APLICAÇÃO DE PROTOCOLOS QUÂNTICOS E ALGORITMO DE SHOR PARA A SEGURANÇA DA INFORMAÇÃO

Mariana Godoy Vazquez Miano[1]

DOI: 10.47283/244670492020080154

Resumo

O objetivo deste artigo é mostrar, de modo objetivo, em especial aos estudantes de Segurança da Informação, algumas formas práticas de se operar com a Criptografia Quântica. Toda a abordagem foi realizada sobre o tripé da Computação Quântica, que é a Matemática, a Física Quântica e a Computação, com vistas às questões atuais e tangíveis relacionadas à segurança da informação. A primeira abordagem é relacionada à geração das chaves quânticas e o uso de protocolos quânticos (BB84 e B92). Em relação ao uso de algoritmos, o Algoritmo Quântico de Shor é implementado na linguagem de programação JavaScript, com seu código parcialmente comentado para a ilustração do funcionamento do algoritmo.

Palavra-chave: Protocolos quânticos. Criptografia quântica. Algoritmo de Shor. JavaScript.

Abstract

The purpose of this article is to show, in an objective way, especially to Information Security students, some practical ways of operating with Quantum Cryptography. The whole approach was carried out on the tripod of Quantum Computing, which is Mathematics, Quantum Physics and Computation, with a view to current and tangible issues related to information security. The first approach is related to the generation of quantum keys and the use of quantum protocols (BB84 and B92). Regarding the use of algorithms, the Shor Quantum Algorithm is implemented in the JavaScript programming language, with its code partially commented to illustrate the operation of the algorithm.

Keywords: *Quantum protocols. Quantum cryptography. Shor's algorithm. Javascript.*

Introdução

A Computação Quântica é um novo paradigma da Computação. Uma nova área de pesquisa inter e multidisciplinar (GRILO, 2014) que necessita, para seu pleno desenvolvimento, o aporte da Matemática, da Física e da Computação.

Segundo (GRILO, 2014), alguns resultados da Teoria de Computação Quântica possuem uma relação estreita com conceitos da Física. O estudo de tais problemas a partir de uma perspectiva diferente pode permitir o melhor entendimento de fundamentos da Mecânica Quântica. Além disso, tem-se que o estudo da Computação Quântica sob as lentes da Teoria da Computação pode trazer novos resultados para a própria Teoria de Computação clássica. O estudo de um problema sob o ponto de vista de um novo modelo computacional pode trazer novas ideias para solucioná-lo sem tais recursos. Encontramos hoje algoritmos clássicos inspirados em algoritmos quânticos, além de provas de teoremas envolvendo somente elementos clássicos, utilizando argumentos inspirados no modelo quântico.

Em um canal clássico de comunicação, por mais segurança que se tenha, ele poderia sofrer espionagem de um agente externo, sem que seja percebido, uma vez que a informação clássica pode ser clonada (RIGOLIN & RIEZNIK, 2005).

[1] Faculdade Tecnologia de Americana. E-mail: vazquez.prof@gmail.com

A segurança da criptografia atual, em especial a criptografia assimétrica, baseia-se na dificuldade de se solucionar alguns problemas matemáticos. As soluções conhecidas para estes problemas têm complexidade não polinomial: apesar de serem, em teoria, solucionáveis, quando se utiliza chaves com tamanho adequado, o tempo previsto de solução ultrapassa as centenas de anos, tornando ataques brutos impraticáveis. Entretanto, a computação quântica permite que estes problemas sejam resolvidos em pouco tempo (chegando à ordem dos segundos), pois várias soluções podem ser testadas ao mesmo tempo, de forma análoga a uma computação paralela, mas com apenas um processador (GRILO, 2014).

De acordo com (NIELSEN &CHUANG, 2005), os computadores quânticos vieram para resolver problemas que são impossíveis de serem resolvidos em computadores clássicos, não porque sejam insolúveis, mas sim pela grande quantidade de recursos necessários para a sua solução.

Atualmente são conhecidas duas classes de algoritmos quânticos, com atuações específicas de resolução de problemas. A primeira classe é baseada na Transformada de Fourier Quântica (TFQ) de Shor e inclui algoritmos notáveis para resolver os problemas de fatoração e de logaritmos discretos, com ganho exponencial de velocidade sobre os melhores algoritmos clássicos conhecidos. A segunda classe é baseada em Algoritmos de Grover para a realização de busca quântica, que oferece um ganho quadrático de tempo sobre os equivalentes clássicos. A importância desses algoritmos está no amplo uso das técnicas de busca por meio de algoritmos clássicos, que em muitos casos podem ser adaptadas para o caso quântico obtendo-se algoritmos mais rápidos.

Em relação à primeira classe, baseada na Transformada de Fourier Quântica (usada para resolver problemas de logaritmo discreto e fatoração), seus resultados permitem que um computador quântico decifre muitos códigos de vários sistemas criptográficos em uso, incluindo o sistema RSA.

No quesito segurança, se a comunicação for feita através de um canal quântico, haverá a certeza de que a transmissão foi realizada com segurança total e que a informação não foi clonada (RIGOLIN & RIEZNIK, 2005), segundo o Teorema da não-Clonagem (NIELSEN & CHUANG, 2005, p.576), uma das bases da Teoria da Informação Quântica.

Nesse artigo serão abordadas a utilização de protocolos quânticos com a distribuição de chaves quânticas e a aplicação do Algoritmo de Shor, implementado na linguagem JavaScript, utilizando bibliotecas quânticas disponíveis.

1 Criptografia

1.1 Criptografia RSA

Criptografia baseada na geração de números aleatórios grandes e primos e na função totiente de Euler (operação com números primos entre si). Pelo fato das chaves serem geradas com números muito grandes e primos, leva muito tempo para fatorar na computação clássica. É nesse ponto que o algoritmo de Shor apresenta riscos para a segurança da informação, uma vez que desacelera o tempo exponencial de fatoração com a atual tecnologia para um tempo polinomial, tornando possível decifrar uma mensagem sem possuir a chave privada (STALLINGS, 2008).

1.2 Criptografia Quântica

A criptografia quântica surgiu para resolver o problema da distribuição de chaves da Criptografia Clássica, através dos postulados da Mecânica Quântica, em especial o Princípio da Incerteza de Heisenberg (MARASCIULO, 2020).

1.2.1 Distribuição de chave quântica

A distribuição de chave quântica (DCQ) é um protocolo provadamente seguro, por meio do qual os bits de uma chave privada podem ser criados por dois parceiros usando um canal público. Os bits da chave podem ser usados para implementar um sistema criptográfico de chave privada (CHAVES, 2018). O único requisito para o protocolo DCQ é que os qubits possam ser comunicados pelo canal público com uma taxa de erro menor do que um certo limiar. A segurança da chave resultante é garantida pelas propriedades da informação quântica e portanto condicionada somente às leis da Física. Assim, um espião não poderia obter qualquer informação dos qubits transmitidos de A para B sem perturbar o estado compartilhado por eles. Primeiramente, pelo Teorema da Não-Clonagem (NIELSEN e CHUANG, 2005, p.576), o espião não poderia clonar o qubit enviado por A e na sequência, temos a seguinte proposição (ganho de informação implica perturbação):

P.1 – Em qualquer tentativa de se distinguirem estados quânticos não ortogonais, a informação obtida só é possível através da introdução de perturbação no sinal.

Dem.: Sejam $|\psi>$ e $|\varphi>$ estados quânticos não-ortogonais, dos quais um espião tenta obter informação. Pode-se supor, sem perda de generalidade, que o processo usado para obter informação é interagir de forma unitária o estado $|\psi>$ ou $|\varphi>$ com um sistema auxiliar preparado em um estado-padrão $|u>$. Supondo que esse processo não perturba os estados, nos dois casos se obtém

$$|\psi>|u> \rightarrow |\psi>|v> \quad e \quad |\varphi>|u> \rightarrow |\varphi>|v'>$$

Se os estados $|v>$ e $|v'>$ fossem diferentes, o espião poderia obter informação sobre a identidade do estado. Porém, como o produto interno é preservado sob transformações unitárias, devemos ter

$$<v|v'><\psi|\varphi> = <u|u><\psi|\varphi>$$

$$<v|v'> = <u|u> = 1$$

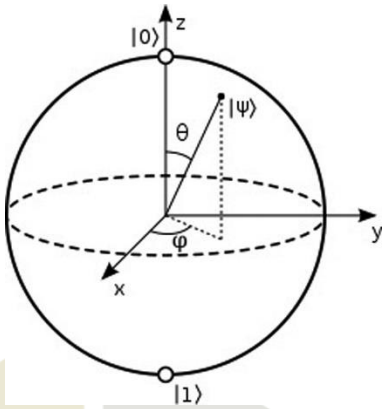
o que implica que $|v>$ e $|v'>$ devem ser idênticos. Logo, a diferença entre $|\psi>$ e $|\varphi>$ perturbará ao menos um dos estados. Esse conceito é usado na transmissão dos qubits em estados não-ortogonais entre A e B. Verificando as perturbações nos estados transmitidos, eles estabelecem um limite superior para o ruído ou espionagem no canal de comunicação. Esses qubits de verificação são aleatoriamente intercalados com os qubits de dados (dos quais os bits a chave serão extraídos posteriormente), de modo que o limite superior também se aplique aos qubits de dados. A e B implementam a restituição da informação da informação e amplificação de privacidade para gerar uma sequência comum para a chave secreta. O limiar para o erro máximo tolerável é então determinado pela eficácia do protocolo utilizado nessa tarefa.

1.3 Protocolos Quânticos

1.3.1 Protocolo BB84

O protocolo BB84 utiliza os estados de polarização dos fótons para a transmissão de chaves criptográficas. Consideremos aqui A e B como os dois elementos comunicantes e a possibilidade de um espião interceptar a comunicação. Consideremos ainda a Esfera de Bloch (CARVALHO et al, 2007), espaço matemático onde ocorre a movimentação dos fótons (ou qubits), conforme figura 1:

Figura 1: Esfera de Bloch



Fonte: (CARVALHO et al, 2007)

A esfera inicia com duas sequências, a e b, cada uma com $(4 + \delta)n$ bits clássicos aleatórios. Ela então codifica essas sequências como um bloco de $(4 + \delta)n$ qubits,

$$|\psi> = \otimes_{k=1}^{(4+\delta)n} |\psi_{a_k b_k}>$$

Em que a_k é o k-ésimo bit de a (analogamente para b), e cada qubit está em um dos quatro estados:

$|\psi_{00}> = |0>$

$|\psi_{10}> = |1>$

$|\psi_{01}> = |+> = (|0> + |1>)/\sqrt{2}$

$|\psi_{11}> = |-> = (|0> - |1>)/\sqrt{2}$

O efeito desse procedimento é codificar a na base X ou Z, como determinado por b (RIGOLION e RIEZNIK, 2005). Observa-se que os quatro estados não são mutuamente ortogonais e, portanto, nenhuma medida pode distinguir um do outro com certeza. A então envia $|\psi>$ para B através do canal quântico público de comunicação.

B recebe $\epsilon(|\psi><\psi|)$, em que ϵ descreve a operação quântica devido ao efeito combinado da ação do ruído e do espião. Ele torna o fato público. Nesse ponto, A, B e o espião têm seus próprios estados descritos por diferentes matrizes densidade. Nesse ponto, como A

não revelou b, o espião não sabe em que base ela deveria ter medido para espionar a comunicação; o melhor que ela pode fazer é tentar adivinhar, mas se ela falhar o estado recebido por B terá sido perturbado. Além disso, enquanto o ruído ϵ pode ser em parte devido ao ambiente (um canal ruim) e em parte à ação do espião, isso não o ajuda a ter o controle do canal, e pode-se considerar que o espião é inteiramente responsável por ϵ .

Obviamente, para B, $\epsilon(|\psi><\psi|)$ também não trará muita informação, uma vez que ele ainda não conhece b. Entretanto, mesmo assim, ele mede cada qubit na base X ou Z, como determinado por uma sequência aleatória de $(4+\delta)n$ bits b' que ele mesmo cria. Seja a' o resultado da medida de B. Na sequência, A anuncia b publicamente e, através de um canal público, B e A descartam todos os bits em {a, a'}, exceto aqueles para os quais b' e b são iguais. Os bits que restarem satisfarão a' = a, pois esses bits foram medidos na mesma base em que A os preparou. Vale ressaltar que b não revela nada sobre a, ou sobre os bits a' que resultaram da medida de B, mas é importante que A não publique b até que B anuncie ter recebido os seus qubits. Para simplificação, supõe-se que A e B mantenham 2n bits dos seus resultados; δ pode ser escolhido suficientemente grande para que isso seja feito com probabilidade exponencialmente alta.

Agora, A e B realizam alguns testes para determinar quanto ruído ou espionagem ocorreu durante a comunicação. A seleciona n bits (dos 2n) aleatoriamente e anuncia publicamente a seleção. B e A publicam e comparam os valores desses bits de verificação. Se houver discordância em mais de t bits, eles abortam a operação e recomeçam o protocolo. T é selecionado de tal forma que, se o teste passar, eles podem aplicar os algoritmos de reconciliação de informação e a amplificação de privacidade para obter m chaves secretas compartilhadas aceitáveis nos n bits restantes.

1.3.2 O protocolo BB84 para DCQ

1. A escolhe aleatoriamente $(4+\delta)n$ bits de dados.
2. A escolhe uma sequência aleatória b de $(4+\delta)n$ bits. Ela codifica cada bit como {|0>, |1>} se o bit correspondente de b for 0 ou {|+ >, |- >} se b for 1.
3. A envia o estado resultante para B.
4. B recebe os $(4+\delta)n$ bits, anuncia o fato e mede cada qubit na base X ou Z aleatoriamente.
5. A anuncia b.
6. A e B descartam quaisquer bits em que B tenha medido em uma base diferente da que foi usada por A na preparação. Com alta probabilidade, 2n bits restarão (caso contrário, o protocolo é abortado). Eles mantêm os 2n bits.
7. A seleciona um subconjunto de n bits que servirá para verificação da interferência do espião e diz para B quais bits ela selecionou.
8. A e B anunciam e comparam os valores de n bits verificadores. Se a discordância for maior do que um número aceitável, eles abortam o protocolo.
9. A e B implementam a reconciliação de informação e amplificação da privacidade nos n bits restantes, para obter uma chave com m bits compartilhados.

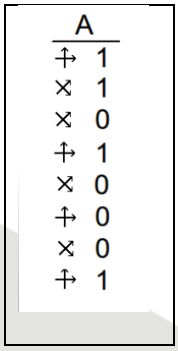
Assim, pode-se dizer que esse protocolo resolve o problema da distribuição de chaves com um algoritmo *one-time pad*, extremamente seguro. A aplicação desse protocolo é dividido em duas etapas: 1ª. Etapa – um canal de comunicação quântico; 2ª. Etapa – um canal de comunicação clássico.

Exemplo de aplicação

1ª. Etapa:

A usa fótons polarizados, que podem ser medidos em 3 bases: retilínea (vertical ou horizontal), circular (à esquerda ou à direita) e diagonal (45 ou 135 graus); A modula aleatoriamente os fótons para um de 4 estados: horizontal, vertical, circular à esquerda ou circular à direita (figura 2); B escolhe aleatoriamente qual polarização usar para medir cada fóton recebido (retilínea ou circular) (MARQUEZINO e HELALEY, 2003).

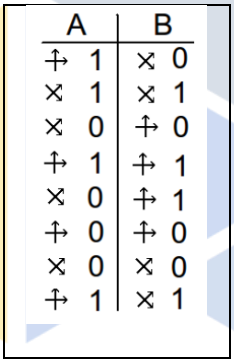
Figura 2: Modulação aleatória dos fótons



Fonte: Marquezino e Helaley, 2003.

Assim, B tenta detectar os bit´s sem saber a base. Se a base fora mesma, o bit é lido corretamente, senão, há 50% de chance de acerto (figura 3).

Figura 3 : Medição de B



Fonte: Marquezino e Helaley, 2003.

2ª. Etapa:

A e B se comunicam através de um canal público; B envia para A a sequência de polarizações utilizada; A verifica quais medidas estão corretas; A e B verificam nas suas sequências quais as corretas.

A 2ª. Etapa é identificar a ocorrência de espionagem através da comparação dos valores observados. Se houver alguma incoerência (polarização correta e valor incorreto), é indicação de espionagem. Se não houver incoerência, a chave é segura. Quando as bases forem diferentes, o resultado é descartado (figura 4).

Figura 4: Resultado da aplicação do protocolo BB84

A	B	A	B	chave
+ 1	× 0	+	×	
× 1	× 1	×	×	1
× 0	+ 0	×	+	
+ 1	+ 1	+	+	1
× 0	+ 1	×	+	
+ 0	+ 0	+	+	0
× 0	× 0	×	×	0
+ 1	× 1	+	×	

Fonte: Marquezino e Helaley, 2003.

1.3.2 Protocolo B92

O protocolo B92 é a generalização do BB84, com a utilização de outros estados e base, utilizando apenas dois estados quânticos não-ortogonais (RIGOLIN e RIEZNIK, 2005). Por simplicidade, será suficiente considerar o que acontece com um único bit de cada vez; a descrição pode facilmente ser generalizada para testes por blocos, como no caso do BB84.

Suponha que A prepara um bit clássico aleatório a e, dependendo do resultado, envia para B o qubit

$$|\psi> = |0> \ se\ a = 0 \text{ ou } |\psi> = \frac{|0> + |1>}{\sqrt{2}} \text{ se a} = 1$$

Dependendo do bit clássico aleatório que ela gera, a', B realiza uma medida no qubit que ele recebe de A ou na base Z, |0> e |1> (se a' = 0), ou na base de X, $|\pm> = (|0> \pm|1>)/\sqrt{2}$ (se a' = 1). Dessa medida, ele obtém o resultado b, que será 0 ou 1, correspondendo aos auto-estados -1 e +1 de X e Z. B então anuncia publicamente b (mas mantem a' em segredo), e A e B comunicam-se publicamente, mantendo apenas os pares {a, a'}, para os quais b =1. Observa-se que quando a = a', tem-se sempre b = 0. Somente se a' = 1-a é que B obterá b=1, o que ocorre com probabilidade ½. A chave final será a para A e 1-a' para B.

Esse protocolo B92 destaca como a impossibilidade de se distinguirem perfeitamente dois estados não-ortonormais está no centro da criptografia quântica. Como no BB84, é impossível que um espião distinga os estados de A sem perturbar a correlação entre seus bits e os de B. Portanto, o protocolo permite que A e B criem bits de chave compartilhados e estabeleçam um limite superior para o ruído e tentativas de invasão durante a comunicação. Eles aplicam a reconciliação de informação e amplificação de privacidade para extrair bits secretos das sequências aleatórias de bits correlacionadas.

Exemplo de aplicação

Inicialmente, A e B geram duas sequência aleatórias; onde o resultado for "0", A envia "vertical" e onde for "1" envia diagonal = 45º ; Na sequência, B procura detectar "0" para diagonal – 45º e "1" para horizontal. Para finalizar, B verifica em quais posições o bit foi detectado, conforme a figura 5.

Figura 5: Resultado da aplicação do protocolo B92.

A	B	A	B	resultado	chave
0	1	↕	↔	não detecta	
0	1	↕	↔	não detecta	
1	1	↗	↔	não detecta	
0	0	↕	↘	detecta	0
1	1	↗	↔	detecta	1
1	0	↗	↘	não detecta	
0	0	↕	↘	não detecta	
1	0	↗	↘	não detecta	

Fonte: Marquezino e Helaley, 2003.

2 Transformada de Fourier quântica

Segundo (Nielsen & Chuang, 2005), uma das formas de resolver problemas matemáticos complexos é transformá-los em outros problemas em que a resolução seja conhecida. Um desses métodos é a Transformada de Fourier. Fourier descobriu que todo sinal pode ser descrito como uma superposição de senóides completas, ou seja, é possível escrevê-lo como uma soma de outros sinais que possuem frequência. Essa função é muito importante na computação atual e o exemplo mais usual é o reconhecimento de fala no qual é usado esse algoritmo em sons digitalizados.

De maneira resumida, a Transformada de Fourier leva um vetor x de N dimensões denotado por (x_0, x_1, x_2, … ,x_{N-1}) para o vetor y denotado por (y_0, y_1, y_2,..., y_{N-1}), no qual cada yk é obtido por

$$yk \equiv \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{2\pi ijk/N}$$

e sua inversa

$$xj \equiv \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} j_k e^{-2\pi ijk/N} .$$

De forma semelhante, o mesmo método pode ser aplicado na mecânica e computação quântica (LULA JR. et al, 2005, p. 18), assim a transformada de Fourier Quântica leva uma base ortonormal | k > para | j > onde | j > é obtido por

$$F|j> \equiv \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi ijk/N} |k>$$

e sua inversa

$$F^{-1}|K> \equiv \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{-2\pi ijk/N} |j>$$

Para entender como é a implementação deste algoritmo, é preciso conhecer os cálculos que lhe deram origem. A Transformada de Fourier mapeia funções do domínio do tempo com período r, para o domínio da frequência, para múltiplos$2\pi/r$. Uma forma específica desse cálculo é a Transformada Discreta de Fourier em q pontos tem o mesmo espaço no intervalo. Desse último, surgiram dois algoritmos. Em ambas qé um múltiplo de 2, a Transformada

Rápida de Fourier é usada em computadores clássicos e a Transformada de Fourier Quântica (QFT) é para computadores quânticos (construída em forma de matrizes de transformações unitárias). A QFT altera as amplitudes do estado quântico, ou seja, altera as probabilidades de medir certo estado.
Obs.: em um estado $|\psi> = \alpha|0> + \beta|1>$ α e β são amplitudes dos estados, ou seja, a possibilidades de medir $|0>$ ou $|1>$.

A QFT atua pode ser representada da seguinte maneira:

$$\sum_{x} \quad g(x)|x> \rightarrow \sum_{c} \quad G(c)|c>$$

Em que:

- G(c) é a função resultante da aplicação QFT na função periódica $g(x)$
- x e q são representações binárias entre 0 e $q-1$
- a probabilidade de observar o estado $|c>$ é $||G(c)||^2$
- $G(c) \neq 0$ somente nos múltiplos de q/r, em q é a quantidade de intervalo e r o período.
- O resultado é um múltiplo de q/r denotado por jq/r

Para se obter uma aproximação é necessário que o período r seja múltiplo de dois, assim como q. Assim, a Transformada Quântica de Fourier pode ser representada por:

$$U_{QFT}: |x> \rightarrow \frac{1}{2^m} \sum_{c=0}^{2^m-1} \quad c^{\frac{2\pi cx}{2m}}|c>$$

A transformada de Fourier Quântica utiliza menos operações que a função original e serve de base para construção de outros algoritmos quânticos tais como a estimativa de fase, fatoração e de busca (NIELSEN; CHUANG, 2005).

3 Algoritmo de fatoração de Shor

O algoritmo de Shor é considerado o *breakthrough* da computação moderna, uma vez que demonstrou efetivamente a importância da computação quântica, pois resolveu um problema matemático que era estudado há anos e em tempo polinomial. O problema da fatoração de grandes números é considerado tão difícil de resolver em um computador clássico que foi implementado no sistema de criptografia RSA (STALLINGS, 2008).

Atualmente, é considerado um dos melhores algoritmos para fatoração. Na versão quântica, é capaz de fatorar números de altas ordens em segundos (LOMONACO, 2002). Essa capacidade pode ser usada para "quebrar" algoritmos de encriptação atuais, como o RSA. O objetivo do algoritmo é achar o período de uma função, e na sequência, encontrar os fatores do valor solicitado.

O algoritmo de Shor é um algoritmo quântico que, dado um inteiro n composto ímpar que não é potência de primo, devolve um fator de n com probabilidade limitada de erro (BERNSTEIN, 1993). As restrições para o valor de n não representam problema algum. De fato, é trivial encontrar um fator de um número par. Além disso, é fácil desenvolver um algoritmo eficiente que decide se $n = a^k$, para inteiros a e $k > 1$, e que devolve a e k neste caso (MARTINS, 2018).

Como n é produto de no máximo log n inteiros, o algoritmo de Shor pode ser utilizado para resolver o problema da fatoração em tempo polinomial no tamanho da entrada. O

algoritmo de Shor baseia-se numa redução do problema da busca de um fator de n ao problema da busca do período de uma sequência. Como a redução utiliza aleatorização é possível que ela falhe, ou seja, que nenhum fator de n seja encontrado. Porém, a probabilidade de ocorrência deste evento é limitada.

3.1 Algoritmo de Shor (n)

1 escolha um inteiro $1 < x < n$ aleatoriamente
2 se mdc(x, n) > 1
3 então devolva mdc(x, n)
4 seja r o período da função $f(a) = x^a \bmod n$
5 se r for ímpar ou $x^{r/2} \equiv -1 \pmod n$
6 então o procedimento falhou
7 devolva mdc($x^{r/2} + 1$, n)

O algoritmo de Shor utiliza um único passo quântico: o cálculo do período da função na linha 4. Os demais passos podem ser efetuados em tempo polinomial no modelo clássico e também no modelo quântico.

4 Implementação em Javascript

JavaScript é uma Linguagem de Programação criada por Brendan Eich, solicitado pela empresa Netscape, em 1995, com o nome Live Script. Tanto o JScript (da Microsoft) quando o JavaScript (Netscape) só podiam ser usados nos navegadores das empresas que os desenvolveram. A empresa ECMA padronizou a linguagem, para que pudesse ser utilizada em vários navegadores. Essa linguagem padronizada foi então denominada ECMAscript. Apesar da mudança da nomenclatura da linguagem, o novo nome foi praticamente ignorado e os desenvolvedores continuam usando a denominação de Javascript (PACIEVITCH, 2020).

O grande diferencial do JavaScript é que esta permite o desenvolvimento dos códigos dentro do código HTML. Para o desenvolvedor, seu uso é muito simples: é só adicionar o código " <script>" e iniciar a programação em Javascript. Também é permitido a utilização de códigos em HTML dentro do código de JavaScript. Assim, é uma linguagem de simples utilização e com muitas bibliotecas dicponíveis.

Para a implementação do algoritmo quântico de Shor, utilizou-se a biblioteca Jsqubits, da linguagem de programação JavaScript (CROCKFORD, 2001). O algoritmo em JavaScript foi desenvolvido com node.js e JavaScript es6. O script é capaz de calcular os fatores comum do número desejado. Todavia, quanto maior a multiplicação entre dois fatores sendo um deles primo, maior o tempo até chegar ao resultado.

Na sequência, destacam-se alguns trechos do código comentado, para a ilustração do funcionamento do Algoritmo de Shor.

1.
```
//Nesta condição, as chances de obter uma resposta obviamente erradas são reduzidas a partir
//da verificação se o fator comum aleatório obtido a partir da função que gera um numero primo
//está no intervalo > 1 e < que o número que queremos descobrir seus fatores.
if (candidateDivisor > 1 && candidateDivisor <= outputRange) {
  if (f(candidateDivisor) === f0) {
    console.log('This is a multiple of the rank.');
    bestSoFar = candidateDivisor;
  } else {
    var lcm = jsqubitsmath.lcm(candidateDivisor, bestSoFar);
    if (lcm <= outputRange) {
      console.log('This is a good candidate.');
      bestSoFar = lcm;
    }
```

2.
```
//Essa condição retorna imediatamente o fator comum '2' caso o número colocado seja par
if (n % 2 === 0) {
  // Is even.  No need for any quantum computing!
  return 2;
}
```

3.
```
//Nessa condição, é verificado se a função powerfactor retorna um número > 1, caso sim
// dentro da função do powerfactor ele verifica se o n (número fatorado) possui
// é um powerfactor. O powerfactor nada mais é que um primo multiplicado por ele mesmo X vezes resultado em n.
var powerFactor = jsqubitsmath.powerFactor(n);
if (powerFactor > 1) {
  // Is a power factor.  No need for anything quantum!
  return powerFactor;
}
```

4.
```
//essa parte do código captura o horário em que dá o 'start' armazenando o horario atual em uma
//variavel para achar os fatores comuns e, ao final, quando o resultado é exibido,
//chama-se uma função que retorna o horario atual e depois substitui pelo valor da variavel
//que armazena o horario em que foi iniciado a busca/1000, retornando em segundos
//o tempo levado pela a operação.
var startTime = new Date();
factor(n, function(result) {
    log("One of the factors of " + n + " is " + result);
    log("Time taken in seconds: " + ((new Date().getTime()) - startTime.getTime()) / 1000);
});
```

A função powerfactor (em 3) é usada na verificação do fator “candidato”. A função fica em *looping* até chegar ao valor desejado. Sabe-se que 11 é um “power factor” de 121, pois os fatores de 121 são iguais (11x11). Assim, o exemplo é validado.

Conclusão

O objetivo deste artigo era mostrar algumas formas de se operar com a Criptografia Quântica, para estudantes da área de Segurança da Informação. Em especial, com o uso de protocolos quânticos (BB84 e B92) e o Algoritmo de Shor. A abordagem foi realizada sobre o tripé da Computação Quântica, que é a Matemática, a Física Quântica e a Computação.

Através da apresentação dos protocolos quânticos BB84 e B92 mostrou-se como garantir a segurança do uso de um canal de comunicação, uma vez que através desses protocolos, tem-se a certeza se a informação sofreu ou não espionagem, e de que não foi clonada (pois isso não é possível no “mundo quântico”).

A implementação do Algoritmo de fatoração de Shor, através da linguagem Javascript (escolhida pela facilidade de implementação, adaptabilidade e por possuir bibliotecas quânticas disponíveis) apenas como ilustração do Algoritmo, mostra o lado “prático” e “tangível” da aplicação dos conceitos quânticos atualmente, ainda que não tenhamos computadores quânticos pessoais disponíveis. A evolução de conceitos e novas práticas computacionais quânticas mostram-se de grande interesse e preocupação mundiais, principalmente das grandes empresas do segmento computacional (IBM, Google, D-Wave, etc) e instituições financeiras.

Referências

BERNSTEIN, E., VAZIRANI U. **Proceedings of the 25th annual ACM symposium on the theory of computing,** ACM, New York 11 (1993).

CARVALHO, L. M., LAVOR, C., MOTTA, V. S. **Caracterização Matemática e Visualização da Esfera de Bloch: Ferramentas para Computação Quântica**. TEMA Tend. Mat. Apl. Comput., 8, n. 3, p. 351 – 360 (2007).

CHAVES, R. O. G. **Táticas de ataque a protocolos quânticos de distribuição de Chaves criptográficas utilizando estratégias de discriminação de estados.** Dissertação (mestrado) – Universidade Federal de Minas Gerais – Departamento de Física. 2018

CROCKFORD, D. ***JavaScript: The World's Most Misunderstood Programming Language***. Disponível em http://www.crockford.com/javascript/javascript.html. Acesso em: 06 de nov. de 2019.

GRILO, A. B. (2014). **Computação Quântica e Teoria da Computação**. Dissertação (mestrado) - UNICAMP - Instituto de Computação.

LOMONACO,S. J. **Shor's quantum factoring algorithm**. Proceedings of Symposia in Applied Mathematics, vol. 58, 2002, pp. 161–180.

LULA JR., B., ALBERT, B. B.; ASSIS, F. M. **Transformadas Quânticas de Fourier e de Hartley**. VII Congresso Brasileiro de Redes Neurais. 2005.

MARASCIULO, M. **Princípio da incerteza: 5 pontos para entender a teoria de Werner Heisenberg.** Disponível em https://revistagalileu.globo.com/Ciencia/noticia/2020/02/principio-da-incerteza-5-pontos-para-entender-teoria-de-werner-heisenberg.html. Acesso em: 02 de fev. de 2020.

MARQUEZINO, F. L., HELALEY-NETO. J. A. **Estudo Introdutório do Protocolo Quântico BB84 para troca segura de chaves**. Centro Brasileiro de Pesquisas Físicas. Série Monografias (2003).

MARTINS, R. C. **O Algoritmo de Fatoração de Shor**. Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Matemática. 2018

NIELSEN. M. A., CHUANG, I. L. **Computação Quântica e Informação Quântica**. Porto Alegre: Bookman, 2005.

PACIEVITCH, Y. ***JavaScript*** . Disponível em https://www.infoescola.com/informatica/javascript-2/. Acesso em: 02 de fev. de 2020.

RIGOLIN, G., RIEZNIK, A. A. Introdução à Criptografia Quântica. **Revista Brasileira de Ensino de Física**, v.27, n.4, p. 517 – 526, 2005.

STALLINGS, W. **Criptografia e segurança de redes – Princípios e práticas**. 4. ed. São Paulo: 2008. Pearson Pratice Hall, 2008.