Hamming Quasi-Cyclical: um esquema pós-quântico baseado em códigos

Luã Jaz R. A.

Novembro de 2025

Laboratório de Arquitetura de Redes e Computadores - LARC EPUSP

Criptografia assimétrica

- Remetente e destinatário possuem chaves diferentes.
- A chave de encriptação é pública, enquanto a de decriptação é secreta.
- Chaves, parâmetros e mensagem devem ter uma relação matemática delicada.
- Proteção das informações secretas imbutidas nas informações públicas é feita por problemas computacionalmente difíceis.

Esquemas e seus problemas

- RSA (1977), problema da fatoração:

$$n = p \cdot q \text{ com } p, q \text{ primos, encontrar } p \text{ e } q$$

- Diffie-Hellman (1976), problema do logaritmo discreto:

$$t = g^s \mod n, \text{ dados } t, g, n \text{ encontrar } s$$

Shor, 1999

- Algoritmos quânticos eficientes para solução do problema da fatoração e do problema do logaritmo discreto.
- Com o desenvolvimento de computadores quânticos, ameaça diretamente a criptografia assimétrica clássica.
- Motivou o desenvolvimento de um novo paradigma da criptografia: criptografia pós-quântica (PQC)

Problemas base para PQC

i. Problemas de reticulado

   Kyber, Dilithium, Falcon,…

ii. Problemas de códigos corretores de erros

   HQC, BIKE, McEliece,…

iii. Problemas de *hashes*

   SPHINCS+,…

iv. Problemas de isogenias

v. Probemas de sistemas multivariados

Em 2016, o NIST (Instituto de padrões e tecnologia dos EUA) organiza um processo de seleção de esquemas pós-quânticos para padronização e recomendação de uso.

Algoritmos selecionados

2022:

- Kyber (troca de chaves, baseado em reticulados)
- Dilithium, Falcon (assinatura, baseados em reticulados)
- SPHINCS+ (assinatura, baseado em *hashes*)

2025:

- HQC (troca de chaves, baseado em códigos)

Envio de mensagem por um canal ruidoso:

$$\overset{\text{Remetente}}{1101} \longrightarrow \overset{\text{Canal ruidoso}}{1101 \oplus \underbrace{1000}_{\text{erro}}} \longrightarrow \overset{\text{Destinatário}}{0101}$$

Para possibilitar detecção do erro, enviamos duas vezes:

$$\overset{\text{Remetente}}{1101\ 1101} \longrightarrow \overset{\text{Canal ruidoso}}{1101\ 1101 \oplus \underbrace{1000\ 0000}_{\text{erro}}} \longrightarrow \overset{\text{Destinatário}}{0101\ 1101}$$

Para possibilitar correção, enviamos três:

$$\overset{\text{Remetente}}{1101\ 1101\ 1101} \longrightarrow \overset{\text{Canal ruidoso}}{\oplus \underbrace{1000\ 0000\ 0000}_{\text{erro}}} \longrightarrow \overset{\text{Destinatário}}{0101\ 1101\ 1101}$$

Matrizes geradoras

Vários processos de codificação podem ser vistos como a multiplicação de um vetor mensagem $\mathbf{m}$ por uma matriz geradora G:

$$\underbrace{[1\ 1\ 0\ 0]}_{\mathbf{m}} \cdot \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}}_{G} = \underbrace{[1\ 1\ 0\ 0 \mid 1\ 1\ 0\ 0]}_{\mathbf{c}}$$

Códigos que podem ser vistos dessa maneira são chamados **códigos lineares**.

Matrizes verificadoras

Também podemos descrever as verificações realizadas pelo destinatário com uma equação matricial, usando uma matriz de verificação H :

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}}_{H} \cdot \underbrace{\begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}}_{\mathbf{c}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \implies \begin{cases} \mathbf{c}_1 = \mathbf{c}_5 \\ \mathbf{c}_2 = \mathbf{c}_6 \\ \mathbf{c}_3 = \mathbf{c}_7 \\ \mathbf{c}_4 = \mathbf{c}_8 \end{cases}$$

Na verificação de um código sem erros, temos

$$H \cdot \mathbf{c} = \vec{0}$$

Caso contrário, o resultado da verificação será um vetor não nulo:

$$H \cdot \mathbf{c} = s \neq \vec{0}$$

Esse vetor é chamado **síndrome** do erro, e ele resume toda a informação que o destinatário tem sobre ele.

Cada posição não nula da síndrome é correspondente a uma verificação que falhou.

O problema usado como base para esquemas baseados em códigos surge quando um destinatário vai decodificar um código qualquer:

Decodificação por síndrome (SD): dado um código aleatório de matriz geradora G conhecida e um erro e desconhecido mas de síndrome s conhecida, encontrar o erro e.

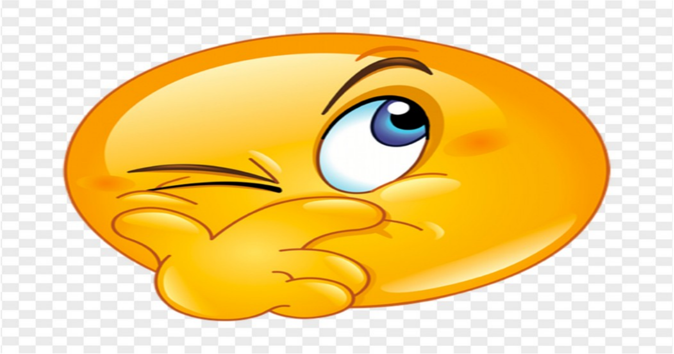
Esse problema é **NP-difícil**, ou seja, existem códigos lineares para os quais não conhecemos algoritmos de decodificação de tempo polinomial.

Além disso, o problema é postulado difícil para o caso médio: um código gerado aleatoriamente (com uma matriz geradora aleatória) provavelmente não possui decodificação eficiente.

Também é postulada difícil uma variação mais fraca do problema, onde o código não é totalmente aleatório, mas é gerado a partir de uma multiplicação com um polinômio aleatório. Essa variação é chamada **decodificação por síndrome de códigos quase-cíclicos (QCSD)**.

Paradoxo: mas daí o código não é inútil?

Se a decodificação de códigos quaisquer é tão difícil, como eles podem ser úteis?



Precisamos de códigos com estrutura, que permitam uma decodificação eficiente!

Códigos Reed-Muller e Reed-Solomon

Sua decodificação é eficiente, pois ela pode ser vista como um problema de interpolação.

i. Dada a mensagem, obtemos um polinômio usando os bits como coeficientes.
ii. Avaliamos o polinômio em todos os vetores possíveis.
iii. A mensagem codificada são os valores do polinômio.
iv. Como um polinômio de grau n está determinado por n+1 pontos, o destinatário decodifica a mensagem como o polinômio determinado pela maioria dos pontos.

Códigos Reed-Muller e Reed-Solomon

São usados, por exemplo, em:

- Códigos QR
- Leitura de CDs
- Transmissão espacial

Além disso, o código usado no HQC é uma combinação dos dois códigos, que denotaremos simplesmente $\mathcal{C}$.

Esse código pode ser visto como uma caixa-preta com decodificação eficiente.

Multiplicação de vetores

Como parte da definição do HQC, é preciso definir uma multiplicação entre vetores. Fazemos isso num anel polinomial:

- Obtemos os polinômios com os bits dos vetores de coeficientes.
- Multiplicamos esses polinômios algébricamente, $\bmod 2$ e $\bmod x^n - 1$.
- Recuperamos o vetor equivalente ao polinômio produto.

$$\mathbf{u} = \begin{bmatrix}\mathbf{u}_1 & \cdots & \mathbf{u}_n\end{bmatrix}$$
$$\mathbf{v} = \begin{bmatrix}\mathbf{v}_1 & \cdots & \mathbf{v}_n\end{bmatrix}$$

$$\mathbf{u}(x) = \mathbf{u}_1 + \mathbf{u}_2 x + \cdots + \mathbf{u}_n x^{n-1}$$
$$\mathbf{v}(x) = \mathbf{v}_1 + \mathbf{v}_2 x + \cdots + \mathbf{v}_n x^{n-1}$$

$$\begin{aligned}\mathbf{u}(x)\cdot\mathbf{v}(x) &= \mathbf{r}_1 + \cdots + \mathbf{r}_n x^{n-1} + \mathbf{r}_{n+1} x^n + \cdots + \mathbf{r}_{2n-1} x^{2n-2} \mod x^n - 1 \mod 2 \\ &= \mathbf{r}_1 + \cdots + \mathbf{r}_n x^{n-1} + \mathbf{r}_{n+1} x^{n \bmod n} + \cdots + \mathbf{r}_{2n-1} x^{2n-2 \bmod n} \mod 2 \\ &= \underbrace{(\mathbf{r}_1 + \mathbf{r}_n)}_{\mathbf{w}_1} + \underbrace{(\mathbf{r}_2 + \mathbf{r}_{n+1})}_{\mathbf{w}_2} x \cdots + \underbrace{(\mathbf{r}_n + \mathbf{r}_{2n-1})}_{\mathbf{w}_n} x^{n-1} \mod 2\end{aligned}$$

$$\mathbf{u}\cdot\mathbf{v} = \mathbf{w} = \begin{bmatrix}\mathbf{w_1} & \cdots & \mathbf{w_n}\end{bmatrix}$$

Construindo um esquema baseado no SD

A ideia por trás dos esquemas baseados em códigos é inserir um erro na mensagem de forma que

- Para um oponente, o erro é grande, então recaimos no SD.
- Para o destinatário com a chave secreta, o erro é pequeno, então é possível corrigí-lo de forma eficiente.

Para gerar as chaves no HQC:

i. Sorteamos x e y com peso pré-definido e público w.
   A chave secreta é o par $sk = (x, y)$.

ii. Sorteamos um vetor h qualquer e calculamos
   $s = x + h \cdot y$. A chave pública é o par $pk = (s, h)$.

Note que x e y estão protegidos em s pelo QCSD: y é a mensagem, a multiplicação por h resulta numa codificação quase-cíclica aleatória e x é um erro de peso conhecido.

Encriptação:

i. Codificamos a mensagem m com o código $\mathcal{C}$,

ii. Sorteamos erros r_1 e r_2 com peso conhecido w_r e um erro e com peso conhecido w_e.

iii. Inserimos um erro grande na mensagem codificada, obtendo

$$v = \text{encode}_{\mathcal{C}}(m) + s \cdot r_2 + e$$

iv. Calculamos uma dica para os erros:

$$u = r_1 + h \cdot r_2$$

v. A mensagem encriptada é o par (u,v).

Note que:

- Tendo r_1 , podemos obter r_2 a partir de u.
- Tendo r_2 e e, podemos obter m a partir de v .

Ou seja, r_1, r_2 e e são quantidade vitais que precisam ser protegidas.

De fato, r_1 e r_2 estão protegidos pelo QCSD em u.

r_1, r_2 e e também estão protegidos pelo QCSD em v.

A melhor opção de um atacante em relação a u e v é fazer a seguinte manipulação:

Seja $z = r_2 \cdot s + e$ o erro total introduzido na mensagem. Então, podemos escrever

$$[u \mid z] = [r_2 \cdot h + r_1 \mid r_2 \cdot s + e] = r_2 \cdot [h \mid s] + [r_1 \mid e]$$

O QCSD garante que s é indistinguível de um vetor aleatório, então r_2 e $[r_1 | e]$ estão protegidos em u e v pelo QCSD.

Decriptação:

i. Calculamos $v - u \cdot y$.

ii. Decdificamos essa quantidade, obtendo

$$m' = \text{decode}_{\mathcal{C}}(v - u \cdot y)$$

iii. Com probabilidade alta, teremos $m' = m$ e a decodificação será bem sucedida.

De fato, expandindo a quantidade $v - u \cdot y$:

$$\begin{aligned} v - u \cdot y &= (\text{encode}_{\mathcal{C}}(m) + s \cdot r_2 + e) - (r_1 + h \cdot r_2) \cdot y \\ &= \text{encode}_{\mathcal{C}}(m) + (x + hy)r_2 + e - r_1 y - h r_2 y \\ &= \text{encode}_{\mathcal{C}}(m) + \underbrace{(x r_2 - y r_1 + e)}_{\text{erro total}} \end{aligned}$$

Maior ameaça ao esquema: resolução do QCSD

Como vimos, várias informações vitais do HQC estão imbutidas nas informações públicas, mas protegidas pelo QCSD.

Dessa maneira, o melhor ataque disponível a um atacante é utilizar o melhor algoritmo disponível para reslolver o QCSD.

Information Set Decoding (ISD)

- Família de algoritmos para resolução do SD com melhor complexidade.
- Inaugurada por Prange (1962).
- Posteriormente aprimorada diversas vezes, mas principalmente por Stern (1989).
- O algoritmo é probabilístico e, no pior caso, nunca termina.
- Sua complexidade é da ordem de 2^w , onde w é o peso de Hamming do vetor erro e.

Ideia do algoritmo original:

Seja G uma matriz geradora $k \times n$ aleatória com $k < n$, m uma mensagem desconhecida, e um erro desconhecido e $c = mG + e$ a mensagem codificada e corrompida.

i. Sorteamos k colunas de G, na esperança de que as respectivas entradas de c não contenham erros.

ii. Seja G' a matriz quadrada dada pelas k colunas escolhidas, c' e e' os vetores dados pelas posições correspondentes de c e e.

iii. Se de fato não escolhemos colunas com erro, teremos:

$$mG' + \underbrace{e'}_{=0} = c' \implies m = c'(G')^{-1}$$

iv. Checamos se isso de fato ocorreu verificando a mensagem obtida:

$$Hm = s$$

v. Se s=0, a decodificação foi bem sucedida. Caso contrário, voltamos ao passo 1.

O ISD resolve o SD e, portanto, resolve também o QCSD. Apesar disso, ele não aproveita a estrutura quase-cíclica dos códigos.

O principal algoritmo que aproveita esse fato é o **Decoding One Out of Many (DOOM)**, que melhora a complexidade em um fator de $\sqrt{n}$, onde n é o tamanho das mensagens codificadas pelo código usado.

Isso não só torna o ataque ao QCSD mais forte, como faz a segurança do HQC depender também do tamanho dos códigos envolvidos.

Outra preocupação em relação aos parâmetros do HQC: o erro final após a redução com a chave secreta deve ser pequeno o suficiente para pode ser corrigido.

Se isso não ocorre, a decriptação falha, o que é tanto um problema de comunicação, quanto um problema de segurança.

A taxa de ocorrência de falhas é chamada **DFR** (Decoding Failure Rate). O HQC se destaca por ser possível deduzir analiticamente um limitante superior para a sua DFR.

Próximos passos

Próximos passos da pesquisa

- Estudar a influência dos parâmetros na segurança do esquema e em outras quantidades como o tamanho das chaves e do texto encriptado, obtendo possíveis trade-offs
- Testar outras combinações de códigos corretores de erro no lugar da combinação Reed-Muller/Reed-Solomon utilizada no HQC em busca de esquemas adjacentes com propriedades interessantes.

Referências

[1] E. Berlekamp, R. McEliece, and H. van Tilborg. On the inherent intractability of certain coding problems (corresp.). *IEEE Transactions on Information Theory*, 24(3):384–386, 1978.

[2] Florence J. MacWilliams and Neil J. A. Sloane. *The theory of error-correcting codes pt. 1-2*. North-Holland Publ. Co, 1977.

[3] National Institute of Standards and Technology. Call for proposals. `https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/Call-for-Proposals`, 2016.

[4] National Institute of Standards and Technology. Selected algorithms. `https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms`, 2017.

Referências

[5] E. Prange. The use of information sets in decoding cyclic codes. *IRE Transactions on Information Theory*, 8(5):5–9, 1962.

[6] Nicolas Sendrier. Decoding one out of many. *Lecture Notes in Computer Science*, page 51–67, 2011.

[7] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review*, 41(2):303–332, 1999.

[8] Jacques Stern. A method for finding codewords of small weight. *Lecture Notes in Computer Science*, page 106–113, 1989.

[9] HQC team. 2025-08-22 hamming quasi-ciclical specification. `https://pqc-hqc.org/doc/hqc_specifications_2025_08_22.pdf`, 2025.

[10] R. Terada. *Segurança de dados: Criptografia em rede de computador*. Editora Blucher, 2008.

Para obter esta apresentação e o documento que a acompanha:

github.com/luajaz/seminario-hqc