UM ESTUDO EXPLORATÓRIO DAS FALHAS DE SEGURANÇA DE ALGORITMOS DE CRIPTOGRAFIA NA ERA DA COMPUTAÇÃO QUÂNTICA

Rodrigo Pigatto Pasquale (IC) e Calebe de Paula Bianchini (Orientador)

Apoio:PIBIC Mackenzie

RESUMO

A segurança de dados é fundamental para a execução de diversas tarefas do cotidiano, seja acessar contas bancárias até o funcionamento de carros autônomos. Algoritmos de criptografia, existente desde o início dos tempos em diferentes formas, foram criados para garantir a integridade destes dados. Um dos algoritmos de segurança mais utilizados, o Rivest-Shamir-Adleman (RSA) tornou importante a fatoração, para saber qual o estado da arte da criptografia e saber quanto tempo levará para fatorar números muito grandes, que é o que torna o RSA seguro atualmente. Há diversas implementações de algoritmos de fatoração, porém, nenhum apresenta uma eficiência adequada para quebrar a criptografia RSA. No entanto, com o avanço da computação quântica, como uma necessidade física resultante do encolhimento dos chips de processamento dos computadores, a segurança da informação é colocada em cheque com o Algoritmo de Shor, desenvolvido por Peter Shor na década de 1990, que é mais eficiente que os algoritmos clássicos de fatoração, e pode ser capaz de quebrar a criptografia RSA em tempo hábil no futuro. O objetivo do presente artigo é apresentar uma análise de diferentes algoritmos de fatoração clássicos e compará-los com o Algoritmo de Shor para mostrar as falhas de segurança que existirão na era da computação quântica.

Palavras-chave: Computação Quântica; Criptografia; Segurança da Informação;

ABSTRACT

Data security is essential for the execution of several daily tasks, whether it is accessing bank accounts to the functioning of autonomous cars. Encryption algorithms, which exists since the beginning of time in different forms, were created to guarantee the integrity of this data. One of the most widely used security algorithms, Rivest-Shamir-Adleman (RSA), made factorization important, knowing the state of art of cryptography and how long it will take to factor very large numbers, which is what makes RSA secure today. There are several implementations of factoring algorithms, but none have an adequate efficiency to break RSA encryption. However, with the advance of quantum computing, as a physical necessity resulting from the shrinking of computer processing chips, information security is challenged with the Shor's Algorithm, developed by Peter Shor in the 1990s, which is more efficient than classic factoring algorithms, and may be able to break RSA encryption in a timely manner in the future. This article presents an analysis of different classical factorization algorithms and compare them with the Shor's Algorithm to show the security flaws that will exist in the quantum computing era.

Keywords: Quantum Computing; Cryptography; Information Security;

1. INTRODUÇÃO

Em 1936, Alan Turing desenvolveu o primeiro modelo computacional: a Máquina de Turing – e mostrou que este modelo, com uma fita infinitamente longa e dividida em quadrados de mesmo tamanho pode escrever, ler e alterar símbolos binários. Em essência, a partir de um dado algoritmo, uma máquina é capaz de computá-lo e produzir algum resultado. Após a publicação do trabalho de Turing, os primeiros computadores construídos com componentes eletrônicos surgiram, principalmente em 1947 com o desenvolvimento do transistor. (CHUANG, 2010)

A partir da década de 1960, Gordon E. Moore descreveu a Lei de Moore, de forma empírica: em um período de aproximadamente dois anos, o número de transistores em um chip de silício dobraria. De maneira menos formal, é possível afirmar que: a velocidade dos processadores dobraria nesse período. Durante cinco décadas, pode-se dizer que Moore esteve correto. No entanto, o encolhimento destes circuitos a cada nova geração, causaria um problema de natureza física: à medida que o tamanho dos transistores se aproxima do nível atômico, efeitos quânticos passam a interferir no funcionamento destes dispositivos eletrônicos. (CHUANG, 2010)

Quando os chips de silício atingirem seu tamanho mínimo, os computadores clássicos estarão em um possível limite de processamento incapaz, ainda, de resolver algumas questões, como da área de química molecular e da ciência dos materiais.

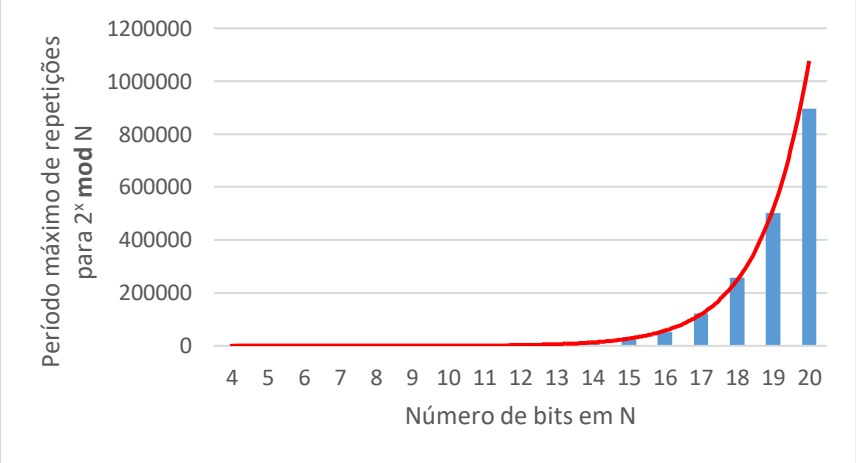
Uma possível solução para os problemas de natureza física dos transistores é mudar o paradigma computacional, e um desses novos paradigmas é a Computação Quântica. Problemas complexos para computadores clássicos, como a fatoração, poderão ser resolvidos facilmente em computadores quânticos. A fatoração, em especial, é parte essencial de alguns dos algoritmos de criptografia mais utilizados na atualidade: o criptossistema RSA. De forma simples, sempre houve uma necessidade, na história humana, de proteger alguma informação (RIGOLIN; RIEZNIK, 2005). Os diversos algoritmos de criptografia, ao longo dos tempos, foram desenvolvidos a partir desta premissa. Hackers que têm por objetivo gravar e forjar informações através de computadores têm suas tarefas dificultadas por estes algoritmos de segurança que, atualmente, dependem de força bruta para serem quebrados. No futuro, com acesso à computadores quânticos, a segurança da comunicação online, como de transações bancárias e até mesmo para carros autônomos, estará em risco. O criptossistema Rivest-Shamir-Adleman (RSA), mesmo com implementações de grandes números é um deles.

O algoritmo RSA é uma implementação de um sistema de criptografia de chave pública em que cada mensagem de um emissor é criptografada com a chave pública do receptor.

Para cada indivíduo, há uma chave pública e disponível, de forma que todos têm a disponibilidade de enviar e receber dados protegidos. Para o conteúdo da mensagem ser lido pelo receptor, ele deve utilizar-se de sua chave privada – que apenas ele tem acesso. As chaves são criadas com base na multiplicação de grandes números primos, ou seja, é correto dizer que o sistema Rivest-Shamir-Adleman é baseado na diferença entre a facilidade de encontrar números primos e na dificuldade de fatorar o produto de dois primos grandes. (CORMEN et al., 2009)

O Algoritmo de Shor, descoberto por Peter Shor em 1994, permite fatorar grandes números de forma rápida através de uma Transformada de Fourier Quântica (QFT), uma variação da Fast Fourier Transform (FFT), para encontrar o período do número que deseja fatorar. No caso da criptografia RSA, este número pode ser uma chave privada. A Figura 2 apresenta um gráfico que demonstra o crescimento exponencial de operações necessárias para um algoritmo de força bruta clássico fatorar um número N.

Figura 1 - Número de repetições necessárias para quebrar a criptografia RSA com um algoritmo de força bruta, em uma CPU clássica.



Fonte: (JOHNSTON, 2019)

Apesar do tempo que os computadores quânticos poderão quebrar esses sistemas ainda estar distante, alternativas, como algoritmos pós-quânticos – uma área de estudo recente da matemática – se tornam extremamente importantes para o futuro da segurança da comunicação.

2. DESENVOLVIMENTO DO ARGUMENTO

2.1 Sistemas de criptografia de chave pública e o criptossistema RSA

Um sistema de criptografia de chave pública permite que mensagens enviadas entre dois participantes de uma comunicação sejam criptografadas de modo que um intruso que as escute não possa decifrá-las. Este sistema permite que cada mensagem tenha uma

assinatura digital única dos participantes, como uma assinatura manuscrita em um papel, e que não pode ser forjada.

Para um sistema de criptografia de chave pública funcionar, cada participante tem uma chave pública e uma chave secreta. Cada chave representa uma informação, no criptossistema RSA, sendo que cada chave consiste em um par de inteiros. (CORMEN et al., 2009). Cada participante cria suas próprias chaves, as secretas são mantidas em segredo enquanto as públicas podem ser divulgadas publicamente. Para um participante A, a função da chave pública é chamada de $\mathrm{P_A}(\,)$ e a da chave secreta, $\mathrm{S_A}(\,)$ (ROSEN, 2009).

Em uma mensagem M, as chaves públicas e secreta formam funções inversas uma da outra, ou seja, $\mathrm{M} = \mathrm{S_A}\big(\mathrm{P_A}(\mathrm{M})\big) = \mathrm{P_A}\big(\mathrm{S_A}(\mathrm{M})\big)$. Para esta mensagem M ser enviada, deve-se seguir o seguinte cenário:

i) Um participante B deseja enviar uma mensagem para o participante A. Para isso, obtém a chave pública de A, P_A;
ii) A mensagem M é criptografada pela função $M' = P_A(M)$ e M' é enviada ao A;
iii) Para o participante A ter acesso à mensagem M, ele aplicar a função da chave secreta em M', ou seja: $M = S_A(M')$

Na década de 1970, Ronald Rivest, Adi Shamir e Leonard Adleman desenvolveram o criptossistema RSA, de chave pública, que é baseado na potenciação modular de dois números primos grandes (ROSEN, 2009). Para as mensagens serem codificadas pelo método RSA, elas devem ser transcritas em sequências de números inteiros, podendo ser feito transcrevendo cada letra em algum número inteiro a partir de uma tabela de conversão. A codificação, então, é feita transformando o número inteiro que representa a mensagem, em outro número inteiro, que representa o texto codificado.

O algoritmo de criptografia de chave pública RSA segue os seguintes passos:

i) Escolha dois números primos grandes p e q, tais que $\mathrm{p} \neq \mathrm{q}$;
ii) Calcule $\mathrm{n} = \mathrm{p} \times \mathrm{q}$, obtenha $\phi(\mathrm{n})$ através da relação $\phi(\mathrm{n}) = (\mathrm{p}-1) \times (\mathrm{q}-1)$;
iii) Escolha um inteiro e, tal que $1 < \mathrm{e} < \phi(\mathrm{n})$;
iv) Calcule d através da relação $\mathrm{d} \times \mathrm{e} = 1 \bmod \phi(\mathrm{n})$;

Seguindo o algoritmo, obtêm-se o par de chave pública RSA, $\mathrm{P}(\mathrm{e},\mathrm{n})$ e o par de chave secreta RSA, $\mathrm{S}(\mathrm{d},\mathrm{n})$. (CORMEN et al, 2009)

Uma mensagem C criptografada pelo RSA a partir de um par de chaves $\mathrm{P(e, n)}$ e $\mathrm{S(d, n)}$ é:

$$C = M^e \, \boldsymbol{mod} \, n$$

Para obter a mensagem original M a partir de C, basta resolver a seguinte relação:

$$M = C^d \boldsymbol{mod} \, n$$

2.2 Aritmética de números inteiros

O Teorema Fundamental da Aritmética diz que todo número inteiro maior que zero pode ser escrito como o produto (ou fatores) de números primos. Um número primo P somente é primo se for divisível por 1 e P, caso contrário é um número composto. (ROSEN, 2009) Como exemplo, o número 20 pode ser escrito da seguinte forma: $\mathbf{20 = 2 \times 2 \times 5 = 2^2 \times 5^1}$, ou seja, é um número composto. Enquanto o número 31 só pode ser escrito como ele mesmo multiplicado por 1, pois é primo.

Os procedimentos para verificar se um número é primo não são eficientes, mesmo com os métodos de fatoração mais poderosos conhecidos, como o Teste de Primalidade de Miller-Rabin, que é um algoritmo probabilístico que determina se um número $\mathbf{n}$ é composto ou primo. Ele é eficiente, por conta de sua natureza probabilística, rodando em $\mathbf{O(k \times log_3 n)}$. No entanto, há uma chance de mostrar um falso positivo – apontando que um número é primo quando ele não é. (CORMEN et al, 2009)

Apesar de úteis, os testes de primalidade só são capazes de determinar se um número é primo ou composto. Para quebrar criptografias, apenas saber se o número é primo não basta, é necessário encontrar fatores de algum número inteiro $\mathbf{N}$. Os métodos mais conhecidos para fatorar números inteiros o fazem em tempo exponencial e variam em complexidade de implementação. O mais simples, é o de divisões sucessivas até $\sqrt{\boldsymbol{n}}$. Há soluções heurísticas para o problema de fatoração, como o Algoritmo Pollard-Rho e métodos mais complexos baseados em sieving, como o *Quadratic Sieve* (QS) e o *General Number Field Sieve* (GNFS).

Sistemas de congruências lineares aparecem em diversos contextos, como no algoritmo de Potenciação Modular. Para a solução destes sistemas, o Teorema do Resto Chinês foi proposto afirmando que: quando os módulos de um sistema de congruência linear são pares de primos entre si, há uma única solução do sistema módulo, o produto dos módulos. (ROSEN, 2009)

Considere os números inteiros positivos de pares primos entre si m_i, com $i = [1 \dots n]$, e a_i números inteiros arbitrários. O sistema de congruências lineares $x \equiv a_i (\boldsymbol{mod}\ m_i)$ tem a solução $m = \prod_{i=1}^{n} m_i$. Desta forma, pode-se representar um elemento de $\mathbb{Z}_{pq}$, inteiros que são congruentes a módulo $p \times q$, como um número de $\mathbb{Z}_p$ e outro de $\mathbb{Z}_q$, ou seja, uma bijeção desses conjuntos. (CORMEN et al, 2009)

Pelo Teorema do Resto Chinês, um número inteiro a pode ser representando pela n-upla que consiste dos restos da divisão por m_i, ou seja, a pode ser representando por: $(a\ \boldsymbol{mod}\ m_1, a\ \boldsymbol{mod}\ m_2, \dots, a\ \boldsymbol{mod}\ m_n)$ (ROSEN, 2009). Para $m = 35$, por exemplo, o número 17 pode ser escrito como $(2,3)$, pois $35 = 7 \times 5$ e $2 \equiv 17\ \boldsymbol{mod}\ 7$ e $3 \equiv 17\ \boldsymbol{mod}\ 5$.

Uma aplicação prática do Teorema do Resto Chinês é computar multiplicações de números grandes através de números menores, pela representação em módulo. Isto pode ser especialmente útil para gerar a assinatura RSA.

Para a criptografia RSA, é importante, também, encontrar $\mathrm{M^e mod\ n}$ de maneira eficiente. Para calcular esta relação, um algoritmo, chamado de Potenciação Modular, foi desenvolvido. Para encontrar $\mathrm{M^e}$, é necessário utilizar-se de uma expansão de e na base M que é expressada da seguinte forma: $\mathrm{e} = \mathrm{a_k M^k} + \mathrm{a_{k-1} M^{k-1}} + \cdots + \mathrm{a_1 M^1} + \mathrm{a_0}$. Depois, multiplica-se os termos $\mathrm{M^{2j}}$ quando $\mathrm{a_j} = 1$, $\mathrm{a_j}$ é o j-ésimo bit na expansão de e na base M.

A Potenciação Modular usa $\mathrm{O((\log m)^2 \log n)}$ operações para encontrar o resultado (ROSEN, 2009). Este algoritmo é descrito pelos seguintes passos:

i) Determine M, e e n;
ii) Verifique se $n = 1$, caso seja, atribua 1 a uma variável x, caso contrário:
iii) Calcule a expressão $x = (x \times M)\boldsymbol{mod}\ e$, para uma variável i de 0 até e, passo 1
iv) Retorne x

O Pequeno Teorema de Fermat diz que se p é um número primo e k um número inteiro qualquer, então $k^p = k\ \boldsymbol{mod}\ p$, se k não for divisível por p, então $k^{p-1} = 1\ \boldsymbol{mod}\ p$. (ROSEN, 2009)

Uma generalização do Pequeno Teorema de Fermat é a função de Euler ($\varphi(n)$), que é definida como o número de inteiros positivos menores que n que são co-primos de n, um exemplo é que todos os números inteiros menores que um primo p são co-primos de p, então $\varphi(p) = p - 1$ (CHUANG, 2010). Os únicos números inteiros que não são co-primos de p são aqueles que são múltiplos dele.

Pelo Teorema Chinês do Resto, se dois inteiros a e b são co-primos, podemos deduzir que $\varphi(ab) = \varphi(a)\varphi(b)$.

2.3 Computação quântica

A mecânica clássica é determinística. As equações de movimento de algum sistema podem ser resolvidas de forma que se obtém a posição e o momento de uma partícula, para qualquer valor de tempo (EISBERG; RESNICK, 1979). No mundo macroscópico, diversos movimentos, como de objetos astronômicos, puderam ser previstos a partir de suas equações. No mundo microscópico, no entanto, a física clássica é limitada pelo princípio da incerteza.

"O princípio da incerteza de Heisenberg afirma que uma experiência não pode determinar simultaneamente o valor exato de uma componente do momento, por exemplo p_x, de uma partícula e o valor exato da coordenada correspondente, x." (EISBERG; RESNICK, 1979)

O ato de observar um sistema gera uma perturbação que não é completamente previsível, alterando-o, fazendo com que se torne desconhecido. Desta forma, não podemos determinar precisamente o comportamento dele e, em vez disso, podemos apenas mostrar os possíveis resultados de uma observação, sua ocorrência, que então, se torna probabilística.

Uma superposição de ondas pode ser uma combinação ou destruição delas. Para duas partículas clássicas de matéria esse fenômeno somente ocorre de uma forma: a combinação delas. A superposição de estados é típica de sistemas quânticos - formadas por fótons, elétrons ou moléculas – que se comportam, concomitantemente, como partícula e onda. (PIVETTA, 2012)

De maneira análoga aos bits, a unidade fundamental da computação clássica, na computação quântica há os bits quânticos, ou *qubits* - que são objetos matemáticos com propriedades específicas (CHUANG, 2010). Os bits quânticos podem ser o spin de um elétron ou um fóton, por exemplo. Enquanto os bits podem assumir apenas dois valores, 0 ou 1, os *qubits* assumem uma superposição, que pode ser representada através de uma combinação linear, de estados entre 0 e 1:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

α e β são números complexos.

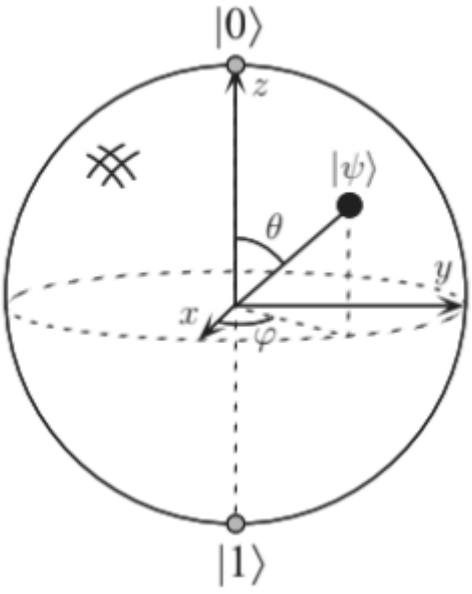
Quando medimos os *qubits*, não podemos determinar, com precisão, em qual estado ele está - recebemos apenas 0 com probabilidade $|\alpha|^2$ ou 1 com probabilidade $|\beta|^2$, $|\alpha|^2 +$

$|\beta|^2 = 1$, ou seja, a sua interpretação é que o vetor está normalizado com comprimento 1 e o estado de um qubit é um vetor unidimensional em um espaço vetorial bidimensional complexo. (CHUANG, 2010)

As superposições de estados fazem parte do funcionamento de *qubits*. Eles podem estar em qualquer valor entre |0⟩ e |1⟩, porém, quando forem medidos, retorna apenas um dos dois valores: |0⟩ ou |1⟩. Desta maneira, enquanto em um conjunto de quatro bits clássicos pode estar em uma configuração de 2^4 por vez – 16 combinações possíveis, das quais somente uma pode ser utilizada – quatro *qubits* em superposição podem estar todos nessas 16 combinações de uma vez e esse número cresce de forma exponencial para cada qubit.

Os *qubits* podem ser representados com a Esfera de Bloch, uma esfera com raio de um e um ponto na sua superfície representa o estado do *qubit*, descrito através de ângulos.

Figura 2 - uma esfera de Bloch representando um qubit.



Fonte: (CHUANG,2010)

Circuitos montados na computação clássica são compostos, em sua maioria, de fios e portas lógicas. Na computação quântica não é diferente: os fios podem não ser, necessariamente, fios físicos, mas fótons movendo de um lugar para outro ou serem correspondentes à passagem do tempo (CHUANG, 2010). Há, também, o equivalente às portas lógicas clássicas que são as portas quânticas.

Uma operação que pode ser obtida com apenas um *qubit* é o análogo a porta NOT da computação clássica: em que os estados são invertidos ($0 \rightarrow 1$ e $1 \rightarrow 0$) – em um sistema microscópico, ela altera os estados de forma linear:

$$\alpha|0\rangle + \beta|1\rangle \rightarrow \alpha|1\rangle + \beta|0\rangle$$

A representação matricial para a porta NOT quântica, X, é:

$$X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

O estado quântico $\alpha|0\rangle + \beta|1\rangle$ pode ser representado como o vetor $\begin{bmatrix} \alpha \\ \beta \end{bmatrix}$.

Aplicando X neste estado, temos:

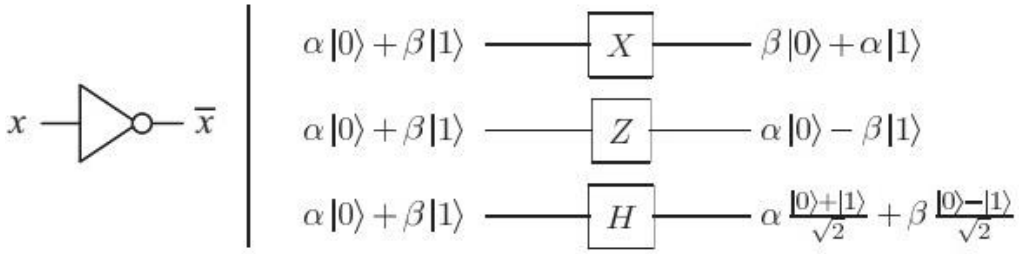
$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} \times \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix}$$

Outras duas portas quânticas importantes são a porta Z e a porta de Hadamard (H):

$$Z \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad H \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

A primeira altera o sinal de $|1\rangle$ para $-|1\rangle$, enquanto a de Hadamard é uma operação que rotaciona no eixo $\hat{y}$ em 90°, seguida por outra rotação no eixo $\hat{x}$ em 180°. Não há equivalente à essas duas nos circuitos lógicos clássicos. (CHUANG, 2010)

Figura 3 - Porta lógica com um único bit (à esquerda) e portas lógicas com um único qubit (à direita).



Fonte: (CHUANG, 2010)

Por conta das propriedades da Física Quântica, o mundo macroscópico pode ser descrito através de princípios quânticos, desta forma, circuitos lógicos clássicos podem ser simulados em circuitos quânticos.

2.4 Algoritmo de Shor

Na Ciência da Computação e na Matemática, uma das maneiras de se resolver um problema difícil é transformá-lo em um problema com uma solução conhecida. Para o problema da fatoração, a solução é transformá-lo em um problema de encontrar o período de uma função através da Transformada Quântica de Fourier.

Seja polinômio $P(x) = \sum_{j=0}^{n-1} a_i x^j$ de grau n $\omega_n^0, \omega_n^1, \dots, \omega_n^{n-1}$ em que $\omega_n = e^{2\pi i/n}$ é uma raiz n-ésima complexa da unidade. O polinômio P pode ser representado como um vetor dos coeficientes $a = (a_0, a_1 \dots, a_{n-1})$. Uma Transformada Discreta de Fourier (DFT – *Discrete Fourier Transform*) pode ser definida por:$DFT_n(P(x)) = \sum_{j=0}^{n-1} a_j \omega_n^{kj}$ para $k = [0 \dots n-1]$. A DFT atua no vetor de coeficiente a e retorna um outro vetor como solução (CORMEN et al, 2009). Em essência, a DFT funciona em amostras discretas de algum sinal, seja uma onda sonora ou a representação digital de uma imagem. Cada amostra é armazenada como um número em ponto flutuante complexo. A Transformada Discreta de Fourier pode informar a frequência, a proporção e os offsets de uma componente senoidal.

Um método que se aproveita das propriedades das raízes complexas da unidade é o da Transformada Rápida de Fourier (FFT – *Fast Fourier Transform*). A FFT utiliza da divisão e conquista para calcular a DFT em tempo hábil, o algoritmo separa os coeficientes de índice par dos de índice ímpar e define dois novos polinômios de grau $n/2$. Assim, $P(x) = P^{[0]}(x^2) + xP^1(x^2)$, ou seja, o problema é reduzido em uma combinação linear dos pontos $(\omega_n^0)^2, (\omega_n^1)^2, \dots, (\omega_n^{n-1})^2$ dos polinômios $P^{[0]}$ e $P^{[1]}$ de grau $n/2$ nos pontos, esse subproblema, apesar de ser exatamente a mesma forma do original, tem metade do tamanho. (CORMEN et al, 2009).

Figura 4 - o algoritmo recursivo da FFT.

```
RECURSIVE-FFT(a)
 1  n ← length[a]                ▷ n is a power of 2.
 2  if n = 1
 3     then return a
 4  ω_n ← e^{2πi/n}
 5  ω ← 1
 6  a^[0] ← (a_0, a_2, ..., a_{n-2})
 7  a^[1] ← (a_1, a_3, ..., a_{n-1})
 8  y^[0] ← RECURSIVE-FFT(a^[0])
 9  y^[1] ← RECURSIVE-FFT(a^[1])
10  for k ← 0 to n/2 − 1
11        do y_k ← y_k^[0] + ω y_k^[1]
12           y_{k+(n/2)} ← y_k^[0] − ω y_k^[1]
13           ω ← ω ω_n
14  return y                     ▷ y is assumed to be a column vector.
```

Fonte: (CORMEN et al, 2009)

O que permite o Algoritmo de Shor ser mais rápido que os algoritmos clássicos de fatoração é uma implementação da DFT em um registrador quântico, chamada de

Transformada Quântica de Fourier (QFT – Quantum Fourier Transform). A partir da transformada discreta $\frac{1}{\sqrt{n}}\sum_{j=0}^{n-1} a_j e^{2\pi ijk/n}$, em que $\frac{1}{\sqrt{n}}$ é o fator normalizante da DFT. A QFT faz exatamente a mesma transformação que a DFT, no entanto, é em uma base ortonormal $|0\rangle, \dots, |n-1\rangle$, ou seja, o estado quântico de um qubit. Ela pode ser definida como: $\sum_{j=0}^{n-1} a_j|j\rangle \rightarrow \sum_{k=0}^{n-1} y_k|k\rangle$. Uma QFT pode ser utilizada no processo de phase estimation, que é estimar o autovalor de um autovetor de um operador unário. Seja U um operador unário de m qubits com um autovetor $|\psi\rangle$, tal que $U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle$ para $0 \leq \theta < 1$. Encontrar o autovalor de $|\psi\rangle$ é equivalente a estimar a fase de θ com um nível finito de precisão. (CHUANG, 2010) Na Figura 9, o algoritmo *phase estimation* é descrito.

Figura 5 - algoritmo de phase estimation.

Algorithm: Quantum phase estimation

Inputs: (1) A black box wich performs a controlled-U^j operation, for integer j, (2) an eigenstate $|u\rangle$ of U with eigenvalue $e^{2\pi i\varphi_u}$, and (3) $t = n + \lceil \log\left(2 + \frac{1}{2\epsilon}\right)\rceil$ qubits initialized to $|0\rangle$.

Outputs: An n-bit approximation $\widetilde{\varphi_u}$ to φ_u.

Runtime: $O(t^2)$ operations and one call to controlled-U^j black box. Succeeds with probability at least $1 - \epsilon$.

Procedure:

1.	$\|0\rangle\|u\rangle$	initial state
2.	$\rightarrow \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} \|j\rangle\|u\rangle$	create superposition
3.	$\rightarrow \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} \|j\rangle U^j\|u\rangle$	apply black box
	$= \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} e^{2\pi ij\varphi_u}\|j\rangle\|u\rangle$	result of black box
4.	$\rightarrow \|\widetilde{\varphi_u}\rangle\|u\rangle$	apply inverse Fourier transform
5.	$\rightarrow \widetilde{\varphi_u}$	measure first register

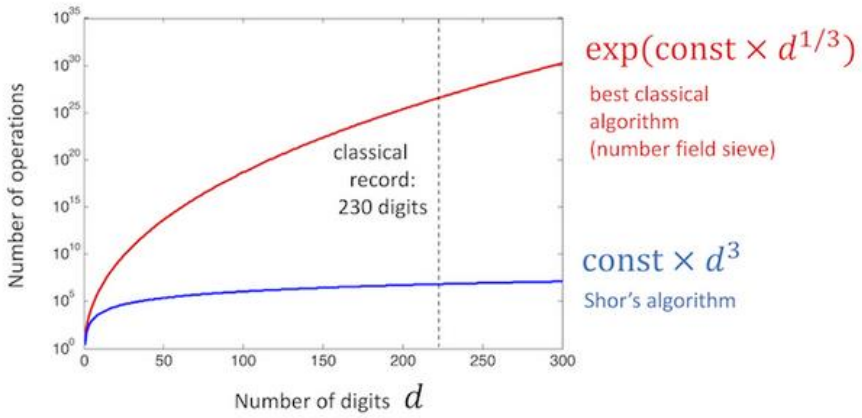
Fonte: (CHUANG, 2010)

Uma aplicação do algoritmo *phase estimation* é resolver o problema de encontrar o período de uma função e o de fatorar números inteiros – os dois problemas são equivalentes. (CHUANG, 2010) Para encontar o período de uma função, basta aplicar o algoritmo de *phase estimation* no operador unário: $U|y\rangle \equiv |xy(\boldsymbol{mod}\ n)\rangle$.

O que trouxe visibilidade para o paradigma da computação quântica foi o algoritmo desenvolvido por Peter Shor no final da década de 1990 (SHOR, 1997). Até então, os algoritmos clássicos que foram criados para resolver o problema da fatoração tinham tempo de execução com crescimento exponencial, mesmo o mais eficiente deles, o *General Number*

Field Sieve, já abordado neste artigo, não era suficiente para abalar a segurança do algoritmo RSA.

Figura 6 - Number Field Sieve em comparação com o algoritmo de Shor.



Disponível em: https://quantum-computing.ibm.com/docs/guide/q-algos/shor-s-algorithm

Em 1997, Peter Shor publicou o artigo intitulado *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer* em que apresentou uma solução para o problema da fatoração de números primos em tempo polinomial em $\log n$, dependente, apenas, de quando os computadores quânticos passariam a existir (SHOR,1997).

O Algoritmo de Shor tem uma parte clássica e uma parte quântica para sua execução. Os computadores clássicos são responsáveis em transformar o problema de fatoração em um problema de encontrar o período de uma função (CHUANG, 2010). Em essência, serve de preparação para a parte quântica, em que se constrói os estados quânticos necessários e é executada a QFT para encontrar o período da função. Por fim, o resultado obtido pela QFT, pode ser utilizado com o algoritmo de Euclides e encontrar o resultado buscado.

Para reduzir um problema de fatoração em um de encontrar o período de uma função, é necessário computar um fator de N se for possível encontrar uma solução não trivial para a equação $x^2 = 1 \, \boldsymbol{mod} \, N$ e mostrar que um inteiro y co-primo de N tem probabilidade de que seu período r ser par tal que $x \equiv y^{\frac{r}{2}} \, \boldsymbol{mod} \, N$ seja uma solução de $x^2 = 1 \, \boldsymbol{mod} \, N$. A função periódica que é utilizada para o Algoritmo de Shor é $f(x) = a^x \boldsymbol{mod} \, n$.

Dado um inteiro $n = p \times q$, o algoritmo de Shor pode ser descrito pela seguinte sequência de passos (SMOLING, SMITH; VARGO, 2013):

i) Escolha, aleatoriamente, um inteiro a, em que $0 < a < n$;

ii) Calcule $mdc(a,n)$ pelo Algoritmo de Euclides. Se for diferente de 1, então $mdc(a,n)$ é um fator não trivial de n. Caso contrário, prossiga para o próximo passo.

iii) Escolha $S \equiv 2^S$ tal que $n^2 \leq S \leq 2n^2$. Construa o estado quântico $S^{-1/2} \sum_{x=0}^{S-1} |x\rangle |0\rangle$ usando dois registradores quânticos, o primeiro tem s qubits e o segundo, $\log n$ qubits.

iv) Mapeie $|x\rangle |0\rangle$ para $|x\rangle |a^x \boldsymbol{mod}\ n\rangle$

v) Faça uma Transformada de Fourier Quântica (QFT) no primeiro registrador para obter o seguinte estado: $S^{-1} \sum_x \sum_y e^{\left(\frac{2\pi i}{S}\right)xy} |y\rangle |a^x \boldsymbol{mod}\ n\rangle$

vi) Meça o primeiro registrador e obtenha o resultado clássico y. Com uma probabilidade razoável, a aproximação, por uma fração contínua, de S/y ou de algum S/y' para algum y' próximo de y será um múltiplo inteiro do período r da função $f_a(x) = a^x \boldsymbol{mod}\ n$. O algoritmo de Euclides pode, então, encontrar eficientemente r.

vii) Se r for ímpar, ou se $a^{r/2} = -1\ \boldsymbol{mod}\ n$, volte para o passo i). Caso contrário, $mdc\left(a^{\frac{r}{2}} \pm 1, n\right)$ é p ou q.

3. CONSIDERAÇÕES FINAIS

A Computação Quântica é um paradigma fundamentalmente diferente da computação clássica e que oferece uma melhora substancial para a execução de alguns problemas, sejam eles problemas de otimização, descoberta de novo fármacos ou a análise de criptografia. O avanço que poderá ser obtido nas mais diversas áreas é notável, principalmente aquelas que necessitam de alto poder de processamento, ou que tem a capacidade de serem paralelizadas. As áreas de ciência de materiais, química, medicina e até mesmo, dentro da própria Computação, a Inteligência Artificial, serão beneficiadas.

O alto poder de processamento dos Computadores Quânticos coloca os algoritmos de segurança mais comumente utilizados em risco de serem quebrados em tempo hábil, quando comparado com os supercomputadores mais poderosos disponíveis na atualidade – uma tarefa computacionalmente difícil, que pode levar ser atingida na Supremacia Quântica, conceito que é definido quando um computador quântico conseguir performar uma tarefa em tempo muito superior ao de um computador clássico.

A fragilidade dos algoritmos de segurança, como o RSA ou o ECC[1] é por conta de serem baseados em problemas matematicamente difíceis de serem resolvidos, a fatoração

[1] Um outro tipo de criptografia de chave pública é a Criptografia de Curva Elíptica (ECC, em inglês). Algoritmos baseados em curvas elípticas para a criptografia de dados existem desde a década de 1980. Em termos de segurança da informação, a criptografia de curvas elípticas pode ser mais eficiente que a RSA. Um procolo que é comumente utilizado e que pode ser quebrado por um computador quântico é o ECDH (Elliptic Curve Diffie-Hellman).

de números inteiros grandes e os logaritmos discretos, respectivamente. O Algoritmo de Shor consegue resolver estes dois problemas em tempo polinomial. Dados confidenciais, sensíveis e pessoais estarão expostos à essa fragilidade a partir do momento em que um computador quântico esteja apto a realizar estas tarefas. É uma corrida contra o tempo para uma solução pós quântica de criptografia, que já tem resultados promissores.

A implementação do RSA não é complexa e o que está por trás da garantia de sua segurança é a dificuldade em fatorar números inteiros muito grandes. Da mesma maneira, algoritmos de fatoração, com exceção dos baseados em sieving, não têm implementação difícil e a melhora do mais simples, a fatoração por divisões sucessivas, a ser implementado para o mais complexo, como o GNFS, é grande mas não tão significativa – eles ainda se mantém em tempo exponencial. Na Figura 7 há uma implementação do algoritmo de divisões sucessivas em C++.

Figura 7 - algoritmo de divisões sucessivas implementado em C++.

```
int divisoes_sucessivas(int N) {
    int d = 3;
    if (N % 2 == 0) return 2;
    while (d <= sqrt(N)) {
        if (N % d == 0) {
            return d;
        }
        d = d + 2;
    }
    return N;
}
```

Fonte: Autoria própria.

De fato, para números muito grandes, acima de 100 dígitos, a diferença entre o algoritmo de divisões sucessivas torna-se notável. Apesar disso, o General Number Field Sieve ainda é incapaz de fatorar em tempo hábil estes números.

É uma questão de tempo e de engenharia que um computador quântico passará a ser capaz de quebrar os criptossistemas mais comumente utilizados. A criptografia é essencial para proteger a informação que trafega pela internet, desde realizar transações online até ler mensagens em algum aplicativo. Como abordado neste artigo, os sistemas de criptografia de chave pública são baseados na dificuldade de fatorar números inteiros. A Figura 8 mostra os sistemas de criptografia que são ameaçados pelo algoritmo de Shor.

Figura 8 - sistemas criptográficos ameaçados pelo algoritmo de Shor.

Public-key cryptography			
RSA-3072[1]	Encryption	128	Broken (Shor)
RSA-3072[1]	Signature	128	Broken (Shor)
DH-3072[42]	Key exchange	128	Broken (Shor)
DSA-3072[63,64]	Signature	128	Broken (Shor)
256-bit ECDH[4–6]	Key exchange	128	Broken (Shor)
256-bit ECDSA[66,67]	Signature	128	Broken (Shor)

Fonte: (BERNSTEIN; LANGE, 2017)

Os novos criptossistemas deverão ser integrados com protocolos já existentes, como o Transport Layer Security (TLS). Para o desenvolvimento de um criptossistema novo, precisarão ser considerados: o tamanho das chaves de criptografia e assinaturas; o tempo para decifrar a chave em cada ponta de um canal de comunicação e a quantidade de dados necessária para criptografar ou decifrar alguma mensagem. O National Institute of Standards and Technology (NIST) e a Microsoft tem um projeto com o nome de Post-Quantum Cryptography. Há quatro protocolos sendo desenvolvido, cada um baseado em um problema matematicamente difícil de ser resolvido e que poderão ser utilizados para diferentes cenários. Eles são: FrodoKEM[2], SIKE[3], Picnic[4], qTESLA[5]. O primeiro e o qTesla, são baseados no problema de learning with errors, que é baseado em reticulados de subgrupos algebricamente não estruturados. O SIKE, baseia-se no protocolo Supersingular Isogeny Diffie-Hellman (SIDH) – ele faz uso de operações aritméticas em curvas elípticas definidas em corpos finitos e computa os chamados isogenies, a dificuldade de encontrar um isogeny específico é o que garante a segurança deste algoritmo.

O protocolo Picnic tem uma abordagem diferente e não se baseia em problemas difíceis da Teoria dos Números. Ele utiliza de um método interativo chamado de provas de conhecimento zero em que um participante A convence um participante B que sabe um segredo, sem revelá-lo, junto disso funções hash e block ciphers (que são seguros contra algoritmos quânticos) são utilizados para melhorar a segurança do protocolo.(CHEN et al, 2016)

Apesar de existirem diversos modelos propostos para computadores quânticos, não há, ainda, uma tecnologia com aceitação unânime para criar qubits: há implementações com o uso de supercondutores (como o Tangle Lake da Intel) e outras com técnicas de ion

[2] Disponível em: https://frodokem.org/
[3] Disponível em: https://sike.org/
[4] Disponível em: https://www.microsoft.com/en-us/research/project/picnic/
[5] Disponível em: https://qtesla.org/

trapping[6]. Por enquanto, o paradigma da computação quântica ainda está distante por conta de sua construção complexa que se deve a perturbações que podem existir durante a operação de um qubit, como ruído, alteração de temperatura, variações de tensão, dentre outras perturbações. No entanto, a mudança para o paradigma quântico será essencial para a resolução de problemas que, atualmente, não podem ser resolvidos em tempo hábil pelos computadores clássicos.

4. REFERÊNCIAS

BERNSTEIN, Daniel J.; LANGE, Tanja. Post-quantum cryptography. **Nature**, [s.l.], v. 549, n. 7671, p. 188-194, set. 2017. Springer Science and Business Media LLC. http://dx.doi.org/10.1038/nature23461.

CHEN, Lily; JORDAN, Stephen; LIU, Yi-kai; MOODY, Dustin; PERALTA, Rene; PERLNER, Ray; SMITH-TONE, Daniel. Report on Post-Quantum Cryptography. National Institute Of Standards And Technology Internal Report, [s.l.], v. [], n. 2105, p. 1-15, abr. 2016. National Institute of Standards and Technology. http://dx.doi.org/10.6028/nist.ir.8105.

CORMEN, Thomas H. et al. Algoritmos: teoria e prática. Teoria e Prática. 3. ed. São Paulo: Elsevier, 2009.

EISBERG, Robert; RESNICK, Robert. Física Quântica: Átomos, Moléculas, Sólidos, Núcleos e Partículas. [s.i]: Campus, 1979. 928 p.

JOHNSTON, Eric R.; HARRIGAN, Nic; GIMENO-SEGOVIA, Mercedes. Programming Quantum Computers: essential algorithms and code samples. Sebastopol: O'reilly, 2019. 317 p.

NIELSEN, Michael A.; CHUANG, Isaac L.. Quantum Computation and Quantum Information. 10. ed. New York: Cambridge University Press, 2010. 704 p.

PIVETTA, Marcos. A nova onda dos qubits: Sistemas à temperatura ambiente podem ter uma porção quântica útil para a computação. Fapesp, [s.l], v. 193, n., p.53-57, mar. 2012.

RIGOLIN, Gustavo; RIEZNIK, Andrés Anibal. Introdução à criptografia quântica. Revista Brasileira de Ensino de Física, [s.l.], v. 27, n. 4, p.517-526, dez. 2005. FapUNIFESP (SciELO). http://dx.doi.org/10.1590/s1806-11172005000400004. Disponível em: <http://www.scielo.br/pdf/rbef/v27n4/a04v27n4.pdf>. Acesso em: 06 mar. 2019.

[6] Esquema proposto por Cirac e Zoller em 1995, que implementa qubits utilizando íons movendo em uma *ion trap* e que interage com luz.

ROSEN, Kenneth H.. Os Fundamentos: Algoritmos, Números Inteiros e Matrizes. In: ROSEN, Kenneth H.. Matemática Discreta e Suas Aplicações. 6. ed. Nova Iorque: Mcgraw-hill, 2009. Cap. 3. p. 167-262.

SHOR, Peter W.. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. Siam Journal On Computing, [s.l.], v. 26, n. 5, p. 1484-1509, out. 1997. Society for Industrial & Applied Mathematics (SIAM). http://dx.doi.org/10.1137/s0097539795293172.

SMOLIN, John A.; SMITH, Graeme; VARGO, Alexander. Oversimplifying quantum factoring. Nature, [s.l.], v. 499, n. 7457, p. 163-165, jul. 2013. Springer Science and Business Media LLC. http://dx.doi.org/10.1038/nature12290.

Contatos: digopasquale@gmail.com e calebe.bianchini@mackenzie.br