

Seminário: HQC

Luã Jaz

November 23, 2025

1 Introdução

Esquemas de criptografia que possuem chaves diferentes para encriptar e decriptar mensagem — os chamados esquemas assimétricos — fazem necessária uma relação matemática delicada entre seus parâmetros e chaves. Nesse caso, a proteção das informações secretas, seja a mensagem, a chave secreta ou algum parâmetro vital do esquema, é baseada em problemas computacionalmente difíceis: extrair essas informações das quantidades públicas deve ser tão difícil quanto resolver esse problema base.

Exemplos disso são os esquemas RSA (1977) e Diffie-Hellman (1976), ambos presentes em protocolos importantes como o TLS, que se baseiam em problemas de teoria dos números: o RSA se baseia no problema da fatoração (dado um produto de dois primos grandes, encontrar os primos) e o Diffie-Hellman no problema do logaritmo discreto (dado o resultado e a base de uma exponenciação em aritmética modular, encontrar o expoente usado) [10].

1.1 A crise na criptografia clássica

Em 1999, Peter Shor publicou [7] algoritmos que resolvem ambos os problemas citados acima em tempo polinomial, o que, junto com os primeiros desenvolvimentos sérios em direção a um computador quântico praticável, colocou o paradigma da criptografia clássica em crise. Isso motivou a busca por algoritmos baseados em outros problemas, que poderiam ser seguros até mesmo contra ataques de computadores quânticos. A área de pesquisa que se preocupa com esses algoritmos é chamada criptografia pós-quântica.

1.2 Bases para a criptografia pós-quântica

Os problemas nos quais os algoritmos pós-quânticos se baseiam podem ser agrupados em (i) algoritmos baseados em reticulados, (ii) algoritmos baseados em códigos corretores de erros, (iii) algoritmos baseados em hashes, (iv) algoritmos baseados em isogenias e (v) algoritmos baseados em sistemas multivariados.

1.3 Processo seletivo do NIST

Em decorrência desse cenário, o NIST (National Institute of Standards and Technology) organizou em 2016 um processo de seleção de esquemas de criptografia pós-quântica para padronização e recomendação de uso [3]. Em 2022, o Kyber, um algoritmo de troca de chaves, e o Dilithium, o Falcon, ambos algoritmos de assinatura, foram selecionados, sendo que todos são baseados em reticulados. O SPHINCS+, algoritmo de assinatura baseada em hash, também foi selecionado em 2022. Por fim, neste ano, o HQC, um algoritmo de troca de chaves baseado em códigos, foi selecionado, motivando nossa pesquisa em relação a sua segurança, seus parâmetros e outros métodos de criptografia adjacentes [4].

2 Códigos Corretores de Erros

Códigos corretores de erros são ferramentas matemáticas que nos permitem transmitir mensagens por canais ruidosos com maior confiança, em troca de algum nível de redundância. Suponha que desejamos enviar a string binária “1101” através de um canal que pode introduzir um erro à mensagem na forma de flipar algum de seus bits com certa probabilidade. Se o primeiro bit é flipado no caminho, o destinatário receberá a mensagem “0101”, e não terá forma alguma de saber se essa era de fato a mensagem original.

Uma forma de mitigar esse problema é simplesmente enviar a mensagem duas vezes, ou seja, codificar a mensagem “1101” como “1101 1101”. Dessa maneira, se o primeiro bit for flipado e o destinatário receber “0101 1101”, ele saberá que um erro foi introduzido na mensagem, pois a string não bate com a sua cópia. Enviando a mensagem mais uma vez ainda, codificando “1101” como “1101 1101 1101”, o destinatário poderá ainda corrigir a mensagem errônea “0101 1101 1101” com probabilidade alta ao tomar a string que ocorre mais frequentemente [2].

2.1 Formalização

Em termos mais matemáticos, o processo de codificação de uma mensagem m em sua forma codificada c pode ser descrito através da multiplicação do vetor mensagem m por uma matriz G , denominada matriz geradora:

$$c = m \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} = \mathcal{C}.\text{encode}(m) \bmod 2$$

De fato, isso significa que o conjunto de todas as mensagens codificadas possíveis formam um espaço vetorial, e por isso dizemos que esse é um código linear.

Ao mesmo tempo, o processo de checagem por erros também pode ser visto como uma série de verificações (i.e. o primeiro bit ser igual ao quinto, o segundo

ser igual ao sexto, etc), o que também pode ser descrito de maneira matricial com uma matriz H , denominada matriz de verificação:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} c = Hc = 0 \bmod 2$$

Dessa forma, um código linear pode ser descrito por sua matriz geradora ou por sua matriz de verificação, e as duas matrizes estão intimamente relacionadas [2].

2.2 Síndromes

Note que a verificação de erros de uma mensagem recebida c consiste em multiplicar c pela matriz de verificação H . Se não há erros, o resultado será o vetor nulo. Entretanto, se houver erros detectáveis na mensagem, o resultado da multiplicação Hc será um vetor não nulo s , que é chamado síndrome. Esse vetor consiste na combinação de todas as verificações realizadas em cada posição da mensagem recebida, e ele sintetiza toda a informação que o destinatário possui sobre o erro introduzido na mensagem. De fato, a síndrome não depende da mensagem codificada em específico, mas apenas do erro introduzido [2].

2.3 Problema da decodificação por síndrome

O problema usado como base para a construção de esquemas criptográficos baseados em códigos surge exatamente quando o destinatário pretende de decodificar uma mensagem errônea a partir de sua síndrome. Esse problema é chamado decodificação por síndrome, abreviado SD (syndrome decoding) e consiste em, dado um código linear conhecido mas aleatório (i.e. obtido através de uma matriz geradora aleatória e, portanto, sem nenhuma estrutura especial) e um erro de tamanho predeterminado desconhecido, determinar esse erro a partir da sua síndrome.

É possível provar que esse problema é NP-difícil, de forma que, no pior caso, não se conhece um algoritmo de tempo polinomial em relação ao tamanho dos parâmetros que resolva o problema. Em outras palavras, isso quer dizer que existem códigos que são muito difíceis de se decodificar. Além disso, é conjecturado que esse problema também é difícil para o caso médio [1].

A partir da formulação desse problema, temos uma espécie de paradoxo: se a decodificação de um código qualquer é intratável, como os códigos corretores podem ser utilizados para a comunicação? Isso motiva a busca por códigos com certas estruturas que nos permitam uma decodificação eficiente.

2.4 Códigos Reed-Muller e Reed-Solomon

Os códigos Reed-Muller e Reed-Solomon são famílias de códigos lineares que, para erros de pouco peso, possuem métodos eficientes de decodificação. Isso

ocorre pois ambos podem ser vistos como um problema de interpolação polinomial, que pode ser resolvido eficientemente em casos de erros pequenos.

A ideia por trás desses algoritmos é tomar os bits da mensagem como os coeficientes de um polinômio e então avaliar esse polinômio em diversos pontos. O valor do polinômio nesses pontos será a mensagem codificada a ser enviada.

Uma vez que um polinômio de grau n está totalmente determinado por $n+1$ pontos distintos, o destinatário pode encontrar o polinômio determinado pela maioria dos pontos obtidos e, assim, recuperar a mensagem a partir de seus coeficientes [2].

Por conta da eficiência desses códigos, eles são amplamente utilizados em diversas modalidades de comunicação ruidosa, como códigos QR, leitura de CD's e transmissão espacial. Além disso, esses dois códigos são concatenados para formar o código \mathcal{C} que é usado no HQC, que a partir de agora trataremos como um código caixa-preta que possui uma decodificação eficiente.

3 HQC

A ideia por trás da construção de um esquema de criptografia baseado em códigos é exatamente utilizar o fato de que uma mensagem codificada com um código adequado pode ser decodificada com alta probabilidade quando o erro é pequeno, mas quando o erro é grande recainos no caso geral de decodificação por síndrome, que é NP-difícil. Assim, precisamos que o erro seja grande para alguém que intercepta a mensagem, mas pequeno para quem possui a chave secreta.

Como parte da especificação do esquema, é preciso definir uma multiplicação entre vetores binários. Isso é feito dentro de um anel polinomial $\mathbb{Z}_n[x]/(x^n - 1)$: a multiplicação entre vetores \mathbf{u} e \mathbf{v} é realizada considerando os polinômios com as entradas de cada vetor nos coeficientes e realizando a multiplicação algébrica mod 2 e mod $x^n - 1$, o que é equivalente a reduzir o expoente de cada termo mod n .

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

3.1 Keygen

Para a chave secreta, sorteamos dois vetores x e y , cada um com peso predeterminado w . Para a chave pública, sorteamos um vetor h qualquer e calculamos $s = x + h \cdot y$. Aqui, a multiplicação de y por h codifica y com um código linear gerado pelo vetor h . Uma vez que h é um vetor aleatório e x é um vetor de peso conhecido, y e x estão protegidos em s por uma variação do problema SD. [9].

3.2 Encriptação e decriptação

Suponha que desejamos encriptar a mensagem m . Primeiro, codificamos a mensagem com o código linear, obtendo $\mathcal{C}.\text{encode}(m)$, e então sorteamos erros r_1 e r_2 com pesos w_r , e o erro e com peso w_e . Finalmente, calculamos os vetores $v = \mathcal{C}.\text{encode}(m) + s \cdot r_2 + e$, que introduz um erro muito grande à mensagem codificada, e $u = r_1 + h \cdot r_2$, que armazena informação sobre os erros introduzidos para possibilitar e futura decodificação. A mensagem é o par (u, v) .

Os erros r_1 , r_2 e e estão novamente protegidos em (u, v) pelo problema SD. Chamando de $z = r_2 \cdot s + e$ o erro introduzido na mensagem codificada, um atacante poderia reorganizar as quantidades da seguinte maneira:

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

Note que nesse caso o vetor $[u \mid z]$ é a síndrome do erro $[r_1 \mid e]$ introduzido à mensagem r_2 codificada pelo código gerado por $[h \mid s]$. Dessa maneira, z , r_1 , r_2 e e estão todos protegidos por uma variação do problema SD.

Para decriptar a mensagem, simplesmente calculamos $v - u \cdot y$ e, supondo que o erro restante é pequeno, podemos aplicar a decodificação e obter a mensagem $m = \mathcal{C}.\text{decode}(v - u \cdot y)$ [9].

3.3 Prova de corretude

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

$$\begin{aligned} v - u \cdot y &= (\mathcal{C}.\text{encode}(m) + s \cdot r_2 + e) - (r_1 + h \cdot r_2) \cdot y \\ &= \mathcal{C}.\text{encode}(m) + (x + hy)r_2 + e - r_1y - hr_2y \\ &= \mathcal{C}.\text{encode}(m) + \underbrace{(xr_2 - yr_1 + e)}_{\text{erro total}} \end{aligned}$$

Ou seja, os parâmetros de peso dos erros e dos componentes x e y da chave secreta devem ser escolhidos de forma que o erro $s \cdot r_2 + e$ presente na mensagem encriptada seja muito grande, enquanto o erro $xr_2 - yr_1 + e$ presente na decriptação seja pequeno o suficiente para ser corrigido. Isso força o atacante a resolver o problema mais geral de decodificação por síndrome, enquanto permite ao destinatário legítimo uma decodificação eficiente.

4 Segurança do HQC

Como vimos previamente, do ponto de vista do atacante que possui apenas a chave pública e a mensagem interceptada, extrair a mensagem m a partir da mensagem encriptada $\mathcal{C}.\text{encode}(m) + s \cdot r_2 + e$ é resolver o problema da decodificação por síndrome. Dessa maneira, os melhores ataques contra o esquema são aqueles que resolvem esse problema de forma mais eficiente.

4.1 Information Set Decoding

A família dos algoritmos mais eficientes na resolução da decodificação por síndrome é chamada Information Set Decoding, que foi inaugurada por Prange em 1962 [5].

A ideia básica por trás desse algoritmo é tentar obter um conjunto de colunas da matriz geradora do código que corresponda a posições do vetor mensagem que não foram afetadas pelo erro. Saberemos que temos um conjunto de colunas adequado, chamado conjunto de informação (information set), calculando a mensagem que seria gerada por ele e utilizando a matriz de verificação do código.

Suponha que a matriz geradora G do código é uma matriz $k \times n$, onde $k < n$, que c é a mensagem a ser decodificada e que e é o erro dessa mensagem. Os passos do algoritmo são:

1. Sorteamos k colunas de G , na esperança de que as respectivas posições de c não contenham erros.
2. Seja G' a matriz quadrada formada por essas colunas, e c' e e' os vetores formados pelas respectivas posições de c e e .
3. Se fomos bem sucedidos em evitar os erros, teremos

3

$$\implies m = c'(G')^{-1}$$

onde m é a mensagem decodificada que obtemos.

4. Checamos se isso de fato ocorreu calculando a síndrome

$$Hm = s$$

Se $s = 0$, m é a mensagem original e a decodificação foi bem sucedida. Caso contrário, voltamos ao passo 1.

Note que este algoritmo é probabilístico, o que torna ele um algoritmo de Las Vegas. Além disso, é importante notar que nenhuma informação é guardada entre diferentes iterações do algoritmo, o que significa que, no pior caso, ele pode sempre escolher as mesmas k colunas contendo algum erro e nunca terminar.

Diversas melhorias foram encontradas para esse algoritmo desde sua concepção, a mais importante delas sendo a publicada por Stern em 1989 [8]. Apesar disso, sua complexidade continua sendo supra-polinomial, mantendo-se na ordem de 2^w , onde w é o peso do erro introduzido na mensagem.

4.2 Decoding One Out of Many

Uma melhoria importante para o caso específico do HQC é a versão do ISD chamada DOOM (Decoding One Out of Many), que usa a estrutura dos códigos

presentes no HQC (mais especificamente, o fato de que esses códigos são quase-cíclicos) para obter uma melhoria significativa na complexidade do algoritmo, resultando em algo da ordem de $2^w/\sqrt{n}$, onde w continua sendo o peso do vetor erro e n é o tamanho das mensagens codificadas com o código em questão [6].

Essa melhoria é importante não só pois torna um potencial ataque por ISD mais eficiente, mas também pois ela torna a segurança do esquema dependente de mais parâmetros: agora ela não só depende o peso dos erros envolvidos, mas também do tamanho do código específico que é usado. Ou seja, se desejamos que o esquema tenha uma segurança, por exemplo, de 128 bits, devemos escolher w e n de maneira que

$$\frac{2^w}{\sqrt{n}} \approx 2^{128}$$

4.3 Decoding Failure Rate

Por outro lado, os parâmetros do esquema devem ser escolhidos de maneira que o erro obtido pelo destinatário após a aplicação da chave secreta sejam adequadamente pequenos com probabilidade muito alta. De fato, se o erro final for grande demais, haverá uma falha na decodificação da mensagem, o que não só é um problema na comunicação do esquema mas também uma vulnerabilidade de segurança, pois diversos ataques são construídos em torno da informação que é obtida quando um oráculo falha em decriptar uma mensagem.

Isso nos leva ao motivo pelo qual o HQC se destacou em relação a outros esquemas baseados em códigos: um limitante superior para o erro final que o destinatário deve corrigir pode ser obtido de maneira analítica, o que permite uma noção mais concreta da segurança do esquema. Dessa manira, é possível estimar um limitante superior para a taxa de falha de decodificação (Decoding Failure Rate, ou DFR) [9].

5 Próximos Passos

Até esse momento, nossa pesquisa se manteve no âmbito da revisão bibliográfica, do cálculo da complexidade dos ataques contra o HQC por meio do ISD e do cálculo da sua taxa de falha de decodificação. Os próximos passos da pesquisa são

1. utilizar essa base construída para estudar a influênciia dos parâmetros na segurança do esquema e em outras quantidades como o tamanho das chaves e do texto encriptado, obtendo assim possíveis trade-offs;
2. 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.

References

- [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.
- [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.