UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE TELEINFORMÁTICA

PAULO REGIS MENEZES SOUSA

APLICAÇÕES DE CRIPTOGRAFIA QUÂNTICA DE CHAVE PÚBLICA EM ASSINATURAS DE MENSAGENS

FORTALEZA
2013

PAULO REGIS MENEZES SOUSA

APLICAÇÕES DE CRIPTOGRAFIA QUÂNTICA DE CHAVE PÚBLICA EM ASSINATURAS DE MENSAGENS

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia de Teleinformática do Departamento de Teleinformática da Universidade Federal do Ceará, como requisito parcial para obtenção do título de Mestre em Engenharia de Teleinformática. Área de concentração: Eletromagnetismo Aplicado.

Orientador: Prof. Dr. José Cláudio do Nascimento.

FORTALEZA
2013

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca de Pós-Graduação em Engenharia - BPGE

S698a Sousa, Paulo Regis Menezes.
Aplicações de criptografia quântica de chave pública em assinaturas de mensagens / Paulo Regis Menezes Sousa. – 2013.
57 f. : il. , enc. ; 30 cm.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia de Teleinformática, Fortaleza, 2013.
Área de concentração: Eletromagnetismo Aplicado.
Orientação: Prof. Dr. José Cláudio do Nascimento.

1. Teleinformática. 2. Assinatura digital. 3. Comunicação eletrônica. 4. Segurança da informação. I. Título.

CDD 621.38

PAULO REGIS MENEZES SOUSA

APLICAÇÕES DE CRIPTOGRAFIA QUÂNTICA DE CHAVE PÚBLICA EM ASSINATURAS DE MENSAGENS

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia de Teleinformática do Departamento de Teleinformática da Universidade Federal do Ceará, como requisito parcial para obtenção do título de Mestre em Engenharia de Teleinformática. Área de concentração: Eletromagnetismo Aplicado.

Aprovada em: 08/08/2013

BANCA EXAMINADORA

Prof. Dr. José Cláudio do Nascimento (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Rubens Viana Ramos
Universidade Federal do Ceará (UFC)

Prof. Dr. João Luzeilton de Oliveira
Universidade Estadual do Ceará (UECE)

FORTALEZA
2013

A Deus. Aos meus pais, Geraldo e Dionéia e a minha esposa Selma.

AGRADECIMENTOS

Agradeço a Deus por ter me dado forças para trilhar os caminhos para mais esta conquista e pelas inúmeras boas amizades que me ajudaram com paciência e compreensão.

Ao Prof. Dr. José Cláudio do Nascimento, pela excelente orientação.

Aos professores participantes da Banca examinadora Dr. Rubens Viana Ramos e Dr. João Luzeilton de Oliveira pelo tempo e pelas valiosas colaborações e sugestões.

À minha esposa Selma, pelo incentivo e paciência nos momentos mais difíceis.

Aos colegas do Grupo de Informação Quântica (GIQ), pelos momentos de inquietantes reflexões.

Finalmente à CAPES, por ter fornecido o apoio financeiro necessário, com a manutenção da bolsa de estudos.

"Quem luta com monstros deve velar para que, ao fazê-lo, não se transforme também em monstro. Pois se olhares muito tempo para dentro de um abismo, o abismo também olha para dentro de ti." (Friedrich Nietzsche)

RESUMO

As assinaturas digitais são de fundamental importância para as comunicações eletrônicas no mundo todo por garantirem a integridade e autenticidade da informação. Com os avanços da ciência nas áreas da mecânica quântica e a introdução destes novos conceitos nas telecomunicações, a segurança da informação também precisou evoluir e cada vez mais se tem buscado novos sistemas de segurança que forneçam maior integridade e autenticidade que os sistemas clássicos. Dessa forma o objetivo deste trabalho é utilizar as propriedades do problema $QSCD_{ff}$, para a criação de um protocolo de assinatura quântica de mensagens. O problema $QSCD_{ff}$ possui propriedades matemáticas e computacionais para garantir a integridade e autenticidade das assinaturas geradas. O protocolo proposto faz uso de chaves descritas na forma de estados quânticos construídos a partir de permutações de um grupo simétrico e de uma função de *hash* para a compressão da mensagem original. Como entrada o protocolo recebe a mensagem clássica e uma chave privada. Para a geração do estado quântico da assinatura utiliza-se uma permutação como chave privada e o *hash* da mensagem. Gerar tal assinatura sem ter uma chave privada consiste em resolver um problema de encontrar automorfismos não triviais de grafos. A validação deste estado é feita através da aplicação do algoritmo quântico de busca de Grover. Por fim é mostrado que a probabilidade de falsificação da assinatura é negligenciável dado o número de cópias do estado da assinatura.

Palavras-chave: Assinatura quântica, problema $QSCD_{ff}$, algoritmo de Grover, automorfismo de grafos.

ABSTRACT

Digital signatures are critical for electronic communications worldwide by ensuring the integrity and authenticity of information. With the advances of science in the fields of quantum mechanics and the introduction of these new concepts in telecommunications, the security of information also had to evolve and increasingly has sought new security systems that provide better integrity and authenticity of classic systems. Thus, the aim of this work is to use the properties of the problem $QSCD_{ff}$ for creation of a signing messages quantum protocol. The problem $QSCD_{ff}$ has mathematical and computational properties to ensure the integrity and authenticity of the signatures generated. The proposed protocol uses keys described in the form of quantum states constructed from permutations at a symmetric group and a function hash to compress the original message. As input, the protocol receives a classic message and a private key. For the generation of the signature quantum state is used a permutation as a private key and the hash of the original message. Generate this signature without a private key is to solve a problem of finding non-trivial graph automorphisms. The validation of this state is done by applying the Grover's quantum search algorithm. Finally, it is shown that the probability of a false signature is negligible given the number of copies of signature state.

Keywords: Quantum signature, $QSCD_{ff}$ problem, Grover's algorithm, graph automorphism.

SUMÁRIO

LISTA DE ILUSTRAÇÕES

Capítulo 1

Introdução

1.1 O que é criptografia?

Uma das definições formais para "Criptografia" é o estudo das técnicas matemáticas relacionadas a aspectos de segurança da informação, tais como confidencialidade, integridade dos dados, autenticação de entidades e verificação da origem [1].

A palavra criptografia vem do grego *kryptós*, "escondido", e *gráphein*, "escrever", o objetivo da criptografia é transformar uma mensagem legível em outra completamente ilegível ("escondendo" a mensagem original), usando para isso funções matemáticas que tornam (idealmente)impossível que uma pessoa, sem o conhecimento de como a mensagem foi gerada, consiga descobrir o texto original em um tempo adequado [2]. A criptografia faz parte da história humana, pois sempre existiram fórmulas secretas, informações confidenciais e interesses dos mais diversos que não deveriam cair no domínio público ou nas mãos de inimigos.

Criptografar é o ato de embaralhar uma mensagem usando uma senha especial (chave), os dados de forma que a saída não faça sentido aparente a um criptoanalista (aquele que tenta descobrir o texto cifrado e/ou a lógica utilizada em sua encriptação).

Descriptografar é o ato de pegar a mensagem cifrada e com o uso de uma senha (chave) ela possa ser revertida para o texto original.

Tanto criptografar como decriptografar exigem em algum momento algo que somente os legítimos interessados devem saber, esta informação secreta é denominada *chave*. A chave é uma informação que não pode ser divulgada a participantes não autorizados na comunicação.

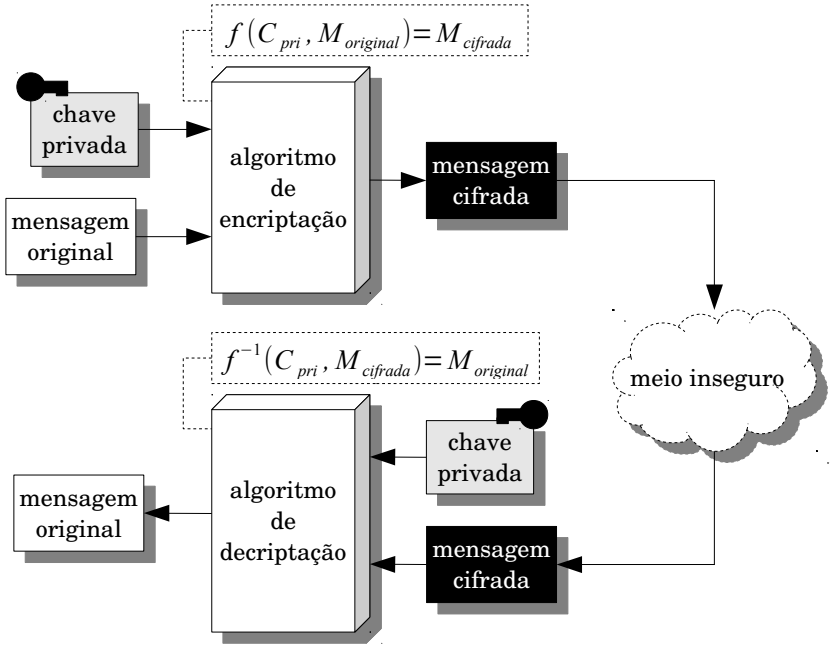
Atualmente, criptografia é mais do que isso, pois além de proteger informações contra pessoas não autorizadas existe, dentre outros, a autenticação cujo objetivo é garantir a identidade do usuário participante.

1.2 Tipos de criptografia em relação ao uso de chaves

Existem dois tipos básicos de criptografia, cada uma com características, vantagens e desvantagens, são estes a criptografia simétrica e a criptografia assimétrica [1].

A criptografia simétrica é basicamente a criptografia onde se utiliza uma única chave tanto para cifrar como para decifrar a mensagem como é mostrado na figura 1.1 abaixo. Esta chave é chamada de *chave secreta* pois apenas os participantes da comunicação devem conhecê-la, dessa forma pode-se manter a comunicação secreta.

Figura 1.1 – Processo de criptografia de chave simétrica



Fonte: Elaborada pelo autor.

Já a criptografia assimétrica é a criptografia onde se utiliza uma chave para encriptar e uma outra diferente para decriptar, sendo que a chave de encriptação é de domínio público (chave pública) e a chave de decriptação é mantida em segredo (chave privada). Por isso, este tipo de criptografia também é conhecido como criptografia de chave pública.

Uma função é unidirecional se é fácil calcular o valor de $f(x)$ da função para qualquer x e é muito oneroso, dado v, determinar um x tal que $f(x) = v$. Em outras palavras podemos dizer que uma função é unidirecional se o seu cálculo for computacionalmente viável, mas o da sua inversa é inviável. Por exemplo, se temos dois números primos da ordem de 10^{100}: atualmente sua multiplicação é realizada em questão de segundos, no entanto, dado o seu produto da ordem de 10^{200}, o melhor algoritmo clássico conhecido leva hoje cerca de 1 bilhão de anos para fatorar o produto dado. Dessa forma

a função produto de dois primos é uma função unidirecional.

Há um tipo de função unidirecional na qual pode-se utilizar uma informação especial chamada de ***trapdoor*** para tornar a computação da sua inversa viável [3]. A função produto de dois primos é unidirecional, como citado anteriormente, mas não possui uma informação *trapdoor* que torne sua inversa viável. Um exemplo de função *trapdoor* é a função modular utilizada no algoritmo RSA [4]. Para transformar uma mensagem $M_{original}$, numa mensagem cifrada usando a chave pública do destinatário, representada por um par de números n e e, basta fazer uma potenciação modular

$$M_{cifrada} = M_{original}^{e} \mod n \tag{1.1}$$

A mensagem então pode ser transmitida em canal inseguro para o receptor.

Para recuperar a mensagem $M_{original}$ da mensagem cifrada $M_{cifrada}$ usando a respectiva chave privada do receptor, um par de números n e d, basta fazer outra potenciação modular

$$M_{original} = M_{cifrada}^{d} \mod n \tag{1.2}$$

Ao escolher uma função unidirecional como função de cifragem, o projetista deve supor que:

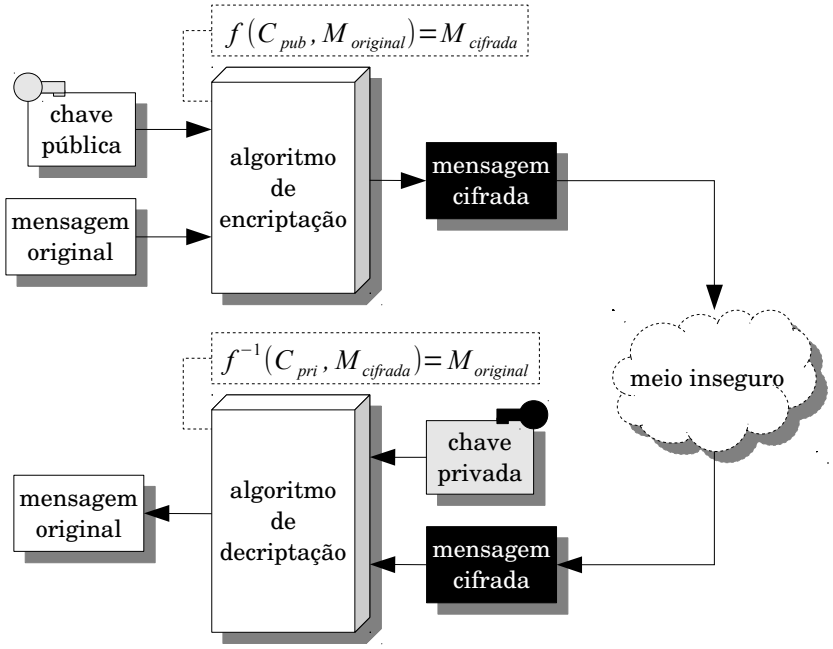
- o algoritmo de cifragem é de domínio público;
- um espião, através de escuta, tem acesso ao texto cifrado.

Diz-se então que a criptoanálise é de texto cifrado conhecido.

A figura 1.2 mostra a estrutura do processo de criptografia de chave pública onde os algoritmos de encriptação e decriptação se utilizam do conceito de função unidirecional *trapdoor*. Uma chave, chamada chave pública, é usada na função f para cifrar a mensagem original, enquanto a outra, chamada chave secreta, é usada para calcular a inversa da função, e assim, decifrá-la.

Uma mensagem cifrada com uma chave pública só pode ser decifrada pela chave secreta com a qual está relacionada.

Figura 1.2 – Processo de criptografia de chave pública



Fonte: Elaborada pelo autor.

1.3 Assinaturas digitais

Com o crescente uso das redes de computadores por organizações para conduzir seus negócios e a massificação do uso da internet, surgiu a necessidade de se utilizar melhores mecanismos para prover a segurança das transações de informações confidenciais.

A questão *segurança* é bastante enfatizada, principalmente, quando imagina-se a possibilidade de ter suas informações expostas a atacantes ou intrusos da *internet*, que surgem com meios cada vez mais sofisticados para violar a privacidade e a segurança das comunicações. De fato um dos principais objetivos da criptografia clássica é garantir a veracidade sobre a origem de uma mensagem [5]. Se A transmite alguma informação para B por um canal inseguro, quer-se ter a certeza de que ela não foi adulterada no trajeto, isto é, a criptografia se preocupa com a autenticidade dos dados que trafegam em um canal e as assinaturas digitais tem fornecido meios para que isso aconteça.

As assinaturas digitais modernas são códigos de autenticação de mensagens que resultam de sistemas de criptografia de chave pública. Para assinar uma mensagem, uma função *Message Digest* (MD) é usada para processar o documento, produzindo um pequeno pedaço de dados, chamado de *hash*, por isso estas funções também são chamadas de funções de *hash* [1].

As **funções de *hash*** são funções que recebem dados de comprimento arbitrário, comprimem estes dados e devolvem um número fixo de bits. Satisfazendo alguns

requisitos adicionais esse tipo de função pode ser usada em aplicações criptográficas como, por exemplo, proteger a autenticidade de mensagens enviadas através de canais inseguros. A ideia básica é que o *hash* forneça uma identidade única para uma mensagem [1].

Este tipo de função precisa ser unidirecional, pois uma vez que se tem o código gerado não deve ser possível a recuperação da mensagem que o gerou. Além disso uma boa função de *hash* precisa ser **resistente a colisões**: isto significa que é "difícil" encontrar duas mensagens distintas que produzam como resultado o mesmo *hash*.

As funções de *hash* são importantes para as assinaturas digitais porque o código gerado por elas é "unido" à mensagem que será transmitida e posteriormente usado para validá-la. Além da validação provida por estas funções as assinaturas de uma forma geral também confiam sua segurança à dificuldade de resolução de certos problemas matemáticos.

As assinaturas digitais foram descritas pela primeira vez no ano de 1974, por Diffie e Hellman [3]. Contudo, apenas em 1978, o primeiro algoritmo que poderia ser usado para gerar assinaturas digitais foi proposto por Rivest, Shamir e Adleman, e ficou conhecido como algoritmo RSA [4]. Eles utilizaram o fato de que é fácil de se obter o resultado da multiplicação de dois números primos extensos, mas é muito difícil de se obter os fatores primos de um número muito extenso.

Com a crescente expansão da capacidade computacional ao longo dos anos e o avanço das técnicas de criptoanálise, alguns dos problemas matemáticos mais utilizados para obtenção da segurança dos sistemas de criptografia de chave pública, deixaram de ser computacionalmente intratáveis. Na década de 90, Peter Shor desenvolveu um algoritmo baseado nos princípios da mecânica quântica capaz de fatorar números muito grandes em tempo polinomial [6]. Este, que ficou conhecido como o Algoritmo de Shor, abalou a base da segurança dos sistemas criptográficos atuais. Isto levou as pesquisas da área de criptografia a se voltarem para a física dos sistemas quânticos, ocasionando o surgimento de grandes revoluções na área de criptografia, entre elas das assinaturas quânticas como equivalentes às assinaturas digitais clássicas, baseadas nos princípios da mecânica quântica.

1.4 Assinaturas quânticas

O artigo seminal para as assinaturas digitais quânticas foi apresentado em 2001 por D. Gottesman e I. L. Chuang que propuseram um sistema de assinatura baseado nos princípios da mecânica quântica com o qual demostraram a existência de um esquema de assinatura de chave pública digital incondicionalmente seguro, algo não realizável classicamente [5].

A entrada desse sistema é uma sequência de bits clássicos e as chaves de assinatura do signatário são estados quânticos. As ideias apresentadas no trabalho demostraram a segurança das assinaturas quânticas e abriram caminho para uma série de inovações nesse ramo.

Em analogia com esquemas de assinatura digital e convencionais, um algoritmo de assinatura quântica deve consistir também de uma assinatura e um algoritmo de verificação. Esses algoritmos devem também, ter uma fase preparatória, que inicializa ou prepara os parâmetros do sistema e cria as chaves [7].

Gottesman e Chuang utilizaram uma abordagem simples em que o emissor seleciona uma cadeia de bits aleatórios e utiliza uma função unidirecional quântica para mapear estes *bits* em estados quânticos usando as chaves públicas. A verificação é realizada quando um destinatário da mensagem realiza o mesmo mapeamento e compara os estados obtidos.

Ainda em 2001, outro trabalho utilizando assinatura quântica para validação de mensagens clássica é proposto por M. Curty e D. J. Santos [8]. Este, se utiliza de um recurso também presente nos sistemas clássicos que é a função de *hash*, usada para comprimir o conteúdo da mensagem em uma *string* de tamanho fixo.

Um ano mais tarde H. Barnum *et al* apresenta um modelo de autenticação para mensagens quânticas [9], que tinha por base a utilização de algoritmos quânticos para validação de uma mensagem também quântica utilizando chaves públicas clássicas [9].

Uma abordagem diferente foi apresentada por G. Zeng e C. H. Keitel, que consiste um esquema de assinatura quântico arbitrado. Sua segurança era baseada na correlação dos estados tripleto GHZ e na utilização de uma cifra de chave única quântica (*one-time pad*)[10]. Contudo neste tipo de esquema de assinatura, todas as comunicações devem envolver um árbitro, que tenha acesso ao conteúdo das mensagens e a segurança da maioria dos esquemas de assinatura arbitrados fica muito dependente da confiança dos árbitros [11][12].

Assim como Gottesman e Chuang, a proposta de trabalho apresentada por X. Lu & D. Feng em 2004 [13] utilizava também o conceito de função de sentido único quântica, neste caso uma *quantum fingerprint* [14], mas o protocolo mostrou-se diferente pela capacidade de assinar estados quânticos gerais.

No ano de 2005 foi proposto por A. Kawachi *et al* [15], um problema computacional, com as propriedades necessárias para o desenvolvimento de um sistema de criptografia de chave pública quântico.

O objetivo deste trabalho é utilizar as propriedades do problema $QSCD_{ff}$, para a criação de um protocolo de assinatura quântica de mensagens. O problema

$QSCD_{ff}$ possui propriedades matemáticas e computacionais para garantir a integridade e autenticidade das assinaturas geradas.

O protocolo proposto faz uso de chaves descritas na forma de estados quânticos construídos a partir de permutações de um grupo simétrico e de uma função de *hash* para a compressão da mensagem original. Como entrada o protocolo recebe a mensagem clássica e uma chave privada.

Para a geração do estado quântico da assinatura utiliza-se uma permutação como chave privada e o *hash* da mensagem original.

A validação deste estado é feita através da aplicação do algoritmo quântico de busca de Grover. Por fim é mostrado que a probabilidade de falsificação da assinatura é negligenciável dado o número de cópias do estado da assinatura.

Esta dissertação foi dividida da seguinte forma: o Capítulo 2 apresenta o funcionamento do algoritmo de busca de Grover utilizado na etapa de verificação da assinatura. No Capítulo 3 serão apresentados alguns conceitos fundamentais afim de auxiliar o entendimento dos problemas descritos no Capítulo 4, onde serão mostrados os principais aspectos do trabalho de Kawachi utilizados aqui. Em seguida, no Capítulo 5, o protocolo de assinatura proposto é descrito. Por fim, o Capítulo 6 apresenta as conclusões e propostas de futuros trabalhos.

Capítulo 2

O Algoritmo quântico de busca de Grover

2.1 Introdução

Um problema de busca é um problema matemático no qual se procura uma solução dentre um espaço de possíveis soluções. Em princípio, bastaria percorrer todo o espaço de possíveis soluções até encontrar a correta. Contudo, tipicamente o tamanho deste espaço é exponencial em relação ao tamanho da sua entrada, tornando o problema computacionalmente intratável [16].

Alguns algoritmos são propostos com base no conhecimento prévio de propriedades de um problema de busca específico para acelerar sua resolução, mas para muitos problemas não se tem um conhecimento profundo de propriedades que facilitem a busca de uma solução. A busca quântica é uma ferramenta para acelerar esses tipos de pesquisas genéricas através de um espaço de possíveis soluções [16].

A estrutura matemática de um problema de busca pode ser definida da seguinte forma. Assume-se que a solução pode ser expressa por uma *string* binária de tamanho n. É definida então uma função $f : \{0,1\}^n \to \{0,1\}$ sendo que $f(x) = 1$ se x é uma codificação binária de uma solução do problema, e $f(x) = 0$ do contrário [16].

A função f é fornecida apenas como uma caixa preta U_f. Assim, são necessárias $\Omega(\sqrt{2^n})$ (notação para o melhor caso) aplicações da caixa preta para resolver o problema de busca com alta probabilidade para qualquer entrada. Dessa forma um algoritmo quântico pode promover um ganho quadrático de velocidade sobre uma busca com o melhor algoritmo clássico [16].

2.2 O algoritmo de Grover

Em um conjunto de possíveis soluções com exatamente uma solução $x = w$, realizando-se apenas uma consulta escolhendo um x_1 uniformemente e checando se $f(x_1) = 1$, tem-se uma probabilidade $\frac{1}{2^n}$ de x_1 ser a solução. Se x_1 não for a solução, realizando uma nova consulta ao conjunto $\{0,1\}^n - \{x_1\}$, se tem x_2 com probabilidade $\frac{2}{2^n}$ de x_2 ser a solução. Continuando a realizar tentativas, para k tentativas com $k < 2^n$ este procedimento retornaria o valor correto $x = w$ com probabilidade $\frac{k+1}{2^n}$ [16].

Uma versão quântica deste procedimento poderia retornar uma resposta correta com uma amplitude de probabilidade de $\frac{1}{\sqrt{2^n}}$. Se houvesse uma forma desta versão quântica incrementar a amplitude em $\frac{1}{\sqrt{2^n}}$ após cada nova consulta então se poderia resolver o problema de busca com somente $O(\sqrt{2^n})$ (notação para o pior caso) consultas. Grover elaborou um algoritmo quântico que alcança este incremento de amplitude [16].

Desta forma a busca se torna quadraticamente mais rápida do que pode ser alcançada com um algoritmo clássico. Existindo exatamente uma solução, uma busca por *força-bruta* clássica precisa fazer $\Omega(2^n)$ consultas, enquanto o algoritmo de Grover precisaria apenas de $O(\sqrt{2^n}) = O(2^{\frac{n}{2}})$ consultas [16].

2.2.1 O oráculo

Assumindo que existe um meio de reconhecer uma solução, e que existe um operador quântico U_f, conhecido como oráculo, da seguinte forma.

$$U_f : |x\rangle|b\rangle \overset{U_f}{\longmapsto} |x\rangle|b \oplus f(x)\rangle \tag{2.1}$$

Supondo que o registrador $|b\rangle$ seja configurado para $|0\rangle$ resultaria em

$$U_f : |x\rangle|0\rangle \overset{U_f}{\longmapsto} |x\rangle|f(x)\rangle \tag{2.2}$$

e pela medição do *qubit* no registrador alvo se obtém a resposta à consulta do oráculo para f da seguinte forma.

$$U_f(|x\rangle|0\rangle) = \begin{cases} |x\rangle|1\rangle, \text{ se } |x\rangle = |w\rangle \\ |x\rangle|0\rangle, \text{ se } |x\rangle \neq |w\rangle \end{cases} \tag{2.3}$$

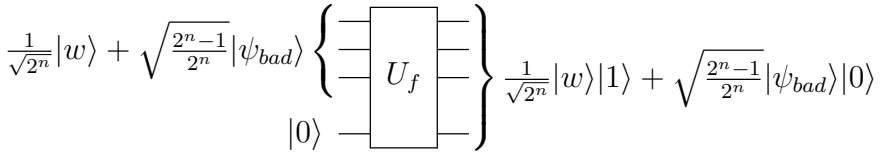
Porém este caso não é melhor que uma aplicação clássica do oráculo para f. A *vantagem quântica* só será alcançada através de superposições.

Pode-se preparar o primeiro registrador com uma superposição de todos os possíveis valores a serem consultados, $\frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$ (onde $N = 2^n$).

A soma $\frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$ pode ser dividida em duas partes. A primeira parte é a soma de todo x para o qual $f(x) = 0$, isto é, todos os valores de x que não são soluções

do problema de busca, este conjunto será denominado X_{bad}. A segunda parte é a soma de todo x para o qual $f(x) = 1$, os valores que são soluções do problema de busca, este conjunto será denominado X_{good}. Por conveniência será assumido que este conjunto tenha apenas uma solução $X_{good} = \{w\}$ [16].

Figura 2.1 – Oráculo U_f para busca quântica



Fonte: Elaborada pelo autor.

Definindo os estados

$$\begin{aligned} |\psi_{good}\rangle &= |w\rangle \\ |\psi_{bad}\rangle &= \frac{1}{\sqrt{N-1}} \sum_{x \in X_{bad}} |x\rangle \end{aligned} \tag{2.4}$$

supondo a preparação do *qubit* alvo de U_f no estado $|0\rangle$ e o registrador da consulta em uma superposição na seguinte forma

$$\frac{1}{\sqrt{N}} \sum\nolimits_{x=0}^{N-1} |x\rangle = \frac{1}{\sqrt{N}}|w\rangle + \sqrt{\frac{N-1}{N}}|\psi_{bad}\rangle \tag{2.5}$$

2.2.2 Operador de mudança de fase

Apenas a criação de uma superposição dos estados de entrada no oráculo resultaria em uma probabilidade de $\frac{1}{N}$ de se obter o estado bom. Contudo o algoritmo de busca quântico é um procedimento iterativo que busca alterar a amplitude do estado bom, então se o valor do segundo registrador for configurado para o estado $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ através da aplicação do operador Hadamard sobre o *qubit* $|1\rangle$

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

o efeito do oráculo é:

$$|x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \xrightarrow{U_f} (-1)^{f(x)} |x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right). \tag{2.6}$$

Como não há ação sobre o segundo registrador deve considerar-se apenas o efeito sobre o primeiro registrador

$$U_f : |x\rangle \longmapsto (-1)^{f(x)}|x\rangle \tag{2.7}$$

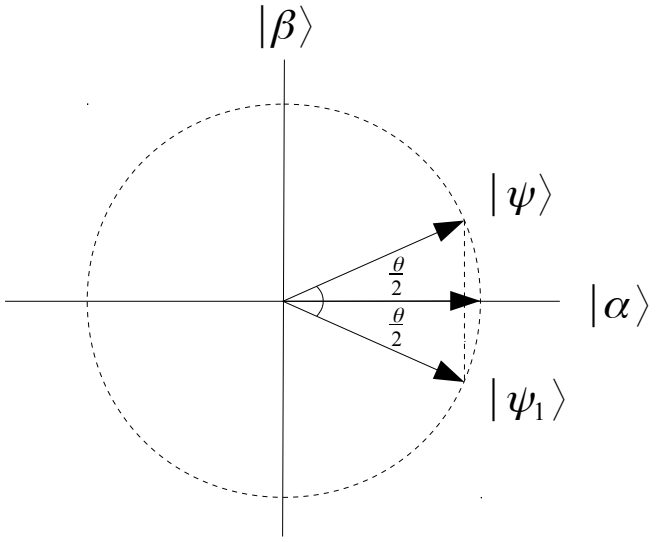
o efeito disto é o de transformar a resposta à consulta do oráculo em uma mudança de fase. Dessa forma será definido um operador de mudança de fase de *n-qubits* $U_{0\perp}$ que age da seguinte forma:

$$U_{0\perp} : \begin{cases} |x\rangle \mapsto -|x\rangle, x \neq 0 \\ |0\rangle \mapsto |0\rangle \end{cases} \tag{2.8}$$

Este operador aplica uma mudança de fase -1 a todo estado que é uma solução do problema [16].

É possível fazer uma interpretação geométrica da aplicação deste operador como mostrado na figura 2.2. O estado do sistema pode ser reescrito utilizando uma

Figura 2.2 – A evolução do sistema para o estado $|\psi_1\rangle = U_{0\perp}(|\psi\rangle)$



Fonte: Elaborada pelo autor.

nova base: $|\beta\rangle$ que representa o vetor que define o elemento procurado pelo oráculo e $|\alpha\rangle$ o seu vetor ortogonal.

Normalmente, nesta nova base, o vetor $|\psi\rangle = a|\alpha\rangle + b|\beta\rangle$ estará mais próximo do vetor $|\alpha\rangle$ pois existem muito mais elementos que não satisfazem o problema. Após a ação do oráculo, os elementos que são soluções são marcados gerando o estado $|\psi_1\rangle = a|\alpha\rangle - b|\beta\rangle$.

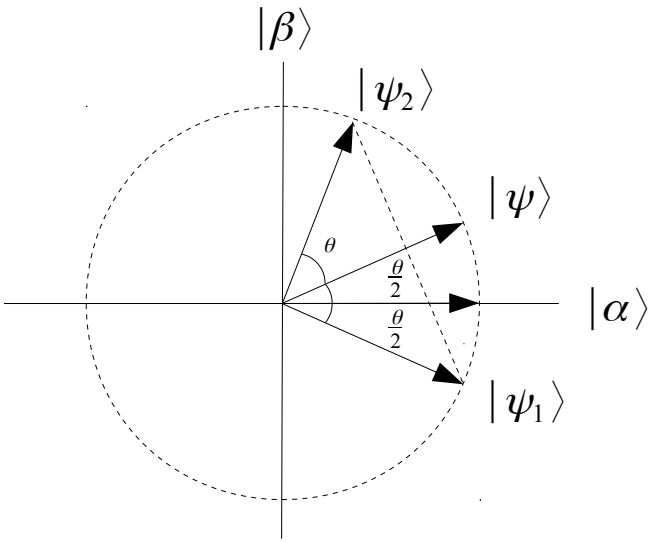
O operador que realiza a inversão de fase é dado pela expressão $HU_{0\perp}H$ que pode ser reformulada como $H^{\otimes n}\left(2|0\rangle\langle 0| - I\right)H^{\otimes n} = 2|\psi\rangle\langle\psi| - I$ onde $|\psi\rangle = \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle$ e I é a matriz identidade correspondente às dimensões de $|\psi\rangle\langle\psi|$ [17]. A sua aplicação em

um estado com elementos previamente marcados, acarreta na amplificação da amplitude de tais elementos [18]. A composição das reflexões impostas pelos operadores U_f e $HU_{0\perp}H$ é conhecida como *iteração de Grover* $G = HU_{0\perp}HU_f = (2|\psi\rangle\langle\psi| - I)\,U_f$.

Iteração de Grover

1. Aplicar o oráculo U_f.
2. Aplicar uma porta Hadamard de *n-qubits*.
3. Aplicar $U_{0\perp}$.
4. Aplicar uma porta Hadamard de *n-qubits*.

Figura 2.3 – A evolução do sistema para o estado $|\psi_2\rangle = G(|\psi_1\rangle)$



Fonte: Elaborada pelo autor.

Dessa forma, a cada iteração de Grover o vetor resultante estará mais próximo do vetor solução.

2.2.3 Número de iterações e desempenho

A iteração de Grover, por ter levado o sistema para mais próximo de $|\beta\rangle$, aumenta a probabilidade de leitura do elemento buscado. Mas existe um limite máximo de aproximação entre os vetores $|\psi\rangle$ e $|\beta\rangle$.

Para demonstrar esse limite de aplicações da iteração (rotação) de Grover, reescreve-se o estado do sistema considerando a possibilidade do oráculo marcar M elementos. Sendo N o número total de elementos do conjuntos, define-se:

- $\sum_x'$ indica a soma dos M elementos que são solução para a busca;
- $\sum_x''$ no caso contrário ($N - M$ elementos).

Usando estas definições os vetores que formam a base do espaço são escritos:

$$|\alpha\rangle = \frac{1}{\sqrt{N-M}} \sum_x'' |x\rangle \qquad \text{e} \qquad |\beta\rangle = \frac{1}{\sqrt{M}} \sum_x' |x\rangle$$

Considerando esta definição dos vetores da base, o estado inicial é definido por: $|\psi\rangle = \sqrt{\frac{N-M}{N}}|\alpha\rangle + \sqrt{\frac{M}{N}}|\beta\rangle$, cada iteração de Grover, composta de duas reflexões (FIGURA 2.3), causa uma rotação de θ graus em direção ao vetor $|\beta\rangle$. Desta forma, a rotação do vetor $|\psi\rangle$ de $\arccos(\sqrt{M/N})$ radianos leva o sistema a uma aproximação máxima de $|\beta\rangle$ [19].

Seja $CI(x)$ o inteiro mais próximo do número real x, no qual por convenção o arredondamento é feito para baixo. Repetindo a iteração de Grover um número de vezes igual a

$$R = CI\left(\frac{\arccos\left(\sqrt{M/N}\right)}{\theta}\right) \tag{2.9}$$

onde $R \leq \lceil \pi/2\theta \rceil$, e por tanto o limite inferior de θ dará um limite superior a R. Supondo que $M \leq N/2$, teremos

$$\frac{\theta}{2} \geq \text{sen } \frac{\theta}{2} = \sqrt{\frac{M}{N}} \tag{2.10}$$

de onde se obtém um limite superior para o número de iterações necessárias:

$$R \leq \left\lceil \frac{\pi}{4}\sqrt{\frac{M}{N}} \right\rceil \tag{2.11}$$

Assim, $R = O(\sqrt{N/M})$ iterações de Grover devem ser realizadas afim de se obter uma solução do problema com alta probabilidade. Isso representa um aumento quadrático em relação às $O(N/M)$ operações clássicas [19].

2.2.4 Amplificação de amplitude

O algoritmo de Grover pode ser generalizado para algum algoritmo A que utilize um estado de entrada genérico $|\psi\rangle = \sum_x \alpha_x |x\rangle|e\rangle$ no qual pode haver alguma informação adicional $|e\rangle$ [16].

$$|\psi\rangle \equiv A|00\ldots0\rangle = \sum_x \alpha_x |x\rangle|e\rangle \tag{2.12}$$

Pode-se dividir este estado $|\psi\rangle$ em duas partes

$$|\psi\rangle = \sum_{x \in X_{good}} \alpha_x |x\rangle|e\rangle + \sum_{x \in X_{bad}} \alpha_x |x\rangle|e\rangle \tag{2.13}$$

Onde

$$p_{good} = \sum_{x \in X_{good}} |\alpha_x|^2 \tag{2.14}$$

é a probabilidade de se medir um estado bom e

$$p_{bad} = \sum_{x \in X_{bad}} |\alpha_x|^2 = 1 - p_{good} \tag{2.15}$$

que é a probabilidade de se medir um estado ruim. Assim os estados $|\psi_{good}\rangle$ e $|\psi_{bad}\rangle$ podem ser redefinidos como

$$|\psi_{good}\rangle = \sum_{x \in X_{good}} \frac{\alpha_x}{p_{good}} |x\rangle|e\rangle \tag{2.16}$$

e

$$|\psi_{bad}\rangle = \sum_{x \in X_{bad}} \frac{\alpha_x}{p_{bad}} |x\rangle|e\rangle \tag{2.17}$$

e o estado $|\psi\rangle$ reescrito como

$$|\psi\rangle = \sqrt{p_{good}}|\psi_{good}\rangle + \sqrt{p_{bad}}|\psi_{bad}\rangle \tag{2.18}$$

ou

$$|\psi\rangle = \text{sen}(\theta)|\psi_{good}\rangle + \cos(\theta)|\psi_{bad}\rangle. \tag{2.19}$$

Partindo de um estado inicial $|\psi_0\rangle$, o estado resultante de k aplicações de Q é

$$|\psi_k\rangle = Q^k|\psi_0\rangle = \text{sen}[(2k+1)\theta]|\psi_{good}\rangle + \cos[(2k+1)\theta]|\psi_{bad}\rangle \tag{2.20}$$

Assim o menor inteiro positivo $k = t$ tal que

$$\text{sen}[(2k+1)\theta]$$

é o mais próximo possível de 1, e portanto $(2k+1)\theta$ o mais próximo de $\pi/2$ possível é

$$t = \left\lfloor \frac{\pi}{4\theta} - \frac{1}{2} \right\rfloor \tag{2.21}$$

Pode-se determinar o angulo θ notando que o angulo ϕ entre $|\psi\rangle$ e $|\psi_{good}\rangle$ é o complementar de θ, dessa forma $\phi + \theta = \pi/2$ e portanto,

$$\frac{1}{\sqrt{N}} = \langle\psi_{good}|\psi\rangle = \cos\alpha = \cos(\frac{\pi}{2} - \theta) = \text{sen}\theta. \tag{2.22}$$

Assim o angulo θ é dado por

$$\theta = \text{arcsen}(\frac{1}{\sqrt{N}}) \tag{2.23}$$

e portanto

$$t = \left\lfloor \frac{\pi}{4\text{arcsen}(\frac{1}{\sqrt{N}})} - \frac{1}{2} \right\rfloor = \left\lfloor \frac{\pi}{4\text{arcsen}(\sqrt{p_{good}})} - \frac{1}{2} \right\rfloor \tag{2.24}$$

Assim se obtém o número de iterações t do algoritmo de Grover em função da probabilidade de se obter a solução do problema, quando esta é conhecida. Isso será útil para a definição do número de iterações que serão necessárias para a validação da assinatura gerada na seção 5.3.

Capítulo 3

Conceitos básicos de permutações e grafos

3.1 Grupos de permutações

Em teoria dos grupos uma **permutação** é uma função bijetiva $f : A \mapsto A$ que mapeia um conjunto A nele próprio. Seja A um conjunto e $S(A) = \{f : A \mapsto A/f \text{ é bijetiva}\}$. Então, $S(A)$, com a operação $\circ$ (composição de funções) é um grupo, não necessariamente abeliano. $S(A)$ é chamado *grupo de simetrias* de A ou *grupo de permutações* de A. Em particular, se $A = \{1, 2, \ldots, n\}$, então $S(A)$ será denotado por S_n, e denominado de grupo simétrico de grau n.

Uma permutação $\alpha/ : A \mapsto A, A = \{1, 2, \ldots, n\}$, em notação matricial, pode ser escrita como

$$\alpha = \begin{pmatrix} 1 & 2 & \ldots & n \\ \alpha(1) & \alpha(2) & \ldots & \alpha(n) \end{pmatrix} \tag{3.1}$$

Observa-se em 3.1 que pode-se obter $\alpha(1)$ de n maneiras, $\alpha(2)$ de $n-1$ maneiras, *etc.* Assim, da análise combinatória, tem-se que a ordem de S_n é dada por $O(S_n) = n(n-1)\cdot\cdots\cdot 2\cdot 1 = n!$. Por exemplo, se $A = \{1, 2, 3\}$, então $S_3 = \{id, \alpha, \beta, \alpha^2, \alpha\beta, \alpha^2\beta\}$, onde

$$id = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix} \alpha = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix} \beta = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix}$$

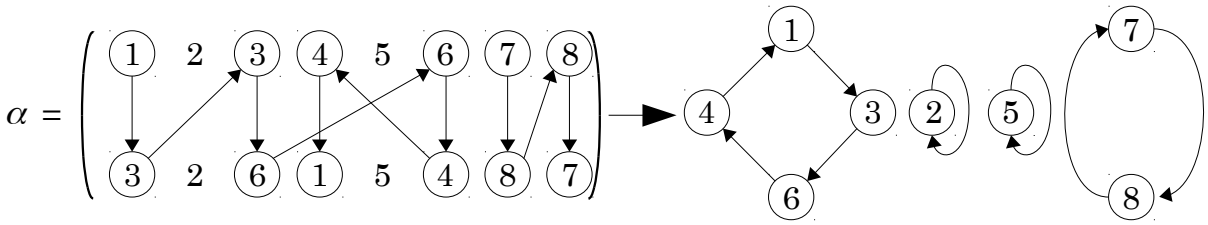
$$\alpha^2 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix} \alpha\beta = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix} \alpha^2\beta = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}$$

Como são bijeções as permutações possuem inversas, denotadas por α^{-1}, por exemplo

$$\alpha^{-1} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 5 & 4 & 3 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} 2 & 5 & 4 & 3 & 1 \\ 1 & 2 & 3 & 4 & 5 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 1 & 4 & 3 & 2 \end{pmatrix}$$

Além da forma matricial, podemos expressar permutações de uma forma mais conveniente através da *notação cíclica*. Um *ciclo* de uma permutação pode ser visto como sendo uma permutação que fixa todos os elementos exceto os que aparecem no ciclo, como mostrado na figura 3.1, a notação cíclica para a permutação que aparece nesta mesma figura é dada por $\alpha = (1364)(2)(5)(78)$, ou simplesmente $\alpha = (1364)(78)$.

Figura 3.1 – Ciclos de uma permutação

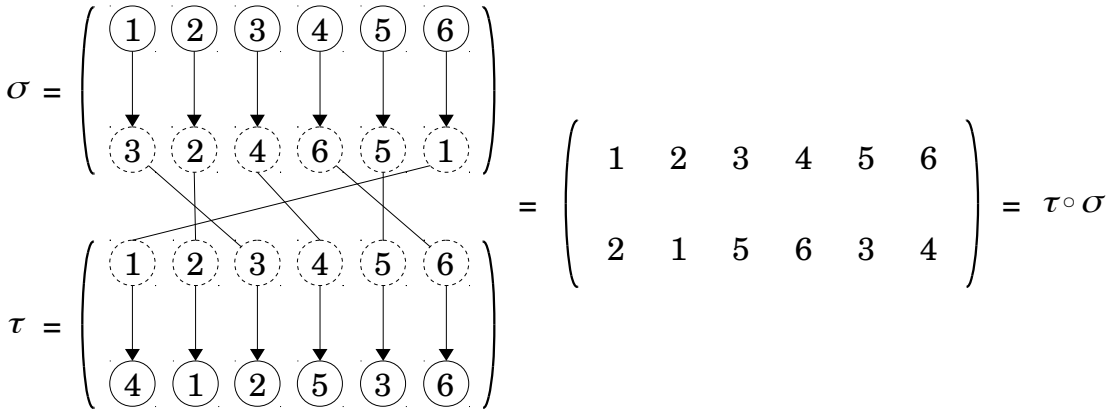


Fonte: Elaborada pelo autor.

Um outro conceito a ser considerado sobre permutações é a composição delas. A composição, $\tau\sigma$, de duas permutações σ e τ, mostradas abaixo, aparece na figura 3.2.

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 2 & 4 & 6 & 5 & 1 \end{pmatrix} \tau = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 4 & 1 & 2 & 5 & 3 & 6 \end{pmatrix}$$

Figura 3.2 – Composição de permutações



Fonte: Elaborada pelo autor.

O comprimento de um ciclo é dado pela quantidade de elementos que ele contém. Uma permutação α pode ser escrita como um produto de seus ciclos, isto é, uma composição das permutações que representam cada ciclo individualmente. Como exemplo, considere a permutação

$$\alpha = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 3 & 2 & 6 & 1 & 5 & 4 & 8 & 7 \end{pmatrix},$$

que pode ser escrita como

$$\alpha = \overbrace{\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 3 & 2 & 6 & 1 & 5 & 4 & 7 & 8 \end{pmatrix}}^{ciclo\ (1364)} \circ \overbrace{\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 4 & 5 & 6 & 8 & 7 \end{pmatrix}}^{ciclo\ (78)} = (1364) \circ (78)$$

Assim a **ordem de uma permutação** é dada pelo menor múltiplo comum (MMC) entre os comprimentos dos ciclos que formam a permutação. Para este exemplo a ordem da permutação α é $MMC(4, 2) = 4$.

3.1.1 Sinal de uma permutação

As permutações que possuem um único ciclo de dois elementos são denominadas **tranposições**. Claramente transposições são permutações de ordem 2 mas nem toda permutação de ordem 2 é uma transposição. Toda permutação pode ainda ser representada por um produto de tranposições, isto é um produto de ciclos de ordem dois.

Uma permutação é chamada de **permutação par**, se ela pode ser representada como um produto de um número par de transposições, ou **permutação ímpar**, se ela pode ser representada pelo produto de um número ímpar de transposições. Isto significa que a permutação é formada por um certo número de inversões de dois elementos e é par se o número de inversões é par e ímpar do contrário. Esta denominação, *par* ou *ímpar*, é conhecida como **sinal da permutação**, que é representado por +1 ou -1 respectivamente.

3.1.2 Matriz de permutação

Uma outra forma de representação de uma permutação é através de uma matriz, chama *matriz de permutação*, ou seja, uma matriz $P_\pi(e_{\pi(i)})_{n\times n}, 1 \leq i \leq n$, onde os elementos das linhas de índice i e coluna de índice $j = \pi(i)$ são iguais a 1 e todos os outros elementos são 0. Assim,

$$\pi = \begin{pmatrix} 1 & 2 & \dots & n \\ \pi(1) & \pi(2) & \dots & \pi(n) \end{pmatrix} = \begin{bmatrix} e_{\pi(1)} \\ e_{\pi(2)} \\ \vdots \\ e_{\pi(n)} \end{bmatrix} = P_\pi.$$

Por exemplo,

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 4 & 2 & 5 & 3 \end{pmatrix} = \begin{bmatrix} e_{\pi(1)} \\ e_{\pi(2)} \\ e_{\pi(3)} \\ e_{\pi(4)} \\ e_{\pi(5)} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} = P_\pi$$

Observe que uma matriz de permutação, nada mais é do que uma permutação das linhas da matriz identidade.

Sob esta representação uma composição de permutações é dada pela multiplicação de duas matrizes, veja o exemplo abaixo. Sejam as permutações α e β, com suas respectivas matrizes de permutações.

$$\alpha = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix} = \overset{P_\alpha}{\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}}, \quad \beta = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix} = \overset{P_\beta}{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}}.$$

Então,

$$\alpha\beta = P_{\beta\circ\alpha} = \overset{P_\beta}{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}} \cdot \overset{P_\alpha}{\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}} = \overset{P_{\alpha\beta}}{\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}}$$

$$P_{\alpha\beta} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix} = \beta \circ \alpha$$

O sinal de uma permutação está diretamente relacionado ao determinante da matriz de permutação associada a ela, pois o determinante resultará em 1 caso a quantidade de inversões nas linhas seja par e -1 caso essa quantidade seja ímpar. Por exemplo, sendo

$$P_\alpha = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix},$$

há uma inversão entre as linhas (tendo como base a matriz identidade) e o determinante de P_α é $\det(P_\alpha) = -1$. Portanto, a permutação associada a esta matriz é dita ímpar, mas para a matriz

$$P_\beta = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix},$$

há duas inversões entre as linhas e o determinante de P_β é $\det(P_\beta) = 1$. Portanto, a permutação associada a esta matriz é dita par.

Para finalizar esta correspondência entre as formas de representação de permutações, como foi dito anteriormente, as permutações possuem a propriedade de inversão. Como as matrizes de permutação são matrizes ortogonais, isto é, $P_\pi P_\pi^T = I$, temos que P_π e P_π^T são matrizes inversíveis e assim suas inversas podem ser escritas como $P_\pi^{-1} = P_{\pi^{-1}} = P_\pi^T$

3.2 O conceito de grafo

Um grafo é uma estrutura matemática $G = (V, E)$ formada por um conjunto finito e não vazio V, cujos elementos são denominados vértices e um conjunto E, de pares de elementos de V, denominados arestas. O número de vértices (também conhecido como ordem do grafo) e o número de arestas de G são indicados, respectivamente, por $n = |V|$ e $m = |E|$. Cada aresta $e \in E$ será denotada por $e = \{v, w\}$, onde os vértices v e w são os extremos de e. Dois vértices que possuem uma aresta em comum são chamados de adjacentes. Da mesma forma arestas adjacentes são aquelas que possuem em comum um mesmo vértice.

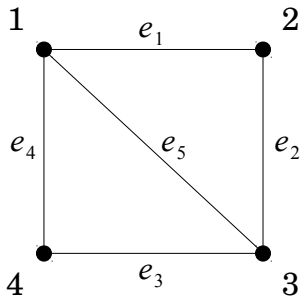
Dizemos que um grafo é orientado se $\{v, w\} \neq \{w, v\}$, porém neste trabalho serão considerados apenas grafos não orientados, isto é, onde $\{v, w\} = \{w, v\}$.

Um grafo não orientado pode ser definido em função dos vértices e arestas, por exemplo $V = \{1, 2, 3, 4\}$ e $E = \{e_1, e_2, e_3, e_4, e_5\}$ como mostrado na figura 3.3.

Um grafo H é dito *subgrafo* de G quando $V(H) \subseteq V(G)$ e $E(H) \subseteq E(G)$ na figura 3.4 podemos ver um subgrafo do grafo mostrado na figura 3.3.

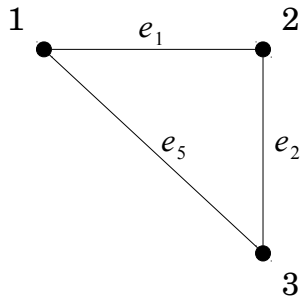
A quantidade de arestas em que um vértice está ligado é chamada de *grau* do vértice e um grafo é dito k-regular se todos os seus vértices têm grau k e um grafo é dito *completo* quando há uma aresta entre cada par de seus vértices. Deste modo todo grafo completo é $(n-1)$-regular. Um grafo completo com n vértices é denotado por K_n.

Figura 3.3 – Grafo simples não orientado



Fonte: Elaborada pelo autor.

Figura 3.4 – Exemplo de subgrafo



Fonte: Elaborada pelo autor.

3.2.1 Matriz de adjacência

Além da representação gráfica o grafo também pode ser representado sob a forma de matrizes. Dado um grafo $G = (V, E)$ não orientado, com n vértices e m arestas. A **matriz de adjacência** do grafo é a matriz $A(G) = (a_{ij})_{n \times n}$ onde $a_{ij} \in \{0, 1\}$, sendo $a_{ij} = 1$ quando os vértices i e j forem adjacentes e $a_{ij} = 0$ do contrário.

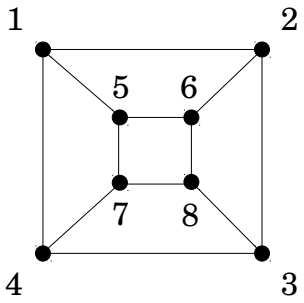
Abaixo pode-se ver a matriz de adjacência do grafo para o grafo da figura 3.3.

$$A(G) = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \tag{3.2}$$

3.2.2 Isomorfismo e automorfismo

Uma função bijetiva $f : V(G) \mapsto V(H)$ que realiza o mapeamento dos vértices de G para os vértices de H, sendo que se dois vértices u e v são adjacentes então $f(u)$

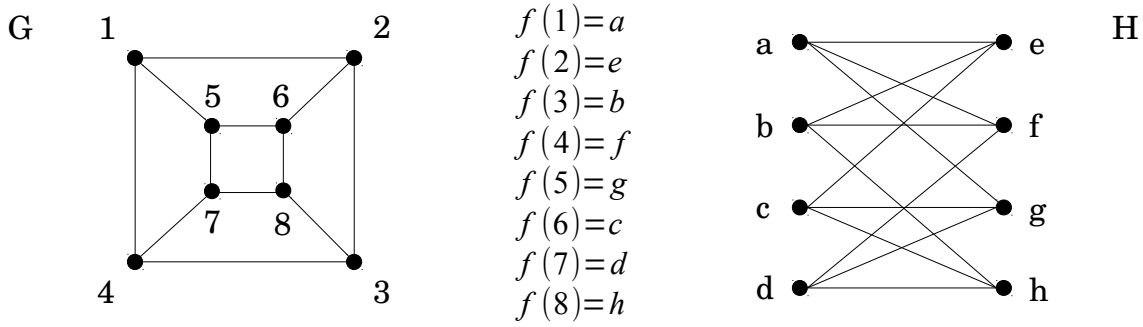
Figura 3.5 – Exemplo de grafo 3-regular



Fonte: Elaborada pelo autor.

e $f(v)$ também são, é chamada de **isomorfismo** entre os grafos G e H. Assim os dois grafos podem apresentar formas diferentes, contudo manterem certos aspectos estruturais em comum.

Figura 3.6 – Grafos isomorfos



Fonte: Elaborada pelo autor.

Em teoria dos grafos o problema computacional de determinar se dois grafos finitos são isomorfos é chamado de *problema do isomorfismo de grafos*. Até então sabe-se que pertence à classe NP de complexidade computacional, porém não se conseguiu definir ainda a qual de seus subconjuntos pertence. A generalização deste problema é o *problema do isomorfismo de subgrafos*, que dados dois grafos G e H, verifica se um grafo H é ou não subgrafo de G. Esta generalização sabe-se ser NP-completo.

Ainda, em teoria dos grafos

Uma forma especial de isomorfismo acontece de quando um grafo é mapeado em si mesmo, preservando a conectividade entre vértices e arestas, e é chamada de automorfismo. Um automorfismo pode ser visto como uma permutação σ do conjunto de vértices V de forma que para toda aresta $e = \{u, v\}$, existe uma aresta correspondente

$\sigma(e) = \{\sigma(u), \sigma(v)\}$.

O mapeamento identidade de um grafo em si é também um automorfismo e é chamado de automorfismo trivial do grafo. O *problema do automorfismo de grafos* é o problema de testar se um grafo tem um automorfismo não trivial. Da mesma forma que no problema do isomorfismo de grafos, não se sabe se ele tem um algoritmo que o resolva em tempo polinomial ou se é NP-completo.

Os conceitos que foram apresentados neste capítulo serão úteis para a compreensão dos conceitos e problemas utilizados nos próximos capítulos.

Capítulo 4

O problema $QSCD_{ff}$

4.1 Introdução

O problema de distinção entre dois estados quânticos específicos, generalizado a partir da distinção entre duas distribuições de probabilidades, foi apresentado em 2006, por Akinori Kawachi *et al* como um novo problema computacional, para a criação de um sistema quântico de criptografia de chave pública, que é seguro contra um adversário quântico de tempo polinomial. Este problema foi intitulado $QSCD_{ff}$ (*quantum state computational distinction with fully flipped permutations*)[15].

Neste capítulo serão apresentados os pontos mais importantes a respeito deste problema, pois o mesmo é de grande importância para este trabalho.

A segurança de um criptossistema construído a partir do problema já foi discutida em outro trabalho [20]. Contudo, apresentaremos de forma sucinta as provas de segurança do sistema além das propriedades que o tornam um problema adequado para a criação de um sistema criptográfico.

4.2 Distinção entre estados quânticos

O problema consiste, basicamente, em distinguir entre conjuntos estatísticos (*ensembles*) específicos de estados quânticos. Para tanto, utiliza-se o pressuposto de que existe um algoritmo quântico de tempo polinomial que consegue distinguir estes *ensembles* com uma vantagem não negligenciável, esta vantagem é definida por Kawachi como segue abaixo [15]. A partir daqui entenda-se $\mathbb{N}$ como o conjunto de todos os inteiros não negativos.

Definição 1. *A* vantagem *de um algoritmo quântico de tempo polinomial $\mathcal{A}$ que distingue entre dois ensembles $\{\rho_0(l)\}_{l\in\mathbb{N}}$ e $\{\rho_1(l)\}_{l\in\mathbb{N}}$ de estados quânticos é a função $\delta_{\mathcal{A}}(l)$ definida*

como:

$$\delta_{\mathcal{A}}(l) = \left| \Pr_{\mathcal{A}} \left[\mathcal{A}\left(\rho_0(l)\right) = 1 \right] - \Pr_{\mathcal{A}} \left[\mathcal{A}\left(\rho_1(l)\right) = 1 \right] \right| \tag{4.1}$$

A saída de $\mathcal{A}$ pode ser arbitrária, neste caso deve-se interpretar 1 como um caso especial indicando que o teste implementado pelo algoritmo $\mathcal{A}$ é "satisfeito", e para qualquer outra saída "não-satisfeito".

Para dois estados de l-qubits $\rho_0(l)$ e $\rho_1(l)$, onde o $\mathcal{A}$ sub-escrito na probabilidade significa que qualquer saída de $\mathcal{A}$, é definida pela medição do seu estado final na base computacional [15]. O problema de distinção tem solução se existe um algoritmo $\mathcal{A}$ e um polinômio p tais que

$$\delta_{\mathcal{A}}(l) = \left| \Pr_{\mathcal{A}} \left[\mathcal{A}\left(\rho_0(l)\right) = 1 \right] - \Pr_{\mathcal{A}} \left[\mathcal{A}\left(\rho_1(l)\right) = 1 \right] \right| > \frac{1}{p(l)} \tag{4.2}$$

para infinitos números l.

Baseado na existência de um algoritmo que possa realizar esta distinção o problema $QSCD_{ff}$ propõe a distinção entre duas sequências idênticas de copias de estados quânticos $\rho_\pi^+(n)$ e $\rho_\pi^-(n)$ de tamanho $n \in N$ onde $N = \{n \in \mathbb{N} : n \text{ é par e } n/2 \text{ é ímpar}\} = \{n \in \mathbb{N} : n \equiv 2 \,(\bmod 4)\}$ e π é uma permutação desconhecida para o adversário. Para todo $n \in N$, seja S_n um grupo simétrico de grau n e seja $\mathcal{K}_n = \{\pi \in S_n : \pi^2 = id \text{ e } \forall i \in \{1, \cdots, n\}[\pi(i) \neq i]\}$, onde id denota a permutação identidade. Dize-se que a permutação é ímpar se ela pode ser representada por um número ímpar de transposições caso contrário, ela é par. Denota-se por sgn a função sinal de uma permutação, definida como $sgn(\pi) = 0$ se π é par e $sgn(\pi) = 1$ se π é ímpar. Para cada $n \in N$, $sign(\pi) = 1$, para toda permutação $\pi \in \mathcal{K}_n$. Assim $\pi \in \mathcal{K}_n$ é uma permutação ímpar já que π consiste de $n/2$ transposições disjuntas, em outras palavras $\pi = (i_1 i_2)(i_3 i_4) \cdots (i_{n-1} i_n)$ para n números distintos $i_1, \cdots, i_n$ em $\{1, \cdots, n\}$.

Definição 2. *Para cada $k \in \mathcal{K}_n$ seja ρ_π^+ e ρ_π^- dois estados quânticos definidos por*

$$\rho_\pi^+(n) = \frac{1}{2n!} \sum_{\sigma \in S_n} \left(|\sigma\rangle + |\sigma\pi\rangle\right)\left(\langle\sigma| + \langle\sigma\pi|\right), \tag{4.3}$$

$$\rho_\pi^-(n) = \frac{1}{2n!} \sum_{\sigma \in S_n} \left(|\sigma\rangle - |\sigma\pi\rangle\right)\left(\langle\sigma| - \langle\sigma\pi|\right). \tag{4.4}$$

O problema $QSCD_{ff}$ tem como objetivo distinguir entre dois estados quânticos $\rho_\pi^+(n)^{\otimes k(n)}$ e $\rho_\pi^-(n)^{\otimes k(n)}$ para cada parâmetro $n \in N$, onde $k(\cdot)$ é um polinômio que pode ser definido por qualquer função de valores inteiros. O parâmetro n é usado para medir a complexidade computacional e é um parâmetro de segurança no criptossistema [15].

4.3 Propriedades criptográficas do $QSCD_{ff}$

Ao longo desta seção, será mostrado que o $QSCD_{ff}$ desfruta das seguintes propriedades criptográficas:

(i) Cada permutação $\pi \in \mathcal{K}_n$ é de ordem 2. Isto fornece a propriedade de função *trapdoor* de $QSCD_{ff}$.

(ii) Para todo $\pi \in \mathcal{K}_n$, a classe conjugada $\{\tau^{-1}\pi\tau : \tau \in S_n\}$ de π é igual a $\mathcal{K}_n$. Esta propriedade permite provar a equivalência entre o caso médio e o pior caso de complexidade de $QSCD_{ff}$.

(iii) O *graph automorphism problem* (GA), ou problema do automorfismo de grafos, é (em uma máquina de Turing de tempo polinomial) equivalente ao seu subproblema *unique graph automorphism problem*, com a premissa de que qualquer grafo dado, ou tem um único automorfismo não trivial em $\mathcal{K}_n$, ou não tem nenhum.

Esta relação de equivalência é usada para dar um limite inferior de complexidade teórica de $QSCD_{ff}$, isto é, o caso médio de complexidade de $QSCD_{ff}$ é limitado inferiormente pelo pior caso de GA.

Para provar estas propriedades, foram introduzidas no trabalho de Kawachi *et al* duas novas técnicas: (i) uma variante do *coset sampling method*, que é largamente usado em várias extensões do algoritmo de Shor [21], e (ii) uma versão quântica do *hybrid argument*, que é uma poderosa ferramenta para muitas reduções de segurança usados na criptografia computacional.

Sejam dois estados $\rho_\pi^+(n) = \frac{1}{2n!}\sum_{\sigma\in S_n}(|\sigma\rangle + |\sigma\pi\rangle)(\langle\sigma| + \langle\sigma\pi|)$ e $\rho_\pi^-(n) = \frac{1}{2n!}\sum_{\sigma\in S_n}|\sigma\rangle - |\sigma\pi\rangle)(\langle\sigma| - \langle\sigma\pi|)$ para uma permutação $\pi \in \mathcal{K}_n$. Por conveniência, será denotado por $\iota(n)$ (ou simplesmente ι), o máximo estado misto $\frac{1}{2n!}\sum_{\sigma\in S_n}|\sigma\rangle\langle\sigma|$ sobre S_n, que aparecerá posteriormente.

4.3.1 Função *trapdoor*

Inicialmente será mostrada a prova apresentada por Kawachi *et al* para garantir que o problema $QSCD_{ff}$ tem uma função *trapdoor* [15]. Para tal, foi suficiente apresentar um algoritmo eficiente de distinção entre ρ_π^+ e ρ_π^- com o conhecimento adicional da permutação oculta $\pi \in \mathcal{K}_n$.

Teorema 1. *(Algoritmo de Distinção). Existe um algoritmo quântico de tempo polinomial que, para qualquer parâmetro $n \in N$ e para qualquer permutação oculta $\pi \in \mathcal{K}_n$, distingue entre $\rho_\pi^+(n)$ e $\rho_\pi^-(n)$ usando π com probabilidade 1 [15].*

Prova. Fixar $n \in N$ arbitrariamente. Seja χ um estado quântico desconhecido que está limitado ou a ρ_π^+ ou ρ_π^-. O algoritmo de distinção desejado para χ funciona da seguinte forma:

(D1) Preparam-se dois registradores quânticos. O primeiro registrador contém um *bit* de controle e o segundo contem χ. Aplica-se a transformação H para o primeiro registrador. O estado do sistema agora torna-se

$$H|0\rangle\langle 0|H \otimes \chi.$$

(D2) Aplica-se o operador Controle-π C_π para ambos os registradores, onde o operador C_π se comporta como $C_\pi|0\rangle|\sigma\rangle = |0\rangle|\sigma\rangle$ e $C_\pi|1\rangle|\sigma\rangle = |1\rangle|\sigma\pi\rangle$ para qualquer dado $\sigma \in S_n$. Assim $\pi^2 = id$ para todo $\pi \in \mathcal{K}_n$, o estado do sistema inteiro pode ser expressado como

$$\frac{1}{n!}\sum_{\sigma\in S_n} \left|\psi_{\pi,\sigma}^+\right\rangle\left\langle\psi_{\pi,\sigma}^+\right| \text{ se } \chi = \rho_\pi^+, \text{ e } \frac{1}{n!}\sum_{\sigma\in S_n} \left|\psi_{\pi,\sigma}^-\right\rangle\left\langle\psi_{\pi,\sigma}^-\right| \text{ se } \chi = \rho_\pi^-,$$

onde $\left|\psi_{\pi,\sigma}^+\right\rangle$ e $\left|\psi_{\pi,\sigma}^-\right\rangle$ são definidos como

$$\begin{aligned} \left|\psi_{\pi,\sigma}^\pm\right\rangle &= C_\pi\left(\frac{1}{2}|0\rangle\left(|\sigma\rangle \pm |\sigma\pi\rangle\right) + \frac{1}{2}|1\rangle\left(|\sigma\rangle \pm |\sigma\pi\rangle\right)\right) \\ &= \frac{1}{2}|0\rangle\left(|\sigma\rangle \pm |\sigma\pi\rangle\right) + \frac{1}{2}|1\rangle\left(|\sigma\pi\rangle \pm \left|\sigma\pi^2\right\rangle\right) \end{aligned}$$

(D3) Aplica-se a transformação Hadamard novamente no primeiro registrador. Como χ é ou ρ_π^+ ou ρ_π^-, o estado do sistema inteiro se torna ou

$$\begin{aligned} (H\otimes I)\left|\psi_{\pi,\sigma}^+\right\rangle &= \frac{1}{\sqrt{2}}|0\rangle\left(|\sigma\rangle + |\sigma\pi\rangle\right) \quad \text{ou} \\ (H\otimes I)\left|\psi_{\pi,\sigma}^-\right\rangle &= \frac{1}{\sqrt{2}}|1\rangle\left(|\sigma\rangle - |\sigma\pi\rangle\right), \end{aligned}$$

respectivamente. Mede-se o primeiro registrador na base computacional. Se o resultado da medição é 0, então a saída é SIM; do contrário, a saída é NÃO.

Está claro que o procedimento acima dá a resposta correta com probabilidade 1.

4.3.2 Uma redução do pior caso para o caso médio

Reduzir o pior caso de complexidade de $QSCD_{ff}$ para o seu caso médio de complexidade implica que $QSCD_{ff}$ com uma permutação aleatória π é pelo menos tão difícil quanto $QSCD_{ff}$ com a permutação fixada π' de maior complexidade. Uma vez que a redução inversa é trivial, o caso médio de complexidade de $QSCD_{ff}$ é portanto equivalente a uma máquina de Turing de tempo polinomial para o pior caso de complexidade (as provas dos lemas e teoremas a seguir foram apresentadas em [15] e encontram-se no Anexo A).

Teorema 2. *Seja k um polinômio qualquer e seja $\mathcal{A}$ um algoritmo quântico de tempo polinomial que resolve $QSCD_{ff}$ com uma vantagem não negligenciável para um $\pi \in \mathcal{K}_n$ aleatório uniformemente distribuído; isto é, existe um polinômio p tal que, para infinitos parâmetros de segurança $n \in N$,*

$$\left| \Pr_{\pi,\mathcal{A}} \left[\mathcal{A}\left(\rho_\pi^+(n)^{\otimes k(n)}\right) = 1 \right] - \Pr_{\pi,\mathcal{A}} \left[\mathcal{A}\left(\rho_\pi^-(n)^{\otimes k(n)}\right) = 1 \right] \right| > \left[\frac{1}{p(n)} \right],$$

onde π é escolhido uniformemente em $\mathcal{K}_n$. Então, existe um algoritmo quântico de tempo polinomial $\mathcal{B}$ que resolve $QSCD_{ff}$ com vantagem não negligível para qualquer permutação $\pi \in \mathcal{K}_n$ [15].

4.3.3 Complexidade computacional

A terceira propriedade do problema $QSCD_{ff}$ é relacionada com a complexidade computacional. Serão apresentadas duas afirmações que mostram sua complexidade relativa frente ao GA ou *graph automorphism problem*. Será mostrado que a complexidade do $QSCD_{ff}$ é inferiormente limitada pela do GA pela construção de uma redução eficiente do GA para o $QSCD_{ff}$. Então será mostrado que $QSCD_{ff}$ não pode ser resolvido por $o(n \log n)$ copias das instâncias de entrada.

A redução do GA para o $QSCD_{ff}$ consiste de duas partes: uma redução do GA para uma variante, chamada $UniqueGA_{ff}$, e uma redução do $UniqueGA_{ff}$ para o $QSCD_{ff}$. Para descrever a redução desejada, o $UniqueGA_{ff}$ será descrito formalmente. Anteriormente Köbler, Schöning e Torán [22] descreveram o $UniqueGA$ da seguinte maneira.

> Unique Graph Automorphism Problem ($UniqueGA$):
> *entrada*: um grafo não direcionado $G = (V, E)$, onde V é um conjunto de vértices e E é um conjunto de arestas;
> *premissa*: G ou tem um único automorfismo não trivial ou não tem automorfismo não trivial;
> *saída*: SIM se G tem um único automorfismo não trivial e NÃO caso contrário.

A premissa do problema $UniqueGA$ é chamada de $(1GA, GA)$ [22]. O *unique graph automorphism with fully-flipped permutation* ($UniqueGA_{ff}$) é uma leve modificação do $UniqueGA$. Lembrando que $N = \{n' \in \mathbb{N} : n' \equiv 2 \pmod 4\}$.

> Unique Graph Automorphism with Fully-Flipped Permutation ($UniqueGA_{ff}$):
> *entrada*: um grafo não direcionado $G = (V, E)$, onde V é um conjunto de vértices e E é um conjunto de arestas;

> *premissa*: o número $n = |V|$ de vértices está em N. Além disso, ou G tem um único automorfismo não trivial $\pi \in \mathcal{K}_n$ ou não tem automorfismo não trivial;
> *saída*: SIM se G tem um único automorfismo não trivial e NÃO caso contrário.

Note que toda instância G de $UniqueGA_{ff}$ é definida apenas quando o número n de nós pertence a N. Em relação a $UniqueGA_{ff}$, foram estabelecidos os dois lemas seguintes, cujas provas foram apresentadas por Kawachi *et al* [15]. Por conveniência será denotado por $\iota = \frac{1}{2n!}\sum_{\sigma\in S_n}|\sigma\rangle\langle\sigma|$, o máximo estado misto sobre S_n.

Lema 1. *Existe um algoritmo quântico de tempo polinomial que, dada uma instância G de $UniqueGA_{ff}$, gera um estado quântico ρ_π^+ se G é "SIM" uma instância com seu único automorfismo não trivial π, ou gera ι se G "NÃO" é instância [15].*

Sendo que $n \in N$. Dada uma instância G de $UniqueGA_{ff}$, primeiro é preparado o estado quântico $\frac{1}{\sqrt{n!}}\sum_{\sigma\in S_n}|\sigma\rangle|\sigma(G)\rangle$, onde $\sigma(G)$ é o grafo resultante do rerotulamento dos seus vértices de acordo com cada permutação π. Pelo descarte do segundo registrador, nós obtemos um estado quântico χ no primeiro registrador.

Se G é "SIM" uma instância com automorfismo único não trivial π, então este estado χ é igual a ρ_π^+ uma vez que

$$\frac{1}{\sqrt{n!}}\sum_{\sigma\in S_n}|\sigma\rangle|\sigma(G)\rangle = \frac{1}{\sqrt{n!}}\sum_{\sigma\in S_n/\langle\pi\rangle}(|\sigma\rangle + |\sigma\pi\rangle)|\sigma(G)\rangle \tag{4.5}$$

.

Caso contrário, como $\sigma(G) \neq \sigma'(G)$ para qualquer $\sigma, \sigma' \in S_n, \chi$ é igual a $\iota = \frac{1}{n!}\sum_{\sigma\in S_n}|\sigma\rangle\langle\sigma|$.

O lema em essência confia no fato de que a permutação oculta $\pi \in \mathcal{K}_n$ é uma permutação ímpar para cada $n \in N$ assim, como uma propriedade especial de $\mathcal{K}_n$, π pode ser expressado como um produto de um número ímpar de transposições.

Lema 2. *Existe um algoritmo de tempo polinomial que, dada uma instância de G de $UniqueGA_{ff}$, gera um estado quântico ρ_π^- se G é "SIM" uma instância com um único automorfismo não trivial π ou gera ι se G "NÃO" é uma instância [15].*

Assim existe uma redução de tempo polinomial de GA para $QSCD_{ff}$. Isto implica que $QSCD_{ff}$ é, pelo menos, tão complexo quanto GA para entradas de tamanhos n infinitamente grandes (e, assim, no pior caso).

Teorema 3. *Se existe um polinômio k e um algoritmo quântico de tempo polinomial que resolve $QSCD_{ff}$ com uma vantagem não negligenciável, então existe um algoritmo quântico de tempo polinomial que resolve GA no pior caso para entradas de tamanhos n infinitamente grandes [15].*

O problema consiste em distinguir os ensembles de estados quânticos $\{\rho_\pi(n)\}_{n\in N}$ e $\{\iota(n)\}_{n\in N}$. Se isto é possível, então a função *trapdoor* utilizada será inversível e não se poderá gerar as assinaturas. O problema de distinguir entre $\{\rho_\pi(n)\}_{n\in N}$ e $\{\iota(n)\}_{n\in N}$ é chamado de $DIST$ em [15]. Hallgren *et al* [23] demonstraram, resolver um DIST de $o(n \log n)$ cópias idênticas é impossível, mesmo para um algoritmo quântico de tempo-ilimitado. A intratabilidade resultante de $DIST$ [23] também é aplicada para o $QSCD_{ff}$ uma vez que $DIST$ pode ser reduzido a $QSCD_{ff}$ em tempo polinomial. Uma solução eficiente para $DIST$ surge da criação de um algoritmo quântico para um certo caso especial do problema do subgrupo oculto em grupos simétricos (SHSP).

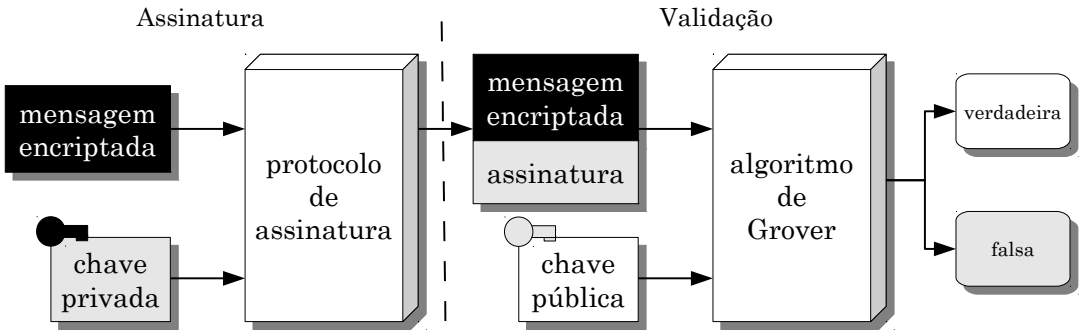
Capítulo 5

Assinatura quântica de mensagens

5.1 Introdução

Neste capítulo será apresentado um esquema de assinatura de mensagens usando o modelo de funções *trapdoor* quânticas proposto por Akinori Kawachi *et al* [15]. Este esquema consiste basicamente na geração de uma assinatura, a partir da mensagem encriptada e de uma chave privada, e da utilização do algoritmo de busca de Grover para a sua validação (FIGURA 5.1).

Figura 5.1 – Fases de assinatura e validação



Fonte: Elaborada pelo autor.

A geração da assinatura é realizável para qualquer máquina quântica de tempo polinomial, pois a entrada tem tamanho polinomial. Contudo gerar tal assinatura sem ter uma chave privada consiste em resolver um problema de encontrar automorfismos não triviais de grafos.

Para a prova de segurança da assinatura deve-se especificar o modelo de ataque do adversário. O modelo de ataque escolhido foi o modelo de ataque de indistinguibilidade contra a escolha de textos abertos para o qual foi construído o seguinte cenário: Alice

(mensageira) quer enviar uma mensagem clássica secreta para Bob via um canal quântico, supondo que Alice e Bob sejam capazes de executar um algoritmo quântico de tempo polinomial.

Bob primeiro gera o estado quântico correspondente à sua chave pública, então Alice requisita esta chave, encripta a mensagem e a envia para Bob. Esta chave faz parte de um criptossistema quântico de chave pública que tem por base o $QSCD_{ff}$.

5.2 Aplicação do $QSCD_{ff}$ a um criptossistema quântico de chave pública

Para a encriptação da mensagem, existe um algoritmo de tempo polinomial que sobre a entrada $\pi \in \mathcal{K}_n$ que gera o estado quântico ρ_π^+ com probabilidade 1 [15]. A prova disto consiste simplesmente na apresentação de um algoritmo capaz de realizar esta tarefa em tempo polinomial de forma determinística. Abaixo é apresentado o protocolo de geração da chave pública [15].

5.2.1 Protocolo de geração da chave pública ρ_π^+

O algoritmo abaixo usa dois registradores quânticos e parte de um estado zero e um estado baseado em uma permutação *identidade*. Estes registradores são usados, através das operações *Hadamard* e π-controlada, para gerar a chave pública.

1. Preparar o estado $|0\rangle|id\rangle$ em dois registradores;
2. Aplicar a operação Hadamard no primeiro registrador, obtendo-se o estado
$$\frac{1}{\sqrt{2}}(|0\rangle|id\rangle + |1\rangle|id\rangle)$$
3. Aplicar a operação π-controlada em ambos os registradores para obter o estado
$$\frac{1}{\sqrt{2}}(|0\rangle|id\rangle + |1\rangle|\pi\rangle)$$
4. Aplicar NOT onde π está presente
$$\frac{1}{\sqrt{2}}(|0\rangle|id\rangle + |0\rangle|\pi\rangle)$$
5. Aplicar uma permutação aleatória σ no segundo registrador
$$\frac{1}{\sqrt{2}}(|0\rangle|\sigma\rangle + |0\rangle|\sigma\pi\rangle)$$

6. Eliminar o primeiro registrador

$$\frac{1}{\sqrt{2}}(|\sigma\rangle + |\sigma\pi\rangle)$$

A segundo Kawachi *et al* [15] existe um algoritmo quântico de tempo polinomial que converte deterministicamente $\rho_\pi^+(n)$ em $\rho_\pi^-(n)$ para $n \in N$ e qualquer $\pi \in \mathcal{K}_n$. Este é o processo que possibilita a encriptação das mensagens, transformando os *bits* da mensagem de zero e um, para ρ_π^+ e ρ_π^-, respectivamente.

5.2.2 Algoritmo de conversão ρ_π^+ em ρ_π^-

Primeiro, recaímos na definição de sinal de uma permutação, denotada por $sign(\cdot)$. Seja $\pi \in K_n$ qualquer permutação fixada. Para seu estado correspondente ρ_π^+, o algoritmo projetado simplesmente inverte a sua fase de acordo com o sinal da permutação. Isso é feito executando a seguinte transformação:

$$|\sigma\rangle + |\sigma\pi\rangle \rightarrow (-1)^{sign(\sigma)}|\sigma\rangle + (-1)^{sign(\sigma\pi)}|\sigma\pi\rangle \tag{5.1}$$

Note que decidir o sinal de uma permutação pode ser feito em tempo polinomial. Dado π ser sempre ímpar, o algoritmo a acima sempre converte $\rho_\pi^+(n)$ em $\rho_\pi^-(n)$. Além disso, o algoritmo não altera a mistura [15]

$$\mathcal{I} = \frac{1}{n!}\sum_{\sigma \in S_n} |\sigma\rangle\langle\sigma| \tag{5.2}$$

5.2.3 Decodificação

Além do algoritmo de conversão, também foi proposto por Kawachi *et al* [15] um algoritmo quântico de tempo polinomial que, para uma permutação conhecida $\pi \in \mathcal{K}_n$, distingue entre ρ_π^+ e ρ_π^- para qualquer $n \in N$ com probabilidade 1, o que possibilita a decodificação da mensagem.

Dessa forma, dado um n fixado arbitrariamente, seja χ um estado desconhecido, que é ρ_π^+ ou ρ_π^-. O algoritmo projetado para distingui-lo funciona da seguinte forma:

1. Preparar dois registradores. O primeiro registrador guarda um bit de controle e o segundo guarda χ. Aplicando a transformação Hadamard H ao segundo registrador. O resultado do sistema torna-se em $H|0\rangle\langle 0|H \otimes \chi$.

2. Aplicando a operação π-controlada C_π nos dois primeiros registradores, em que o operador C_π satisfaz: $C_\pi|0\rangle|\sigma\rangle = |0\rangle|\sigma\rangle$ e $C_\pi|1\rangle|\sigma\rangle = |1\rangle|\pi\sigma\rangle$ para qualquer $\sigma \in S_n$.

3. Dado $\pi^2 = id$ para $\pi \in K_n$, o estado pode ser expresso por:

$$\frac{1}{n!}\sum_{\sigma\in S_n} \left|\psi^+_{\pi,\sigma}\right\rangle\left\langle\psi^+_{\pi,\sigma}\right| \; se \; \chi = \rho^+_\pi \tag{5.3}$$

ou

$$\frac{1}{n!}\sum_{\sigma\in S_n} \left|\psi^-_{\pi,\sigma}\right\rangle\left\langle\psi^-_{\pi,\sigma}\right| \; se \; \chi = \rho^-_\pi \tag{5.4}$$

Em que $\left|\psi^+_{\pi,\sigma}\right\rangle$ é definido por:

$$\begin{aligned}\left|\psi^+_{\pi,\sigma}\right\rangle &= C_\pi\left(\frac{1}{2}|0\rangle\left(|\sigma\rangle + |\sigma\pi\rangle\right) + \frac{1}{2}|1\rangle\left(|\sigma\rangle + |\sigma\pi\rangle\right)\right) \\ &= \frac{1}{2}\left(|0\rangle\left(|\sigma\rangle + |\sigma\pi\rangle\right) + |1\rangle\left(|\sigma\pi\rangle + \left|\sigma\pi^2\right\rangle\right)\right)\end{aligned} \tag{5.5}$$

4. Aplicando a transformação Hadamard para o primeiro registro. Se χ é ρ^+_π , então o estado do sistema torna-se:

$$\begin{aligned}&H \otimes I\left(\frac{1}{2}\left(|0\rangle\left(|\sigma\rangle + |\sigma\pi\rangle\right) + |1\rangle\left(|\sigma\pi\rangle + \left|\sigma\pi^2\right\rangle\right)\right)\right) \\ &= \frac{1}{2\sqrt{2}}\left(\left(|0\rangle + |1\rangle\right)\left(|\sigma\rangle + |\sigma\pi\rangle\right) + \left(|0\rangle - |1\rangle\right)\left(|\sigma\pi\rangle + \left|\sigma\pi^2\right\rangle\right)\right) \\ &= \frac{2}{2\sqrt{2}}|0\rangle\left(|\sigma\rangle + |\sigma\pi\rangle\right) \\ &= |0\rangle\frac{|\sigma\rangle + |\sigma\pi\rangle}{\sqrt{2}}\end{aligned} \tag{5.6}$$

De forma análoga:

$$H \otimes I\left|\psi^-_{\pi,\sigma}\right\rangle = |1\rangle\frac{|\sigma\rangle - |\sigma\pi\rangle}{\sqrt{2}} \tag{5.7}$$

Finalmente, mede-se o primeiro registro na base computacional. Se o resultado é 0, então o estado quântico é ρ^+_π. Caso contrário, o estado é ρ^-_π.

5.2.4 Protocolo quântico de criptografia de chave pública

Agora será descrito o criptossistema quântico de chave pública, dividido em duas fases, a de transmissão da chave e a transmissão da mensagem, como é mostrado na figura 5.2.

Transmissão da chave

1. Bob escolhe uma chave secreta π com distribuição de probabilidade uniforme sobre o conjunto $\mathcal{K}_n$;

2. Bob gera uma quantidade suficientemente grande de cópias da chave pública ρ^+_π;

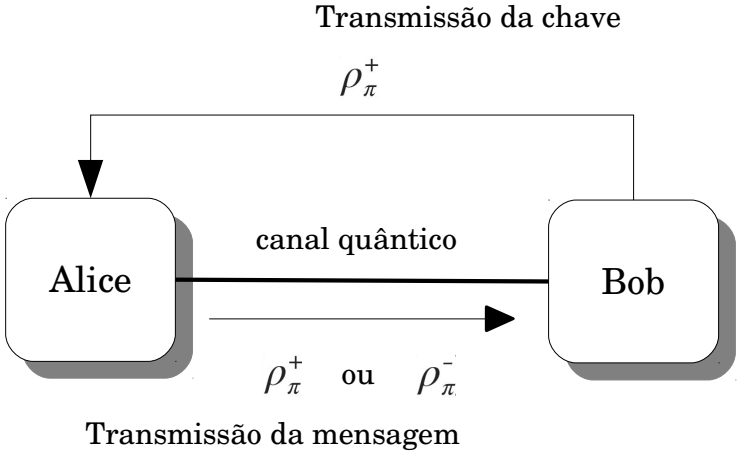
3. Alice obtém uma cópia da chave pública de Bob.

Transmissão da mensagem

4. Alice encripta 0 e 1 em ρ_π^+ e ρ_π^-, respectivamente, e envia a mensagem encriptada a Bob;

5. Bob decripta a mensagem de Alice usando a chave π.

Figura 5.2 – Protocolo de criptografia de chave pública



Fonte: Elaborada pelo autor.

5.3 Protocolo para assinatura digital quântica

Agora será descrito um protocolo de assinatura criptográfica de chave pública usando estados quânticos. Neste esquema, Bob possui uma chave privada π que será usada para assinar uma mensagem m. A assinatura é uma autenticação de que esta mensagem m foi realmente enviada e assinada por Bob. Esta assinatura pode ter a autoria verificada por qualquer usuário a partir da chave pública ρ_π^+ disponibilizada por Bob na rede.

5.3.1 Geração da assinatura a partir da chave privada

Neste esquema de assinatura, Bob aplica a transformação unitária U tal que $U|k\rangle|\sigma\rangle = |k\rangle|\sigma_k\rangle$. Assim, usando o algoritmo de geração da chave pública, ele produz:

$$|\psi_{Sign}\rangle = \frac{|\sigma_k\rangle + |\sigma_k\pi\rangle}{\sqrt{2}} \tag{5.8}$$

Sendo $k = h(m)$ é um *hash* da mensagem m. A função *hash* aqui usada é uma função que mapeia qualquer mensagem de m bits em $|S_n|$ bits, ou seja, $h : \{0,1\}^* \rightarrow$

$\{0,1\}^{|S_n|}$. Assim o que é publicado é um estado quântico $\rho_\pi^{\otimes z}$, onde z obedece o limite de $o(n \log n)$ cópias do estado. Assim o estado da assinatura é dado por

$$|\psi_{Sign}\rangle^{\otimes z} = \left(\frac{|\sigma_k\rangle + |\sigma_k\pi\rangle}{\sqrt{2}}\right)^{\otimes z}. \tag{5.9}$$

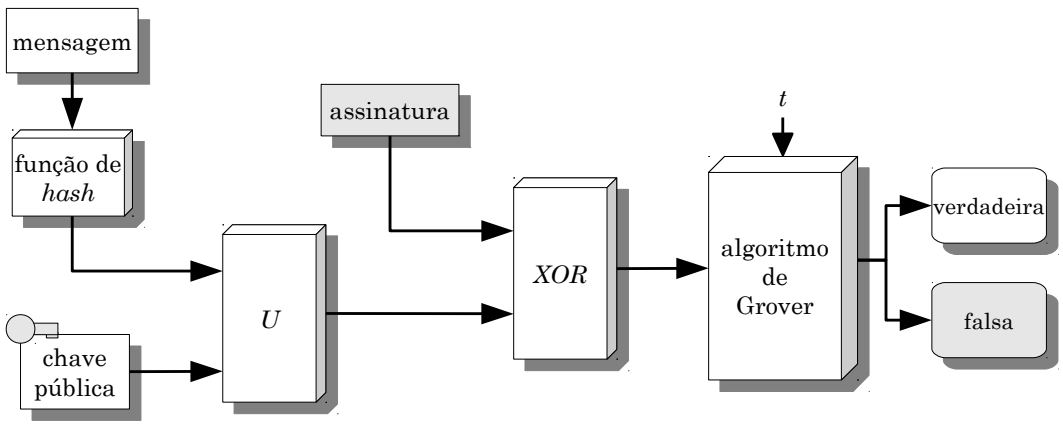
5.3.2 Verificação da assinatura

Para verificar a assinatura, Alice deve gerar um estado quântico a partir da chave pública. É necessário aplicar a operação unitária U no estado quântico $|k\rangle$ e na chave pública $\rho_\pi^{\otimes z}$. Como resultado obtém-se o seguinte estado:

$$\begin{aligned} \left|\psi_{(\sigma\pi)k}\right\rangle &= U|k\rangle \left(\frac{|\sigma\rangle + |\sigma\pi\rangle}{\sqrt{2}}\right) \\ &= |k\rangle \left(\frac{|\sigma_k\rangle + |(\sigma\pi)_k\rangle}{\sqrt{2}}\right). \end{aligned} \tag{5.10}$$

Ainda é necessário realizar uma operação XOR tal que $XOR|a,b,0\rangle = |a,b,a \oplus b\rangle$. Assim Quando Alice desejar verificar a assinatura deve executar os seguintes passos ilustrados na figura 5.3

Figura 5.3 – Verificação da assinatura de Bob



Fonte: Elaborada pelo autor.

1. calcular o *hash* da mensagem m, $h(m) = k$;
2. Realiza a operação $U|k\rangle|\psi_{\sigma,\pi}\rangle$ para obter o estado $\left|\psi_{(\sigma\pi)k}\right\rangle$;
3. Agora realiza uma operação XOR entre os estados quânticos $|\psi_{Sign}\rangle$ e $\left|\psi_{(\sigma\pi)k}\right\rangle$ expressando o resultado num registrador auxiliar;

4. Por fim, o algoritmo de Grover é executado e buscando pelo *qubit* $|0\rangle$ no terceiro registrador, onde é armazenado o resultado da operação XOR entre os $|\psi_{Sign}\rangle$ e $\left|\psi_{(\sigma\pi)k}\right\rangle$. O número de passos usados no algoritmo é fixo e dado pela equação:

$$t = \frac{\pi}{4 \operatorname{arcsen}(\sqrt{p_{good}})} - \frac{1}{2} = 1, \tag{5.11}$$

descrita no Capítulo 2, onde $p_{good} = 1/4$ e t é um valor inteiro positivo representando o número de passos. Assim, Alice aceita a mensagem m assinada por Bob com apenas uma execução do Algoritmo de Grover.

Para justificar a necessidade de apenas uma execução do Algoritmo de Grover é necessário lembrar brevemente os princípios desta operação vistos na seção 2.2.4. Como o algoritmo de Grover é ajustável de acordo com a amplitude do estado quântico. De uma forma simples, consideremos o seguinte estado quântico

$$\operatorname{sen}(\theta)|\psi_{good}\rangle + \cos(\theta)|\psi_{bad}\rangle, \tag{5.12}$$

onde $\theta \in \{0, \pi/2\}$ e $p_{good} = \operatorname{sen}^2(\theta)$. Então o estado $|\psi_{good}\rangle$ representa o estado que se quer encontrar e $|\psi_{bad}\rangle$ são os resultados indesejáveis. O Algoritmo de Grover aplicado t vezes resulta em

$$\operatorname{sen}((2t+1)\theta)|\psi_{good}\rangle + \cos((2t+1)\theta)|\psi_{bad}\rangle, \tag{5.13}$$

para obtermos um bom resultado com alta probabilidade devemos ter $(2t+1)\theta \approx \frac{\pi}{2}$. Como temos $p_{good} = 1/4$, então o número de execuções necessárias para a verificação é dada pelo inteiro

$$t = \frac{\pi}{4 \operatorname{arcsen}(\sqrt{p_{good}})} - \frac{1}{2} = 1. \tag{5.14}$$

Dessa forma o número de execuções do Algoritmo de Grover também é um parâmetro de segurança na verificação da assinatura.

5.4 Análise de segurança

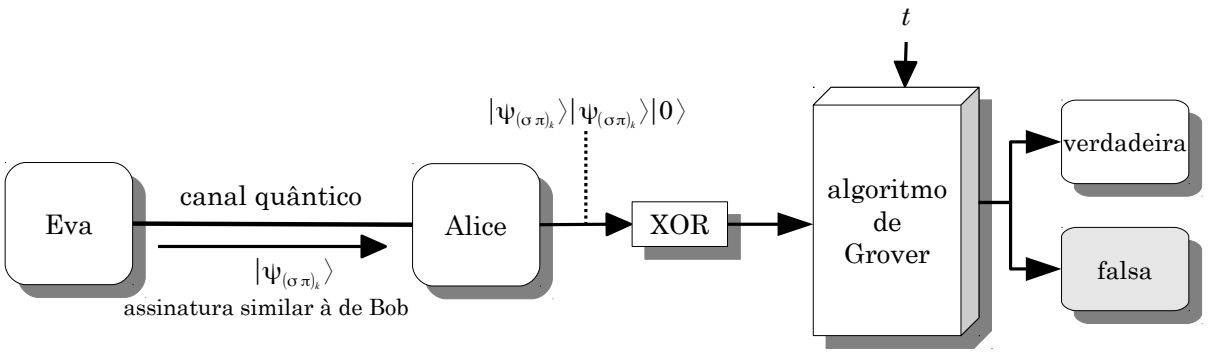
Em muitos senários, o emissor e o receptor de uma mensagem, precisam da garantia de que a mensagem não será alterada durante a transmissão. Se a mensagem é assinada digitalmente, qualquer alteração na mesma irá invalidar a assinatura. Isso se dá, devido à propriedade de resistência a colisões de uma função de *hash* descrita na seção 1.3.

Assinaturas digitais autenticam a fonte das mensagens. Quando a posse da chave secreta de uma assinatura digital está vinculada a um usuário específico, uma

assinatura válida mostra que a mensagem foi enviada realmente por este usuário. A importância de autenticidade do emissor é especialmente óbvia no contexto financeiro. Por exemplo, supondo que um escritório da filial de um banco envie instruções para o escritório central requisitando a alteração no balanço de uma conta. Se o escritório central não tiver certeza de que a mensagem foi realmente enviada por uma fonte autorizada, realizar tal ação poderia ser um grande erro.

A dificuldade computacional necessária para um adversário gerar um estado ρ_π^+ sem conhecimento da chave secreta π, que é equivalente à dificuldade de encontrar um único automorfismo, apresentada na seção 4.3.3, é a mesma de se gerar o estado quântico $|\psi_{Sign}\rangle$. Contudo o dono da chave secreta π pode gerar tanto ρ_π^+, como $|\psi_{Sign}\rangle$ em tempo polinomial. Esta questão é discutida com mais detalhes no capítulo 4. Neste trabalho foi escolhida uma outra estratégia. Em nossa análise de segurança consideramos o caso em que Eva realiza um ataque a partir da chave pública, tentando gerar uma assinatura válida através dela.

Figura 5.4 – Cenário de ataque de Eva



Fonte: Elaborada pelo autor.

5.4.1 Gerando uma assinatura falsa a partir da chave pública

Embora qualquer usuário que não possui a chave privada π não seja capaz de gerar o estado (5.9), a partir da chave pública pode-se gerar o seguinte estado

$$\left|\psi_{(\sigma\pi)_k}\right\rangle = U|k\rangle\frac{|\sigma\rangle + |\sigma\pi\rangle}{\sqrt{2}} = |k\rangle\frac{|\sigma_k\rangle + |(\sigma\pi)_k\rangle}{\sqrt{2}}, \tag{5.15}$$

que pode ser usado como tentativa de falsificação da assinatura por se assemelhar a uma assinatura verdadeira.

Então considere o cenário mostrado na figura 5.4: Eva quer enviar uma mensagem com a assinatura de Bob. Pela dificuldade computacional, ela não é capaz de gerar a mesma a assinatura de Bob, mas a partir da chave pública de Bob ela pode gerar um assinatura semelhante a de Bob.

A verificação é baseada no Algoritmo de Grover e este algoritmo é sintonizável no número de passos t. Desta forma, quando Eva enviar uma mensagem para Alice com uma assinatura falsa de Bob, Alice deverá executar o algoritmo de Grover um certo número de vezes afim de verificar a assinatura.

Temos que a probabilidade a *priori* de se encontrar um estado $|0\rangle$ no terceiro registrador é $p_{good} = 1/4$ para uma assinatura autêntica de Bob. Assim o estado $|\psi_{Sign}\rangle\big|\psi_{(\sigma\pi)_k}\big\rangle|0\rangle$ após a operação XOR é:

$$\begin{aligned} XOR|\psi_{Sign}\rangle\big|\psi_{(\sigma\pi)_k}\big\rangle|0\rangle &= \frac{1}{2}(XOR|\sigma_k,\sigma_k,0\rangle + XOR|\sigma_k,(\sigma\pi)_k,0\rangle \\ &+ XOR|\sigma_k\pi,\sigma_k,0\rangle + XOR|\sigma_k\pi,(\sigma\pi)_k,0\rangle) \\ &= \frac{1}{2}(|\sigma_k,\sigma_k,0\rangle + |\sigma_k,(\sigma\pi)_k,\sigma_k\oplus(\sigma\pi)_k\rangle \\ &+ |\sigma_k\pi,\sigma_k,\sigma_k\pi\oplus\sigma_k\rangle + |\sigma_k\pi,(\sigma\pi)_k,\sigma_k\pi\oplus(\sigma\pi)_k\rangle). \quad (5.16) \end{aligned}$$

A sequência abaixo descreve melhor o que acontece

$$XOR|\psi_{Sign}\rangle\big|\psi_{(\sigma\pi)_k}\big\rangle|0\rangle$$

$$\downarrow$$

$$\frac{1}{2}(XOR|\sigma_k,\sigma_k,0\rangle + XOR|\sigma_k,(\sigma\pi)_k,0\rangle + XOR|\sigma_k\pi,\sigma_k,0\rangle + XOR|\sigma_k\pi,(\sigma\pi)_k,0\rangle)$$

$$\downarrow$$

$$\frac{1}{2}(|\sigma_k,\sigma_k,\underbrace{0}_{good}\rangle + |\sigma_k,(\sigma\pi)_k,\underbrace{\sigma_k\oplus(\sigma\pi)_k}_{bad}\rangle + |\sigma_k\pi,\sigma_k,\underbrace{\sigma_k\pi\oplus\sigma_k}_{bad}\rangle + |\sigma_k\pi,(\sigma\pi)_k,\underbrace{\sigma_k\pi\oplus(\sigma\pi)_k}_{bad}\rangle)$$

Aplicando p_{good} na equação 5.11 temos que o algoritmo só precisa de uma execução para encontrar o estado $|0\rangle$.

Usando esta probabilidade ainda para calcular a probabilidade de erro, para uma assinatura verdadeira temos,

$$\begin{aligned} p_{erro} &= \cos^2\left((2t+1)\ \text{arcsen}\left(\sqrt{p}\right)\right) \\ &= \cos^2\left((2(1)+1)\ \text{arcsen}\left(\sqrt{\frac{1}{4}}\right)\right) \\ &= 2.5861\times10^{-32}. \quad (5.17) \end{aligned}$$

Por outro lado, se Bob repudia sua assinatura, ele então mede seu registador gravado quando ele gerou as assinaturas. Se o resultado é 0, então o estado quântico

guardado por Alice é $|\sigma_k\rangle$. Se o resultado é 1, então o estado quântico guardado por Alice é $|\sigma_k\pi\rangle$. Assim, zeros aparecem com probabilidade $p = 1/2$ quando o verificador da assinatura (Alice) realizar a operação XOR. Esta é expressa pela seguinte equação:

$$\begin{aligned} XOR|\psi_{(\sigma\pi)_k}\rangle|\psi_{(\sigma\pi)_k}\rangle|0\rangle &= \frac{1}{2}(XOR|\sigma_k,\sigma_k,0\rangle + XOR|\sigma_k,(\sigma\pi)_k,0\rangle \\ &+ XOR|(\sigma\pi)_k,\sigma_k,0\rangle + XOR|(\sigma\pi)_k,(\sigma\pi)_k,0\rangle) \\ &= \frac{1}{2}(|\sigma_k,\sigma_k,0\rangle + |\sigma_k,(\sigma\pi)_k,\sigma_k\oplus(\sigma\pi)_k\rangle \\ &+ |(\sigma\pi)_k,\sigma_k,(\sigma\pi)_k\oplus\sigma_k\rangle + |(\sigma\pi)_k,(\sigma\pi)_k,0\rangle) \end{aligned} \tag{5.18}$$

A sequência abaixo descreve melhor o que acontece

$$XOR|\psi_{(\sigma\pi)_k}\rangle|\psi_{(\sigma\pi)_k}\rangle|0\rangle$$

$$\downarrow$$

$$\frac{1}{2}(XOR|\sigma_k,\sigma_k,0\rangle + XOR|\sigma_k,(\sigma\pi)_k,0\rangle + XOR|(\sigma\pi)_k,\sigma_k,0\rangle + XOR|(\sigma\pi)_k,(\sigma\pi)_k,0\rangle)$$

$$\downarrow$$

$$\frac{1}{2}(|\sigma_k,\sigma_k,\underbrace{0}_{good}\rangle + |\sigma_k,(\sigma\pi)_k,\underbrace{\sigma_k\oplus(\sigma\pi)_k}_{bad}\rangle + |(\sigma\pi)_k,\sigma_k,\underbrace{(\sigma\pi)_k\oplus\sigma_k}_{bad}\rangle + |(\sigma\pi)_k,(\sigma\pi)_k,\underbrace{0}_{good}\rangle)$$

Quando o algoritmo de Grover for executado uma única vez neste estado por Alice, ela encontrará o estado $|0\rangle$ com uma probabilidade a *priori* $p = 1/2$. Assim a probabilidade de erro é:

$$\begin{aligned} p_{erro} &= \cos^2\left((2t+1)\text{arcsen}(\sqrt{p})\right) \\ &= \cos^2\left((2(1)+1)\ \text{arcsen}\left(\sqrt{\frac{1}{2}}\right)\right) \\ &= \frac{1}{2}. \end{aligned} \tag{5.19}$$

Portanto, a probabilidade de sucesso e falha na tentativa de falsificação de Eva é a mesma. Neste caso, p é a probabilidade de encontrar zero numa assinatura falsa. Como são enviadas z cópias do estado $|\psi_{Sign}\rangle$, então a probabilidade de Eva enganar Alice é $1/2^z$, um valor negligenciável no número de cópias.

5.4.2 Repúdio

O não-repúdio é uma importante propriedade dos esquemas de assinaturas digitais clássicas. Através dessa propriedade uma entidade que assina digitalmente alguma informação não pode posteriormente negar que realizou a assinatura. O não-repúdio é importante em muitos casos onde assinaturas digitais são aplicadas, no entanto o repúdio pode ser necessário em alguns casos. Por exemplo, suponhamos que uma entidade emita

tickets. Geralmente é definido o quê é permitido a cada ticket, ele pode autenticar a passagem em algum transporte, entrada em um cinema ou teatro, etc. Quando o período de validade expira, então a entidade providenciará novos tickets. Se o usuário não utilizar os tickets no devido tempo, ele vai à entidade e realiza a troca dos tickets antigos por novos tickets. Então os novos tickets terão a data anterior, mas o período de validade e assinatura digital tiveram que ser mudados.

Agora imaginemos que uma entidade emite tickets sem período de validade, mas ela quer ter o poder de invalidá-los a qualquer momento. Uma maneira é invalidar a assinatura. Na criptografia clássica, a entidade poderia emitir uma nova chave pública para invalidar a anterior. Portanto, um esquema de assinatura em uma assinatura onde se pode repudiar uma assinatura a qualquer momento é muito útil neste caso e em muitos outros casos similares.

Agora, consideremos que os passos 5 e 7 na fase de geração da assinatura, mostrados na seção 5.3.1, não sejam executados. A assinatura seria então o estado quântico

$$\frac{|0\rangle|\sigma_k\rangle + |1\rangle|\sigma_k\pi\rangle}{\sqrt{2}}. \tag{5.20}$$

Quando o primeiro registrador é medido o resultado é um estado misto. Tornando impraticável a verificação da assinatura. Por exemplo, sem perda de generalidade, considerar que o resultado da medição no primeiro registrador é zero. Então no segundo registrador é guardado o estado quântico $|\sigma_k\rangle$. Zeros aparecem uma vez quando a operação XOR é realizada na verificação da assinatura como mostra a equação a baixo

$$\begin{aligned} XOR|\sigma_k\rangle\left|\psi_{(\sigma\pi)_k}\right\rangle|0\rangle &= \frac{1}{2}(XOR|\sigma_k,\sigma_k,0\rangle + XOR|\sigma_k,(\sigma\pi)_k,0\rangle) \\ &= \frac{1}{2}(|\sigma_k,\sigma_k,0\rangle + |\sigma_k,(\sigma\pi)_k,\sigma_k\oplus(\sigma\pi)_k\rangle) \end{aligned} \tag{5.21}$$

Quando o algoritmo de Grover for executado uma única vez neste estado por Alice, a probabilidade de erro é:

$$\begin{aligned} p_{erro} &= \cos^2\left((2t+1)\text{arcsen}(\sqrt{p})\right) \\ &= \frac{1}{2}. \end{aligned} \tag{5.22}$$

Como são enviadas z cópias do estado $|\psi_{Sign}\rangle$, então a probabilidade de Eva enganar Alice é $1/2^z$, um valor negligenciável no número de cópias.

Capítulo 6

Conclusão

Neste trabalho foi proposto um protocolo de assinatura de menagens cuja base criptográfica é o problema $QSCD_{ff}$ e o sistema de criptografia de chave pública criado a partir dele.

No protocolo de assinatura de mensagens proposto uma mensagem é assinada através de uma sequência, de tamanho z, de estados quânticos, a partir do texto público da mensagem pela aplicação de uma função de *hash* e de uma chave privada π.

Foi ainda demonstrado que o protocolo proposto é seguro contra um adversário quântico de tempo polinomial cujo melhor estado possível que pode ser construído como uma assinatura falsa é o estado $\left|\psi_{(\sigma\pi)_k}\right\rangle$. Onde a probabilidade do adversário ser bem sucedido na tentativa de falsificação da assinatura é igual a $\frac{1}{2^z}$, valor este que é negligenciável para o número de cópias.

O esquema de assinatura digital quântica proposto possui ainda a propriedade de repúdio. Uma entidade pode gerar a assinatura e depois repudiá-la em outro momento. Esta é a principal das vantagens do protocolo proposto.

A propriedade de repúdio presente neste esquema de assinatura é um resultado de não-localidade. Analisar esta propriedade em assinaturas quânticas é importante por que sistemas quânticos podem ser implementados com entrelaçamento. Operações locais em sistemas entrelaçados podem gerar outras propriedades ou falhas de segurança. Neste trabalho não foi respondido se todas as assinaturas digitais propostas possuem a propriedade de repúdio. Basta verificar essa possibilidade.

REFERÊNCIAS

[1] MENEZES, A. J.; VAN OORSCHOT, P. C.; VANSTONE, S. A. *Handbook of applied cryptography*. CRC press, 2010.

[2] BRUCE, S. Applied cryptography. *2nd John Wiley and Sons, Inc*, 1996.

[3] DIFFIE, W.; HELLMAN, M. New directions in cryptography. *Information Theory, IEEE Transactions on*, v. 22, n. 6, p. 644–654, 1976.

[4] RIVEST, R.; SHAMIR, A.; ADLEMAN, L. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, v. 21, n. 2, p. 120–126, 1978.

[5] GOTTESMAN, D.; CHUANG, I. Quantum digital signatures. *arXiv preprint quant-ph/0105032*, 2001.

[6] SHOR, P. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, v. 26, n. 5, p. 1484–1509, 1997.

[7] Lü, X.; FENG, D.-G. An arbitrated quantum message signature scheme. In: ZHANG, J.; HE, J.-H.; FU, Y. (Eds.) *Computational and Information Science*. Springer Berlin / Heidelberg, 2005. v. 3314 of *Lecture Notes in Computer Science*, p. 1054–1060.

[8] CURTY, M.; SANTOS, D. J. Quantum authentication of classical messages. *Physical Review A*, Woodbury, v. 64, n. 6, p. 062309, 2001.

[9] BARNUM, H.; CRÉPEAU, C.; GOTTESMAN, D.; SMITH, A.; TAPP, A. Authentication of quantum messages. In: . c2002. p. 449–458.

[10] ZENG, G.; KEITEL, C. H. Arbitrated quantum-signature scheme. *Phys. Rev. A*, v. 65, p. 042312, Apr 2002.

[11] LEE, H.; HONG, C.; KIM, H.; LIM, J.; YANG, H. J. Arbitrated quantum signature scheme with message recovery. *Physics Letters A*, Amsterdam, v. 321, n. 5âĂŞ6, p. 295 – 300, 2004.

[12] MEIJER, H.; AKL, S. Digital signature schemes for computer communication networks. In: . c1981. v. 11. p. 37–41.

[13] LU, X.; FENG, D. Quantum digital signature based on quantum one-way functions. In: . c2004. v. 1. p. 514–517.

[14] BUHRMAN, H.; CLEVE, R.; WATROUS, J.; DE WOLF, R. Quantum fingerprinting. *Physical Review Letters*, Woodbury, v. 87, n. 16, p. 167902, 2001.

[15] KAWACHI, A.; KOSHIBA, T.; NISHIMURA, H.; YAMAKAMI, T. Computational indistinguishability between quantum states and its cryptographic application. *Journal of cryptology*, Aarhus, Dinamarca, v. 25, n. 3, p. 528–555, 2012.

[16] PHILLIP, K.; LAFLAMME, R.; MOSCA, M. *An introduction to quantum computing.* Oxford University Press, New York, NY, USA, 2007.

[17] GROVER, L. K. A fast quantum mechanical algorithm for database search. In: . STOC '96. New York, NY, USA: ACM, c1996. p. 212–219.

[18] BRASSARD, G.; HOYER, P.; MOSCA, M.; TAPP, A. Quantum amplitude amplification and estimation. *arXiv preprint quant-ph/0005055*, 2000.

[19] NIELSEN, M.; CHUANG, I. *Quantum computation and quantum information.* Cambridge university press, 2010.

[20] KOSHIBA, T. Security notions for quantum public-key cryptography. *arXiv preprint quant-ph/0702183*, 2007.

[21] REGEV, O. Quantum computation and lattice problems. *SIAM Journal on Computing*, v. 33, n. 3, p. 738–760, 2004.

[22] KÖBLER, J.; SCHÖNING, U.; TORÁN, J. *The graph isomorphism problem: its structural complexity.* Birkhauser Verlag, 1994.

[23] HALLGREN, S.; MOORE, C.; RÖTTELER, M.; RUSSELL, A.; SEN, P. Limitations of quantum coset states for graph isomorphism. In: . c2006. p. 604–617.

ANEXO A - PROVAS DOS LEMAS E TEOREMAS

Prova do teorema 2

Fixe um parâmetro arbitrário $n \in N$ que satisfaça a suposição do teorema. Assuma que a entrada é ou $\rho_\pi^+(n)^{\otimes k(n)}$ ou $\rho_\pi^-(n)^{\otimes k(n)}$. Para cada $i \in 1, 2, ..., k(n)$, seja χ_i o i-ésimo estado dos $k(n)$ estados. Claramente χ_i é ou ρ_π^+ ou ρ_π^-. Para o caso médio dado algoritmo A, nós construímos o pior caso desejado do algoritmo B da seguinte maneira:

(R1) Escolhe-se uma permutação $\tau \in S_n$ uniformemente ao acaso.

(R2) Aplica-se τ para cada χ_i, onde $i \in 1, 2, ..., k(n)$, a partir da direita. Se $\chi_i = \rho_\pi^+$, então nós obtemos o estado quântico

$$\begin{aligned} \chi_i' &= \frac{1}{2n!} \sum_{\sigma \in S_n} \left(|\sigma\tau\rangle + \left|\sigma\tau\tau^{-1}\pi\tau\right\rangle \right) \left(\langle\sigma\tau| + \left\langle\sigma\tau\tau^{-1}\pi\tau\right| \right) \\ &= \frac{1}{2n!} \sum_{\sigma' \in S_n} \left(|\sigma'\rangle + \left|\sigma'\tau^{-1}\pi\tau\right\rangle \right) \left(\langle\sigma'| + \left\langle\sigma'\tau^{-1}\pi\tau\right| \right). \end{aligned}$$

(R3) Invoca-se o caso médio do algoritmo $\mathcal{A}$ na entrada $\bigotimes_{i=1}^k \chi_i'$.

(R4) Retorna o resultado de $\mathcal{A}$.

Seja $\pi \in \mathcal{K}_n$. Note que, para cada $\tau \in S_n$, $\tau^{-1}\pi\tau$ pertence a $\mathcal{K}_n$. Além disso, para todo $\pi' \in \mathcal{K}_n$, existe um $\tau \in S_n$ que satisfaz $\tau^{-1}\pi\tau = \pi'$, de onde resulta que a classe conjugada $\{\tau^{-1}\pi\tau : \tau \in S_n\}$ de π é igual a $\mathcal{K}_n$. Como mostrado abaixo, o número de todas as permutações $\tau \in S_n$, para os quais $\tau^{-1}\pi\tau = \pi'$ é independente da escolha de $\pi' \in \mathcal{K}_n$.

Afirmação 1. *Para qualquer permutação* $\pi, \pi', \pi'' \in \mathcal{K}_n$, $|\{\tau \in S_n : \tau^{-1}\pi\tau = \pi'\}| = |\{\tau \in S_n : \tau^{-1}\pi\tau = \pi''\}|$.

Prova. Defina um mapeamento $\mu_\tau : \mathcal{K}_n \to \mathcal{K}_n$ como $\mu_\tau(\sigma) = \tau^{-1}\sigma\tau$ e um conjunto $\mathcal{T}_{\pi\pi'} := \mu_\tau : \mu_\tau(\pi) = \pi'$. É óbvio que, por definição a operação "$\cdot$" em um grupo como $\mu_\tau \cdot \mu_{\tau'}(\cdot) = \mu_\tau(\mu_{\tau'}(\cdot)), \mathcal{T}_{\pi,\pi}$ resulta em um subgrupo $S_n := \mu_\tau : \tau \in S_n$. Portanto, S_n tem uma decomposição de *coset* com o seu respectivo subgrupo $\tau_{\pi,\pi}$ para todo $\pi \in \mathcal{K}_n$ e cada *coset* coincide com $\mathcal{T}_{\pi,\pi'}$ para um determinado π'. Isto mostra que $|\mathcal{T}_{\pi,\pi'}| = |\mathcal{T}_{\pi,\pi''}|$ para

todo par π, π''. Como μ_τ e τ tem uma relação um-para-um, segue-se que, para todo par π, π'', $|\tau \in S_n : \tau^{-1}\pi\tau = \pi'| = |\tau \in S_n : \tau^{-1}\pi\tau = \pi''|$.

As propriedades mencionadas acima implicam que $\tau^{-1}\pi\tau$ é de fato uniformemente distribuído sobre $\mathcal{K}_n$. Portanto, alimentando a entrada $\bigotimes_{i=1}^{k} \chi_i'$ para o algoritmo $\mathcal{A}$, nós podemos alcançar a vantagem não negligível desejada de $\mathcal{A}$. Isto completa a prova.

Prova do lema 2

Seja $n \in N$. Similar ao algoritmo dado na prova do lema 1, nós começamos com o estado $\frac{1}{\sqrt{n!}} \sum_{\sigma \in S_n} |\sigma\rangle\langle\sigma(G)|$ em dois registradores. Calculamos o sinal de cada permutação no primeiro registrador e então invertemos a sua faze apenas onde a permutação é ímpar. Consequentemente, nós obtemos o estado quântico $\frac{1}{\sqrt{n!}} \sum_{\sigma \in S_n} (-1)^{\text{sign}(\sigma)} |\sigma\rangle\langle\sigma(G)|$. Lembrando que $sign(\sigma) = 0$ se σ é par e $sign(\sigma) = 1$ do contrário. Pelo descarte do segundo registrador, nós imediatamente obtemos um certo estado, chamado, χ no primeiro registrador. Note que, como π é ímpar, se σ é ímpar (par, respectivamente) então $\sigma\pi$ é par (ímpar, respectivamente). Portanto, se segue que $\chi = \rho_\pi^-$ se G é "SIM" uma instância com o único automorfismo não trivial π, e $\chi = \iota$ do contrário.

Prova do teorema 3

Primeiro foi mostrado que GA é equivalente em tempo polinomial ao $UniqueGA_{ff}$. Depois, foi mostrda uma redução de tempo polinomial do $UniqueGA_{ff}$ para o $QSCD_{ff}$. A redução do GA para $UniqueGA_{ff}$ é definida de forma similar a que foi dada por Köbler, Schöning e Torán [22], que apresentou uma redução de tempo polinomial do GA para $UniqueGA$. O algoritmo de tempo polinomial deles para GA faz consultas a um dado oráculo que representa corretamente $UniqueGA_{ff}$ nas entradas dadas. Este algoritmo funciona corretamente por que todas as consultas feitas pelo algoritmo satisfazem a premissa do $UniqueGA$, isto é, toda consulta é um grafo com um número par de vertices, com um único automorfismo não trivial, sem nenhum ponto fixado, ou não há nenhum automorfismo não trivial de qualquer modo. Por uma leve modificação da redução deles, pode-se obter uma redução de GA para $UniqueGA_{ff}$. Além disso, isto também é possível fazendo o parâmetro de tamanho n satisfazer a equação específica $n = 2(2n' + 1)$, onde $n' \in \mathbb{N}$.

Lema 3. (Algoritmo de Conversão). Existe um algoritmo quântico de tempo polinomial que, com certeza, converte $\rho_\pi^+(n)$ em $\rho_\pi^-(n)$ e mantém $\iota(n)$ como é para qualquer parâmetro $n \in N$ e qualquer permutação $\pi \in \mathcal{K}_n$.

Prova Dado um $n \in N$ arbitrário. Primeiro lembremos da definição de $\mathrm{sign}(\sigma)$: $\mathrm{sign}(\sigma) = 0$ se σ é par e $\mathrm{sign}(\sigma)$: $\mathrm{sign}(\sigma) = 1$ se σ é ímpar. Seja $\pi \in \mathcal{K}_n$ uma permutação oculta qualquer e ρ_π^+ seu estado quântico correspondente. Sobre uma entrada ρ_π^+, nosso algoritmo quântico desejado apenas inverte a fase de acordo com o sinal da permutação. Isto é feito efetuando a seguinte transformação:

$$|\sigma\rangle + |\sigma\pi\rangle \longmapsto (-1)^{\mathrm{sign}(\sigma)}|\sigma\rangle + (-1)^{\mathrm{sign}(\sigma\pi)}|\sigma\pi\rangle$$

Note-se que a determinação do sinal de uma dada permutação leva apenas tempo polinomial em n. Como π é ímpar, $\mathrm{sgn}(\sigma)$ e $\mathrm{sgn}(\sigma\pi)$ são diferentes; assim, o algoritmo acima obviamente converte ρ_π^+ em ρ_π^-. Além disso, o algoritmo não altera o estado quântico ι.

Teorema 4. *Seja k um polinômio. Se existe um algoritmo quântico $\mathcal{A}$ tal que*

$$\left|\Pr_{\mathcal{A}}\left[\mathcal{A}(\rho_\pi^+(n)^{\otimes k(n)}) = 1\right] - \Pr_{\mathcal{A}}\left[\mathcal{A}(\rho_\pi^-(n)^{\otimes k(n)}) = 1\right]\right| > \varepsilon(n)$$

Para qualquer parâmetro de segurança $n \in N$, então existe um algoritmo quântico $\mathcal{B}$ tal que, para cada $n \in N$,

$$\left|\Pr_{\mathcal{B}}\left[\mathcal{B}(\rho_\pi^+(n)^{\otimes k(n)}) = 1\right] - \Pr_{\mathcal{B}}\left[\mathcal{B}(\iota(n)^{\otimes k(n)}) = 1\right]\right| > \frac{\varepsilon(n)}{4}$$

Prova do teorema 4

Fixado um $n \in N$ arbitrariamente, e que daqui por diante omitiremos esse parâmetro n. Assumimos que o algoritmo quântico $\mathcal{A}$ distingue entre $\rho_\pi^{+\otimes k}$ e $\rho_\pi^{-\otimes k}$ com vantagem de pelo menos $\varepsilon(n)$. Seja $\mathcal{A}'$ o algoritmo que aplica o algoritmo de conversão do Lema 3 a um dado estado χ (que é ou $\rho_\pi^{+\otimes k}$, ou $\iota^{-\otimes k}$) e em seguida alimenta o estado resultante χ' ($\rho_\pi^{+\otimes k}$, ou $\iota^{-\otimes k}$) para $\mathcal{A}$. Disto resulta que $\mathcal{A}'(\rho_\pi^{+\otimes k}) = \mathcal{A}(\rho_\pi^{-\otimes k})$ e $\mathcal{A}'\iota^{\otimes k}) = \mathcal{A}(\iota^{\otimes k})$. Pela desigualdade triangular, temos

$$\left|\Pr_{\mathcal{A}}\left[\mathcal{A}(\rho_\pi^{+\otimes k}) = 1\right] - \Pr_{\mathcal{A}}\left[\mathcal{A}(\iota^{\otimes k}) = 1\right]\right| + \left|\Pr_{\mathcal{A}'}\left[\mathcal{A}'(\rho_\pi^{+\otimes k}) = 1\right] - \Pr_{\mathcal{A}'}\left[\mathcal{A}'(\iota^{\otimes k}) = 1\right]\right| > \varepsilon(n)$$

para qualquer parâmetro $n \in N$. Esta desigualdade nos leva ou a

$$\left|\Pr_{\mathcal{A}}\left[\mathcal{A}(\rho_\pi^{+\otimes k}) = 1\right] - \Pr_{\mathcal{A}}\left[\mathcal{A}(\iota^{\otimes k}) = 1\right]\right| > \frac{\varepsilon(n)}{2}$$

ou

$$\left|\Pr_{\mathcal{A}'}\left[\mathcal{A}'(\rho_\pi^{+\otimes k}) = 1\right] - \Pr_{\mathcal{A}'}\left[\mathcal{A}'(\iota^{\otimes k}) = 1\right]\right| > \frac{\varepsilon(n)}{2}$$

Para completar a prova, foi projetado o algoritmo $\mathcal{B}$ da seguinte maneira: primeiro escolhe-se $\mathcal{A}$ ou $\mathcal{A}'$ ao acaso e depois é simulado o algoritmo escolhido. É fácil verificar que $\mathcal{B}$ distingue entre $\rho_\pi^{+\otimes k}$ e $\iota^{\otimes k}$ com vantagem de pelo menos $\varepsilon(n)/4$.