Otimizações em operações de criptografia pós-quântica baseada em reticulados para ambientes restritos

Felipe J. A. Rampazzo , Marco A. A. Henriques
{f233261@dca.unicamp.br, maah@unicamp.br}

Departamento de Engenharia de Computação e Automação (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (Unicamp)
Campinas, SP, Brasil

Resumo – Os esquemas de criptografia de chave pública estão fadados à obsolescência com a possibilidade do surgimento de computadores quânticos, visto que a segurança desses sistemas é sustentada em problemas matemáticos que estes computadores conseguiriam resolver em tempo polinomial. Devido aos efeitos catastróficos que o surgimento desse computador traria para a computação de um modo geral, novos esforços têm sido feitos na criação de algoritmos pós-quânticos que sejam resistentes aos futuros ataques dos computadores quânticos. O National Institute for Standards and Technology - NIST - EUA tem coordenado os estudos neste tema e já selecionou alguns algoritmos baseados em reticulados como padrões de criptografia pós-quântica. Nesse contexto, este trabalho busca avaliar as possibilidades de melhoria em relação ao desempenho e custos de implementação de algoritmos criptográficos pós-quânticos baseados em reticulados em ambientes restritos. A justificativa para tal abordagem se deve às importantes limitações desses ambientes, que geralmente carecem do poder computacional e funcionalidades necessárias às operações criptográficas pós-quânticas.

Palavras-chave – criptografia pós-quântica, reticulados, ambientes computacionais restritos, processadores ARM

1. Introdução

Nos últimos anos é visto uma crescente de projetos para a construção de um computador quântico funcional. Todavia, seu surgimento colocaria em risco a segurança dos criptossistemas de chave pública atuais, mais especificamente os de troca de chaves e assinaturas.

Com a publicação do trabalho de [1], verificou-se que um computador quântico conseguiria quebrar esses criptossistemas em tempo polinomial, tornando-os fadados à obsolescência. Desde então, novos esforços têm sido feitos na criação de algoritmos pós-quânticos que sejam resistentes aos futuros ataques desses computadores.

Na busca desses novos esquemas criptográficos, o NIST iniciou em 2016 o Processo de Padronização de Criptografia Pós-Quântica (PQC Standardization Process), um concurso com o objetivo de selecionar os novos padrões de criptografia de chave pública que sejam resistentes aos computadores quânticos.

Após três rodadas, os primeiros escolhidos foram CRYSTALS-Kyber na categoria de mecanismos de encapsulamento de chave (public-key encapsulation mechanism - KEM) e os de assinatura selecionados foram CRYSTALS–Dilithium, FALCON e SPHINCS, sendo o primeiro deles o recomendado para uma implementação inicial [2]. Uma quarta rodada está em andamento para a definição de outros algoritmos a serem padronizados.

Os métodos matemáticos que garantem a segurança vistos nesses algoritmos incluem os baseados em códigos, reticulados, multivariados, hashes e isogenias. Contudo, ao analisar os algoritmos já padronizados, percebe-se que os construídos sobre problemas de reticulados são predominantes visto que, dos quatro definidos, três sustentam sua segurança nessas estruturas.

Com o intuito de explorar esses problemas, este trabalho procura avaliar as possibilidades de otimização das principais operações presentes em algoritmos criptográficos pós-quânticos baseados em problemas de reticulados em ambientes restritos de software e hardware.

2. Criptografia baseada em reticulados

Os reticulados são conjuntos de pontos discretos no espaço n-dimensional Euclidiano R^n , descritos como todas as combinações lineares inteiras de vetores independentes [3]. Na Figura 1, é mostrado um exemplo de reticulado formado por todas as combinações lineares de $\mathbf{b_1}$ e $\mathbf{b_2}$, como por exemplo o vetor $\mathbf{s}$ formado pela combinação $-2b_1 + b_2$. **Definição 1:** seja $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m$ vetores linearmente independentes em R^n. Um reticulado Λ com bases $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m\}$, em que $m \leq n$ [4, p. 50], é definido por:

$$\Lambda = \{u_1\mathbf{b}_1 + \ldots + u_m\mathbf{b}_m : u_1, \ldots, u_m \in Z\}$$

A criptografia baseada em reticulados usa conjecturas de difícil solução em reticulados de R^n como prova

de segurança para a construção de sistemas criptográficos [3]. O primeiro trabalho desta área é datado em 1996 com a proposta de [5], definida como Short Integer Solution (SIS), que consiste em encontrar o vetor mais curto de um reticulado inteiro dado uma função de mão única baseada neste reticulado. Ainda neste trabalho, foi definido que a complexidade de resolver este problema no caso médio é tão difícil quanto no pior caso.

Contudo, é no ano de 2005 com o trabalho de [6] que a ligação entre sistemas criptográficos e reticulados ganham mais atenção. Regev propõe um método chamado Learning With Errors (LWE) que se torna a base dos criptossistemas que se respaldam em reticulados. No LWE, há duas instâncias do problema consideradas NP-difícil, ou seja, que são tão difíceis de resolver quanto os problemas mais difíceis em NP. Esses problemas são conhecidos como Shortest Vector Problem (SVP) e o Closest Vector Problem (CVP).

No SVP, dado dois vetores bases $\mathbf{b_1}$ e $\mathbf{b_2}$ de um reticulado, deve-se encontrar o vetor mais curto que pertença ao reticulado a partir dessas bases. Na Figura 1, esse vetor é representado por $\mathbf{s}$.

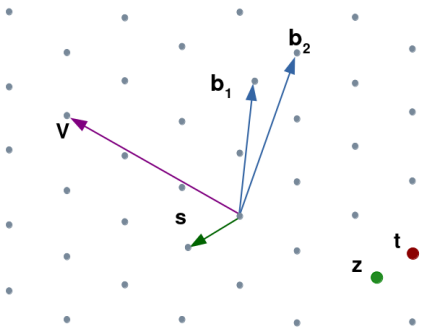

Figura 1. Shortest Vector Problem (SVP) e Closest Vector Problem (CVP). Fonte: `https://blog.cloudflare.com/towards-post-quantum-cryptography-in-tls/`

Já no CVP, a dificuldade está em encontrar um vetor $\mathbf{z}$ que não pertença necessariamente ao reticulado e que seja o mais próximo de um vetor t que é pertencente ao reticulado. Ambos esses problemas se mostraram seguros tanto em ataques de computadores clássicos quanto, na teoria, dos quânticos. Vemos um reticulado de apenas duas dimensões na figura 1. Porém, em modelos reais, as dimensões dos reticulados são muito grandes e, somada a escolha de bases pouco ortogonais entre si, dificultam muito a solução do problema.

2.1. Problema Learning With Errors

O LWE é um problema baseado em reticulados voltado a aplicações criptográficas, em que é necessário resolver um sistema de equações lineares sobre um módulo primo inteiro q. Para montar o sistema criptográfico utilizando o LWE deve-se:

1. Escolher um vetor secreto s, tal que $\mathbf{s} \in Z_q^n$.
2. Vetores públicos (matriz) $\mathbf{a}^i = (a_1^i, \ldots, a_n^i)$, tal que $1 \leq i \leq n$, obtidos por meio de uma distribuição uniforme em Z_q^n.
3. Por fim, deve-se escolher um vetor de erro com coeficientes pequenos, tal que $\mathbf{e} = (e^1, \ldots e^n) \in Z_q^n$ via uma distribuição χ.
4. Assim, a distribuição LWE pode ser dada como

$$b^i = \sum_{j=1}^{n} a_j^i s_j + e^i q$$

A dificuldade em encontrar o vetor secreto $\mathbf{s}$ é aumentada com a adição dos erros ao sistema. Sem esses ruídos, esse sistema poderia ser resolvido a partir da eliminação de Gauss. Assim, para todo $\mathbf{a} \cdot \mathbf{s} \approx b$, uma adição suscetiva de erros sempre resulta em uma aproximação cumulativa que cresce tanto a ponto de não sobra nenhuma informação na "aproximação" obtida, algo que poderia favorecer a descoberta de $\mathbf{s}$. Essa afirmação vale mesmo quando o erro é pequeno.

Para cifrar um bit x é necessário gerar um vetor $\mathbf{v} \in 0, 1^m$ e gerar o par que representa o texto cifrado

$$(\mathbf{c}_1, c_2) = (A\mathbf{v}, \mathbf{b} \cdot \mathbf{v} + x \cdot \left\lfloor \frac{q}{2} \right\rfloor)$$

. Já para decifrar, usamos a chave secreta $\mathbf{s}$ calculada anteriormente e computamos
$c_2 - \mathbf{c}_1 \cdot \mathbf{s} = (A\mathbf{s} + \mathbf{e}) \cdot \mathbf{v} + x \left\lfloor \frac{q}{2} \right\rfloor - (A\mathbf{v}) \cdot \mathbf{s}$
$= (A\mathbf{s}) \cdot \mathbf{v} - (A\mathbf{v}) \cdot \mathbf{s} + \mathbf{e} \cdot \mathbf{v} + x \left\lfloor \frac{q}{2} \right\rfloor \approx x \left\lfloor \frac{q}{2} \right\rfloor$.

Se x for 1, o valor estaria mais próximo de $\left\lfloor \frac{q}{2} \right\rfloor$ do que de zero, visto que os vetores de erro adicionados são pequenos. Do contrário, o valor estaria mais próximo de zero.

Novas variantes do LWE surgiram com o tempo, como o Ring-LWE proposto por [7], em que as operações são sobre anéis polinomiais, e não vetores n-dimensionais do LWE, o que reduz o tamanho das chaves, uma vez que somente a primeira linha da matriz precisa ser armazenada e o restante é calculado partindo da linha anterior, tornando-as próximas do tamanho das chaves do RSA e melhorando a eficiência das construções baseadas em reticulados.

Outra variante do LWE pode ser encontrada em [8] e [9] chamada Module-LWE, visando resolver fragilidades tanto do LWE quanto do RLWE. A diferença principal para o RLWE é a substituição dos elementos do anel único por elementos modulares de um mesmo anel. Este é geralmente um anel ciclotômico de potência de

dois, ou seja, um anel em $Z[\mathbf{X}]/\langle \mathbf{X}^n + 1\rangle$ com $n = 2^k$ [10]. O Module-LWE se tornou modelo mais utilizado desde então nos algoritmos criptográficos baseados em reticulados, implementado nos padrões já definidos de algoritmos pós quânticos, como o CRYSTALS-Kyber, pois algumas operações são facilitadas neste tipo de anel, tornando-os mais performáticos.

2.2. CRYSTALS-Kyber

O CRYSTALS-Kyber é um algoritmo para realizar troca de chaves de forma segura a ataques de computadores pós-quânticos, por meio de um mecanismo de encapsulamento de chaves. O algoritmo é do tipo IND-CCA2 e baseado em um variante do LWE em reticulados modulares, o Module-LWE. A chave encapsulada é formada por 256 bits, permitindo seu uso por algoritmos de chave simétrica como o AES, o que facilita o desenvolvimento de protocolos híbridos, que unem a criptografia pós-quânticas e clássica. Todo o processo de KEM do CRYSTALS-Kyber pode ser dividido em três operações principais: geração de chaves, encapsulamento (ou encriptação) e desencapsulamento (ou decriptação).

Na etapa de de geração de chaves, um par de chaves pública e privada é gerado pela parte que deseja inciar a comunicação segura. Os parâmetros iniciais a serem definidos são um módulo q para os coeficientes e um módulo polinomial $x^n + 1$ (para reduzir o grau do polinômio). A chave pública é formada por duas partes: uma matriz e um vetor. As entradas da matriz $\mathbf{A}$ são polinômios em que os coeficientes são randômicos sobre o módulo q.

Então, é gerado a chave secreta $\mathbf{s}$, porém, com os coeficientes do polinômios "pequenos", da mesma forma que o vetor de erros $\mathbf{e}$. Por fim, para obter a segunda parte da chave pública, o vetor $\mathbf{t}$, realizamos uma multiplicação e adição de matrizes, tal que $\mathbf{t} = \mathbf{As} + \mathbf{e}$.

A seguir, é realizado o encapsulamento da chave secreta compartilhada em que são necessários dois vetores de polinômios $\mathbf{s}$ e $\mathbf{e_1}$ e um polinômio e_2. Para cifrar a mensagem, primeiro, é preciso transformá-la em um polinômio. No caso do CRYSTALS-Kyber, cada bit representa um coeficiente do polinômio. Antes de cifrar a mensagem, precisamos "expandir"o polinômio, para evitar os erros adicionados, multiplicando-o por $\lceil q/2 \rfloor$.

Após esse processo, cifra-se *m* com a chave pública $(\mathbf{A}, \mathbf{t})$, tal que $\mathbf{u} = \mathbf{A}^T\mathbf{r} + \mathbf{e_1}$ e $v = \mathbf{t}^T\mathbf{r} + e_2 + m$. Tanto o vetor $\mathbf{u}$ quanto o polinômio *v* são enviados e consistem no texto cifrado. A dificuldade de se obter m se baseia em quão complexo é resolver o problema do SVP.

A última etapa consiste no desencapsulamento da chave secreta compartilhada. Em posse da chave secreta $\mathbf{s}$, podemos decifrar a mensagem da seguinte forma: $m = v - \mathbf{s}^T\mathbf{u}$. Porém, o resultado ainda não é o que se espera, visto que $m = \mathbf{e}^T\mathbf{r} + e_2 + m + \mathbf{s}^T\mathbf{e_1}$. É necessário desfazer a "expansão", verificando se os valores são mais próximos de $\lceil q/2 \rfloor$ ou de (0 ou q) e assim definir se os bits são 0 ou 1.

Todas essas etapas passam por um processo de conversão que aumenta o nível de segurança do algoritmo de IND-CPA para IND-CCA2, por meio da Transformação Fujisaki-Okamoto.

O CRYSTALS-Kyber conta ainda com funções hash padronizadas pelo NIST na FIPS 202, que inclui as pertencentes à família SHA3 e SHAKE. Além disso, a multiplicação de vetores e matrizes são realizadas a partir de uma implementação da transformada rápida de Fourier para operar sobre anéis e corpos finitos, chamada Number Theoretic Transform (NTT).

3. Resultados

Neste trabalho, dois ambientes restritos são estudados para verificar as possibilidades de melhoria e implementação do algoritmo CRYSTALS-Kyber. O primeiro ambiente é em hardware, com mudanças nas operações de hash em placas ARM M0+. Já o segundo ambiente restrito é baseado em software, em que levantou-se problemas iniciais da implementação para switches programáveis em P4. Ainda não foi definido qual dos ambientes serão alvo da pesquisa, apenas estudos iniciais acerca das características e desafios de cada um deles. Os resultados já obtidos desses ambientes podem ser vistos a seguir.

3.1. Ambiente restrito em hardware - ARM M0+

Os testes iniciais foram realizados em uma placa Freedom FRDM-KL25Z, equipada com MCU KL25Z128, ARM Cortex-M0+, 128 KB de memória FLASH, clock de 48 MHz e 16 KB de memória SRAM, sendo que desses, 3 KB são dedicados ao uso do ambiente de desenvolvimento.

Como pode ser visto em [11], foi possível executar as três funções principais do algoritmo. Contudo, algumas modificações foram necessárias, como a substituição da biblioteca OpenSSL para a geração de números aleatórios pela função *srand* da linguagem C. Embora haja impactos na segurança, essa ação foi precisa dada as limitações do ambiente.

Outra modificação foi aplicada nas funções hash do CRYSTALS-Kyber como a de saída extensível (XOF), as H e G, uma função pseudo-aleatória (PRF) e uma de derivação de chave (KDF). O intuito dessas modificações foi medir o impacto de cada uma na eficiência do algoritmo

em ambientes que carecem de aceleradores em hardware.

3.2. Ambiente restrito em software - P4

O P4 é uma linguagem utilizada para definir como os pacotes serão processados no plano de dados por dispositivos de rede programáveis. Esses dispositivos também precisam prover serviços de segurança para diversos contextos e é possível aplicá-las para a detecção de intrusão, mitigação de DDoS, implantação de ACLs, firewalls, entre outras [12].

Todavia, há uma dificuldade de se aplicar sistemas criptográficos nestes dispositivos, uma vez que apenas operações de aritmética simples [13] são permitidas, além da ausência de funções hashes seguras criptograficamente. Há trabalhos que implementam hashes seguros em P4, como em [14] e [15], porém estes não são os padrões definidos pelo NIST no concurso dos novos algoritmos PQC.

Outra barreira do P4 é a ausência de *loops*, ainda que a recirculação de pacotes possa ser usada para simular essa propriedade. Contudo, o número de vezes que um pacote pode recircular é limitado em alguns modelos de switches. A funcionalidade *#define*, nativa do P4 e presente também na linguagem C, ameniza essa limitação ao criar um *loop* "desenrolado"e evitar a redundância de linhas de código, que naturalmente ocorre devido as características intrínsecas da linguagem P4.

4. Conclusões

Com o iminente surgimento de um computador quântico e os riscos que este trará para as comunicações seguras, novos esforços estão sendo empregados na busca por sistemas criptográficos que sejam seguros a ataques desses dispositivos.

Entretanto, a construção matemática desses criptossistemas são pesados do ponto de vista computacional, o que exige implementações mais otimizadas, principalmente se forem aplicados em ambientes restritos, tanto em software como em hardware.

Dadas essas motivações, este trabalho busca efetuar otimizações nas operações mais pesadas dos algoritmos PQC, como as funções hash e NTT. Já em um ambiente restrito em software como visto na linguagem P4, a contribuição reside na implementação do algoritmo PQC em si, visto que este cenário possui grandes limitantes para o desenvolvimento.

Referências

[1] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484–1509, 1997.

[2] D. Moody, "Status report on the third round of the NIST post-quantum cryptography standardization process," tech. rep., National Institute of Standards and Technology, jul. 2022.

[3] S. I. Costa, F. Oggier, A. Campello, J.-C. Belfiore, and E. Viterbo, *Lattices Applied to Coding for Reliable and Secure Communications*. Springer International Publishing, 2017.

[4] M. F. Bollauf, "Códigos, reticulados e aplicações em criptografia," Master's thesis, Universidade Estadual de Campinas, Campinas-SP, Brasil, 2015.

[5] M. Ajtai, "Generating hard instances of lattice problems (extended abstract)," in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, (New York, NY, USA), p. 99–108, Association for Computing Machinery, 1996.

[6] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *Journal of the ACM (JACM)*, vol. 56, no. 6, 2009.

[7] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," *Journal of the ACM (JACM)*, vol. 60, nov 2013.

[8] A. Langlois and D. Stehlé, "Worst-case to average-case reductions for module lattices," *Designs, Codes and Cryptography*, vol. 75, 06 2014.

[9] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "Fully homomorphic encryption without bootstrapping," *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 18, p. 111, jan. 2011.

[10] M. R. Albrecht and A. Deo, "Large modulus ring-lwe $\geq$ module-lwe," in *Advances in Cryptology – ASIACRYPT 2017* (T. Takagi and T. Peyrin, eds.), (Cham), pp. 267–296, Springer International Publishing, 2017.

[11] L. F. C. Ferro, F. J. A. Rampazzo, and M. A. A. Henriques, "Estudos de otimização do algoritmo de criptografia pós-quântica CRYSTALS-KYBER," in *Anais Estendidos do XXI Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSeg Estendido 2021)*, Sociedade Brasileira de Computação - SBC, 2021.

[12] E. F. Kfoury, J. Crichigno, and E. Bou-Harb, "An exhaustive survey on p4 programmable data plane switches: Taxonomy, applications, challenges, and future trends," *IEEE Access*, vol. 9, pp. 87094–87155, 2021.

[13] Y. Gao and Z. Wang, "A review of p4 programmable data planes for network security," *Mobile Information Systems*, vol. 2021, pp. 1–24, 11 2021.

[14] D. Scholz, A. Oeldemann, F. Geyer, S. Gallenmüller, H. Stubbe, T. Wild, A. Herkersdorf, and G. Carle, "Cryptographic hashing in p4 data planes," pp. 1–6, 09 2019.

[15] S. Yoo and X. Chen, "Secure keyed hashing on programmable switches," SPIN '21, (New York, NY, USA), p. 16–22, Association for Computing Machinery, 2021.