores*. Ou seja, faz-se necessário que um computador quântico seja capaz de representar informação quântica de forma consistente; ser capaz de (de forma consistente) permitir a preparação de um estado inicial; performar uma variedade de transformações unitárias; oferecer medições confiáveis dos seus registradores [5].

Esses desafios permitem uma gama ampla de possibilidades de representações de um qubit e, consequentemente, uma gama de diferentes arquiteturas possíveis. Mas, esses diferentes modelos são limitados fundamentalmente pelas propriedades físicas das representações escolhidas, por exemplo o sistema quântico escolhido não pode ser tão desacoplado que não pode ser mensurado, mas não pode ser acoplado a outros sistemas de forma que seu estado seja destruído muito rapidamente.

Por exemplo, a Tabela 5.1, mostra algumas possibilidades de representações de qubit e seus tempos de decoerência quântica (τ_Q), ou seja, tempo que o sistema evolui de forma coerente sem colapsar (parcial ou totalmente); Tempo por operação (τ_{op}) e o número máximo de operações $n_{op} = \frac{\tau_Q}{\tau_{op}}$

Tabela 5.1: Tempos característicos em segundos de diversos sistemas

Sistema	τ_Q	τ_{op}	n_{op}
Spin - Elétron	10^{-3}	10^{-7}	10^4
Armadilha de Ion	10^{-1}	10^{-14}	10^{13}
Elétron - Au	10^{-8}	10^{-14}	10^6
Elétron - GaAs	10^{-10}	10^{-13}	10^3
Quantum Dot	10^{-6}	10^{-9}	10^3
Cavidade Óptica	10^{-5}	10^{-14}	10^9
Cavidade de Micro-ondas	10^0	10^{-4}	10^4

Fonte: [5]

Que exemplifica parte dos *trade-offs* que tem que ser considerados na produção de um computador quântico. Assim, fica evidente que a Computação Quântica é uma área com grande potencial de aplicações e impactos nos métodos de segurança modernos. Além disso, diversos aspectos, como a produção de software voltados para a computação quântica, estão ainda no início do seu desenvolvimento amplo ferramentas como o Qiskit da IBM exemplificam isso. Outros trabalhos recentemente apresentados no Instituto de Física da Universidade Federal de Mato Grosso do Sul (UFMS) justamente nos tópicos do Qiskit e da correção de erros quânticos também apresentam a relevância do tópico atualmente (2026) e a importância do seu estudo e análise [28] [29].

Capítulo 6

Conclusão

O presente estudo introduziu os conceitos fundamentais das técnicas de criptografia simétrica e assimétrica com exemplos diversos e ênfase no algoritmo RSA. O texto também abordou uma breve introdução à computação quântica, dando destaque aos algoritmos de Deutsch-Jozsa, Shor e aos seus potenciais impactos nas técnicas de criptografia moderna. Discutiram-se, também, os princípios fundamentais da mecânica quântica, como superposição, operadores unitários e medições. Em seguida, aplicaram-se esses conceitos a uma série de algoritmos quânticos relevantes, destacando-se o algoritmo de Shor, capaz de resolver o problema da fatoração de um número inteiro de forma exponencialmente mais eficiente que algoritmos clássicos. Além da introdução conceitual dos algoritmos quânticos, ilustrou-se o método de busca de ordem utilizado no algoritmo de Shor através de uma implementação clássica utilizando a linguagem de programação Python, permitindo observar experimentalmente o processo de fatoração.

Apesar do potencial teórico desses algoritmos, a implementação prática com grandes conjuntos de qubits ainda enfrenta limitações significativas, especialmente relacionadas à estabilidade dos qubits, correção de erros quânticos e escalabilidade dos dispositivos atuais. Como perspectivas futuras, destaca-se a investigação de algoritmos de criptografia pós-quântica, bem como a implementação prática de algoritmos quânticos em simuladores ou dispositivos quânticos disponíveis atualmente. Dessa forma, a computação quântica se apresenta como uma área de pesquisa promissora, com potencial para transformar significativamente diversos campos da ciência da computação e da segurança da informação nas próximas décadas.

Apêndice A

Aritmética Modular

Diz-se que um número a é *congruente* a um número b módulo m se

$$\exists\ k\ \ \in \mathbb{Z} : a = b + km \tag{A.1}$$

Referências Bibliográficas

[1] AHMAD, D. R. M. et al. Chapter 6 - cryptography. In: *Hack Proofing Your Network (Second Edition)*. Second edition. Burlington: Syngress, 2002. p. 165–203. ISBN 978-1-928994-70-1. Disponível em: <https://www.sciencedirect.com/science/article/pii/B9781928994701500094>.

[2] BRASIL. Lei nº 14.063, de 23 de setembro de 2020. *Diário Oficial [da] República Federativa do Brasil*, Brasília, DF, 2020. Disponível em: <http://www.planalto.gov.br/ccivil$_0$3/$_A$to2011 − 2014/2012/*Lei*/*L*12651.*htm*>.

[3] FEYNMAN, R. P. Simulating physics with computers. *Int. J. Theor. Phys.*, Springer Science and Business Media LLC, v. 21, n. 6-7, p. 467–488, jun. 1982.

[4] DEUTSCH, D. Quantum theory, the church–turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, The Royal Society, v. 400, n. 1818, p. 97–117, jul. 1985. ISSN 2053-9169. Disponível em: <http://dx.doi.org/10.1098/rspa.1985.0070>.

[5] NIELSEN, M. A.; CHUANG, I. L. *Quantum Computation and Quantum Information*. Cambridge, England: Cambridge University Press, 2012.

[6] ORúS, R.; MUGEL, S.; LIZASO, E. Quantum computing for finance: Overview and prospects. *Reviews in Physics*, v. 4, p. 100028, 2019. ISSN 2405-4283. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2405428318300571>.

[7] RAVI, P.; CHATTOPADHYAY, A.; BHASIN, S. Security and quantum computing: An overview. In: *2022 IEEE 23rd Latin American Test Symposium (LATS)*. [S.l.: s.n.], 2022. p. 1–6.

[8] TERADA, R. *Segurança de dados: criptografia em redes de computador*. [S.l.]: Edgard Blucher, 2008.

[9] KERCKHOFFS, A. La cryptographie militaire. *Journal des Sciences Militaires*, p. 161–191, 1883.

[10] Disponível em: <https://www.nist.gov/glossary-term/18756>.

[11] Information Technology Standards Commission of Japan. *ISO-IR-006: ASCII Graphic character set*. [S.l.], 1975. Disponível em: <https://itscj.ipsj.or.jp/ir/006.pdf>.

[12] DIVISION, I. T. L. C. S. *Block cipher techniques: CSRC*. Disponível em: <https://csrc.nist.gov/projects/block-cipher-techniques>.

[13] PIZA, A. F. R. de T. *Mecânica quântica*. [S.l.]: Edusp, 2003.

[14] COHEN-TANNOUDJI, C.; DIU, B.; LALOë, F. *Quantum mechanics*. New York, NY: Wiley, 1977. v. 1. Trans. of : Mécanique quantique. Paris : Hermann, 1973. Disponível em: <https://cds.cern.ch/record/101367>.

[15] Nobel Prize Outreach. *Press release*. 2026. Accessed: 2026-02-25. Disponível em: <https://www.nobelprize.org/prizes/physics/2025/press-release/>.

[16] WILLIAMS, C. P. *Explorations in Quantum Computing*. Springer London, 2011. ISSN 1868-095X. ISBN 9781846288876. Disponível em: <http://dx.doi.org/10.1007/978-1-84628-887-6>.

[17] GRIFFITHS, D. J.; SCHROETER, D. F. *Introduction to Quantum Mechanics*. 3. ed. [S.l.]: Cambridge University Press, 2018.

[18] SHOR, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, v. 26, n. 5, p. 1484–1509, 1997. Disponível em: <https://doi.org/10.1137/S0097539795293172>.

[19] ROSEN, K. H. *Elementary number theory and its applications (3. ed.)*. [S.l.]: Addison-Wesley, 1993. I-XV, 1-544 p. ISBN 978-0-201-57889-8.

[20] KLEINJUNG, T. et al. *Factorization of a 768-bit RSA modulus*. 2010. Cryptology ePrint Archive, Report 2010/006. Disponível em: <https://eprint.iacr.org/2010/006.pdf>.

[21] GIDNEY, C.; EKERÅ, M. How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. *Quantum*, Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften, v. 5, p. 433, abr. 2021. ISSN 2521-327X. Disponível em: <https://doi.org/10.22331/q-2021-04-15-433>.

[22] National Institute of Standards and Technology (NIST). *Announcing Approval of Three Federal Information Processing Standards (FIPS) for Post-Quantum Cryptography*. 2024. `https://csrc.nist.gov/News/2024/postquantum-cryptography-fips-approved`. Accessed: 2026-02-27.

[23] ABANIN, D. A. et al. Observation of constructive interference at the edge of quantum ergodicity. *Nature*, Springer Science and Business Media LLC, v. 646, n. 8086, p. 825–830, out. 2025. ISSN 1476-4687. Disponível em: <http://dx.doi.org/10.1038/s41586-025-09526-6>.

[24] ACHARYA, R. et al. Suppressing quantum errors by scaling a surface code logical qubit. *Nature*, Springer Science and Business Media LLC, v. 614, n. 7949, p. 676–681, fev. 2023. ISSN 1476-4687. Disponível em: <http://dx.doi.org/10.1038/s41586-022-05434-1>.

[25] ARUTE, F. et al. Quantum supremacy using a programmable superconducting processor. *Nature*, Springer Science and Business Media LLC, v. 574, n. 7779, p. 505–510, out. 2019. ISSN 1476-4687. Disponível em: <http://dx.doi.org/10.1038/s41586-019-1666-5>.

[26] BORDIN, A. et al. Enhanced majorana stability in a three-site kitaev chain. *Nature Nanotechnology*, Springer Science and Business Media LLC, v. 20, n. 6, p. 726–731, mar. 2025. ISSN 1748-3395. Disponível em: <http://dx.doi.org/10.1038/s41565-025-01894-4>.

[27] KOTIL, A. et al. Quantum approximate multi-objective optimization. *Nature Computational Science*, Springer Science and Business Media LLC, v. 5, n. 12, p. 1168–1177, out. 2025. ISSN 2662-8457. Disponível em: <http://dx.doi.org/10.1038/s43588-025-00873-y>.

[28] MELO, G. R. *Protocolo Baseado em Feedback para Correção de Ruído Quântico: Proposta e Avaliação Teórica*. Trabalho de Conclusão de Curso — Universidade Federal de Mato Grosso do Sul, 2024. Disponível em: <https://repositorio.ufms.br/handle/123456789/10965>.

[29] MACEDO, G. F. L. d. *Conceitos Básicos de Computação Quântica Aplicados no Qiskit*. Trabalho de Conclusão de Curso — Universidade Federal de Mato Grosso do Sul, 2024. Disponível em: <https://repositorio.ufms.br/handle/123456789/9810>.