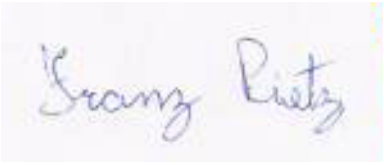
MS777 - Projeto Supervisionado I

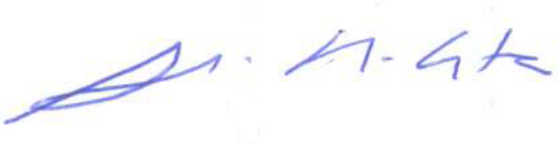
Projeto: Reticulados, Codigos e Criptografia

Período 01/03/2008 à 30/06/2010 (*)

Aluno: Franz Pietz



Orientadora: Prof[a]. Dra. Sueli Irene Rodrigues Costa



Instituto de Matemática, Estatística e Computação Científica

Universidade Estadual de Campinas

(*) Trabalho desenvolvido com o suporte do Projeto de Iniciação Científica Bolsa FAPESP - 2008 / 06171-3, com período de 01/08/2008 à 31/07/2010

Sumário

Resumo de Estado

O propósito é o de introduzir o aluno à teoria de códigos corretores de erros e a tópicos de geometria discreta associados. Naturalmente o objetivo subjacente é o de complementar sua formação em matemática nestas subáreas e propiciar sua interação com um grupo de pesquisa ativo. Este projeto está vinculado ao Projeto Temático FAPESP "Teoria da Informação e Códigos (2007/56052-8) com vigência de 04/2008 a 03/2012, coordenado pela orientadora. Integram este projeto 9 pesquisadores de Matemática, Eng. Elétrica e Computação, pesquisadores colaboradores e alunos de pós-graduação da FEEC, do IMECC e do IC da Unicamp.

Introdução - Descrição geral dos temas abordados no projeto

A teoria de códigos corretores de erros tem como objetivo conseguir sistemas de codificação que permitam uma maior confiabilidade nos canais de transmissão de informação. Ela integra a grande área do conhecimento chamada de "Teoria da Informação", cujo marco inicial é o famoso artigo de C.E. Shannon intitulado "A Mathematical Theory of Communications" (1948) que influenciou o desenvolvimento de teorias matemáticas relacionadas à tecnologias de comunicação e informação. Desde a publicação do artigo, diversas sub-áreas da matemática têm sido utilizadas para a resolução de problemas relacionados à Teoria da Informação (dentre elas a Álgebra Linear de corpos finitos e Geometria Discreta), onde destacamos os códigos lineares e os reticulados, objetos de estudo deste prejeto de iniciação científica. Códigos lineares são, grosso modo, códigos obtidos pela adição de redundância (com propriedades lineares) em um conjunto de "palavras" de um certo tamanho em um alfabeto. De maneira mais precisa, são códigos gerados por transformações lineares injetivas entre dois espaços vetoriais sobre corpos finitos. São fortemente utilizados na prática por herdarem todas as propriedades de espaços vetoriais, o que facilita a busca por códigos "bons" (que corrigem/detectam o maior número de erros com a menor redundância) e o desenvolvimento de algoritmos de codificação/decodificação. Dentre os códigos lineares, destacam-se os códigos de Hamming, Golay e Reed-Solomon, que podem ser vistos em [2].

1 Tópicos desenvolvidos

Inicialmente, foram estudados os capítulos 1 de [1] e os capítulos 1 e 5 de [2], que apresentam os fundamentos básicos de códigos corretores de erros e o processo de decodificação. Em seguida, foram estudados os capítulos 3 e 4 de [5], para aprofundamento dos conceitos de decodificação e códigos de Hamming, respectivamente.

Para a seção de reticulados foram estudados o capítulo 2 de [1] e os capítulos 2 e 4 de [8], que apresentam os fundamentos básicos e os tipos mais conhecidos de reticulados. Para uma introdução em técnicas de decodificação em reticulados, foram estudados tópicos dos artigos [10], [11] e [12], que tratam do problema da decodificação esférica e o problema de mínimos quadrados inteiros.

Para a seção de criptografia, foi estudado [6], principalmente o capítulo 4, sobre criptografia baseada em códigos e o criptossistema de McEliece, e o capítulo 5, que apresenta uma proposta para criptografia baseada em reticulados.

2 Metodologia

A metodologia utilizada foi o estudo individual dos capítulos dos livros e artigos citados, resolução de questões para fixação das idéias e aprofundamento de tópicos, além de reuniões regulares com o professora orientadora para discussão dos temas. Foi utilizado, também, o espaço e recursos do Laboratório Matemática Discreta e Códigos, IMECC - Unicamp, incluindo contato com alunos de pós-graduação que desenvolvem pesquisas em áreas relacionadas ao projeto.

Além disso, participei dos seminários do grupo de pesquisa, com a presença de professores e alunos de graduação e pós graduação dos Intituto de Computação e Faculdade de Engenharia Elétrica e Computação, os quais envolveram diversos tópicos em Teoria de Códigos Corretores de Erros e Criptografia. No primeiro semestre de 2010, me matriculei na matéria MC889 - Introdução à Criptografia, para aprofudamento no estudo de criptografia e seus fundamentos básicos.

Dentro dos mesmos temas, participei nos seguintes eventos no semestre anterior:

- Palestra do professor Dr. Johannes A. Buchmann, da Technische Universität Darmstadt, Alemanha, com o título "Post-Quantum Signatures", realizada no dia 10 de Junho de 2009, no Instituto de Computação da Unicamp.
- 9º Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais, realizado no Centro de Convenções da Unicamp, entre os dia 28 de Setembro e 02 de Outubro de 2009. Nesses simpósio, assisti palestras de professores convidados, como Kenny Paterson, e participei dos Minicursos "Introdução a ataques por canais secundários" e "Segurança em redes colaborativas: desafios e prospostas de soluções", ambos com carga horária de quatro horas.

3 Códigos lineares e decodificação

Nessa seção, apresentaremos os conceitos básicos de códigos lineares vistos em capítulos de [1], [2], [4] e [5]. Foi estudado, particularmente, o processo de decodificação por síndrome, sua complexidade em dimensões maiores e o uso dessa característica no criptossistema de McEliece, apresentado em [6].

3.1 Conceitos Básicos

Dado um corpo finito K, um **código linear** C é um susbespaço de dimensão k em K^n. Este código pode ser caracterizado como imagem de uma função linear injetora de K^k em K^n, e , portanto, por uma **matriz geradora** G, que é a matriz $n \times k$ da transformação linear em relação às bases de K^k e K^n. Um código linear também pode ser considerado como núcleo de uma transformação linear de K^n em K^{n-k}, associando ao código, uma **matriz de paridade** de

ordem $(n-k) \times n$ de maneira que $y = (y_1, y_2, ..., y_n) \in K^n$ pertence ao código C se, e somente se, $H.y^t = 0$.

Dizemos que G e H estão na forma padrão se seguem a seguinte estrutura:

$$G = \left[\begin{array}{c} I_{k\times k} \\ \hline A_{n-k\times k} \end{array} \right]; \; H = \left[\begin{array}{c|c} -A^t_{n-k\times k} & I_{n-k\times n-k} \end{array} \right]$$

Seja $x \in K^n$. Definimos o **peso** de x como $\omega(x) = |\{i : x_i \neq 0\}|$, ou seja, a quantidade de posições não nulas de um vetor. O peso de um código linear C é dado por $\omega(C) = \min\{\omega(x) : x \in C \backslash \{0\}\}$.

A **distância de Hamming** é definida como a quantidade de posições diferentes entre dois elementos de um código. Dados x e y, ambos pertencentes a C, temos que $d_h(x, y) = |\{i;\, x_i \neq y_i\}|$[1]. A **distância mínima** entre as palavra do código C é dada por $d(C) = mind_h(x, y);\, x \neq y$. Nos códigos lineares, a distância mínima entre as palavras do codigo e o peso mínimo das palavras do código coincidem.

Um código é capaz de corrigir $t = \left\lfloor \frac{d-1}{2} \right\rfloor$ e detectar até $d-1$ erros [1], onde t é chamada de **capacidade de correção** do código.

Denotamos por $C(n, k, d)$ um código linear C de dimensão k em K^n que possui distância mínima d.

Proposição [2]: Seja H uma matriz de paridade de C. O peso de C é maior ou igual a s se, e somente se, quaisquer $s-1$ colunas de H são linearmente independentes (LI).

Demonstração: Suponha $s-1$ colunas de H linearmente independentes. Seja $c = (c_1...c_n) \neq 0$ e $h^1, ..., h^n$ colunas de H. Como H.$c' = 0 \Rightarrow 0 = H.c' = \sum c_i.h^i$. Daí, como $\omega(c)$ é o número de componentes não nulas de c, segue que se $\omega(c) \leq s-1$, teríamos uma combinação nula de um número t, $1 \leq t \leq s-1$, de colunas de H, o que é uma contradição. Reciprocamente, para $\omega(c) \geq s$ e $s-1$ colunas linearmente dependentes (LD), $(h^{i1}, h^{i2}, ...h^{is-1})$, então existiriam $c_i1, ..., c_is-1$, nem todos não nulos, tais que $c_{i1}.h^{i1} + ... + c_is-1.h^{is-1} = 0$. Assim, $c = (0, ..., ci1, 0, ...cis-1, 0, ..., 0) \in C$ e, assim, $\omega(c) \leq s-1 < s$, o que é um absurdo.

Teorema 1 [2] **:** Dada uma matriz de paridade H. A distância mínima entre as palavras do código C é igual a $d = s$ se $s-1$ colunas de H são LI e s colunas são LD.

Demonstração: Seja $\omega(c) = s = d$, logo, $s-1$ colunas de H são LI. No entanto, existem s colunas de H que são LD, pois, pela proposição, teríamos $\omega(c) \geq s+1$. Para $s-1$ colunas LI e s colunas LD, temos $\omega(c) \geq s$, mas $\omega(c)$ não pode ser maior que s, pois, pela proposição, teríamos s colunas LI.

3.1.1 Códigos de Hamming

Os **códigos de Hamming**, propostos por Richard Hamming em 1950, são códigos binários ($K = \mathbb{Z}_2$) com parâmetros $(2^r - 1, 2^r - r - 1, 3)$, com $r \geq 2$, sendo capazes de detectar e corrigir um erro, já que a distância mínima é 3[1]. As colunas da matriz de paridade H_r são formada por todas as r-tuplas não nulas de $\mathbb{Z}_2^r$. Por exemplo, para r = 3, temos:

$$H_3 = \left[\begin{array}{cccc|ccc} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{array}\right]$$

Por serem econômicos, códigos de Hamming são normalmente utilizados na memória RAM (Random Acess Memory ou Memória de Acesso Aleatório) dos computadores e no sistema RAID (Redundant Array of Independent Drives ou Conjunto Redundante de Discos Independentes) de armazenamento, que utiliza dois ou mais discos rígidos para armazenar uma informação, visando a segurança dos dados utilizando redundâncias.

3.2 Decodificação em códigos

A decodificação códigos é o processo de analisar a mensagem recebida, corrigindo possíveis erros. Existem vários métodos conhecidos para decoficação de códigos lineares, tais como dicionário de decodificação, matriz de decodificação e a decodificação por síndrome, que será discutida a seguir.

Se transmitimos um código linear C com **comprimento** n, a transmissão é sujeita a interferência/ruido, ou seja, ocorre a adição de um vetor erro.

Seja $c \in C$ a mensagem transmitida e $r \in F^n$ o **vetor recebido**. Para $c \neq r$, definimos o **vetor erro** como $e = r - c \in F^n$. O peso de e, $\omega(e)$, é igual ao número de posições diferentes entre c e r. Seja H a matriz de paridade de C. O resultado do produto $H.x^t = \vec{s}$ é chamado de **síndrome**. Como $e = r - c$, temos $He^t = H(r^t - c^t) = Hr^t - Hc^t \Rightarrow He^t = Hr^t$, ou seja, o vetor e possui mesma síndrome que o vetor r, que é um fato importante para a decodificação e será abordado mais adiante.

Seja $v \in F^n$. Chamamos classe lateral a direita o conjunto $v + C = \{v + c : c \in C\}$

Lema [2]: $u, v \in K^n$ tem a mesma síndrome se, e somente se, $u \in v + C$.

Demonstração: $Hu^t = Hv^t \Leftrightarrow H(u^t - v^t) = 0 \Leftrightarrow u - c \in C \Leftrightarrow u \in v + C$. Ou seja, os elementos de uma classe lateral possuem a mesma síndrome. Assim, podemos definir conjuntos com os possíveis erros, realizando deslocamentos em cada uma das posições dos elementos de C.

Chamamos de **líder de classe** o elemento com o menor peso entre os elementos de uma classe lateral do código. Como os elementos de uma classe lateral possuem mesma síndrome, podemos montar um dicionário (ou tabela) de síndromes, contendo o líder de cada classe lateral e suas respectivas síndromes [5].

Ex: Seja $C(5,2,3)$ um código linear sobre $\mathbb{Z}_2$. Cujas matrizes geradora e paridade são:

$$G = \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \\ \hline 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{array}\right]; H = \left[\begin{array}{cc|ccc} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{array}\right]$$

Então $C = \{00000, 10011, 01101, 11110\}$ e suas classes laterais, além da nula, são:

$$00001 + C = \{00001, 10010, 01100, 11111\}$$

$$00010 + C = \{00010, 10001, 01111, 11100\}$$

$$00100 + C = \{00100, 10111, 01001, 11010\}$$

$$01000 + C = \{01000, 11011, 00101, 10110\}$$

$$10000 + C = \{10000, 00011, 11101, 01110\}$$

Tabela de síndrome:

Líder	Síndrome
00001	001
00010	010
00100	100
01000	101
10000	011

Após encontrar o vetor e, obtemos a palavra correta realizando $c = r - e$.

A utilização da tabela de síndromes é recomendada por apresentar um custo computacional, muito menor do que outros métodos, além de poder ser ordenada de uma maneira conveniente, para facilitar a busca do padrão de erro [5]. Na próxima seção há um comentário sobre os problemas relacionados à esse método de decodificação.

Caso um código possua capacidade de correção $t > 1$, a síndrome obtida pode ser a soma de outras síndromes da tabela, ou seja, a soma de dois ou mais padrões de erro.

Ex: Código (8,2,5):

$$G = \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \\ \hline 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \\ 1 & 1 \end{array}\right]; H = \left[\begin{array}{cc|cccccc} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

Tabela de síndromes:

Líder	Síndrome
00000001	000001
00000010	000010
00000100	000100
00001000	001000
00010000	010000
00100000	100000
01000000	101011
10000000	010111

Vamos supor que $r = (11110111)$. Fazendo $Hr^t = (001011) = (101011) + (100000)$, que é a menor combinação entre os padrões de erro. Assim, com os valores da tabela, temos que o padrão de erro $e = (01100000) \Leftrightarrow c = r - e = (10010111)$, que pertence ao código. Vamos supor agora $r = (01110111)$. Temos $Hr^t = (011100) = (010000) + (001000) + (000100)$ ou $(010111) + (101011) + (10000)$, sendo que os erros iguais a $e_1 = (00011100)$ e $e_2 = (11100000)$, fornecendo $c_1 = (01101011)$, que não pertence ao código, e $c_2 = (10010111)$, que pertence ao código.

Por ter distância mínima 5, esperamos que esse código corrija dois e detecte até quatro erros. Entretanto, o resultado anterior mostra que o código pode corrigir dois erros de maneira segura e que, com o tratamento correto, pode corrigir até três erros. Para isso, poderiamos criar um algoritmo que busque as menores combinações lineares entre as síndromes, decoficando a mensagem recebida utilizando o padrão de erro adequado.

3.2.1 Dificuldades da decodificação por síndrome

A decodificação por síndrome nem sempre é viável. Em dimensões grandes, surgem problemas como a quantidade de cálculos necessários para encontrar um padrão de erro ou a quantidade de espaço necessária para armazenar todos os padrões de erro possíveis.

No primeiro caso, geramos apenas os padrões de erro de uma posição, ou seja, apenas os líderes de classe. Assim, para um código de comprimento n e capacidade de correção t, geramos apenas n padrões de erro e suas possíveis síndromes. Considerando que a capacidade de correção, temos que combinar de 1 até t padrões de erro para encontrar a síndrome adequada. Nesse caso, a quantidade de combinações é dada por

$$\sum_{k=1}^{t} \binom{n}{k}$$

Para $n = 64$ e $t = 5$, temos valores da ordem de 2^{22} (ou 10^6), que são valores aceitáveis. Já para $n = 512$ e $t = 30$, temos um valor da ordem de 2^{160} (ou 10^{48}) possibilidades, o que é computacionalmente intratável.

Outra possibilidade é calcular e armazenar os possíveis padrões de erro, o que tem um custo de memória de

$$\sum_{k=1}^{t}(2n-k)\binom{n}{t}$$

Para obter o padrão de erro de uma determinada síndrome, poderiamos usar um método de busca eficiente. Entretanto, a quantidade de possibilidades para dimensões grandes é da mesma ordem dos valores apresentados anteriormente, tornando o cálculo e armazenagem das síndromes inviável.

Conforme aumentamos a capacidade de correção e a dimensão das mensagens, o problema de decodificaçao por síndrome torna-se mais intratável. Na próxima seção, mostraremos como a dificuldade no cálculo da síndrome pode ser utilizada para construir um sistema criptográfico seguro.

3.3 Criptossistema de McEliece

O **criptosistema de McEliece** é um criptossistema baseado em códigos capaz de, segundo demonstrado até o momento, resistir a ataques de computadores quanticos[6]. Sua segurança baseia-se na dificuldade de detectar e corrigir erros de um código genérico, sem propriedades especiais. O sistema utiliza códigos de Goppa, uma classe de códigos lineares que possue métodos de decoficação eficientes, para codificar uma mensagem m e adiciona exatamente t erros em posições aleatórias, com o objetivo de dificultar os processos de criptoanalise conhecidos, já que uma mensagem pode receber $\binom{n}{t}$ erros diferentes.

O algoritmo do sistema de McEliece é apresentado a seguir[6]:

- **Entradas:** n,$t \in \mathbb{N}$, com $n \gg t$
- **Geração das chaves:**
  - G: uma matriz $k \times n$ geradora de um código de Goppa de dimensão k, irredutível e com distância mínima $d \geq 2t+1$.
  - S: uma matriz $k \times k$ aleatória não singular.
  - P: uma matriz $n \times n$ aleatória de permutação.
  - Faça $G^{pub} = SGP$.
- **Chave Pública:** (G^{pub}, t)
- **Chave Privada:** (S,D_G,P), onde D_G é um algoritmo eficiente para decodificar códigos de Goppa.
- **Encriptação:** dada uma mensagem $m \in \mathcal{F}^k$, escolha um vetor $z \in \mathcal{F}^n$ aleatório e de peso exatamente t e calcule $c = mG^{pub} \oplus z$, onde c é a mensagem cifrada e $\oplus$ é a operação "ou exclusivo".

- **Decriptação:** Para decifrar uma mensagem cifrada c:
  - Calcule $cP^{-1} = (mS)G \oplus zP^{-1}$
  - Em seguida, aplique o algoritmos de decodificação D_G para obter $D_G(cP^{-1}) = mSG$
  - Seja $J \subseteq \{1, ..., n\}$ um conjunto de colunas de G^{pub}tal que G_J^{pub} seja inversível. Calcule $m = (mSG)_J(G_J)^{-1}S^{-1}$.

Os parâmetros base para que o sistema seja considerado seguro são $k = 644$, $n = 1024$ e $t = 38$ [7], mas, naturalmente, valores maiores podem ser utilizados.

Em comparação com outros sistemas criptograficos, o sistema de McEliece possue um grau de segurança maior e complexidade de encriptação e decriptação semelhante ao RSA, por exemplo. O grande empecílho para sua utilização é o tamanho da chave, devido à dimensão das matrizes envolvidas, gerando chaves de alguns quilobytes até vários megabytes[6].

4 Reticulados e decodificação via esferas

Nessa seção apresentamos os conceitos básicos de reticulados, usando como referências [1] e [8], e abordamos o métodos de decodificação via esferas, estudado tópicos de [9], [10], [11] e [12].

4.1 Conceitos básicos

O problema de encontrar o melhor código em $\mathbb{Z}_2^n$ está associado ao problema de **empacotamento esférico**, que trata da distribuição de esferas de raio r de maneira que duas esferas se toquem em apenas um ponto da casca ou não haja intersecção, ocupando a maior área possível[1]. O centro dessas esferas pode ser tomado em pontos de um reticulado. Um subconjunto Λ de $\mathbb{R}^n$ é um **reticulado** se existe uma base $\beta = \{\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_n\}$ de $\mathbb{R}^n$ tal que $\mathbf{x} \in \Lambda$ se, e somente se, $\mathbf{x} = a_1\mathbf{u}_1 + a_2\mathbf{u}_2 + ... + a_n\mathbf{u}_n$ com $a_i \in \mathbb{Z}$ para todo i. A base de um reticulado não é única e a matriz mudança de base M entre duas bases tem coeficientes inteiros e determinante ± 1.

As figuras a seguir mostram dois exemplos de reticulados:

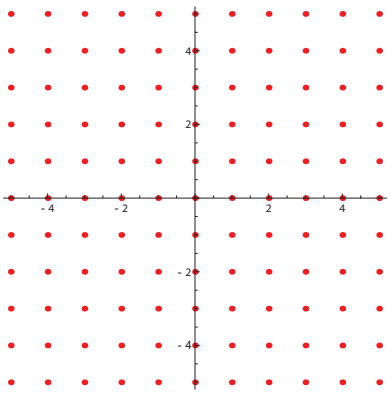


Figura 1: $\beta = \{(0,1),(1,0)\}$

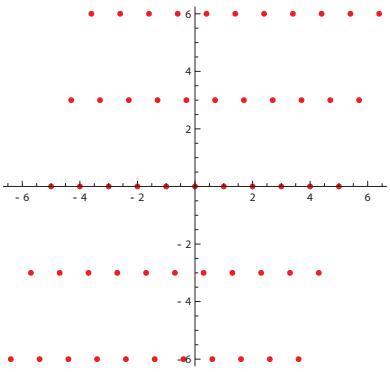


Figura 2: $\beta = \{(0,1),(0.7,3)\}$

Definimos como **raio de empacotamento** o maior raio possível para as esferar sem que haja sobreposição das mesmas centradas nos pontos dos reticulados. Para a base $\beta = \{(0,1),(1,0)\}$, por exemplo, temos o reticulado $\mathbb{Z}^2$ e a distância entre os pontos do reticulado é exatamente 1, ou seja, o maior raio de empacotamento é $\rho = \dfrac{1}{2}$.

Decodificar em reticulados é, basicamente, encontrar o ponto do reticulado mais próximo à um ponto y recebido. Essa tarefa nem sempre é simples para dimensões grandes, podendo se tornar um problema muito dificil. Existem algumas técnicas para diminuir a complexidade desse problema, sendo uma delas a decodificação via esferas, que será detalhada na seção a seguir.

4.2 Decodificação via esferas em reticulados

Nessa seção apresentaremos o método de decodificação chamado **decodificação via esferas** (Sphere Decoding, no original), apresentado por Fincke e Pohst [9] em 1985. Consiste, basicamente, em encontrar todos os pontos localizados dentro de uma esfera S de raio d e centro em y, e encontrar o ponto em S mais próximo de y. A busca dentro de uma esfera reduz o número de operações necessárias para decodificar um ponto. O principal passo desse método é a escolha do raio da esfera [10, 11] e existem alguns problemas relacionados, como:

- Escolha do raio d: um raio muito pequeno pode gerar uma esfera que não contém pontos do reticulado; um raio muito grande pode gerar uma esfera com muitos pontos, aumentando a complexidade do cálculo.
- Listar os pontos que estão dentro de uma esfera: requer o cálculo da distância de cada ponto do reticulado ao centro da esfera. Uma escolha natural para o raio d seria o raio de empacotamento do reticulado [10], definido anteriormente. Entretanto, o calculo do raio de empacotamente de um reticulado genérico é considerado um problema NP-dificil.

Um método de resolução do problema é tratá-lo como um problema de mínimos quadrados inteiros.

4.2.1 Mínimos quadrados inteiros

O problema de **mínimos quadrados inteiros** aparece em aplicações de diversas áreas, tais como comunicações, criptografia e GPS. É definido por:

$$\min_{s\in\mathbb{R}^m} \|Hs - y\| \qquad (1),$$

com $y \in R^n$ e $H \in R^{n\times m}$, sendo s inteiro e y e H reais.

Utilizando a fatoração QR em H, geramos $H = Q\begin{bmatrix}R\\0\end{bmatrix}$, com Q em $\mathbb{R}^{n\times n}$ ortogonal e R em $\mathbb{R}^{m\times n}$ triangular superior. Aplicando QR em (1), temos [11]:

$$\begin{aligned}
&\|Hs - y\|_2^2 \\
&= \left\| Q\begin{bmatrix}R\\0\end{bmatrix} s - y \right\|_2^2 \\
&= \left\| \begin{bmatrix}R\\0\end{bmatrix} s - Q^T y \right\|_2^2 \\
&= \left\| \begin{bmatrix}R\\0\end{bmatrix} s - \begin{bmatrix}Q_1^T\\Q_2^T\end{bmatrix} y \right\|_2^2 \\
&= \left\|Rs - Q_1^T y\right\|_2^2 + \left\|Q_2^T y\right\|_2^2
\end{aligned}$$

Podemos ver que o segundo termo independe de s, reduzindo (1) para

$$\min_{s \in \mathbb{R}^m} \|Rs - \hat{y}\|_2^2 \qquad (2),$$

onde $\hat{y} = Q_1^T y$.

Para o problema de decodificação esférica, queremos calcular $\min_{s \in \mathbb{R}^m} \|Rs - \hat{y}\|_2^2$ com $s \in S\{s \mid \|Rs - \hat{y}\|_2 \leq d\}$, mas para isso, precisamos encontrar um raio d que permita resolver o problema.

4.2.2 Estimativa do raio da esfera

Em [12] é apresentado um método determinístico de estimar um raio d da esfera S. Esse método leva em conta que existe uma restrição do canal para o tamanho das mensagens em aplicações reais. Isso faz com que $\|Hs\|^2$ e $\|s\|^2$ sejam da mesma magnitude.

Utilizando o problema de mínimos quadrados reduzido (2), resolvemos o sistema linear triangular $Rs = y$, que fornece uma solução real. Em seguida, arredondamos os valores de s, para obtermos coordenadas inteiras, obtendo um $\hat{s}$. Calculamos, então o raio da esfera como sendo $\hat{d} = \|R\hat{s} - \hat{y}\|_2^2$. Com base nisso, podemos construir o seguinte algoritmo:

- **Entradas**: Matriz R triangular superior e $\hat{y} = Q_1^T y$, obtido a partir de y reduzido pela fatoração QR.
- **Saida:** O raio de busca $\hat{d}$

1. resolva $s \in \mathbb{R}^m$ em $Rs = \hat{y}$
2. arredonde $\hat{s} = \lceil s \rfloor$
3. defina $\hat{d} = \|R\hat{s} - y\|_2$

Teoricamente, para o método descrito anteriormente, existe pelo menos um ponto s na esfera. Mais precisamente, como $d = \|y–s\|^2$, então s está na superfície da esfera. Na prática, devido a problemas de arredondamento no cálculo dos valores, é possível que não exista nenhum ponto dentro dessa esfera.

Em [11] há a proposta de analise dos erros cometidos, gerando uma estimativa para um valor que, ao ser acrescentado ao raio de empacotamento, inclui pelo menos um ponto ao interior da esfera. Chamando $u = R\hat{s}$, temos:

$$\|u - \bar{u}\|_2 \leq \gamma_m \sqrt{m} \, \|R\|_2 \, \|\hat{s}\|_2,$$

onde, $\gamma_m = \dfrac{m\mu}{1 - m\mu}$ e μé a capacidade de arredondamento.

Assim, o erro no raio calculado é:
$\left|\hat{d} - \bar{d}\right| = \left|\|R\hat{s} - \hat{y}\|_2 - \bar{d}\right|$
$= \left|\|u - \tilde{u} + \tilde{u} - \hat{y}\|_2 - \bar{d}\right|$
$\leq \|u - \tilde{u}\|_2 + \|\tilde{u} - \hat{y}\|_2 - \bar{d}$
$\leq \gamma\sqrt{m}\,\|R\|_2\,\|\hat{s}\|_2 + \|\tilde{u} - \hat{y}\|_2 - \bar{d}$
Onde o termo $\left|\|\tilde{u} - \hat{y}\|_2 - \bar{d}\right|$é o erro computacional cometido ao calcular a diferença $\tilde{u} - \hat{y}$. Assim, podemos modificar o algoritmo anterior para considerar também os erros:

- **Entradas**: Matriz R triangular superior, $\|H\|_2$ e $\hat{y} = Q_1^T y$, obtido a partir de y reduzido pela fatoração QR.
- **Saida:** O raio de busca $\hat{d}$

1. resolva $s \in \mathbb{R}^m$ em $Rs = \hat{y}$
2. arredonde $\hat{s} = \lceil s \rfloor$
3. calcule $\hat{d} = \|R\hat{s} - y\|_2$
4. defina $\hat{d} = \tilde{d} + 2m\left(\sqrt{m}\,\|H\|_2\,\|\hat{s}\|_2 + \bar{d}\right)\mu$

Após obter o raio inicial, sabemos que $\hat{d}^2 \geq \|Rs - \hat{y}\|_2^2$. Como R é triangular superior, podemos escrever a inequação como $\hat{d}^2 \geq \sum_{i=1}^{m}\left(\sum_{j=1}^{m} r_{i,j}s_j - \hat{y}_i\right)^2$, onde $r_{i,j}$representa a posição (i,j) da matriz R.

Expandindo o somatório, temos $\hat{d}^2 \geq (\hat{y}_m - r_{m,m}s_m)^2 + (\hat{y}_{m-1} - r_{m-1,m}s_m - r_{m-1,m-1}s_{m-1})^2 + \ldots$

Podemos ver que o primeiro termo do somatório depende apenas da m-ésima coordenada do ponto s, o segundo termo depende da m e $(m-1)$-ésima entradas, e assim por diante. Assim, para que s pertenca à esfera S, é necessário que $\hat{d}^2 \geq (\hat{y}_m - r_{m,m}s_m)^2$. Assim, temos que $\left\lceil \frac{-\hat{d} + \hat{y}_m}{r_{m,m}} \right\rceil \leq s_m \leq \left\lfloor \frac{\hat{d} + \hat{y}_m}{r_{m,m}} \right\rfloor$, ou seja, temos um intervalo de valores válidos para s_m.

Para cada valor de s_m , definimos $\hat{d}_{m-1}^2 = \hat{d}_{m-1}^2 - (\hat{y}_m - r_{m,m}s_m)^2$ e $\hat{y}'_{m-1} = \hat{y}_{m-1} - r_{m-1,m}s_m$. Temos que, se $\hat{d}_{m-1}^2 \geq \left(\hat{y}'_{m-1} - r_{m-1,m-1}s_{m-1}\right)^2$, então podemos definir também um intervalo para $s_m - 1$, dado por

$$\left\lceil \frac{-\hat{d}_{m-1} + \hat{y}_{m-1}}{r_{m-1,m-1}} \right\rceil \leq s_{m-1} \leq \left\lfloor \frac{\hat{d} + \hat{y}_{m-1}}{r_{m-1,m-1}} \right\rfloor$$

Repetindo esse procedimento, podemos obter os intervalos para todas as coordenadas de s. Calculando o valor de s_1, podemos calcular a norma de cada vetor s e obter, assim, o ponto do reticulado mais próximo do ponto y recebido e, finalmente, realizar a decodificação.

A desvantagem desse método está na necessidade de calcular um intervalo para cada valor do intervalo obtido para o nível anterior. Ao final do processo, temos uma árvore, onde cada folha representa um valor de s_1 e podemos calcular a norma de cada vetor obtido percorrendo a árvore de cada folha até a raiz.

No momento, estamos estudando , juntamente com alunos do Laboratório de Otimização Combinatória do Instituto de Computação, uma proposta para otimização desse algoritmo de busca, utilizando fundamentos de projetos de algoritmos e manipulação de árvores, fizando um melhor desempenho para o algoritmo proposto anteriormente. Infelizmente, até o momento de entrega deste relatório, não houve uma melhora significativa para o tempo de execução do algoritmo.

4.3 Proposta de criptosistema baseado em reticulados

No Capítulo 5 de [6] , é apresentado e há uma proposta de melhoria de um sistema criptográfico baseado em reticulados, entitulado GGH, proposto inicialmente em [13]. A estrutura desse sistema se assemelha ao sistema de McEliece, discutido anteriormente, mas sua segurança se baseia na dificuldade de decodificar em um reticulado genérico.

Basicamente, o sistema GGH utiliza como chave privada uma de base "boa" B para um reticulado Λ, formada por vetores curtos e quase ortogonais. Essa escolha permite decodificar de maneira eficiente, pois apresenta uma melhora significativa no cálculo de problemas como encontrar o vetor mais curto, problema computacionalmente dificil para reticulados genéricos. Como chave privada, temos uma base "ruim" H para o mesmo reticulado Λ, ou seja, temos $\Lambda(B) = \Lambda(H)$, de maneira que seja dificil decodificar um ponto recebido utilizando H. Em [14], há uma proposta para que essa base "ruim" seja a Forma Normal de Hermite de $\Lambda(B)$, que é uma matriz triangular inferior essencialmente única obtida por um método eficiente, similar à eliminação gaussiana. Assim, teríamos uma representação geral de todas as bases para Λ, não conhecendo uma base "boa" para decodificar nesse reticulado.

No processo de encriptação, multiplicamos a mensagem por H, obtendo um ponto v do reticulado. Em seguida, adicionamos um erro r a v, ou seja, realizamos $v + r$. Logo após, calculamos $(v + r) \bmod H$, obtendo um valor chamado $r \bmod H$, que torna difícil o processo de criptoanálise, já que é simples calcular $(v + r) \bmod H$ para qualquer ponto v.

Para decriptação, devemos encontrar o ponto v'mais próximo de $(v + r) \bmod H$, que só é simples conhecendo B. Como dito anteriormente, a segurança desse criptosistema está na dificuldade de encontrar o vetor mais próximo a $(v + r) \bmod H$ sem conhecimento prévio de uma base B com propriedades que permitam decodificar eficientemente pontos recebidos.

Referências

[1] Lavor C. C., Alves M. M. S, Siqueira R. M e Costa S. I. R, "Uma Introdução à Teoria de Códigos", Notas em Matemática Aplicada SBMAC vol. 21 2006

[2] A. Hefez e M. A. T. Villela, "Códigos corretores de erros", IMPA, 2002.

[3] F.J. Mac-Williams e N.J.A. Sloane, "Theory of Error Correcting Codes", North-Holland, 1977.

[4] W.C. Huffman e V. Pless, "Fundamentals of Error-Correcting Codes",Cambridge University Press, 2003.

[5] J.I. Hall, "Notes on Coding Tremovendoheory", http://www.mth.msu.edu/%7Ejhall/classes/codenotes/coding-notes.html.

[6] J.I. Hall, "Notes on Coding Tremovendoheory", http://www.mth.msu.edu/%7Ejhall/classes/codenotes/coding-notes.html.Bernstein, D.J.; Buchmann, J; Dahem, E. "Post-Quantum Cryptography", Springer-Verlag, 2009.

[7] Menezes,A.J.; Vanstone,S.A.; van Oorschot, A.C. "Handbook of Applied Cryptography", CRC Press, 2001.

[8] J.H. Conway and N.J.A. Sloane. "Sphere Packings, Lattices and Groups". Springer-Verlag, New York, 1988.

[9] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis", Mathematics of Computation, vol. 44, no. 170, pp. 463-471, Apr. 1985.

[10] B. Hassibi and H. Vikalo, "On the Sphere Decoding Algorithm I. Expected Complexity", IEEE Transactions on Signal Processing, vol. 53, no. 8, pp. 2806-2818, Aug. 2005.

[11] Zhao, F. and Qiao, S. 2009. Radius selection algorithms for sphere decoding. In Proceedings of the 2nd Canadian Conference on Computer Science and Software Engineering (Montreal, Quebec, Canada, May 19 - 21, 2009). B. C. Desai and C. K. Leung, Eds. C3S2E '09. ACM, New York, NY, 169-174

[12] Sanzheng Qiao, "Integer least squares: Sphere decoding and the LLL algorithm", Proceedings of C3S2E-08, ACM International Conference Proceedings Series, pp. 23-28, May 2008.

[13] Goldreich, O., Goldwasser, S., and Halevi, S.: Public-key cryptosystems from lattice reduction problems. In Advances in cryptology, volume 1294 of Lecture Notes in Comput. Sci., pages 112–131. Springer (1997).

[14] Micciancio, D.: Improving lattice based cryptosystems using the hermite normal form. In J. Silverman, editor, Cryptography and Lattices Conference — CaLC 2001, volume 2146 of Lecture Notes in Computer Science, pages 126–145. Springer-Verlag, Providence, Rhode Island (2001).