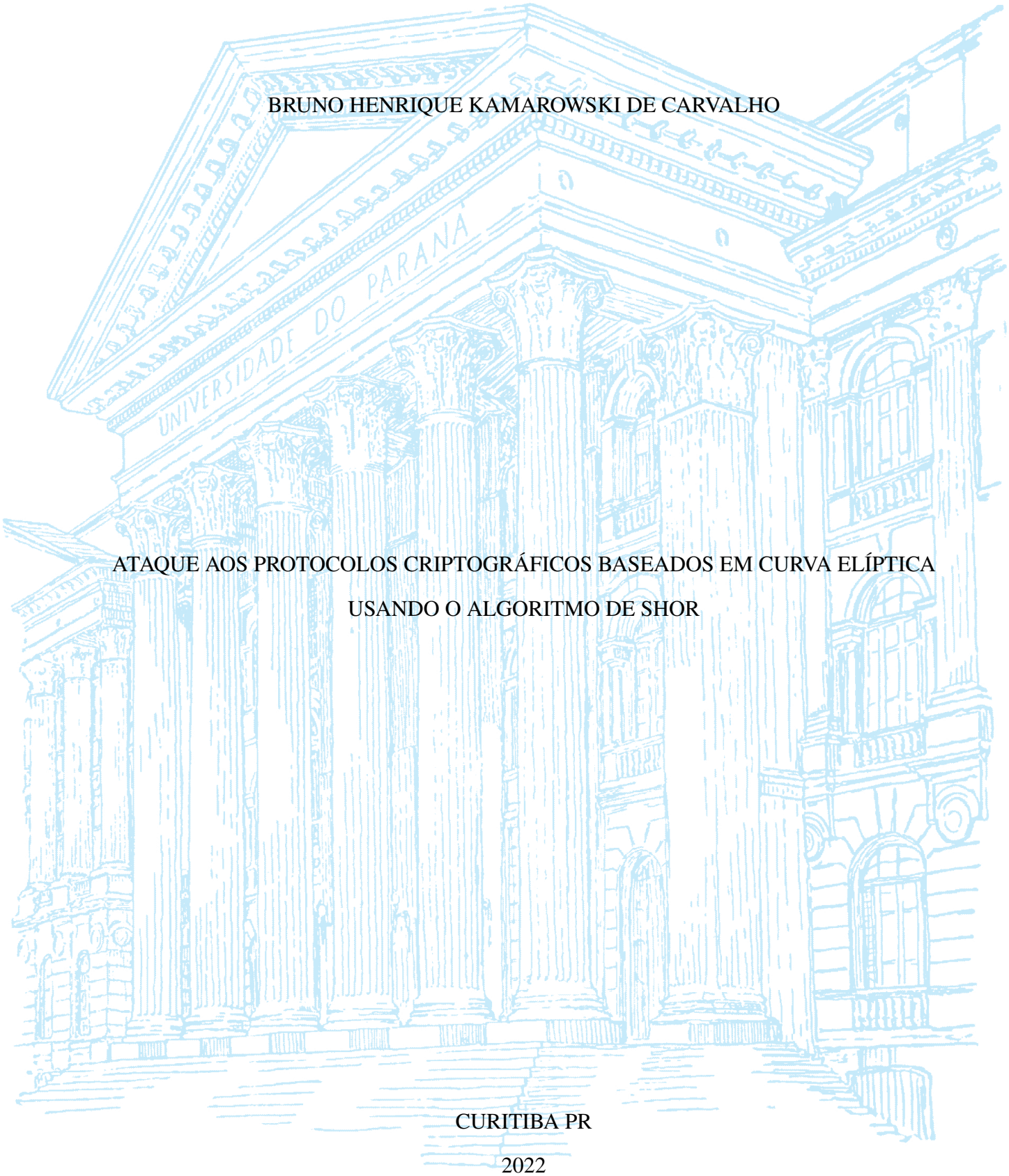
UNIVERSIDADE FEDERAL DO PARANÁ

BRUNO HENRIQUE KAMAROWSKI DE CARVALHO



ATAQUE AOS PROTOCOLOS CRIPTOGRÁFICOS BASEADOS EM CURVA ELÍPTICA USANDO O ALGORITMO DE SHOR

CURITIBA PR

2022

BRUNO HENRIQUE KAMAROWSKI DE CARVALHO

ATAQUE AOS PROTOCOLOS CRIPTOGRÁFICOS BASEADOS EM CURVA ELÍPTICA USANDO O ALGORITMO DE SHOR

Trabalho apresentado como requisito parcial à conclusão do Curso de Bacharelado em Ciência da Computação, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: Murilo Silva.

CURITIBA PR

2022

Ficha catalográfica

Substituir o arquivo `0-iniciais/catalografica.pdf` pela ficha catalográfica fornecida pela Biblioteca da UFPR (PDF em formato A4).

Instruções para obter a ficha catalográfica e fazer o depósito legal da tese/dissertação (contribuição de André Hochuli, abril 2019):

1. Estas instruções se aplicam a dissertações de mestrado e teses de doutorado. Trabalhos de conclusão de curso de graduação e textos de qualificação não precisam segui-las.
2. Verificar se está usando a versão mais recente do modelo do PPGInf e atualizar, se for necessário (`https://gitlab.c3sl.ufpr.br/maziero/tese`).
3. conferir o *checklist* de formato do Sistema de Bibliotecas da UFPR, em `https://portal.ufpr.br/teses_servicos.html`.
4. Enviar e-mail para "`referencia.bct@ufpr.br`" com o arquivo PDF da dissertação/tese, solicitando a respectiva ficha catalográfica.
5. Ao receber a ficha, inseri-la em seu documento (substituir o arquivo `0-iniciais/catalografica.pdf` do diretório do modelo).
6. Emitir a Certidão Negativa (CND) de débito junto a biblioteca (`https://www.portal.ufpr.br/cnd.html`).
7. Avisar a secretaria do PPGInf que você está pronto para o depósito. Eles irão mudar sua titulação no SIGA, o que irá liberar uma opção no SIGA pra você fazer o depósito legal.
8. Acesse o SIGA (`http://www.prppg.ufpr.br/siga`) e preencha com cuidado os dados solicitados para o depósito da tese.
9. Aguarde a confirmação da Biblioteca.
10. Após a aprovação do pedido, informe a secretaria do PPGInf que a dissertação/tese foi depositada pela biblioteca. Será então liberado no SIGA um link para a confirmação dos dados para a emissão do diploma.

Ficha de aprovação

Substituir o arquivo 0-iniciais/aprovacao.pdf pela ficha de aprovação fornecida pela secretaria do programa, em formato PDF A4.

Dedico este trabalho à minha querida irmã, que tanto admiro e que sempre esteve ao meu lado.

AGRADECIMENTOS

Ao professor Murilo, por ter sido meu orientador e me guiar durante toda a realização deste trabalho. Aos meus familiares por me incentivarem a me dedicar aos estudos e me concederem condições para tal. Aos meus amigos pelas palavras de apoio e por todos os momentos juntos. À minha namorada que me motivou, me ajudou em todos os momentos difíceis.

RESUMO

Este trabalho explora uma variação do algoritmo de Shor capaz de resolver o problema do logaritmo discreto para curvas elípticas e com isso quebrar protocolos criptográficos baseados em curvas elípticas. Para isso é formalizada a ideia da criptografia simétrica e assimétrica, também é mostrada a definição de curvas elípticas, como elas são usadas para geração de chaves públicas e como o problema do logarítmo discreto é a base matemática de protocolos criptográficos baseados em curva elítptica.

Além disso, esse trabalho apresenta conceitos básicos do modelo de computação quântica, algumas notações usuais da área e como usar o modelo de computação quântica para criar algoritmos que resolvem problemas computacionais de maneira mais eficiente que os melhores algoritmos clássicos conhecidos. Por último, é mostrado como usar uma variação do algoritmo de Shor para resolver o problema do logaritmo discreto para curvas elípticas.

Palavras-chave: Computação Quântica, Criptografia, Curva Elíptica

ABSTRACT

The use of cryptography is essential to guarantee the security of the information that circulates in public, so that mathematical mechanisms are used to define protocols cryptographic Some of these protocols use the concept of elliptic curve points over a finite field because one of its interesting properties is that there is an efficient algorithm that allows easy computing operations with curve points but no known algorithm efficient classic that allows to do the opposite way and in view of these properties it is stated the problem of the discrete logarithm in elliptic curves which is the mathematical principle guarantees the security of encryptions that use elliptic curves as a mechanism for generating asymmetric keys.

Besides that, in the mid-1980s, computing models were proposed. that use quantum properties, it was then discovered that this model allows the creation of efficient algorithms for problems not known efficient classical algorithm capable to solve. Shor's algorithm is one of the best known algorithms that uses the model quantum computing, it allows solving the discrete logarithm problem in curves ellipticals efficiently and therefore threatens the security of systems that employ elliptic curve based cryptographics.

This work uses a bibliographic base referring to the quantum computing model and elliptic curve cryptography to demonstrate how to use Shor's algorithm to calculate the private keys of elliptic curve-based cryptographic protocols.

Keywords: Quantum Computing. Cryptography. Eliptic Curve

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ACRÔNIMOS

PEF	Problema de estimação de fase
PEA	Problema de estimação do autovalor
PLDCE	Problema do logaritmo discreto para curvas elípticas
PLDCEP	Problema do logaritmo discreto para curvas elípticas onde r é primo
MMC	Mínimo múltiplo comum
MDC	Maior divisor comum

LISTA DE SÍMBOLOS

e	Número de Euler
i	$\sqrt{-1}$
$\sum$	Símbolo matemático de uma sequência de somas
$\otimes$	Símbolo matemático para produto tensorial
$\equiv$	Símbolo matemático para congruência modular
mod	Símbolo matemático para operação modular
log	Símbolo matemático para logaritmo na base 2
sin	Símbolo matemático para a função seno
$\in$	Símbolo matemático para pertence
$\mapsto$	Símbolo matemático de mapeamento
$\mathbb{Z}$	Conjunto dos números inteiros
$\mathbb{Z}_k$	Conjunto dos números inteiros não negativos até k
$\mathbb{Z}_{>0}$	Conjunto dos números inteiros e positivos
$\mathbb{R}$	Conjunto dos números reais
$\mathbb{C}$	Conjunto dos números complexos
α	alfa, primeira letra do alfabeto grego
β	beta, segunda letra do alfabeto grego
δ	delta, quarta letra do alfabeto grego
λ	lambda, décima letra do alfabeto grego
ρ	rho, décima sétima letra do alfabeto grego
ϕ	phi, vigésima letra do alfabeto grego
ψ	psi, penúltima letra do alfabeto grego
Ψ	psi, penúltima letra do alfabeto grego
ω	ômega, última letra do alfabeto grego

SUMÁRIO

1 INTRODUÇÃO

Após a demonstração do modelo de criptografia assimétrico por Whitfeld Diffie e Martin Hellman em 1976 (Diffie e Hellman, 1976), diversos objetos matemáticos foram propostos para serem usados como mecanismo para geração de chaves assimétricas, dentre eles as curvas elípticas sobre corpos finitos, proposta de maneira independente por Victor Miller e Neal Keblitz em 1985 (Miller, 1985; Koblitz, 1987). Com a crença de que os sistemas criptográficos baseados em curvas elípticas proveem o mesmo nível de segurança dos protocolos baseados em outros grupos e somado ao fato de que as chaves geradas por curvas elípticas são muito pequenas, os estudos em cima de curvas elípticas aumentaram significativamente. Atualmente, protocolos baseados em curvas elípticas são amplamente utilizados, principalmente em dispositivos com baixo poder de processamento. No Capítulo 2 é mostrado algumas estruturas matemáticas que são usadas no Capítulo 5, neste são apresentadas curvas elípticas sobre um corpo finito, a maneira de usar esse objeto matemático como mecanismo para geração de chaves assimétricas e também é apresentado o problema do logaritmo discreto para curvas elípticas como sendo a base matemática para a segurança desses sistemas.

Em paralelo à proposta de criptografias assimétricas, o modelo de computação quântico era proposto. Esse modelo conta com propriedades quânticas, inexistentes em computadores clássicos que são exploradas no Capítulo 3. Neste capítulo também é apresentado o modelo de computação quântica, passando por definições, propriedades fundamentais, notações usuais, portas e circuitos quânticos.

Em 1994, Peter Shor propôs o algoritmo de Shor (Shor, 1999), que utiliza o modelo de computação quântica para resolve o problema de fatoração de inteiros e o problema do logaritmo discreto de maneira eficiente. Neste contexto, no Capítulo 4 são apresentados alguns algoritmos quânticos que resolvem partes do problema do logaritmo discreto enquanto no Capítulo 6 é apresentada uma variação do algoritmo de Shor capaz de resolver o problema do logarítmo discreto para curvas elípticas.

Tendo em vista a ideia de que o modelo de computação quântico permite a criação de algoritmos capazes de resolver eficientemente problemas que não existe algoritmo clássico eficiente conhecido, temos que o objetivo deste estudo é o de mostrar como uma variação do algorítmo de Shor é capaz de resolver o problema do logaritmo discreto para curvas elípticas e com isso quebrar protocolos criptográficos baseados em curva elíptica.

2 PRELIMINARES MATEMÁTICOS

Para melhor compreensão deste trabalho, será apresentado neste capítulo operações de aritmética modular e definições de algumas estruturas algébricas que serão usadas nos capítulos seguintes.

O conteúdo da seção 2.1 foi baseado no livro (Weil, 2013) a prova do teorema da seção 2.2 pode ser encontrada no livro (Scott, 2012), o conteúdo desta seção foi baseado nos livros (Scott, 2012; Beachy e Blair, 2019).

2.1 ARITMÉTICA MODULAR

Seja $n \in \mathbb{Z}_{>0}$ e $a, b \in \mathbb{Z}$, é dito que a é congruente a b módulo n se existe $k \in \mathbb{Z}$ tal que $a = b + kn$. A congruência módulo n é denotada por

$$a \equiv b \ (mod\ n).$$

Se a é congruente a b módulo n, então o resto da divisão de a por n é igual ao resto da divisão de b por n.

Seja $a \in \mathbb{Z}$ e $n \in \mathbb{Z}_{>0}$, o inverso multiplicativo de a módulo n, denotado por a^{-1} é o número inteiro tal que $aa^{-1} \equiv 1 \ (mod\ n)$. O inverso multiplicativo de um número pode ser calculado eficientemente utilizando o algoritmo de Euclides extendido, que não será apresentado neste estudo.

2.2 ESTRUTURAS ALGÉBRICAS

Definição 2.2.1 *Um grupo é uma estrutura algébrica composta por um par ordenado $(G, *)$, onde G é um conjunto de elementos e $*$ é uma operação binária fechada em G tal que*

- *A operação $*$ é associativa, isto é, $(a * b) * c = a * (b * c)$, para $a, b, c \in G$*
- *G possui um elemento $'e'$ que é o elemento neutro da operação $*$*
- *Para todo $a \in G$ existe a^{-1} tal que $a * a^{-1} = e$.*

Quando o contexto é claro, é comum nos referirmos ao grupo $(G, *)$ apenas por G.

Se a operação do grupo também é comutativa, então esse grupo é dito ser abeliano. Se a operação do grupo for a operação de adição, independentemente de quais elementos compõem o conjunto do grupo, então é comum dizer que este grupo é um grupo aditivo. Se for a operação de multiplicação, é dito que este grupo é multiplicativo.

Além disso, seja o par $(G, *)$ um grupo, $a \in G$ e $r \in \mathbb{Z}$ o resultado de $a * a * ... * a$ onde a ocorre r vezes é denotado por ra ou a^r.

Definição 2.2.2 *Seja $(G, *)$ um grupo onde $a \in G$ e cujo elemento neutro da operação é $'e'$. A ordem do elemento a é o menor inteiro n tal que $a^n = e$.*

Definição 2.2.3 *Seja $(G, *)$ um grupo e $H \subseteq G$ tal que $(H, *)$ forma um grupo, então $(H, *)$ é dito subgrupo de $(G, *)$.*

Definição 2.2.4 *Seja $(G, *)$ um grupo e $a \in G$ cuja ordem de a é n, então a é dito gerador do grupo $(H, *)$ onde $H = a^i, i \in [1..n]$.*

A ordem de um grupo é a quantidade de elementos do conjunto do grupo.

Teorema 2.2.5 *Seja G um grupo de ordem p e a um elemento de G de ordem r, então $p > r$.*

Definição 2.2.6 *Um corpo é uma estrutura algébrica composta pela tripla $(F, +, \cdot)$, onde F é um conjunto de elementos, $+$ e $\cdot$ são operações binárias fechadas em F tais que*

- *As operações do corpo são ambas associativas e comutativas sobre F*
- *A operação $\cdot$ é distributiva sobre operação $+$, isto é, $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$, com $a, b, c \in F$*
- *Existem elementos $e, o \in F$ onde $e \neq o$ tais que para qualquer $a \in F$*
  1. $a + o = a$
  2. $a \cdot e = a$
- *Para qualquer $a \in F$ existe elementos $a^{-1}, -a \in F$ tais que*
  1. $a + (-a) = 0$
  2. $a \cdot a^{-1} = e$

Tipicamente nos referimos às operações $+$ e $\cdot$ como adição e multiplicação, respectivamente, porém ressaltamos que um corpo pode possuir qualquer operação algébrica sobre os elementos do corpo que satisfaçam as propriedades citadas.

Definição 2.2.7 *A característica de um corpo $(F, +, \cdot)$ é o menor número n tal que n somas consecutivas do elemento neutro da multiplicação resulta no elemento neutro da adição*

Neste trabalho, será usado a notação $\mathbb{F}_k$ onde $k \in \mathbb{Z}_{>0}$ para se referir ao corpo definido pela tripla $(\mathbb{Z}_k, +, \cdot)$ onde as operações $+$ e $\cdot$ são as operações adição módulo k e multiplicação módulo k, respectivamente.

3 FUNDAMENTOS DA COMPUTAÇÃO QUÂNTICA

Computação quântica é um ramo da computação teórica que explora propriedades de estados quânticos como superposição e emaranhamento, permitindo obter uma redução significativa da complexidade computacional de problemas tipicamente difíceis como fatoração de inteiros, um problema para qual não existe atualmente algoritmos clássicos eficientes (De Wolf, 2017). Os melhores algoritmos conhecidos para solucionar esses problemas requerem a análise de uma grande quantidade de combinações, que com computadores clássicos seriam computadas de maneira a exaurir todas as possibilidades enquanto em computadores quânticos são representados em espaços multidimensionais capazes de representar múltiplas combinações juntas, que podem fornecer informação de maneira menos custosa computacionalmente, mitigando o efeito de análise exaustiva de todas as possibilidades.

Esse capítulo foi construído com base no conteúdo em (Kaye et al., 2006),(Nielsen e Chuang, 2002),(Cleve et al., 1998) e (VAZIRANI, 2013).

3.1 BITS E QUBITS

Um bit clássico pode possuir dois valores que podem ser 0 ou 1. Além disso, quando se mede o valor de um bit clássico, o mesmo não é modificado. No caso quântico, usa-se qubits, que possuem comportamento diferente de um bit clássico pois podem estar em sobreposição, o que significa que está representando dois estados ao mesmo tempo e ao mensurá-lo o seu valor colapsa para um dos estados.

3.2 REPRESENTAÇÃO DE QUBITS

Para se descrever sistemas quânticos usualmente é usada a notação de Dirac, também conhecida como notação bra-ket, que usa os seguintes símbolos para descrever um estado quântico:

- ket: $|a\rangle := \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$
- bra: $\langle b| := \begin{bmatrix} b_1^* & b_2^* \end{bmatrix}$
- bra-ket: $\langle b|a\rangle := a_1 b_1^* + a_2 b_2^*$

Os possíveis estados, chamados de bases do sistema quântico, são descritos como vetores ortonormais. Usualmente são usados os estados $|0\rangle$ e $|1\rangle$, descritos a seguir:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad\qquad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Como dito anteriormente, um qubit pode estar representando dois estados, "0"e "1", ao mesmo tempo. Essa dualidade de estados é descrita em forma de uma amplitude de probabilidade associada a cada estado. Isto é, é possível associar um número complexo à cada possível estado do sistema quântico e com isso podemos escrever um estado $|\psi\rangle$ da seguinte forma:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \text{ onde } \alpha, \beta \in \mathbb{C} \text{ e } |\alpha|^2 + |\beta|^2 = 1.$$

É possível descrever um sistema com mais de um qubit, e portanto mais estados possíveis, aplicando o produto tensorial entre os vetores que representam os qubits. Assim um sistema quântico de dois qubits tem a seguinte base:

$$|00\rangle = \begin{bmatrix}1\\0\\0\\0\end{bmatrix} \qquad |01\rangle = \begin{bmatrix}0\\1\\0\\0\end{bmatrix} \qquad |10\rangle = \begin{bmatrix}0\\0\\1\\0\end{bmatrix} \qquad |10\rangle = \begin{bmatrix}0\\0\\1\\0\end{bmatrix} \qquad |11\rangle = \begin{bmatrix}0\\0\\0\\1\end{bmatrix}.$$

O estado quântico $|\psi\rangle$ formado pelos qubits $|\psi_1\rangle$ e $|\psi_2\rangle$ é obtido pelo produto tensorial de $|\psi_1\rangle$ e $|\psi_2\rangle$, isto é, com os estados $|\psi_1\rangle$ e $|\psi_2\rangle$ descritos como:

$$|\psi_1\rangle = \alpha_1|0\rangle + \beta_1|1\rangle \qquad\qquad |\psi_2\rangle = \alpha_2|0\rangle + \beta_2|1\rangle.$$

Tem-se que:

$$|\psi\rangle = (\alpha_1|0\rangle + \beta_1|1\rangle) \otimes (\alpha_2|0\rangle + \beta_2|1\rangle) = \alpha_1\alpha_2|00\rangle + \alpha_1\beta_2|01\rangle + \beta_1\alpha_2|10\rangle + \beta_1\beta_2|11\rangle.$$

Naturalmente é possível generalizar essa representação para um sistema quântico com quantidade arbitrária n de qubits.

Seja $|\Psi_1\rangle$ e $|\Psi_2\rangle$ estados quânticos, é comum denotar o estado $|\Psi_1\rangle \otimes |\Psi_2\rangle$ como $|\Psi_1\rangle|\Psi_2\rangle$ ou ainda $|\Psi_1, \Psi_2\rangle$. Além disso, seja $|\psi\rangle$ um estado quântico de um qubit, é denotado por $|\psi\rangle^{\otimes k}$ o produto tensorial de $|\psi\rangle$ k vezes, isto é $|\psi\rangle \otimes |\psi\rangle \otimes ... \otimes |\psi\rangle$ onde $|\psi\rangle$ ocorre k vezes.

3.3 SISTEMAS QUÂNTICOS

Nesta seção será discutido as três propriedades importantes de um sistema quântico: sobreposição, medição e emaranhamento.

3.3.1 Sobreposição

Um sistema quântico $|\psi\rangle$ de n qubits pode estar em 2^n estados distintos possíveis, então pode ser descrito como uma combinação linear das bases do sistema quântico satisfazendo:

$$|\psi\rangle = \sum_{s=0}^{2^n-1} a_s|s\rangle, a_s \in \mathbb{C}, n \in \mathbb{Z}_{>0}, \sum_{s=0}^{2^n-1} |a_s|^2 = 1.$$

3.3.2 Medição

Seja $|\psi\rangle$ um sistema quântico no estado:

$$|\psi\rangle = \sum_{s=0}^{2^n-1} a_s|s\rangle, a_s \in \mathbb{C}, n \in \mathbb{Z}_{>0}, \sum_{s=0}^{2^n-1} |a_s|^2 = 1.$$

Ao medir o sistema o resultado será um estado associado a uma das 2^n bases do sistema quântico e o resultado k pode ser observado com probabilidade $|a_k|^2$. Além disso, o novo estado do sistema passa a ser o estado medido, isto é, supondo que o resultado da medida seja k, então o novo estado do sistema será $|k\rangle$

3.3.3 Emaranhamento

Em um sistema quântico com dois ou mais qubits, é possível obter situações onde o estado do sistema não pode ser escrito como o produto tensorial de dois ou mais sistemas quânticos, dessa forma os valores dos qubits que compõem um sistema assim podem estar correlacionados. Por exemplo no estado $|\psi\rangle$ descrito a seguir:

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle).$$

Note que os estados $|00\rangle$ e $|11\rangle$ podem ser obtidos com probabilidade de 50%. Entretanto, dado que o sistema está em sobreposição desses estados, medir um qubit revela informação sobre o valor do outro qubit, isto é, ao medir um qubit o outro qubit colapsa para um determinado valor, nesse caso colapsa para o mesmo valor lido.

3.4 PORTAS QUÂNTICAS

Portas quânticas são os componentes lógicos responsáveis por alterar o estado de um ou mais qubits e são a base para a construção de um algoritmo quântico. É comum se referir à portas quânticas como operadores quânticos ou simplesmente operadores. Nessa seção serão apresentadas algumas portas quânticas comuns e algumas técnicas úteis de associá-las.

3.4.1 Portas com um qubit

Portas quânticas são descritas por matrizes unitárias, isto é, matrizes cujo seu conjugado transposto é também a sua inversa. A seguir serão apresentadas algumas portas quânticas comuns que atuam sobre apenas um qubit:

- Porta identidade: a porta identidade é descrita pela matriz identidade e usualmente é denotada pela letra I. Essa porta não altera o estado quântico. A matriz que descreve essa porta é a seguinte:

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

- Porta Hadamard: é uma das portas mais importantes, normalmente é denotada pela letra H. A matriz que descreve essa porta é a seguinte:

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

- Porta NOT: essa porta troca a amplitude do estado $|0\rangle$ com a do estado $|1\rangle$. A matriz que a descreve é a seguinte:

$$NOT = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

- Porta de mudança de fase: Essa porta mapeia o estado $|0\rangle$ para $|0\rangle$ e o estado $|1\rangle$ para $e^{i\rho}|1\rangle$, com $\rho \in \mathbb{R}$. A matriz que descreve essa porta é a seguinte:

$$P(\rho) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\rho} \end{bmatrix}$$

3.4.2 Portas com múltiplos qubits

Portas quânticas que atuam em mais de um qubit são representadas por matrizes quadradas de 2^n por 2^n elementos onde n é a quantidade de qubits da porta. É importante citar que as matrizes que representam portas que atuam em mais de um qubit ainda são matrizes unitárias. A seguir serão apresentadas algumas das portas mais comuns que atua em múltiplos qubits.

- Porta CNOT: a porta CNOT ou porta não-controlado, é uma porta que usa um qubit como controle e atua invertendo ou não o outro qubit, com base no qubit de controle. Isto é, se o qubit de controle está no estado $|1\rangle$, é equivalente a ter aplicado uma porta NOT no segundo qubit porém caso o qubit de controle esteja no estado $|0\rangle$, é equivalente a ter aplicado a porta identidade. A matriz que descreve a porta CNOT é a seguinte:

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

  Note que o qubit de controle não é afetado após aplicada a porta CNOT.

  Em desenhos de circuitos quânticos, a porta CNOT é usualmente representada pela imagem da Figura 3.1

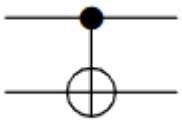


Figura 3.1: Representação da porta CNOT

- Porta U-controlada: A porta U-controlada, também denotada por CT-U, é uma generalização da porta CNOT onde o qubit de controle determina se será aplicada a porta U ao outro qubit ou não.

  Seja a porta U definida pela matriz:

$$U = \begin{bmatrix} u_{00} & u_{01} \\ u_{10} & u_{11} \end{bmatrix}$$

  Então a matriz da porta quântica CT-U é dada por:

$$\text{CT-}U = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{00} & u_{01} \\ 0 & 0 & u_{10} & u_{11} \end{bmatrix}$$

  Em desenhos de circuitos quânticos, a porta CT-U é usualmente representada pela imagem da Figura 3.2.

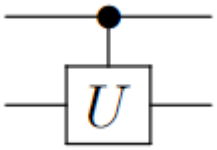


Figura 3.2: Representação da porta CT-U

Também podemos construir uma porta CT-G onde G é uma porta quântica qualquer que atua sobre uma quantidade arbitrária de qubits como mostrado (Kaye et al., 2006).

Neste trabalho será usada a notação U^k para representar a porta U aplicada consecutivamente k vezes.

3.4.3 Autovetores e Autovalores

A seguir será apresentada a definição de autovetores e autovalores que são conceitos que serão utilizados nas próximas seções.

Definição 3.4.1 *Seja U um operador que atua sobre n qubits e $|\psi\rangle$ um estado quântico de n qubits, se $U|\psi\rangle = e^{\alpha i}|\psi\rangle, \alpha \in \mathbb{R}$, então $|\psi\rangle$ é dito um autovetor de U com autovalor $e^{\alpha i}$.*

Teorema 3.4.2 *Seja U um operador, $|\psi\rangle$ um autovetor de U cujo autovalor é $e^{\alpha i}$ e $\alpha \in \mathbb{R}$, então $|\psi\rangle$ é autovetor de $U^k, k \in \mathbb{Z}_{>0}$ com autovalor igual a $(e^{\alpha i})^k$.*

3.4.4 Circuitos quânticos

Um circuito quântico é uma descrição temporal das modificações a serem feitas nos qubits. De maneira semelhante aos circuitos clássicos, são compostos por inicializações de qubits, portas quânticas associadas e medidas de qubits.

É usual chamar os qubits de entrada de um circuito quântico de registrador, quando conveniente também é possível agrupar os qubits em mais de um registrador. Em diagramas de circuitos quânticos, os qubits são levados da saída de uma porta para a entrada de outra porta por fios, representados por linhas horizontais ou verticais. Normalmente, a leitura desses diagramas se dá da esquerda para a direita, ou seja, após a inicialização dos qubits, eles são propagados ao longo do tempo da esquerda para a direita através de portas quânticas e fios.

3.4.5 Phase kickback

Nesta seção será apresentada a descrição de um fenômeno chamado de phase kickback. Este fenômeno é utilizado em diversos algoritmo quânticos que serão apresentados nos próximos capítulos. Esse efeito é quando o autovalor de um operador é passado para o qubit de controle de uma porta controlada (Cleve et al., 1998).

Na argumentação que virá abaixo, será usado a seguinte igualdade que é válida para produtos tensoriais:

$$c(|\psi_1\rangle|\psi_2\rangle) = (c|\psi_1\rangle)|\psi_2\rangle = |\psi_1\rangle(c|\psi_2\rangle), c \in \mathbb{C}.$$

Seja U um operador, $|\psi\rangle$ um autovetor de U cujo autovalor é $e^{\phi i}$ com $\phi \in [0, 2\pi)$. Da definição de autovalor, observe que o seguinte é válido

$$U|\psi\rangle = e^{\phi i}|\psi\rangle \tag{3.1}$$

Dado o operador controlado por um qubit CT-U, definido na seção anterior, temos da definição de CT-U e da equação 3.1

$$\begin{aligned} \text{CT-}U|1\rangle|\psi\rangle &= |1\rangle U|\psi\rangle \\ &= |1\rangle e^{\phi i}|\psi\rangle \\ &= e^{\phi i}|1\rangle|\psi\rangle. \end{aligned}$$

É notável que para um qubit dado por

$$|x\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$$

quando usado como qubit de controle do operador CT-U e o estado $|\psi\rangle$ como registrador alvo resulta no seguinte estado quântico

$$\text{CT-}U|x\rangle|\psi\rangle = \frac{|0\rangle + e^{\alpha i}|1\rangle}{\sqrt{2}}|\psi\rangle. \tag{3.2}$$

3.4.6 Operador controlado por múltiplos qubits

Nesta seção será generalizado o conceito de operador controlado, de forma que teremos um registrador de controle $|x\rangle$, ao invés de termos apenas um qubit controlando a operação. A ideia é que o operador CTR-U^x faça o seguinte mapeamento

$$|x\rangle|s\rangle \mapsto |x\rangle U^x|s\rangle, x \in \mathbb{Z}_{\geq 0} \tag{3.3}$$

onde U^x é o operador equivalente a aplicar x vezes o operador U consecutivamente. Perceba que da equação 3.3 a quantidade de vezes que o operador U é aplicado no registrador $|s\rangle$ é determinado pelo valor do registrador $|x\rangle$, chamaremos o registrador $|x\rangle$ de registrador de controle e o registrador $|s\rangle$ de registrador alvo. Suponha que o registrador de controle e o registrador alvo possuam n e m qubits, respectivamente.

Seja U um operador qualquer, considere o operador CT-U^{2^k} com $k \in [0..n]$, ou seja, o operador U^{2^k} controlado por apenas um qubit. Para construir o operador CTR-U^x podemos usar n operadores da forma CT-U^{2^i} usando o i-ésimo qubit do registrador de controle para controlar o operador CT-U^{2^i} e ligar a saída desse operador à entrada do operador CT-$U^{2^{(i+1)}}$, criando o cicuito mostrado na Figura 3.3.

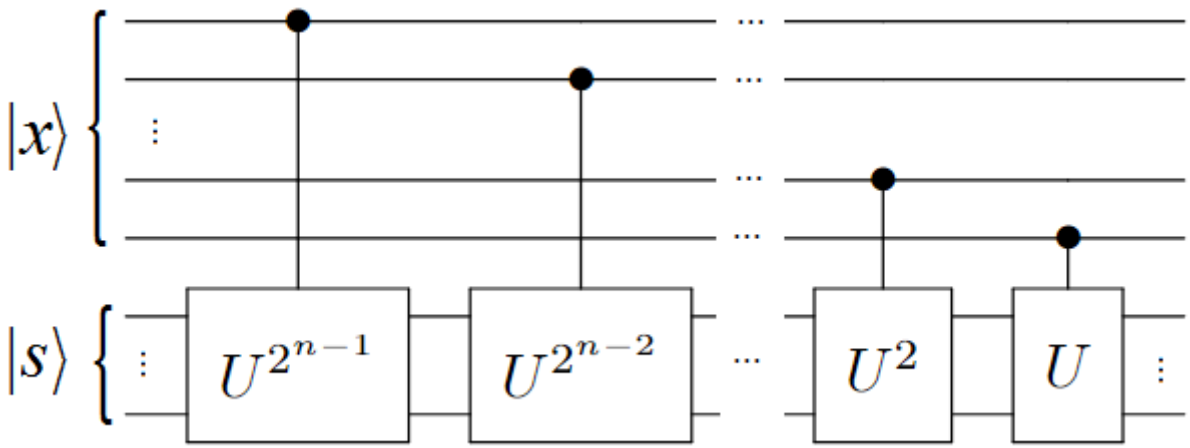


Figura 3.3: Circuito quântico do operador CTR-U^x

Perceba que a ideia por trás desse circuito é utilizar a representação binária do valor no registrador $|x\rangle$ para aplicar x vezes o operador U e com isso fazer o mapeamento especificado na equação 3.3.

Seja o estado quântico de n qubits $|\alpha\rangle$ o registrador de controle do operador CTR-U^x tal que

$$|\alpha\rangle = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)^{\otimes n}. \tag{3.4}$$

E seja $|\psi\rangle$ um autovetor do operador U com autovalor $e^{2\pi i\omega}$ onde $\omega \in \mathbb{R}$ e $\omega \in [0, 1)$. Do Teorema 3.4.2 temos que $|\psi\rangle$ é autovetor de U^{2^k} para qualquer $k \in [0..n-1]$ com autovetor $e^{2\pi i(2^k\omega)}$. Note que o i-ésimo qubit do registrador de controle $|\alpha\rangle$ está no estado

$$\frac{|0\rangle + |1\rangle}{\sqrt{2}}.$$

Como apresentado anteriormente, dentro do operador CTR-U^x é aplicado o operador CT-U^{2^i} controlado pelo i-ésimo qubit de $|\alpha\rangle$, da equação 3.2 teremos que o qubit de controle estará no seguinte estado

$$\frac{|0\rangle + e^{2\pi i(2^i\omega)}|1\rangle}{\sqrt{2}}.$$

E portanto após a aplicação do operador CTR-U^x o registrador $|\alpha\rangle$ estará no estado

$$\begin{aligned}|\alpha\rangle &= \left(\frac{|0\rangle + e^{2\pi i(2^{n-1}\omega)}|1\rangle}{\sqrt{2}}\right) \otimes \left(\frac{|0\rangle + e^{2\pi i(2^{n-2}\omega)}|1\rangle}{\sqrt{2}}\right) \otimes \ldots \otimes \left(\frac{|0\rangle + e^{2\pi i(\omega)}|1\rangle}{\sqrt{2}}\right) \\ &= \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} e^{2\pi i\omega x}|x\rangle.\end{aligned}$$

Perceba que o autovalor de U foi parar na amplitude do registrador de controle após a aplicação da porta CTR-U^x, esse fenômeno será usado na construção dos algoritmos que serão apresentados.

Por vezes, circuitos quânticos podem ficar muito grandes e complexos, nesses casos é comum agrupar partes do circuito representado a porção modularizada como se fosse uma porta quântica, a Figura 3.4 é a representação simplificada do circuito mostrado na Figura 3.3

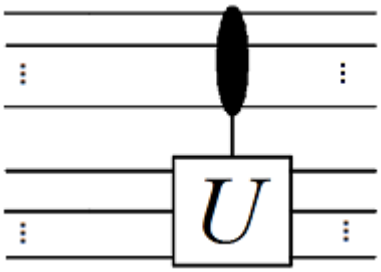


Figura 3.4: Representação do operador CTR-U^x

3.4.7 Transformada quântica de Fourier

A transformada quântica de Fourier, proposta por Don Coppersmith (Coppersmith, 1994), é a implementação quântica da transformada discreta de Fourier. Em suma, a transformada

discreta de Fourier é uma ferramenta para decompor uma função em termos das suas frequências, a sua versão quântica segue uma ideia parecida.

É importante destacar que a transformada quântica de Fourier e a transformada discreta de Fourier são semelhantes porém não são iguais. A transformada discreta de Fourier retorna um conjunto de valores enquanto a transformada quântica de Fourier faz uma transformação dos estados dos qubits de entrada de forma que a medição desses qubits transformados retorna uma informação referente ao estado de entrada.

A transformada quântica de Fourier de n qubits é uma transformação unitária que transforma um estado quântico de n qubits $|x\rangle$ em outro estado quântico de n qubits $|y\rangle$ fazendo o seguinte mapeamento

$$|x\rangle \mapsto \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i \frac{x}{2^n} y} |y\rangle.$$

A transformada quântica de Fourier de n qubits é denotada por QFT_{2^n} porém é usual denotar apenas por QFT quando o contexto é claro.

Seja $N = 2^n$ e $\omega = e^{\frac{2\pi i}{2^n}}$, a QFT_{2^n} é definida pela matriz:

$$QFT_{2^n} = \begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \dots & \omega^N \\ 1 & \omega^2 & \omega^4 & \omega^6 & \dots & \omega^{N-1} \\ 1 & \omega^3 & \omega^6 & \omega^9 & \dots & \omega^{2N-2} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2N-1} & \omega^{3N-3} & \dots & \omega^{(N-1)(N-1)} \end{bmatrix}$$

Como citado anteriormente, a transformada de Fourier de n qubits é uma transformação unitária, sendo assim existe uma operação inversa que faz o seguinte mapeamento

$$\frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i \frac{x}{2^n} y} |y\rangle \mapsto |x\rangle.$$

A partir disso, define-se a transformada inversa de Fourier de n qubits que faz o seguinte mapeamento

$$|y\rangle \mapsto \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{-2\pi i \frac{x}{2^n} y} |x\rangle. \tag{3.5}$$

O circuito quântico que implementa a transformada inversa de Fourier de n qubits é usualmente denotado por $\mathrm{QFT}_{2^n}^{-1}$. O circuito que implementa QFT^{-1} é basicamente o mesmo circuito que implementa a QFT trocando todas as portas por suas respectivas inversas.

Outro aspecto notável da transformada quântica de Fourier e da transformada inversa de Fourier é que as suas implementações com operadores quânticos possuem um custo computacional exponencialmente menor que a implementação da transformada discreta de Fourier. Em (Nielsen e Chuang, 2002) é mostrado como construir uma QFT_{2^n} usando $O(n^2)$ portas quânticas, além disso argumenta que as melhores implementação clássicas da transformada de Fourier discreta possuem ordem assintótica $O(n2^n)$.

4 ALGORITMOS QUÂNTICOS

Um algoritmo quântico é especificado por um circuito quântico, como visto na seção 3.4.4, e são usados para resolver problemas computacionais. Os algoritmos que serão apresentados possuem uma parte de sua execução que não depende de efeitos quânticos, isto é, partes que um computador clássico seria capaz de computar com o mesmo custo computacional.

Nesta seção serão apresentados alguns problemas computacionais que serão usados como passo intermediário para argumentações que virão nos próximos capítulos e algoritmos quânticos capazes de resolver esses problemas de maneira mais eficiente em relação aos algoritmos clássicos conhecidos. O conteúdo desse capítulo foi baseado em (Kaye et al., 2006) e (VAZIRANI, 2013).

4.1 O PROBLEMA DA ESTIMAÇÃO DE FASE

Seja $\omega \in \mathbb{R}$ onde $\omega \in [0, 1)$. Considere a representação de números em binário com a notação de ponto fixo, temos que se for usada uma quantidade infinita de bits, ω é representado por

$$\omega = 0, x_1x_2x_3... \text{ com } x_i \in \{0, 1\}, i \in \mathbb{Z}_{>0}$$

ou seja

$$\omega = \sum_{i=1}^{\infty} x_i 2^{-i}.$$

De maneira semelhante a computação clássica, no contexto da computação quântica é fixado um número n de qubits para representar números. É trivial verificar que um registrador quântico de n qubits pode representar 2^n valores distintos. Usando a notação de ponto fixo, esse registrador consegue representar todos os números decimais da forma $\frac{k}{2^n}$ com $k \in [0..2^n - 1]$, isto é, múltiplos inteiros de $\frac{1}{2^n}$.

Pela natureza discreta e finita da representação dos números em computadores, sejam eles clássicos ou quânticos, não é possível representar todos os valores válidos de ω de maneira precisa. Dessa forma, usando n qubits a melhor representação de um número ω será aquela que gere menor erro, isto é, a melhor representação de ω em n qubits é $\widetilde{\omega} = \frac{k}{2^n}$ com $k \in [0..2^n - 1]$ tal que o erro $|\omega - \widetilde{\omega}|$ obtido é o mínimo possível. Perceba que para um número decimal representado em n qubits a menor variação de um número representável para o próximo número representável é de $\frac{1}{2^n}$ pois para qualquer $k \in [0..2^n - 2]$ temos que $\frac{k+1}{2^n} - \frac{k}{2^n} = \frac{1}{2^n}$. Com isso, concluímos que a melhor representação de ω em n qubits será $\widetilde{\omega}$ onde $\widetilde{\omega} = \frac{k}{2^n}$ com $k \in [0..2^n - 1]$ tais que $|\omega - \widetilde{\omega}| \leq \frac{1}{2^{n+1}}$. Note que ω pode possuir duas melhores representações $\widetilde{\omega}_1 = \frac{k}{2^n}$ e $\widetilde{\omega}_2 = \frac{k+1}{2^n}$ se $|\omega - \widetilde{\omega}_1| = |\omega - \widetilde{\omega}_2|$.

Seja $n \in \mathbb{Z}_{>0}$ e $\omega \in \mathbb{R}$ tal que $\omega \in [0, 1)$, o problema de estimação de fase (PEF) é definido da seguinte forma

Problema de estimação de fase (PEF)

Entrada: Um estado quântico de n qubits da forma

$$\frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i \omega y} |y\rangle.$$

Saída: Obter uma estimativa para ω representável em n qubits.

4.2 ALGORITMO PARA ESTIMAÇÃO DE FASE

Seja $\omega \in \mathbb{R}$ onde $\omega \in [0, 1)$ e n a quantidade de qubits que o estado de entrada do PEF, considere $x \in \mathbb{Z}_{\geq 0}$ e $\delta \in \mathbb{R}$ tais que $\omega = \frac{x}{2^n} + \delta$ onde $|\delta| \leq \frac{1}{2^{n+1}}$. Não é difícil perceber que $\frac{x}{2^n}$ é a melhor representação de ω em n qubits. Além disso, considere o estado $|x\rangle$ de n qubits, da definição do mapeamento da QFT_{2^n} temos que

$$QFT_{2^n}|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i \frac{x}{2^n} y} |y\rangle.$$

A ideia do algoritmo para resolver o PEF consiste em perceber que o estado quântico de entrada é muito semelhante à saída de uma QFT_{2^n} aplicada a um registrador contendo o inteiro x onde $\frac{x}{2^n}$ é a melhor representação de ω em n qubits. Com isso, ao aplicar a operação inversa, isto é, a $\text{QFT}_{2^n}^{-1}$ no estado de entrada recuperamos o valor de x e com esse resultado calculamos com uma divisão por 2^n uma boa estimativa para ω.

Algoritmo 1 Estimação de fase($|x\rangle$):

1: $\psi \leftarrow QFT_{2^n}^{-1}|x\rangle$
2: Medir ψ obtendo $\widetilde{x}$
3: $\widetilde{\omega} \leftarrow \frac{\widetilde{x}}{2^n}$
4: **return** $\widetilde{\omega}$

Aplicando a QFT^{-1} no estado quântico $|x\rangle$ obtemos

$$\begin{aligned} QFT^{-1}|x\rangle &= QFT^{-1} \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i \omega y} |y\rangle \\ &= \frac{1}{2^n} \sum_{y=0}^{2^n-1} \sum_{k=0}^{2^n-1} e^{2\pi i \omega y} e^{-2\pi i \frac{k}{2^n} y} |k\rangle \\ &= \frac{1}{2^n} \sum_{y=0}^{2^n-1} \sum_{k=0}^{2^n-1} e^{2\pi i y(\omega - \frac{k}{2^n})} |k\rangle \end{aligned}$$

Fazendo $\omega = \frac{\widetilde{x}}{2^n} + \delta$ tais que $\widetilde{x} \in \mathbb{Z}_{\geq 0}$ e $|\delta| \leq \frac{1}{2^{n+1}}$, perceba que $\frac{\widetilde{x}}{2^n}$ é uma boa representação de ω. Com isso, temos que a probabilidade de medir $\widetilde{x}$ na saída da $\mathrm{QFT}_{2^n}^{-1}$ é de

$$\begin{aligned} Prob(\widetilde{x}) &= \left| \frac{1}{2^n} \sum_{y=0}^{2^n-1} \sum_{x=0}^{2^n-1} e^{2\pi i y(\omega - \frac{\widetilde{x}}{2^n})} \right|^2 \\ &= \left| \frac{1}{2^n} \sum_{y=0}^{2^n-1} \sum_{x=0}^{2^n-1} e^{-\frac{2\pi i y}{2^n}(\widetilde{x} - 2^n\omega)} \right|^2 \\ &= \left| \frac{1}{2^n} \sum_{y=0}^{2^n-1} \sum_{x=0}^{2^n-1} e^{-\frac{2\pi i y}{2^n}(\widetilde{x} - \widetilde{x})} e^{2\pi i \delta y} \right|^2 \\ &= \frac{1}{2^{2n}} \left| \sum_{y=0}^{2^n-1} e^{2\pi i \delta y} \right|^2 \end{aligned} \tag{4.1}$$

Note que da equação 4.1, se $\delta = 0$ temos que a probabilidade de obter $|\widetilde{x}\rangle$ é 1. Nesse caso $\omega = \frac{\widetilde{x}}{2^n}$, isto é, ω pode ser representado com exatidão em n qubits e daí o Algoritmo 1 retorna sempre o valor exato de ω. Caso $\delta \neq 0$, aplicando a fórmula da soma geométrica em 4.1 e utilizando a identidade matemática $|1 - e^{2ix}|^2 = 4|\sin(x)|^2$, obtemos

$$\begin{aligned} Prob(\widetilde{x}) &= \frac{1}{2^{2n}} \left| \frac{1 - e^{2\pi i 2^n \delta}}{1 - e^{2\pi i \delta}} \right|^2 \\ &= \frac{1}{2^{2n}} \left| \frac{2\sin(\pi 2^n \delta)}{2\sin(\pi\delta)} \right|^2 \\ &= \frac{1}{2^{2n}} \frac{|2\sin(\pi 2^n \delta)|^2}{|2\sin(\pi\delta)|^2} \end{aligned}$$

dado que $|\sin(\pi\delta)| \leq |\pi\delta|$ e que $|2^{n+1}\delta| \leq |\sin(\pi 2^n \delta)|$ para qualquer $|\delta| \leq \frac{1}{2^n}$, chegamos em

$$Prob(\widetilde{x}) = \frac{1}{2^{2n}} \frac{|2\sin(\pi 2^n \delta)|^2}{|2\sin(\pi\delta)|^2} \geq \frac{1}{2^{2n}} \frac{|2^{n+1}\delta|^2}{|\pi\delta|^2} = \frac{4}{\pi^2}. \tag{4.2}$$

Como nesse caso $\frac{\widetilde{x}}{2^n}$ é a melhor representação de ω em n qubits, concluímos que o Algoritmo 1 retorna a melhor representação de ω em n qubits com probabilidade de pelo menos $\frac{4}{\pi^2}$.

4.3 O PROBLEMA DA ESTIMAÇÃO DO AUTOVALOR

O problema de estimação de autovalor (PEA) é semelhante ao PEF, enunciado na seção 4.1, porém possui um contexto diferente. Dado $n \in \mathbb{Z}_{>0}$, um operador unitário de n qubits U e o estado quântico $|\psi\rangle$ que é um autovetor de U cujo autovalor é $e^{2\pi i\omega}$ para $\omega \in \mathbb{R}$ tal que $\omega \in [0, 1)$, o PEA é definido da seguinte forma

Problema de estimação do autovalor (PEA)

Entrada: Um circuito quântico que implementa o operador U, um estado quântico de n qubits com o autovetor $|\psi\rangle$ cujo autovalor é $e^{2\pi i\omega}$.
Saída: Obter uma boa estimativa para ω representável em n qubits.

Novamente, entende-se por estimativa de ω a melhor representação de ω em n qubits.

4.4 ALGORITMO PARA ESTIMAÇÃO DO AUTOVALOR

A ideia central do algoritmo da estimação de autovalor é utilizar o efeito phase kickback para que o autovalor de U passe para o registrador de controle e então usar o algoritmo de estimação de fase para obter ω.

O Algoritmo 2 usa dois registradores de n qubits, o registrador de controle e o registrador alvo. Também é usado a porta CTR-U^x, que foi mostrado como pode ser construída na seção 3.4.6.

Algoritmo 2 Estimação de autovalor(U,$|\psi\rangle$):

1: Inicializar o registrador de controle com o estado $|0\rangle^{\otimes n}$.
2: Aplicar a QFT_{2^n} no registrador de controle.
3: Aplicar CTR-U^x no autovetor $|\psi\rangle$ controlado pelo registrador de controle.
4: Aplicar $QFT_{2^n}^{-1}$ no registrador de controle.
5: Medir o registrador de controle obtendo $\widetilde{x}$.
6: $\widetilde{\omega} \leftarrow \frac{\widetilde{x}}{2^n}$.
7: **return** $\widetilde{\omega}$.

Após aplicar a QFT_{2^n} no registrador de controle, o estado quântico do sistema é

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle|\psi\rangle.$$

Do Teorema 3.4.2, $|\psi\rangle$ é autovetor do operador U^x. Então em seguida é aplicado ao operador CTR-U^x o autovetor de U^x ocorrendo o efeito de phase kickback, descrito na seção 3.4.5, obtendo o estado

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} e^{2\pi i\omega x}|x\rangle|\psi\rangle,$$

note que o registrador de controle possui exatamente a entrada do problema de estimação de fase.

Por último é aplicado a $\text{QFT}_{2^n}^{-1}$ e com a mesma análise da seção 4.2, concluímos que o algoritmo de estimação de autovalor retorna a melhor representação de ω em n qubits com probabilidade de pelo menos $\frac{4}{\pi^2}$. O circuito que computa o Algoritmo 2 está representado na Figura 4.1.

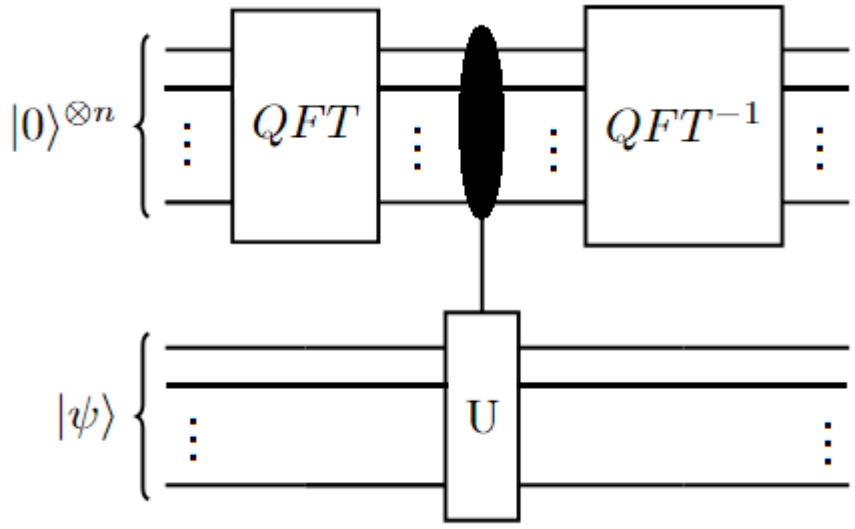


Figura 4.1: Circuito para estimação de autovalor

4.5 O PROBLEMA DA ESTIMAÇÃO DA ORDEM

Seja G um grupo qualquer, definimos o problema de estimação da ordem da seguinte maneira

Problema de estimação da ordem (PEO)

Entrada: Um elemento $g \in G$.
Saída: Um inteiro r representando a ordem de g.

4.6 ALGORITMO PARA ESTIMAÇÃO DA ORDEM

A título de simplicidade e para ilustrar a constução do algoritmo com um grupo concreto usaremos o grupo $\mathbb{Z}_p^+$, sendo p um número primo[1], porém o algoritmo apresentado pode ser adaptado para encontrar a ordem de um elemento em qualquer grupo (Kaye et al., 2006).

Considere a porta U_g que faz o seguinte mapeamento

$$U_g : |s\rangle \mapsto |s+g\rangle,\ g, s \in \mathbb{Z}_p^+ \tag{4.3}$$

Podemos usar múltiplos da U_g para criar o operador CTR-U_g^x de forma similar à descrita na seção 3.4.6, onde o operador CTR-U_g^x faz o seguinte mapeamento

$$\text{CTR-}U_g^x : |x\rangle|s\rangle \mapsto |x\rangle U_g^x|s\rangle = |s\rangle \tag{4.4}$$

Perceba que do Teorema 3.4.2, temos que qualquer autovetor de U_g também é autovetor de U_g^x. Com isso, considere o estado quântico de n qubits

$$|u_k\rangle = \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{-2\pi i \frac{k}{r} s} |sg\rangle \tag{4.5}$$

[1]O grupo $\mathbb{Z}_p^+$ é composto pelos elementos do conjunto $\mathbb{Z}_p$, isto é, pelo conjunto $\{0, 1, ..., p-1\}$ e possui como operação a adição módulo p.

Temos que

$$\begin{aligned}
U_g|u_k\rangle &= \frac{1}{\sqrt{r}}\sum_{s=0}^{r-1} e^{-2\pi i\frac{k}{r}s}U_g|sg\rangle \\
&= \frac{1}{\sqrt{r}}\sum_{s=0}^{r-1} e^{-2\pi i\frac{k}{r}s}|(s+1)g\rangle \\
&= e^{2\pi i\frac{k}{r}}\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1} e^{-2\pi i\frac{k}{r}(s+1)}|(s+1)g\rangle \\
&= e^{2\pi i\frac{k}{r}}\frac{1}{\sqrt{r}}\left(\sum_{s=0}^{r-2} e^{-2\pi i\frac{k}{r}(s+1)}|(s+1)g\rangle + e^{-2\pi i\frac{k}{r}r}|rg\rangle\right) \\
&= e^{2\pi i\frac{k}{r}}\frac{1}{\sqrt{r}}\left(\sum_{s=1}^{r-1} e^{-2\pi i\frac{k}{r}s}|sg\rangle + e^{-2\pi i\frac{k}{r}0}|0g\rangle\right) \\
&= e^{2\pi i\frac{k}{r}}\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1} e^{-2\pi i\frac{k}{r}s}|sg\rangle \\
&= e^{2\pi i\frac{k}{r}}|u_k\rangle
\end{aligned}$$

Portanto, $|u_k\rangle$ é autovetor de U_g cujo autovalor é $e^{2\pi i\frac{k}{r}}$. Perceba que poderíamos usar o algoritmo de estimação de autovalor tendo como entrada o estado $|u_k\rangle$ para qualquer $k \in [0..r-1]$ e o operador U_g para conseguir

$$|0\rangle|u_k\rangle \mapsto |\widetilde{k/r}\rangle|u_k\rangle$$

Porém como não é conhecido o valor de r, não é possível criar o estado $|u_k\rangle$.

Considere o estado quântico

$$\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}|u_k\rangle = \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1} e^{-2\pi i\frac{k}{r}s}|sg\rangle \tag{4.6}$$

Perceba que $|sg\rangle = |0\rangle$ se e somente se $s \equiv 0\ (mod\ r)$. A amplitude de $|0\rangle$ no estado 4.6 é a soma dos termos em que $s = 0$. Ou seja

$$\frac{1}{\sqrt{r}}\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1} e^{-2\pi i\frac{k}{r}0} = \frac{1}{r}\sum_{k=0}^{r-1} 1 = 1$$

Como a equação em 4.6 é um estado quântico, então pela definição de estados quânticos a soma do quadrado das amplitudes deve ser 1. Como a amplitude do estado $|0\rangle$ é 1, a amplitude dos demais estados deve ser 0. Concluimos que

$$\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}|u_k\rangle = |0\rangle.$$

Com isso, temos que

$$|0\rangle|0\rangle = |0\rangle\left(\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}|u_k\rangle\right) = \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}|0\rangle|u_k\rangle. \tag{4.7}$$

Antes de apresentar o algoritmo de estimação de ordem é necessário apresentar o funcionamento do algoritmo de frações contínuas que pode ser encontrado com mais detalhes em (Hardy et al., 1979).

Algoritmo 3 Frações contínuas(w, n):

1: $k', r' \leftarrow$ valores que satisfazem $\left|w - \frac{k'}{r'}\right| \leq \frac{1}{2^n}$ e que $MDC(k', r') = 1$.
2: **return** k', r'.

O Algoritmo 4 recebe um elemento $g \in \mathbb{Z}_p^+$ e a ordem p do grupo $\mathbb{Z}_p^+$.

Algoritmo 4 Estimação da ordem(g, p):

1: $n \leftarrow \lceil 2\log_2(p)\rceil$.
2: Inicializar o registrador de controle com $|0\rangle^{\otimes n}$.
3: Inicializar o registrador alvo com $|0\rangle^{\otimes n}$.
4: Aplicar a QFT_{2^n} no registrador de controle.
5: Aplicar CTR-U_g^x no registrador de controle e no registrador alvo.
6: Aplicar a $\mathrm{QFT}_{2^n}^{-1}$ no registrador de controle
7: Medir o registrador de controle obtendo x_1 para algum $k \in [0..r-1]\, \frac{x_1}{2^n} = \widetilde{\frac{k}{r}}$.
8: $c_1, r_1 \leftarrow$Frações contínuas$\left(\frac{x_1}{2^n}, n\right)$
9: Repetir os passos 2-7 até obter $x_2 \neq x_1$.
10: $c_2, r_2 \leftarrow$Frações contínuas$\left(\frac{x_2}{2^n}, n\right)$
11: $r \leftarrow MMC(r_1, r_2)$
12: **return** r

A linha 1 do Algoritmo 4 garante que é escolhido n grande o suficiente para que o algoritmo de frações contínuas gere valores suficientemente precisos.

Como discutido anteriormente o estado $|0\rangle^{\otimes n}|0\rangle^{\otimes n}$ pode ser reescrito como

$$|0\rangle^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}|0\rangle^{\otimes n}|u_k\rangle$$

E como $|u_k\rangle$ é autovetor do operador U_g^x, a sequência das linhas 4,5,6 e 7 funcionam igual ao algoritmo de estimação de fase(algoritmo 4.2). Nesse estágio o estado do sistema é

$$\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}|\widetilde{k/r}\rangle|u_k\rangle, \text{ para algum } k \in [0..r-1]$$

Considere o seguinte lema demonstrado em (Cheung, 2003).

Lema 4.6.1 *Dado um operador U e um autovetor $|\psi\rangle$ de U com autovalor $e^{2\pi i\omega}$, onde $\omega \in \mathbb{R}$ e $\omega \in [0, 1)$. Então com probabilidade $\frac{8}{\pi^2}$ o algoritmo de estimação de fase retornará $\widetilde{x} \in [0..2^n-1]$ tal que*

$$\left|\frac{\widetilde{x}}{2^n} - \omega\right| \leq \frac{1}{2^n}$$

Utilizando o Lema 4.6.1, obtemos com probabilidade $\frac{8}{\pi^2}$ $x_1 \in [0..2^n - 1]$ satisfazendo

$$\left|\frac{x_1}{2^n} - \frac{k}{r}\right| \leq \frac{1}{2^n}$$

Com isso o algoritmo das frações contínuas retorna o único valor possível de k_1 e r_1 tais que $\frac{k_1}{r_1} = \frac{k}{r}$, onde k_1 e r_1 não possuem fatores em comum.

Com análise semelhante aos passos anteriores, a execução da linha 10 retorna k_2 e r_2 únicos e que não possuem fatores em comum. Finalizamos, então, o Algoritmo 4 que com probabilidade $\frac{6}{\pi^2}$ calcula $MMC(r_1, r_2) = r$. Fazendo uma análise do Algoritmo 4, temos que a probabilidade de estimar corretamente a ordem do elemento g é de $\frac{384}{\pi^6}$ (Kaye et al., 2006).

5 CRIPTOGRAFIA

Neste capítulo serão introduzidos os conceitos de criptografia simétrica e assimétrica, além de uma introdução a curvas elípticas sobre corpos finitos e seu uso na geração de chaves para protocolos criptográficos com criptografia assimétrica.

5.1 MODELOS CRIPTOGRÁFICOS

Modelos criptográficos são usados para alterar uma mensagem de modo que seu conteúdo original se torna inacessível, porém de forma que, com as operações apropriadas, seja possível recuperar a mensagem original. Esse tipo de técnica é comumente usado para permitir a transmissão de uma mensagem por um canal não seguro, isto é, um canal onde não é garantido que apenas a origem e o destino da mensagem são capazes de acessar o seu conteúdo. Para isso, é necessário que a origem seja capaz de cifrar a mensagem e apenas o destino seja capaz de decifrá-la, para garantir essas propriedades são usados protocolos criptográficos que, usualmente, utilizam chaves para cifrar ou decifrar a mensagem.

O conteúdo desta seção também pode ser encontrado no livro “Cryptography and network security” (Forouzan e Mukhopadhyay, 2015), o qual compreende, também, outros aspectos importantes para a segurança da informação.

5.1.1 Criptografia simétrica

A criptografia simétrica usa a mesma chave para criptografar e descriptografar uma mensagem. Esse modelo contém cinco elementos essenciais:

- Uma mensagem m, que deve chegar ao destino sem que seu conteúdo seja exposto.
- Uma mensagem cifrada m', que é uma mensagem que não revela informações sobre a mensagem original m.
- Uma chave secreta k, que deve ser conhecida apenas pela origem e pelo destino da mensagem
- Uma função de cifragem f, que com base na chave secreta k, modifica a mensagem m produzindo a mensagem m'.
- Uma função de decifragem f^{-1}, que a partir de m', recupera a mensagem original m se usada a chave secreta k.

Em protocolos criptográficos que utilizam o modelo de chaves simétricas, inicialmente a origem e destino concordam com um valor da chave secreta k, em seguida a origem faz a cifragem da mensagem usando f.

$$f(k, m) = m'$$

Então a origem envia a mensagem m' para o destino, note que caso a mensagem m' seja interceptada será necessário descobrir a chave secreta k para recuperar a mensagem m.

A seguir, o destino decifra a mensagem m' usando a chave k e f^{-1}.

$$f^{-'}(k, m') = m$$

E assim, a mensagem m chega ao destino sem que seu conteúdo seja exposto.

5.1.2 Criptografia assimétrica

As criptografias assimétricas, também chamadas de criptografias de chave pública, usam um par de chaves ligadas matematicamente: a chave pública, usada para encriptar e a chave privada, usada para decriptar. Além da funcionalidade de gerar chaves para encriptar mensagens, esse modelo também pode ser usado para gerar assinaturas. Modelos de criptografias assimétricas em geral são compostos por oito elementos:

- Uma mensagem m, que deve chegar ao destino sem que seu conteúdo seja exposto.
- Uma mensagem cifrada m', que é uma mensagem que não revela informações sobre a mensagem original m.
- O par de chaves pri_O e pub_O, que são as chaves privada e pública da origem, respectivamente
- O par de chaves pri_D e pub_D, que são as chaves privada e pública do destino, respectivamente
- Uma função de cifragem f, que com base numa chave pública passada, cifra a mensagem m produzindo m'.
- Uma função de decifragem d, que com base numa chave privada passada, decifra a mensagem m' produzindo m.
- Uma função autenticadora g, que recebe uma mensagem m e uma chave privada gerando uma assinatura.
- Uma assinatura s, que é a mensagem assinada.

No processo de geração das chaves pública e privada é usado as chamadas funções de mão única, que são funções computacionalmente fáceis de computar porém computacionalmente difíceis de calcular a sua inversa. Isso faz com que dado uma chave privada seja possível computar uma chave pública associada, porém, a partir de uma chave pública, descobrir a chave privada é uma tarefa tipicamente intratável devido a dificuldade de computar a função inversa.

Em protocolos criptográficos de chave assimétrica, inicialmente a origem e o destino geram seus pares de chave pública e privada. Cada um divulga a sua chave pública.

Para a origem mandar uma mensagem m para o destino, primeiro é usado a chave pública do destino pub_D junto da função f para produzir uma mensagem cifrada.

$$f(pub_D, m) = m'$$

Então a origem envia a mensagem m' para o destino. É interessante notar que algum usuário que intercepte a mensagem precisaria da chave privada do destino, que não foi divulgada, é perceptível também que o interceptador tem conhecimento das chaves públicas da origem e do destino, pub_O e pub_D respectivamente, porém, como citado anteriormente, descobrir a chave privada a partir de uma chave pública é computacionalmente difícil.

Em seguida o destino usa sua chave privada, pri_D, junto da função d para recuperar a mensagem original.

$$d(pri_D, m') = m$$

E assim, a mensagem m chega ao destino sem que seu conteúdo seja exposto.

Modelos de chave assimétrica também podem ser usados para garantir a autenticidade de uma informação por meio de assinaturas, isto é, garantir que uma informação foi realmente enviada por um determinado usuário.

Supondo que a origem deseja autenticar uma mensagem m, então usando a função autenticadora g e a chave privada da origem pri_O é computado a assinatura da mensagem.

$$g(pri_O, m) = s$$

Utilizando a chave pública pub_O o destino pode verificar que a assinatura s pertence à origem.

5.2 CURVAS ELÍPTICAS SOBRE UM CORPO FINITO

Esta seção e as próximas serão dedicadas a apresentar o funcionamento de curvas elípticas sobre um corpo da forma $\mathbb{F}_p$, sendo p um número primo, bem como as operações entre pontos da curva e como usá-las para gerar chaves em protocolos criptográficos assimétricos.

Definição 5.2.1 *Seja $\mathbb{F}_p$ um corpo finito com p elementos de característica diferente de 2 e 3, onde p é um número primo. Uma curva elíptica sobre $\mathbb{F}_p$, denotada por $E(\mathbb{F}_p)$, é o conjunto de pontos $(x, y) \in \mathbb{F}_p \times \mathbb{F}_p$ que satisfazem a equação modular*

$$y^2 \equiv x^3 + ax + b \ (mod\ p) \tag{5.1}$$

junto de um ponto especial chamado de ponto no infinito, denotado por O, onde $a, b \in \mathbb{F}_p$ são constantes tais que $4a^3 + 27b^2 \neq 0$.

Em outras palavras, o conjunto dos pontos E que pertencem à curva elíptica $E(\mathbb{F}_p)$ é dado por:

$$E = \{(x, y) \in \mathbb{F}_p \times \mathbb{F}_p | y^2 \equiv x^3 + ax + b \ (mod\ p)\} \cup \{O\}$$

A Figura 5.1 mostra os pontos da curva elíptica sobre o corpo $\mathbb{F}_{13}$ que satisfazem a equação modular 5.1 onde $a = 1, b = 0$ e $p = 13$.

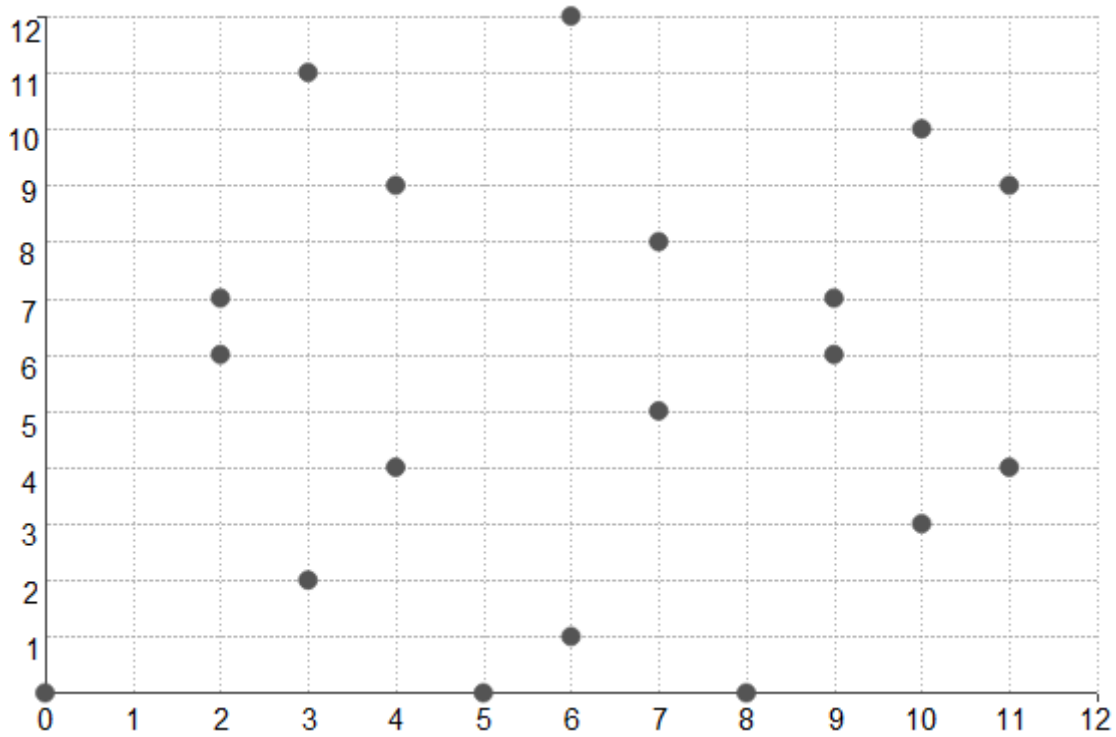


Figura 5.1: Exemplo de curva elíptica sobre um corpo finito

É dito que a ordem de $E(\mathbb{F}_p)$ é igual a $|E|$, isto é, a ordem da curva elíptica é dada pela quantidade de pontos na curva.

Existem diversos formatos de equações que definem curvas elípticas, a equação 5.1 está no formato de Weierstrass, as restrições impostas sobre a e b são para que não exista pontos singulares, isto é, não possuir auto-interseção, nem pontas ou pontos isolados quando a equação for definida sobre o plano $\mathbb{R}^2$.

5.2.1 Operações sobre curvas elípticas

Será definido a operação de soma de pontos que pertencem a uma curva elíptica sobre um corpo finito e apresentado uma notação útil para descrever múltiplas somas.

Definição 5.2.2 *Seja E o conjunto dos pontos da curva $E(\mathbb{F}_p)$, $P, Q \in E$ e O o ponto no infinito. Seja $P = (x_P, y_P)$ e $Q = (x_Q, y_Q)$, para a soma de dois pontos é definido que*

1. $O + O = O$
2. $P + O = P$
3. $P + Q = \begin{cases} O & , se\ (x_P, y_P) = (x_Q, -y_Q) \\ (x_R, y_R) & , caso\ contrário \end{cases}$

   onde $x_R = \lambda^2 - x_P - x_Q \ (mod\ p)$, $y_R = \lambda(x_P - x_R) - y_Q \ (mod\ p)$ e λ é dado por

$$\lambda = \begin{cases} (y_Q - y_P)(x_Q - x_P)^{-1} \ (mod\ p)\ , se\ P \neq Q \\ (3x_P^2 + a)(2y_P)^{-1} \ (mod\ p)\ , se\ P = Q \end{cases} \tag{5.2}$$

É importante enfatizar que na equação 5.2, é denotado por $(x_Q - x_P)^{-1}$ e $(2y_p)^{-1}$ o inverso multiplicativo dos elementos $(x_Q - x_P)$ e $(2y_p)$ módulo p, respectivamente. Além disso, é notável que a operação de soma é comutativa, isto é, $P + Q = Q + P$.

Não é difícil de verificar que se $P, Q \in E$ então $(x_R, y_R) \in E$, concluíndo que a soma de dois pontos na curva resulta em um ponto na curva, portanto a operação de soma é fechada sobre E. Como provado em (Silverman, 2009), o conjunto E junto da operação de adição de pontos, definida acima, formam um grupo abeliano cujo termo neutro da operação é o ponto $\mathcal{O}$.

Note que se $P = (x, y) \in E$ então, diretamente de 5.1, temos que $R = (x, -y) \in E$. Perceba também que o inverso aditivo P é R, portanto todo ponto da curva possui um inverso aditivo que pertence à curva. O ponto inverso de P é usualmente denotado por $-P$.

Seja m um número inteiro positivo, é denotado por mP a soma consecutiva de P m vezes, isto é $mP = P + P + ... + P$ onde P ocorre m vezes. Se $m = 0$, então $mP = \mathcal{O}$. Se m representa um numero negativo, então $mP = -m(-P)$, ou seja, é equivalente a somar $-P$ a ele mesmo m vezes. Em alguns contextos a soma repetida do ponto P a ele mesmo é chamada de multiplicação de ponto por escalar.

Seja $m, n \in \mathbb{Z}$, temos as seguintes propriedades úteis para computar a multiplicação por escalar:

- $mP + nP = (m + n)P$
- $m(nP) = n(mP) = (mn)P$

Além diso, temos as definições da ordem de um ponto e do grupo gerado por um ponto.

Definição 5.2.3 *A ordem de um ponto P é definida como o menor inteiro positivo r tal que $rP = \mathcal{O}$.*

Definição 5.2.4 *O conjunto gerado por P, denotado por $\langle P \rangle$, é definido por*

$$\langle P \rangle = \{kP \mid k \in [0..r-1]\}$$

É dito que P é gerador de $\langle P \rangle$. Se $Q \in \langle P \rangle$, então é dito que Q é gerado por P.

5.2.2 Multiplicação de ponto por escalar de forma eficiente

Seja P um ponto em uma curva elíptica e k um número inteiro não negativo, kP pode ser trivialmente computado fazendo sucessivas somas do ponto P, portanto esse algoritmo faria uma quantidade de somas da ordem $O(k)$. Porém kP pode ser computado de maneira eficiente usando o Algoritmo 5.

Algoritmo 5 Double and Add(k,P):

1: **if** $k = 0$ **then**
2: **return** O
3: **end if**
4: Escrever k em sua representação binária $k = e_{\beta-1}e_{\beta-2}...e_1e_0$ onde cada e_i possui valor 0 ou 1
5: $P' \leftarrow O$
6: **for** $i \in \{\beta - 1, ..., 0\}$ **do**
7: $P' \leftarrow 2P'$
8: **if** $e_i = 1$ **then**
9: $P' \leftarrow P' + P$
10: **end if**
11: **end for**
12: **return** P'

Seja $e_{\beta-1}e_{\beta-2}...e_1e_0$ a representação binária de k. Então para cada i começando de $e_{\beta-1}$ até e_0, é verificado o valor de e_i. Se $e_i = 1$, então deve dobrar e somar o ponto P ao resultado, caso contrário, realiza-se apenas a operação de dobrar o ponto. O bit mais significativo é $e_{\beta-1}$ e será usado para inicialização com ponto P, seu valor será sempre 1 exceto quando $k = 0$.

A Tabela 5.1 exemplifica como computar $77P$ eficientemente, note que a representação binária de 77 é 1001101.

e_6	1	P	Inicialização
e_5	0	2P	Dobrar
e_4	0	2(2P)	Dobrar
e_3	1	2(2(2P)) + P	Dobrar e somar
e_2	1	2(2(2(2P)) + P) + P	Dobrar e somar
e_1	0	2(2(2(2(2P)) + P) + P)	Dobrar
e_0	1	2(2(2(2(2(2P)) + P) + P)) + P	Dobrar e somar

2(2(2(2(2(2P)) + P) + P)) + P = 77P

Tabela 5.1: Exemplo de como computar 77P eficientemente

Com o Algoritmo 5 o processo de computar kP faz na ordem de $O(log\ k)$ somas de pontos, portanto é possível realizar a multiplicação por escalar de maneira eficiente. É importante citar que os cálculos das coordenadas foram omitidos, deve-se usar as fórmulas apresentados na seção 5.2.1 para realizar a soma de pontos em curvas elípticas e efetivamente computar as coordenadas.

5.2.3 O problema do logaritmo discreto para curvas elípticas

Seja $p \in \mathbb{Z}$ um número primo, F_p um corpo finito e $E(F_p)$ uma curva elíptica cujos pontos pertencem ao conjunto E, o problema do logaritmo discreto para curvas elípticas (PLDCE) é definido da seguinte forma:

Problema do logaritmo discreto para curvas elípticas (PLDCE)

Entrada: Dois pontos $P, Q \in E$ onde $Q \in \langle P \rangle$ e $r, p \in \mathbb{Z}_{>0}$ representando a ordem de P e a ordem de $\mathrm{E}(F_p)$, respectivamente.
Saída: O valor $t \in \mathbb{Z}$ tal que $Q = tP$.

É interessante citar que não é conhecido algoritmo polinomial no tamanho de p, portanto esse problema é considerado intratável para computadores clássicos.

5.2.4 Curvas elípticas para a geração de chaves assimétricas

Protocolos criptográficos que utilizam curvas elípticas para a geração de chaves assimétricas definem um ponto especial na curva elíptica chamado de ponto gerador P, também chamado de base. Como dito anteriormente, esse ponto pode gerar qualquer ponto em $\langle P \rangle$ com a operação de multiplicação por escalar.

Protocolos criptográficos assimétricos baseados em curva elíptica possuem uma chave pública e uma chave privada. A chave privada é um inteiro k de tamanho suficientemente grande sorteado no momento da geração das chaves, onde k é menor que a ordem de P. A chave pública é dada pelo ponto $Q = kP$.

É interessante notar que computar o ponto Q a partir de P e k é computacionalmente fácil enquanto para computar k a partir de Q e P é equivalente a resolver o PLDCE que com os algoritmos clássicos conhecidos atualmente não pode ser resolvido eficientemente.

Neste capítulo foi definido curvas elípticas sobre um corpo finito da forma $\mathbb{F}_p$, sendo p um número primo, bem como as operações entre pontos da curva. Contudo, é importante mencionar que existem também curvas elípticas sobre corpos finitos da forma $\mathbb{F}_{2^m}$, para algum m inteiro. Nesses casos a equação que define os pontos da curva elíptica e as fórmulas para operação entre pontos são diferentes porém a forma com que a curva elíptica é usada para gerar chaves é invariante.

6 INTERAÇÃO DE ALGORITMOS QUÂNTICOS COM PROTOCOLOS CRIPTOGRÁFICOS BASEADOS EM CURVAS ELÍPTICAS

Em (Shor, 1999), é apresentado dois algoritmos quânticos, um para fatoração de números inteiros e outro para computar o logaritmo discreto em grupos finitos. O segundo algoritmo pode ser aplicado para computar o logaritmo discreto no grupo formado pelos pontos de uma curva elíptica sobre um corpo finito.

Neste capítulo será mostrada uma redução do PLDCE, uma adaptação do algoritmo de Shor para computar o logaritmo discreto em curvas elípticas e uma análise desse algoritmo. O conteúdo deste capítulo foi baseado nos algoritmos e nas análises apresentadas em (Shor, 1999; Kaye et al., 2006).

6.1 REDUÇÃO DO PROBLEMA DO LOGARITMO DISCRETO PARA CURVAS ELÍPTICAS

Nesta seção será definido um subproblema do PLDCE e mostrado como reduzir o PLDCE para este subproblema.

Seja $p \in \mathbb{Z}_{>0}$ um número primo, $\mathbb{F}_p$ um corpo finito, $\mathrm{E}(\mathbb{F}_p)$ uma curva elíptica cujo conjunto de pontos é E, o problema do logaritmo discreto para curvas elípticas onde r é primo (PLDCEP) é definido por

Problema do logaritmo discreto para curvas elípticas onde r é primo (PLDCEP)

Entrada: Dois pontos $P, Q \in E$ onde $Q \in \langle P \rangle$ e dois números primos $r, p \in \mathbb{Z}_{>0}$ que representam respectivamente a ordem de P e a ordem de $\mathrm{E}(F_p)$.
Saída: O valor $t \in \mathbb{Z}$ tal que $Q = tP$.

Dada uma instância do PLDCE, isto é, dado dois pontos $P, Q \in \mathrm{E}$ onde $Q = tP$, $r, p \in \mathbb{Z}_{>0}$ representando a ordem de P e a ordem de $\mathrm{E}(\mathbb{F}_p)$, respectivamente. Podemos reduzir a instância do PLDCE em uma instância do PLDCEP como especificado a seguir. Caso r seja primo, a redução está completa. Caso contrário, $r = r_1 r_2$ onde $1 < r_1, r_2 < r$. É possível computar r_1 e r_2 de maneira eficiente com o algoritmo de Shor para fatoração de inteiros apresentado em $(Shor,\ 1999)$. A seguir será considerado que r_1 e r_2 são números primos, porém caso não sejam o método pode ser aplicado recursivamente.

Como $t < r$, existem c_1, c_2 onde $0 \leq c_1 < r_1$ e $0 \leq c_2 < r_2$, tais que

$$t = c_1 r_2 + c_2 \tag{6.1}$$

Fazemos $Q' = r_1 Q$ e $P' = r_1 P$. Por definição $Q = tP$ e com isso temos que

$$\begin{aligned} Q &= tP \\ r_1 Q &= r_1 (tP) \\ r_1 Q &= t(r_1 P) \\ Q' &= tP' \end{aligned} \tag{6.2}$$

Como $r = r_1 r_2$ é a ordem de P, então $r_1 r_2 P = O$, portanto $r_2 P' = O$. Com isso, partindo de 6.2 e usando 6.1, temos que

$$\begin{aligned} Q' &= tP' \\ &= (c_1 r_2 + c_2)P' \\ &= (c_1 r_2)P' + c_2 P' \\ &= c_1(r_2 P') + c_2 P' \\ &= c_1 O + c_2 P' \\ &= c_2 P' \end{aligned}$$

Perceba que encontrar c_2 dada a ordem p de $\mathrm{E}(F_p)$, os pontos $Q', P' \in E$ onde $Q' \in \langle P' \rangle$ e o número primo r_2 que é ordem de P' configura uma instância do PLDCEP. O valor c_2 pode ser usado para resolver a instância do PLDCE original.

Além disso, fazemos $P'' = r_2 P$ e $Q'' = Q + (-c_2)P$. Como $Q = tP$, temos

$$\begin{aligned} Q'' &= tP + (-c_2)P \\ &= (c_1 r_2 + c_2)P + (-c_2)P \\ &= (c_1 r_2 + c_2 - c_2)P \\ &= (c_1 r_2)P \\ &= c_1(r_2 P) \\ &= c_1 P'' \end{aligned}$$

Perceba que encontrar c_1 dada a ordem p de $\mathrm{E}(F_p)$, os pontos $Q'', P'' \in E$ onde $Q'' \in \langle P'' \rangle$ e o número primo r_1 que é ordem de P'' configura uma instância do PLDCEP.

Por último, a partir de c_1, c_2 e r_2 podemos calcular t com a equação 6.1. Portanto o PLDCE pode ser reduzido ao PLDCEP.

6.2 ALGORITMO PARA O PROBLEMA DO LOGARITMO DISCRITO PARA CURVAS ELÍPTICAS

O algoritmo que será apresentado recebe uma instância do PLDCEP, isto é, definido um corpo finito $\mathbb{F}_p$ e uma curva elíptica $E(\mathbb{F}_p)$ são dados dois pontos $P, Q \in E(\mathbb{F}_p)$ onde $Q \in \langle P \rangle$ e dois inteiros r, p representado respectivamente a ordem de P e a ordem de $\mathrm{E}(\mathbb{F}_p)$ com r sendo um número primo, o algoritmo encontra t tal que:

$$Q = tP$$

Note que no PLDCE e no PLDCEP é conhecida a ordem de P. Para uma variação dos problemas onde não fosse conhecido esse valor é possível calcula-lo com o algoritmo de estimação de ordem (Algoritmo 4), como citado em (Shor, 1999).

Além disso, o algoritmo que será apresentado também resolve o PLDCE, eliminando a restrição de r ser um número primo, porém a análise do funcionamento do algoritmo seria mais complexa.

Seja U_P o operador que mapeia

$$U_P : |S\rangle \mapsto |S + P\rangle, \text{ com } S \in \langle P \rangle$$

Considere o estado $|u_k\rangle$ dado por

$$|u_k\rangle = \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{-2\pi i \frac{k}{r} s} |sP\rangle$$

temos que

$$\begin{aligned}
U_P|u_k\rangle &= \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{-2\pi i \frac{k}{r} s} U_P|sP\rangle \\
&= \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{-2\pi i \frac{k}{r} s} |sP + P\rangle \\
&= \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{-2\pi i \frac{k}{r} s} |(s+1)P\rangle \\
&= e^{2\pi i \frac{k}{r}} \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{-2\pi i \frac{k}{r}(s+1)} |(s+1)P\rangle \\
&= e^{2\pi i \frac{k}{r}} \frac{1}{\sqrt{r}} \left(\sum_{s=0}^{r-2} e^{-2\pi i \frac{k}{r}(s+1)} |(s+1)P\rangle + e^{-2\pi i \frac{k}{r} r} |rP\rangle \right) \\
&= e^{2\pi i \frac{k}{r}} \frac{1}{\sqrt{r}} \left(\sum_{s=1}^{r-1} e^{-2\pi i \frac{k}{r} s} |sP\rangle + e^{-2\pi i \frac{k}{r} 0} |0P\rangle \right) \\
&= e^{2\pi i \frac{k}{r}} \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{-2\pi i \frac{k}{r} s} |sP\rangle \\
&= e^{2\pi i \frac{k}{r}} |u_k\rangle
\end{aligned}$$

Conclui-se que $|u_k\rangle$ é um autovetor de U_P com autovalor $e^{2\pi i \frac{k}{r}}$. Agora considere o seguinte estado quântico

$$\frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} |u_k\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{-2\pi i \frac{k}{r} s} |sP\rangle \tag{6.3}$$

Perceba que $|sP\rangle = |O\rangle$ se e somente se $s \equiv 0 \ (mod\ r)$, a amplitude do estado $|O\rangle$ é a soma dos termos onde $s = 0$. Ou seja

$$\frac{1}{\sqrt{r}} \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i \frac{k}{r} 0} = \frac{1}{r} \sum_{k=0}^{r-1} 1 = 1$$

Como a equação 6.3 é um estado quântico, então pela definição de estados quânticos a soma do quadrado das amplitudes deve ser 1. Como a amplitude do estado $|O\rangle$ é 1, a amplitude dos demais estados deve ser 0. Concluímos que

$$\frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} |u_k\rangle = |O\rangle \tag{6.4}$$

Como para qualquer $k \in [0..r-1]$ temos que $|u_k\rangle$ é autovetor de U_P, então o estado quântico da equação 6.4 é uma sobreposição de autovetores de U_P.

Generalizando o operador U_P para um ponto $X \in \langle P \rangle$, temos o operador U_X definido por:

$$U_X : |S\rangle \mapsto |S + X\rangle, \text{ com } S \in \langle P \rangle$$

Como $Q \in \langle P \rangle$ é possível usar o operador genérico para definir U_Q e de maneira similar à demostração do autovalor e autovetor de U_P é possível verificar que $|u_k\rangle$ também é um autovetor de U_Q com autovalor $e^{2\pi i \frac{kt}{r}}$. Perceba que $kt = zr + (kt \ (mod\ r))$ para algum $z \in \mathbb{Z}_{\geq 0}$, então temos que

$$\begin{aligned} e^{2\pi i \frac{kt}{r}} &= e^{2\pi i \frac{zr + (kt \ (mod\ r))}{r}} \\ &= e^{2\pi i \left(z + \frac{kt \ (mod\ r)}{r}\right)} \\ &= e^{2\pi i z + 2\pi i \frac{kt \ (mod\ r)}{r}} \\ &= e^{2\pi i \frac{kt \ (mod\ r)}{r}} \end{aligned}$$

Concluimos que o autovalor de U_Q com respeito ao autovetor $|u_k\rangle$ é equivalente a $e^{2\pi i \frac{kt \ (mod\ r)}{r}}$.

Será empregado no algoritmo os operadores CTR-U_P^x e CTR-U_P^x que utilizam registradores como controle e realizam os mapeamentos descritos nas equações 6.5 e 6.6, respectivamente.

$$\text{CTR-}U_P^x : |x\rangle|S\rangle \mapsto |x\rangle|S + xP\rangle, \text{ com } S \in \langle P \rangle \text{ e } x \geq 0 \tag{6.5}$$

$$\text{CTR-}U_Q^x : |x\rangle|S\rangle \mapsto |x\rangle|S + xQ\rangle, \text{ com } S \in \langle P \rangle \text{ e } x \geq 0 \tag{6.6}$$

Para construir esses operadores é necessário pré-computar todos os pontos da forma $2^k P$ e $2^k Q$ com $k \in [1..n]$, onde n é a quantidade de qubits no registrador de controle. Esses pontos podem ser calculados classicamente em tempo polinomial com o Algoritmo 5. Em seguida, utiliza-se os pontos calculados para criar operadores da forma CT-$U_{2^k P}$ e CT-$U_{2^k Q}$, por fim associa-los de maneira análoga ao circuito descrito na seção 3.4.6.

A ideia geral do algoritmo é aplicar o algoritmo de estimação de autovalor para estimar suficientemente bem os autovalores de U_P^x e U_Q^x e determinar $\frac{k}{r}$ e $\frac{kt \ (mod\ r)}{r}$. Se $k \neq 0$ basta computar

$$t \equiv k^{-1} kt \ (mod\ r) \equiv (k \ (mod\ r))^{-1} (kt \ (mod\ r)) \ (mod\ r) \tag{6.7}$$

Com isso temos os as definições básicas para compreender o algoritmo para resolver o PLDCEP que será apresentado a seguir.

Algoritmo 6 Algoritmo para PLDCEP(P,Q,r,p):

1: $n \leftarrow \lceil log_2(2p) \rceil + 1$
2: Inicializar dois registradores de n qubits (que serão referidos como $|\alpha\rangle$ e $|\beta\rangle$) com $|0\rangle$.
3: Inicializar um registrador (que será referido como $|\psi\rangle$) com o estado $|O\rangle$.
4: Aplicar a QFT_{2^n} sobre $|\alpha\rangle$ e $|\beta\rangle$.
5: Aplicar a operação CTR-U_P^x no registrador $|\psi\rangle$ usando o registrador $|\alpha\rangle$ como controle.
6: Aplicar a operação CTR-U_Q^x no registrador $|\psi\rangle$ usando o registrador $|\beta\rangle$ como controle.
7: Aplicar a $\text{QFT}_{2^n}^{-1}$ nos registradores $|\alpha\rangle$ e $|\beta\rangle$.
8: Medir o registrador $|\alpha\rangle$ para obter o valor x onde $\frac{x}{2^n}$ é uma estimativa de $\frac{k}{r}$ para um $k \in \{0, 1, ..., r-1\}$ aleatório.
9: Medir o registrador $|\beta\rangle$ para obter o valor y onde $\frac{y}{2^n}$ é uma estimativa de $\frac{kt \ (mod\ r)}{r}$ para o mesmo k do passo anterior.
10: Arredondar $\frac{ry}{2^n}$ para o inteiro mais próximo $\widetilde{y}$.
11: Arredondar $\frac{rx}{2^n}$ para o inteiro mais próximo $\widetilde{x}$.
12: **if** $\widetilde{x} = 0$ **then**
13: **return** 'FALHA'
14: **else**
15: $\widetilde{t} \leftarrow \widetilde{y}\,\widetilde{x}^{-1} \ (mod\ r)$
16: **if** $\widetilde{t}\,P \neq Q$ **then**
17: **return** 'FALHA'
18: **else**
19: **return** $\widetilde{t}$
20: **end if**
21: **end if**

Após a inicializção dos registradores, o estado do sistema é o seguinte

$$|0\rangle^{\otimes n}|0\rangle^{\otimes n}|O\rangle = |0\rangle^{\otimes n}|0\rangle^{\otimes n}\left(\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}|u_k\rangle\right)$$

Ao aplicar a QFT_{2^n} sobre cada um dos registradores $|\alpha\rangle$ e $|\beta\rangle$ na linha 4 o circuito se encontra no estado

$$\left(\frac{1}{\sqrt{2^n}}\sum_{a=0}^{2^n-1}|a\rangle\right)\left(\frac{1}{\sqrt{2^n}}\sum_{b=0}^{2^n-1}|b\rangle\right)\left(\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}|u_k\rangle\right)$$

Após a aplicação dos operadores CTR-U_P^x e CTR-U_Q^x, como $|u_k\rangle$ é autovetor dos operadores citados, com a mesma argumentação apresentada no Algoritmo 2 temos que os registradores de controle tem em suas amplitudes o autovalor dos operadores, obtendo o seguinte estado

$$\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}\left(\frac{1}{\sqrt{2^n}}\sum_{a=0}^{2^n-1}e^{2\pi i\frac{k}{r}a}|a\rangle\frac{1}{\sqrt{2^n}}\sum_{b=0}^{2^n-1}e^{2\pi i\frac{kt \ (mod\ r)}{r}b}|b\rangle\right)|u_k\rangle$$

Na linha 7 do algoritmo, é aplicado a $\text{QFT}_{2^n}^{-1}$ sobre $|\alpha\rangle$ e $|\beta\rangle$, obtendo um efeito semelhante ao que é feito no algoritmo de estimação de fase, isto é, $|\alpha\rangle$ possui uma sobreposição da estimativa para o autovalor $e^{2\pi\frac{k}{r}}$ com $k \in [0..r-1]$ e $|\beta\rangle$ possui uma estimativa para o autovalor $e^{2\pi i\frac{kt \ (mod\ r)}{r}}$ para o mesmo k. O estado do sistema quântico após essa operação é o seguinte

$$\frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \left|\widetilde{\frac{k}{r}}\right\rangle \left|\widetilde{\frac{kt \bmod r}{r}}\right\rangle \left|u_k\right\rangle$$

Após a medição dos registradores $|\alpha\rangle$ e $|\beta\rangle$ os autovalores colapsam para algum valor de $k \in [0..r-1]$ e obtemos os valores x e y que do Lema 4.6.1 satisfazem

$$\left|\frac{x}{2^n} - \frac{k}{r}\right| \leq \frac{1}{2^n} \text{ e } \left|\frac{y}{2^n} - \frac{kt \bmod r}{r}\right| \leq \frac{1}{2^n}$$

Ou seja,

$$\left|\frac{xr}{2^n} - k\right| \leq \frac{r}{2^n} \text{ e } \left|\frac{ry}{2^n} - kt \ (mod\ r)\right| \leq \frac{r}{2^n}$$

Fazendo $n = \lceil log_2(p) \rceil + 1$ temos $p < 2^{log_2(p)+1} \leq 2^n$, do Teorema 2.2.5 temos que $r < p$ e portanto $2r < 2p \leq 2^n$. Com isso, verificamos que $\frac{r}{2^n} \leq \frac{1}{2}$, isto é

$$\left|\frac{xr}{2^n} - k\right| \leq \frac{1}{2} \text{ e } \left|\frac{ry}{2^n} - kt \ (mod\ r)\right| \leq \frac{1}{2}$$

Então é possível determinar k e $kt \ (mod\ r)$ apenas arredondando $\frac{rx}{2^n}$ e $\frac{ry}{2^n}$ para o inteiro mais próximo, computar eficientemente k^{-1} com o algoritmo de Euclides extendido e usar a equação modular 6.7 para determinar t.

Como o valor de k é aleatório e $k \in \{0, 1, ..., r-1\}$, então existe uma probabilidade de $\frac{r-1}{r}$ de $k \neq 0$, consequentemente do arredondamento $\widetilde{x}$ ser diferente de 0 e nesse caso não falhar o algoritmo. Com isso, obtém-se uma probabilidade de pelo menos $\left(\frac{r-1}{r}\right)\left(\frac{8}{\pi^2}\right)^2$ de estimar corretamente t. A Figura 6.1 mostra o circuito que resolve PLDCEP.

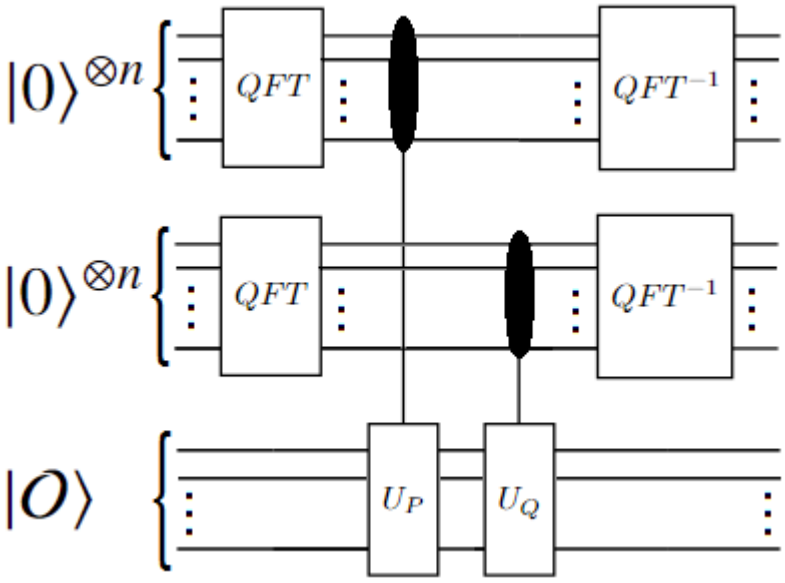


Figura 6.1: Circuito para resolver o PLDCEP

6.3 ANÁLISE DO ALGORITMO

Esta seção é dedicada em fazer uma análise do algoritmo quântico que resolve o PLDCEP.

Segundo (Coppersmith, 2002), os circuitos que implementam a QFT e a QFT^{-1} podem ser construídos com $O((log\ N)^2)$ portas elementares, onde $N = 2^n$. Em (Roetteler et al., 2019) é mostrado uma descrição detalhada de como implementar as operações aritméticas

básicas como computar inverso multiplicativo e multiplicação modular, requeridas no algoritmo utilizando apenas $O(log(p)log\ log(p)log\ log\ log(p))$ portas elementares. Ainda em (Roetteler et al., 2019) é mostrado como o implementar os operadores CTR-U_P^x e CTR-U_Q^x usando $O(n\ log(p)log\ log(p)log\ log\ log(p))$ portas elementares, como $n = O(log(p))$, temos que o custo para implementar os operadores citados é $O(log^2(p)log\ log(p)log\ log\ log(p))$. Portanto o custo total do Algoritmo 6 é de $O(log^2(p)log\ log(p)log\ log\ log(p))$, em comparação o melhor algoritmo clássico conhecido é o algoritmo Pollard's ρ que é da ordem $O(\sqrt{r})$, que é exponencial em relação à quantidade de bits necessária para representar r (Yan, 2015), ou seja o algoritmo de Shor adaptado para curvas elípticas é exponencialmente mais rápido que o melhor algoritmo clássico conhecido.

7 CONCLUSÃO

Criptografias baseadas em curvas elípticas, desde o começo de sua adoção no início dos anos 2000, foram consideradas como uma melhoria comparado aos modelos criptográficos anteriores devido permitirem reduzir consideravelmente o tamanho das chaves geradas fornecendo o mesmo nível de segurança. Contudo, como apresentado nesse estudo, podemos reduzir o PLDCE para o PLDCEP e utilizar um algoritmo quântico para resolvê-lo eficientemente, permitindo calcular a chave privada em tempo polinomial em relação ao tamanho da chave. Portanto criptografias baseadas em curvas elípticas estão ameaçadas pelo avanço da computação quântica e provavelmente se tornarão obsoletas quando esse modelo de computação for implementado na prática de modo eficiente tendo em vista que o modelo de computação quântica permite a criação de algoritmos capazes de resolver o PLDCE de maneira eficiente.

REFERÊNCIAS

Beachy, J. A. e Blair, W. D. (2019). *Abstract algebra*. Waveland Press.

Cheung, D. (2003). *Using generalized quantum Fourier transforms in quantum phase estimation algorithms*. Tese de doutorado, Citeseer.

Cleve, R., Ekert, A., Macchiavello, C. e Mosca, M. (1998). Quantum algorithms revisited. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 454(1969):339–354.

Coppersmith, D. (1994). An approximate fourier transform useful in quantum computing. Relatório técnico, Technical report, IBM Research Division.

Coppersmith, D. (2002). An approximate fourier transform useful in quantum factoring. *arXiv preprint quant-ph/0201067*.

De Wolf, R. (2017). The potential impact of quantum computers on society. *Ethics and Information Technology*, 19(4):271–276.

Diffie, W. e Hellman, M. (1976). New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654.

Forouzan, B. A. e Mukhopadhyay, D. (2015). *Cryptography and network security*, volume 12. New York, NY, USA:: Mc Graw Hill Education (India) Private Limited.

Hardy, G. H., Wright, E. M. et al. (1979). *An introduction to the theory of numbers*. Oxford university press.

Kaye, P., Laflamme, R. e Mosca, M. (2006). *An introduction to quantum computing*. OUP Oxford.

Koblitz, N. (1987). Elliptic curve cryptosystems. *Mathematics of computation*, 48(177):203–209.

Miller, V. S. (1985). Use of elliptic curves in cryptography. Em *Conference on the theory and application of cryptographic techniques*, páginas 417–426. Springer.

Nielsen, M. A. e Chuang, I. (2002). Quantum computation and quantum information.

Roetteler, M., Lauter, K. e Svore, K. (2019). Quantum resource estimates for computing elliptic curve discrete logarithms. US Patent 10,430,162.

Scott, W. R. (2012). *Group theory*. Courier Corporation.

Shor, P. W. (1999). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332.

Silverman, J. H. (2009). *The arithmetic of elliptic curves*, volume 106. Springer.

VAZIRANI, U. (2013). Quantum mechanics and quantum computation.

Weil, A. (2013). *Basic number theory.*, volume 144.

Yan, S. Y. (2015). *Quantum computational number theory*. Springer.