Capítulo 3

Introdução à Computação Quântica e Impactos em Criptografia

Victor Takashi Hayashi (USP), Bryan Kano Ferreira (USP), Reginaldo Arakaki (USP), Jonatas Faria Rossetti (Bradesco), Routo Terada (USP), Ever Costa (Inteli), Wildisley Filho (Inteli), Giovanna Vieira (Inteli), Luiza Petenazzi (Inteli), Priscila Falcão (Inteli)

Abstract

Quantum Computing is an emerging technology with the potential to solve some complex problems considered intractable by classical computers. However, this potential poses risks to information security, since quantum algorithms can break encryption methods widely used in current systems. This short course aims to introduce the fundamentals of quantum computing and examine its impacts on cryptography, supporting a deeper understanding of the reasons for these impacts based on an understanding of its mathematical foundations, in addition to providing a more comprehensive view of possible solutions beyond post-quantum cryptography. Given the relevance of the scenario of breaking asymmetric algorithms such as RSA, Quantum Computing is one of the main tools for understanding the need and importance of post-quantum cryptography and its standardization by NIST.

Resumo

A Computação Quântica é uma tecnologia emergente com potencial para resolver alguns problemas complexos considerados intratáveis por computadores clássicos. No entanto, esse potencial representa riscos à segurança da informação, uma vez que algoritmos quânticos podem quebrar métodos de criptografia amplamente utilizados em sistemas atuais. Este minicurso tem como objetivo introduzir os fundamentos da computação quântica e examinar seus impactos na criptografia, apoiando uma compreensão mais profunda das razões para esses impactos com base na compreensão de seus fundamentos matemáticos, além de fornecer uma visão mais abrangente de possíveis soluções além da criptografia pós-quântica. Dada a relevância do cenário de quebra de algoritmos de criptografia assimétrica como o RSA, a Computação Quântica é uma das principais ferramentas para a compreensão da necessidade e da importância da criptografia pós-quântica e de sua padronização pelo NIST.

3.1. Introdução

A Computação Quântica é uma tecnologia emergente com potencial para resolver alguns problemas complexos considerados intratáveis por computadores clássicos. Entretanto, esse potencial traz riscos à segurança da informação, já que algoritmos quânticos podem quebrar métodos de criptografia amplamente utilizados nos sistemas atuais [Gamble 2019, Khan et al. 2024].

Este minicurso tem como objetivo geral introduzir os fundamentos da computação quântica e examinar seus impactos em criptografia, suportando um aprofundamento maior nas razões para esses impactos a partir do entendimento de seus fundamentos matemáticos, além de fornecer uma visão mais abrangente sobre possíveis soluções além da criptografia pós-quântica, que são os principais diferenciais em relação aos minicursos apresentados em edições anteriores do SBSeg [Barreto et al. 2013, Paiva et al. 2023].

Além disso, o minicurso está alinhado com propósito similar ao workshop internacional ACM QSec[1], para fomentar a sinergia entre comunidades de pesquisa em Computação Quântica (principalmente aquelas relacionadas à Criptografia Quântica com *Quantum Key Distribution*) e de Segurança da Informação (que vêm pesquisando por muitos anos abordagens de Criptografia Pós-Quântica).

O minicurso destina-se a estudantes de graduação e pós-graduação, pesquisadores e profissionais de Computação, Engenharias e áreas afins, especialmente aqueles interessados em Segurança da Informação e em Computação Quântica. Dado o caráter interdisciplinar do tema de Impactos da Computação Quântica em Criptografia, que combina fundamentos de Mecânica Quântica, Matemática e Ciência da Computação, o conteúdo foi planejado para ser acessível a participantes com conhecimentos básicos de Ciência da Computação e Criptografia, não exigindo formação aprofundada em Mecânica Quântica. O público esperado inclui pessoas que desejam se atualizar sobre o impacto dos computadores quânticos na segurança da informação e conhecer as estratégias para proteger sistemas frente a essa tecnologia emergente.

Dada a relevância do cenário de quebra de algoritmos de criptografia assimétrica como o RSA e a padronização do NIST sobre criptografia pós-quântica [Alagic et al. 2022], o curso abordará desde os fundamentos da criptografia clássica na Seção 3.2, passando pelos princípios da computação quântica na Seção 3.3, até os potenciais ataques quânticos sobre algoritmos criptográficos atuais e as soluções em desenvolvimento para mitigar esses riscos na Seção 3.4. O tema será tratado de forma didática e acessível, conectando teoria e prática, de modo a ressaltar tanto a necessidade de atualizar os mecanismos de segurança diante da era quântica quanto as novas oportunidades tecnológicas que emergem desse contexto na Seção 3.5. As considerações finais com reflexões, aspectos éticos tangenciais e exemplos de capacitação na área são apresentadas na Seção 3.6.

3.2. Fundamentos de Criptografia

Nesta seção são apresentados os fundamentos de criptografia, trazendo os principais requisitos de segurança da informação que devem ser suportados pelos mecanismos cripto-

[1] `https://acm-qsec.com/`

gráficos: Confidencialidade, Integridade, Autenticidade e Irretratabilidade. Também são apresentados os fundamentos matemáticos de funções *hash* criptográficas, criptografia simétrica e criptografia assimétrica [Menezes et al. 2018].

3.2.1. Requisitos de Segurança

A Segurança da Informação é uma área multidisciplinar que tem como objetivo a proteção da informação e, consequentemente, dos sistemas que a processam contra ameaças que possam vir a comprometer seu valor. Uma definição usualmente adotada consta na legislação norte-americana, utilizada por órgãos como o *National Institute of Standards and Technology* (NIST), o *Department of Homeland Security* (DHS), o *Office of Management and Budget* (OMB) e a *Federal Information Security Modernization Act* (FISMA) como base para suas diretrizes técnicas:

> Segurança da Informação significa proteger informações e sistemas de informação contra acesso, uso, divulgação, interrupção, modificação ou destruição não autorizados, com o objetivo de assegurar:
>
> (A) **Integridade**, que significa proteger contra modificação ou destruição indevida da informação, incluindo a garantia de *irretratabilidade* e *autenticidade*;
>
> (B) **Confidencialidade**, que significa preservar as restrições autorizadas de acesso e divulgação, incluindo mecanismos de proteção da privacidade pessoal e de informações proprietárias;
>
> (C) **Disponibilidade**, que significa assegurar o acesso e uso oportunos e confiáveis da informação.
>
> [44 U.S. Code § 3542 – Definitions 2013]

Temos dois requisitos complementares à integridade. A *irretratabilidade*, também conhecida pelo termo, *não-repúdio*, pode ser definida como a propriedade que um emissor legítimo não possa negar a autoria de uma ação. A autenticidade, por sua vez, é a propriedade que garante que uma parte é, de fato, quem afirma ser [Menezes et al. 2018].

O esforço para assegurar os requisitos objetivados depende de ações coordenadas multidisciplinares que ultrapassam soluções técnicas isoladas [Stallings 2013]. De acordo com Anexo A da ISO/IEC 27001:2022, os controles necessários estão organizados em quatro categorias principais, sendo estes: controles organizacionais, relacionados a pessoas, físicos e tecnológicos. Em resumo, os controles de natureza organizacional concentram-se na formulação e condução de diretrizes e políticas de caráter estratégico e tático. Os controles voltados às pessoas abrangem dimensões relacionadas ao comportamento humano e à atribuição de responsabilidades. Já os controles físicos têm como foco a segurança dos espaços e dos ativos materiais. Por fim, os controles tecnológicos tratam de instrumentos técnicos empregados no contexto digital. Nesta última categoria, encontra-se a Criptografia.

3.2.2. Introdução à Criptografia

A Criptografia é a área do conhecimento que estuda as técnicas matemáticas que asseguram requisitos da Segurança da Informação [Menezes et al. 2018]. Tal definição pode ser

melhor especificada quando incluímos um aspecto central que a distingue de outras abordagens com o mesmo objetivo, que é o estudo de proteções que atuam diretamente sobre os dados, através de transformações na própria informação [Stallings 2013, Terada 2008]. Isto significa que a criptografia, como ferramenta, é um subconjunto de mecanismos técnicos que atuam na camada de dados, diferenciado-se de outros controles que protegem o contexto em que a informação circula, como *firewalls*, *antimalwares* e sistemas de autenticação de usuários.

A camada de Segurança de Dados por meio do uso da criptografia pode ser entendida como uma configuração primária em qualquer ecossistema de mecanismos técnicos de segurança. Essa visão é corroborada pelo conceito de defesa em profundidade, amplamente conhecido na indústria e referenciado pelo NIST *Cybersecurity Framework* [Pascoe 2023], onde essa atuação no nível mais baixo da pilha de segurança confere à criptografia um papel fundacional, em relação aos demais controles técnicos.

Usualmente, os requisitos de segurança tidos como objetivos a serem fornecidos por mecanismos criptográficos estão associados à confidencialidade, integridade, autenticação e irretratabilidade da informação [Menezes et al. 2018]. Embora, tradicionalmente a disponibilidade não fosse considerada uma propriedade diretamente assegurada pela criptografia, arquiteturas distribuídas têm ampliado esse escopo, revelando o papel de mecanismos criptográficos na manutenção da disponibilidade confiável de dados e serviços [Bonneau et al. 2015].

Os cenários adversariais considerados pelo estudo da criptografia tomam cada fase do ciclo de vida da informação como objeto de análise. Durante a transmissão (Figura 3.1), onde os dados são enviados de um ponto a outro através de um canal de comunicação, os principais problemas envolvem a interceptação da mensagem por terceiros (quebra da confidencialidade), a modificação do conteúdo em trânsito (quebra da integridade), a falsificação da identidade do remetente (quebra da autenticidade) e a negação do envio após a entrega (quebra da irretratabilidade). Com exceção da negação do envio após a entrega, que toma uma ação de má índole de uma parte legítima da comunicaçaõ, todos os outros problemas, se tomados como intencionais, podem ser englobados no cenário adversarial clássico em criptografia, onde uma parte terceira e desautorizada *C*, denominada na literatura como Carlos, tem acesso ao canal de comunicação [Terada 2008].

Tomando a fase do ciclo de vida da informação, onde a informação está em re-

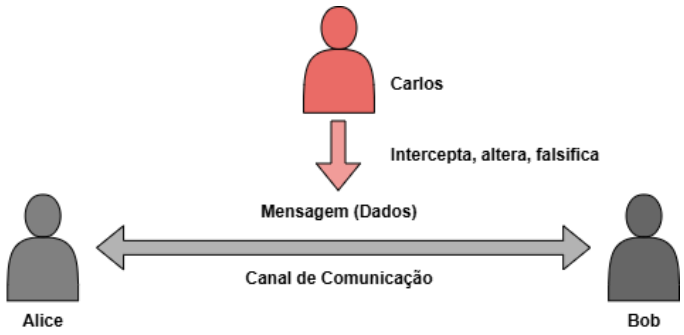


Figura 3.1. Cenário de Transmissão de Dados. Adaptado de [Terada 2008]

pouso, também chamada de armazenamento (Figura 3.2), esse cenário pode ser entendido como uma simplificação estática do problema da transmissão, na medida em que os dados permanecem parados, mas ainda sujeitos a ameaças semelhantes. Temos como problemas e cenários adversariais o acesso indevido por terceiros não autorizados à base de dados (quebra da confidencialidade) e a modificação ilegítima dos dados (quebra da integridade). Adicionalmente, para fins de auditabilidade e perícia, temos problemas relacionados à atribuição incorreta da autoria ou origem dos dados (quebra da autenticidade) e a tentativa de negar a autoria de um conteúdo previamente salvo (quebra da irretratabilidade). Uma modelagem clássica adversarial para o cenário está presente na Figura 3.2, onde o atacante deseja conseguir acesso à base de dados para perpetrar ações que podem impactar todos os requisitos de segurança considerados [Terada 2008].

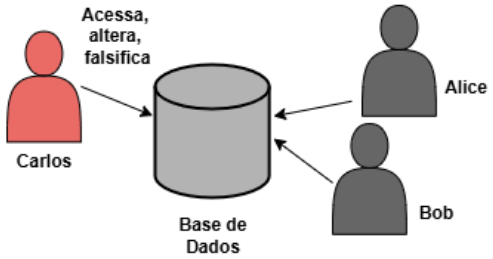


Figura 3.2. Cenário de Dados em Repouso. Adaptado de [Terada 2008]

Para fins de ciência, a etapa de processamento dos dados, embora inviável até o momento, também está sendo explorada pela comunidade acadêmica. Isso significa que os problemas considerados passam não só a considerar atacantes externos, mas mudam o paradigma de confiança no processador. Dessa forma, os dados permanecem encriptados em processamento. Esse conjunto de algoritmos criptográficos é conhecido como criptografia homomórfica [Gentry 2009].

Para assegurar todos os requisitos elencados nos múltiplos cenários adversariais, torna-se evidente a necessidade da composição de múltiplos mecanismos criptográficos. Em uma comunicação segura, por exemplo, utiliza-se criptografia assimétrica para troca de chaves, criptografia simétrica para proteger os dados, autenticação e assinaturas digitais para garantir a identidade. No armazenamento, combina-se criptografia de disco, funções *hash* criptográficas e assinaturas.

Esses mecanismos podem ser classificados, de forma simplificada, em três tipos de primitivas criptográficas (Figura 3.3). A primitiva sem o uso de chaves trata de mecanismos que não dependem de nenhum tipo de chave secreta para funcionar, como funções *hash* e geradores pseudoaleatórios. A primtiva de chave simétrica trata de mecanismos que utilizam a mesma chave secreta para encriptar e desencriptar. Nesta primitiva, destacam-se os encriptadores de dados como cifras de bloco e cifras de fluxo, além de autenticadores do tipo MACs (*Message Authentication Codes*). Por fim, a primitiva de chaves assimétricas utiliza um par de chaves, sendo uma pública e uma privada. Mecanismos assimétricos são geralmente empregados para o encapsulamento de chaves simétricas (*Key Encapsulation Mechanism - KEM*) e assinaturas digitais [Menezes et al. 2018].

Nas próximas seções exploraremos os fundamentos dos mecanismos mais usuais

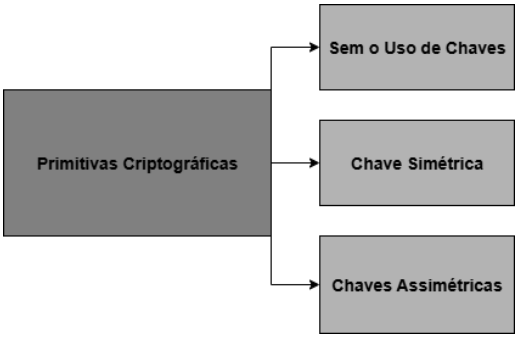


Figura 3.3. Primitivas Criptográficas de Segurança. Adaptado de [Menezes et al. 2018]

associados a cada tipo de primitiva.

3.2.3. Fundamentos Matemáticos de Funções Hash Criptográficas

Funções *hash* são os mecanismos mais usuais da primitiva de segurança sem o uso de chaves criptográficas. Originalmente foram concebidas dentro da área de Estrutura de Dados como estruturas associativas de chave e valor, utilizadas para otimizar o acesso eficiente a informações [Knuth 1997]. Formalmente, uma função *hash* pode ser definida como uma função determinística que produz uma saída de comprimento fixo.

$$H : \{0,1\}^* \rightarrow \{0,1\}^n \tag{1}$$

Onde H é a função *hash* propriamente dita, $\{0,1\}^*$ representa o conjunto de todas as cadeias binárias de comprimento arbitrário de entrada e $\{0,1\}^n$ representa o conjunto de todas as cadeias binárias de comprimento fixo de saída.

Nesse contexto primário, o uso dessas funções têm o fim de permitir operações rápidas de busca, inserção e remoção em bancos de dados. Nesse uso, não há preocupações com adversários ou segurança formal, colisões são aceitáveis e tratadas de forma eficiente. A transição do uso das funções para uso como mecanismos criptográficos surge em trabalhos seminais nas décadas de 70 e 80 [Merkle 1979, Lamport 1981]. Essa mudança de paradigma, inicialmente, deslocou o foco da indexação eficiente de dados para a verificação da integridade de dados, criando o tipo de função *hash* que pode ser classificado como MDCs (*Modification Detection Codes*) [Stallings 2013].

Em termos objetivos, se determinado conjunto de dados x é alterado resultando em x', então a saída da função *hash* $H(x')$ também será alterada de forma significativa em relação à saída original de $H(x)$, mesmo que a modificação em x seja mínima [Menezes et al. 2018]. Essa propriedade é chamada de efeito avalanche e é essencial para detectar alterações acidentais ou maliciosas [Terada 2008].

Tratando-se, agora, de uma ferramenta de segurança, tornou-se necessário uma formalização e consolidação dos requisitos de segurança que uma função *hash* criptográfica deve possuir [Menezes et al. 2018]:

1. **Resistência à pré-imagem**: dado um valor de saída $h \in \{0,1\}^n$, é difícil encontrar qualquer entrada $x \in \{0,1\}^*$ tal que $H(x) = h$.

2. **Resistência à segunda pré-imagem**: dado um valor de entrada $x \in \{0,1\}^*$, é difícil encontrar outro valor $x' \neq x$ tal que $H(x') = H(x)$.

3. **Resistência a colisões**: é difícil encontrar quaisquer dois valores distintos $x, x' \in \{0,1\}^*$ tais que $H(x) = H(x')$.

Esses requisitos elencados não são apenas teóricos, mas estão associados à proteção contra ataques reais. A resistência à pré-imagem é essencial para defesa contra ataques de inversão, como nos casos de armazenamento de senhas, onde mesmo que os *hashes* sejam expostos, não deve ser possível recuperar as senhas originais. Na quebra da segunda pré-imagem, o atacante reage a uma mensagem já existente; na quebra de colisão, ele atua proativamente, criando duas mensagens distintas que compartilham o valor *hash*. A quebra de ambas implicam em uma miríade de potenciais ações maliciosas, como falsificação de documentos, manipulação de registros e evasão de verificação de integridade. Ataques viáveis a esses dois requisitos de funções *hash* criptográficas já foram demonstrados na prática contra funções amplamente utilizadas como MD5 e SHA-1 [Wang et al. 2004, Wang et al. 2005], descartadas posteriormente.

A família de algoritmos SHA-3 é o padrão mais recente de funções *hash* criptográficas aprovado pelo NIST [FIPS PUB 202 2015]. Enquanto funções como SHA-1 e SHA-2 têm como base a construção de Merkle–Damgård, o SHA-3 utiliza a construção esponja do algoritmo Keccak [Katz and Lindell 2014]. Essa construção é diferente das abordagens anteriores por operar sobre um estado interno fixo dividido em duas partes, a *rate* (r) e a *capacity* (c), de forma que $r + c = b$, onde b representa o tamanho total do estado, usualmente 1600 *bits*. Na fase de absorção, a operação XOR é utilizada para combinar os blocos da entrada com os *r bits* do estado interno, posteriormente ocorrendo uma permutação não linear. Após toda a entrada ser processada, a fase de extração (em inglês, *squeezing*) gera o valor de saída a partir dos mesmos *r bits*. Na prática, esse funcionamento confere propriedades de segurança adequadas. Por conta da estrutura modular, a construção esponja ainda permite a adição de outros mecanismos como funções de autenticação do tipo KMAC (*Keccak Message Authentication Code*), consolidando sua versatilidade e segurança.

A capacidade de representar grandes volumes de dados usando saídas compactas aliada a sua simplicidade conceitual e eficácia computacional, consolida esse tipo de mecanismo como ferramenta essencial para assegurar a integridade dos dados em um ecossistema de Segurança da Informação.

3.2.4. Fundamentos Matemáticos da Criptografia Simétrica

O conceito principal que permeia os algoritmos de criptografia simétrica é a utilização de uma mesma chave para o processo de encriptação e desencriptação de dados [Terada 2008]. Apesar da primitiva de chave simétrica embarcar não só encriptadores, como também funções *hash* de tamanho arbitráriocomo os MACs (*Message Authentication Codes*), sistemas de asssinatura, sequências pseudoaleatórias e primitivas de identificação [Menezes et al. 2018], o foco desta subseção é exclusivamente didático e visa apresentar os fundamentos dos encriptadores simétricos, que constituem a principal aplicação prática dessa classe de algoritmos.

A criptografia simétrica foi historicamente formulada para assegurar a confidencialidade de dados em cenários adversariais, tanto para o uso durante a transmissão de informações quanto no armazenamento de dados sensíveis [Stallings 2013]. Nesse modelo, uma única chave primária é compartilhada entre as partes autorizadas, permitindo que todas possam encriptar e desencriptar os dados enviados ou guardados. O pressuposto de segurança deste tipo de mecanismo está em assegurar o segredo da chave compartilhada. A quebra ou acesso à chave implica na quebra total da segurança do sistema. Um sistema de criptografia simétrica pode ser denotado conceitualmente como uma tríade de algoritmos:

$$\mathsf{Gen}(1^n),\ \mathsf{Enc}_k(m),\ \mathsf{Dec}_k(c),$$

onde $\mathsf{Gen}(1^n)$ é o algoritmo de geração de chave, que recebe um parâmetro de segurança n e retorna uma chave secreta k a ser compartilhada entre as partes. $\mathsf{Enc}_k(m)$ é o algoritmo de encriptação, que recebe a chave k e uma mensagem m, e tem como saída um encriptado c. $\mathsf{Dec}_k(c)$ é o algoritmo de desencriptação, que têm como entrada a chave secreta k e o encriptado c, retornando a mensagem original m [Katz and Lindell 2014, Terada 2008].

A Figura 3.4 mostra a aplicação do modelo simétrico em um cenário adversarial clássico, onde após a combinação da chave secreta, Alice realiza a encriptação da mensagem m que deseja enviar com o algoritmo Enc e a chave secreta k, gerando o encriptado c que é enviado para Bob através de um canal comprometido pelo atacante Carlos. Bob, ao receber c, utiliza o algoritmo de desencriptação Dec junto com a chave secreta k para recuperar a mensagem às claras m e ter acessso aos dados originais. Note que neste modelo, Carlos só possui acesso ao encriptado c, não tendo o acesso à informação verdadeira.

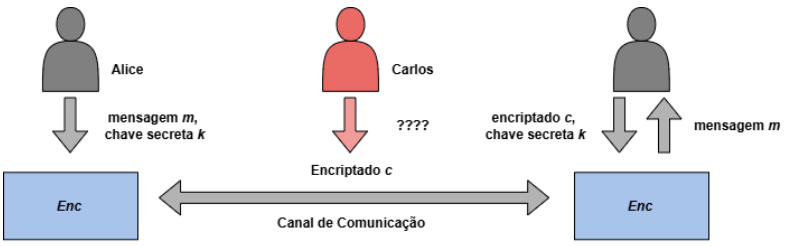


Figura 3.4. Modelo Simétrico de Criptografia. Adaptado de [Terada 2008]

Do ponto de vista de segurança, os algoritmos de criptografia simétrica buscam assegurar que a relação entre a chave secreta k, a mensagem m e o encriptado c seja suficientemente obscura matematicamente, de modo a inviabilizar qualquer ataque de reversão sem a posse da chave [Menezes et al. 2018]. Para isso, duas propriedades fundamentais postuladas por Claude Shannon são objetivadas: confusão, tem como objetivo tornar a relação entre a chave secreta e o encriptado o mais não-linear possível; difusão, por sua vez, busca espalhar a influência de cada *bit* da mensagem por vários *bits* do encriptado, viabilizando o efeito avalanche e aumentando a entropia [Terada 2008].

Essas duas propriedades são usualmente implementadas por meio de duas técnicas elementares em sistemas simétricos utilizadas desde o início da história da criptografia [Singh 1999], a substituição e a transposição. A substituição é uma operação criptográfica

em que os elementos da mensagem original, como *bits* ou *bytes* são trocados por outros segundo uma regra definida. A transposição é uma técnica que reorganiza a ordem dos elementos da mensagem ou de intermediários no processamento. Os *bits* ou blocos permanecem os mesmos, mas suas posições são alteradas [Menezes et al. 2018]. Algoritmos modernos contam com uma composição inteligente desses dois tipos de operações.

O AES (*Advanced Encryption Standard*) padronizado como encriptador simétrico de dados pelo NIST [of Standards et al. 2023], e amplamente adotado pela indústria, trabalha através de rodadas (em inglês, *rounds*) de procedimentos eficientes de substituição e transposição. O método SubBytes tem como objetivo aplicar uma transformação linear sobre cada *byte* do bloco de dados. O principal componente utilizado nesta etapa é a $S-box$ que é uma tabela de substituição baseada em inverso multiplicativo e transformação afim, que envolvem aritmética modular em corpo finito e operações booleanas com XOR, sendo extremamente eficientes computacionalmente [Menezes et al. 2018]. O ShiftRows baseia-se em transposição, rotacionando ciclicamente cada linha da matriz de estado. O método de MixColumns realiza uma mistura linear dos *bytes* de cada coluna, ao invés das linhas, da matriz. Por fim, a operação AddRoundKey baseia-se em aplicar a operação XOR entre o bloco de dados e uma subchave derivada da chave secreta, preparando o novo *round*.

Combinando essas transformações em múltiplas rodadas, o AES representa uma aplicação segura dos princípios de Shannon na prática da criptografia simétrica moderna. Em resumo, esquemas simétricos tem sua principal vantagem na simplicidade de suas operações e na alta performance computacional, sendo ideais para a encriptação de grande volume de dados [Menezes et al. 2018]. Porém, seu modelo pressupõe o compartilhamento seguro da chave secreta que torna sua aplicação exclusiva inviável em contextos com o estabelecimento frequente e dinâmico de conexões, como a Internet. Esse é o problema conhecido como compartilhamento de chaves e para endereçá-lo foi criada a primitiva de chaves assimétricas que é discutida na próxima subseção.

3.2.5. Fundamentos Matemáticos da Criptografia Assimétrica

Como discutido na subseção 3.2.4, apesar de encriptadores simétricos serem altamente eficientes, o modelo pressupõe e tem como limitação que todas as partes envolvidas na comunicação compartilhem previamente a chave secreta k, idealmente, por algum tipo de canal seguro. Em ambientes militares ou para uso em armazenamento, isso pode ser viável. Porém com a democratização da tecnologia e com o crescimento da Internet e outros tipos de redes distribuídas, tornou-se inviável estabelecer esses canais prévios de forma segura. O problema do compartilhamento de chaves tornou-se o desafio a ser superado. Nesse cenário, a criptografia de chave pública surge como uma solução [Menezes et al. 2018].

Introduzida seminalmente por meio de um protocolo de combinação de chave que permitia que duas partes estabelecessem um segredo comum [Diffie and Hellman 2022], e posteriormente, com a apresentação do algoritmo RSA [Rivest et al. 1978], que ampliou as funcionalidades e aplicações permitindo a encriptação direta de mensagens e a criação de assinaturas digitais, essa abordagem modificou o rumo da criptografia.

A criptografia assimétrica se baseia no uso de duas ou mais chaves com ações

diferentes. No caso de encriptadores com o fim de confidencialidade, temos o uso da chave pública *pk* para encriptação e o uso de uma chave privada *sk* para desencriptação. Um sistema de encriptação assimétrica pode ser denotado de forma generalista como uma tríade de algoritmos,

$$\mathsf{Gen}(1^n),\ \mathsf{Enc}_{pk}(m),\ \mathsf{Dec}_{sk}(c),$$

onde $\mathsf{Gen}(1^n)$ é o algoritmo que, a partir de um parâmetro de segurança n, gera um par de chaves, pública e privada (pk, sk). O algoritmo $\mathsf{Enc}_{pk}(m)$ é utilizado junto à chave pública para encriptar uma mensagem m, gerando o encriptado c. Ao usar o algoritmo $\mathsf{Dec}_{sk}(c)$ com a chave privada sk no encriptado c, a mensagem original m é recuperada [Katz and Lindell 2014, Terada 2008].

A Figura 3.5 exibe a aplicação do modelo assimétrico no modelo adversarial de confidencialidade. Alice, ao enviar uma mensagem m, utiliza a chave pública pk de Bob e o algoritmo de encriptação Enc para gerar um encriptado c, que é enviado através do canal de comunicação. Bob, ao receber c, realiza a desencriptação com uso de sua chave privada sk no algoritmo de desencriptação Dec, conseguindo a mensagem original m. Nesse esquema, Bob e Alice não precisaram combinar uma chave previamente: Alice usa a chave pública de Bob. Carlos, ao tentar interceptar a mensagem, consegue apenas o encriptado c que não possui nenhuma informação útil. Note que ao contrário de um esquema simétrico, a encriptação assimétrica não funciona para ambos os lados de forma direta. Se Bob deseja mandar uma mensagem para Alice, ele terá que obter a chave pública dela.

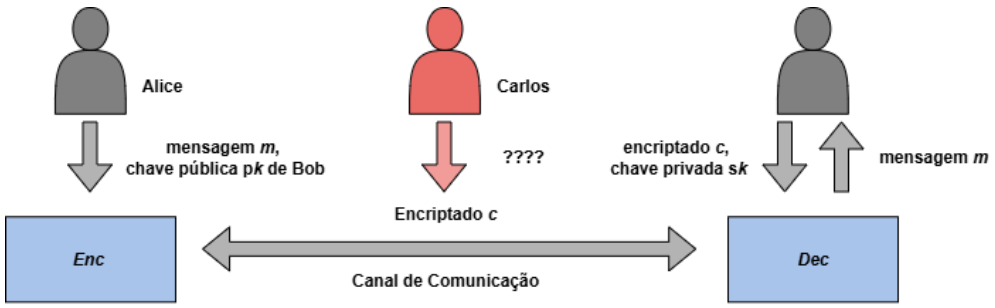


Figura 3.5. Modelo Assimétrico de Criptografia. Adaptado de [Terada 2008]

Invertendo o fluxo de encriptação, é possível derivar outro uso da criptografia assimétrica, a construção de assinaturas digitais, que tem como objetivo garantir autenticidade de dados. Nesse tipo de mecanismo, o remetente utiliza sua própria chave privada (sk) para assinar a mensagem, e qualquer terceiro, como Bob, pode verificar a autenticidade dessa assinatura utilizando a chave pública (pk) disponibilizada. Adicionalmente, a criptografia assimétrica é empregada em protocolos de acordo de chaves, que não envolvem encapsulamento via encriptação, além de esquemas de desafio-resposta, como os protocolos de Conhecimento Zero (ZK, do inglês *Zero Knowledge*) [Katz and Lindell 2014].

A ideia para construção de esquemas com base na primitiva assimétrica é o uso de uma função que é fácil de calcular em uma direção, encriptação por exemplo, mas difícil de inverter (desencriptação) sem uma informação necessária, a chave secreta *sk* [Menezes et al. 2018]. Essa assimetria de complexidade motivou a busca por problemas matemáticos computacionalmente difíceis, mas que permitissem uma solução eficiente quando a informação secreta está disponível. Diversas classes de problemas foram exploradas pela comunidade, as mais adotadas contemplam três tipos de abordagens.

O problema da fatoração de inteiros pode ser definido como: dado um número grande $N = pq$, onde p e q são primos grandes, o objetivo é encontrar p e q sem conhecer previamente um deles [Stallings 2013, Katz and Lindell 2014]. Isso era tido como computacionalmente inviável até a chegada do paradigma quântico. Porém, se um dos fatores é conhecido, a operação é trivial [Bernstein et al. 2009]. O problema do logaritmo discreto toma que dado um grupo finito G, um elemento $g \in G$ e outro elemento $h \in G$, o objetivo é encontrar um inteiro x tal que $g^x = h$. Uma variante do mesmo problema é o logaritmo discreto em curvas elípticas, que é mais eficiente para certas aplicações.

O problema fundador da fatoração dos inteiros é a base que sustenta o RSA [Rivest et al. 1978]. Para fins didáticos, o funcionamento simplificado e sem padronização do criptossistema é descrito abaixo [Terada 2008]. O processo é iniciado através da geração do par de chaves pública-privada (*pk* e *sk*):

1. Dois primos grandes p e q são escolhidos.
2. O módulo N comum das operações é calculado por $N = pq$.
3. A função totiente de Euler é calculada por $\phi(N) = (p-1)(q-1)$.
4. O expoente público e é escolhido sendo um inteiro tal que $1 < e < \phi(N)$ e $\gcd(e, \phi(N)) = 1$.
5. O expoente privado d é calculado tal que $d \equiv e^{-1} \ (mod\ \phi(N))$, isto é, o inverso multiplicativo de e módulo $\phi(N)$.

A chave pública resultante é o par (N, e) e a chave privada é (N, d). Para realizar a encriptação de uma mensagem $m \in \mathbb{Z}_N$, basta o remetente utilizar o expoente público e para calcular $c = m^e \bmod N$. É de trivial entendimento que a mesagem m pode ser recuperada com o uso do expoente secreto d através de $m = c^d \bmod N$.

O mesmo paper seminal [Rivest et al. 1978] mostra, ainda, como um esquema de assinatura digital pode ser derivado. A assinatura de um valor *hash* h de uma mensagem ou documento m pode ser feita utilizando a chave privada $sk = d$ por meio do cálculo $\sigma = h^d \ mod\ N$. Para verificação, basta o receptor receber m e σ, calcular $h = \sigma^e \ mod\ N$ e $h' = H(m)$, e fazer a comparação $h \stackrel{?}{=} h'$ para decidir se a assinatura é válida ou não. Note que o sistema é seguro porque fatorar N é difícil, mas se já se conhece p e q, todo o restante se torna trivial.

Diversos outros esquemas assimétricos foram desenvolvidos a partir das outras abordagens de escolha do problema. A abordagem do logaritmo resultou em algoritmos como ElGamal e DSA. Sua variação em curvas elípticas (ECC - *Elliptic Curve Cryptography*), fundamentou esquemas como o ECDSA (*Elliptic Curve Digital Signature Algorithm*) e ECDH (*Elliptic Curve Diffie–Hellman*), que são utilizados em protocolos

modernos como TLS (*Transport Layer Security*). Seus fundamentos são similares, mas apresentam diferenças de eficiência em cenários diferentes.

Usualmente, sistemas assimétricos são mais complexos em suas formulações quando comparados com sistemas simétricos. Enquanto os últimos apresentam operações eficientes (como XOR, substituições e permutações), sistemas assimétricos precisam garantir o artifício seguro de correlação entre as chaves, utilizando operações como exponenciação, custosas computacionalmente [Terada 2008]. Dessa forma, em termos práticos, a criptografia assimétrica não é utilizada para processar grandes volumes de dados. Esquemas assimétricos são combinados com sistemas simétricos, assumindo o papel de Mecanismos de Encapsulamento de Chave (KEM - *Key Encapsulation Mechanism*), responsáveis pela entrega da chave simétrica entre as partes [Katz and Lindell 2014].

A segurança dos algoritmos assimétricos está ligada de forma fundamental à dificuldade desses problemas em condições adversariais, ou seja, sem a posse da chave secreta. Isso significa que a segurança dos sistemas assimétricos toma a inviabilidade computacional da resolução dos problemas em tempo hábil como pressuposto. Se o problema computacional é resolvido em tempo hábil, o sistema é quebrado em sua premissa fundamental [Menezes et al. 2018].

3.3. Introdução à Computação Quântica

Esta seção apresenta uma visão geral da computação quântica, noções básicas de mecânica quântica, análise de algoritmos e complexidade computacional, além de conceitos sobre *qubits* e seus estados, portas quânticas, circuitos quânticos e como a informação pode ser representada e manipulada através dessas estruturas [Nielsen and Chuang 2010]. Serão exercitados circuitos e algoritmos quânticos utilizando ferramentas da IBM (exemplos disponíveis em `https://github.com/vthayashi/quantum-crypto`).

3.3.1. Conceitos Básicos

Na computação tradicional, agora chamada de clássica para a diferenciar da quântica, utilizamos algoritmos clássicos para resolver problemas computacionais. Esses algoritmos, ao serem implementados como programas e suportados por diversas camadas de abstração, modificarão os estados dos *bits* até obtermos uma possível solução para o problema. O *bit* é a unidade básica de informação clássica, podendo representar em determinado instante apenas um de dois valores possíveis, usualmente denotados como 0 e 1. Isso constitui uma abstração, pois naturalmente o *bit* não possui existência física: ele precisa ser implementado por meio de algum sistema físico específico. A forma predominante de se fazer isso na computação clássica é utilizando o transistor em conjunto com outros dispositivos eletrônicos. O transistor, apesar de ser construído a partir de semicondutores cuja descrição e entendimento completo só é possível com a mecânica quântica, opera sob um regime clássico e podemos descrevê-lo e utilizá-lo valendo-se de fenômenos clássicos como corrente elétrica, tensão elétrica, resistência elétrica, entre outros, conforme estabelecidos pela teoria eletromagnética.

Por outro lado, a computação quântica constitui-se através da combinação da ciência da computação com a mecânica quântica, tendo por objetivo construir sistemas computacionais que possam explorar fenômenos quânticos como superposição, emara-

nhamento e interferência para realizar computações úteis. Abaixo, temos uma breve intuição sobre cada um desses fenômenos, considerados principais dentro da computação quântica. Entretanto, como nosso foco será na parte computacional, uma introdução técnica à mecânica quântica pode ser encontrada em [Susskind and Friedman 2014] e uma exposição mais avançada em [Sakurai and Napolitano 2020]:

- **Superposição:** possibilidade de um sistema quântico ser descrito através da combinação de múltiplos estados (ou configurações) durante sua operação. Assim, para um sistema de dois níveis, além de poder assumir estados bem definidos, como 0 ou 1, pode apresentar configurações que são combinações desses dois estados de acordo com uma certa distribuição de probabilidades.

- **Emaranhamento:** um tipo especial de superposição envolvendo os estados de dois ou mais sistemas quânticos, de forma a não ser mais possível descrever cada sistema de forma independente dos demais, criando correlações. Assim, o sistema completo não pode ser totalmente descrito em termos de cada parte individual.

- **Interferência:** quando o estado de dois ou mais sistemas quânticos são combinados para formar um novo estado composto, podemos ter interações entre eles, com cada sistema influenciando o outro, que amplificam a probabilidade associada a certos estados, chamada de interferência construtiva, ou diminuem/eliminam a de outros, chamada de interferência destrutiva.

Para fazer a exploração desses fenômenos para computar, construímos dispositivos baseados em sistemas quânticos diversos (e.g., átomos, íons, elétrons, fótons), além de outros componentes, com comportamentos distintos do transistor (há diversas abordagens para se construir computadores quânticos, porém este tópico está além do escopo deste trabalho; informações podem ser obtidas em [Kasirajan 2021]). A partir desses dispositivos, operando como um sistema de dois níveis, emerge uma abstração chamada *qubit (quantum bit)*, que é a unidade básica de informação quântica, podendo representar em determinado instante tanto estados bem definidos como 0 e 1 quanto estados que são superposições deles. Com a computação quântica podemos projetar novos algoritmos que são fundamentalmente distintos dos algoritmos clássicos. Os *qubits* serão então manipulados pelos algoritmos quânticos de forma a resolver problemas computacionais de uma maneira diferente. Ao mudarmos os fenômenos físicos que alicerçam a computação, abrimos novas possibilidades, o que pesquisas já realizadas nas últimas décadas nos mostraram que permite à computação quântica resolver certos problemas computacionais de forma mais eficiente do que a computação clássica [Shafique et al. 2024].

3.3.2. Análise de Algoritmos e Complexidade Computacional

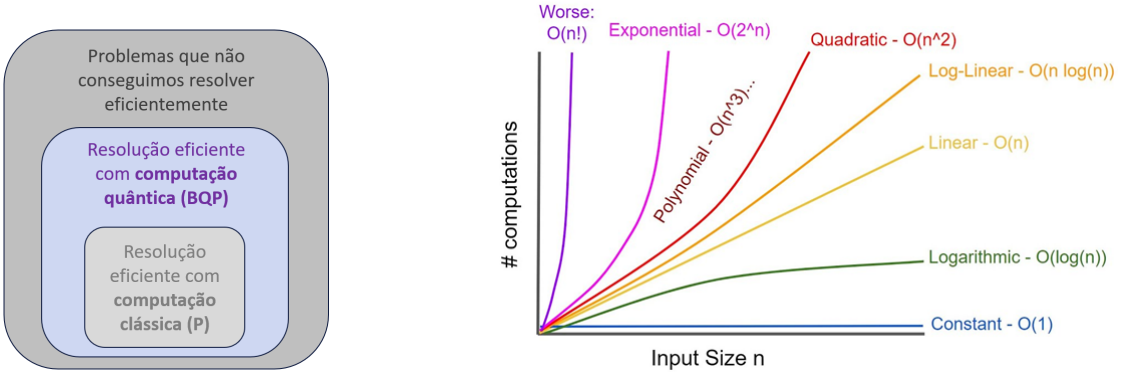
Ao falarmos sobre eficiência de algoritmos, estamos nos referindo ao seu padrão de consumo de recursos computacionais em função do tamanho do problema que se deseja resolver, uma análise matemática que é possível realizar e que nos permite classificá-los em termos de seu comportamento. Importante frisar a diferença dessa ideia para a de desempenho de programas, ou seja, das implementações de algoritmos utilizando determinadas

linguagens de programação, compiladores, sistemas operacionais, até sua conversão completa para a linguagem de máquina a ser executada pelo hardware (ou uma implementação diretamente em hardware), fatores estes que podem impactar negativamente o comportamento de um algoritmo na prática e serem mitigadores de desempenho. Ao longo da discussão sobre o potencial da computação quântica, seja neste texto ou na literatura em geral, muito se fala em termos da eficiência dos algoritmos quânticos conhecidos mesmo que ainda não existam computadores quânticos de larga escala e tolerantes a falhas para implementá-los e concretizar esse potencial.

Dessa forma, um dos objetivos durante a análise de algoritmos é estimar a quantidade de recursos necessários para resolver determinado problema conforme o tamanho da entrada varia. Também de interesse é a análise comparativa de algoritmos, onde dados ao menos dois algoritmos diferentes, identificamos qual deles é mais eficiente para resolver determinado problema. Esse estudo dos recursos necessários para executar um algoritmo pode empregar como métricas recursos temporais (e.g., número de passos computacionais ou de operações básicas necessárias) e espaciais (e.g., a quantidade de memória). A medida exata do tempo de processamento ou de utilização de memória só será possível após a implementação e execução do algoritmo em um computador real. A partir daqui, o texto irá priorizar a métrica de tempo, chamada de complexidade temporal, e por meio de uma análise matemática formal do algoritmo, tentaremos encontrar uma função matemática $f(n)$ que represente seu consumo de recursos em relação ao tamanho n da entrada.

A análise assintótica é uma ferramenta matemática que nos permite determinar essa função e tirar conclusões sobre o comportamento do algoritmo, contando o número de passos computacionais a serem executados conforme n varia. Essa análise não depende de detalhes de implementação, nos dando um indicativo sobre o comportamento desse algoritmo quando n se torna muito grande. Existem 3 notações básicas para análise assintótica de funções: Big-O, Big-Ω e Big-Θ. Vamos defini-las brevemente, porém nosso uso neste minicurso se limitará à notação Big-O. Outras notações e aprofundamentos teóricos podem ser encontrados em [Sipser 2012]:

- **Big-O**: representa o limite superior para o consumo de recursos de um algoritmo. Geralmente utilizada para estudar o comportamento do algoritmo no pior caso. Formalmente, dizemos que $f(n) = O(g(n))$ se tivermos constantes $c > 0$ e $n_0 > 0$ tais que:
$$f(n) \leq cg(n) \quad \forall n \geq n_0 \tag{2}$$
- **Big-Ω**: representa o limite inferior para o consumo de recursos de um algoritmo. Geralmente utilizada para estudar o comportamento do algoritmo no melhor caso. Formalmente, dizemos que $f(n) = \Omega(g(n))$ se tivermos constantes $c > 0$ e $n_0 > 0$ tais que:
$$f(n) \geq cg(n) \quad \forall n \geq n_0 \tag{3}$$
- **Big-Θ**: representa simultaneamente os limites inferior e superior para o consumo de recursos de um algoritmo. Geralmente utilizada para estudar o comportamento no caso médio. Formalmente, dizemos que $f(x) = \Theta(g(n))$ se tivermos constantes $c > 0$, $d > 0$ e $n_0 > 0$ tais que:
$$cg(n) \leq f(x) \leq dg(n) \quad \forall n \geq n_0 \tag{4}$$



(a) Classificação simplificada dos problemas computacionais.

(b) Exemplos de funções em notação Big-O. Fonte: [Salvi 2023]

Figura 3.6. Classificação de problemas e complexidade computacional

Uma das propriedades interessantes da notação Big-O é que ao considerar duas funções tais que $f(n) = O(g(n))$ e $x(n) = O(y(n))$, então teremos que $f(n) + x(n) = O(max\{g(n), y(n)\})$. A mesma ideia se aplica para um número arbitrário de funções. Além disso, essa notação nos indica a ordem de magnitude dessa função, ou seja, o termo dominante, para representar o que acontece com a função $f(n)$ quando n cresce de forma arbitrária. Por exemplo, se para determinado algoritmo A_1 tivermos $f_1(n) = n+1$ diremos que esse algoritmo é $O(n)$ ou também que $f_1(n) = O(n)$. No caso de um algoritmo A_2 ter $f_2(n) = n^2 + 4n + 3$ diremos que $f_2(n) = O(n^2)$. Comparando os dois, concluímos que o algoritmo A_1 é o mais eficiente, pois sua função de consumo de recursos cresce mais lentamente (é possível verificar isso através do cálculo de limites e derivadas). Como a notação Big-O denota um limite superior, não importa qual o computador real utilizado, o consumo de recursos de um determinado algoritmo nunca será melhor do que esse limite para valores de n muito grandes. A mesma ideia vale para a comparação de algoritmos: não importa o contexto prático de implementação, dado que ambos são executados sob as mesmas condições, o algoritmo A_1 será mais rápido do que A_2 para n suficientemente grande, que é o caso assintótico. Naturalmente, para valores pequenos podem até ocorrer variações.

Sabemos que a computação clássica pode resolver certos tipos de problemas de forma eficiente (classe de problemas P), enquanto a computação quântica expande essa possibilidade (classe de problemas BQP) abarcando alguns problemas considerados insolúveis por computadores clássicos, como a fatoração de números inteiros [Shor 1997]. Entretanto, vale frisar que a computação quântica não vai resolver todos os problemas considerados intratáveis atualmente. A Figura 3.6(a) representa de maneira simplificada as relações citadas acima, sendo uma interpretação lúdica dos diagramas de classes de complexidade computacional e a Figura 3.6(b) mostra alguns dos principais tipos de funções que representam o padrão de consumo de recursos, em notação Big-O, utilizadas na análise de algoritmos.

Por fim, quando falamos sobre resolução eficiente de um problema computacional, estamos nos referindo a algoritmos cuja complexidade computacional é polinomial ou melhor (o consumo de recursos cresce mais lentamente), enquanto as demais, como a

exponencial, consideramos ineficientes [Sipser 2012]. Naturalmente, um polinômio com grau muito alto não pode ser considerado eficiente na prática. Assim, mesmo um algoritmo com complexidade polinomial pode resultar num desempenho inadequado após ser implementado dependendo do tamanho do problema, sendo preferidos aqueles que apresentem complexidades constante, logarítmica ou linear. Porém, restringiremos nossa discussão à distinção entre eficientes e ineficientes em termos de complexidade computacional. Além disso, qualquer problema computacional para o qual se conhece um algoritmo pode ser fácil de resolver se o tamanho da entrada for suficientemente pequeno. As diferenças se tornam mais significativas conforme n cresce rumo ao regime assintótico, revelando a verdadeira natureza do comportamento do algoritmo.

3.3.3. Estados, Portas e Circuitos Quânticos

Nesta subseção, apresentamos uma breve introdução ao formalismo matemático e conceitual da computação quântica. Para aprofundamentos e exposições mais gerais, consultar os trabalhos de [Nielsen and Chuang 2010, Watrous 2025].

Utilizamos a notação de Dirac [Watrous 2025] para representar matematicamente os estados quânticos e seus operadores na computação quântica. Um vetor ψ escrito como $|\psi\rangle$ é chamado de *ket* e escrito como $\langle\psi|$ é chamado de *bra* nessa notação. Podemos representar um vetor na forma de coluna, correspondente ao *ket*, ou na forma de linha, corresponde ao *bra*, conforme exemplificado abaixo para dois vetores **v** e **u**:

$$|v\rangle = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} \qquad \langle u| = \begin{pmatrix} u_1 & u_2 & \cdots & u_n \end{pmatrix} \tag{5}$$

Na computação quântica, usualmente temos dois estados que representam quanticamente nossos conhecidos valores 0 e 1, agora chamados de $|0\rangle$ e $|1\rangle$ (dois estados ortogonais e normalizamos que juntos formam o que chamamos de base computacional, que será nossa referência para descrevermos os estados e resultados durante a execução de circuitos quânticos). Na maior parte do tempo, vamos utilizar os *kets*, com os *bras* aparecendo apenas quando necessários em determinadas operações. Dessa forma, os vetores coluna que representam esses estados da base computacional são definidos como:

$$|0\rangle := \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle := \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{6}$$

Representaremos os estados quânticos utilizando vetores e os operadores, que modificam tais estados, por meio de matrizes. Essa formulação é suficiente para compreendermos as bases da área e a maior parte dos algoritmos quânticos. Vale notar que há uma descrição mais geral do que essa, utilizada na teoria quântica da informação, que faz uso de matrizes densidade para descrever os estados quânticos, ferramental necessário quando desejamos modelar o efeito do ruído das interações dos nossos sistemas com o ambiente a sua volta [Watrous 2025]. Porém, essa abordagem generalizada não faz parte do escopo deste texto, pois não é necessária para nossos propósitos. Assim, o estado quântico de

um *qubit* pode ser então representado de maneira geral como uma combinação linear (ou superposição) dos estados de uma base, neste exemplo a base computacional:

$$|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha|0\rangle + \beta|1\rangle \quad , \quad \alpha, \beta \in \mathbb{C} \tag{7}$$

Os números complexos α e β são chamados de amplitudes de probabilidade e é a partir deles que conseguimos construir superposições, explorar interferências e calcular as probabilidades de se obter cada estado da base após o processo de medição. Esse processo nos permite extrair o resultado de uma computação a partir da interação entre um equipamento de medida e o computador quântico de maneira a determinar em qual configuração ele se encontra e associá-la a um determinado estado quântico de referência. A partir desse resultado, podemos associar o estado identificado com uma sequência de *bits* que representam em binário a saída do computador quântico para determinada computação. Dessa forma, o resultado obtido após a medição sempre será um valor específico, nunca obteremos superposições, e a probabilidade de o obtermos estará de acordo com a respectiva amplitude de probabilidade. Naturalmente, só conseguiremos visualizar experimentalmente tal probabilidade após repetir a execução diversas vezes. Após a medição, o estado do qubit se torna bem determinado e será o mesmo para medições subsequentes, a não ser que o modifiquemos deliberadamente, que alteremos a configuração do aparato de medida ou que ruídos no sistema causem alguma mudança inadvertida. Para obter a probabilidade de medir cada dos estados da base de referência, calcula-se o módulo de cada número complexo e eleva-se o resultado ao quadrado. Pela Regra de Born [Watrous 2025], que nos fornece uma condição de normalização, devemos ter todas essas probabilidades somando para a unidade conforme a relação $|\alpha|^2 + |\beta|^2 = 1$.

As mesmas ideias se aplicam para sistemas formados por múltiplos *qubits*, com seu estado e condição de normalização sendo representados como (para *n qubits* teremos no máximo 2^n estados na superposição):

$$|\psi\rangle = \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{2^n-1} \end{pmatrix} = \sum_{j=0}^{2^n-1} c_j|j\rangle \quad , \quad \sum_{j=0}^{2^n-1} |c_j|^2 = 1 \quad , \quad c_j \in \mathbb{C} \tag{8}$$

Em sistemas de múltiplos *qubits*, precisamos definir a ordenação que será utilizada para eles. No nosso caso, os *qubits* menos significativos serão representados por índices menores. Vamos também ordená-los da esquerda para a direita em ordem decrescente, ou seja, se tivermos um estado $|011\rangle$ fica subentendido que ele representa o estado composto dos *qubits* $q_2q_1q_0$, com q_2 sendo o qubit mais significativo e q_0 sendo o menos significativo. Disso decorre uma outra possibilidade de representação, conforme utilizado na Equação 8, que é substituir a notação binária pela decimal dentro dos *kets*. Assim, o estado $|011\rangle$ se torna $|3\rangle$ e o estado $|111\rangle$ se torna $|7\rangle$. Para evitar confusões, caso não esteja claro pelo contexto do uso, um subscrito pode ser utilizado para informar a quantos *qubits* correspondem essa notação decimal, ou seja, $|7\rangle_3$ nos indica o estado $|111\rangle$ e $|7\rangle_5$ o estado $|00111\rangle$.

Cada amplitude de probabilidade complexa pode também ser escrita como $c_j = r_j e^{i\theta_j}$, onde r_j é uma amplitude real, com $r_j > 0$, e θ_j é uma fase com $0 \leq \theta_j < 2\pi$. Assim, podemos também representar o estado de um sistema de múltiplos *qubits* como:

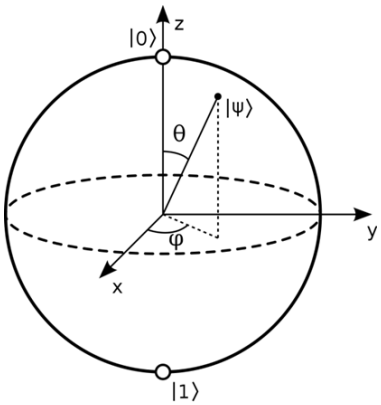
$$|\psi\rangle = \sum_{j=0}^{2^n-1} r_j e^{i\theta_j} |j\rangle \quad , \quad \sum_{j=0}^{2^n-1} r_j^2 = 1 \quad , \quad r_j, \theta_j \in \mathbb{R} \tag{9}$$

A partir dessas descrições e através de algumas manipulações algébricas, podemos representar o estado de um qubit por meio de coordenadas polares usando números reais e visualizá-lo geometricamente na chamada esfera de Bloch [Zhang et al. 2011]:

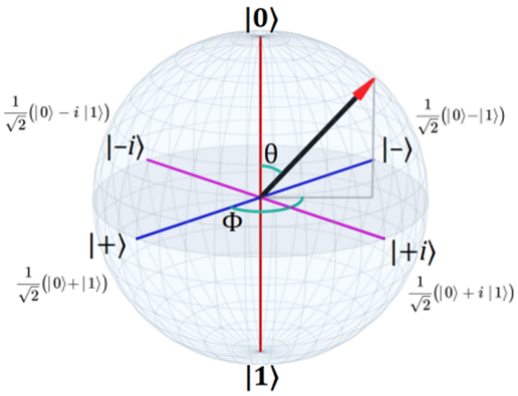
$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle \quad , \quad \theta, \phi \in \mathbb{R} \tag{10}$$

A Figura 3.7(a) mostra a esfera de Bloch, uma representação geométrica útil para o estado de um qubit, com $0 \leq \theta \leq \pi$ e $0 \leq \phi < 2\pi$, que nos permite mapear um vetor de $\mathbb{C}^2$ para $\mathbb{R}^3$. As operações codificadas pelas portas quânticas podem ser vistas como rotações em relação aos eixos dessa esfera [Kosmann-Schwarzbach and Singer 2010]. Nos seus polos temos os estados $|0\rangle$ e $|1\rangle$ da base computacional (por se localizarem sobre o eixo z, essa base muitas vezes também é chamada de base Z), sendo uma escolha arbitrária, porém de uso comum. Os dois ângulos θ e ϕ determinam a localização do vetor de estado. Para estados apresentados nos próximos exemplos, eles vão se localizar na superfície da esfera, porém é possível construir estados que se localizem em qualquer região interna dela. O ângulo ϕ é chamado de fase relativa e contém a diferença de fase entre os números complexos β e α. O fator $e^{i\phi}$ pode ser relacionado com senos e cossenos através da Fórmula de Euler, sendo i a unidade imaginária:

$$e^{i\phi} = cos(\phi) + isen(\phi) \tag{11}$$



(a) Esfera de Bloch. Fonte: [Smite-Meister 2023]



(b) Estados notáveis na Esfera de Bloch. Fonte: [Smythe 2021]

Figura 3.7. Representação de estados na Esfera de Bloch

Em relação aos estados da base computacional, a não ser pelos polos, qualquer outro ponto da superfície da esfera irá representar uma superposição deles. No equador

teremos as chamadas superposições uniformes, nas quais as probabilidades associadas a cada um desses estados será a mesma. Alguns estados notáveis que representam superposições uniformes de estados da base computacional e recebem nomes especiais são mostrados na Figura 3.7(b) e na equação abaixo:

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix} \qquad |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}$$
$$|+i\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle) = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\i\end{pmatrix} \qquad |-i\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle) = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-i\end{pmatrix} \tag{12}$$

A partir da expressão para o estado quântico utilizando coordenadas polares, conseguimos representar tais estados na esfera com a fase relativa determinando a posição do vetor de estado em relação ao eixo x (além do uso de seu nome).

Para modificar os estados dos *qubits* ao longo do tempo precisamos de operadores. Aqueles de interesse para a computação quântica são os chamados unitários, que preservam a norma unitária dos vetores de estado, de acordo com a Regra de Born. Por serem aplicados aos estados dos *qubits*, essas matrizes serão quadradas e com dimensão $2^n \times 2^n$ com n representando o número de *qubits*:

$$U = \begin{pmatrix} u_{11} & \cdots & u_{12^n} \\ \vdots & \ddots & \vdots \\ u_{2^n 1} & \cdots & u_{2^n 2^n} \end{pmatrix} \tag{13}$$

Para aplicar o operador a um determinado estado quântico, basta realizar a multiplicação padrão de matrizes, obtendo assim um novo estado:

$$|\psi'\rangle = U|\psi\rangle \tag{14}$$

Operadores unitários apresentam propriedades importantes. Por exemplo, sua matriz inversa, quando existe, é igual a sua matriz transposta conjugada ($U^{-1} = U^\dagger$). Ou seja, para inverter o efeito de um operador, ao invés de precisar calcular matrizes inversas, algo que se torna mais custoso conforme as dimensões da matriz aumentam, basta aplicar o operador transposto conjugado. A relação abaixo mostra essa ideia, onde um operador multiplicado pelo seu inverso resulta na matriz identidade.

$$UU^{-1} = U^{-1}U = I \quad \rightarrow \quad UU^\dagger = U^\dagger U = I \tag{15}$$

Dentro do conjunto dos operadores unitários temos aqueles que também são hermitianos, ou seja, que são iguais a sua conjugada transposta. Nesse caso, para realizar a operação inversa de um operador hermitiano, basta aplicar o operador novamente. Acumulando as duas propriedades, temos $U = U^\dagger = U^{-1}$. Os operadores serão representados a partir daqui como portas lógicas quânticas (que são a base para computadores

quânticos de propósito geral; para outras abordagens de propósito específico, consultar [Kasirajan 2021]) e a implicação de ter um determinado operador unitário e hermitiano é que para reverter o efeito de uma determinada porta desse tipo basta aplicá-la novamente, enquanto para as demais portas quânticas que não são hermitianas, precisamos aplicar portas quânticas diferentes que representarão os respectivos operadores transpostos conjugados.

Para representarmos estados e combinarmos operadores envolvendo múltiplos *qubits*, precisamos de estruturas matemáticas chamadas tensores, que são generalizações de estruturas como escalares, vetores e matrizes. Um escalar (real ou complexo) é entendido como um tensor de ordem 0, um vetor como um tensor de ordem 1 e uma matriz como um tensor de ordem 2. Para representarmos o produto tensorial entre dois tensores A e B, utilizaremos a notação $A \otimes B$. De forma geral, nos restringindo a vetores e matrizes, para dois deles de dimensões arbitrárias ($n \times m$ e $r \times s$) seu produto tensorial será dado pela seguinte expressão:

$$A \otimes B = \begin{pmatrix} A_{11}\begin{pmatrix} B_{11} & \cdots & B_{1s} \\ \vdots & \ddots & \vdots \\ B_{r1} & \cdots & B_{rs} \end{pmatrix} & \cdots & A_{1m}\begin{pmatrix} B_{11} & \cdots & B_{1s} \\ \vdots & \ddots & \vdots \\ B_{r1} & \cdots & B_{rs} \end{pmatrix} \\ \vdots & \ddots & \vdots \\ A_{n1}\begin{pmatrix} B_{11} & \cdots & B_{1s} \\ \vdots & \ddots & \vdots \\ B_{r1} & \cdots & B_{rs} \end{pmatrix} & \cdots & A_{nm}\begin{pmatrix} B_{11} & \cdots & B_{1s} \\ \vdots & \ddots & \vdots \\ B_{r1} & \cdots & B_{rs} \end{pmatrix} \end{pmatrix} \tag{16}$$

Anteriormente, definimos a base computacional para um qubit como sendo composta pelos estados $|0\rangle$ e $|1\rangle$. Podemos obter a base computacional para sistemas com 2 ou mais *qubits* utilizando o produto tensorial. Para o caso de 2 *qubits*, teremos $2^2 = 4$ estados possíveis de serem representados e podemos obtê-los através da combinação dos estados da base computacional de um qubit conforme expressões abaixo:

$$|00\rangle = |0\rangle \otimes |0\rangle = \begin{pmatrix} 1\begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ 0\begin{pmatrix} 1 \\ 0 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad |01\rangle = |0\rangle \otimes |1\rangle = \begin{pmatrix} 1\begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ 0\begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$|10\rangle = |1\rangle \otimes |0\rangle = \begin{pmatrix} 0\begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ 1\begin{pmatrix} 1 \\ 0 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \qquad |11\rangle = |1\rangle \otimes |1\rangle = \begin{pmatrix} 0\begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ 1\begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{17}$$

A representação de circuitos quânticos segue o formato representado na Figura 3.8. As linhas simples representam *qubits*, nomeados q_i, enquanto as linhas duplas re-

presentam *bits*, no caso mostrados de forma coletiva como um registrador de três *bits*, nomeado *c*, que guardam os resultados das medições que são indicadas pelo símbolo de um medidor. As demais caixas com símbolos representam portas lógicas quânticas, responsáveis por modificar os estados dos *qubits* e implementar as operações desejadas (as diferentes cores não são relevantes para nossa análise). As portas mais simples geralmente atuam sobre um ou dois *qubits*, mas podemos ter outras portas mais complexas construídas a partir das mais simples e que operam sobre um maior número deles. A seguir, veremos alguns exemplos, frisando apenas que há diversas outras portas bastante conhecidas, mas que não farão parte da exposição.

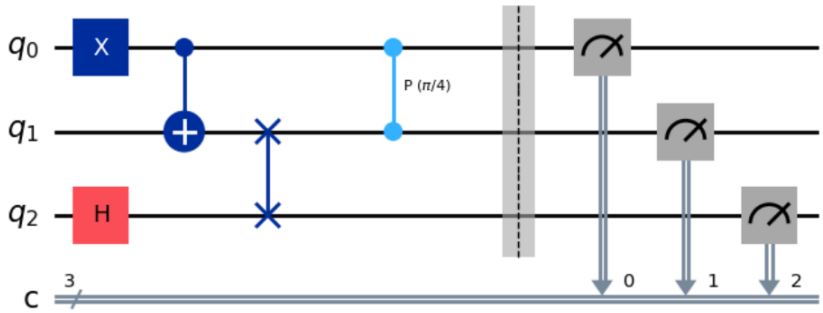


Figura 3.8. Exemplo de Circuito Quântico

Uma das portas quânticas mais importantes é a Hadamard, representada pela letra H. Ela tem por efeito criar superposições uniformes se for aplicada a estados bem definidos como os da base computacional e modificar superposições se for aplicada a estados quaisquer. Sua matriz correspondente é apresentada na Equação 18 e os resultados de sua aplicação aos estados $|0\rangle$ e $|1\rangle$ nas Equações 19 e 20, respectivamente:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{18}$$

$$H|0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle \tag{19}$$

$$H|1\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle \tag{20}$$

Equivalentemente, podemos visualizar essa mudança através da representação do estado do qubit na esfera de Bloch conforme Figura 3.9.

As próximas três portas que veremos são chamadas de portas de Pauli e são representadas pelas letras X, Y e Z, aplicando aos estados quânticos rotações de π radianos em torno dos eixos x, y e z da esfera de Bloch, respectivamente. Em relação à base computacional, a porta X tem por efeito trocar os estados da base computacional entre si ($|0\rangle \rightarrow |1\rangle$, $|1\rangle \rightarrow |0\rangle$), também chamado de *bit-flip*. A porta Z tem por efeito aplicar uma troca de sinal na fase do estado $|1\rangle$, também chamado de *phase-flip*. Por fim, a porta Y

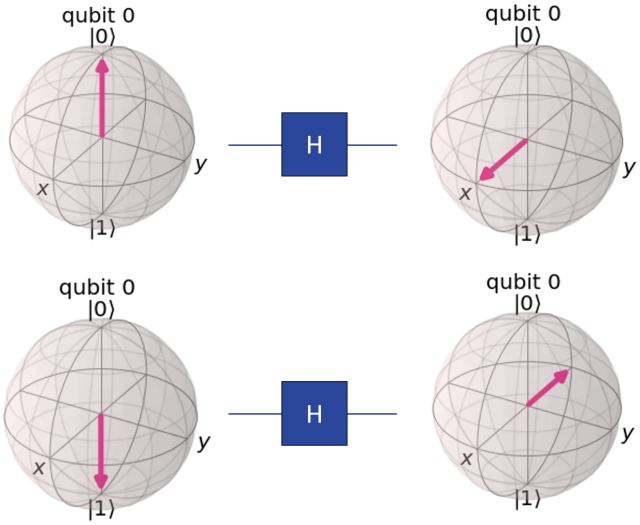


Figura 3.9. Porta H e seu efeito visualizado na esfera de Bloch

equivale a combinar os dois efeitos anteriores, realizando um *bit-flip* e um *phase-flip*. As matrizes correspondentes a essas portas são apresentadas na Equação 21 e suas representações gráficas na Figura 3.10, mostrando os resultados de sua aplicação aos estados $|0\rangle$ e $|1\rangle$.

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad , \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad , \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{21}$$

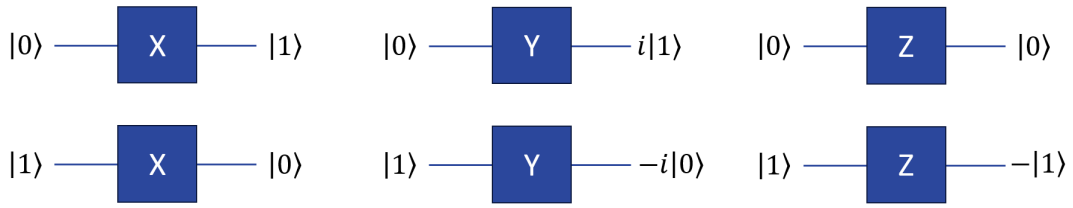


Figura 3.10. Portas de Pauli e seus efeitos sobre a base computacional

O efeito da porta X pode ser visto como o equivalente quântico da porta clássica NOT, sendo também representado como $X|j\rangle = |j \oplus 1\rangle$, com o símbolo $\oplus$ denotando a operação XOR ou adição módulo 2. Disso decorre que a porta X também é representada em muitos locais com esse símbolo. Ao ser aplicada a uma superposição, por linearidade, basta aplicá-la a cada um dos estados constituintes (mesma ideia para as demais portas). Nesse caso, a operação então se torna equivalente a trocarmos as amplitudes de probabilidade:

$$X(\alpha|0\rangle + \beta|1\rangle) = \alpha X|0\rangle + \beta X|1\rangle = \alpha|1\rangle + \beta|0\rangle = \beta|0\rangle + \alpha|1\rangle \tag{22}$$

As próximas três portas quânticas (SWAP, CX e CP) são bastante presentes em circuitos quânticos e operam sobre dois *qubits*. Suas matrizes correspondentes são:

$$SWAP = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, CX_{q_0,q_1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, CP(\phi) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\phi} \end{pmatrix} \quad (23)$$

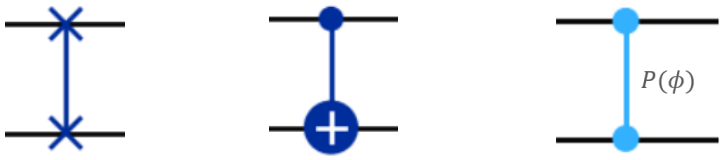
A porta SWAP tem por efeito trocar os estados dos dois *qubits* envolvidos na operação. Sua representação gráfica é apresentada na Figura 3.11(a) e os resultados de sua aplicação aos estados da base computacional de 2 *qubits* são mostrados abaixo:

$$SWAP|00\rangle = |00\rangle \;\; , \;\; SWAP|01\rangle = |10\rangle \;\; , \;\; SWAP|10\rangle = |01\rangle \;\; , \;\; SWAP|11\rangle = |11\rangle \quad (24)$$

Na Figura 3.11(b) temos a porta CX, que possui um qubit de controle (círculo menor) e um qubit alvo (círculo maior, representando a porta X). Quando o controle estiver no estado $|1\rangle$, a porta é ativada e o alvo receberá a aplicação da porta X, caso contrário nada acontece, ou seja, é a aplicação condicional da porta X ao alvo. O seu efeito nos estados da base computacional de 2 *qubits* são mostrados na Equação 25. O subscrito indica qual qubit controla (q_0) e qual é alvo (q_1). Dependendo da escolha, a matriz será diferente, porém com mesmo funcionamento geral.

$$CX_{q_0,q_1}|00\rangle = |00\rangle \;\; , \;\; CX_{q_0,q_1}|01\rangle = |11\rangle \;\; , \;\; CX_{q_0,q_1}|10\rangle = |10\rangle \;\; , \;\; CX_{q_0,q_1}|11\rangle = |01\rangle \quad (25)$$

A combinação de portas CX, H e X nos permite construir estados emaranhados, especialmente aqueles chamados de estados de Bell [Watrous 2025], essenciais em diversas tarefas na computação quântica e na comunicação quântica.



(a) Porta SWAP (b) Porta CX (c) Porta CP

Figura 3.11. Representação gráfica das portas SWAP, CX e CP

Por fim, a porta CP permite rotacionar o estado de um qubit de forma parametrizada em torno do eixo z, aplicando uma fase ao estado do qubit alvo. Quando o controle estiver no estado $|1\rangle$, a porta é ativada e o alvo receberá a fase, caso contrário nada acontece (a Figura 3.11(c) mostra a representação gráfica dessa porta). Porém, por motivos que ficarão claros na subseção 3.4.1.4, seu símbolo não faz uma distinção clara entre controle e alvo. Ao lado dele, aparece a parametrização indicada como $P(\phi)$ visto que a porta CP é a versão controlada da porta P (de único qubit cuja aplicação da fase sempre acontece;

não detalhada aqui). Um exemplo de aplicação da porta é mostrado abaixo para o estado $|q_1 q_0\rangle = |+1\rangle = |+\rangle \otimes |1\rangle$:

$$CP(\pi/4)|+1\rangle = CP(\pi/4)(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |1\rangle) = \frac{1}{\sqrt{2}}(|0\rangle + e^{i\pi/4}|1\rangle) \otimes |1\rangle \tag{26}$$

Agora que aprendemos algumas portas quânticas fundamentais, vamos exercitar a aplicação desses conceitos para identificar qual o estado final do circuito da Figura 3.8 antes da medição. Por convenção, temos o estado inicial $|q_2 q_1 q_0\rangle = |000\rangle$ e por comodidade vamos usar índices em todas as portas para nos dizer sobre quais *qubits* elas estão aplicadas:

$$\begin{aligned} |000\rangle \xrightarrow{X_{q_0}} |001\rangle \xrightarrow{H_{q_2}} \frac{1}{\sqrt{2}}(|001\rangle + |101\rangle) \xrightarrow{CX_{q_0,q_1}} \frac{1}{\sqrt{2}}(|011\rangle + |111\rangle) \\ \xrightarrow{SWAP_{q_1,q_2}} \frac{1}{\sqrt{2}}(|101\rangle + |111\rangle) \xrightarrow{CP(\pi/4)_{q_0,q_1}} \frac{1}{\sqrt{2}}(|101\rangle + e^{i\pi/4}|111\rangle) \end{aligned} \tag{27}$$

Se realizarmos a medição desse estado, na ausência de erros quaisquer no processo, obteremos o estado $|101\rangle$ com probabilidade $1/2$ (basta usar a Regra de Born) e podemos associá-lo à cadeia de *bits* 101, guardando esse valor no registrador clássico. Equivalentemente, poderíamos obter o estado $|111\rangle$ também com probabilidade $1/2$ e associá-lo à cadeia de *bits* 111 (o fator $e^{i\pi/4}$ não influencia essa probabilidade visto que $|e^{i\pi/4}| = 1$). Podemos então reconfigurar nosso circuito e executá-lo novamente para que tenhamos amostras suficientes para reconstruir a distribuição de probabilidades associada ao estado do sistema. Perceba que mesmo exemplos simples podem ser desafiadores quando se fala de computação quântica, pois a lógica subjacente a esse modelo de computação é diferente da que estamos acostumados na computação clássica. Mesmo o entendimento de circuitos básicos e dos algoritmos existentes constitui tarefa complexa e pensar em novas soluções, melhorando ou até mesmo criando novos algoritmos, é um desafio não trivial.

3.4. Impactos em Criptografia e Soluções

Esta seção busca expor como algoritmos quânticos afetam a segurança dos sistemas criptográficos atuais (RSA, ECC, AES), destacando as justificativas para os diferentes níveis de impactos em mecanismos de criptografia simétrica e assimétrica. Esta seção também apresenta a criptografia pós-quântica (PQC) [Alagic et al. 2022] e quântica (QKD) como soluções em desenvolvimento [Mosca 2018], e um estudo de caso de impactos da computação quântica ao Bitcoin. Serão explorados exemplos práticos de uso de uma biblioteca de Criptografia Pós-Quântica (exemplos disponíveis em `https://github.com/vthayashi/quantum-crypto`).

3.4.1. Impactos dos Algoritmos de Grover e Shor

Os dois algoritmos quânticos mais conhecidos são os de Shor [Shor 1994] e de Grover [Grover 1996]. O algoritmo de Shor foi desenvolvido pelo matemático Peter Shor (na realidade ele propôs um conjunto de algoritmos, porém é de uso comum se referir a esse conjunto apenas como algoritmo de Shor) em 1994 e pode ser utilizado para resolver eficientemente problemas como a fatoração de números inteiros e o cálculo de logaritmos discretos, apresentando uma aceleração exponencial se comparado ao melhor algoritmo clássico conhecido para o mesmo problema. Lembrando que, para falarmos sobre aceleração em termos computacionais, comparamos a complexidade computacional para o mesmo problema entre ao menos dois algoritmos e analisamos suas diferenças.

O algoritmo de Grover foi proposto pelo cientista da computação Lov Grover em 1996 e pode ser utilizado para realizar buscas num conjunto de dados não estruturado, ou seja, um conjunto que não está ordenado ou possui qualquer tipo de atalho para encontrar o item desejado. Nesse caso, o algoritmo apresenta uma aceleração quadrática se comparado ao melhor algoritmo clássico para o mesmo problema. Ao combinar um computador quântico de larga escala e tolerante a falhas, também chamado de Computador Quântico Criptograficamente Relevante (CRQC - *Cryptographically Relevant Quantum Computer*) [NSA 2021] neste contexto de criptografia, com os algoritmos de Shor e de Grover, pode-se causar impactos substanciais em boa parte da criptografia utilizada atualmente, conforme descrito na próxima subseção.

3.4.1.1. Impactos do Algoritmo de Grover na Criptografia

Suponha que temos uma lista com $N = 2^n$ itens com N sendo um número grande e n o número de *bits* necessário para representar todos os itens. Se dentre eles existe um que possui uma propriedade única e que queremos localizar (por exemplo, uma chave criptográfica), vamos representar esse item pela letra m. Os N itens podem ser representados equivalentemente por n *qubits*, com cada item correspondendo a um estado da base computacional que podemos representar. Para encontrar o item m usando computação clássica, precisaríamos checar na média $N/2$ itens e, no pior caso, todas os N itens da lista ($O(N)$ operações). Por outro lado, em um computador quântico, podemos encontrar m com cerca de ($O(\sqrt{N})$) operações utilizando o algoritmo de Grover [Grover 1996]. O circuito quântico para executar o algoritmo tem a estrutura apresentada na Figura 3.12. A ideia principal é iniciar o circuito com todos os estados possíveis em superposição uniforme dado um certo número n de *qubits* e aplicar transformações dependentes do problema para que as amplitudes correspondentes aos estados desejados sejam amplificadas às custas das amplitudes dos estados não desejados. Vamos agora explicar em linhas gerais o que cada parte realiza.

Começamos com um registrador de n *qubits* inicializados no estado $|0\rangle$ e podendo ser representados coletivamente como $|0\rangle^{\otimes n}$. Essa notação indica o produto tensorial de n *qubits* configurados no estado quântico $|0\rangle$. Na primeira parte das operações, aplicamos portas Hadamard a todos esses *qubits*, resultando no estado $|s\rangle = H^{\otimes n}|0\rangle^{\otimes n} = |+\rangle^{\otimes n}$ que é uma superposição uniforme de todos os estados da base computacional de n *qubits*. Em seguida, repetiremos t vezes a execução da dupla de transformações chamadas de Orá-

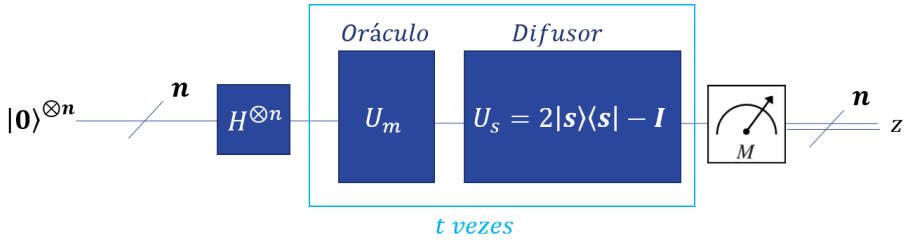


Figura 3.12. Circuito geral para executar o Algoritmo de Grover

culo U_m e Difusor U_s. A estrutura do bloco oráculo dependerá do problema em questão e vai variar de acordo com o que queremos encontrar, enquanto a estrutura do difusor pode ser fixa como $2|s\rangle\langle s| - I$. O papel do oráculo é identificar quais estados são candidatos a soluções e os marcar utilizando uma alteração de fase ($U_m|x\rangle = -|x\rangle$ se $x = m$), mantendo os demais inalterados ($U_m|x\rangle = |x\rangle$ se $x \neq m$). O papel do difusor é amplificar as amplitudes de probabilidade dos estados candidatos em detrimento dos demais de maneira que as respectivas probabilidades desses estados serem medidos aumentem. Essa sequência de operações é repetida um determinado número de vezes até que o estado resultante represente o item (ou itens) desejado com alta probabilidade. No caso de mais de 1 solução, teremos um estado que será uma superposição das soluções possíveis, porém sempre medindo apenas uma delas a cada execução.

O operador resultante a cada iteração da sequência oráculo/difusor é dado por $G = U_s U_m$, sendo G chamado comumente de operador (ou iterador) de Grover. O número t determina uma estimativa da quantidade de repetições desse bloco para obter o resultado desejado com alta probabilidade. Ao fazer essa aplicação sucessiva, a expressão final para o estado quântico do circuito será $|\psi_t\rangle = (U_s U_m)^t |s\rangle$. Para os casos de solução única m ou múltiplas soluções $M = \{m_1, m_2, ...\}$ [Nielsen and Chuang 2010], teremos:

$$t_{única} \approx \frac{\pi}{4}\sqrt{N} \rightarrow O(\sqrt{N}) \quad , \quad t_{múltiplas} \approx \frac{\pi}{4}\sqrt{N/M} \rightarrow O(\sqrt{N/M}) \tag{28}$$

Por exemplo, suponha que estamos utilizando o algoritmo AES e temos uma determinada quantidade de pares mensagem-encriptado (m_i, c_i) de forma a minimizar a probabilidade de encontrarmos chaves falsas durante a busca [Menezes et al. 2018]. Construíndo o bloco Oráculo para implementar o AES, com o algoritmo de Grover conseguimos marcar a solução desejada (a chave criptográfica k) de forma que $U_m|k\rangle = -|k\rangle$ se $Enc(k, m_i) = c_i$. Após $O(\sqrt{N})$ iterações, sendo N o tamanho do espaço de chaves, teremos o estado correspondente à chave k com alta probabilidade na saída do circuito.

Na criptografia simétrica, o nível de segurança estabelece diretamente o esforço necessário (usualmente medido em *bits*) em termos de passos computacionais para descobrirmos a chave secreta utilizada em alguma cifra simétrica ou para calcular pré-imagens de resumos criptográficos para uma determinada função de *hash*. Ou seja, dizer que um algoritmo tem 128 *bits* de nível de segurança significa que a melhor abordagem para quebrá-lo demanda um esforço da ordem de 2^{128} passos computacionais. Na ausência de outros tipos de ataques mais eficientes, podemos associar esse nível de segurança ao

esforço computacional de realizar uma busca exaustiva. No contexto criptográfico, esse tipo de busca nada mais é que uma busca num conjunto de dados não estruturado, tipo de problema onde o algoritmo de Grover pode ser aplicado. No pior caso, analisando apenas a diferença na complexidade dos melhores algoritmos clássico ($O(N) = O(2^n)$) e quântico ($O(\sqrt{N}) = O(2^{n/2})$) para esse mesmo problema, com N representando o número de itens no conjunto e n sendo o número de *bits* necessários para representar todos os itens, obtemos uma aceleração quadrática de eficiência. Neste caso, os algoritmos criptográficos podem ser impactados, mas não de forma crítica, podendo apenas demandar ajustes no tamanho da chave ou da saída da função de *hash* para retornar ao mesmo nível de segurança. Entretanto, vamos analisar em mais detalhes esse potencial impacto em funções de *hash* e nos demais tipos de algoritmos criptográficos simétricos.

De forma geral, para funções de *hash* criptográficas consideradas seguras podemos identificar dois tipos de ataque: o cálculo de pré-imagens e o cálculo de colisões. No primeiro caso, a melhor abordagem acaba sendo a busca exaustiva com complexidade temporal $O(2^n)$ sendo n o tamanho do resumo criptográfico. Um ataque de busca exaustiva, também chamado de força bruta, pode ter sua solução acelerada pelo algoritmo de Grover para $O(2^{n/2})$. No segundo caso, já existe um algoritmo clássico conhecido, explorando um ataque de aniversário, para o cálculo de colisões com complexidade temporal $O(2^{n/2})$ [Katz and Lindell 2014], então o algoritmo de Grover acaba não sendo melhor do que isso. Dessa forma, uma função de *hash* cuja saída possui 256 *bits* (por exemplo SHA-256, BLAKE-256, SHA3-256) acaba tendo sua segurança efetiva para ataques de pré-imagem reduzida para 128 *bits*, o que ainda é considerado seguro em termos dos padrões atuais de nível de segurança, que recomendam um mínimo de 112 *bits* [Barker et al. 2020]. Ou seja, em termos práticos, o algoritmo de Grover não fornece uma ameaça substancial nesse caso. O mesmo tipo de análise pode ser feito para funções de *hash* com saídas de 224 *bits*, o que nos leva a níveis de segurança efetivos de 112 *bits*, no limiar da margem recomendada. Assim, deve-se analisar cada função de *hash* utilizada e verificar se o seu nível de segurança para ataques de pré-imagem, ao ser verificado frente ao algoritmo de Grover, cai abaixo do padrão recomendado (independente de qual for). Além disso, há estudos combinando ideias do ataque de aniversário com o algoritmo de Grover para tentar melhorar o esforço computacional para o cálculo de colisões [Brassard et al. 1998, Hosoyamada and Sasaki 2020]. Dadas todas essas considerações, se o nível de segurança resultante da análise frente ao algoritmo de Grover cair abaixo do adequado, se for desejado um nível de segurança com maior expectativa de vida ou se objetiva-se manter exatamente o mesmo nível de segurança atual para ataques de pré-imagem, bastaria dobrar o tamanho do resumo criptográfico.

No caso de algoritmos criptográficos simétricos que utilizam chaves, como as cifras de bloco, cifras de fluxo e códigos de autenticação de mensagens, se não conhecermos abordagens analíticas mais eficientes, nossa melhor estratégia para recuperar a chave simétrica é a busca exaustiva no espaço de chaves, equivalente a um problema de busca num conjunto de dados não estruturado. Dessa forma, o algoritmo de Grover pode também ser utilizado nesse caso para acelerar essa tarefa para $O(2^{n/2})$. Assim, algoritmos como o AES utilizando chaves de 128 e 192 *bits* teriam seu nível de segurança efetivo reduzido para 64 e 96 *bits*, respectivamente, abaixo dos 112 *bits* recomendado atualmente. Por outro lado, algoritmos como o AES-256 e ChaCha20, que utilizam chaves de 256

bits, teriam sua segurança teoricamente reduzida para 128 *bits*, ainda considerada segura. Assim, para responder a esse potencial impacto, o ideal é que o tamanho das chaves criptográficas simétricas utilizadas no caso dos algoritmos impactados seja dobrada caso seu nível de segurança frente ao algoritmo de Grover caia abaixo do recomendado.

Importante notar que há trabalhos que analisam as dificuldades que a implementação do algoritmo de Grover traria e argumentam que seria inviável concretizar na prática essa aceleração quadrática [Sarah and Peter 2024]. Em todo caso, para teoricamente impactar a segurança de um algoritmo simétrico, a aceleração não necessariamente precisa ser quadrática, basta que o nível de segurança efetivo seja reduzido abaixo do recomendado. Por fim, a análise do impacto do algoritmo de Grover é feita em termos de análise de algoritmos sob o regime assintótico, onde não estamos preocupados com questões de implementação, nos dando um limite superior para o desempenho que pode ser obtido e, nesse caso, a recomendação de dobrar o tamanho das chaves simétricas deve ser adotada.

3.4.1.2. Impactos do Algoritmo de Shor na Criptografia

O algoritmo de Shor pode ser utilizado para fatorar números inteiros (dado um número inteiro N, encontrar seus fatores primos p e q tal que $N = pq$) e calcular logaritmos discretos (dados um número primo p e outros números inteiros g e y, encontrar x em $y = g^x \ mod \ p$), ambos de forma eficiente e com acelerações exponenciais se comparados aos melhores algoritmos clássicos para os respectivos problemas [Shor 1997]. Ou seja, mostrou que os computadores quânticos poderiam ser mais eficientes do que os computadores clássicos para problemas de interesse prático, até então um desafio para os pesquisadores da área.

A ideia principal é transformar o problema da fatoração ou do cálculo de logaritmos discretos em um problema de encontrar o período de uma função em tempo polinomial utilizando um computador quântico. Dessa forma, um algoritmo que resolva esse problema de forma eficiente pode ser utilizado para resolver esses outros problemas de forma eficiente também [Nielsen and Chuang 2010]. Por exemplo, o algoritmo de Shor para fatorar números inteiros apresenta complexidade temporal $O(poly(n))$, onde $poly(n)$ denota uma função polinomial (veremos com mais detalhes algumas possibilidades para essa função na Subseção 3.4.1.4). O melhor algoritmo clássico para resolver o mesmo problema quando N é grande é o *General Number Field Sieve* (GNFS) [Boudot et al. 2020a], que possui complexidade subexponencial dada por $O(e^{(c)(n)^{1/3}(log\ n)^{2/3}})$, onde c é uma constante, em função do tamanho n do número N. Podemos também ignorar a constante e representar de forma mais geral essa complexidade subexponencial como $O(e^{(n)^{\alpha}(log\ n)^{1-\alpha}})$, sendo $0 < \alpha < 1$.

Na criptografia assimétrica, para conseguirmos recuperar a chave privada a partir da chave pública, precisamos resolver determinado problema matemático considerado intratável para computadores clássicos. Três desses principais problemas cuja dificuldade clássica é utilizada como pilar de segurança são o problema da fatoração de números inteiros (IFP - *Integer Factorization Problem*), o problema do logaritmo discreto (DLP - *Discrete Logarithm Problem*) e o problema do logaritmo discreto elíptico (ECDLP - *Elliptic Curve Discrete Logarithm Problem*) [Aumasson 2017]. Eles são considerados intratáveis, pois os melhores algoritmos clássicos apresentam complexidade temporal su-

bexponencial ou exponencial. Entretanto, todos esses problemas podem ser endereçados de forma eficiente pelo algoritmo de Shor e resolvidos em tempo polinomial. Com isso, nossa premissa de que são problemas intratáveis acaba sendo invalidada e não podemos mais utilizar esses algoritmos criptográficos de forma segura, devendo ser substituídos por novos algoritmos cujo modelo de atacante tem como uma de suas premissas a posse de um computador quântico criptograficamente relevante, coletivamente chamados de criptografia pós-quântica (PQC - *Post-Quantum Cryptography*).

A segurança atual de algoritmos como o RSA reside no fato de que p e q, os dois números primos utilizados por cada usuário para criação do seu par de chaves, estão escondidos no valor N, que é público, e fatorar esse número (que na prática é grande, com centenas/milhares de dígitos decimais) é considerado um problema difícil classicamente. O último número fatorado utilizando computação clássica foi o RSA-250 com 250 dígitos decimais, equivalente a um tamanho de 829 *bits*, em 2020 [Boudot et al. 2020b]. A tarefa foi completada em alguns meses e o algoritmo clássico utilizado foi o GNFS. Com a ajuda do algoritmo de Shor, podemos fatorar N e obter p e q de forma eficiente. Com isso, conseguiríamos realizar os seguintes cálculos (e pode ser facilmente obtido e d pode ser calculado de forma eficiente com computação clássica):

$$N = pq \quad , \quad \phi(N) = (p-1)(q-1) \quad , \quad d = e^{-1} mod\ \phi \tag{29}$$

A partir disso, poderíamos forjar assinaturas digitais e decifrar mensagens (por exemplo uma chave que tenha sido encapsulada). Um ataque conhecido que tem por objetivo preparar as informações necessárias para se valer dessa segunda possibilidade é o chamado *Store Now Decrypt Later (SNDL)* ou *Harvest Now Decrypt Later (HNDL)* [Joseph et al. 2022]. Dessa forma, com o uso do algoritmo de Shor, o IFP se torna tratável em tempo polinomial, o que implica que nossa premissa de segurança se torna inválida e faz com que esse tipo de algoritmo criptográfico, que faz uso desse problema, deva ser abandonado.

Para o caso de algoritmos como o DH e o DSA, que utilizam instâncias do problema do logaritmo discreto, suas chaves públicas são geradas como $y = g^x\ mod\ p$ com chave privada x que é o logaritmo discreto. Dados g, p e y, desejamos encontrar x, e este problema também é considerado intratável para computadores clássicos. Porém, novamente Shor propôs um algoritmo quântico que pode resolver esse problema com eficiência polinomial $O(poly(n))$ enquanto classicamente temos complexidade subexponencial dada por $O(e^{(n)^{\alpha}(log\ n)^{1-\alpha}})$, ignorando constantes e com $0 < \alpha < 1$ [Boudot et al. 2020a]. Assim, para o caso do DH, poderíamos recuperar a chave privada de um dos participantes da comunicação e, ao combinar essa informação com a chave pública do outro participante, que pode ser recuperada facilmente, seria possível calcular o segredo compartilhado. Para o caso de assinaturas digitais, como o DSA, resolver esse problema nos permite recuperar a chave privada de assinatura.

Por fim, o algoritmo de Shor para o problema do logaritmo discreto pode ser adaptado para o caso elíptico (ECDLP). Nesse problema, dado dois pontos P e Q na curva elíptica utilizada que satisfazem $Q = xP$ sendo x um escalar e representando a chave privada, o algoritmo pode calculá-lo com complexidade temporal $O(poly(n))$, enquanto as

melhores alternativas clássicas possuem complexidade temporal exponencial $O(e^{poly(n)})$ [Roetteler et al. 2017]. Dessa forma, algoritmos como ECDH e ECDSA também devem ser abandonados dada a existência do algoritmo de Shor. Para trabalhos que discutem em profundidade a aplicação e melhorias do algoritmo de Shor para o DLP e para o ECDLP, consultar [Roetteler et al. 2017, Häner et al. 2020, Aono et al. 2022, Hhan et al. 2023].

A partir dessa discussão, as implicações são que os melhores algoritmos clássicos para resolver instâncias dos três problemas comentados para os tamanhos de parâmetros utilizados atualmente acabariam resultando em escalas de tempo muito grandes e impraticáveis, enquanto o algoritmo de Shor estaria em escalas menores e factíveis. Naturalmente, o tempo exato de execução só seria possível determinar levando-se em conta a implementação específica adotada em computadores quânticos reais de larga escala, ainda não existentes. Em suma, frente ao algoritmo de Shor, todos os criptossistemas que baseiam sua segurança na dificuldade desses três problemas devem ser abandonados e substituídos pelos novos algoritmos pós-quânticos.

3.4.1.3. Resumo dos Impactos em Criptografia

A Tabela 3.1 sumariza o que foi discutido nas subseções anteriores. Lembrando que todo e qualquer sistema atual será impactado em maior ou menor grau, independente de fabricantes e de aplicações, e que os esforços de migração, sejam para impactos mais modestos como no caso simétrico ou para impactos críticos como no caso assimétrico, constituem tarefa desafiadora. Além disso, métricas como tempo de processamento, consumo de energia e uso de memória também são afetadas, criando desafios diversos para o processo de migração como um todo.

Tabela 3.1. Resumo dos Impactos em Criptografia.

Primitiva Criptográfica	Exemplos de Algoritmos	Problema Subjacente	Complexidade Temporal Clássica	Complexidade Temporal Quântica	Contramedida
Hash	SHA-2, SHA-3	Cálculo de pré-imagem	$O(2^n)$	$O(2^{n/2})$	Ajustar o tamanho do *hash* se necessário
Simétrica	AES, ChaCha20	Busca de chave	$O(2^n)$	$O(2^{n/2})$	Ajustar o tamanho da chave se necessário
Assimétrica	RSA	IFP	$O(e^{(n)^{\alpha}(log\ n)^{1-\alpha}})$ $0<\alpha<1$	$O(poly(n))$	Substituir algoritmos
	DH, DSA, ElGamal	DLP			
	ECDH, ECDSA, EdDSA	ECDLP	$O(e^{poly(n)})$		

3.4.1.4. Exemplo de Execução do Algoritmo de Shor

Vamos mostrar a aplicação prática do algoritmo de Shor para o problema da fatoração de números inteiros. Veremos um exemplo completo de como utilizar o algoritmo para fatorar o número $N = 15$. Naturalmente, esse número é pequeno e muitos dos detalhes técnicos serão omitidos visto que seu pleno entendimento demandaria uma exposição mais longa e aprofundada sobre o assunto, o que está além do escopo deste minicurso. O exemplo completo, incluindo o código associado, está disponível em: `https://github.com/vthayashi/quantum-crypto`.

3.4.2. Fundamentos de Criptografia Pós-Quântica

A principal diferença entre a criptografia moderna tradicional e a pós-quântica está em seus modelos adversariais. Enquanto uma assume a capacidade limitada de computadores convencionais, a PQC considera um adversário com acesso a computadores quânticos capazes de executar, de forma eficiente, os algoritmos apresentados na Seção 3.3.

A criptografia pós-quântica, ao contrário da criptografia quântica, não depende e não almeja o uso de sistemas quânticos para sua implementação, pois seus algoritmos são executáveis em computadores clássicos. Em resumo, a criptografia pós-quântica trata de mitigar as vulnerabilidades inseridas pela computação quântica [Bernstein et al. 2009].

O centro das preocupações atuais e do desenvolvimento da área da criptografia pós-quântica são os algortimos que tomam a primitiva do uso de chaves assimétricas para o seu funcionamento. Isso se deve pois, de forma diferente do impacto quadrático do algoritmo de Grover em funções *hash* criptográficas e algoritmos simétricos, o algoritmo de Shor resolve os problemas matemáticos subjacentes (fatoração de inteiros e logaritmo discreto) de algoritmos assimétricos em tempo polinomial, como discutido na Subseção 3.4.1. Isso implica que quando um computador quântico criptograficamente relevante vier a existir, os criptossistemas baseados nesses problemas estarão completamente quebrados [Mosca 2018].

Ao contrário dos casos das mitigações à ameaça quântica das outras primitivas, aumentar o tamanho da chave significa uma ineficiência crescente do uso de recursos computacionais sem ganho de segurança significativo no caso assimétrico, já que o algoritmo de Shor continuará a resolver os problemas fundamentais em tempo polinomial. Para isso, diferentes classes de problemas matemáticos têm sido estudadas para substituir a segurança baseada na fatoração de inteiros e no logaritmo discreto [Beullens et al. 2021].

O processo de criação de algoritmos pós-quânticos começa com a seleção de uma classe de problemas matemáticos que sejam computacionalmente intratáveis em computadores clássicos e para os quais não existam algoritmos quânticos eficientes. Sobre essa base teórica escolhida, pesquisadores desenvolvem e propõem diversos algoritmos concretos dentro de cada família. As principais classes de problemas computacionais e algoritmos considerados para PQC podem ser encontrados na Tabela 3.2.

A dificuldade da criação de certos problemas computacionais em reticulados é a

Tabela 3.2. Principais classes de problemas em criptografia pós-quântica, suas dificuldades matemáticas e exemplos de algoritmos. Adaptado de [Beullens et al. 2021]

Classe	Fonte da Dificuldade	Exemplos de Algoritmos
Baseada em Reticulados	Problemas de reticulados euclidianos, como LWE e Module-LWE	Kyber, Dilithium, Falcon, SABER
Baseada em Códigos	Problema de Decodificação de Erros	Classic McEliece, BIKE, HQC
Baseada em Hashes	Colisões e pré-imagens em funções *hash* seguras (ex. SHA-3)	SPHINCS+
Baseada em Isogenias	Problema da Isogenia Supersingular	SIKE
Baseada em Sistemas Multivariados	Problema MQ (Multivariate Quadratic) sobre $\mathbb{F}_q$	Rainbow, GeMSS

base de segurança para a construção de alguns esquemas PQC relevantes. Um reticulado em $\mathbb{R}^n$ é o conjunto de todas as combinações lineares com coeficientes inteiros de n vetores linearmente independentes em $\mathbb{R}^n$ [Ajtai 1996]. Trata-se de uma malha regular de pontos no espaço n-dimensional (Figura 3.13). Problemas clássicos construídos são de simples entendimento, como o *Shortest Vector Problem* (SVP), que se baseia em encontrar o vetor não nulo mais curto em um reticulado, ou o *Closest Vector Problem* (CVP) que tem como objetivo encontrar o vetor do reticulado mais próximo de um ponto arbitrário definido.

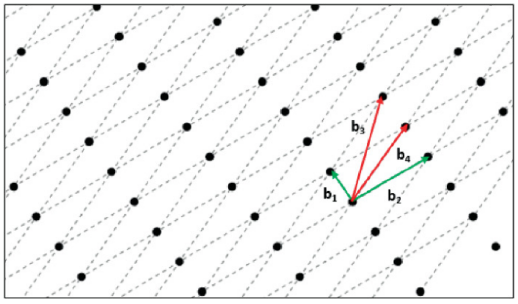


Figura 3.13. Representação de um reticulado (*lattice*) no plano com duas bases distintas. Os vetores $\mathbf{b}_1$ e $\mathbf{b}_2$ (em verde) formam uma base do reticulado, enquanto os vetores $\mathbf{b}_3$ e $\mathbf{b}_4$ (em vermelho) representam uma base alternativa. Todos os pontos pretos no plano representam os pontos do reticulado gerado por combinações lineares inteiras dessas bases. Adaptado de [Shah et al. 2025].

Na classe baseada em reticulados (*lattice-based*), o problema *Learning With Errors* (LWE), juntamente com variações como o *Module-LWE* e *Ring-LWE*, ganham destaque na construção de algoritmos PQC. Embora o problema LWE não seja, em formulação direta, um problema de reticulados como o SVP e o CVP, ele pode ser reduzido para problemas fundamentais desse conjunto. Foi demonstrado formalmente que resolver instâncias aleatórias do problema LWE é tão difícil quanto resolver problemas clássicos de reticulados no pior caso, como o *Gap Shortest Vector Problem* (GapSVP) e o *Shortest Independent Vectors Problem* (SIVP) [Regev 2005]. O LWE estabelece que dado um sistema linear com um ruído, tentar recuperar o vetor original é computacionalmente difícil. Suas variações, como o *Ring-LWE* e o *Module-LWE*, foram desenvolvidas para otimizar desempenho e reduzir o tamanho das chaves na construção de criptossistemas [Lyubashevsky et al. 2010]. Esquemas baseados em reticulados, como o CRYSTALS-KYBER e o CRYSTALS-DILITHIUM, comentados na Subseção 3.4.3 são tidos como eficientes e seguros.

Enquanto em reticulados o ruído está num espaço geométrico, na classe baseada em Códigos (*code-based*) ele é adicionado a palavras codificadas em espaço finito (por exemplo, *bits*). Essa abordagem remonta à uma proposta clássica [McEliece 1978], que utilizou códigos de Goppa para construir um criptossistema assimétrico que competiu por adoção em sua época. A dificuldade do problema principal toma que dado um código linear aleatório e uma palavra corrompida por erros, determinar a mensagem original é um problema considerado intratável por algoritmos conhecidos [Bernstein et al. 2009]. Esquemas baseados em códigos geralmente apresentam tamanhos de chave pública significativamente maiores do que outras abordagens, sendo uma limitação em comparações

para adoção prática.

Na criptografia baseada em *hashes* (*hash-based*) a ideia é usar as dificuldades provenientes dos requisitos de segurança de funções *hash* criptográficas, apresentadas na Subseção 3.2.3, como problema computacional para assegurar criptossistemas assimétricos. O principal problema utilizado é que dado somente o valor de uma função *hash*, encontrar uma pré-imagem primária, secundária ou colisão é computacionalmente inviável. Em vez de confiar em uma estrutura matemática complexa, essa abordagem parte da premissa de que funções como SHA-3 são seguras contra ataques quânticos, portanto constituem um ponto para construção de outros mecanismos [Beullens et al. 2021].

A pesquisa sobre a classe de algoritmos PQC baseados em Isogenia (*isogeny-based*) tornou-se desafiadora após a quebra pública do principal algoritmo (SIKE) da família [Castryck and Decru 2023]. Essa abordagem explora a dificuldade de encontrar isogenias entre curvas elípticas supersingulares. Uma isogenia é, de forma imprecisa e simples, um morfismo entre duas curvas elípticas que preserva a estrutura algébrica dos grupos dessas curvas. Curvas elípticas supersingulares demonstram propriedades específicas em relação ao conjunto genérico, quando definidas sobre corpos finitos. O problema pode ser formulado dado que econtrar uma isogenia de grau pequeno entre duas curvas elípticas supersingulares E e E' é computacionalmente intratável, mesmo para computadores quânticos [Jao and De Feo 2011].

A criptografia multivariada (*multivariate*) têm como fundamento a dificuldade de resolver sistemas de equações polinomiais multivariadas sobre corpos finitos, problema conhecido como problema MQ (*Multivariate Quadratic problem*), NP-difícil em geral. O problema pode ser formulado dado que para uma função polinomial $P : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$, composta por equações quadráticas em variáveis sobre um corpo finito $\mathbb{F}_q$, determinar uma pré-imagem x tal que $P(x) = y$ é difícil [Ding et al. 2006]. Um exemplo destaque dessa classe foi o esquema Rainbow que foi quebrado antes de sua padronização [Beullens et al. 2021].

Como discutido na próxima subseção, as padronizações indicam que a classe de algoritmos baseados em reticulados será adotada como principal mecanismo PQC.

3.4.3. Padronização NIST de Criptografia Pós-Quântica

Dada a relevância do cenário de quebra de algoritmos de criptografia assimétrica baseados nos problemas de logaritmo discreto e de fatoração de números inteiros, além dos impactos em algoritmos simétricos citados anteriormente, esforços de padronização de algoritmos criptográficos pós-quânticos estão em andamento [Alagic et al. 2025].

O estabelecimento de padrões criptográficos é um processo relevante para suportar o correto uso dos algoritmos. O NIST (*National Institute of Standards and Technology*) é uma agência do departamento de Comércio dos Estados Unidos que vêm apoiando a padronização de algoritmos criptográficos (e.g., SHA-3), e que conta com uma reputação relevante na comunidade de segurança da informação.

O processo de padronização do NIST envolve uma análise de segurança e considerações sobre eficiência computacional dos algoritmos candidatos. Em relação à criptografia pós-quântica, há uma chamada pública para mecanismos de encapsulamento de

chaves (KEM - *Key Encapsulation Mechanism*) e assinatura digital resilientes aos ataques potenciais de computadores quânticos. No decorrer das rodadas de avaliação, a resiliência a ataques clássicos e quânticos foi avaliada pela comunidade, em um esforço conjunto para a seleção dos melhores candidatos [Paar et al. 2024].

A iniciativa do NIST em PQC teve início em 2012 com a criação de um grupo de trabalho. O primeiro workshop sobre o tema ocorreu em 2015, o primeiro relatório veio em 2016, e a chamada pública inicial se encerrou em Novembro de 2017 [Chen et al. 2017]. Desde então, outras três novas rodadas ocorreram [Alagic et al. 2025]. A Tabela 3.3 mostra a evolução das submissões e seleção de finalistas no decorrer das rodadas de padronização.

Tabela 3.3. Resumo da evolução dos candidatos no processo de padronização NIST para criptografia pós-quântica (PQC). Em negrito, o número de algoritmos finalistas; entre parênteses, o número de algoritmos com avaliação pendente. Adaptado de [Paar et al. 2024] e [Alagic et al. 2025].

Etapa	Data de Anúncio	# KEM	# Assinatura Digital
Submissões Iniciais	Dez 2017	40	29
Após 1ª Rodada	Jan 2019	17	9
Após 2ª Rodada	Jul 2020	4 + (5)	3 + (3)
Após 3ª Rodada	Set 2022	**4** + (5)	**3** + (3)

Dentre os finalistas, há 5 algoritmos selecionados para padronização: 2 para encapsulamento de chaves e 3 para assinatura digital [Alagic et al. 2025]. CRYSTALS-KYBER é um mecanismo baseado em reticulados que foi selecionado devido ao seu compromisso entre segurança e eficiência quando comparado aos mecanismos de criptografia assimétrica existentes, e se tornou o padrão FIPS 203 adotando o nome *Module-Lattice-Based Key-Encapsulation Mechanism* (ML-KEM). CRYSTALS-DILITHIUM é um mecanismo utilizado para assinatura digital que também é baseado em reticulados, e compõe o padrão FIPS 204 com o nome *Module-Lattice-Based Digital Signature Algorithm* (ML-DSA). SPHINCS+ é um mecanismo utilizado para assinatura digital baseado em *hash*, e que se tornou o padrão FIPS 205 sob o nome *Stateless Hash-Based Digital Signature Algorithm* (SLH-DSA) [Paar et al. 2024, Alagic et al. 2025].

Os algoritmos Falcon (baseado em reticulados) e HQC (baseado em códigos corretores de erros) também foram selecionados, e estão atualmente aguardando a publicação de seus padrões associados. Como a maioria dos algoritmos finalistas na rodada 3 são baseados em reticulados, a rodada 4 obteve 3 candidatos baseados em outros problemas matemáticos (desconsiderando na contagem o candidato SIKE, que foi provado inseguro). Considerando que a segurança dos mecanismos criptográficos é baseada em problemas matemáticos complexos com resolução inviável em computadores clássicos, e que computadores quânticos tem o potencial de resolver novas classes de problemas, abordagens complementares baseadas em outros tipos de problemas matemáticos são desejáveis [Alagic et al. 2025].

Em relação à transição dos algoritmos criptográficos, o NIST não autoriza mais o uso de ECDSA, EdDSA (*Edwards-curve Digital Signature Algorithm*) e RSA do padrão FIPS 186 para assinatura digital após 2035, sendo que RSA e ECDSA são considera-

dos depreciados após 2030 (i.e., podem ser usados, mas há risco potencial associado) [Moody et al. 2024]. Para assinatura digital com níveis de segurança de 128, 192 e 256 *bits*, o NIST passa a recomendar o uso de ML-DSA (i.e., CRYSTALS-DILITHIUM) nas configurações ML-DSA-44, ML-DSA-65 e ML-DSA-87 da FIPS 204, além de padrões baseados em *hash* [Cooper et al. 2020]. Para o estabelecimento seguro de chaves, algoritmos como Diffie-Hellman e RSA não são mais autorizados para uso pelo NIST a partir de 2035, e o padrão recomendado passa a ser o ML-KEM (i.e., CRYSTALS-KYBER) nas configurações ML-KEM-512, ML-KEM-768, e ML-KEM-1024 para níveis de segurança de 128, 192 e 256 *bits*, respectivamente [Moody et al. 2024].

Conforme os novos padrões são adotados pelas organizações com a transição dos mecanismos criptográficos, se atentar a aspectos de implementação passa a ser relevante, uma vez que os algoritmos podem suportar um nível adequado de segurança nas aplicações, mas uma configuração inadequada e erros intencionais/inadvertidos podem resultar em vulnerabilidades. Por exemplo, ataques de canal lateral (*side channel attacks*) podem impactar a segurança de mecanismos PQC de assinatura digital e encapsulamento de chaves [Jedlicka et al. 2022, Ravi et al. 2024]. A análise da temporização, sinais eletromagnéticos, e níveis de consumo de energia podem levar ao comprometimento de mecanismos de criptografia pós-quântica baseados em reticulados, inclusive com o uso de redes neurais para execução dos ataques [Wang et al. 2023, Huang et al. 2024]. Neste cenário, o uso de métodos de avaliação (e.g., FIPS 140-3) é um processo relevante [Saarinen 2022]. Contramedidas como adição de ruído, introdução de atrasos aleatórios, e outras medidas de mascaramento podem contribuir para mitigar estas ameaças, tanto em software quanto em hardware [Zhao et al. 2023, Dobias et al. 2025].

3.4.4. Criptografia Quântica

A criptografia quântica surge como uma alternativa aos métodos clássicos de proteção de dados, pois transfere a segurança da matemática para as leis fundamentais da mecânica quântica [Duim and Portácio 2023]. Em vez de confiar na dificuldade de calcular fatores numéricos ou logaritmos discretos, ela explora fenômenos como superposição, emaranhamento e o princípio da incerteza de Heisenberg para garantir que qualquer tentativa de espionagem seja detectável. Essa abordagem desvincula-se das vulnerabilidades expostas pelos algoritmos de Shor e Grover, pois mesmo um adversário com poder computacional ilimitado não consegue clonar estados quânticos sem introduzir ruídos mensuráveis no canal de comunicação. O componente central da criptografia quântica é a Distribuição Quântica de Chaves (QKD - *Quantum Key Distribution*), na qual duas partes compartilham uma sequência de *bits* aleatórios e secretos codificada na polarização de fótons [Duim and Portácio 2023]. O processo completo pode ser visualizado na Figura 3.14, que apresenta um esquema geral dessa comunicação segura.

Protocolos consagrados, como o BB84, combinam bases distintas (por exemplo as bases X e Z) para codificar 0s e 1s em estados quânticos distintos [Marquezino and Helayel-Neto 2003]. Durante a troca, o receptor seleciona aleatoriamente uma base para medir cada fóton; só quando a base coincide com a do emissor é possível recuperar corretamente o *bit* original. Ao comparar, por um canal clássico, as bases usadas (sem revelar os valores dos *bits*), ambas as partes conseguem eliminar medições incongruentes, reduzindo o tamanho da chave mas assegurando sua inviolabilidade

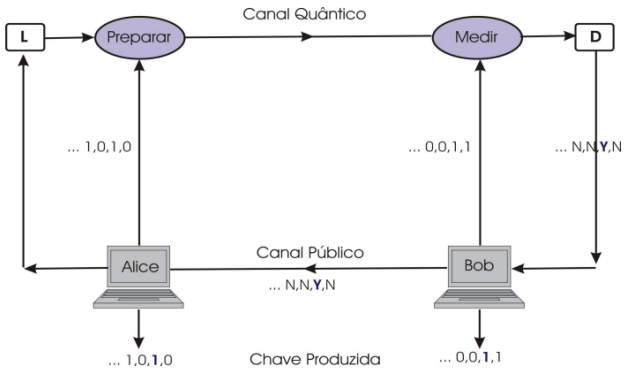


Figura 3.14. Processo completo do QKD. Adaptado de [Takagi 2003]

[Mendes et al. 2011]. Se a taxa de erro quântico (QBER - *Quantum Bit Error Rate*) exceder um limiar pré-definido, indica-se a presença de um intruso e todo o processo deve ser reiniciado para preservar a confidencialidade da comunicação.

Além do BB84, outras implementações, como o protocolo E91 (baseado em emaranhamento quântico), reforçam a robustez da QKD ao permitir que correlações quânticas garantam a geração de chaves idênticas mesmo à distância. Em aplicações práticas, combina-se a QKD com cifradores de uso único (*one-time pad*), obtendo-se segurança teórica perfeita: a interceptação administrativa de uma chave quântica deixa vestígios irrefutáveis, fazendo da criptografia quântica uma barreira efetiva contra ataques futuros que utilizem computador quântico [Mendes et al. 2011].

3.4.5. Estudo de Caso: Impactos no Bitcoin

O Bitcoin, protocolo proposto por [Nakamoto 2008], fundamenta-se em princípios de criptografia clássicos, como o algoritmo de assinatura digital ECDSA e as funções de *hash* SHA-256 e RIPEMD-160, que buscam garantir autenticidade, integridade e irretratabilidade das transações. Embora tais mecanismos sejam considerados seguros frente a ataques por computadores clássicos, estudos têm demonstrado que algoritmos quânticos, como os de Shor e Grover, podem comprometer sua segurança, evidenciando a necessidade de reavaliação da resiliência do protocolo Bitcoin em um cenário pós-quântico.

No contexto de assinaturas digitais, o Bitcoin emprega ECDSA sobre a curva elíptica secp256k1 como mecanismo padrão desde sua criação. A proposta de melhoria Taproot, ativada em 2021, introduziu o suporte a assinaturas Schnorr, que, embora baseadas na mesma curva, possuem características formais distintas, como linearidade e eficiência em agregação de chaves [Maxwell et al. 2020]. Apesar das sutis diferenças, ambos os esquemas se fundamentam na dificuldade do ECDLP, considerado computacionalmente intratável sob o paradigma clássico [Menezes et al. 2018].

Contudo, o algoritmo de Shor permite resolver o ECDLP em tempo polinomial em um computador quântico com *qubits* e correção de erros suficientes [Shor 1997], conforme vimos anterioremente. Dessa forma, qualquer chave pública que se torne visível na blockchain pode ser utilizada por um agente adversário, com acesso a um computador quântico suficientemente avançado, para derivar a chave privada associada e realizar uma

transação maliciosa [Aggarwal et al. 2017].

Além das assinaturas, o protocolo Bitcoin também utiliza funções de *hash*, sendo o algoritmo SHA-256 empregado em diversas camadas, incluindo a construção do cabeçalho de blocos no mecanismo de prova de trabalho (PoW - *Proof-of-Work*), a geração dos identificadores de transações (TXIDs) e a verificação de integridade de dados [Narayanan et al. 2016].

Diante deste cenário, algumas pesquisas têm investigado as vulnerabilidades do Bitcoin e explorado possíveis alternativas de adaptação do protocolo. Em um relatório da Chaincode Labs, foi avaliado os impactos de computadores quânticos criptograficamente relevantes sobre o ecossistema, estimando que aproximadamente 6,26 milhões de BTC encontram-se vulneráveis à extração de chaves privadas por meio do algoritmo de Shor [Milton and Shikhelman 2025]. O estudo categoriza os scripts de saída quanto à sua suscetibilidade, alinhando-se à análise de que P2PK, P2MS e P2TR são vulneráveis, enquanto P2PKH e P2WPKH se tornam vulneráveis durante o processo de gasto.

Entre as soluções propostas para mitigar esses riscos, destaca-se o `BIP-360` [Beast 2024], que introduz o conceito de *Pay-to-Quantum-Resistant-Hash* (P2QRH). Essa proposta visa permitir a utilização de assinaturas pós-quânticas diretamente em scripts de gasto por meio de algoritmos como Falcon, CRYSTALS-DILITHIUM e SPHINCS+, padronizados pelo NIST [National Institute of Standards and Technology (NIST) 2024]. A proposta prevê uma combinação de assinatura Schnorr com assinaturas pós-quânticas em um esquema híbrido e a adoção de novos formatos de endereço (com prefixo "bc1r").

Outra proposta é o `BIP-347`, que sugere a reintrodução do opcode `OP_CAT` com o objetivo de viabilizar construções baseadas em assinaturas de Lamport, uma classe de esquemas baseados em *hash* considerados resistentes a ataques quânticos. Ao possibilitar a concatenação de elementos na pilha de execução, o opcode permitiria a criação de scripts Taproot com caminhos de gasto alternativos baseados em assinaturas de Lamport [Heilman and Sabouri 2023]. De forma complementar, [Corallo 2024] propôs uma abordagem por meio da introdução de um opcode dedicado à verificação de assinaturas SPHINCS+ (`OP_SPHINCS`). A inclusão de tal verificação permitiria que carteiras adotassem, de maneira compatível, saídas com gastos criptograficamente seguros mesmo diante de CRQCs, ainda que com um aumento significativo no tamanho das transações.

3.5. Oportunidades em Computação Quântica

Esta seção explora aplicações da Computação Quântica em áreas como Processamento de Linguagem Natural (PLN) utilizando Quantum Machine Learning (QML), saúde e finanças, como por exemplo otimização de portfólios de investimento [Biamonte et al. 2017]. Esta seção possui como objetivo trazer uma visão complementar de oportunidades desta tecnologia emergente, além dos desafios e riscos associados aos impactos na criptografia.

3.5.1. QNLP

O PLN é o campo de estudo que desenvolve soluções computacionais para problemas e tarefas que envolvam o uso da linguagem, seja ela escrita ou falada, sendo dividido em duas vertentes: NLU (*Natural Language Understanding*), voltada à análise e interpreta-

ção de texto, e NLG (*Natural Language Generation*), dedicada à geração de linguagem coerente [Caseli and Nunes 2024].

Segundo [Nausheen et al. 2025], o uso da computação quântica se dá em aproveitar as características destes sistemas, como a superposição e o emaranhamento, para otimizar tanto a representação de características de um texto quanto a própria inferência de acordo com o tipo de tarefa adotada. Assim, otimizando seja a representação e extração de características, seja o próprio custo computacional, como demonstrado por [Correia et al. 2022], a incorporação do algoritmo de Grover ao procedimento de *Question Answering* reduziu a complexidade da busca de $O(N)$ para $O(\sqrt{N})$, proporcionando, portanto, uma aceleração quadrática na identificação da resposta correta. Ainda segundo [Nausheen et al. 2025], que mapearam o uso da computação quântica em tarefas de PLN, identificam-se três principais aplicações desta tecnologia no campo:

1. *Encoding*: é a codificação dos dados clássicos — no caso de PLN, o texto — em informação quântica, mapeando-os para estados quânticos. Essas técnicas de encoding facilitam uma representação eficiente dos dados e ajudam a capturar relações entre eles.

2. Modelagem: a utilização da computação quântica em nível de modelagem pode variar conforme a tarefa de linguagem natural adotada, mas, de modo geral, os modelos classificam-se em:

    - *Categorical QNLP models*: utilizam uma representação estrutural e gramatical da língua, combinando essa estrutura com o significado das palavras por meio de conceitos matemáticos como *compact closed categories*, *tensor products* e *diagrammatic calculus*. Subdivide-se em modelos Categóricos (DisCoCat) e modelos de Circuito (DisCoCirc).
    - *Probabilistic models*: exploram a representação geométrica do modelo de probabilidade quântica para maior flexibilidade na tomada de decisão, permitindo transições entre diferentes bases do espaço vetorial.
    - *Quantum circuit models*: usam circuitos parametrizados para realizar inferências. Incluem os VQA (*Variational Quantum Algorithms*), adequados aos dispositivos da era NISQ (*Noise Intermediate-Scale Quantum*), nos quais um estimador quântico avalia a função de custo do problema enquanto um otimizador clássico ajusta os parâmetros; e as QNNs (*Quantum Neural Networks*), variantes dos VQA que codificam dados via *feature map*, aplicam um modelo variacional para otimizar a *loss function* e, por fim, pós-processam classicamente os resultados de medida.
    - *Quantum Kernel models*: mapeiam dados clássicos em espaços de alta dimensão via *kernel* quântico, extraindo mais características e reduzindo ruído. Um exemplo é o QSVM (*Quantum Support Vector Machine*), que reformula a SVM (*Support Vector Machine*) em mínimos quadrados para inverter matrizes eficientemente e acelerar treinamento e classificação em *big data*.
    - *Quantum language models*: usam princípios quânticos para representar palavras como estados quânticos, capturando relações semânticas de forma mais

rica; variantes de inspiração quântica (*quantum-inspired*) simplificam isso mapeando cada palavra para um dos vetores da base ortogonal de referência e modelos como o SQLM (*Session-based Quantum Language Model*) aplicam certas transformações para rastrear a evolução das buscas dos usuários.

- *Hybrid models*: combinam processamento clássico e quântico, distribuindo etapas entre ambos, de modo a aproveitar os benefícios da computação quântica mesmo com as limitações de hardware da era NISQ.

3. *Hyperparameter tuning*: utiliza as características dos sistemas quânticos para melhorar a eficiência e a eficácia em tarefas de otimização. Dentre os principais métodos estão o *Quantum-Accelerated Hyperparameter Tuning*, o *Quantum Annealing* e abordagens híbridas.

A computação quântica já possui pesquisas de PLN em andamento com provas de conceito práticas [Correia et al. 2022, Omar and El-Hafeez 2023, Nausheen et al. 2025]. Por exemplo, [Omar and El-Hafeez 2023] desenvolveram um classificador de sentimentos para textos de redes sociais em árabe, utilizando QSVM para inferência em comparação a um modelo clássico de *Random Forest*, avaliando o desempenho de ambos em diversos conjuntos de dados. De qualquer forma, por estar em estágio inicial, ainda demanda pesquisas adicionais para consolidar métodos e ampliar seu impacto.

3.5.2. Aplicações em Saúde

Os recentes avanços da computação clássica, com o desenvolvimento de algoritmos e aumento no poder computacional amplamente disponível, permitiram a análise de dados no contexto das ciências da saúde e medicina de forma inédita. Com a computação quântica, diversas técnicas vêm sendo testadas e aprimoradas para acelerar algoritmos, reduzir o consumo de energia e flexibilizar os requisitos de dados para análise [Flöther 2023]. Nesta seção, delineiam-se os principais usos da computação quântica na área da saúde, destacando os algoritmos empregados e perspectivas futuras.

Inicialmente, como apontado em [Flöther 2023], os primeiros usos da computação quântica nas ciências biológicas se concentraram na resolução de problemas da bioquímica e biologia computacional, sobretudo por meio de técnicas de simulação quântica de moléculas, como proteínas. Esse caso de uso atraiu grande interesse por representar um dos principais gargalos em plataformas clássicas, em razão de sua elevada complexidade. Além disso, a viabilidade de execução em hardware quântico ruidoso atualmente disponível, somado ao potencial de escalabilidade com o desenvolvimento de dispositivos mais poderosos, motivou a busca de soluções quânticas para a superação dos impasses correntes [Robert et al. 2021]. Nesse cenário, esperam-se aplicações ainda mais avançadas, por exemplo, na predição de estruturas de proteínas com aminoácidos sintéticos, situação na qual mesmo técnicas de *machine learning* falham devido a falta de dados para treinamento, o que reduziria a dependência de experimentos laboratoriais caros e demorados.

Embora o maior destaque recaia sobre a simulação de compostos, as aplicações quânticas na saúde não se limitam a isso. Nesse contexto, a genômica logo passou a explorar o poder de processamento dessas máquinas trabalhar com grandes volumes de dados com estruturas complexas. Tanto algoritmos quânticos consolidados quanto

técnicas de QML foram avaliadas para acelerar tarefas genômicas. Nesse âmbito, em [Sarkar et al. 2021] propõe-se a utilização do algoritmo de Grover para acelerar o alinhamento de sequências de DNA. Já em [Prousalis and Konofaos 2019] elabora-se sobre a utilização de um algoritmo de *Quantum Pattern Recognition* (QPR) para pareamento de sequências de material genético. Curiosamente, o QPR utiliza como subrotina a Transformada Quântica de Fourier (QFT - *Quantum Fourier Transform*), também empregada no algoritmo de Shor, evidenciando a versatilidade dessas técnicas.

Por fim, quando lançamos o olhar para a utilização de algoritmos de *quantum machine learning*, de maneira semelhante ao que ocorre com suas contrapartes clássicas, observa-se ampla variedade de aplicações, principalmente em diagnóstico. *Quantum Neural Networks* (QNNs), *Quantum K-Nearest Neighbors*, *Quantum Support Vector Classifiers* (QSVCs), *Quantum Random Forest*, entre outras abordagens, têm sido usadas para detectar diversos distúrbios de saúde. Destacam-se os trabalhos que utilizam QNNs em exames de imagem para o diagnóstico de Alzheimer [Shahwar et al. 2022] e que fazem um comparativo de técnicas de machine learning clássico e quântico para a predição de insuficiência cardíaca a partir de indicadores clínicos [Kumar et al. 2021].

É interessante notar que, desde o seu surgimento, a medicina vem se apropriando das tecnologias mais recentes de cada momento histórico [Flöther 2023]. Assim, o atual entusiasmo na criação de aplicações médicas com computação quântica, seja para aprimorar métodos existentes ou para fomentar descobertas inéditas, revela o potencial transformador dessa tecnologia, prometendo alterar profundamente o cuidado à saúde e contribuir para a melhora na qualidade de vida da população.

3.5.3. Aplicações em Finanças

A computação quântica tem emergido como uma tecnologia com grande potencial de aplicação em diversos setores, especialmente nas finanças, onde a complexidade dos modelos matemáticos e a demanda por cálculos intensivos representam desafios significativos para a computação clássica [Chang et al. 2023]. Vale destacar, algoritmos quânticos têm sido propostos para o cálculo de medidas de risco, como o valor em risco (VaR) e o valor condicional em risco (CVaR), tendo assim forte atuação na gestão de carteiras de crédito e na tomada de decisão sob incerteza [Miyamoto 2022]. Essas inovações sugerem uma computação quântica extensiva, podendo impactar não só um mercado local, mas em escala de economia global, trazendo melhorias na precisão e na eficiência do processamento de grandes volumes de dados financeiros.

Dentro da diversidade de aplicações possíveis com a tecnologia, vale detalhar aqui a aplicação na análise de risco de portfólios de investimentos. O trabalho [Woerner and Egger 2019] apresenta um estudo que utiliza utiliza a plataforma de computação quântica da IBM para precificar um título do tesouro americano sob cenários de aumento de juros, além de simular o cálculo de risco financeiro para uma carteira composta por dois ativos de dívida pública com diferentes vencimentos. Os resultados apresentaram, para os problemas de escala reduzida que foram estudados, uma taxa de convergência superior aos métodos clássicos (Monte Carlo) mesmo com interferência de erros quânticos. Já em [Rebentrost and Lloyd 2024] é proposto algoritmo quântico para otimização de carteiras que permitem determinar a curva de *trade-off* entre risco e retorno

e amostrar a partir do portfólio ideal. Esse algoritmo utiliza dados históricos de retornos dos ativos e afirmam alcançar um tempo de execução de $log(N)$, superando os algoritmos clássicos com tempo de execução polinomial N.

A aplicação de técnicas de computação quântica para a otimização de portfólios também tem se expandido por meio de abordagens híbridas, como o *Variational Quantum Eigensolver* (VQE). Em [Buonaiuto et al. 2023] é apresentada uma formulação geral do problema de otimização quadrática restrita, transformado em um problema de Otimização Binária Quadrática Sem Restrições (QUBO - *Quadratic Unconstrained Binary Optimization*). Essa abordagem permite lidar com a complexidade que existe na seleção de carteiras sob restrições financeiras. Em outro trabalho [Veselỳ 2022], é explorado uma estratégia híbrida clássico-quântica usando recozimento quântico no computador D-Wave 2000Q, onde soluções iniciais geradas por métodos clássicos são refinadas por meio de recozimento quântico reverso, o que permite obter melhores tempos de convergência conforme o número de variáveis aumenta. Além disso, em [Lang et al. 2022], é proposto um fluxo de trabalho que combina pré-processamento clássico com uma nova formulação do modelo QUBO, permitindo flexibilidade quanto ao número de ativos e ao valor investido em cada um. Esse modelo foi testado em diferentes plataformas de recozimento, incluindo simulação clássica (*simulated annealing*), recozimento digital (*Fujitsu Digital Annealer*) e recozimento quântico (*D-Wave Advantage*), utilizando dados reais da bolsa de Nova York.

Diante desse cenário, é possível observar pesquisas recentes que demonstram avanços na aplicação de algoritmos quânticos híbridos e modelos de otimização adaptados a hardwares reais, indicando que as finanças podem ser uma das primeiras áreas a colher benefícios tangíveis dessa tecnologia [Khang et al. 2025]. No entanto, nota-se também que será necessário superar barreiras como os erros quânticos, os custos elevados de implementação e ainda pensar em questões éticas no uso da computação quântica. Embora os computadores quânticos ainda estejam em fase de desenvolvimento , seu potencial para transformar o setor financeiro é evidente, especialmente no que diz respeito à otimização de portfólios, precificação de ativos e gestão de risco [Ciacco et al. 2025].

3.6. Considerações Finais

Diante das ameaças da tecnologia emergente de Computação Quântica aos mecanismos criptográficos atuais, sobretudo aqueles baseados nos problemas de logaritmo discreto e de fatoração de números inteiros (e.g., cifras assimétricas como o RSA), este minicurso teve como objetivo introduzir os fundamentos da computação quântica e examinar seus impactos em criptografia, suportando um aprofundamento maior nas razões para esses impactos a partir do entendimento de seus fundamentos matemáticos, além de fornecer uma visão mais abrangente sobre possíveis soluções além da criptografia pós-quântica, que são os principais diferenciais em relação aos minicursos apresentados em edições anteriores do SBSeg [Barreto et al. 2013, Paiva et al. 2023].

Este material abordou fundamentos da criptografia clássica na Seção 3.2, passando pelos princípios da computação quântica na Seção 3.3, até os potenciais ataques quânticos sobre algoritmos criptográficos atuais e as soluções em desenvolvimento para mitigar esses riscos na Seção 3.4, buscando suportar uma compreensão não só dos impac-

tos associados, mas também de suas justificativas. Caso a Computação Quântica venha a impactar outras classes de problemas atuais considerados intratáveis em computadores clássicos, este entendimento detalhado pode fomentar novas perspectivas de exploração de outros problemas matemáticos em novos algoritmos de criptografia pós-quântica, ou até novas abordagens como a criptografia quântica. De forma complementar, a Seção 3.5 traz algumas oportunidades de aplicações de Computação Quântica para mostrar que esta tecnologia também fomenta inovação.

Um aspecto não menos importante e que vale para reflexão é sobre a ética associada à Computação Quântica, considerando que devemos endereçar o uso de recursos como hardware, software, infraestrutura e sistemas de informação em ciência, engenharia e tecnologia para as pessoas físicas e jurídicas. Além das táticas de resistir apresentadas neste texto, o NIST também recomenda as táticas de Detectar e Recuperar [Johnson 2016, Pascoe 2023]. Os ataques são elaborados e executados com sofisticação por parte daqueles que não tem nenhum compromisso ético, e que fazem parte do nosso convívio. Podem estar próximos como membros de equipes internas ou fornecedores atacando ou até cooptando colaboradores por engenharia social para obter dados (e.g., credenciais) que os aproximam dos ativos (e.g., ativos criptográficos[2]).

O minicurso é resultado de esforços de capacitação em Computação Quântica que ocorrem no Banco Bradesco, como disciplina de Pós-Graduação na Escola Politécnica da Universidade de São Paulo (USP), e em cursos optativos para a Graduação em Ciência da Computação, e de especialização no Instituto de Tecnologia e Liderança (Inteli). Para o uso desta tecnologia emergente, a formação de mão-de-obra especializada é algo essencial, e iniciativas de educação como este minicurso se tornam relevantes, sobretudo considerando que 2025 foi eleito como o Ano Internacional da Ciência e Tecnologia Quânticas pelas Nações Unidas [Bongs 2025]. Espera-se que este material possa fomentar novos cursos, workshops e outros formatos de capacitação nesta área.

Referências

[44 U.S. Code § 3542 – Definitions 2013] 44 U.S. Code § 3542 – Definitions (2013). United states code. U.S. Code. Disponível em: `https://www.govinfo.gov/app/details/USCODE-2013-title44/USCODE-2013-title44-chap35-subchapIII-sec3542`. Acesso em: 15 jul. 2025.

[Aggarwal et al. 2017] Aggarwal, D., Brennen, G. K., Lee, T., Santha, M., and Tomamichel, M. (2017). Quantum attacks on bitcoin, and how to protect against them. Technical report, arXiv preprint arXiv:1710.10377. `https://arxiv.org/abs/1710.10377`.

[Ajtai 1996] Ajtai, M. (1996). Generating hard instances of lattice problems. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 99–108.

[2] `https://github.com/IBM/CBOM`

[Alagic et al. 2022] Alagic, G., Alagic, G., Apon, D., Cooper, D., Dang, Q., Dang, T., Kelsey, J., Lichtinger, J., Liu, Y.-K., Miller, C., et al. (2022). Status report on the third round of the nist post-quantum cryptography standardization process.

[Alagic et al. 2025] Alagic, G., Bros, M., Ciadoux, P., Cooper, D., Dang, Q., Dang, T., Kelsey, J., Lichtinger, J., Liu, Y.-K., Miller, C., et al. (2025). *Status report on the fourth round of the nist post-quantum cryptography standardization process*. US Department of Commerce, National Institute of Standards and Technology.

[Aono et al. 2022] Aono, Y., Liu, S., Tanaka, T., Uno, S., Meter, R. V., Shinohara, N., and Nojima, R. (2022). The present and future of discrete logarithm problems on noisy quantum computers. *IEEE Transactions on Quantum Engineering*, 3:1–21.

[Aumasson 2017] Aumasson, J.-P. (2017). *Serious Cryptography: A Practical Introduction to Modern Encryption*. No Starch Press, USA.

[Barker et al. 2020] Barker, E., Roginsky, A., and National Institute of Standards and Technology (NIST) (2020). Recommendation for key management: Part 1 – general (revision 5). Technical Report NIST SP 800-57pt1r5, National Institute of Standards and Technology. Supersedes NIST SP 800-57 Part 1 Rev. 4 (2016).

[Barreto et al. 2013] Barreto, P., Biasi, F. P., Dahab, R., César, J., Pereira, G., and Ricardini, J. E. (2013). Introdução à criptografia pós-quântica. *Minicursos do XIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais, SBSeg*.

[Beast 2024] Beast, A. (2024). BIP-360: Pay to Quantum Resistant Hash (P2QRH). `https://github.com/bitcoin/bips/pull/1670`.

[Bernstein et al. 2009] Bernstein, D. J., Buchmann, J., and Dahmen, E., editors (2009). *Post-Quantum Cryptography*. Mathematics and Statistics. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 1 edition. eBook ISBN: 978-3-540-88702-7.

[Beullens et al. 2021] Beullens, W., D'Anvers, J.-P., Hülsing, A. T., Lange, T., Panny, L., de Saint Guilhem, C., and Smart, N. P. (2021). Post-quantum cryptography: Current state and quantum mitigation. Technical report, ENISA, Attiki, Greece.

[Biamonte et al. 2017] Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., and Lloyd, S. (2017). Quantum machine learning. *Nature*, 549(7671):195–202.

[Bongs 2025] Bongs, K. (2025). Celebrating the international year of quantum science and technology.

[Bonneau et al. 2015] Bonneau, J., Miller, A., Clark, J., Narayanan, A., Kroll, J. A., and Felten, E. W. (2015). Sok: Research perspectives and challenges for bitcoin and cryptocurrencies. In *2015 IEEE Symposium on Security and Privacy*, pages 104–121.

[Boudot et al. 2020a] Boudot, F., Gaudry, P., Guillevic, A., Heninger, N., Thomé, E., and Zimmermann, P. (2020a). Comparing the difficulty of factorization and discrete logarithm: a 240-digit experiment. Cryptology ePrint Archive, Paper 2020/697.

[Boudot et al. 2020b] Boudot, F., Gaudry, P., Guillevic, A., Heninger, N., Thomé, E., and Zimmermann, P. (2020b). Factorization of rsa-250.

[Brassard et al. 1998] Brassard, G., HØyer, P., and Tapp, A. (1998). *Quantum cryptanalysis of hash and claw-free functions: Invited paper*, page 163–169. Springer Berlin Heidelberg.

[Buonaiuto et al. 2023] Buonaiuto, G., Gargiulo, F., De Pietro, G., Esposito, M., and Pota, M. (2023). Best practices for portfolio optimization by quantum computing, experimented on real quantum devices. *Scientific Reports*, 13(1):19434.

[Caseli and Nunes 2024] Caseli, H. d. M. and Nunes, M. d. G. V. (2024). *Processamento de Linguagem Natural: Conceitos, Técnicas e Aplicações em Português*. Brasileiras em PLN (BPLN).

[Castryck and Decru 2023] Castryck, W. and Decru, T. (2023). An efficient key recovery attack on sidh. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 423–447. Springer.

[Chang et al. 2023] Chang, Y.-J., Sie, M.-F., Liao, S.-W., and Chang, C.-R. (2023). The prospects of quantum computing for quantitative finance and beyond. *IEEE Nanotechnology Magazine*, 17(2):31–37.

[Chen et al. 2017] Chen, L., Moody, D., and Liu, Y. (2017). Nist post-quantum cryptography standardization. *Transition*, 800(131A):164.

[Ciacco et al. 2025] Ciacco, A., Guerriero, F., and Macrina, G. (2025). Review of quantum algorithms for medicine, finance and logistics. *Soft Computing*, 29(4):2129–2170.

[Cooper et al. 2020] Cooper, D. A., Apon, D. C., Dang, Q. H., Davidson, M. S., Dworkin, M. J., Miller, C. A., et al. (2020). Recommendation for stateful hash-based signature schemes. *NIST Special Publication*, 800(208):800–208.

[Corallo 2024] Corallo, M. (2024). Proposal for OP_SPHINCS as Post-Quantum Signature Opcode. Bitcoin-dev mailing list.

[Correia et al. 2022] Correia, A. D., Moortgat, M., and Stoof, H. T. C. (2022). Quantum computations for disambiguation and question answering. *arXiv preprint arXiv:2106.05299*.

[Diffie and Hellman 2022] Diffie, W. and Hellman, M. E. (2022). New directions in cryptography. In *Democratizing cryptography: the work of Whitfield Diffie and Martin Hellman*, pages 365–390.

[Ding et al. 2006] Ding, J., Gower, J. E., and Schmidt, D. S. (2006). *Multivariate public key cryptosystems*. Springer.

[Dobias et al. 2025] Dobias, P., Rezaeezade, A., Chmielewski, Ł., Malina, L., and Batina, L. (2025). Sok: Reassessing side-channel vulnerabilities and countermeasures in pqc implementations. *Cryptology ePrint Archive*.

[Duim and Portácio 2023] Duim, J. L. and Portácio, R. G. (2023). Segurança criptográfica: Combinando métodos clássicos e quânticos. *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, 10(1).

[FIPS PUB 202 2015] FIPS PUB 202 (2015). SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. `https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf`. FIPS PUB 202.

[Flöther 2023] Flöther, F. F. (2023). The state of quantum computing applications in health and medicine. *Research Directions: Quantum Technologies*, 1:e10.

[Gamble 2019] Gamble, S. (2019). Quantum computing: What it is, why we want it, and how we're trying to get it. In *Frontiers of Engineering: Reports on Leading-Edge Engineering from the 2018 Symposium*. National Academies Press (US).

[Gentry 2009] Gentry, C. (2009). Fully homomorphic encryption using ideal lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 169–178.

[Grover 1996] Grover, L. K. (1996). A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, page 212–219, New York, NY, USA. Association for Computing Machinery.

[Heilman and Sabouri 2023] Heilman, E. and Sabouri, A. (2023). BIP-347: Reintroducing OP_CAT for Lamport Signatures. https://github.com/bitcoin/bips/blob/master/bip-0347.mediawiki.

[Hhan et al. 2023] Hhan, M., Yamakawa, T., and Yun, A. (2023). Quantum complexity for discrete logarithms and related problems. Cryptology ePrint Archive, Paper 2023/1054.

[Hosoyamada and Sasaki 2020] Hosoyamada, A. and Sasaki, Y. (2020). Finding hash collisions with quantum computers by using differential trails with smaller probability than birthday bound. Cryptology ePrint Archive, Paper 2020/213.

[Huang et al. 2024] Huang, Z., Wang, H., Cao, B., He, D., and Wang, J. (2024). A comprehensive side-channel leakage assessment of crystals-kyber in iiot. *Internet of Things*, 27:101331.

[Häner et al. 2020] Häner, T., Jaques, S., Naehrig, M., Roetteler, M., and Soeken, M. (2020). Improved quantum circuits for elliptic curve discrete logarithms. Cryptology ePrint Archive, Paper 2020/077.

[Jao and De Feo 2011] Jao, D. and De Feo, L. (2011). Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In *International workshop on post-quantum cryptography*, pages 19–34. Springer.

[Jedlicka et al. 2022] Jedlicka, P., Malina, L., Socha, P., Gerlich, T., Martinasek, Z., and Hajny, J. (2022). On secure and side-channel resistant hardware implementations of post-quantum cryptography. In *Proceedings of the 17th International Conference on Availability, Reliability and Security*, pages 1–9.

[Johnson 2016] Johnson, C. (2016). Guide to cyber threat information sharing. *NIST Special Publication*, pages 800–150.

[Joseph et al. 2022] Joseph, D., Misoczki, R., and Manzano, M. e. a. (2022). Transitioning organizations to post-quantum cryptography. *Nature*, 605:237–243.

[Kasirajan 2021] Kasirajan, V. (2021). *Fundamentals of Quantum Computing: Theory and Practice*. Springer Cham.

[Katz and Lindell 2014] Katz, J. and Lindell, Y. (2014). *Introduction to Modern Cryptography, Second Edition*. Chapman & Hall/CRC, 2nd edition.

[Khan et al. 2024] Khan, S., Krishnamoorthy, P., Goswami, M., Rakhimjonovna, F. M., Mohammed, S. A., and Menaga, D. (2024). Quantum computing and its implications for cybersecurity: A comprehensive review of emerging threats and defenses. *Nanotechnology Perceptions*, 20:S13.

[Khang et al. 2025] Khang, A., Rath, K. C., Madapana, K., Rao, J., Panda, L. P., and Das, S. (2025). Quantum computing and portfolio optimization in finance services. In *Shaping Cutting-Edge Technologies and Applications for Digital Banking and Financial Services*, pages 27–45. Productivity Press.

[Knuth 1997] Knuth, D. E. (1997). *The art of computer programming, volume 1 (3rd ed.): fundamental algorithms*. Addison Wesley Longman Publishing Co., Inc., USA.

[Kosmann-Schwarzbach and Singer 2010] Kosmann-Schwarzbach, P. Y. and Singer, S. F. (2010). *Lie Groups SU(2) and SO(3)*, pages 71–80. Springer New York, New York, NY.

[Kumar et al. 2021] Kumar, Y., Koul, A., Sisodia, P. S., Shafi, J., Verma, K., Gheisari, M., and Davoodi, M. B. (2021). Heart failure detection using quantum-enhanced machine learning and traditional machine learning techniques for internet of artificially intelligent medical things. *Wireless Communications and Mobile Computing*, 2021(1):1616725.

[Lamport 1981] Lamport, L. (1981). Password authentication with insecure communication. *Communications of the ACM*, 24(11):770–772.

[Lang et al. 2022] Lang, J., Zielinski, S., and Feld, S. (2022). Strategic portfolio optimization using simulated, digital, and quantum annealing. *Applied Sciences*, 12(23):12288.

[Lyubashevsky et al. 2010] Lyubashevsky, V., Peikert, C., and Regev, O. (2010). On ideal lattices and learning with errors over rings. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 1–23. Springer.

[Marquezino and Helayel-Neto 2003] Marquezino, F. and Helayel-Neto, J. (2003). Estudo introdutório do protocolo quântico bb84 para troca segura de chaves. *Centro Brasileiro de Pesquisas Fısicas, Série Monografias*.

[Maxwell et al. 2020] Maxwell, G., Poelstra, A., Seurin, Y., and Wuille, P. (2020). BIP-340: Schnorr Signatures for secp256k1. `https://github.com/bitcoin/bips/blob/master/bip-0340.mediawiki`.

[McEliece 1978] McEliece, R. J. (1978). A public-key cryptosystem based on algebraic. *Coding Thv*, 4244(1978):114–116.

[Mendes et al. 2011] Mendes, Á. J. B., Paulicena, E. H., and Souza, W. A. R. d. (2011). Criptografia quântica: uma abordagem direta. *Revista de Sistema de Informação da FSMA*, (7):39–48.

[Menezes et al. 2018] Menezes, A. J., Van Oorschot, P. C., and Vanstone, S. A. (2018). *Handbook of applied cryptography*. CRC press.

[Merkle 1979] Merkle, R. C. (1979). *Secrecy, authentication, and public key systems.* Stanford university.

[Milton and Shikhelman 2025] Milton, A. and Shikhelman, C. (2025). Bitcoin and the quantum threat: A comprehensive risk analysis. Technical report, Chaincode Labs Research Report. `https://chaincode.com/bitcoin-post-quantum.pdf`.

[Miyamoto 2022] Miyamoto, K. (2022). Quantum algorithm for calculating risk contributions in a credit portfolio. *EPJ Quantum Technology*, 9(1):1–16.

[Moody et al. 2024] Moody, D., Perlner, R., Regenscheid, A., Robinson, A., and Cooper, D. (2024). Transition to post-quantum cryptography standards. Technical report, National Institute of Standards and Technology.

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

[Nakamoto 2008] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. `https://bitcoin.org/bitcoin.pdf`.

[Narayanan et al. 2016] Narayanan, A., Bonneau, J., Felten, E., Miller, A., and Goldfeder, S. (2016). *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press.

[National Institute of Standards and Technology (NIST) 2024] National Institute of Standards and Technology (NIST) (2024). Post-quantum cryptography standardization project – finalist algorithms. Technical report, US Department of Commerce. `https://csrc.nist.gov/projects/post-quantum-cryptography`.

[Nausheen et al. 2025] Nausheen, F., Ahmed, K., and Khan, M. I. (2025). Quantum natural language processing: A comprehensive review of models, methods, and applications. *arXiv preprint arXiv:2504.09909*. Preprint.

[Nielsen and Chuang 2010] Nielsen, M. A. and Chuang, I. L. (2010). *Quantum computation and quantum information*. Cambridge university press.

[NSA 2021] NSA (2021). *Frequently Asked Questions: Quantum Computing and Post-Quantum Cryptography*. National Security Agency.

[of Standards et al. 2023] of Standards, N. I., (NIST), T., Dworkin, M. J., Turan, M. S., and Mouha, N. (2023). Advanced encryption standard (aes).

[Omar and El-Hafeez 2023] Omar, A. and El-Hafeez, T. A. (2023). Quantum computing and machine learning for arabic language sentiment classification in social media. *Scientific Reports*, 13(1):17305.

[Paar et al. 2024] Paar, C., Pelzl, J., and Güneysu, T. (2024). *Understanding cryptography: from established symmetric and asymmetric ciphers to post-quantum algorithms*. Springer Nature.

[Paiva et al. 2023] Paiva, T. B., Ponciano, V., Moreira, E., Oliveira, R., Rufino, V., Lima, C., López, J., Ueda, E., and Terada, R. (2023). Explorando esquemas criptográficos pós-quânticos considerados pelo nist com implementação em sage. *Anais*.

[Pascoe 2023] Pascoe, C. E. (2023). Public draft: The nist cybersecurity framework 2.0. *National Institute of Standards and Technology*.

[Prousalis and Konofaos 2019] Prousalis, K. and Konofaos, N. (2019). A quantum pattern recognition method for improving pairwise sequence alignment. *Scientific Reports*, 9(1):7226.

[Ravi et al. 2024] Ravi, P., Chattopadhyay, A., D'Anvers, J. P., and Baksi, A. (2024). Side-channel and fault-injection attacks over lattice-based post-quantum schemes (kyber, dilithium): Survey and new results. *ACM Transactions on Embedded Computing Systems*, 23(2):1–54.

[Rebentrost and Lloyd 2024] Rebentrost, P. and Lloyd, S. (2024). Quantum computational finance: quantum algorithm for portfolio optimization. *KI-Künstliche Intelligenz*, pages 1–12.

[Regev 2005] Regev, O. (2005). On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '05, page 84–93, New York, NY, USA. Association for Computing Machinery.

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

[Robert et al. 2021] Robert, A., Barkoutsos, P. K., Woerner, S., et al. (2021). Resource-efficient quantum algorithm for protein folding. *npj Quantum Information*, 7(1):38.

[Roetteler et al. 2017] Roetteler, M., Naehrig, M., Svore, K. M., and Lauter, K. (2017). Quantum resource estimates for computing elliptic curve discrete logarithms. Cryptology ePrint Archive, Paper 2017/598.

[Saarinen 2022] Saarinen, M.-J. O. (2022). Wip: Applicability of iso standard side-channel leakage tests to nist post-quantum cryptography. In *2022 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pages 69–72. IEEE.

[Sakurai and Napolitano 2020] Sakurai, J. J. and Napolitano, J. (2020). *Modern Quantum Mechanics (3rd ed.)*. Cambridge University Press.

[Salvi 2023] Salvi, P. (2023). How to calculate big o notation time complexity.

[Sarah and Peter 2024] Sarah, D. and Peter, C. (2024). *On the practical cost of Grover for AES key recovery*. UK National Cyber Security Centre.

[Sarkar et al. 2021] Sarkar, A., Al-Ars, Z., Almudever, C. G., and Bertels, K. L. M. (2021). Qibam: Approximate sub-string index search on quantum accelerators applied to dna read alignment. *Electronics*, 10(19).

[Shafique et al. 2024] Shafique, M. A., Munir, A., and Latif, I. (2024). Quantum computing: Circuits, algorithms, and applications. *IEEE Access*, 12:22296–22314.

[Shah et al. 2025] Shah, P., Prajapati, P., and Patel, D. (2025). Lattice-based post quantum cryptography using variations of learning with error (lwe). In Patel, K. K., Santosh, K., Gomes de Oliveira, G., Patel, A., and Ghosh, A., editors, *Soft Computing and Its Engineering Applications*, pages 58–72, Cham. Springer Nature Switzerland.

[Shahwar et al. 2022] Shahwar, T., Zafar, J., Almogren, A., Zafar, H., Rehman, A. U., Shafiq, M., and Hamam, H. (2022). Automated detection of alzheimer's via hybrid classical quantum neural networks. *Electronics*, 11(5).

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

[Shor 1997] Shor, P. W. (1997). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509.

[Singh 1999] Singh, S. (1999). *The Code Book: The Evolution of Secrecy from Mary, Queen of Scots, to Quantum Cryptography*. Doubleday, USA, 1st edition.

[Sipser 2012] Sipser, M. (2012). *Introduction to the Theory of Computation, 3rd Edition*. Thomson Course Technology.

[Smite-Meister 2023] Smite-Meister (2023). Bloch sphere.

[Smythe 2021] Smythe, W. (2021). Qm 101: Bloch sphere.

[Stallings 2013] Stallings, W. (2013). *Cryptography and Network Security: Principles and Practice*. Prentice Hall Press, USA, 6th edition.

[Susskind and Friedman 2014] Susskind, L. and Friedman, A. (2014). *Quantum Mechanics: The Theoretical Minimum*. Basic Books.

[Takagi 2003] Takagi, N. H. (2003). Fundamentos matemáticos da criptografia quântica.

[Terada 2008] Terada, R. (2008). *Segurança de dados: criptografia em redes de computador*. Edgard Blücher, São Paulo, 1 edition. 1ª reimpressão: 2011.

[Veselỳ 2022] Veselỳ, M. (2022). Application of quantum computers in foreign exchange reserves management. *arXiv preprint arXiv:2203.15716*.

[Wang et al. 2023] Wang, R., Ngo, K., Gärtner, J., and Dubrova, E. (2023). Single-trace side-channel attacks on crystals-dilithium: Myth or reality? *Cryptology ePrint Archive*.

[Wang et al. 2004] Wang, X., Feng, D., Lai, X., and Yu, H. (2004). Collisions for hash functions md4, md5, haval-128 and ripemd. *Cryptology ePrint Archive*.

[Wang et al. 2005] Wang, X., Yin, Y. L., and Yu, H. (2005). Finding collisions in the full sha-1. In *Annual international cryptology conference*, pages 17–36. Springer.

[Watrous 2025] Watrous, J. (2025). Understanding quantum information and computation.

[Woerner and Egger 2019] Woerner, S. and Egger, D. J. (2019). Quantum risk analysis. *npj Quantum Information*, 5(1):15.

[Zhang et al. 2011] Zhang, M., Xi, Z., and Wei, J.-H. (2011). Manipulating quantum information on the controllable systems or subspaces.

[Zhao et al. 2023] Zhao, Y., Pan, S., Ma, H., Gao, Y., Song, X., He, J., and Jin, Y. (2023). Side channel security oriented evaluation and protection on hardware implementations of kyber. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 70(12):5025–5035.