Análise do Criptossistema de McEliece

Artur Marzano, Jeroen van de Graaf
Universidade Federal de Minas Gerais (UFMG)
{artur.marzano,jvdg}@dcc.ufmg.br

***Resumo*—A evolução na construção de grandes computadores quânticos, embora devagar, constitui uma preocupação crescente para a comunidade criptográfica. Algoritmos quânticos que ameaçam a segurança de criptossistemas em amplo uso atualmente já foram desenvolvidos, como o algoritmo de Shor, e os algoritmos de criptografia pós-quântica ainda não tiveram tempo para amadurecer em termos de eficiência, confiança e usabilidade. O criptossistema de McEliece é uma alternativa forte para substituir algoritmos tradicionais, como o RSA, nesse futuro hipotético. Neste trabalho inicial estudamos e discutimos as principais características da literatura existente a respeito do McEliece e estabelecemos uma base sólida de conhecimento para a construção de um protótipo na segunda etapa do trabalho.**

***Index Terms*—McEliece PKC, Post-Quantum Cryptography, Coding Theory, Public-Key Encryption, Error Correcting Codes**

I. Introdução

O problema da *comunicação segura* – comunicar uma mensagem a alguém sem que um terceiro possa descobrir o conteúdo da mensagem – é um dos mais antigos problemas criptográficos. Esse problema vem sendo explorado desde a simples cifra usada por Júlio César para se comunicar secretamente com o exército romano até primitivos modernos, como o AES, utilizado para assegurar a privacidade do internauta em grande parte do tráfego da internet mundial.

A criptografia assimétrica é uma forma de abordar esse problema baseada em pares de chaves – uma pública, usada para encriptação, e uma privada, usada para decriptação. Expressa-se a dificuldade de se violar a segurança desse tipo de sistema associando, a cada sistema, um conjunto de hipóteses computacionais a respeito da dificuldade de resolver, em tempo polinomial, algum outro problema mais bem conhecido. Essas hipóteses são condições necessárias para a segurança do sistema e devem ser, em geral, bem aceitas. Alguns dos sistemas de chave pública mais utilizados da atualidade tem hipóteses computacionais ancoradas em problemas tais como o *problema da fatoração de inteiros* (RSA) e o *problema do logaritmo discreto* (DHE, ECDHE, ECDSA, etc).

Com o desenvolvimento das pesquisas no campo da Computação Quântica, no entanto, a segurança de sistemas ancorados nesses problemas corre risco de ser severamente enfraquecida e até quebrada. Há mais de duas décadas atrás, foram apresentados algoritmos quânticos para resolver esses problemas em tempo polinomial [18], o que introduziu a necessidade pelo desenvolvimento de algoritmos de criptografia pós-quântica.

O criptossistema de McEliece é um dos candidatos para esse posto. Recentemente, uma versão do McEliece (*Classic McEliece*) foi submetida para o *Round 2* do processo realizado pelo NIST para a avaliação e padronização de algoritmos criptográficos de chave pública resistentes a computadores quânticos. No trabalho proposto será apresentada uma explicação do funcionamento do criptossistema de McEliece, bem como discussões acerca de suas características, vantagens e desvantagens. Na segunda etapa do trabalho planeja-se desenvolver uma implementação do McEliece, de forma a estimular o uso do sistema e facilitar a prototipação de protocolos criptográficos baseados nesse sistema.

II. Referencial Teórico

O criptossistema analisado neste trabalho é um sistema de criptografia assimétrica baseado na Teoria dos Códigos [15]. Assim, os dois campos representam o principal referencial teórico do trabalho.

A. Criptografia Assimétrica

A **criptografia simétrica** inclui todos os sistemas criptográficos que utilizam uma única chave compartilhada entre as partes para que ambos possam trocar mensagens cifradas. Essa abordagem introduz um problema, pois depende, contraditoriamente, de um **canal seguro** para comunicação da chave. Uma solução clássica para esse problema é utilizar um protocolo de **estabelecimento de chaves**, que é capaz de estabelecer uma chave compartilhada entre duas partes, de maneira secreta, sem depender de um canal seguro. Ao invés disso, essa abordagem requer apenas um **canal autenticado**, isso é, deve ser possível assegurar que as mensagens trocadas pelo protocolo realmente vieram das partes envolvidas. Caso contrário, um atacante poderia se posicionar no meio da comunicação e estabelecer uma chave falsa com cada uma das partes – o chamado ataque MiTM (*man-in-the-middle*).

A **criptografia assimétrica**, por outro lado, utiliza pares de chaves para as partes envolvidas. Nessa abordagem, a preparação do sistema é simplificada: cada parte publica uma única vez a sua chave para as demais (custo linear), enquanto na criptografia simétrica cada parte envolvida deve estabelecer uma chave com cada uma das demais (custo quadrático). Além da construção de algoritmos para a comunicação segura, a criptografia assimétrica tem outras aplicações de interesse, como a geração de assinaturas digitais e o próprio estabelecimento de chaves.

B. Códigos corretores de erros

Um **código corretor de erros** $\mathcal{C}$ é uma codificação aplicada a uma mensagem, antes de sua transmissão por um canal ruidoso, de forma a adicionar redundância à mensagem que

permita detectar e corrigir um conjunto de erros ocorridos na transmissão. Se x é uma mensagem, chamaremos de $\phi(x)$ a palavra-código de $\mathcal{C}$ associada a essa mensagem. Diferentes códigos corretores de erros são capazes de detectar e corrigir diferentes conjuntos de erros.

A partir daqui, estaremos lidando apenas com **códigos binários**, nos quais o domínio de ϕ é $\mathbb{F}_2^k$ e o contra-domínio é $\mathbb{F}_2^n$. Esse arranjo é útil pois modela a forma como computadores representam informação (arranjos de bits).

A **distância de Hamming** $d(x, y)$, definida como a quantidade de posições distintas entre dois vetores x e y, tem um papel fundamental na teoria dos códigos. Se a menor distância de Hamming entre duas palavras-código é dada por:

$$m = \min_{\forall x,y} d(x, y)$$

Então o código é capaz de detectar todos os erros de até $m - 1$ bits, pois todo erro de até $m - 1$ bits na transmissão levará a uma palavra que não faz parte da imagem de ϕ (denominadas de **palavras inválidas**). Adicionalmente, $\mathcal{C}$ é capaz de corrigir todos os erros de até $t = \left\lfloor \frac{m-1}{2} \right\rfloor$ bits, pois, se uma mensagem x é transmitida com um erro de até t bits, a palavra-código mais próxima da palavra recebida no contra-domínio será $\phi(x)$.

Chamamos de (n, k, m)**-código** um código que gera palavras-código de n elementos, operando sobre palavras de k elementos, no qual a distância mínima entre duas palavras-código quaisquer é m.

A **taxa** do código, dada por $r = k/n$, mede a proporção da mensagem codificada que corresponde a informação útil. Ao escolher um código para aplicação em correção de erros, em geral buscamos um *trade-off* entre a taxa (quanto maior a taxa, menor o overhead de espaço), capacidade de correção e eficiência do algoritmo de correção associado.

C. Códigos lineares

Um **código linear** é um código corretor de erros no qual qualquer combinação linear de duas palavras-código também é uma palavra-código. O **peso** de uma palavra-código é a quantidade de elementos não-nulos nela. Note que o peso de uma palava-código x é igual à distância entre x e a palavra $\vec{0} = \begin{pmatrix} 0 & 0 & \dots & 0 \end{pmatrix}$. Em um código linear, a distância mínima entre duas palavras-código corresponde ao menor peso dentre as palavras-código, pois:

$$\begin{aligned} m &= \min_{\forall x,y} d(x, y) \\ &= \min_{\forall x,y} d(x - y, y - y) \\ &= \min_{\forall x,y} d(x - y, \vec{0}) \\ &= \min_{\forall z \in \mathbb{F}_2^n} d(z, \vec{0}) \\ &= \min_{\forall z \in \mathbb{F}_2^n} w(z) \end{aligned}$$

1) Codificação: Em códigos lineares, $\mathcal{C}$ (a imagem de ϕ) é um **subespaço** de $\mathbb{F}_2^n$, e chamamos de **geradora** uma matriz G cujas linhas são uma base desse subespaço. Definimos, portanto, a função codificadora como $\phi(x) = xG$.
2) Detecção de erros: Outra matriz notável no estudo de códigos lineares é a matriz de **verificação de paridade**. Dada uma matriz de verificação de paridade H para o (n, k, m)-código $\mathcal{C}$, pode-se identificar se uma palavra-código $w \in \mathbb{F}_2^n$ possui erros de transmissão de até $m - 1$ bits se o vetor Hw, denominado de **vetor síndrome**, é não-nulo. Por outro lado, se $Hw = \vec{0}$, então $w \in \mathcal{C}$ e a palavra-código foi transmitida sem erros de até $m - 1$ bits.

Exemplo. Suponha que temos um $(7, 4, 3)$-código especificado pela matriz G a seguir:

$$G = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Calculamos a matriz de verificação de paridade H:

$$H = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

O código $\mathcal{C}$ é um subespaço de $\mathbb{F}_2^7$ e o conjunto original de blocos possíveis é $\mathbb{F}_2^4$. Embora existam palavras com distância 1 em F_2^4, como $a = \begin{pmatrix} 1 & 0 & 1 & 0 \end{pmatrix}$ e $b = \begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix}$, a distância entre quaisquer duas palavras de $\mathcal{C}$ é no mínimo 3. De fato, para as palavras escolhidas, as palavras-código associadas são $\phi(a) = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 0 \end{pmatrix}$ e $\phi(b) = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$, com distância 3.

Assim, se transmitimos $\phi(b)$ e ela sai no outro lado do canal como $w = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$, com um erro de 2 bits, sabemos que ela não é uma palavra-código válida, pois $w \notin \mathcal{C}$.
3) Decodificação: No código do exemplo anterior, se uma mensagem fosse transmitida com um erro de até 1 bit, sabemos que a palavra-código válida em $\mathcal{C}$ mais próxima de w é $\phi(b)$, pois, como as palavras-código estão a uma distância mínima de 3 umas das outras, seria necessário um erro de no mínimo 2 bits para que a palavra-código mais próxima pudesse deixar de ser a palavra-código da palavra original.

Associa-se, a algumas classes de códigos lineares, algoritmos específicos para o processo de decodificação. Para um código linear genérico, no entanto, não há algoritmo de tempo polinomial para a decodificação, e esse problema é conhecidamente NP-difícil. Esse fato é um dos principais pilares da segurança do McEliece.

III. O Criptossistema de McEliece

O criptossistema de McEliece faz uso de uma classe particular de códigos, os códigos de Goppa, para os quais existem decodificadores eficientes.

Um decodificador eficiente *Dec* para um código $\mathcal{C}$ é um algoritmo de tempo polinomial que realiza a *decodificação*

de uma mensagem codificada por esse código e acrescida de ruído, revertendo o efeito da codificação e do ruído.

Seja $\mathbf{c}$ uma mensagem $\mathbf{m}$ codificada pelo código determinado por G e acrescida de um ruído $\mathbf{z}$, cujo peso não excede a capacidade de correção t do código gerado por G. Então *Dec* realiza a operação:

$$\text{Dec}(\mathbf{c}) = \text{Dec}(\mathbf{m}G + \mathbf{z}) = \mathbf{m}$$

A. Parâmetros do sistema

O sistema primeiro é preparado com parâmetros fixos t, k e n, que determinam o grau de segurança desejado.

B. Geração de chaves

Suponha que Bob quer enviar uma mensagem a Alice. Alice deve ter gerado um par de chaves e publicado sua chave pública para que Bob seja capaz de encriptar mensagens para Alice. A geração de chaves é realizada como descrito abaixo:

1) Alice escolhe uma matriz G de dimensões $n \times k$. G deve ser uma geradora de um código binário $\mathcal{C}$ capaz de corrigir t erros.
2) Alice gera uma matriz aleatória e inversível S de dimensões $k \times k$.
3) Alice gera uma matriz de permutação aleatória P de dimensões $n \times n$.
4) Alice computa a matriz $G' = SGP$.
5) A chave pública de Alice é (G', t) e a chave privada é (S, G, P).

No McEliece original, utiliza-se um código de Goppa no lugar de $\mathcal{C}$. A ideia é que, para gerar a chave pública, o código G é mascarado por uma transformação secreta para gerar G', um código linear indistinguível de uma matriz binária aleatória. Assim, o recipiente da mensagem cifrada, que conhece os parâmetros da transformação (sua chave privada), será capaz de anular o efeito da transformação e obter a mensagem original com o algoritmo decodificador do código original, mas um adversário que tenha apenas a geradora do código transformado (chave pública) não será capaz de decodificar mensagens encriptadas utilizando esse código.

C. Encriptação

Suponha que Bob quer enviar para Alice uma mensagem dada por $\mathbf{m} \in \mathrm{F}_2^k$.

Bob gera um vetor de ruído $\mathbf{z} \in \mathrm{F}_2^n$ aleatório, de peso t, e computa o texto cifrado $\mathbf{c}$ codificando a mensagem $\mathbf{m}$ com o código definido por G' e introduzindo $\mathbf{z}$, um ruído artificial de t erros.

$$c \coloneqq \mathbf{m}G' + \mathbf{z}$$

D. Decriptação

Para decriptar a mensagem cifrada ($\mathbf{c}$), Alice, em posse de sua chave privada (S, G, P), realiza o seguinte procedimento:

1) Computa $\mathbf{c}' \coloneqq \mathbf{c}P^{-1}$.
2) Decodifica $\mathbf{c}'$ com o decodificador do código, obtendo $\mathbf{m}' \coloneqq Dec(c') = \mathbf{m}S$.
3) Computa a mensagem original: $\mathbf{m} \coloneqq \mathbf{m}'S^{-1}$.

E. Corretude do sistema

A seguir verificamos a corretude do algoritmo – em outras palavras, provamos que a decriptação funciona. Note que:

$$\begin{aligned}\mathbf{c}' &= \mathbf{c}P^{-1} \\ &= \mathbf{m}G'P^{-1} + \mathbf{z}P^{-1} \\ &= \mathbf{m}SGPP^{-1} + \mathbf{z}P^{-1} \\ &= \mathbf{m}SG + \mathbf{z}P^{-1}\end{aligned}$$

Note que $d(\mathbf{c}', \mathbf{m}SG) = t$, pois $\mathbf{z}P^{-1}$ é simplesmente uma permutação das entradas de $\mathbf{z}$ (portanto, $\mathbf{z}P^{-1}$ também tem peso t).

Assim, o algoritmo decodificador do código pode ser utilizado para obter $\mathbf{m}S$:

$$\begin{aligned}\mathbf{m}' = \text{Dec}(\mathbf{c}') &= \text{Dec}((\mathbf{m}S)G + (\mathbf{z}P^{-1})) \\ &= \mathbf{m}S\end{aligned}$$

Como S é inversível, segue que $\mathbf{m} = \mathbf{m}'S^{-1}$.

F. Formato das mensagens

O sistema é projetado para lidar com mensagens de k bits que, ao serem encriptadas, se tornam mensagens cifradas de n bits. Para mensagens de tamanho variável, embora possamos separar as mensagens em blocos de tamanho k e realizar a encriptação bloco-a-bloco, esse mecanismo não é popular para sistemas de criptografia assimétrica, uma vez que as cifras de bloco de criptografia simétrica são usualmente mais eficientes para esse tipo de tarefa.

G. Tamanho das chaves

O elevado tamanho das chaves é, muitas vezes, apontado como o principal problema do McEliece. A chave pública de 32kB para os parâmetros recomendados pelo artigo original já era muito maior que chaves comuns de outros criptossistemas. Uma implementação mais atual do McEliece, por exemplo, estipulou parâmetros que levam a chaves públicas de 65KB a 1MB a depender do nível de segurança desejado [5].

IV. Segurança

O *problema do McEliece* consiste em, dado uma chave pública (G', t) do McEliece e uma mensagem cifrada $\vec{c} \in \mathbb{F}_2^n$, encontrar a mensagem única $\vec{m} \in \mathbb{F}_2^k$ tal que $d(\vec{m}G', \vec{c}) = t$.

McEliece afirma, no artigo original, que não acredita haver esperança para ataques baseados em reverter a chave pública

para obter a chave privada (ataques estruturais). Essa conclusão é apoiada no fato de que, para parâmetros suficientemente grandes de n e k, há um número astronômico de possibilidades para as matrizes S, G e P, de forma que não é factível realizar um procedimento de força-bruta focado em adivinhar essas matrizes. De fato, com exceção de ataques estruturais capazes de enfraquecer a segurança de instâncias do McEliece baseadas em classes particulares de códigos de Goppa, como as que vieram a ser chamadas de *weak keys* [13], não foram propostos ataques estruturais para a classe geral de códigos de Goppa utilizáveis no McEliece.

Dessa forma, restam ataques cujo alvo é obter a mensagem diretamente, sem reverter a chave pública. Por exemplo, dada uma mensagem criptografada $\vec{c}$, podemos verificar, para cada palavra-código válida $\vec{s}$, se $d(\vec{s}, \vec{c}) \leq t$, a fim de determinar a mensagem original. Para testar as palavras-código correspondentes a cada possível mensagem de $\mathbb{F}_2^k$, teríamos 2^k possibilidades, um custo impeditivo para valores suficientemente grandes de k.

Embora o exemplo citado não represente ameaça para parâmetros razoáveis do McEliece, podem haver ataques que, ao invés de quebrar o criptossistema por completo, elevam o tamanho das chaves necessárias para um mesmo padrão de segurança. Em 2008, por exemplo, o McEliece com os parâmetros originais $(n, k, t) = (1024, 524, 50)$ foi efetivamente quebrado [7] por uma versão melhorada do ataque de Stern [20], o que elevou significativamente os requisitos sobre os tamanho necessário dos parâmetros para o mínimo de segurança aceitável.

A. Hipóteses

A segurança do McEliece, assim como outros criptossistemas, é ancorada em um conjunto de hipóteses computacionais a respeito de problemas que acredita-se serem difíceis. As hipóteses do McEliece são:

1) A matriz geradora da chave pública (G') é indistinguível de uma matriz binária aleatória.
2) A decodificação de códigos lineares aleatórios é difícil.

A primeira hipótese não é verdade para todos os tipos de códigos de Goppa utilizados no McEliece [10]. No entanto, para a classe geral de códigos binários de Goppa, não há, até então, um algoritmo capaz de distinguir G' de uma matriz binária aleatória. O problema base da segunda hipótese é NP-completo [4] e não há, até então, algoritmos eficientes para resolvê-lo.

B. Information-set decoding

A ideia básica de ataque que inspirou os ataques mais eficientes contra o sistema da atualidade, agrupados sob o nome de *ataques de information-set decoding*, se baseia em encontrar um conjunto de k posições sem erros (um *information set*) na mensagem cifrada. Ao restringir $\mathbf{c}$ e G' às colunas correspondentes a essas posições sem erro, obtendo $\mathbf{c}_*$ e G'_*, obtemos também um sistema linear:

$$\mathbf{c}_* = \mathbf{m}G'_*$$

Note que, ao restringir as posições das matrizes iniciais, $\mathbf{c}_*$ e $\mathbf{m}$ são matrizes $1 \times k$. Se G'_* for uma matriz inversível, a mensagem pode ser obtida simplesmente resolvendo o sistema. Caso contrário, podemos tentar adivinhar outros conjuntos de k posições sem erro, até que encontremos um G'_* inversível. Esse processo é razoavelmente simples e as matrizes inversíveis ocorrem aproximadamente 29% das vezes; o gargalo está em adivinhar um conjunto de posições sem erro.

Uma série de variações e otimizações dessa ideia básica, como [20], [3], e [14], foram propostas desde então. No entanto, nenhuma delas causou uma ruptura significativa do sistema geral.

C. Segurança semântica

Denomina-se **segurança semântica** a noção informal de que o texto cifrado não vaza informação acerca do texto pleno além de seu tamanho.

Um sistema pode possuir segurança semântica sob uma variedade de ataques, notavelmente *chosen-plaintext*, *chosen-ciphertext* e *adaptative chosen-ciphertext*. Quando um sistema possui segurança semântica sob cada um desses ataques, respectivamente, o sistema é dito ter segurança IND-CPA, IND-CCA ou IND-CCA2. Cada uma das propriedades citadas generaliza a anterior para uma propriedade mais forte – sistemas IND-CCA2 são IND-CCA e sistemas IND-CCA são IND-CPA. Um dos objetivos do projetista de algoritmos criptográficos para uso prático é propor sistemas que possuem, provadamente, segurança semântica para a visão mais avançada possível com relação às capacidades do atacante.

Definição (IND-CPA). Um sistema de criptografia assimétrica é dito IND-CPA se um adversário com capacidades polinomiais não é capaz de vencer o seguinte jogo com probabilidade superior a $\frac{1}{2}$:

1) O *challenger* gera, aleatoriamente, um par de chaves do sistema (pk e sk).
2) O *adversary* recebe a chave pública do *challenger* (pk), com a qual ele é livre para realizar quaisquer operações em tempo polinomial.
3) O *adversary* gera duas mensagens distintas de mesmo tamanho, m_0 e m_1, e as envia para o *challenger*.
4) O *challenger* escolhe, aleatoriamente e de maneira justa, uma das duas mensagens (m_b), que ele encripta com sua chave pública e responde ao *adversary*.
5) O *adversary* tenta adivinhar qual das mensagens foi encriptada pelo *challenger*.

O McEliece original não possui segurança IND-CPA, pois o adversário consegue adivinhar a mensagem escolhida pelo *challenger* no jogo apresentado em 100% dos casos realizando o seguinte procedimento:

1) Após receber a mensagem cifrada $C = m_b G' + e$, o adversário calcula $x_0 = C - m_0 G'$ e $x_1 = C - m_1 G'$.

2) O adversário verifica os pesos de x_0 e x_1. Se x_0 tem peso t, o atacante escolhe m_0. Se x_1 tem peso t, o atacante escolhe m_1.

O método funciona pois temos que ou $x_0 = e$ ou $x_1 = e$. Suponha que $x_0 = e$. Então x_0 é um vetor de peso t e $x_1 = (m_0 - m_1)G'$ é uma palavra-código não-nula e, portanto, x_1 possui peso pelo menos $2t+1$. O caso $x_1 = e$ é perfeitamente análogo.

Esse problema pode ser resolvido de duas formas principais:

1) **Adaptar o criptossistema adicionando mais aleatoriedade à mensagem, de forma que o sistema resultante possua propriedades de segurança semântica**. Um exemplo real de adaptações dessa natureza é o chamado *Randomized McEliece* [16], adaptação IND-CPA do McEliece proposta em 2008, que recentemente foi base para uma adaptação IND-CCA por outros autores [1]. No *Randomized McEliece*, aleatoriedade é introduzida inserindo bits aleatórios à esquerda da mensagem ($m' = r|m$) antes da encriptação.

2) **Projetar um mecanismo de encapsulação de chaves (KEM) com propriedades de segurança semântica para o McEliece**. Nessa abordagem, o McEliece serviria apenas para encriptar uma chave de sessão gerada aleatoriamente a ser transmitida antes da comunicação real, e a comunicação em si seria encriptada por algum outro algoritmo utilizando a chave transmitida. O *Classic McEliece* [8] é um exemplo de uma proposta atual desse tipo.

D. Segurança pós-quântica

O algoritmo quântico de Grover [11] é um algoritmo capaz de, com alta probabilidade e realizando $\mathcal{O}(\sqrt{n})$ avaliações de uma função f, encontrar, dentre n possibilidades, a entrada de f que produz determinada saída (pré-imagem). No modelo tradicional de computação, seria necessário $\mathcal{O}(n)$ avaliações de f para determinar essa entrada. O algoritmo representa a possibilidade, no modelo quântico, de uma redução quadrática no custo de ataques contra uma variedade de primitivos criptográficos, como funções de hash criptográfico e algoritmos de criptografia simétrica. Bernstein, em 2010, apresentou uma forma de utilizar o algoritmo de Grover para reduzir a complexidade de um ataque contra o McEliece e demonstrou que, frente a esse método, seria necessário quadruplicar o tamanho das chaves para reter um mesmo nível de segurança [6].

V. Direções Futuras

A academia vêm seguindo diferentes linhas de pesquisa a fim de melhorar sistemas baseados no McEliece e torná-los mais eficientes e práticos. Dentre os principais objetivos de pesquisa, destacamos:

1) Reduzir o tamanho das chaves necessárias para um mesmo nível de segurança.

2) Projetar variantes do McEliece e KEMs derivados com prova de segurança semântica.

O primeiro objetivo é frequentemente abordado a partir da exploração de classes de códigos alternativas aos códigos de Goppa. Embora haja uma longa linha de pesquisa a respeito criptossistemas similares baseados em outros códigos, essa linha também vem com uma longa história de falhas, como visto, por exemplo, em [19] [12] [2] e [9]. Embora nem todas as variantes tenham sofrido ataques significativos, os códigos binários de Goppa permanecem como a alternativa mais madura até então, principalmente em virtude de seu impressionante histórico de resistência a ataques.

O segundo objetivo está em constante pesquisa e há artigos de diversos autores que apresentam sistemas derivados com provas de segurança semântica sob uma variedade de ataques, como visto em [17] e [8].

VI. Conclusão

O algoritmo de McEliece, embora ainda possua um longo caminho pela frente, continua uma forte alternativa para substituir primitivos criptográficos tradicionais na eventual ocasião de grandes computadores quânticos se tornarem práticos. À medida que os recursos computacionais aumentam, inclusive, espera-se que os elevados tamanhos das chaves do criptossistema sejam cada vez menos uma preocupação. Mesmo assim, a pesquisa sobre métodos de redução do tamanho das chaves deve continuar e podem chegar a viabilizar o uso prático do sistema mais cedo que o esperado. Além disso, há grande expectativa quanto aos resultados que virão do processo de padronização de algoritmos de criptografia pós-quântica do NIST. Neste trabalho estudamos e discutimos os principais aspectos do funcionamento e da segurança do McEliece. Na segunda etapa do trabalho, com esse conhecimento agregado, planejamos implementar um protótipo do criptossistema de McEliece, de forma a contribuir para a comunidade estimulando e facilitando a experimentação e utilização do sistema.

AGRADECIMENTOS

Os autores gostariam de agradecer a todos os colegas do laboratório INSCRYPT que contribuíram com discussões interessantes acerca do tema em questão.

REFERÊNCIAS

[1] Franz Aguirre Farro e Kirill Morozov. "On IND-CCA1 Security of Randomized McEliece Encryption in the Standard Model". Em: *Code-Based Cryptography*. Ed. por Marco Baldi, Edoardo Persichetti e Paolo Santini. Cham: Springer International Publishing, 2019, pp. 137–148. ISBN: 978-3-030-25922-8.

[2] Magali Bardet et al. "Cryptanalysis of the McEliece Public Key Cryptosystem Based on Polar Codes". Em: *Post-Quantum Cryptography*. Ed. por Tsuyoshi Takagi. Cham: Springer International Publishing, 2016, pp. 118–143. ISBN: 978-3-319-29360-8.

[3] Anja Becker et al. "Decoding Random Binary Linear Codes in 2n/20: How $1 + 1 = 0$ Improves Information Set Decoding". Em: *Advances in Cryptology – EUROCRYPT 2012*. Ed. por David Pointcheval e Thomas Johansson. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 520–536. ISBN: 978-3-642-29011-4.

[4] E. Berlekamp, R.J. McEliece e H.C.A. van Tilborg. "On the inherent intractability of certain coding problems (Corresp.)" Em: *Information Theory, IEEE Transactions on* 24.3 (1978), pp. 384–386. ISSN: 0018-9448. DOI: 10.1109/TIT.1978.1055873. URL: http://authors.library.caltech.edu/5607/1/BERieeetit78.pdf.

[5] Daniel J. Bernstein, Tung Chou e Peter Schwabe. "McBits: Fast Constant-Time Code-Based Cryptography". Em: *Cryptographic Hardware and Embedded Systems - CHES 2013*. Ed. por Guido Bertoni e Jean-Sébastien Coron. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 250–272. ISBN: 978-3-642-40349-1.

[6] DanielJ. Bernstein. "Grover vs. McEliece". English. Em: *Post-Quantum Cryptography*. Ed. por Nicolas Sendrier. Vol. 6061. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010, pp. 73–80. ISBN: 978-3-642-12928-5. DOI: 10.1007/978-3-642-12929-2_6. URL: http://cr.yp.to/codes/grovercode-20100303.pdf.

[7] DanielJ. Bernstein, Tanja Lange e Christiane Peters. "Attacking and Defending the McEliece Cryptosystem". English. Em: *Post-Quantum Cryptography*. Ed. por Johannes Buchmann e Jintai Ding. Vol. 5299. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2008, pp. 31–46. ISBN: 978-3-540-88402-6. DOI: 10.1007/978-3-540-88403-3_3. URL: http://cr.yp.to/codes/mceliece-20080807.pdf.

[8] *Classic McEliece: conservative code-based cryptography*. 2019. URL: https://classic.mceliece.org/nist/mceliece-20190331.pdf.

[9] Alain Couvreur, Irene Marquez Corbella e Ruud Pellikaan. "A Polynomial Time Attack against Algebraic Geometry Code Based Public Key Cryptosystems". Em: *CoRR* abs/1401.6025 (2014). arXiv: 1401.6025. URL: http://arxiv.org/abs/1401.6025.

[10] J. Faugère et al. "A Distinguisher for High-Rate McEliece Cryptosystems". Em: *IEEE Transactions on Information Theory* 59.10 (2013), pp. 6830–6844. ISSN: 1557-9654. DOI: 10.1109/TIT.2013.2272036.

[11] Lov K. Grover. *A fast quantum mechanical algorithm for database search*. 1996. arXiv: quant-ph/9605043 [quant-ph].

[12] Grégory Landais e Jean-Pierre Tillich. "An Efficient Attack of a McEliece Cryptosystem Variant Based on Convolutional Codes". Em: *Post-Quantum Cryptography*. Ed. por Philippe Gaborit. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 102–117. ISBN: 978-3-642-38616-9.

[13] Pierre Loidreau e Nicolas Sendrier. "Weak keys in the McEliece public-key cryptosystem". Em: *Information Theory, IEEE Transactions on* 47 (abr. de 2001), pp. 1207 –1211. DOI: 10.1109/18.915687.

[14] Alexander May e Ilya Ozerov. "On Computing Nearest Neighbors with Applications to Decoding of Binary Linear Codes". Em: *Advances in Cryptology – EUROCRYPT 2015*. Ed. por Elisabeth Oswald e Marc Fischlin. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 203–228. ISBN: 978-3-662-46800-5.

[15] R. J. McEliece. "A Public-Key Cryptosystem Based On Algebraic Coding Theory". Em: *Deep Space Network Progress Report* 44 (jan. de 1978), pp. 114–116.

[16] Ryo Nojima et al. "Semantic security for the McEliece cryptosystem without random oracles". Em: *Designs, Codes and Cryptography* 49.1 (2008), pp. 289–305. ISSN: 1573-7586. DOI: 10.1007/s10623-008-9175-9. URL: https://doi.org/10.1007/s10623-008-9175-9.

[17] Edoardo Persichetti. "Code-based Key Encapsulation from McEliece's Cryptosystem". Em: *CoRR* abs/1706.06306 (2017). arXiv: 1706.06306. URL: http://arxiv.org/abs/1706.06306.

[18] P.W. Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". Em: *SIAM Review* 41 (1995), pp. 303–332.

[19] V. M. SIDELNIKOV e S. O. SHESTAKOV. "On insecurity of cryptosystems based on generalized Reed-Solomon codes". Em: *Discrete Mathematics and Applications* 2.4 (1992). DOI: 10.1515/dma.1992.2.4.439. URL: https://doi.org/10.1515/dma.1992.2.4.439.

[20] Jacques Stern. "A method for finding codewords of small weight". English. Em: *Coding Theory and Applications*. Ed. por Gérard Cohen e Jacques Wolfmann. Vol. 388. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1989, pp. 106–113. ISBN: 978-3-540-51643-9. DOI: 10.1007/BFb0019850. URL: http://dx.doi.org/10.1007/BFb0019850.