UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CIÊNCIA DA COMPUTAÇÃO

Gustavo de Castro Biage

Estudo de Esquema de Assinatura Digital Dilithium

Florianópolis
2022

Gustavo de Castro Biage

Estudo de Esquema de Assinatura Digital Dilithium

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de Bacharel em Ciência da Computação e aprovado em sua forma final pelo curso de Graduação em Ciência da Computação.

Florianópolis, 2022.

Banca Examinadora:

Prof. Thaís Bardini Idalino, Dra.
Orientadora
Universidade Federal de Santa Catarina

Gustavo Zambonin, Me.
Coorientador
University of Ottawa

Prof. Alexandre Giron, Me.
Avaliador
Universidade Federal de Santa Catarina

Prof. Daniel Panario, Dr.
Avaliador
Carleton University

Gustavo de Castro Biage

Estudo de Esquema de Assinatura Digital Dilithium

Trabalho de Conclusão de Curso submetido ao Programa de Graduação em Ciência da Computação do Centro Tecnológico da Universidade Federal de Santa Catarina como requisito para obtenção do título de Bacharel em Ciência da Computação.
Orientadora: Prof. Thaís Bardini Idalino, Dra.
Coorientador: Gustavo Zambonin, Me.

Florianópolis
2022

Aos meus pais.

AGRADECIMENTOS

Primeiro, eu gostaria de agradecer a professora Thaís Bardini Idalino por ter me orientado neste trabalho. Não achei que teria a sorte de ter uma orientadora tão atenciosa. Segundo, quero agradecer a participação da banca, composta pelo professor Alexandre Augusto Giron e o professor Daniel Panario. Obrigado por suas avaliações. Terceiro, gostaria de agradecer meu amigo e coorientador Gustavo Zambonin. Ele não só me auxiliou no trabalho com ótimos conselhos, mas, me forneceu a oportunidade de ser bolsista no Laboratório de Segurança em Computação (LabSEC). Gosto muito de seu jeito peculiar de fazer as coisas. Acredito que ele é uma das razões pela qual eu me encaixei muito bem no laboratório e esta foi uma das melhores oportunidades que eu tive dentro da universidade. Além dele, quero agradecer a Dúnia Marchiori. Ela parece nunca hesitar em manifestar sua opinião quando se trata do bem-estar dos outros. Tive muito prazer em conhecer estes dois e todos os demais companheiros de laboratório. Cada um de vocês completa os meus semestres de maneira única.

Quero agradecer aos meus excelentes amigos de curso, principalmente o Gilson Trombetta Magro, o Matheus Eyng de França e o Arthur Gabriel Crippa Milanez. Passamos pela maioria dos momentos difíceis e divertidos do curso juntos. Apesar do distanciamento da pandemia, a gente se esforçou para não perder o contato e ser aprovado nas matérias. Espero que daqui a quatro anos ainda mantenhamos este contato.

Finalmente, quero expressar meus sentimentos pelas pessoas que conheci antes de meu trajeto dentro da universidade. São eles meus amigos de infância, Gabriel de Castro Biage (grande irmão), João Pedro Bittar de Freitas, Matheus Machado, Nícolas Goeldner, Eduardo Piazza Margarida e Matheus de Oliveira Saldanha. Acho incrível como nós ainda gostamos uns dos outros. Em nenhum momento eu pensei pouco sobre nossa amizade. Por fim, agradeço à minha família. Sou grato pelos meus irmãos, que estiveram comigo por toda minha vida, especialmente o Gabriel e o Guilherme. Eles são pessoas bem maneiras e definiram todos os meus gostos. Além deles, agradeço profundamente os meus pais, Milton e Marina. Eu sei que vocês fizeram todo o possível para que eu tenha as melhores oportunidades do mundo. Cada um com seu jeito especial.

Nunca vou me esquecer de todos vocês!

"*Words you say never seem to live up to the ones inside your head.*"
Chris Cornell

RESUMO

Em criptografia, os esquemas de assinatura digital mais utilizados atualmente baseiam sua segurança na dificuldade de fatorar inteiros grandes ou computar logaritmo discreto. Estes esquemas podem ser quebrados por computadores quânticos poderosos com o algoritmo de Shor. Portanto, esquemas pós-quânticos procuram novas abordagens que forneçam segurança contra tais computadores. Esquemas baseados na complexidade de problemas de reticulados (*lattices*) são foco de pesquisas há algumas décadas. Duas propostas de esquemas de assinatura digital com esta abordagem foram selecionados para a padronização após a terceira etapa do processo de padronização do Instituto Nacional de Padrões e Tecnologia dos EUA (NIST). Um desses esquemas, o Dilithium, é baseado em problemas que envolvem estruturas algébricas de reticulados. O esquema faz uso da heurística de *Fiat-Shamir with aborts*, que rejeita assinaturas que levam ao vazamento de informações privadas. Este trabalho realiza um estudo dos fundamentos de esquemas baseados em reticulados, como também, apresenta a teoria e matemática computacional utilizada. Como resultado, oferece um material que permite a compreensão e implementação do esquema, incentivando o uso da criptografia pós-quântica em assinaturas digitais.

Palavras-chave: Criptografia pós-quântica. Esquema Baseado em Reticulado. Criptografia de Chave Pública.

ABSTRACT

In cryptography, the most common digital signature schemes base their security on the hardness of factoring large integers or computing discrete logarithmic. These schemes may be broken by powerful quantum computers with the help of Shor's algorithm. Therefore, post-quantum cryptography schemes search for new ways to obtain security against such computers. Schemes based on the complexity of certain lattices problems have maintained the interest of researchers for a few decades. Two signature schemes based on lattices have been selected to be standardized after the third round of NIST's post-quantum process. One of these schemes, Dilithium, make use of the Fiat-Shamir with aborts heuristic to obtain a signature scheme from an identification scheme. The aborting technique allows to avoid the leakage of information of the private key, keeping the scheme secure. This work study the fundaments of cryptography schemes based on the algebraic structure of lattices, presenting the computational theory and mathematic behind, and uses Dilithium as learning tool. As a result, we offer a material that allows an easy comprehension of the scheme, incentivizing the use and implementation of post-quatum cryptography algorithms.

Keywords: Post-quantum Cryptography. Lattice Based Scheme. Public Key Cryptography.

LISTA DE FIGURAS

LISTA DE ALGORITMOS

SUMÁRIO

1 INTRODUÇÃO

Em criptografia, os esquemas de assinatura digital mais utilizados atualmente baseiam sua segurança na dificuldade de fatorar inteiros grandes ou computar logaritmo discreto. Como exemplo disso encontra-se o RSA (Rivest–Shamir–Adleman) e o ECDSA (*Elliptic Curve Digital Signature Algorithm*). Estes esquemas podem ser quebrados em tempo polinomial por computadores quânticos poderosos com o uso do algoritmo de Shor (1999). Assim, a partir do avanço da era pós-quântica, existe a necessidade de desenvolver e ter conhecimento da segurança de esquemas criptográficos capazes de garantir autenticidade, integridade e não-repúdio por todo futuro previsto. Historicamente, levou-se quase duas décadas para a implantação de esquemas criptográficos modernos (STANDARDS; TECHNOLOGY, 2022). Portanto, o Instituto Nacional de Padrões e Tecnologia dos EUA (NIST) lançou um esforço de padronização pós-quântico, em que a terceira rodada terminou recentemente, com o objetivo de normatizar um ou mais esquemas de cifragem de chave pública (*public key encryption*), assinatura digital e acordo de chaves (*key exchange*).

Conforme apresentado por Moody (2017) e Moody (2019), na primeira rodada do processo criado pelo NIST houve a submissão de 23 esquemas de assinatura digital. Os dados das submissões de esquema de assinatura digital aceitas para participar na primeira e demais rodadas do esforço, classificados por família, podem ser encontrados na Tabela 1. As famílias mencionadas são conjuntos de esquemas que utilizam princípios de segurança similares. Neste conjunto de famílias encontramos: criptografia baseada em sistemas de equações quadráticas multivariadas, criptografia baseada em reticulados, criptografia baseada em resumos criptográficos e criptografia baseada em códigos de correção de erros. Em cada rodada do esforço, os requisitos e critérios de avaliação para padronização analisam (i) a possibilidade de implementação de cada esquema em diversas plataformas; (ii) o desempenho do esquema; (iii) a publicação do esquema e disponibilização gratuita do material publicado; (iv) a presença de evidência teórica e empírica, acompanhado de indicações de segurança contra ataques convencionais e pós-quânticos; por fim, (v) a descrição de parâmetros que caracterizam diversos níveis de segurança.

Na terceira rodada do esforço de padronização, entre os esquemas finalistas voltados para assinaturas digitais, encontravam-se um que utiliza uma abordagem baseada em sistemas de equações quadráticas com várias variáveis, Rainbow, e dois com abordagens baseadas em reticulados, FALCON e Dilithium. Nesta rodada, uma propriedade importante a ser avaliada é a segurança dos esquemas contra ataques de *side-channel* e a exploração de outros meios de ataques não percebidos até o momento. Também leva-se em consideração a dificuldade de implementação, como erros de ponto flutuante, ruídos de transmissão, entre outros. Recentemente (GORJAN et al., 2022), os dois esquemas de assinatura digital baseado em reticulados foram selecionados para serem normatizados.

Esquemas baseados em reticulados são construídos com base em conjecturas sobre a dificuldade de problemas de reticulados, como por exemplo os problemas LWE (*Learning with*

Errors) e SIS (*Short Integer Solution*), ou suas variações. Estas conjecturas, de acordo com (REGEV, 2005), são suportadas pela ausência de uma solução quântica de tempo polinomial encontrada para os problemas. Um exemplo de esquema construído dessa forma é o Dilithium (DUCAS et al., 2021), que faz uso da heurística *Fiat-Shamir with aborts* para obter um esquema de assinatura digital. Esta heurística se comporta de forma similar à heurística de construção de esquemas de assinatura Fiat-Shamir, porém, transferida para o contexto de reticulados. Neste contexto, existe a possibilidade de abortar-se o processo de assinatura ao identificar o vazamento de informações da chave privada. A segurança do esquema Dilithium é formalizada pelo QROM (*quantum random oracle model*) e baseada na dificuldade dos problemas MLWE (*Modular LWE*) e MSIS (*Modular SIS*). Por isso, similar a outros esquemas criptográficos, segundo (ZHANDRY, 2019), presume-se que este esquema, seguro no QROM, continue seguro no "mundo real".

Em relação ao seu concorrente, o FALCON apresenta melhores resultados em geral, relacionado a tamanho de chave pública, chave privada e assinatura. Contudo, o esquema possui diversas dificuldades de implementação. Um exemplo disso está relacionado com a técnica de geração de números aleatórios utilizada. Por este motivo, o Dilithium propõe a padronização de um esquema mais fácil de ser implementado e que ainda apresenta bons resultados de desempenho, tamanho de chave e assinatura; e fornece evidências de segurança ainda não questionadas. Por estes motivos o Dilithium foi escolhido como tópico de estudo deste trabalho.

Família	Rodadas do NIST			
	Primeira	Segunda	Terceira	Padronização
Esquemas baseados em reticulados	5	3	2	2
Esquemas baseados em sistemas de equações quadráticas multivariadas	7	4	2	0
Esquema baseados em resumos criptográficos	3	2	2	1
Esquemas baseados em código de correção de erro	2	0	0	0
Outras famílias	2	0	0	0

Tabela 1 – Classificação de esquemas de assinaturas aceitos nas três primeiras rodada do NIST e selecionados para padronização.

1.1 OBJETIVOS

Objetivo geral: o objetivo geral do trabalho é fornecer um estudo sobre esquemas criptográficos baseados em reticulados, tendo em vista esquemas de assinatura digital. Dessa forma, a fundamentação teórica do trabalho decorrerá da definição de reticulado e alguns dos problemas difíceis mais importantes que são utilizados na construção de esquemas baseados nesta estrutura algébrica. Por este motivo, é importante demonstrar o empenho realizado sobre esses problemas, que vem muito antes de suas aplicações na criptografia pós-quântica e

levou a fortes conjecturas sobre suas dificuldades. Uma vez que explicado a fundamentação de reticulados e problemas relacionados, prossegue a associação destes problemas difíceis com a construção de primitivas criptográficas, como funções de resumo criptográfico resistentes a colisões, esquemas de identificação e esquemas de assinatura. Como resultado, o estudo deve ser capaz de introduzir com uma descrição detalhada a criptografia baseada em reticulados para alunos próximos de obter um nível de graduação ou alunos recentemente graduados.

Objetivos específicos: como o Dilithium é um esquema promissor e selecionado para ser padronizado no esforço criptográfico do NIST, o trabalho tem como objetivo específico mostrar a aplicação dos princípios modernos da criptografia pós-quântica baseada em reticulados na construção deste esquema. Ou seja, apresentar a matemática computacional por trás da geração de um par de chaves assimétricas, o processo de assinar uma mensagem com uma chave privada, o processo de verificar uma assinatura com uma chave pública. Dessa forma, por meio das primitivas criptográficas apresentadas, procura-se também esclarecer o motivo por trás da credibilidade do esquema ser seguro, isto significa, ser computacionalmente impossível descobrir a chave privada a partir da chave pública e assinaturas; e ser computacionalmente impossível construir uma assinatura válida sem a chave privada. Além do funcionamento, o trabalho deve demonstrar os obstáculos que o esquema enfrenta relacionado ao tamanho das chaves e assinaturas; ao desempenho computacional; e ao passível vazamento de informações da chave privada ao assinar uma mensagem, originado pela heurística *Fiat-Shamir with aborts*. No trabalho, a teoria do esquema de assinatura será acompanhada por uma implementação que tornará concreto os conceitos teóricos, mostrando uma implementação dos algoritmos de assinatura, verificação e geração de par de chaves; contendo as principais técnicas intermediárias que visam otimizar o tempo de execução e o espaço de memória utilizado. Contudo, diferente da implementação fornecida pelos autores da publicação do esquema, não executará em tempo constante. Está característica torna a implementação deste trabalho sujeita a ataques de *side-channel* de análise de tempo e não deve ser utilizada em ambientes de produção.

Escopo do trabalho: o trabalho engloba os principais fundamentos da criptografia baseada em reticulado. Apesar do escopo do trabalho ser exclusivo a esquemas de assinatura, muitas das primitivas apresentadas permitem a compreensão de outras aplicações da criptografia baseada em reticulado. Em relação aos esquemas apresentados, o escopo é definido em volta dos esquemas de assinatura participantes do esforço criptográfico do NIST, principalmente nos esquemas de assinatura padronizados após o final da terceira rodada. Mesmo assim, apesar de mencionar características de outros esquemas, o aprofundamento teórico ocorrerá exclusivamente sobre o esquema Dilithium, selecionado pelos motivos apresentados na introdução do trabalho. Dito isso, o escopo do trabalho não se aplica à implementação de infraestruturas organizacionais responsáveis pela geração de chaves ou revogação de certificados.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 ESTRUTURAS ALGÉBRICAS

Esta seção descreve as estruturas algébricas relevantes para o entendimento do trabalho. Assim, as principais estruturas descritas aqui são anéis quocientes (Definição 2.1.12), corpos finitos (Definição 2.1.15) e espaço euclideano (Definição 2.1.17). Espaços euclideanos são importantes para definir reticulados. Anéis quocientes são utilizados na descrição de reticulados cíclicos. Por fim, corpos finitos, não só são importantes para definir espaços vetoriais, são partes fundamentais de vários esquemas criptográficos. Além destas três principais estruturas algébricas, são definidas demais estruturas que auxiliam em suas definições. Inicia-se, abaixo, a descrição de anéis quocientes com a definição de grupos.

Definição 2.1.1 (Grupo)**.** Um grupo $(G, *)$ é um conjunto de elementos G munido de uma operação binária $*$ que comporta-se de acordo com as seguintes propriedades:

- **Associatividade**: $(a * b) * c = a * (b * c)$, para todo $a, b, c \in G$.
- **Elemento Neutro** : Existe um elemento identidade à esquerda o_l e um elemento identidade à direita o_r, onde $o_l, o_r \in G$ e $o_l * a = a * o_r = a$ para todo $a \in G$. Quando $o_l = o_r$, esse elemento é chamado de o.
- **Elemento Inverso** : Para todo $a \in G$, existe um elemento inverso à esquerda a_l^{-1} e um elemento inverso à direita a_r^{-1}, onde $a_l^{-1}, a_r^{-1} \in G$ e $a_l^{-1} * a = o_l$ e $a * a_r^{-1} = o_r$. Quando $a_l^{-1} = a_r^1$, esse elemento é chamado de a^{-1}.

Se um grupo possui uma identidade à esquerda e um inverso à esquerda, isto implica que o grupo possui uma identidade à direita idêntica à sua identidade à esquerda e um inverso à esquerda idêntico ao seu inverso à esquerda. Contudo, apesar de existir essa comutatividade consequente envolvendo elementos neutros e elementos inversos, as propriedades de grupos não garantem comutatividade para quaisquer elementos de G.

Definição 2.1.2 (Grupo abeliano)**.** O grupo $(G, *)$ é um grupo *abeliano* se possui a propriedade de comutatividade para todo $a, b \in G$, ou seja, $a * b = b * a$.

Com a definição de grupos, pode-se descrever subgrupos e subgupos normais.

Definição 2.1.3 (Subgrupo)**.** Seja $(G, *)$ um grupo. O par ordenado $(H, *)$ é um *subgrupo* de G se e somente se $H \subseteq G$ e $(H, *)$ também forma um grupo.

Definição 2.1.4 (Subgrupo normal)**.** Um subgrupo $(H, *)$ de $(G, *)$ é um *subgrupo normal* se, para qualquer elemento $g \in G$, os conjuntos gH e Hg coincidem. Ou seja, $\forall g \in G$ e $\forall h \in H$, existe um $h' \in H$ onde $h * g = g * h'$. Da mesma forma, existe um $h' \in H$ onde $g * h = h' * g$.

Proposição 2.1.5. Seja $(G,*)$ um grupo abeliano e seja $(H,*)$ um subgrupo de $(G,*)$. Então $(H,*)$ é um subgrupo normal.

Demonstração. Um grupo abeliano fornece a propriedade de comutatividade. Dessa forma, conforme a definição de subgrupo normal, ao selecionarmos $h' = h$, então temos que $\forall g \in G$ e $\forall h \in H$, $g*h = h'*g = h*g$ e $h*g = g*h' = g*h$. □

A definição de coclasse encontra-se abaixo. Esta definição permite simplificar a demonstração da Proposição 2.1.5, pois, sempre que a coclasse à esquerda é igual à coclasse à direita, o subgrupo é um subgrupo normal.

Definição 2.1.6 (Coclasse)**.** Seja $(G,*)$ um grupo e $(H,*)$ um subgrupo normal de G. Para qualquer elemento $a \in G$, uma coclasse à esquerda de H é o conjunto $aH = \{a*h \mid h \in H\}$. Analogamente, uma coclasse à direita é o conjunto $Ha = \{h*a \mid h \in H\}$, onde $a \in G$.

Grupos quocientes são grupos que agregam elementos com características *semelhantes* (matematicamente) em classes. Assim, cada classe de elementos semelhantes pode ser tratada como um único elemento. Por este motivo, uma classe inteira, definida pela coclasse aH, pode ser representada por qualquer elemento singular desta classe. Por exemplo, o próprio elemento a.

Definição 2.1.7 (Grupo quociente)**.** Seja $(G,*)$ um grupo e $(H,*)$ um subgrupo normal de G. Define-se um grupo quociente G/H como toda coclasse à esquerda do subgrupo H no grupo G, isto é, $G/H = \{aH \mid a \in G\}$.

Exemplo 2.1.8. Seja $(G,+)$ um grupo, onde $G = \mathbb{Z}_6 = \{0,1,2,3,4,5\}$, o conjunto de inteiros módulo 6, e $+$ a operação de soma módulo 6. Seja $(H,+)$ um subgrupo normal de G, onde $H = \{0,2,4\}$. Segundo a definição de grupos quocientes e coclasse:

$$
\begin{aligned}
G/H = \{0H,\, 1H,\, \cancel{2H},\, \cancel{3H},\, \cancel{4H},\, \cancel{5H}\} &= \{0H,\, 1H\},\\
0H = \{0+0,\, 0+2,\, 0+4\} &= \{0,\, 2,\, 4\},\\
1H = \{1+0,\, 1+2,\, 1+4\} &= \{1,\, 3,\, 5\},\\
2H = \{2+0,\, 2+2,\, 2+4\} &= \{2,\, 4,\, 0\},\\
3H = \{3+0,\, 3+2,\, 3+4\} &= \{3,\, 5,\, 1\},\\
4H = \{4+0,\, 4+2,\, 4+4\} &= \{4,\, 0,\, 2\},\\
5H = \{5+0,\, 5+2,\, 5+4\} &= \{5,\, 1,\, 3\}.
\end{aligned}
$$

Perceba que, na construção de G/H, os resultados de várias coclasses podem ser idênticos. No exemplo anterior, $0H = 2H = 4H$ e $1H = 3H = 5H$. Portanto, consideramos que existem apenas duas coclasses.

No Exemplo 2.1.8, qualquer soma de elemento em $0H$ resulta em um elemento pertencente a $0H$ e qualquer soma de elementos em $1H$ resulta também em um elemento pertencente a $0H$. Ao prestar mais atenção, estes grupos obedecem o comportamento de paridade na soma de números módulo 6. Por exemplo, ao somar dois números pares, o resultado será um número par. Pode-se concluir que o exemplo de grupo quociente define as classes dos inteiros módulo 6, divididos em pares ou ímpares.

Definição 2.1.9 (Anel)**.** Um anel é uma tripla $(R,+,*)$, onde R é um conjunto e $+$ e $*$ são operadores binários definidos em R, satisfazendo as seguintes propriedades:

- **Grupo abeliano** : $(R,+)$ é um grupo abeliano. Ou seja, possui as propriedades de associatividade, existência e unicidade do elemento neutro, existência e unicidade de inversas para todos os elementos, e comutatividade.
- **Associatividade de** $*$: O operador binário $*$ deve ser associativo, portanto, $(a*b)*c = a*(b*c)$ para todo a, b e $c \in R$.
- **Distributividade de** $*$ **com respeito a** $+$: O operador binário $*$ deve ser distributivo com respeito a $+$, isto é, para todo $a,b,c \in R$, $a*(b+c) = (a*b)+(a*c)$ e $(a+b)*c = (a*c)+(b*c)$.

Definição 2.1.10 (Anel comutativo)**.** Um anel $(R,+,*)$ é dito comutativo se possui a propriedade de comutatividade da multiplicação. Isto é, $\forall a,b \in R$, $(a*b) \in R$.

Definição 2.1.11 (Ideal)**.** Seja $(R,+,*)$ um anel comutativo. O conjunto $I_l \subseteq R$ é um ideal à esquerda com as três propriedades:

(i) O elemento $o_l \in I_l$, onde o_l é o elemento neutro da adição à esquerda do grupo *abeliano* $(R,+)$.

(ii) $\forall a,b \in I_l$, $(a+(-b)) \in I_l$, onde $-b$ é o elemento inverso de b.

(iii) $\forall a \in R$ e $\forall b \in I_l$, $(a*b) \in I_l$.

Da mesma forma, o conjunto $I_r \subseteq R$ é um ideal a direita analogamente. Se os ideais à esquerda e à direita são iguais, então pode-se definir o ideal $I \subset R$, tal que $I = I_l = I_r$.

A descrição de ideais fornece uma ideia similar a de elementos que, juntos, são congruentes a zero. Ou seja, ao somar dois elementos *zero*, o resultado é um elemento *zero* (propriedade (i)). Ao multiplicar qualquer outro elemento do anel por um elemento zero, o resultado será um elemento zero (propriedade (ii)).

A definição de anéis quocientes se assemelha a definição de anéis, mas com o uso de grupos quocientes. Portanto, em um anel quociente, define-se dois operadores binários entre as classe de semelhança de um grupo quociente.

Definição 2.1.12 (Anel quociente)**.** Seja $(R, \bigoplus, \bigotimes)$ um anel, $I \subset R$ um ideal, $(R, \bigoplus)$ um grupo abeliano e $(I, \bigoplus)$ um subgrupo normal. O anel definido pela tripla $(R/I, +, *)$ é um anel quociente, onde R/I é um grupo quociente e os operadores binários $+$ e $*$ definem as operações binárias das classes de equivalência pertencentes a R/I.

Seja $S \subset R$ um ideal de um anel $(R, +, *)$. A notação $\langle S \rangle$ representa o menor ideal deste anel que contém S. Caso S contenha um único elemento $s \in S$, pode-se também utilizar a notação $\langle s \rangle$. Neste caso, refere-se a s como o único gerador do ideal e $\langle s \rangle = \{rs : r \in R\}$. Dessa forma, se $S = \{s_1, s_2, \ldots, s_m\}$ conter diversos elementos, o ideal possui diversos geradores e $\langle S \rangle = \langle s_1, s_2, \ldots, s_m \rangle = \{r_1 s_1 + r_2 s_2 + \cdots + r_m s_m | r_i \in R \text{ para todo } 1 \leq i \leq m\}$. Para definir anéis quocientes polinomiais, todas as notações R_s, $R/\langle s \rangle$ e R/sR possuem o mesmo significado e são viáveis de serem utilizadas.

Considere que $\mathbb{Z}[x]$ é o conjunto de todos os polinômios com coeficientes inteiros. Seja $f \in \mathbb{Z}[x]$ um polinômio. Então, $\langle f \rangle$ é o ideal que contém todos os polinômios múltiplos de f. Observando a definição de grupos quocientes, o exemplo $(\mathbb{Z}/2\mathbb{Z}, \bigoplus)$ define duas classes de inteiros, os que quando operados em módulo 2 resultam em 0 e os que resultam em 1. Semelhante a este comportamento, o grupo quociente $(\mathbb{Z}[x]/\langle f \rangle, \bigoplus)$, utilizando polinômios, define classes com base no resto da divisão por f. Assim, ao considerar o anel quociente $(\mathbb{Z}[x]/\langle f \rangle, \bigoplus, \bigotimes)$ e que cada classe de semelhança do grupo seja representada pelo seu elemento de menor grau, pode-se descrever as operações de soma e multiplicação entre as classes do anel. A soma de dois elementos neste anel é igual ao resto da divisão da soma dos dois operandos pelo polinômio f. A multiplicação de dois elementos neste anel é igual ao resto da divisão da multiplicação dos dois operandos pelo polinômio f. Note que o resultado destas operações geram o polinômio representante de menor grau de uma classe de semelhança em $\mathbb{Z}[x]/\langle f \rangle$.

$$\text{classe}_1 \bigoplus \text{classe}_2 = \text{classe}_3$$
$$p_1(x) + p_2(x) \equiv p_3(x) \pmod{f},$$

onde $p_i(x)$ é o polinômio de menor grau representante da $classe_i$. O mesmo se aplica a operação de multiplicação entre classes.

Na próxima seção serão discutidos reticulados e reticulados cíclicos. Quando se trata de reticulados cíclicos, existe o interesse especial no anel quociente $(\mathbb{Z}[x]/\langle x^n - 1 \rangle, \bigoplus, \bigotimes)$. A razão disso é que as operações de soma e multiplicação podem ser, ambas, representadas matematicamente com muita facilidade. Considere que, neste anel, as operações sejam realizadas sobre o elemento representante de menor grau de cada classe. Ao realizar a soma entre dois elementos, o resultado da soma será um polinômio de grau igual ao máximo entre o grau do primeiro e do segundo operando. Pelo fato do grau máximo se manter, sempre, menor que n, o resto da divisão deste valor com $x^n - 1$ será a própria soma dos polinômios originais. Assim, realiza-se a soma, simplesmente, pela adição dos coeficientes que multiplicam indeterminantes x de potências iguais. Em relação à multiplicação, precedente a operação de módulo, o polinômio resultante pode possuir um grau maior que o grau dos dois operandos. Assim, a operação

de módulo encontra-se necessária para que sempre obtenha-se um polinômio de menor grau de alguma classe de semelhança. Esta multiplicação modular entre dois polinômios neste anel, onde o divisor é $x^n - 1$, pode ser computada diretamente pela convolução dos coeficientes dos operandos. Sejam A e B polinômios de grau k, cada coeficiente do polinômio C, resultante da multiplicação modular, pode ser expressado da forma

$$\begin{aligned} A &= a_0 + a_1 x + \cdots + a_k x^k, \\ B &= b_0 + b_1 x + \cdots + b_k x^k, \\ C = A * B &= c_0 + c_1 x + \cdots + c_k x^k. \end{aligned}$$

Assim, a operação de convolução, que define a multiplicação de resultado C, pode ser calculado pelo seguinte somatório. Os subscritos de a e b são inteiros módulo k (o grau dos operandos).

$$c_i = \sum_{j=0}^{k} a_j * b_{i-j}, \text{ onde } i \in [0,k].$$

A seguir apresenta-se um exemplo de soma e um de multiplicação de elementos no anel $(\mathbb{Z}[x]/\langle x^4 - 1\rangle, \bigoplus, \bigotimes)$.

Exemplo 2.1.13. Soma dos polinômios $1x^3 + 3x^2 + 8x$ e $4x^3 + 4x^2 + 2$ pertencentes ao anel quociente $(\mathbb{Z}[x]/\langle x^4 - 1\rangle, \bigoplus, \bigotimes)$:

$$\begin{aligned} (1x^3 + 3x^2 + 3x) + (4x^3 + 4x^2 + 2) &= (1+4)x^3 + (3+4)x^2 + (3+0)x + (0+2) \\ &= 5x^3 + 7x^2 + 3x + 2. \end{aligned}$$

Deve-se então buscar o resto da divisão do resultado da soma com o polinômio $x^4 - 1$. Como o grau do polinômio resultante da soma não é alterado, então este resultado será o próprio resto da divisão.

$$\begin{aligned} \frac{5x^3 + 7x^2 + 3x + 2}{x^4 - 1} &= \text{ quociente } + \frac{\text{resto}}{x^4 - 1} \\ &= \quad 0 \quad + \frac{5x^3 + 7x^2 + 3x + 2}{x^4 - 1} \end{aligned}$$

Por consequência, o resultado final é:

$$(1x^3 + 3x^2 + 3x) \bigoplus (4x^3 + 4x^2 + 2) = 5x^3 + 7x^2 + 3x + 2$$

Exemplo 2.1.14. Multiplicação dos polinômios $1x^3 + 3x^2 + 8x$ e $4x^3 + 4x^2 + 2$ pertencentes ao anel quociente $(\mathbb{Z}[x]/\langle x^4 - 1\rangle, \bigoplus, \bigotimes)$:

$$\begin{aligned}(1x^3+3x^2+3x)*(4x^3+4x^2+2) &= (3x^2+3x)*(4x^3+4x^2+2)+(4x^6+4x^5+2x^3)\\ &= (3x)*(4x^3+4x^2+2)+(4x^6+4x^5+2x^3)+\\ &\quad (12x^5+12x^4+6x^2)\\ &= (12x^4+12x^3+6x)+(4x^6+4x^5+2x^3)+\\ &\quad (12x^5+12x^4+6x^2)\\ &= 4x^6+16x^5+24x^4+14x^3+6x^2+6x\end{aligned}$$

Semelhante à soma, deve-se agora obter o resto da divisão do resultado da multiplicação com o polinômio x^4-1.

$$\begin{aligned}\frac{4x^6+16x^5+24x^4+14x^3+6x^2+6x}{x^4-1} &= \text{quociente} + \frac{\text{resto}}{x^4-1}\\ &= 14x^3+10x^2+22x+24+\frac{14x^3+10x^2-22*x+24}{x^4-1}\end{aligned}$$

Por consequência, o resultado final é:

$$(1x^3+3x^2+3x)\bigotimes(4x^3+4x^2+2) = 14x^3+10x^2+22*x+24$$

Semelhante ao anel $(\mathbb{Z}[x]/\langle x^n-1\rangle, \bigoplus, \bigotimes)$, existem polinômios diferentes de x^n-1 que facilitam a multiplicação de elementos. O polinômio que tem um papel importante na multiplicação modular de polinômios no esquema de assinatura Dilithium é o x^n+1.

Outras estruturas algébricas importantes são corpos finitos. Define-se um corpo finito da seguinte forma.

Definição 2.1.15 (Corpo finito)**.** Um corpo é uma tripla $(F,+,*)$, onde F é um conjunto e $+$ e $*$ são operações binárias definidas em F satisfazendo as seguintes propriedades:

(i) $(F,+)$ é um grupo abeliano com elemento neutro o.

(ii) $(F',*)$ é grupo abeliano, onde $F' = F \setminus \{o\}$.

(iii) Para quaisquer a, b, $c \in F$, a propriedade de distributividade de $*$ com $+$ é satisfeita, isto é, $a*(b+c) = (a*b)+(a*c)$.

Se o conjunto F do corpo for finito, então o corpo é chamado de corpo finito.

Veja que da definição de grupos *abelianos* obtém-se elementos inversos da adição e multiplicação. Assim, define-se as operações de subtração e divisão nos corpos.

Finalmente, relevantes para a definição de reticulados, descreve-se espaços euclideanos. Acompanhado de espaços euclideanos, a seguir, define-se, também, espaços vetoriais, normas e módulos.

Definição 2.1.16 (Espaço vetorial)**.** Seja $\mathbb{K} = (K, +, *)$ um corpo. Um espaço vetorial $(V, \bigoplus, \bigotimes, \mathbb{K})$ é uma quádrupla onde V é um conjunto, $\bigoplus$ é um operador binário entre elementos de V e $\bigotimes$ é um operador binário entre um elemento de K e um vetor de V. Dentro deste espaço vetorial, as seguintes propriedades são obedecidas:

- **Propriedades de adição:** $(V, \bigoplus)$ deve formar um grupo *abeliano*.
- **Propriedades de multiplicação:**
  - (i) Existe a distributividade do operador $\bigotimes$ com $+$. Portanto, $\forall v \in V$ e $\forall \alpha, \beta \in K$, $(\alpha + \beta) \bigotimes v = (\alpha \bigotimes v) \bigoplus (\beta \bigotimes v)$.
  - (ii) Existe a distributividade do operador $\bigotimes$ com o operador $\bigoplus$. Dessa forma, $\forall v_1, v_2 \in V$ e $\forall \alpha \in K$, $\alpha \bigotimes (v_1 \bigoplus v_2) = (\alpha \bigotimes v_1) \bigoplus (\alpha \bigotimes v_2)$.
  - (iii) Os operadores $\bigotimes$ e $*$ contêm a propriedade de associatividade, ou seja, $\forall v \in V$ e $\forall \alpha, \beta \in K$, $\alpha \bigotimes (\beta \bigotimes v) = (\alpha * \beta) \bigotimes v$.
  - (iv) Existe um elemento neutro $o_K \in K$ onde $o_K \bigotimes v = v$ para todo $v \in V$.

No restante do trabalho, considera-se que um vetor qualquer $v \in \mathbb{R}^n$ possa ser escrito como $v = (v_1, v_2, \ldots, v_n)$. Considere $B = \{b_1, b_2, ..., b_m\}$ uma base composta por elementos pertencentes a algum espaço vetorial. Por ser uma base, B é formado por vetores linearmente independentes. Além disso, considere span(B) uma combinação linear de B com coeficientes inteiros. Por fim, considere que qualquer variável representada por um caractere em negrito, por exemplo $\boldsymbol{v}$, se refere a um vetor de um espaço vetorial ou, futuramente, a um polinômio equivalente a um vetor no reticulado (definido na seção 2.2).

Definição 2.1.17 (Espaço euclideano)**.** Seja $n \in \mathbb{Z}$. O espaço euclideano de dimensão n é o espaço vetorial $(\mathbb{R}^n, \bigotimes, \bigoplus, \mathbb{R})$. Os operadores binários de adição e multiplicação são definidos da seguinte forma:

- **Operação de adição:** $\boldsymbol{a} \bigoplus \boldsymbol{b} = (a_1 + b_1, a_2 + b_2, \ldots, a_n + b_n)$.
- **Operação de multiplicação:** $\alpha \bigotimes \boldsymbol{a} = (\alpha * a_1, \alpha * a_2, \ldots, \alpha * a_n)$.

Definição 2.1.18 (Norma)**.** Seja V o conjunto de um espaço vetorial e $\boldsymbol{v} \in V$ um vetor qualquer. A norma de $\boldsymbol{v}$, denotada por $\|\boldsymbol{v}\|$, é uma função $f : V \to \mathbb{R}^+$ que representa o comprimento de $\boldsymbol{v}$ e respeita as seguintes propriedades:

(i) $\|\alpha \boldsymbol{v}\| = |\alpha| * \|\boldsymbol{v}\|$ para todo $\boldsymbol{v} \in V$;

(ii) $\|\boldsymbol{v}_1 + \boldsymbol{v}_2\| \leq \|\boldsymbol{v}_1\| + \|\boldsymbol{v}_2\|$ para todo $\boldsymbol{v}_1, \boldsymbol{v}_2 \in V$;

(iii) para todo $\boldsymbol{v} \in V$, $\|\boldsymbol{v}\| = 0$ se, e somente se, $\boldsymbol{v}$ é o elemento neutro do espaço vetorial V.

Definição 2.1.19 (Norma eclideana)**.** Seja V um espaço euclideano de n dimensões e um vetor $\boldsymbol{v} \in V$ qualquer. A norma euclideana de $\boldsymbol{v}$, conhecida como ℓ_2, é denotada por $\|\boldsymbol{v}\| = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^2}$.

Na álgebra, módulos são uma generalização de espaços vetoriais e seu conceito será importante na definição de reticulados modulares. Para definir um módulo, basta substituir na definição de espaços vetoriais a estrutura algébrica de corpos por anéis comutativos. Assim, a definição de módulo é a seguinte.

Definição 2.1.20 (Módulo)**.** Seja $(R, +, *)$ um anel comutativo. Um módulo $(M, \bigoplus, \bigotimes, \mathcal{R})$, também chamado de R-módulo M, é uma quádrupla onde M é um conjunto, $\bigoplus$ é um operador binário entre elementos de M e $\bigotimes$ é um operador binário entre um elemento de R e um vetor de M. O módulo possui as mesma propriedades de espaços vetoriais, contudo, os escalares, ao invés de serem corpos, são elementos do anel R.

Aqui, define-se o anel do módulo como um anel comutativo. Assim, não precisa-se definir R-módulo M à esquerda e R-módulo M à direita. Contudo, para certos aspectos da álgebra, esta distinção pode se tornar relevante.

2.2 RETICULADOS

Reticulados são frequentemente utilizados como um método matemático para a construção de algoritmos criptográficos resistentes a computadores quânticos. Dessa forma, se torna importante conhecer sua definição, princípios e problemas computacionais relacionados. A definição de reticulados se assemelha a de espaços vetoriais, com o uso de bases e combinações lineares. Diferente de espaços vetoriais, a combinação linear é realizada estritamente com coeficientes inteiros. Um reticulado de duas e três dimensões pode ser apresentado graficamente em um plano cartesiano, conforme, respectivamente, os exemplos da Figura 1 e da Figura 2.

Definição 2.2.1. Seja $B = \{b_1, b_2, \ldots, b_m\}$ uma base composta por elementos de um espaço euclideano de dimensão n. O *reticulado* $\mathcal{L}(B)$ formado pela base B, é o span(B). Matematicamente, isto é a combinação linear da base com coeficientes inteiros. Assim, $\boldsymbol{r} \in \mathcal{L}(B)$ se existem coeficientes $c_i \in \mathbb{Z}$ tal que

$$\boldsymbol{r} = (c_1 \bigotimes \boldsymbol{b}_1) \bigoplus (c_2 \bigotimes \boldsymbol{b}_2) \bigoplus \ldots \bigoplus (c_m \bigotimes \boldsymbol{b}_m).$$

Veja que quando $m = n$, o reticulado $\mathcal{L}(B) = \mathbb{Z}^n$. Ainda, para qualquer valor de m, $\mathcal{L}(B) \subseteq \mathbb{Z}^n$.

Ajtai (1996) apresentou em a primeira primitiva criptográfica com segurança baseada em fortes conjecturas de problemas de reticulados. Seu trabalho será referenciado em várias seções e melhor discutido ao introduzir esquemas de assinaturas baseados em reticulados na

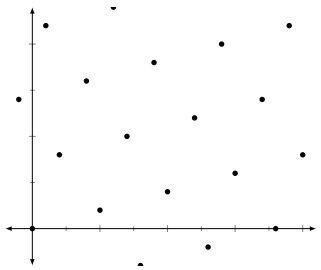

Figura 1 – Exemplo gráfico de reticulado $\mathcal{L}$ com duas dimensões

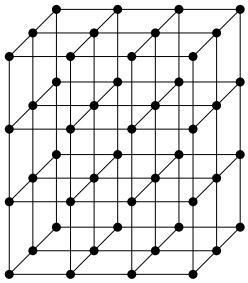

Figura 2 – Exemplo gráfico de reticulado $\mathcal{L}$ com três dimensões

Seção 3.1. Apesar da contribuição de Ajtai, o desempenho de primitivas criptográficas baseadas em reticulado era questionado quando aplicado em situações do mundo real. Dessa forma, trabalhos como (PEIKERT; ROSEN, 2006a) e (LYUBASHEVSKY; MICCIANCIO, 2006) introduziram o estudo do uso de reticulados cíclicos para construção de resumos criptográficos. Até certo ponto, estes reticulados ainda não haviam atraído o interesse de pesquisadores como os reticulados normais. Operações com reticulados cíclicos podem ser computadas de maneira eficiente e, em certas circunstâncias, sem sacrificar a segurança do esquema baseada em fortes conjecturas de problemas de reticulados. A melhor eficiência de realizar computações em reticulados cíclicos está relacionada a sua forma estruturada, que permite computar com facilidade operações sobre seus elementos por um isomorfismo com o anel quociente $(\mathbb{Z}[x]/\langle x^n-1\rangle,+,*)$. Isto é, pode-se mostrar que o anel quociente $(\mathbb{Z}[x]/\langle x^n-1\rangle,+,*)$ é isomórfico a $\mathbb{Z}^n$. Qualquer elemento de um reticulado cíclico, subconjunto de $\mathbb{Z}^n$, pode ser mapeado para uma classe de semelhança do anel quociente. Ainda, a classe de semelhança do anel pode ser representada por um de seus elementos, como o polinômio de menor grau, e operações entre as classes, como a multiplicação, são realizadas de maneira computacionalmente eficiente.

Considere que um polinômio mônico de grau n é um polinômio cujo coeficiente dominante $c_n = 1$. Além disso, considere que um polinômio irredutível é um polinômio que não tem fatores exceto os trivial.

Proposição 2.2.2. Seja $f \in \mathbb{Z}[x]$ um polinômio mônico irredutível de grau n. O anel quociente $(\mathbb{Z}[x]/\langle f\rangle,+,*)$ é isomórfico a $\mathbb{Z}^n$. Para relacionar os elementos dos dois conjuntos, basta escolher uma base $B = \{\boldsymbol{b}_0,\ldots,\boldsymbol{b}_{n-1}\} \subset \mathbb{Z}^n$. Logo, o isomorfismo é dado pelas funções σ e sua inversa σ^{-1}

$$\begin{aligned}\sigma &: \mathbb{Z}[x]/\langle F\rangle \to \mathbb{Z}^n \\ &: (a_0 * x^0 + \cdots + a_{n-1} * x^{n-1}) \to (a_0 * \boldsymbol{b}_0 + \cdots + a_{n-1} * \boldsymbol{b}_{n-1}).\end{aligned}$$

A seguir serão descritos reticulados cíclicos e reticulados ideais. A definição de reticulado ideal é uma generalização da definição de reticulado cíclico.

Definição 2.2.3. Seja $f \in \mathbb{Z}[x]$ um polinômio mônico e irredutível. Seja I um ideal do anel quociente $(\mathbb{Z}[x]/\langle f\rangle, +, *)$. Considere o isomorfismo dado por $\sigma : \mathbb{Z}[x]/\langle f\rangle \to \mathbb{Z}^n$. Define-se um reticulado ideal $\mathcal{L}(I) = \{\sigma(i) \mid i \in I\}$.

Definição 2.2.4. Um *reticulado cíclico* é semelhante a um reticulado ideal, porém, f é um polinômio mônico no formato $x^n - 1$ para algum grau n e f não é irredutível.

A definição original de reticulados cíclicos vem do conceito de rotação de vetores.

Estas definições de reticulados e a proposição de isomorfismo impactam o restante das demonstrações neste trabalho. Por exemplo, na Seção 2.2.1, que discute-se a complexidade de problemas envolvendo reticulados, usufrui-se do isomorfismo para apresentar os problemas RLWE e RSIS, que são outras versões dos problemas de reticulado *Lerning With Errors* (LWE) e *Short Integer Solution* (SIS), dentro do contexto de reticulados ideais. Assim, polinômios podem ser designados como vetores e vice-versa.

A segurança de esquemas como o Dilithium não é baseada em problemas que envolvem reticulados ideais. Os problemas supostamente difíceis que caracterizam o Dilithium envolvem uma outra estrutura algébrica, os reticulados modulares. Estudos recentes mostram que complexidade de tempo e complexidade de espaço de algoritmos que tentam resolver problemas fundamentais de reticulados são melhores quando se trata somente de reticulados ideais (LAARHOVEN; MOSCA; POL, 2015) (CRAMER et al., 2016) (CRAMER; DUCAS; WESOLOWSKI, 2016). Ainda, demonstra-se que, em teoria, certos algoritmos que solucionam problemas de reticulados ideais se tornam menos complexos com o uso de computadores quânticos. Isto é ruim para os esquemas criptográficos baseados em reticulados, pois a segurança destes esquemas depende da impraticabilidade de solucionar estes problemas. Os reticulados modulares possuem uma estrutura mais complicada, porém, permitem construir esquemas mais seguros contra computadores quânticos. Devido a maior complexidade, fornecer um estudo aprofundado sobre propriedades de reticulados modulares vai muito além do que pode ser realizado em uma única fundamentação teórica. Portanto, considere somente a seguinte definição de reticulados modulares.

Definição 2.2.5. Define-se número algébrico como a raiz de um polinômio de uma única variável e coeficientes inteiros.

Definição 2.2.6. Seja r um número algébrico, um corpo de números algébricos $\mathbb{F}[r]$ é o conjunto de todas as expressões construídas por uma sequência de adições, subtrações, multiplicações e divisões em r.

Definição 2.2.7 ((LANGLOIS; STEHLÉ, 2015))**.** Reticulados modulares correspondem a módulos gerados por um conjunto finito sobre um anel de inteiros algébricos de um corpo de números algébricos.

Um anel importante para sistemas criptográficos é o anel quociente $R = \mathbb{Z}[x]/\langle f\rangle$, onde $f \in \mathbb{Z}[x]$ é um polinômio mônico irredutível de grau n. Além de sua aplicação nos reti-

culados ideais, este anel permite a definição de módulos para os reticulados modulares. Assim, considere $q \in \mathbb{Z}$ um número primo. Pode-se obter k elementos arbitrários do anel quociente $R_q = R/qR$, onde $k \in \mathbb{Z}$. Com isto, forma-se elementos em $(R_q)^k$ que compõem reticulados modulares. Dessa forma, devido ao isomorfismo entre R_q e $\mathbb{Z}_q[x]/\langle x^n + 1\rangle$, o uso de reticulados modulares em primitivas criptográficas são beneficiados pelas mesmas técnicas de convolução de polinômios que os reticulados ideais. O comportamento das operações no anel $(R_q, +_q, *_q)$, adicionar e multiplicar polinômios, ocorre de maneira extremamente similar ao anel $(R, +, *)$. A única diferença é que a adição e o produto de coeficientes destes polinômios são resultados de uma soma ou multiplicação modular em q.

Reticulados cíclicos, reticulados ideais e reticulados modulares possuem uma grande importância na área da criptografia pós-quântica. Os reticulados ideais e cíclicos estão presentes em diversos esquemas de criptografia, como por exemplo o NTRU (HOFFSTEIN; PIPHER; SILVERMAN, 1998). O mesmo se aplica aos reticulados modulares, que estão presentes em esquemas como o Dilithium. Além da facilidade de expressar matematicamente as operações nos anéis $(R, +, *)$ e $(R_q, +_q, *_q)$, o desempenho computacional de realizar a operação de multiplicação pode ser reduzido a $O(n)$, aplicando a técnica de *Number Theoretic Transform* (NTT). Ou seja, apresenta um melhor desempenho do que os algoritmos triviais de convoluções que realizam a multiplicação de polinômios em um anel quociente. A transformação com NTT exige um processo inicial de decomposição que possui uma complexidade de $O(n \log n)$ e precisa ser revertido posteriormente. Contudo, devido ao grande número de multiplicações de polinômios no anel, presentes na multiplicação de matrizes, os resultados ainda são melhores que as técnicas triviais.

A *Number Theoretic Transform* pode ser construída de maneira semelhante a *Fast Fourier Transform* (FFT), porém, diferente da FFT, a NTT não transforma os valores para o domínio da frequência. Utiliza-se, na verdade, teoremas análogos que envolvem teorias de convoluções, permitindo realizar rápidas convoluções em sequências de inteiros. Esta transformação será melhor discutida com detalhes na Seção 2.3.

2.2.1 Problemas computacionais difíceis

O interesse no estudo de reticulados em criptografia e teoria da computação existe pela ausência de algoritmos conhecidos que solucionam, em tempo viável, problemas de grande escala que envolvem essa estrutura algébrica. A primeira parte desta seção discute os problemas SVP (*Shortest Vector Problem*), SVP_γ (*Approximate Shortest Vector Problem*) e o CVP (*Closest Vector Problem*), que são problemas fundamentais de reticulados. Enquanto isso, a segunda parte desta seção aprofunda problemas mais expostos nas construções de primitivas criptográficas, mas ainda diretamente relacionados a alguns dos três problemas fundamentais apresentados. Estes problemas são o SIS (*Shortest Integer Solution*), LWE (*Learning With Errors*) e suas variantes.

Definição 2.2.8 (*Shortest Vector Problem* (SVP))**.** Seja B uma base do reticulado $\mathcal{L}$, o SVP consiste em encontrar o menor vetor não nulo no reticulado $\mathcal{L}(B)$, ou seja, um vetor $\boldsymbol{v} \in \mathcal{L}$ tal que $\|\boldsymbol{v}\| = \lambda$ seja minimizado.

Definição 2.2.9 (*Approximate Shortest Vector Problem (SVP$_\gamma$)*)**.** Seja B uma base que forma o reticulado $\mathcal{L}(B)$. Seja λ o comprimento do vetor solução para o SVP com a base B. Para um escalar γ, o SVP$_\gamma$ consiste em encontrar qualquer vetor $\boldsymbol{v} \in \mathcal{L}(B)$ tal que $\|\boldsymbol{v}\| \neq 0$ e $\|\boldsymbol{v}\| \leq \gamma \cdot \lambda$.

Definição 2.2.10 (*Closest Vector Problem* (CVP))**.** Seja B uma base, $\mathcal{L}(B) \subseteq \mathbb{Z}^n$ um reticulado e $\boldsymbol{w} \notin \mathcal{L}(B)$ um vetor qualquer de $\mathbb{Z}^n$. O CVP consiste em encontrar um vetor $\boldsymbol{v} \in \mathcal{L}(B)$ de forma que $\|\boldsymbol{v} - \boldsymbol{w}\|$ seja o menor possível.

As soluções de instâncias destes três problemas dependem de como o cálculo de distância é realizado. Assim, este cálculo de distância é determinado pela norma e também influencia a qual classe de complexidade o problema pertence. As duas normas mais comuns e consideradas ao longo do trabalho é a ℓ_2, também conhecida como norma Euclideana (Definição 2.1.17), e a ℓ_∞. A norma ℓ_∞ é definida pelo termo do vetor de maior magnitude. No Dilithiu, conforme a Definição 4.3.5, a norma ℓ_∞ depende dos termos em sua forma centralizada.

Um ponto importante para começar uma discussão da complexidade destes problemas é a incerteza da existência de um algoritmo em tempo polinomial que solucione os problemas SVP ou SVP$_\gamma$, onde γ é um valor pequeno, para a maioria das normas. Até hoje, somente foi demonstrado que o problema SVP$_\gamma$ pertence ao conjunto de problemas NP-Difíceis para a norma ℓ_∞ (DINUR, 2002), enquanto que para outras normas se mantém uma questão aberta na área da computação. Dessa forma, vários estudos tentam levar a uma melhor compreensão da complexidade do SVP quando considerado cenários específicos (KHOT, 2004). Um exemplo é encontrado no trabalho (AJTAI, 1996), que apresenta uma contextualização da dificuldade de solucionar o SVP. No trabalho, demonstra-se uma redução não determinística dos casos mais complexos do SVP$_\gamma$ para os casos de complexidade mediana do SVP. Ou seja, se existir um algoritmo randomizado que solucione casos médios (*average case*) do SVP em tempo polinomial com uma probabilidade não negligenciável, então existirá um algoritmo randomizado que solucione os piores casos (*worst case*) do SVP$_\gamma$, com uma aproximação pequena n^c, onde $c \in \mathbb{N}$ e n é a dimensão do reticulado, em tempo polinomial com uma probabilidade não negligenciável.

Definição 2.2.11. Uma redução não determinística (*randomized reduction*) polinomial de um problema A para um problema B define que existe uma máquina de Turing determinística M que execute em tempo polinomial um algoritmo randomizado e, portanto, exista uma probabilidade não negligenciável de que a solução do problema A com a entrada x seja igual a solução do problema B com a entrada $M(x)$.

Algoritmos aleatórios resultam em soluções corretas com certa probabilidade e podem exigir um número grande de execuções para atingir certo índice de certeza sobre o resultado. Dessa forma, reduções não determinísticas, que fazem uso de algoritmos randomizados, não

possuem a mesma força que reduções determinísticas. Porém, estas reduções ainda contribuem para construção de suposições mais fortes sobre problemas de reticulados. Um exemplo disso é a construção da suposição de que os casos medianos do SVP não são fáceis de serem solucionados. Para complementar essa ideia, trabalhos como o de Micciancio (2001) utilizam reduções determinísticas que dependem de conjecturas sobre distribuições de inteiros sem fatores quadráticos para especular sobre a dificuldade do SVP_γ. Devido a relação entre os dois problemas, especular a complexidade do SVP_γ auxilia a acreditar que o SVP seja difícil de ser solucionado. Existem, ainda, conjecturas que enfatizam o SVP pertencer a classe de problemas NP-Difícil (EMDE-BOAS, 1981) para qualquer norma ℓ_p. Apesar de serem apenas conjecturas e não existir uma prova concreta disso para o SVP ou SVP_γ, acredita-se que estas suposições sejam fortes e levam a estudos sobre suas aplicações em primitivas criptografia.

Diferente dos dois problemas anteriores, o CVP é o único destes três problemas que foi provado pertencer à classe de complexidade NP-Difícil (MICCIANCIO; GOLDWASSER, 2002) para qualquer norma ℓ_p, demonstrado sobre reduções determinísticas. Apesar da classificação de complexidade do CVP, este problema ainda contém instâncias específicas de melhor caso que, quando identificados, são solucionados em tempo polinomial. A Figura 3 demonstra uma instância do problema CVP que necessita encontrar o vetor pertencente ao reticulado de base B mais próximo do vetor representado pelo ponto vermelho. Para encontrar a solução, basta construir o losango que cerca o ponto com ambos os vetores da base e selecionar o vetor representado pelo vértice do losango mais próximo do ponto vermelho. Os piores casos do CVP são encontrados quando a base que define o reticulado não é considerada uma "boa" base (*good basis*).

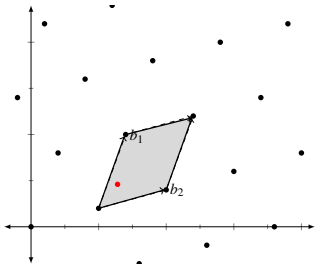

Figura 3 – Representação gráfica do reticulado $\mathcal{L}_1(B_1)$ com uma base $B_1 = \{b_1, b_2\}$ e $\mathcal{L} = \mathcal{L}_1$.

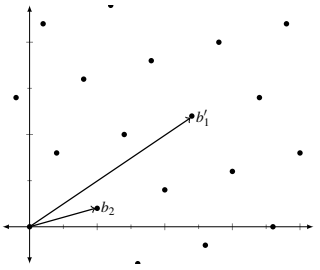

Figura 4 – Representação gráfica do reticulado $\mathcal{L}_2(B_2)$ com uma base $B_2 = \{b'_1, b_2\}$ e $\mathcal{L} = \mathcal{L}_2$.

A Figura 4 apresenta uma base diferente e um pouco *pior* do que a primeira, porém, após obter o span da base, resulta no mesmo reticulado do exemplo à esquerda. Como algumas bases são mais difíceis de se trabalhar, existem estratégias que encontram novas bases equivalentes para o mesmo reticulado, contudo, que sejam *melhores*. Define-se *boas* bases, ou bases *melhores*, como bases com vetores pequenos e *quase* ortogonais. Por este motivo, a dificuldade de decidir problemas como o CVP está relacionado com a dificuldade de encontrar bases menores e mais ortogonais. O motivo da definição exigir bases quase ortogonais é porque nem todo

reticulado pode ser formado por bases perfeitamente ortogonais.

O problema de encontrar bases pequenas está relacionado com o SVP_γ, pois, uma vez que são obtidos vetores pequenos do reticulado, estes vetores podem ser utilizados para montar bases pequenas. Um algoritmo utilizado para redução de base é o LLL. Este algoritmo foi apresentado em (LENSTRA; LENSTRA; LOVÁSZ, 1982) e resolve certas instâncias do SBP_γ (*Approximate Shortest Basis Problem*), apresentado a seguir.

Definição 2.2.12 (Approximate Shortest Basis Problem (SBP_γ))**.** Seja B uma base que forme o reticulado $\mathcal{L}(B)$. O SBP_γ consiste em encontrar uma base B' que forme $\mathcal{L}(B') = \mathcal{L}(B)$ onde o comprimento do maior vetor em B' seja no máximo γ vezes o comprimento do maior vetor pertencente a menor base que forme $\mathcal{L}(B)$.

O algoritmo LLL possui a seguinte proposição sobre sua complexidade.

Proposição 2.2.13. Seja $B = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n\}$ uma base de dimensão n. Seja $B_{\max} \in \mathbb{R}$, sendo $B_{\max} \geq 2$, de forma que, $\forall i \in [1,n], \|\boldsymbol{b}_i\|^2 \leq B_{max}$. O número de operações aritméticas realizadas nesta instância do algoritmo LLL de redução de base é $O(n^4 \log B_{max})$ e os inteiros representados possuem um comprimento binário de no máximo $O(n \log B_{max})$ bits.

Portanto, o algoritmo LLL pode ser executado em tempo polinomial em n. Contudo, o algoritmo somente encontra bases a partir de aproximações do menor vetor existente no reticulado. O fator de aproximação γ deste algoritmo obedece o formato 2^{cn}, onde n é o número de dimensões do reticulado. Assim, para grandes dimensões, as bases encontradas podem não ser tão pequenas quanto possíveis. Apesar da aproximação do algoritmo piorar para reticulados de várias dimensões, o LLL levou à criação de novos algoritmos que tentam solucionar o CVP. Um destes algoritmos é conhecido como *Nearest Plane* (BABAI, 1986) e utiliza o LLL para encontrar uma solução aproximada do CVP.

Embora existam várias estratégias que tentam aproximar soluções para estes problemas, ainda se mantêm fortes suposições de complexidade sobre problemas como SVP e SVP_γ. Em relação ao CVP, que é um problema provadamente difícil, o algoritmo LLL auxilia a encontrar suas soluções aproximadas, porém, além de sozinho não ser o suficiente para solucioná-lo, com grandes dimensões, realiza reduções de bases com piores aproximações. Por consequência, o entendimento que pode ser obtido dos estudos relacionados a problemas de reticulado é que existe uma credibilidade de suas complexidades e isso levou a pesquisas sobre esquemas criptográficos com provas de segurança baseada nestas dificuldades.

A criptografia baseada em reticulados possui como base de segurança problemas difíceis de reticulados, como os problemas apresentados anteriormente. Contudo, quando observados algoritmos criptográficos pós-quânticos atuais, os problemas comumente mencionados são outros, como o LWE (*Learning with Errors*) e SIS (*Shortest Integer Solution*). Os problemas LWE e SIS se tornaram ótimas ferramentas para a construção de primitivas criptográficas. Pode-se mostrar que são tão difíceis quanto os problemas apresentados anteriormente. Ademais, suas variações com reticulados cíclicos, RLWE e RSIS, permitem esquemas baseados na

dificuldade de solucioná-los utilizarem uma estrutura algébrica isomórfica ao espaço euclideano de n dimensões. Esta estrutura possui operações de melhor desempenho computacional.

Considere que χ_σ é a distribuição Gaussiana com média zero e desvio padrão σ. Neste trabalho será comentado somente a versão de busca do LWE (*Search LWE*).

Definição 2.2.14 (*Learning with Errors (LWE)*)**.** Sejam m,n dois números inteiros. Seja $A \in ((\mathbb{Z}_q)^n)^m$ uma sequência de m vetores escolhidos aleatoriamente de maneira uniforme. Seja $\boldsymbol{e} \in (\mathbb{Z}_q)^m$ um vetor de erros desconhecidos, escolhidos de maneira aleatória pela distribuição χ_σ. Seja $\boldsymbol{b} \in (\mathbb{Z}_q)^m$ um vetor resultante da operação $A * \boldsymbol{s} + \boldsymbol{e}$, onde $\boldsymbol{s} \in (\mathbb{Z}_q)^n$. Considere que $\boldsymbol{a} * \boldsymbol{s} = \langle \boldsymbol{a}, \boldsymbol{s} \rangle$ para todo vetor $\boldsymbol{a}$ da sequência A e que $\langle \boldsymbol{a}, \boldsymbol{s} \rangle \in \mathbb{Z}_q$ seja o produto interno dos dois vetores. O LWE consiste em encontrar, com alta probabilidade, o $\boldsymbol{s}$ que define a igualdade $\boldsymbol{b} = A * \boldsymbol{s} + \boldsymbol{e}$.

Definição 2.2.15 (*Ring Learning with Errors (RLWE)*)**.** Sejam m e n dois números inteiros. Seja $R_q = \mathbb{Z}_q[x]/\langle x^n + 1 \rangle$ e $(R_q, +, *)$ um anel quociente. Seja $A \in (R_q)^m$ uma sequência de m vetores escolhidos aleatoriamente de maneira uniforme. Seja $\boldsymbol{e} \in (R_q)^m$ uma sequência de m vetores de erros desconhecidos, escolhidos de maneira aleatória pela distribuição χ_σ. Seja $\boldsymbol{b} \in (R_q)^m$ uma sequência de m vetores resultantes da operação $A * \boldsymbol{s} + \boldsymbol{e}$, onde $\boldsymbol{s} \in R_q$. Considere que $\boldsymbol{a} * \boldsymbol{s}$ é a multiplicação definida no anel quociente, para todo vetor $\boldsymbol{a}$ na sequência A. O RLWE consiste em encontrar, com alta probabilidade, o $\boldsymbol{s}$ que define a igualdade $\boldsymbol{b} = A * \boldsymbol{s} + \boldsymbol{e}$.

Definição 2.2.16 (*Module Learning with Errors (MLWE)*)**.** Sejam d,m,n três números inteiros. Seja $R_q = \mathbb{Z}_q[x]/\langle x^n + 1 \rangle$ e $(R_q, +, *)$ um anel quociente. Seja $A \in ((R_q)^d)^m$ uma sequência de m vetores escolhidos aleatoriamente de maneira uniforme. Seja $\boldsymbol{e} \in (R_q)^m$ um vetor de erros desconhecidos, escolhidos de maneira aleatório pela distribuição χ_σ. Seja $\boldsymbol{b} \in (R_q)^m$ uma sequência de m vetores resultante da operação $A * \boldsymbol{s} + \boldsymbol{e}$, onde $\boldsymbol{s} \in (R_q)^d$. Considere que para cada vetor $\boldsymbol{a}$ na sequência A, $\boldsymbol{a} * \boldsymbol{s} = \sum \boldsymbol{a'} * \boldsymbol{s'}$, para cada um dos d termos $\boldsymbol{a'}, \boldsymbol{s'}$, respectivamente, de $\boldsymbol{a}$ e $\boldsymbol{s}$. O MLWE consiste em encontrar, com alta probabilidade, o s que define a igualdade $\boldsymbol{b} = A * \boldsymbol{s} + \boldsymbol{e}$.

O problema LWE envolve encontrar uma função que melhor mapeia um conjunto de entradas para um conjunto de saídas, assumindo a existência de pequenos erros aleatórios. Sua complexidade está relacionada com a dificuldade de solucionar equações lineares com pequenos erros. Isto pode ser facilmente observado na Definição 2.2.14. Os coeficientes (fixos) da equação são identificados pelos valores em A e as variáveis pela solução $\boldsymbol{s}$. Diferente do LWE, quando se trata de solucionar equações lineares triviais, sem erros, pode-se facilmente utilizar a técnica de eliminação Gaussiana. Mesmo que a técnica não seja aplicável ao LWE, ainda existem estratégias que tentam solucionar o problema. Como o LWE exige encontrar a função mais provável (formada por $\boldsymbol{s}$) que, considerando a distribuição χ_σ e a sequência de valores A, obtém-se a sequência de valores de $\boldsymbol{b}$, não basta somente solucionar as equações para erros quaisquer. Porém, quando o número de equações é, pelo menos, próximo de n ($m \geq n$), torna-se provável que exista um único $\boldsymbol{s}$ que mapeie apropriadamente a entrada à saída. Por

isso, é praticável procurar entre todos os vetores de $(\mathbb{Z}_q)^n$ por um vetor que seja solução das equações lineares com pequenos erros. Qualquer que seja o vetor encontrado, que resolva as equações, possivelmente formará, com significativa probabilidade, a solução do LWE. A complexidade dessa estratégia é $2^{O(n\log n)}$, mas somente pode ser aplicada com um número definido de equações. Existem outros algoritmos com melhores resultados, cada um com suas restrições de parâmetros. O melhor algoritmo que soluciona o LWE exige uma quantidade polinomial de equações e possui uma complexidade de $2^{O(n)}$ (HEROLD; KIRSHANOVA; MAY, 2018). Semelhante ao SVP, a dificuldade do LWE é melhor compreendida somente em cenários específicos. Em certos cenários, acredita-se que o problema LWE é tão complexo quanto os casos mais difíceis de problemas de reticulados e difícil de ser solucionado por computadores quânticos. A construção de fortes suposições, nestes cenários, sobre a dificuldade do problema LWE, pode ser feita de diversas formas. A forma apresentada relaciona o LWE com outro problema de reticulado, *Bounded Distance-Decoding* (BDD_α). O problema BDD_α, descrito a seguir, é bem similar ao problema CVP.

Definição 2.2.17 (*Bounded Distance-Decoding* (BDD_α))**.** Sejam $\alpha \in \mathbb{R}$, B uma base, $\mathcal{L}(B) \subset \mathbb{Z}^n$ um reticulado e $\boldsymbol{w} \notin \mathcal{L}(B)$ um vetor de $\mathbb{Z}^n$. Garantido a existência de pelo menos um vetor, o BDD_α consiste em encontrar qualquer vetor $\boldsymbol{v} \in \mathcal{L}(B)$ de forma que $\|\boldsymbol{v} - \boldsymbol{w}\| \leq \alpha$.

O problema BDD_α se assemelha bastante ao CVP, pois quando o valor de α é pequeno o suficiente, existirá uma solução única dentro da distância definida. Por consequência, o resultado obtido será uma solução para instâncias similares do CVP, formados pelo mesmo ponto alvo e reticulado. Outro detalhe sobre o BDD_α é a promessa de existência de solução. Algoritmos que solucionam o BDD_α podem assumir que o reticulado contém pelo menos um vetor dentro da região de raio α, em volta de $\boldsymbol{w}$. Apesar do problema se demonstrar mais fácil que o CVP, diversos estudos apresentam que solucionar o BDD_α se mantém difícil, como (LYUBASHEVSKY; MICCIANCIO, 2009), (LIU; LYUBASHEVSKY; MICCIANCIO, 2006) e (BENNETT; PEIKERT, 2020). Assumindo essa dificuldade, conjectura-se que o problema LWE também seja difícil. A prova mostra que, para certos cenários, oráculos do LWE permitem solucionar instâncias complexas do BDD_α. Ou seja, se o problema LWE pode ser solucionado eficientemente, nestes cenários, pode-se solucionar eficientemente as mais difíceis instâncias do BDD_α.

Definição 2.2.18 (*Gaussian Sampling*)**.** Seja $\mathcal{L}$ um reticulado e r um parâmetro da distribuição Gaussiana. A amostragem discreta de vetores em uma distribuição normal (*Gaussian Sampling*) consiste em obter um vetor $\boldsymbol{v} \in \mathcal{L} \subseteq \mathbb{Z}^n$ de forma que, qualquer vetor $\boldsymbol{x} \in \mathcal{L}$, tenha uma probabilidade $D_r(\boldsymbol{x}) = e^{\|\boldsymbol{x}/r\|^2}$ de ser escolhido.

Vale notar que, na amostragem discreta de vetores em uma distribuição normal, para um r não tão pequeno, o módulo dos vetores coletados é aproximadamente $r\sqrt{n}$.

Teorema 2.2.19. [Regev (2010)] Seja D_r uma distribuição Gaussiana centrada em zero com parâmetro r. Se existe um algoritmo eficiente que resolva LWE, então, ao obter um número

polinomial em n de amostras da distribuição D_r, pode-se resolver eficientemente o problema $\text{BDD}_{2\sqrt{n}/r}$ para qualquer reticulado $\mathcal{L}$. Em outras palavras, se for possível aproximar a função em LWE, então, é possível resolver $\text{BDD}_{2\sqrt{n}/r}$.

A prova do Teorema 2.2.19 é bastante complexa e pode ser consultada no trabalho (REGEV, 2010, pg. 7).

Esta demonstração se aplica somente para cenários em que o vetor de erros $\boldsymbol{e}$ é amostrado de uma distribuição Gaussiana. Descobrir com quais distribuições estatísticas o erro utilizado no LWE e suas variações se mantêm difíceis é um problema em aberto na área da computação. Contudo, trabalhos como (CABARCAS; GÖPFERT; WEIDEN, 2014) mostram que, se o erro for amostrado por uma distribuição uniforme, então o problema se mantém difícil com base na dificuldade de solucionar o SBP_γ.

Definição 2.2.20 (*Shortest Integer Solution (SIS)*)**.** Sejam m,n dois números inteiros e $\beta \in \mathbb{R}$. Seja $A \in ((\mathbb{Z}_q)^n)^m$ uma sequência de m vetores obtidos de uma distribuição aleatória e uniforme. O SIS consiste em encontrar um vetor não nulo $\boldsymbol{z} \in \mathbb{Z}^m$ de modo que $\|\boldsymbol{z}\| \leq \beta$ e a multiplicação $A * \boldsymbol{z} \in (\mathbb{Z}_q)^n$ é igual ao vetor nulo.

Definição 2.2.21 (*Ring Shortest Integer Solution (RSIS)*)**.** Sejam d,m,n três números inteiros. Seja $R_q = \mathbb{Z}_q[x]/\langle x^n+1\rangle$ e $(R_q,+,*)$ um anel quociente. Seja $\beta \in \mathbb{R}$ e $A \in (R_q)^m$ uma sequência de m vetores obtidos de uma distribuição aleatória e uniforme. O RSIS consiste em encontrar um vetor não nulo $\boldsymbol{z} \in (R_q)^m$ de modo que $\|\boldsymbol{z}\| \leq \beta$ e a multiplicação $A * \boldsymbol{z} \in R_q$ é igual ao vetor nulo.

Definição 2.2.22 (*Module Integer Solution (MSIS)*)**.** Sejam d,m,n três números inteiros. Seja $R_q = \mathbb{Z}_q[x]/\langle x^n+1\rangle$ e $(R_q,+,*)$ um anel quociente. Seja $\beta \in \mathbb{R}$ e $A \in ((R_q)^d)^m$ uma sequência de m vetores obtidos de uma distribuição aleatória e uniforme. O MSIS consiste em encontrar um vetor não nulo $\boldsymbol{z} \in (R_q)^m$ de modo que $\|\boldsymbol{z}\| \leq \beta$ e a multiplicação $A * \boldsymbol{z} \in (R_q)^d$ é igual ao vetor nulo.

Suposições da dificuldade de solucionar certos cenários do problema SIS podem ser construídas por uma redução ao SVP_γ. Isto é demonstrado pelo teorema apresentado no trabalho (GOLDREICH; GOLDWASSER; HALEVI, 2011). Este teorema é particularmente importante para a Seção 3.1, que mostra a propriedade de resistência à colisão de uma função de resumo criptográfico com base na dificuldade do problema SIS. Esta é uma importante demonstração que permite enxergar a relação entre solucionar problemas difíceis de reticulados e provar insegurança em primitivas criptográficas baseadas em reticulados. O teorema utiliza a redução dos piores casos do SVP_γ para os casos medianos do SVP de Ajtai.

Teorema 2.2.23. [Goldreich, Goldwasser e Halevi (2011)] Se os piores casos do SVP_γ forem difíceis de serem solucionados, então, os casos medianos do *SVP* serão difíceis de serem solucionados. Por consequência, isto implica que os casos medianos do SIS, também, serão difíceis de serem solucionados.

Vale notar que o teorema vai além da redução apresentada por Ajtai (1996), entre os problemas SVP_γ e SVP, que não mostra sua relação com o SIS.

O Teorema 2.2.23 demonstra uma redução referenciando o problema SIS e o SVP_γ. Quando se trata do problema RSIS ou MSIS, existem demonstrações similares válidas, contudo, não implica que versões do SVP_γ restritas a reticulados ideais ou modulares sejam difíceis de serem solucionados. O mesmo vale em outros teoremas. Ao construir variações e redefinir os problemas como SIS ou LWE, para que estes problemas se mantenham relevantes na área de criptografia, requer-se a construção de novas provas concretas ou fortes suposições sobre suas dificuldade. Isto é demonstrado, possivelmente, com reduções a problemas conhecidos e supostamente difíceis de reticulados. Um exemplo disso é a redefinição destes dois problemas com reticulados ideais. Por exemplo, como foi mostrado em (MICCIANCIO, 2002), utilizar um reticulado cíclico definido pelo polinômio mônico $x^n - 1$ torna, na verdade, o problema SIS mais fácil de ser solucionado. A maior facilidade se relaciona com o polinômio não ser irredutível. Diferentes deste exemplo, ainda existem estruturas que fornecem credibilidade de segurança para primitivas criptográficas por meio de problemas baseados em reticulados. Especulações sobre a dificuldade de solucionar RLWE são descritas pelo trabalho (LYUBASHEVSKY; PEIKERT; REGEV, 2010) e conjecturas da dificuldade de solucionar RSIS são exploradas pelos trabalhos (LYUBASHEVSKY; MICCIANCIO, 2006) e (PEIKERT; ROSEN, 2006b).

2.3 NUMBER THEORETIC TRANSFORM (NTT)

A técnica de *Number Theoretic Transform* (NTT) permite realizar convolução de uma sequências de termos em tempo linear. Existem diversas técnicas de convolução que se classificam como NTT. Assim, uma possível utilização do termo NTT é para referenciar a generalização da *Deterministic Fourier Transformation* para corpos finitos. Contudo, a versão mais relevante na criptografia baseada em reticulados, e que será apresentada, não transforma os valores para o domínio da frequência, mas, apropria-se de teoremas análogos que envolvem teorias de convoluções.

Em diversos esquemas de assinatura digital baseados em reticulados, necessita-se que os reticulados relacionados ao esquema sejam isomórficos a anéis quocientes polinomiais em que a multiplicação de elementos equivale à convolução dos coeficientes. Isto ocorre, pois, ao considerar o isomorfismo, o maior fator da complexidade dos algoritmos de assinatura e verificação se torna devido à diversas multiplicações de polinômios nestes anéis e podem ser otimizados com a NTT. Com a ausência de técnicas como esta, diversas primitivas criptográficas baseadas em reticulados, mesmo que suportadas por provas de segurança, se tornam ineficientes e obsoletas. Um exemplo disso é a função de resumo criptográfico de Ajtai (1996), que inspirou o estudo de reticulados no âmbito da criptografia, contudo, a função é desconsiderada ao construir esquemas criptográficos pós-quânticos modernos.

A técnica de NTT utilizada pelos esquemas criptográficos Dilithium e Kyber surge da generalização do Teorema Chinês do Resto para anéis. O Teorema Chinês do Resto permite

uma construção estruturada de isomorfismo entre anéis, sendo mais conhecido sua aplicação no anel $\mathbb{Z}_N$, para algum $N \in \mathbb{Z}$.

Proposição 2.3.1. Sejam $(R,+,*)$ e $(S,\bigoplus,\bigotimes)$ dois anéis. Um homomorfismo entre anéis, representado pela função $f : R \to S$, preserva as funções de adição e multiplicação. Ou seja, $\forall r_1, r_2 \in R$, $f(r_1 + r_2) = f(r_1) \bigoplus f(r_2)$ e $f(r_1 * r_2) = f(r_1) \bigotimes f(r_2)$.

Definição 2.3.2. Sejam $x, y \in \mathbb{Z}$. Então, x e y são coprimos se e somente se o máximo divisor comum de x e y for 1.

Teorema 2.3.3 (Teorema Chinês do Resto para $\mathbb{Z}_N$)**.** Considere $x, N, k \in \mathbb{Z}$, tal que $N = n_1 * \cdots * n_k$. Considere que para todo $i \in \mathbb{Z}$, $n_i \in \mathbb{Z}$. Se $\forall i, j \in \mathbb{Z}$, $i \neq j \to n_i$ e n_j são coprimos, então a função de mapeamento

$$f : x \pmod{N} \to (x \pmod{n_1}, \ldots, x \pmod{n_k})$$

define o isomorfismo $\mathbb{Z}_N \cong \mathbb{Z}_{n_1} \times \cdots \times \mathbb{Z}_{n_k}$ entre anéis.

Este isomorfismo, em relação a desempenho computacional, é importante quando o valor de N é muito grande. Neste caso, consegue-se construir um isomorfismo com diversos divisores n_i pequenos. Em certos casos, torna-se computacionalmente mais eficiente realizar diversas multiplicações com pares de valores $x \pmod{n_i}$ pequenos, do que uma única multiplicação com dois valores $x \pmod{N}$ grandes. Um segundo motivo para utilizar a aplicação deste teorema na criptografia é para obter implementações que executam em tempo constante. Quando uma implementação criptográfica não possui um tempo de execução constante, torna-se sujeito a ataques de tempo (*timing side-channel attack*). Nestes ataques, observadores utilizam dados como o tempo de execução para obter, por exemplo, informações da chave privada do assinante. Esta aplicação do teorema em $\mathbb{Z}_N$ é capaz de ser generalizada para outros anéis, por exemplo, os anéis quocientes isomórficos a reticulados. Assim, o teorema pode ser vantajoso para outros anéis utilizados em esquemas criptográficos.

Definição 2.3.4. Sejam f e g dois polinômios. Estes dois polinômios são coprimos se, e somente se, o máximo divisor comum de f e g for 1.

Definição 2.3.5. Sejam I e J dois ideais de um anel R. Estes ideais são *comaximais* se $I + J = R$. Define-se $I + J = \{i + j : i \in I \text{ e } j \in J\}$.

Proposição 2.3.6. Seja R um anel e $f, g \in R$. Se f e g são coprimos, então os ideais $I = \langle f \rangle$ e $J = \langle g \rangle$ são comaximais.

Teorema 2.3.7 (Teorema Chinês do Resto para anéis (DUMMIT; FOOTE, 2004))**.** Seja R um anel e $I \subseteq R$ um ideal deste anel. Considere que, para um $k \in \mathbb{Z}$, $I = I_1 \cap \cdots \cap I_k$, onde, para todo $i \in \mathbb{Z}$, $I_i \subseteq R$ é um ideal. Se $\forall i, j \in \mathbb{Z}$, $i \neq j \to I_i$ e I_j são comaximais, então a função de mapeamento

$$f : x \pmod{I} \to (x \pmod{I_1}, \ldots, x \pmod{I_k})$$

define o isomorfismo $R/I \cong (R/I_1) \times \cdots \times (R/I_k)$ entre anéis.

Um elemento do anel é levado à sua forma NTT pela função de isomorfismo. Para todo $a \in R_q$, utiliza-se a notação $\hat{a} = f(a)$ para explicitar a forma NTT de a.

Considere que o ideal I seja gerado por um polinômio de grau n grande e que a multiplicação no anel $(R/I_1) \times \cdots \times (R/I_k)$ é a multiplicação ponto-a-ponto de seus termos. Para reduzir a complexidade de multiplicar polinômios no anel R/I, pode-se escolher diversos ideais coprimos I_i gerados por polinômios de grau m pequeno. Assim, realizar diversas multiplicações clássicas (*Schoolbook multiplication*) de complexidade quadrática com operandos em R/I_i pode ser menos complexo do que realizar uma única multiplicação quadrática com operandos em R/I. Ou seja, $k * m^2$ pode ser significativamente menor que n^2.

A complexidade linear é obtida quando todos os ideais I_i são gerados por polinômios de grau 1. Neste caso, todos os polinômios nos anéis R/I_i possuem grau 0 e o produto de dois polinômios equivale a multiplicar dois números inteiros. A complexidade final de calcular o produto de elementos em $(R/I_1) \times \cdots \times (R/I_k)$, neste caso, é linear ($k * 1^2$). Portanto, para atingir a multiplicação linear de polinômios no anel do Dilithium, basta encontrar um conjunto de 256 ideais coprimos e gerados por polinômios de grau 1. Estes polinômios podem ser encontrados de maneira recursiva por uma série de fatorações. Por consequência, define-se a fatoração de polinômios (ou divisão de polinômios) da seguinte forma.

Definição 2.3.8 (Fatoração de polinômios)**.** Seja β um número inteiro, considere o polinômio $p(x) = x^n + \beta$. A fatoração do polinômio $p(x)$ consistem em encontrar dois polinômios $(x^{n/2} - \alpha)$ e $(x^{n/2} + \alpha)$, tais que $p(x) = (x^{n/2} - \alpha) * (x^{n/2} + \alpha)$. O valor de α se relaciona com a constante β da forma abaixo.

$$\begin{aligned} x^n + \beta &= (x^{n/2} - \alpha) * (x^{n/2} + \alpha), \\ x^n + \beta &= x^n + \alpha x^{n/2} - \alpha x^{n/2} - \alpha^2, \\ x^n + \beta &= x^n - \alpha^2, \\ \beta &= -\alpha^2, \\ \sqrt{-\beta} &= \alpha, \\ \sqrt{-1 * \beta} &= \alpha. \end{aligned} \tag{2.0}$$

Ao aplicar a mesma lógica ao polinômio $x^n - \beta$, tem-se que $\sqrt{\beta} = \alpha$.

A fatoração de polinômios é relevante, pois, pode-se mostrar que, ao fatorar da forma acima recursivamente o polinômio gerador de um ideal, os fatores finais geram ideais coprimos que respeitam os requisitos do Teorema do Resto Chinês. O anel utilizado no Dilithium é o $R_q = \mathbb{Z}_q[x]/\langle x^{256} + 1 \rangle$. Logo, apresenta-se a fatoração do polinômio $x^{256} + 1$.

Definição 2.3.9 (n-ésima raiz)**.** Seja $w \in \mathbb{Z}_q$, w é n-ésima raiz $\iff w^n = 1$.

Definição 2.3.10 (n-ésima raiz primitiva)**.** Uma n-ésima raiz primitiva é uma n-ésima raiz w onde $\forall 1 \leq i < n$, $w^i \neq 1$.

Proposição 2.3.11. Seja $n \in \mathbb{Z}$ uma potência de 2. Se $w \in \mathbb{Z}_q$ é uma n-ésima raiz, então $w^{1/2}$ (ou $\sqrt{w}$) é uma $(2*n)$-ésima raiz e w^2 é uma $(n/2)$-ésima raiz.

Proposição 2.3.12. Seja $n \in \mathbb{Z}$ um número par. Se $w \in \mathbb{Z}_q$ é uma n-ésima raiz, então $w^{n/2} = -1 = q-1$.

Considere, daqui para frente, que ζ_n representa uma n-ésima raiz primitiva. Veja que, ao fatorar o polinômio $x^{256}+1$, deve-se dividi-lo em outros dois polinômios, seja eles $(x^{128} + \alpha_1)$ e $(x^{128} - \alpha_1)$. Conforme definição fornecida de fatoração, obtêm-se

$$\begin{aligned} -(\alpha_1)^2 &= 1, \\ (\alpha_1)^2 &= -1, \\ (\alpha_1)^4 &= 1. \end{aligned}$$

Perceba que α_1 é uma 4-ésima raiz primitiva, denotada por ζ_4. É importante notar que, o polinômio gerador $x^{256}+1 \in \mathbb{Z}[x]$ é um polinômio irredutível e não deve existir $\alpha_1 \in \mathbb{Z}$ que defina esta fatoração. Contudo, este caso se refere ao polinômio $x^{256}+1 \in \mathbb{Z}_q[x]$, com coeficientes inteiros módulo q. Sendo assim, diferente do polinômio irredutível anterior, existe, sim, um número $\alpha_1 = \zeta_4 \in \mathbb{Z}_q$ que realize a fatoração desejada.

De maneira recursiva, os dois polinômios obtidos podem ser fatorados novamente. Semelhante a primeira fatoração, ao fatorar os polinômios $x^{128} - \zeta_4$ e $x^{128} + \zeta_4$, individualmente, de cada um deve-se obter dois novos polinômios. Tome $x^{64} - \alpha_2$ e $x^{64} + \alpha_2$ como os dois fatores de $x^{128} - \zeta 4$; e tome $x^{64} - \alpha_3$ e $x^{64} + \alpha_3$ como os dois fatores de $x^{128} + \zeta_4$. Assim, encontra-se os seguintes valores para α_2 e α_3:

$$\begin{aligned} \alpha_2 &= \sqrt{\zeta_4}, \\ \alpha_2 &= \zeta_8. \end{aligned} \qquad \text{e} \qquad \begin{aligned} \alpha_3 &= \sqrt{-1 * \zeta_4}, \\ \alpha_3 &= \sqrt{(\zeta_4)^2 * \zeta_4}, \\ \alpha_3 &= \sqrt{(\zeta_4)^3}, \\ \alpha_3 &= (\sqrt{\zeta_4})^3, \\ \alpha_3 &= (\zeta_8)^3; \end{aligned}$$

Em cada recursão, o grau dos polinômios é cortado pela metade. Dessa forma, este processo é realizado até obter polinômios de grau 1 no formato $x \pm \alpha_i$. Perceba que, $\alpha_i = (\zeta_n)^m$ é a exponenciação de alguma n-ésima raiz primitiva, para algum $m \in \mathbb{Z}$.

Quaisquer que sejam os polinômios encontrados, os ideais gerados por estes serão coprimos e o Teorema do Resto Chinês garantem a existência de um isomorfismo. Portanto,

$$R_q \cong \mathbb{Z}_q/\langle x^{128} - \zeta_4 \rangle \times \mathbb{Z}_q/\langle x^{128} + \zeta_4 \rangle \cong \mathbb{Z}_q/\langle x^{64} - \zeta_8 \rangle \times \ldots \cong [\ldots] \,. \tag{2.1}$$

Uma vez que o isomorfismo esteja definido, qualquer polinômio em R_q pode ser transformado eficientemente para sua forma NTT. O mapeamento pode ser computado espelhando a fatoração recursiva dos polinômios. Considere que, para todo $a \in R_q$, o resultado da transformação $\hat{a} = NTT(a)$ é dado pelo último isomorfismo, onde todos os geradores dos ideais são polinômios mônicos coprimos de grau 1. A função de mapeamento de cada um dos isomorfismos na Eq. 2.1 pode ser computada em O(n), onde n é o grau do polinômio gerador do ideal em R_q. Assim, um *passo* da transformação tem complexidade O(n). Além disso, como mencionado, em cada passo dos isomorfismos, o grau dos polinômios é cortado pela metade, caracterizando o número de *passos* O($\log n$). Portanto, para qualquer $a \in R_q$, a complexidade computacional de computar todos os *passos* e calcular $\hat{a} = NTT(a)$ é O($n \log n$). Define-se a seguir um *passo* da transformação NTT e sua inversa.

Definição 2.3.13 (Passo da Transformação de NTT)**.** Considere $n \in \mathbb{Z}$ uma potência de 2 maior que 1, os números $\beta, \alpha \in \mathbb{Z}_q$ e o vetor $v \in \mathbb{Z}_q/\langle x^n + \beta \rangle$. Mapear v pelo isomorfismo $\mathbb{Z}_q/\langle x^n + \beta \rangle \cong \mathbb{Z}_q/\langle x^{n/2} - \alpha \rangle \times \mathbb{Z}_q/\langle x^{n/2} + \alpha \rangle$ é equivalente a computar (u, w), onde u é igual a v módulo $(x^{n/2} - \alpha)$ e w é igual a v módulo $(x^{n/2} + \alpha)$. Como, para computar as duas operações de módulo, basta realizar substituições de $x^{n/2} = \alpha$ e $x^{n/2} = -\alpha$ em v. Isto é, computar u, corresponde a substituir x^i por $\alpha x^{i-n/2}$, para todo $i \geq n/2$. Enquanto que computar w corresponde a substituir x^j por $-\alpha x^{j-n/2}$, para todo $j \geq n/2$. Logo,

$$\begin{aligned} u &= (v_0 + \alpha v_{n/2})x^0 + (v_1 + \alpha v_{n/2+1})x^1 + \cdots + (v_{n/2-1} + \alpha v_{n-1})x^{n/2-1}, \\ w &= (v_0 - \alpha v_{n/2})x^0 - (v_1 + \alpha v_{n/2-1})x^1 + \cdots - (v_{n/2-1} + \alpha v_{n-1})x^{n/2-1}. \end{aligned}$$

Definição 2.3.14 (Passo da Transformação Inversa de NTT)**.** Considere $n \in \mathbb{Z}$ uma potência de 2 positiva, os números $\beta, \alpha \in \mathbb{Z}_q$; por fim, considere os dois vetores $u \in \mathbb{Z}_q/\langle x^n - \alpha \rangle$ e $w \in \mathbb{Z}_q/\langle x^n + \alpha \rangle$, tais que (u, w) foram obtidos por um passo da transformação de NTT. Mapear (u, w) pelo isomorfismo $\mathbb{Z}_q/\langle x^n - \alpha \rangle \times \mathbb{Z}_q/\langle x^n + \alpha \rangle \cong \mathbb{Z}_q/\langle x^{2*n} + \beta \rangle$ é equivalente a computar o único vetor $v \in \mathbb{Z}_q/\langle x^{2*n} + \beta \rangle$ em que u é igual a v módulo $x^n - \alpha$ e w é igual a v módulo $x^n + \alpha$. Isto é obtido realizando soma e subtração dos coeficientes de u e w. Veja que, para todo $i \in \mathbb{Z}$ tal que $0 \leq i < n$, $u_i = v_i + \alpha v_{i+n}$ e $w_i = v_i - \alpha v_{i+n/2}$. Logo, temos que

$$u_i + w_i = (v_i + \alpha v_{i+n}) + (v_i - \alpha v_{i+n}) = v_i + v_i = 2 * v_i$$

e

$$u_i - w_i = (v_i + \alpha v_{i+n}) - (v_i - \alpha v_{i+n}) = \alpha v_{i+n} + \alpha v_{i+n} = 2 * \alpha v_{i+n}.$$

Portanto, temos que

$$v = \frac{u_0 + w_0}{2} + \cdots + \frac{u_{n-1} + w_{n-1}}{2} + \frac{u_0 - w_0}{2 * \alpha} + \cdots + \frac{u_{n-1} - w_{n-1}}{2 * \alpha}.$$

Uma transformação, ou sua inversa, equivale a executar $\log n$ passos. Assim, consegue-se melhorar o desempenho de primitivas criptográficas baseadas em reticulados.

2.4 PRIMITIVAS CRIPTOGRÁFICAS

2.4.1 Função de resumo criptográfico

Funções de resumo criptográfico são funções de caminho único que geram uma impressão digital da entrada fornecida. Estas funções devem seguir um conjunto de propriedades que garantam este comportamento.

Definição 2.4.1. Seja $f : \mathcal{D} \to \mathcal{I}$ uma função de resumo criptográfico com domínio $\mathcal{D}$ e imagem $\mathcal{I}$. Esta função deve conter três propriedades para ser considerada segura.

- **Resistência à primeira pré-imagem**: Seja $h \in \mathcal{I}$ um resumo criptográfico. Deve ser computacionalmente impraticável encontrar $m \in \mathcal{D}$ tal que $f(m) = h$.
- **Resistência à segunda pré-imagem**: Seja $m_1 \in \mathcal{D}$. Deve ser computacionalmente impraticável encontrar algum $m_2 \in \mathcal{D}$, sendo $m_1 \neq m_2$, tal que $f(m_1) = f(m_2)$.
- **Resistência à colisão**: Deve ser computacionalmente impraticável encontrar quaisquer m_1 e m_2, sendo $m_1, m_2 \in \mathcal{D}$, tal que $f(m_1) = f(m_2)$.

A resistência à primeira pré-imagem dita que é computacionalmente inviável reverter a função de resumo criptográfico. Não deve ser viável encontrar uma função f^{-1} que reverta um resumo h. Funções de resumo criptográfico possuem o domínio $\mathcal{D}$ muito maior que o contra-domínio $\mathcal{I}$. Algumas funções de resumo criptográfico possuem até mesmo um domínio composto por infinitos valores. Por este motivo, podem existir colisões entre saídas da função. Em uma colisão, duas entradas diferentes resultam no mesmo resumo. A resistência à segunda pré-imagem e resistência à colisão impõem uma dificuldade de encontrar entradas que geram o mesmo resumo. A resistência à colisão de um função dita que deve ser computacionalmente impraticável encontrar uma colisão entre quaisquer entradas. De forma similar, a resistência à segunda pré-imagem dita o mesmo sobre a dificuldade de encontrar colisão para uma entrada específica. Além dessas propriedades, funções de resumo criptográfico possuem a propriedade

de avalanche. A propriedade de avalanche dita que pequenas modificações nos dados de entrada resultam em grandes mudanças no resumo final.

Com o passar dos anos, conforme a velocidade de computadores aumentam ou fraqueza em funções de resumo criptográfico são encontrados, a utilização de certas funções não é mais recomendada. Duas funções bastante utilizadas no passado, porém, cuja utilização não é recomendada atualmente, são o MD5 e o SHA-1. O esquema MD5 não deve ser utilizado por apresentar vulnerabilidades que permitam encontrar colisões (LIU; LIU, 2011). O SHA-1 sofre de possíveis ataques que permitem encontrar colisões de resumo criptográfico (LEURENT; PEYRIN, 2020) (LEURENT; PEYRIN, 2019).

Dessa forma, para contornar este problema, um detalhe a ser observado em diversas funções é o possível incremento do contra-domínio para dificultar o processo de encontrar colisões. Assim, funções de resumo criptográfico utilizados hoje podem ser seguras contra computadores pós-quânticos. Mesmo que computadores quânticos encontrem colisões com mais facilidade através do algoritmo de Grover (GROVER, 1996), ao aumentar o tamanho do contra-domínio, pode-se facilmente multiplicar a dificuldade do problema. A família SHA-2, que consiste de 6 funções de resumo criptográfico com diferentes tamanho, é um exemplo de funções que ajustam com facilidade o tamanho do contra-domínio.

2.4.2 Assinatura Digital

Assinaturas digitais têm o mesmo objetivo de assinaturas manuscritas, um pedaço de informação que fornece identificação do signatário e não pode ser reproduzida por outra pessoa. Assinaturas auxiliam na garantia de autenticidade e não-repúdio do assinante, e integridade do documento.

- **Autenticidade**: Identificar ou autenticar se a assinatura foi realizada por uma pessoa ou entidade específica.
- **Integridade**: Garante que o documento não foi modificado depois de ser assinado e se encontra íntegro.
- **Não-repúdio**: O assinante não pode negar a autoria da assinatura.

Esquemas de assinatura digital são esquemas baseados em cifras assimétricas, apresentadas a seguir.

Definição 2.4.2. Um sistema criptográfico que utiliza um par de chaves para cifrar e decifrar mensagens é chamado de cifra assimétrica. Mensagens cifradas por uma chave podem ser decifradas somente pela outra chave. Um par de chaves gerado é composto por uma chave pública e uma privada. A chave pública pode ser compartilhada enquanto a chave privada deve ser sempre mantida em segredo. Este sistema pode ser aplicado em protocolos para fornecer sigilo, autenticação ou não-repúdio.

Esquemas de assinaturas digitais descrevem três operações básicas: geração de um par de chaves, construção de assinatura e verificação de assinatura. Assim, o processo de construção de assinatura utilizará a chave privada do assinante, não compartilhada com ninguém, para assinar o documento. Devido ao segredo da chave privada, consegue-se autenticar o assinante no momento da verificação. De maneira semelhante, como a chave privada pertence a somente uma única entidade, obtém-se o não-repúdio, onde o assinante não pode negar a autoria da assinatura. A propriedade de integridade é obtida de diferentes formas em esquemas criptográficos.

Esquemas de assinaturas convencionais geralmente utilizam o paradigma de *hash and sign*. Neste paradigma, calcula-se o resumo criptográfico do documento a ser assinado, então, o cálculo da assinatura é feito a partir da chave privada e deste resumo e, opcionalmente, anexa-se este dado ao documento. Ao verificar a assinatura, a chave pública é aplicada ao valor de assinatura obtido e o resultado é comparado com um novo resumo do documento. Como mencionado, a autenticidade é obtida pelo uso do par de chaves assimétricos, porém, neste caso a integridade é obtida com o cálculo de resumos criptográficos. Devido às propriedades da função de resumo criptográfico, quaisquer mudanças realizadas no documento são detectadas, por mais que pequenas.

Existem outras propriedades sobre assinaturas digitais que ainda não foram mencionadas. Estas outras propriedades estão diretamente relacionados com requisitos não funcionais, que ditam a segurança, desempenho e implementação destes esquemas. Estas propriedades dizem respeito, por exemplo, a existência de um vínculo com a assinatura e o documento assinado; o desempenho de produzir chaves, assinaturas e verificá-las; por último, a segurança, que dita ser computacionalmente impraticável forjar uma assinatura digital.

Entre esquemas de assinatura digital, há esquemas que não permitem assinar múltiplos documentos com a mesma chave. A reutilização da chave privada para gerar assinaturas vaza informações o suficiente e contribui para a possibilidade de forjar uma assinatura. Por este motivo, estes esquemas específicos, conhecidos como esquemas de *one-time signature*, se restringem a produzir no máximo uma única assinatura por chave privada.

2.4.3 Esquema de Identificação

Esquemas de identificação são esquemas em que, a partir de um par de chaves assimétricas, um *verificador* (*verifier*) que possui a chave pública requisitará do *fornecedor* (*prover*) uma prova de sua posse da chave privada correspondente. Estes esquemas são compostos por um algoritmo de geração de chaves e uma descrição da comunicação entre *fornecedor* e *verificador*. O principal objetivo dos esquemas é que, em nenhuma situação, o *verificador* deve conseguir descobrir a chave privada. Dessa forma, o *fornecedor* deve apresentar uma prova de conhecimento zero (*zero-knowledge proof*), onde, apesar do compartilhamento da prova, a chave privada ainda se mantém em segredo. Assim, como descrito em (FIAT; SHAMIR, 1987), esta é a principal diferença entre esquema de identificação e esquema de autenticação. Esque-

mas de autenticação assumem a cooperação dos participantes e que a ameaça é externa. A seguir apresenta-se uma simples descrição da diferença de cada um dos esquemas.

Definição 2.4.3 (Esquema de autenticação (FIAT; SHAMIR, 1987))**.** Sejam A e B duas entidades quaisquer. Em um *esquema de autenticação*, A consegue provar sua identidade para B, porém, uma outra entidade não consegue se passar por A para B.

Definição 2.4.4 (Esquema de identificação (FIAT; SHAMIR, 1987))**.** Sejam A e B duas entidades quaisquer. Em um *esquema de identificação*, A consegue provar sua identidade para B, porém, B não consegue se passar por A para outra entidade.

Os principais esquemas de identificação, como Schnorr (SCHNORR, 1991), Okamoto (OKAMOTO, 1993), Girault (GIRAULT, 1991) e GQ (GUILLOU; QUISQUATER, 1990); fazem uso da estratégia *commit-challenge-response*, derivada do *challenge-response* presente em diversos esquemas de autenticação. Nos esquemas exemplificados, a identificação da identidade do *fornecedor*, com a chave pública, é segura pela dificuldade de resolver problemas com números inteiros, bem conhecidos na criptografia convencional. Os esquemas GQ e Okamoto dependem da dificuldade de fatorar inteiros grandes e os esquemas Schnorr e Girald dependem da dificuldade computar logaritmos discretos. As três etapas do *commit-challenge-response* são descritas a seguir.

- **Commit:** O *fornecedor* escolhe um y de maneira uniforme e aleatória e se compromete ao *verificador* que a prova de posse da sua chave privada envolverá um dado $Y := f(y)$, para alguma função f.
- **Challenge:** O *verificador* constrói uma pergunta c, também conhecida como desafio, possivelmente escolhida de maneira uniformemente aleatória e a envia para o *fornecedor*.
- **Response:** O *fornecedor* utiliza a chave privada para computar uma resposta z para a pergunta c, envolvendo y, e devolve a resposta ao *verificador*.

Após receber a resposta, o *verificador* verifica se a resposta retornada pelo *fornecedor* condiz com a pergunta c, o comprometimento Y e sua correspondente chave pública. No esquema de identificação, encontra-se necessário que o *fornecedor* escolha o valor de Y. Isto ocorre, pois, caso a resposta dependa somente da pergunta c e a mesma pergunta seja realizada duas vezes, ambas respostas seriam iguais. Assim, um adversário que observa o canal de comunicação conseguiria memorizar os pares de perguntas e respostas, e então se passar pelo proprietário da chave privada caso o *verificador* forneça um desafio usado anteriormente. Assim, o comprometimento permite que o verificador possua influência sobre a resposta que o verifica. Dessa forma, um adversário que observa o canal de comunicação entre *verificador* e *fornecedor* deverá memorizar não somente a pergunta e a resposta, mas o comprometimento. Uma vez que Y é uniformemente aleatório, existe uma probabilidade negligenciável de que Y

seja repetido pelo verificador em novas execuções do esquema e o processo de identificação não pode ser replicado por um adversário.

Uma característica importante destes esquemas é que eles podem ser convertidos em esquemas de assinatura digital. Um exemplo é a heurística Fiat-Shamir, que foi inspirada pelo esquema de assinatura presente no trabalho de Amos Fiat e Adi Shamir (FIAT; SHAMIR, 1987). Na heurística, o assinante faz o papel de *fornecedor* e *verificador* dos esquemas de identificação. Seja m a mensagem a ser assinada e H uma função de resumo criptográfico. O assinante escolhe um y de maneira uniforme e aleatória para ser utilizado na construção da assinatura e computa $Y := H(y)$. Depois, o assinante constrói a pergunta $c = H(m,Y)$ dependente da mensagem. No final, o assinante monta a resposta z envolvendo c, Y e a chave privada e devolve a assinatura descrita pela tripla (z,c,Y). O destinatário da assinatura, diferente do assinante, somente verifica a resposta do esquema de identificação. Assim, verificar a assinatura, que contém o comprometimento Y e a pergunta c, equivale a verificar a resposta z com a chave pública. Obviamente, existem variações desta apresentação, contudo, a descrição anterior expressa a ideia principal por trás da representação de um esquema de identificação como um esquema de assinatura.

Semelhante à Definição 2.4.3 de esquema de autenticação e à Definição 2.4.4 de esquema de identificação, o trabalho (FIAT; SHAMIR, 1987) traz uma descrição para esquemas de assinatura digital.

Definição 2.4.5 (Esquema de assinatura (FIAT; SHAMIR, 1987))**.** Sejam A e B duas entidades quaisquer. Em um *esquema de assinatura*, A consegue provar sua entidade para B, porém, B não consegue se passar por A nem para si mesmo.

Esta descrição é claramente bastante abstrata. Mas, a diferença de esquemas de identificação e esquemas de assinatura é que o verificador B pode ler o roteiro de uma comunicação passada com a entidade A e reproduzir a identificação válida. Nos esquemas de assinaturas, isto não ocorre. Na prática, as duas descrições de esquemas podem ser utilizados de maneira intercalável.

A propriedade de *witness-indistinguishability* está presente em diversos esquemas de identificação. Este é um tipo específico de prova de conhecimento zero e permite que o *fornecedor* prove sua identidade para o *verificador* sem que o *verificador* tome conhecimento de sua chave privada.

Definição 2.4.6. Um esquema de identificação é dito ser *witness-indistinguishable* caso, dado uma chave pública *pk* e duas possíveis chaves privadas *sk* e sk', um adversário não deve conseguir distinguir qual é a chave privada utilizada pelo *fornecedor*.

Isto implica que a resposta, ou assinatura, deve ser independente da pergunta. Assim, um adversário que só tenha conhecimento da pergunta e da chave pública que construiu a resposta de identificação, não consegue assumir informações da chave privada e, muito menos,

distingui-la das demais prováveis chaves privadas. Como será comentado nas seções seguintes, não são todos os esquemas de identificação que possuem perfeitamente a propriedade de *witness-indistinguishable* e ocorre algum vazamento de informação da chave privada. Contudo, o vazamento é considerado negligenciável e continua computacionalmente impraticável que um adversário identifique a chave privada do *fornecedor*. Estes esquemas continuam seguros. O importante é compreender que, em certas ocasiões, a propriedade *witness-indistinguishable* em teoria se difere da prática.

3 ASSINATURA DIGITAL BASEADA EM RETICULADO - PARADIGMA FIAT-SHAMIR WITH ABORTS

Entre os esforços realizados para construir esquemas seguros de assinatura digital baseados em reticulados, encontra-se, principalmente, os esquemas de assinatura com função de *alçapão*, que seguem o paradigma *hash-and-sign*, e os esquemas de assinatura com a heurística de Fiat-Shamir. Os esquemas de assinatura modelados por esta heurística são conhecidos como esquemas de assinatura Fiat-Shamir. Entre estes esquemas está presente o Dilithium.

Esta heurística, discutida brevemente na Seção 2.4.3, apresenta um modelo para construir esquemas de assinatura que originam de esquemas de identificação. Assim, uma vez que o esquema de identificação possua sua segurança baseada na dificuldade de solucionar problemas difíceis de reticulados, o esquema de assinatura obtido pela heurística de *Fiat-Shamir* se apropria destas mesmas propriedades. Com isso, o trabalho (FIAT; SHAMIR, 1987) descreve um padrão presente na construção de diversos esquemas de identificação que possuem uma segurança baseada na dificuldade de fatorar inteiros grandes ou computar logaritmo discreto. Este padrão é descrito com uma série de reduções de problemas, da seguinte forma.

Problemas difíceis $\leq$ CRHF $\leq$ *one time signatures* $\leq$ *ID Scheme* $\leq$ Assin. Fiat-Shamir.

Considere que CRHF (*collision resistant hash function*) se refere ao problema de encontrar colisões em resumos criptográficos; *one-time signatures* se refere ao problema de forjar assinaturas em um esquema de assinatura única; *ID Scheme* se refere ao problema de um adversário se identificar de maneira indevida como uma entidade cuja chave privada não o pertence; e *assin. Fiat-Shamir* se refere ao problema de forjar assinaturas em um esquema de assinatura, derivado da heurística de Fiat-Shamir.

Nestas reduções, quando se trata da criptografia convencional, os problemas difíceis são problemas de fatorar inteiros grandes ou computar logaritmos discretos. Contudo, este processo de construção de esquemas criptográficos pode ser aplicado a outros problemas e estruturas algébricas que diferem da criptografia convencional. Assim, surge-se trabalhos que exploram diferentes problemas difíceis, como os que envolvem o uso de reticulados. A aplicação de reticulados na criptografia se tornou visível após a contribuição de Ajtai (1996). Como já mencionado, o trabalho forneceu a primeira primitiva criptográfica acompanhada por uma prova de segurança baseada na dificuldade de solucionar problemas de reticulados. O trabalho propõe uma função de caminho único, que, se for possível encontrar uma primeira pré-imagem de um resumo criptográfico, então, será possível resolver algum dos piores casos do SVP_γ de maneira eficiente. Foi somente em trabalhos futuros a este que apresentou-se a demonstração de que, para certos parâmetros, esta função de caminho único é também resistente à segunda pré-imagem e resistente à colisão. Após esta contribuição, surgiram novas funções de resumos criptográficos resistentes a colisões acompanhadas por provas de segurança sobre a dificuldade deste e outros problemas de reticulados. Entre estes, encontram-se trabalhos que fazem uso de reticulados cíclicos, reticulados ideais e reticulados modulares. Estes reticulados permitem que

os esquemas sejam executados de maneira mais eficiente e sem sacrificar a segurança baseado em fortes conjecturas sobre problemas de reticulados.

Um exemplo da construção de um esquema de *one-time signature* com o uso de funções de resumo criptográfico baseados em reticulados foi proposta em (LYUBASHEVSKY; MICCIANCIO, 2008). Neste esquema, a chave privada é uma matriz de elementos de um anel. A chave pública é composta por uma função de resumo criptográfico, definida por uma matriz, e o resultado de uma expressão envolvendo a função de resumo criptográfico e a chave privada. Assim, existe uma relação matemática entre as duas chaves, e propriedades da função de resumo criptográfico, definida por uma multiplicação com matrizes, permite verificar a assinatura. Pode-se mostrar que forjar uma assinatura é tão ou mais difícil que encontrar colisões de resumo nesta função. Portanto, o esquema de assinatura única é seguro com base em problemas de reticulados. Este esquema é considerado *one-time signature*, pois, múltiplas assinaturas construídas com a mesma chave privada reduzem drasticamente a segurança da mesma.

O esquema de identificação de Lyubashevsky (2009) utiliza a mesma ideia por trás do esquema de *one-time signature*, porém, a possibilidade de vazamento de informações que reduz a segurança não é aceitável. Desta forma, o trabalho introduz a estratégia de abortar o processo de identificação ao perceber que informações sensíveis sobre a chave privada seriam vazadas. O esquema de identificação proposto, como outros, pode aproveitar da heurística de Fiat-Shamir para montar um esquema de assinatura baseado em reticulados, que permite realizar múltiplas assinaturas com o mesmo par de chaves. Cada um dos esquemas mencionados serão mais aprofundados e discutidos nas seções seguintes. A Seção 3.1 aprofundará funções de resumo criptográfico. A Seção 3.2 tratará do esquema de *one-time signature*. Por fim, a Seção 3.3 discutirá sobre o esquema de identificação e sua conversão para um esquema de assinatura.

3.1 FUNÇÕES DE RESUMO CRIPTOGRÁFICO BASEADO EM RETICULADOS

A função de caminho único de Ajtai (1996) inspirou uma família de funções de resumo criptográfico que hoje é importante para vários esquemas criptográficos. A definição da função no trabalho de Ajtai permite escolher parâmetros específicos para calcular o resumo criptográfico de uma mensagem definida por um conjunto de bits.

Definição 3.1.1 (Ajtai (1996))**.** Considere inteiros m, n, q, $R = \mathbb{Z}_q$, $D \subset R$ e uma matriz $A \in R^{n \times m}$. Uma função de resumo criptográfico f_A é definida por

$$\begin{aligned} f_A(x) : D^m &\to R^n, \\ : \boldsymbol{x} &\to A\boldsymbol{x}. \end{aligned}$$

Ao considerar $D = \{0,1\}$ e descrever uma mensagem binária de tamanho m como D^m, a função f_A passa a mapear essa mensagem para uma saída de $n\lceil \log_2 q \rceil$ bits. Dessa forma, pode-se perceber que, sempre que $m > n\lceil \log_2 q \rceil$, existirá colisão em f_A. Ainda, ao fixar o

valor de m, a função somente permite o cálculo de resumos criptográficos de mensagens com tamanho fixo, precisando redefinir a função caso a mensagem seja maior que m, por exemplo. Na Figura 5, observamos que a função f_A realiza a multiplicação $A\boldsymbol{x}$. Conforme as operações no anel $R = \mathbb{Z}_q$ e esta multiplicação, o cálculo do resumo criptográfico é expressado como $f_A(\boldsymbol{x}) = (\sum_i^m x_i a_{1,i} \pmod q, \cdots, \sum_i^m x_i a_{n,i} \pmod q)$. Assim, o resumo criptográfico obtido pertence ao anel R^n. Pelo comportamento da operação de módulo, o processo de inverter f_A para obter a mensagem original não é tão simples quanto montar a matriz inversa A^{-1} e realizar uma multiplicação $A^{-1}(A\boldsymbol{x})$.

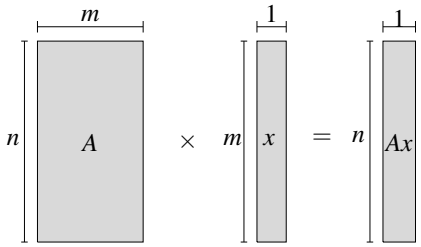

Figura 5 – Multiplicação realizada ao calcular $f_A(\boldsymbol{x})$

Foi provado no trabalho de Ajtai a redução de um problema difícil de reticulado ao problema de encontrar uma inversa para essa função. De forma que, dado uma saída $\boldsymbol{y}$, se for simples encontrar uma primeira pré-imagem desta função, então, será simples encontrar uma solução para alguma instância de pior caso de problemas difíceis de reticulados. Ajtai somente havia mostrado a dificuldade de inverter a função. Foi somente o trabalho de (GOLDREICH; GOLDWASSER; HALEVI, 2011) que apresentou, para certos parâmetros, que a função de Ajtai também é resistente à colisão e resistente à segunda pré-imagem. Como será apresentado, a segurança dessas três propriedades de funções de resumo criptográfico depende da dificuldade de solucionar os casos medianos de SIS, que relaciona-se ao problema SVP_γ e SBP_γ pelo teorema de Goldreich e Goldwasser (Teorema 2.2.23) . Abaixo segue uma descrição destes casos medianos.

Problema 3.1.2 (*Average Shortest Integer Solution (ASIS)*)**.** Define-se casos medianos de SIS como todos os casos em que $\boldsymbol{z} \in \mathbb{Z}^m$, tal que $\|\boldsymbol{z}\|_\infty \leq 1$. Ou seja $\boldsymbol{z} \in \{-1, 0, 1\}^m$.

Esta definição é o suficiente para mostrar a dificuldade de encontrar uma colisão, ou encontrar uma pré-imagem, para alguma função de resumo criptográfico de Ajtai. Seja f_A uma função de resumo criptográfico de Ajtai definida na Definição 3.1.1. É possível demonstrar que, se for difícil solucionar o problema de reticulados ASIS, então é difícil encontrar colisões em f_A.

Teorema 3.1.3 ((GOLDREICH; GOLDWASSER; HALEVI, 2011))**.** Seja f_A uma função de resumo criptográfico da Definição 3.1.1 onde $q = n^c$, para algum inteiro c, e $n \log_2 q < m < \frac{q}{2n^4}$. Se existe um algoritmo probabilístico que encontre colisões em f_A em tempo polinomial, então existirá um algoritmo probabilístico que solucione ASIS em tempo polinomial.

Demonstração. Sejam $\boldsymbol{x}, \boldsymbol{y} \in \{0,1\}^m$ dois vetores, onde $\boldsymbol{x} \neq \boldsymbol{y}$ e $f_A(\boldsymbol{x}) = f_A(\boldsymbol{y})$. Por consequência, pode-se definir $\boldsymbol{w} = (\boldsymbol{x} - \boldsymbol{y}) \in \{-1, 0, 1\}^m$ de modo que $f_A(\boldsymbol{w}) = 0$. Isto é verdade pela propriedade de distributividade das operações de matriz, onde $A(B+C) = AB + AC$ e $A\boldsymbol{w} = f_A(\boldsymbol{w}) = f_A(\boldsymbol{x} - \boldsymbol{y}) = f_A(\boldsymbol{x}) - f_A(\boldsymbol{y}) = 0$. Assim, $\boldsymbol{w}$ será uma solução para o problema ASIS. □

A prova de resistência à colisão do trabalho de (GOLDREICH; GOLDWASSER; HALEVI, 2011), descrita pela redução anterior, somente foi mostrada para certos parâmetros n, m, q da função. As restrições são $q = \mathrm{O}(n^c)$, para algum inteiro c, e $n\log_2 q < m < \frac{q}{2n^4} < \frac{q}{n^4}$. O primeiro motivo das restrições é para garantir que exista colisão na função de resumo criptográfico. Assim, os parâmetros $m > n\log_2 q$ (i) implica que o domínio seja maior que a imagem. Queremos que isto seja verdade, pois, caso o domínio for menor ou igual que a imagem, não existe garantia de que a função de resumo criptográfico possua colisões, e o ASIS possua solução. Um segundo motivo é relacionado à parametrização da dificuldade de solucionar SBP_γ, utilizado na demonstração do teorema de Goldreich e Goldwasser. Nesta demonstração, as instâncias trabalhadas de SBP_γ aceitam soluções com uma aproximação γ no formato qn^6. Caso seja reconhecido como difícil resolver este problema com uma aproximação n^k, é sempre possível escolher o valor de c igual a $k-6$. Assim, a redução do problema SBP_{n^k} ao problema ASIS implica que o problema de encontrar colisões de resumo criptográfico é difícil. A última restrição está relacionado com o número de instâncias de ASIS que precisam ser solucionadas para solucionar uma instância de SBP_γ. Conforme o teorema, o número de instâncias solucionadas de ASIS para solucionar SBP_γ é aproximadamente $n^2/\log(\frac{q/n^4}{m})$. Portanto, quando $\frac{q/n^4}{m}$ é próximo de 1, pode-se perceber que o número de instâncias que devem ser solucionadas de ASIS cresce rapidamente. Por causa disso, exige-se que $m < q/n^4$ (ii), pois $\frac{q/n^4}{q/n^4} = 1$. De (i) e (ii), obtém-se a inequação $n\log_2 q < m < \frac{q}{2n^4}$. Agora, note que esta inequação impõe uma restrição sobre o número inteiro q

$$\begin{aligned} n\log_2 q &< \frac{q}{2n^4}, \\ n^5 &< \frac{q}{2\log_2 q}. \end{aligned}$$

Apesar da existência de uma prova de segurança para a função de resumo criptográfico da Definição 3.1.1, apresentado em (AJTAI, 1996), esta função não consegue ser implementada de maneira eficiente a ponto de permitir seu uso em aplicações realistas. Contudo, o impacto desse trabalho gerou a formulação de uma família de funções de resumo criptográfico, descrita na Definição 3.1.4, que supera estes problemas de eficiência computacional. Essa família de funções faz o uso de um anel *estruturado* R, que permite computar resumos criptográficos com uma maior eficiência computacional.

Definição 3.1.4 (Família de funções de resumo criptográficos baseados em problemas de reticulados ideais)**.** Sejam m, n dois inteiros, $R = \mathbb{Z}[x]/\langle x^n + 1\rangle$ e $R_q = R/qR$ dois anéis. Seja $D \subseteq R_q$

e $A \in (R_q)^m$ uma matriz composta por m vetores. Define-se uma família de funções de resumo criptográfico com segurança baseada em problemas difíceis de reticulados da seguinte forma

$$\mathcal{H}_{R,D,m} = \left\{ \begin{array}{ll} & f_A : D^m \rightarrow R_q \\ f_A : & \\ & : x \rightarrow A\boldsymbol{x} \end{array} \right\}.$$

A família de funções de resumo criptográfico da Definição 3.1.4 se resume a realizar uma multiplicação de matriz, similar à função da Definição 3.1.1. Contudo, a matriz A da Definição 3.1.4 é composta de m elementos em R_q. Conforme foi apresentado na fundamentação teórica, a multiplicação de elementos em R_q pode ser realizada de forma eficiente utilizando técnicas como a NTT. Portanto, apesar das duas definições serem visualmente semelhantes, as funções de resumo criptográfico da família $\mathcal{H}_{R,D,m}$ não sofrem de problemas extremos relacionados à eficiência computacional. Ainda, a segurança com base na dificuldade de solucionar problemas difíceis de reticulados permanece. Pode-se mostrar que encontrar uma colisão para alguma função desta família é igual ou mais difícil que solucionar casos do problema difícil de reticulado RSIS (Definição 2.2.21). A próxima seção mostra como as funções de resumo criptográficos baseados em reticulados influenciaram a construção de esquemas de assinatura única, baseados na dificuldade de solucionar os mesmos problemas.

3.2 ESQUEMA DE ASSINATURA ÚNICA BASEADO EM RETICULADO

O trabalho de Lyubashevsky e Micciancio (2008) propõe um *framework* para a construção de um esquema de assinatura única com o uso de funções de resumo criptográfico resistentes à colisão. A proposta de um *framework* permite que a ideia fundamental do esquema possa ser aplicada em diversos contextos. Assim, o esquema é descrito em um formato genérico e com o uso de operações de estruturas algébricas conhecidas. As propriedades destas estruturas permitem descrever todos os três algoritmos do esquema. São eles, a geração de par de chaves, construção e verificação de assinaturas. Somente no final desta seção estas estruturas algébricas serão relacionadas aos reticulados estudados no trabalho, até este momento. Dessa forma, o funcionamento do esquema pode ser interpretado pelas propriedades de anéis e operações binárias entre matrizes. Logo, considere um anel R arbitrário e três inteiros k, m, n. Considere também as seguintes matrizes e vetores $\mathcal{H} \subseteq R^{n \times m}$, $\mathcal{K} \subseteq R^{m \times k}$, $\mathcal{M} \subseteq R^k$ e $\mathcal{S} \subseteq R^m$. Com isso, o Algoritmo 1 descreve a geração do par de chaves.

Algoritmo 1 Geração de chaves (*Asymptotically Efficient Lattice-Based Digital Signatures*)

Entrada: Considere que os valores são obtidos de uma distribuição uniforme.
Saída: Par de chaves $(pk = (H, \hat{K}), sk = K)$

1. $K \leftarrow \mathcal{K}$
2. $H \leftarrow \mathcal{H}$
3. $\hat{K} := HK$
4. **retorne** $(pk = (H, \hat{K}), sk = K)$

Pela descrição acima, note que, por consequência do produto HK, $\hat{K} \in R^{n \times k}$. A dimensão desta matriz será importante para ver a viabilidade da multiplicação de matrizes ao utilizar a chave pública. Assim, a construção de assinaturas com a chave privada é descrita pelo Algoritmo 2 e a verificação de assinaturas com a chave pública é descrita pelo Algoritmo 3.

Algoritmo 2 Criação de assinatura (*Asymptotically Efficient Lattice-Based Digital Signatures*)

Entrada: Seja $M \in \mathcal{M}$ a mensagem a ser assinada. Seja $sk = K \in \mathcal{K}$ a chave privada do assinante.
Saída: Uma assinatura $S \in R^{m \times 1}$
1▸ $S := KM$
2▸ **retorne** S

Algoritmo 3 Verificação de assinatura (*Asymptotically Efficient Lattice-Based Digital Signatures*)

Entrada: Seja $M \in \mathcal{M}$ uma mensagem, $pk = (H \in \mathcal{H}, \hat{K} \in R^{n \times k})$ a chave pública do assinante e $S \in R^{m \times 1}$ a assinatura que será verificada.
Saída: Resultado de verificação.
1▸ $h := HS$
2▸ $h' := \hat{K}M$
3▸ **retorne** $[h = h']$

Considere as dimensões das matrizes e vetores apresentados. Veja que a verificação da assinatura consiste em comparar o resultado da expressão HS com $\hat{K}M$. Esta lógica utilizada pelo esquema permite que a assinatura seja verificada pela propriedade de associatividade da multiplicação de matrizes e as propriedades do anel R. Deste modo, observe que

$$\begin{aligned} HS &= H(KM), && \text{pela definição de } S; \\ H(KM) &= (HK)M, && \text{por associatividade;} \\ (HK)M &= \hat{K}M, && \text{pela definição de } \hat{K}. \end{aligned}$$

Portanto, tem-se que $HS = \hat{K}M$. Assim, se a assinatura não foi editada, nem a mensagem, a igualdade é obtida e a assinatura é considerada válida.

A corretude e segurança deste esquema depende de três propriedades: a propriedade de fecho, a propriedade de resistência à colisão e a propriedade de ocultação. Assim, segue suas descrições.

- **Fecho:** Para qualquer chave privada $K \in \mathcal{K}$ e mensagem $M \in \mathcal{M}$, a construção da assinatura $S := KM$ por este assinante é fechada sobre $\mathcal{S}$, logo, sempre ocorre que $S \in \mathcal{S}$.
- **Resistência à colisão:** O esquema requer que a matriz H defina uma função com a propriedade resistência à colisão. A propriedade de resistência à colisão é a mesma descrita

na definição de funções de resumo criptográfico. Note que resistência à colisão implica em resistência à segunda pré-imagem, que implica em resistência à primeira pré-imagem, em casos não degenerados. Assim, o esquema requer que a matriz H defina uma função de resumo criptográfico.

- **Ocultação:** Esta propriedade está relacionada com a identificação da chave privada do assinante por meio de assinaturas previamente construídas e a demonstração de evidência de segurança do esquema. Seja H e K duas matrizes que constroem um par de chaves. Considere que $D_H(K,M) = \{\tilde{K} \in \mathcal{K} : HK = H\tilde{K} \text{ e } KM = \tilde{K}M\}$ seja o conjunto de todas as possíveis chaves privadas que um adversário pode inferir pela assinatura de uma mensagem M. Para algum $\varepsilon, \delta \in \mathbb{R}$, tal que $\varepsilon, \delta \leq 1$, a propriedade de ocultação diz que:

$$\Pr(\forall\, \tilde{M} \neq M,\, |D_h(K,M) \,\cap\, D_h(K,\tilde{M})| \leq \varepsilon \cdot |D_H(K,M)|) \geq \delta.$$

  Ou seja, ao construir a assinatura de uma nova mensagem, o tamanho do conjunto prévio de todas as possíveis chaves privadas é reduzido por pelo menos ε com uma probabilidade de δ. Para tornar mais concreto a exemplificação do esquema, pode-se imaginar $\varepsilon = 1/2$ e $\delta \approx 1$. Assim, a construção de uma nova assinatura possui uma probabilidade próxima de 1 de que o conjunto de possíveis chaves privadas seja reduzido pelo menos à metade.

A propriedade de Ocultação é necessária para fornecer evidência de segurança para o esquema. Considere $\varepsilon = 1/2$ e $\delta \approx 1$. Considere também que $H \in \mathcal{H}$ e $K \in \mathcal{K}$ sejam as duas matrizes que geraram o par de chaves do assinante. Agora, assuma que o proprietário deste par de chaves tenha assinado uma única mensagem M e, portanto, tenha produzido uma assinatura $S := KM$. Um adversário capaz de forjar uma assinatura de uma mensagem $\tilde{M} \in \mathcal{M}$ encontra um $\tilde{S} \in \mathcal{S}$ tal que $H\tilde{S} = \hat{K}\tilde{M}$. Logo, esta assinatura forjada é considerada válida pelo esquema. Contudo, veja que a assinatura construída sem a chave privada, em sua origem, pode ser resultado de uma multiplicação por uma chave privada $\tilde{K} \in D_H(K,M)$ correspondente a K, mas, diferente de K. Além disso, assume-se que matriz $\tilde{K}$ é obtida de forma uniforme e aleatória em $D_H(K,M)$, conforme as informações obtidas pelo adversário da primeira assinatura de M com K. Agora, perceba que existem duas possibilidades para $\tilde{S}$, $\tilde{S} = K\tilde{M}$ ou $\tilde{S} \neq K\tilde{M}$. Devido a definição de $D_H(K,M)$, temos que $\tilde{K} \in D_H(K,M)$ e $H\tilde{K} = \hat{K}$. E isto implica que

$$\tilde{S} = K\tilde{M} \longleftrightarrow \tilde{K} \in D_H(K,M) \cap D_H(K,\tilde{M}) \longleftrightarrow \tilde{K} \in D_H(K,\tilde{M}),$$
$$\tilde{S} \neq K\tilde{M} \longleftrightarrow \tilde{K} \notin D_H(K,M) \cap D_H(K,\tilde{M}) \longleftrightarrow \tilde{K} \notin D_H(K,\tilde{M}).$$

Considerando a distribuição de $\tilde{K}$ e considerando a propriedade de ocultação, para qualquer par de chaves $(K,(H,\hat{K}))$, para qualquer mensagem $M \in \mathcal{M}$ e para qualquer $\tilde{K} \in D_H(K,M)$, a probabilidade que $\tilde{K} \notin D_H(K,M)$ e $\tilde{S} \neq K\tilde{M}$ é no mínimo $\varepsilon = 1/2$.

Agora, para mostrar que o esquema é seguro, note que $\tilde{S} \neq K\tilde{M}$ implica que existem dois valores diferentes tais que, quando inseridos na função de resumo criptográfico formada

por H, obtém-se o mesmo resultado. Esta é a definição de uma colisão de resumo criptográfico. Portanto, pode-se concluir que, se existe uma probabilidade não negligenciável de que um adversário forje uma assinatura, então, existe uma probabilidade não negligenciável de que uma colisão de resumo criptográfico seja encontrada. Da mesma forma, uma vez que é conjecturado ser computacionalmente impraticável encontrar uma colisão de resumo criptográfico, tem-se que é computacionalmente impraticável forjar uma assinatura.

Em um último detalhe sobre a construção indevida de assinaturas, veja que, pelas propriedades do algoritmo de verificação de assinatura, se S é uma assinatura válida de uma mensagem M, então $c \cdot S$ é uma assinatura válida para $c \cdot M$. Exceto que a propriedade de ocultação para $\varepsilon = 1/2$ e $\delta \approx 1$ implica que M e $c \cdot M$ não podem pertencer a $\mathcal{M}$ ao mesmo tempo. Por isso, apesar da igualdade obtida pelo algoritmo de verificação, este abuso da multiplicação de matrizes não pode ocorrer.

Mesmo que a propriedade de ocultação permita mostrar a segurança contra construção indevida de assinaturas por um adversário, por consequência, a construção de diversas assinaturas com uma mesma chave privada K reduz bastante o tamanho do conjunto de possíveis chaves privadas do assinante. Logo, o esquema é um esquema de assinatura única e a construção de várias assinaturas resulta no vazamento de informações e na insegurança do esquema. Existem outros ataques que tentam demonstrar insegurança ao esquema. Contudo, pode-se facilmente perceber que são todos inviáveis pelas propriedades de funções de resumo criptográfico. Assim, encontrar a chave privada a partir da chave pública é equivalente a inverter o resumo criptográfico $\hat{K}$, para a função de resumo criptográfico definida por H. A dificuldade de substituir a mensagem assinada por uma outra mensagem é a mesma que forjar uma assinatura, pois resulta também em uma colisão de resumo criptográfico.

Como foi mencionado, o esquema é definido em forma de *framework*. Para obter uma implementação que seja baseada em problemas de reticulados, basta selecionar o anel R e parâmetros k, m e n corretamente.

Definição 3.2.1 (Aplicação do *framework* com funções de Ajtai)**.** Um esquema de assinatura única pode ser construído com as funções de resumo criptográfico de Ajtai. Estas funções possuem segurança baseada na dificuldade de solucionar o problema SIS. Considere que o esquema de assinatura única seja construído com o *framework* apresentado. Logo, define-se o esquema de assinatura única uma vez que os parâmetros sejam:

- $R = \mathcal{Z}_p$,
- $\mathcal{H} \in R^{n \times m}$,
- $\mathcal{K} = \{K \in R^{m \times k} : \|K\|_\infty \leq b\}$.
- $\mathcal{M} \subseteq \{m \in \{0,1\}^k : \|m\|_\infty = w\}$,
- $S = \{s \in R^m : \|s\|_\infty \leq wb\}$;

para quaisquer $b, k, m, n, p, w \in \mathbb{Z}$. Além disso, para estes parâmetros, o esquema possui as propriedades desejadas de fecho, resistência a colisão e ocultação, para $\varepsilon = 1/2$ e $\delta \approx 1$.

Lema 3.2.1 (Ocultação (LYUBASHEVSKY; MICCIANCIO, 2008))**.** *Considere que* $\gamma \in \mathbb{R}$ *cresça linearmente com* $wb\sqrt{nm}$*. Deste modo, para qualquer* $H \in \mathcal{H}$*, se a matriz* $K \in \mathcal{K}$ *for obtida de maneira uniforme e aleatória, então, com probabilidade mínima de* $1 - k2^{\gamma}$*, existe pelo menos uma chave* $\tilde{K} \in \mathcal{K}$ *tal que* $HK = H\tilde{K}$ *e* $K \neq \tilde{K}$ *são diferentes em todas as colunas.*

Por definição, pode-se mostrar que $\|KM\|_\infty \leq \|K\|_\infty \cdot \|M\|_1 \leq b \cdot w$ e toda assinatura $KM \in \mathcal{S}$. Logo, a propriedade de fecho vale para a Definição 3.2.1. A propriedade de resistência a colisão é demonstrada pelo Teorema 3.1.3, onde encontrar colisões nas funções de resumo criptográficos de Ajtai é igual ou mais complexo que solucionar o problema SIS de reticulados. Por fim, a propriedade de ocultação com $\varepsilon = 1/2$ e $\delta \approx 1$ é consequência do Lema 3.2.1 e a explicação disso pode ser encontrada em (LYUBASHEVSKY; MICCIANCIO, 2008). De forma análoga pode-se construir esquemas de assinatura única com reticulados ideais e modulares.

3.3 ESQUEMA DE IDENTIFICAÇÃO BASEADO EM RETICULADO E ASSINATURAS FIAT-SHAMIR WITH ABORTS

O trabalho (LYUBASHEVSKY, 2009) compreende um padrão na construção de esquemas de identificação e, da mesma forma, constrói um esquema de identificação baseado em reticulados. O esquema de identificação proposto pode ser transformado em um esquema de assinatura pela transformação de *Fiat-Shamir*. Contudo, devido a obstáculos encontrados ao transferir essa heurística para o contexto de reticulados, exige-se que a transformação seja adaptada. Esta adapatação introduz o conceito de *abort* e permite que o esquema se mantenha *witness-indistinguishable*, como desejado.

Primeiramente, observe a descrição do esquema de identificação e sua relação com o esquema de assinatura única apresentado. Os conjuntos $\mathcal{H}, \mathcal{K}, \mathcal{M}, \mathcal{D}_y$ e $\mathcal{A}$ estão descritos no texto seguinte.

Fornecedor		Verificador
$\boldsymbol{y} \leftarrow D^m, \boldsymbol{Y} := H\boldsymbol{y}$	$\xrightarrow{\boldsymbol{Y}}$	
	$\xleftarrow{M}$	$M \leftarrow \mathcal{M}$
$\boldsymbol{S} := \boldsymbol{K}M + \boldsymbol{y}$ se $\boldsymbol{S} \in \mathcal{A}$, então aborte	$\xrightarrow{\boldsymbol{S}}$	aceite sse $\boldsymbol{S} \notin \mathcal{A}$ e $H\boldsymbol{S} = \hat{\boldsymbol{K}}M + \boldsymbol{Y}$

Figura 6 – Esquema de identificação baseado em reticulado onde a chave privada do *fornecedor* é $sk = \boldsymbol{K} \leftarrow \mathcal{K}$ e a chave pública do *fornecedor* é $pk = (H \leftarrow \mathcal{H}, \hat{\boldsymbol{K}} := H\boldsymbol{K})$.

O esquema de identificação descrito na Figura 6, se comparado com a apresentação original, possui suas variáveis renomeadas. Isto ocorre, pois, deseja-se manter similar com o conteúdo apresentado nas seções anteriores deste trabalho. Assim, perceba que o esquema

é construído em cima do *framework* de assinaturas únicas da seção anterior. Os parâmetros utilizados definem um esquema baseado na dificuldade de solucionar o problema RSIS, com reticulados estruturados. Seja $n, m, \sigma, \kappa, q \in \mathbb{Z}$, onde n é uma potência de dois, $2^{\kappa} \cdot \binom{n}{\kappa} \geq 2^{160}$ e $p \approx (2\sigma+1)^m \cdot 2^{-\frac{2^{128}}{n}}$. Seja $R = \mathbb{Z}_q[x]/\langle x^n+1\rangle$ o conjunto que forma um anel. Os parâmetros que definem o esquema são

- $\mathcal{K} = \{k \in R^m : \|k\|_\infty \leq \sigma\}$,
- $\mathcal{M} = \{m \in R : \|m\|_\infty \leq \kappa\}$,
- $\mathcal{S} = \{s \in R^m : \|s\| \leq \kappa \cdot \sigma\}$,
- $\mathcal{H} = \{h \in R^m\}$,
- $D_y = \{y \in R^m : \|y\|_\infty \leq mn\sigma\kappa\}$,
- $\mathcal{A} = \{s \in R : \|s\|_\infty > mn\sigma\kappa - \sigma\kappa\}$.

A prova de evidência de segurança do esquema de identificação é dividida em partes. As partes consistem em mostrar três propriedades: (i) mostrar que o esquema é *witness-indistinguishable* (Definição 2.4.6); (ii) mostrar que existe uma probabilidade de $1-2^{-128}$ de que exista duas chaves $\boldsymbol{K}$ e $\tilde{\boldsymbol{K}} \neq \boldsymbol{K}$ tal que $H\boldsymbol{K} = H\tilde{\boldsymbol{K}}$; por fim, (iii) mostrar que se um adversário consegue *forjar uma assinatura* com uma probabilidade não negligenciável, então, pode-se encontrar uma colisão de resumo criptográfico com uma probabilidade não negligenciável.

Essencialmente, a apresentação de segurança com estas três propriedades é análoga à realizada no esquema de assinatura única anterior, mas com pequenas mudanças. Perceba que o passo (ii) é equivalente ao Lema 3.2.1 para $k = 1$ e $\gamma = 128$, só que no contexto de reticulados ideais. Da mesma maneira que antes, o objetivo é garantir a existência da propriedade de ocultação sobre as chaves privadas e encontra-se necessário para evidenciar dificuldade de *forjar uma assinatura*. Neste caso, *forjar uma assinatura* equivale a um adversário se identificar como uma entidade a qual ele não possui a chave privada, indevidamente. Por este motivo, no esquema de identificação, conforme a Definição 2.4.3, refere-se a $\boldsymbol{S}$ como resposta, não assinatura; e refere-se a M como desafio, não mensagem.

O passo (iii) demonstra que se um adversário possuir sucesso em se identificar de maneira indevida, então, existe uma probabilidade não negligenciável de que seja encontrada uma colisão para a função de resumo criptográfico definida por H. Na demonstração do esquema de assinatura única, a colisão de resumo criptográfico é obtida por uma assinatura forjada $\tilde{S}$ de uma mensagem $\tilde{M}$, tal que $\tilde{\boldsymbol{S}} \neq \boldsymbol{K}\tilde{M}$. Lyubashevsky (2009) faz isso de maneira diferente, pois seria exigido do proprietário da chave privada computar $\boldsymbol{K}M + \boldsymbol{y}$, para algum $\boldsymbol{y}$ escolhido de maneira aleatória pelo adversário. Contudo, a máscara $\boldsymbol{y}$ não é compartilhada ao *verificador* no esquema de identificação. Portanto, a demonstração de segurança utiliza uma técnica chamada *rewind*. Uma demonstração formal não será apresentada aqui. Porém, esta técnica é bastante utilizada para mostrar segurança em esquemas interativos.

De maneira resumida, o proprietário da chave privada faz o papel de *fornecedor* e identifica-se para o adversário, que faz o papel de *verificador*. Depois disso, o proprietário da

chave e o adversário trocam de papel. O proprietário da chave privada faz o papel de *verificador* e avalia se o adversário, agora fazendo o papel de *fornecedor*, consegue se passar por ele de maneira indevida. Considere que um adversário forje uma resposta válida $\tilde{\boldsymbol{S}}_1$ para um desafio $\tilde{M}_1$ e uma chave pública $(H, \hat{\boldsymbol{K}})$. A técnica de *rewind* dita que, com uma probabilidade não negligenciável, se o proprietário da chave privada rebobinar no tempo (realiza um *rewind*) e submeter um outro desafio, seja ele $\tilde{M}_2$, então existe uma probabilidade não negligenciável de que o adversário consiga forjar uma resposta $\tilde{\boldsymbol{S}}_2$ para $\tilde{M}_2$ e com as mesmas propriedades. Com isso, o proprietário da chave privada obtém duas respostas forjadas, $\tilde{\boldsymbol{S}}_1$ e $\tilde{\boldsymbol{S}}_2$, para dois desafios, $\tilde{M}_1$ e $\tilde{M}_2$, tal que $\tilde{M}_1 \neq \tilde{M}_2$. Assim, uma colisão na função de resumo criptográfico formada pela matriz H é definida pelos elementos $\tilde{\boldsymbol{S}}_1 - \boldsymbol{K}\tilde{M}_1$ e $\tilde{\boldsymbol{S}}_2 - \boldsymbol{K}\tilde{M}_2$, com probabilidade não negligenciável. Logo, ao conjecturar que é computacionalmente impraticável encontrar colisões de resumo criptográfico nesta função, conjectura-se ser computacionalmente impraticável que um adversário se passe de maneira indevida por uma outra entidade no esquema de identificação.

O passo (i), não comentado até o momento, requer evidência da propriedade de *witness-indistinguishable*, exigido pelos esquemas de identificação. Em outras palavras, exige-se evidência de que o vazamento de informações do esquema de assinatura única não ocorre neste esquema de identificação. Para isto, basta demonstrar que, para toda chave privada $\boldsymbol{K}$ e desafio M, a resposta $\boldsymbol{S} := \boldsymbol{K}M + \boldsymbol{y}$ é desconexa do desafio. Veja que $\boldsymbol{y}$ faz o papel de uma máscara que *esconde* $\boldsymbol{K}M$ do *verificador*. Assim, uma vez que y é obtido de maneira uniforme e aleatória, não pode-se assumir informações sobre $\boldsymbol{K}M$ e, qualquer que seja o adversário, ele não consegue assumir informações sobre $D_H(\boldsymbol{K}, M)$. Contudo, é deste contexto que surge o conceito de *abort*. Em vários esquemas de identificação, obter valores para $\boldsymbol{y}$ que completamente escondem $\boldsymbol{K}M$ é computacionalmente impossível ou desvantajoso. Um exemplo disso é quando o intervalo em que amostra-se $\boldsymbol{y}$ é infinito, como ocorre no esquema de identificação de Girault (1991). Neste esquema, deve-se amostrar uma valor aleatório e uniforme do conjunto dos números inteiros. Para solucionar este problema, este esquema não pós-quântico obtém valores aleatórios de um intervalo bem grande. Assim, mesmo que o intervalo exigido seja infinito, devido a magnitude de $\boldsymbol{y}$, encontra-se impraticável obter informações úteis de $\boldsymbol{K}M$ na expressão $\boldsymbol{K}M + \boldsymbol{y}$. Esta abordagem não pode ser utilizada, mesmo que adaptada, pelo esquema de identificação baseado em reticulado proposto por Lyubashevsky (2009). O motivo disso se dá pelas conjecturas sobre dificuldade de solucionar o SVP_γ e fornece credibilidade à segurança do esquema de identificação. Se for aplicado a mesma técnica presente no esquema Girault, as conjecturas sobre a dificuldade de encontrar uma solução com aproximação γ do SVP (SVP_γ) se tornam mais fracas e o esquema passa a depender de conjecturas mais extremas sobre problemas de reticulados.

Desta forma, surgiu-se o conceito de abortar o processo de identificação para que a resposta não forneça informações sobre a chave privada. A ideia é: mesmo que certas informações sobre $\boldsymbol{K}M$ sejam vazadas, todos os possíveis pares $(\boldsymbol{K}M, \boldsymbol{y})$ são igualmente prováveis de terem formado $\boldsymbol{S} := \boldsymbol{K}M + \boldsymbol{y}$, para algum desafio M. Portanto, muito pouco pode ser assumido sobre $\boldsymbol{K}M$ e a chave privada $\boldsymbol{K}$. Assim, obtém-se a propriedade de *witness-indistinguishable*. Um *abort* no esquema ocorre se $[\boldsymbol{S} \in \mathcal{A}]$, onde a condição para uma resposta $\boldsymbol{S}$ pertencer a $\mathcal{A}$

depende da norma ℓ_∞, conforme a definição do conjunto $\mathcal{A}$.

Algoritmo 4 Geração de chaves (*Fiat-Shamir with Aborts*)

Entrada: Considere que os valores são obtidos de uma distribuição uniforme.
Saída: Par de chaves $(pk = (H, \hat{\boldsymbol{K}}), sk = \boldsymbol{K})$

$K \leftarrow \mathcal{K}$
$H \leftarrow \mathcal{H}$
$\hat{K} := H\boldsymbol{K}$
retorne $(pk = (H, \hat{\boldsymbol{K}}), sk = \boldsymbol{K})$

Algoritmo 5 Criação de assinatura (*Fiat-Shamir with Aborts*)

Entrada: Seja $M \in \{0,1\}*$ mensagem a ser assinada. Seja $sk = \boldsymbol{K} \in \mathcal{K}$ uma chave privada.
Saída: Assinatura $\sigma = (\boldsymbol{S}, c)$.

$\boldsymbol{S} := \perp$
enquanto $\boldsymbol{S} = \perp$ **faça**
    $\boldsymbol{y} \leftarrow D_1^m$
    $\boldsymbol{Y} := H\boldsymbol{y}$
    $c := h(\boldsymbol{Y}, M) \in \mathcal{M}$ ▷ *Onde h é uma função de resumo criptográfico qualquer*
    $S := \boldsymbol{K}c + \boldsymbol{y}$
    se $\boldsymbol{S} \in \mathcal{A}$ **então**
        $\boldsymbol{S} := \perp$
    fim se
fim enquanto
retorne $\sigma = (\boldsymbol{S}, c)$

Algoritmo 6 Verificação de assinatura (Fiat-Shamir with Aborts)

Entrada: Seja $M \in \{0,1\}*$ uma mensagem, $pk = (H, \hat{\boldsymbol{K}}$ uma chave pública e $\sigma = (\boldsymbol{S}, c)$ uma assinatura.
Saída: Resultado da verificação. ▷ *Onde h é uma função de resumo criptográfico qualquer*

retorne $\boldsymbol{S} \notin \mathcal{A}$ e $c = h(H\boldsymbol{S} - \hat{\boldsymbol{K}}c, M)$

Uma vez que o esquema de identificação é descrito, consegue-se aplicar a heurística de *Fiat-Shamir* no contexto de reticulados para obter o esquema de assinatura desejado. Assim, os algoritmos de geração de chaves, criação e verificação de assinaturas podem ser vistos, respectivamente, no Algoritmo 4, Algoritmo 5 e Algoritmo 6. Se o esquema de identificação é considerado um esquema seguro, o esquema de assinatura resultante comporta as mesmas propriedades de segurança.

4 ESQUEMA DILITHIUM E IMPLEMENTAÇÃO

4.1 DILITHIUM

Dilithium é um esquema de assinatura digital da família CRYSTALS e sua segurança é baseada na dificuldade de solucionar problemas difíceis de reticulados. O esquema segue a heurística *Fiat-Shamir with Aborts* e, por este motivo, muito se assemelha ao que foi apresentado anteriormente (Seção 3). Ao computar a assinatura de uma mensagem com a chave privada, a mesma ideia de *abort* é adotada. Uma vez que identifica-se o vazamento de informações sensíveis da chave privada, o processo de construção de assinatura é abortado e reiniciado. Assim, ataques de recuperação da chave privada se mantêm computacionalmente impraticáveis de serem realizados. Outros dois ataques que um esquema de assinatura encontra-se sujeito são ataques de substituição de mensagem e construção indevida de assinaturas digitais. Todos estes três ataques se resumem a problemas difíceis de serem solucionados tanto por computadores clássicos quanto por computadores quânticos. Assim, estes problemas são problemas difíceis de reticulados ou, em certos casos, o problema de encontrar uma colisão de resumo criptográfico em uma função de resumo criptográfico resistente a colisão.

A credibilidade de que os problemas relacionados ao esquema sejam realmente difíceis de serem solucionados origina-se de conjecturas sobre o *Short Integer Solution* e *Learning With Errors* com reticulados modulares (nomeados, respectivamente, MSIS e MLWE). Os reticulados estruturados permitem que a álgebra presente no esquema Dilithium, que envolve a estrutura algébrica de anéis quocientes polinomiais, seja isomórfica aos reticulados modulares. Assim, devido a teorias de convolução de vetores e o Teorema Chinês do Resto, obtém-se um esquema de assinatura que realiza multiplicações rápidas de polinômios (com mais baixa complexidade assintótica), apresenta bons resultados de performance e com uma segurança baseada na dificuldade de problemas de reticulados. Onde boa performance é, tempo de execução e espaço de memória ocupado pelos pares de chaves e pelas assinaturas.

A Tabela 2 apresenta os parâmetros utilizados no Dilithium junto com seus respectivos valores. Os valores dos parâmetros apresentados estão separados conforme os níveis de segurança submetidos ao esforço de criptografia pós-quântica do NIST. O NIST requisitou que sejam definidos 5 níveis de segurança, onde quanto maior o número do nível maior a segurança. Os níveis 4 e 5 representam altos níveis de segurança. Diversos níveis de segurança, e parâmetros que os definem, foram propostos na primeira rodada e atualizados nas demais rodadas do esforço de criptografia. Apesar do esquema ser selecionado para a padronização, atualmente, só tem-se conhecimento dos parâmetros definidos na Tabela 2, que foram introduzidos no início da terceira rodada do NIST. O significado de cada parâmetro, apesar de brevemente discutido em sua coluna, será melhor aprofundado ao longo da explicação do Dilithium e os algoritmos que o compõem. Por enquanto, os parâmetros que devem ser compreensíveis, pois são associados à fundamentação teórica apresentada nas seções anteriores, são q e n e correspondem ao anel quociente polinomial $R_q = \mathbb{Z}_q[x]/\langle x^n + 1\rangle$.

Nível de Segurança do NIST:	2	4	5
Parâmetros			
n [*quantidade de coeficientes nos polinômios*]	256	256	256
q [módulo]	8380417	8380417	8380417
d [bits ignorados de t]	13	13	13
τ [*quantidade de* ± 1*'s no vetor* c]	39	49	60
γ_1 [intervalo da distribuição uniforme de y]	2^{17}	2^{19}	2^{19}
γ_2 [parâmetro de centralização dos valores baixos]	$(q-1)/88$	$(q-1)/32$	$(q-1)/32$
(k,l) [comprimento para vetores e matrizes]	$(4,4)$	$(6,5)$	$(8,7)$
η [intervalo da distribuição uniforme de s_1 e s_2]	2	4	2
β $[\tau \cdot \eta]$	78	196	120
ω [*quantidade de 1's na dica presente na assinatura*]	80	55	75

Tabela 2 – Parâmetros para níveis de segurança do NIST.

Neste último capítulo do trabalho, será definido o esquema de assinatura Dilithium, com detalhes sobre a geração de chaves, geração e construção de assinaturas. Serão apresentados as propriedades do esquema, permitindo enxergar o dimensionamento das estruturas algébricas, vazamento de informações, etc. E será clarificado quais são os problemas que, se solucionados, levam um adversário a invalidar as propriedades de autenticidade, não-repúdio e integridade asseguradas pelos esquemas criptográficos de assinatura digital. Além do mais, serão apresentadas as principais evidências de segurança, discutindo também detalhes de implementação. No fim, planeja-se concluir os objetivos propostos no início do trabalho, fornecendo um material teórico, acompanhado de implementação, que auxilie em compreender a aplicação de reticulados no contexto de esquemas pós-quânticos de assinatura digital.

4.2 DEFINIÇÃO DO ESQUEMA

Segue abaixo a descrição dos algoritmos de geração de chaves, construção e verificação de assinaturas que compõem o esquema Dilithium. Considere que as explicações fornecidas aqui, sobre estes processos, são parciais e que servirão somente como uma introdução ao conceitos discutidos posteriormente neste trabalho.

Geração de chaves. As principais características a serem notadas em respeito da geração de chaves são: a compressão do par de chaves, a relação matemática entre a chave privada e a chave pública; e, por último, o objetivo de cada termo presente nas chaves do par. A razão por trás da compressão de chaves é bem simples de se perceber. A matriz $A \in (R_q)^{k \times l}$ é composta por $k \times l$ polinômios com n coeficientes cada. Esta matriz é armazenada tanto na chave pública quanto na chave privada, pois ela é utilizada na geração e verificação de assinaturas. Assim, cada um dos $k \times l \times n$ coeficientes são armazenados em $\log_2(q)$ bits, o que faz a matriz A ocupar um espaço majoritário dentro das duas chaves. Dessa forma, os métodos do esquema relacionados à expansão de vetores utilizam uma semente ρ para expandir a matriz A. Assim, consegue-se

compactar ambas as chaves ao armazenar A não com $k \times l \times n \times \log_2(q)$ bits, mas, com uma única semente ρ de 32 bytes. Os métodos relacionados à expansão dos vetores e reduções dos tamanhos das chaves, que também envolve outras estruturas além de A, são aprofundados ao discutir os algoritmos de suporte (Seção 4.3).

A relação matemática entre a chave privada e a chave pública se dá pela construção do vetor de polinômios $\boldsymbol{t} := A\boldsymbol{s}_1 + \boldsymbol{s}_2$ e permite que uma assinatura seja verificada pela chave pública do assinante. Para perceber que esta relação não expõe os segredos da chave privada assuma que um adversário tome conhecimento de $\boldsymbol{t}$. Isto não é errado de se fazer, pois $\boldsymbol{t}$ é uma informação da chave pública em versões simplificadas do Dilithium e parcialmente conhecido na versão demonstrada ($\boldsymbol{t}_1$ pertence à chave pública). Assim, o segredo da chave privada representado pelo par $(\boldsymbol{s}_1, \boldsymbol{s}_2)$ se mantém protegido pela dificuldade de solucionar instâncias difíceis do MLWE. De maneira breve perceba que, ao tomar o vetor $\boldsymbol{s}_2$ como os vetores de erros uniformemente aleatórios desconhecidos e ao tomar o vetor $\boldsymbol{t}$ como a solução da expressão cuja entrada é a matriz A, descreve-se o problema de reticulado MLWE. Logo, este problema consiste em encontrar a função de maior probabilidade, neste caso definida por $\boldsymbol{s}_2$, que mapeia a entrada à saída.

Os últimos elementos da chave a serem comentados são a sequência de bytes tr e K. Os bytes de tr representam o resumo criptográfico da chave pública e, a menos que sejam encontradas colisões de resumo criptográfico, garantem que no momento da verificação realmente é a chave pública do assinante que está verificando a assinatura. Os bytes de K funcionam como entropia, junto com a mensagem a ser assinada, para obter de maneira uniforme e aleatória o vetor y na versão determinística (a versão apresentada) do Dilithium. A sequência K não é utilizada em versões não-determinísticas do Dilithium.

Algoritmo 7 Geração de chaves (Dilithium$_{n,q,d,\tau,\gamma_1,\gamma_2,k,l,\eta,\beta,\omega}$)

Saída: Um par de chaves (pk, sk).

1. $\zeta \leftarrow \{0,1\}^{256}$
2. $(\rho, \rho', K) \in \{0,1\}^{256} \times \{0,1\}^{512} \times \{0,1\}^{256} := \mathrm{H}(\zeta)$
3. $A \in (R_q)^{k \times l} :=$ ExpandA(ρ) ▷ *Assuma que a matriz A é obtida em sua forma NTT*
4. $(\boldsymbol{s}_1, \boldsymbol{s}_2) \in (S_\eta)^l \times (S_\eta)^k :=$ ExpandS(ρ') ▷ *Onde $S_\eta \subset R_q$ é o conjunto de todos os polinômios cuja norma ℓ_∞ é menor ou igual a η*
5. $\boldsymbol{t} := A\boldsymbol{s}_1 + \boldsymbol{s}_2$
6. $(\boldsymbol{t}_1, \boldsymbol{t}_0) :=$ Power2Round$_q(t, d)$
7. $tr \in \{0,1\} := \mathrm{H}(\rho \| \boldsymbol{t}_1)$
8. **retorne** $(pk = (\rho, \boldsymbol{t}_1), sk = (\rho, K, tr, \boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{t}_0))$

Construção e verificação de assinaturas. A assinatura digital deve criar um relacionamento matemático que envolve a chave privada, a chave pública e a mensagem assinada. Com isto têm-se o objetivo de permitir que uma assinatura construída pela chave privada seja verificada unicamente pela chave pública do assinante. Não só isso, deseja-se que uma assinatura verificada pela chave pública do assinante somente seja capaz de ter sido criada pela respectiva

chave privada. Assim, estabelece-se autenticidade e não-repúdio, que são duas propriedades de segurança fornecidas por esquemas de assinatura.

A terceira propriedade de segurança, integridade da mensagem, é obtida pelo forte vínculo da assinatura com a mensagem assinada. Para enxergar o relacionamento no Dilithium que obtém estes resultados, considere que M e $\boldsymbol{w}_1 := \text{HighBits}_q(A\boldsymbol{y}, 2\gamma_2)$, linha 9 do Algoritmo 9, sejam, respectivamente, a mensagem assinada e o vetor produzido no processo de construção da assinatura. Considere que M' e $\boldsymbol{w}'_1 := \text{UseHint}_q(h, A\boldsymbol{z} - c\boldsymbol{t}_1 \cdot 2^d, 2\gamma_2)$, linha 4 do Algoritmo 8, sejam a mensagem em verificação e o vetor produzido com a chave pública mais a assinatura no processo de verificação, respectivamente. Assim, se a mensagem assinada não for modificada e a respectiva chave pública do assinante for fornecida à verificação, então tem-se que o resumo da chave pública e a mensagem resultam, ao serem entregues à função de resumo criptográfico H, na mesma saída que foi obtida dentro do processo de criação da assinatura. Isto é, $\mu' := \text{H}(\textit{resumo da chave pública} \parallel M')$ é idêntico a $\mu := \text{H}(tr \parallel M)$ utilizado para assinar a mensagem. Portanto, a menos que colisões de resumo criptográfico sejam encontradas, considerado impraticável de se realizar, quaisquer mudanças na mensagem ou na chave pública resultam em uma assinatura inválida.

Além disso, a relação matemática de ambas as chaves, definida por t, e os dados $(\boldsymbol{z}, \boldsymbol{h})$ inclusos na assinatura permitem computar $\boldsymbol{w}_1$, linha 9 da construção de assinatura, e $\boldsymbol{w}'_1$, linha 4 da verificação de assinaturas, tal que $\boldsymbol{w}'_1 = \boldsymbol{w}_{1s}$. A explicação da igualdade destas duas expressões é obtida pelo comportamento de UseHint_q e HighBits_q e a matemática envolvida é apresentada na Seção 4.5. Uma vez que os dois vetores sejam idênticos, a comparação $\text{H}(\mu' \| \boldsymbol{w}'_1) = \text{H}(\mu \| \boldsymbol{w}_1)$ identifica a assinatura como válida. Contudo, ainda deseja-se que a assinatura somente possa ter sido construída com o uso da chave privada. O esquema fornece esta segurança, pois um adversário que tente construir uma assinatura indevidamente, sem a chave privada, encontra-se buscando uma tripla $(\tilde{c}, \boldsymbol{z}, \boldsymbol{h})$ que satisfaça os requisitos do esquema e solucione instâncias difíceis do problema SelfTargetMSIS. O problema SelfTargetMSIS, cuja definição encontra-se na Seção 4.6, é demonstrado ser de complexidade igual ou superior ao problema MSIS de reticulado por uma redução não-rigorosa, mas, que ainda fornece evidências de segurança. Logo, o esquema encontra-se seguro contra a construção indevida de assinaturas com base em fortes conjecturas sobre os problemas de reticulados.

Algoritmo 8 Verificação de assinatura ($\text{Dilithium}_{n,q,d,\tau,\gamma_1,\gamma_2,k,l,\eta,\beta,\omega}$)

Entrada: Uma chave pública $pk := (\rho, t_1)$, uma mensagem $M \in \{0,1\}*$ e uma assinatura $\sigma := (\tilde{c}, \boldsymbol{z}, \boldsymbol{h})$

Saída: Um valor lógico que corresponde à validade da assinatura.

1► $A \in (R_q)^{k \times l} :=$ ExpandA(ρ) ▷ *Assuma que a matriz A é obtida em sua forma NTT*

2► $\mu \in \{0,1\}^{512} := \text{H}(\text{H}(\rho \| t_1) \| M)$

3► $c :=$ SampleInBall$(\tilde{c})$ ▷ $\hat{c} := NTT(c)$

4► $\boldsymbol{w}'_1 :=$ $\text{UseHint}_q(\boldsymbol{h}, A\boldsymbol{z} - c\boldsymbol{t}_1 \cdot 2^d, 2\gamma_2)$ ▷ $A\boldsymbol{z} - c\boldsymbol{t}_1 := NTT^{-1}(A * NTT(\boldsymbol{z}) - \hat{c} * NTT(\boldsymbol{t}_1))$

5► **retorne** $\|\boldsymbol{z}\|_\infty < \gamma_1 - \beta$ e $\tilde{c} = \text{H}(\mu \| \boldsymbol{w}'_1)$ e *"a quantidade de 1's em* $\boldsymbol{h}$ *é menor ou igual a* ω*"*

Algoritmo 9 Construção de assinatura (Dilithium$_{n,q,d,\tau,\gamma_1,\gamma_2,k,l,\eta,\beta,\omega}$)

Entrada: Uma chave privada $sk := (\rho, K, tr, \boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{t}_0)$ e uma mensagem $M \in \{0,1\}^*$.
Saída: Uma assinatura $\sigma = (\tilde{c}, \boldsymbol{z}, \boldsymbol{h})$

1. $A \in (R_q)^{k \times l} :=$ ExpandA(ρ) ▷ *Assuma que a matriz A é obtida em sua forma NTT*
2. $\mu \in \{0,1\}^{512} :=$ H$(tr \| M)$
3. $\kappa := 0$
4. $(\boldsymbol{z}, \boldsymbol{h}) := \perp$
5. $\rho' \in \{0,1\}^{512} :=$ H$(K \| \mu)$
6. **enquanto** $(\boldsymbol{z}, \boldsymbol{h}) = \perp$ **faça**
7. $\quad \boldsymbol{y} \in (\tilde{S}_{\gamma_1})^l :=$ ExpandMask(ρ', κ) ▷ *Onde $\tilde{S}_{\gamma_1} \subset R_q$ é o conjunto de polinômios resultados de centered modular reduction com γ_1*
8. $\quad \boldsymbol{w} := A\boldsymbol{y}$ ▷ $\boldsymbol{w} := NTT^{-1}(\hat{A} * NTT(y))$
9. $\quad \boldsymbol{w}_1 :=$ HighBits$_q(\boldsymbol{w}, 2\gamma_2)$
10. $\quad \tilde{c} \in \{0,1\}^{256} :=$ H$(\mu \| \boldsymbol{w}_1)$
11. $\quad c \in B :=$ SampleInBall$(\tilde{c})$ ▷ $\hat{c} := NTT(c)$
12. $\quad \boldsymbol{z} := \boldsymbol{y} + c\boldsymbol{s}_1$ ▷ $c\boldsymbol{s}_1 := NTT^{-1}(\hat{c} * NTT(\boldsymbol{s}_1))$
13. $\quad \boldsymbol{r}_0 :=$ LowBits$_q(\boldsymbol{w} - c\boldsymbol{s}_2, 2\gamma_2)$ ▷ $c\boldsymbol{s}_2 := NTT^{-1}(\hat{c} * NTT(\boldsymbol{s}_2))$
14. $\quad$ **se** $\|\boldsymbol{z}\|_\infty \geq \gamma_1 - \beta$ ou $\|\boldsymbol{r}_0\|_\infty \geq \gamma 2 - \beta$ **então**
15. $\quad\quad (\boldsymbol{z}, \boldsymbol{h}) := \perp$
16. $\quad$ **senão**
17. $\quad\quad \boldsymbol{h} :=$ MakeHint$_q(-c\boldsymbol{t}_0, \boldsymbol{w} - c\boldsymbol{s}_2 + c\boldsymbol{t}_0, 2\gamma_2)$ ▷ $c\boldsymbol{t}_0 := NTT^{-1}(\hat{c} * NTT(\boldsymbol{t}_0))$
18. $\quad\quad$ **se** $\|c\boldsymbol{t}_0\|_\infty \geq \gamma_2$ ou *"a quantidade de 1's em h é maior que ω"* **então**
19. $\quad\quad\quad (\boldsymbol{z}, \boldsymbol{h}) := \perp$
20. $\quad\quad$ **fim se**
21. $\quad$ **fim se**
22. $\quad \kappa := \kappa + l$
23. **fim enquanto**
24. **retorne** $\sigma := (\tilde{c}, \boldsymbol{z}, \boldsymbol{h})$

Tanto as características mencionadas quanto novas características sobre a construção e verificação de assinaturas e a geração de chaves serão explicadas nas próximas seções. A Seção 4.3 aprofunda os métodos que foram omitidos nos algoritmos acima e explica com mais detalhes passos como a geração de valores aleatórios, expansão de vetores e decomposição dos coeficientes. A Seção 4.4 discute o vazamento de informações herdado da heurística de *Fiat-Shamir with aborts*, mencionando a solução que mantém a chave privada em segredo e o número de *aborts* esperados ao assinar uma mensagem. A Seção 4.5 demonstra a matemática do esquema. Com isto, consegue-se computar na verificação da assinatura o vetor $\boldsymbol{w}_1'$ que participa na atribuição de validade à assinatura. As Seções 4.6 e 4.7, discutem a segurança do esquema, apresentando os principais ataques e a razão pelo qual acredita-se que estes ataques sejam computacionalmente impraticáveis de serem realizados. As duas últimas seções (a Seção 4.8 e a Seção 4.9) discutem pequenas variações do esquema Dilithium, características de implementação e a contribuição fornecida por este trabalho.

4.3 ALGORITMOS DE SUPORTE

Diversas partes dos três algoritmos que compõem o esquema de assinatura digital foram omitidos por métodos ainda não descritos. Isto foi feito para que as propriedades do esquema sejam separadas e explicadas com mais clareza. Deste modo, abaixo apresenta-se os algoritmos de suporte do Dilithium. Todos estes métodos estão relacionados de alguma forma com a decomposição de coeficientes conforme um parâmetro de centralização, cuja descrição formal será ainda apresentada. Os métodos HighBits_q e LowBits_q retornam um elemento do par que define a decomposição. O método Power2Round_q decompõe a entrada para um parâmetro de centralização que é uma potência de 2. O método MakeHint_q cria as dicas inclusas na assinatura e UseHint_q faz uso destas dicas para que seja possível reconstruir parcialmente a decomposição de coeficientes que verifica a assinatura. Além dos algoritmos de suporte, serão discutidas características consequentes destas e demais operações do esquema. Portanto, separa-se esta seção nos seguinte tópicos: geração de valores pseudo-aleatórios, multiplicação de polinômios, expansão de vetores, reduções modulares, decomposição de valores, norma de polinômios, construção do desafio e empacotamento de chaves e assinaturas.

Geração de valores pseudo-aleatórios. A geração de valores pseudo-aleatórios (relacionados ao uso da função de resumo criptográfico H) está presente em diversas partes dos algoritmos de geração de chaves, construção de assinaturas e verificação de assinaturas. A versão do esquema Dilithium que foi descrita utiliza os algoritmos SHAKE-128 e SHAKE-256, pertencentes ao padrão de funções de resumo criptográfico FIPS 202 (STANDARDS; TECHNOLOGY, 2015). Conforme as propriedades das funções de resumo criptográfico, pode-se utilizar a saída destas funções para obter valores imprevisíveis e uniformemente pseudo-aleatórios. Contudo, o uso estes algoritmos são custosos e ainda não existe suporte abundante em *hardware* para a família SHAKE. Assim, na terceira etapa do esforço criptográfico, foi proposta uma versão alternativa do Dilithium que utiliza o algoritmo *Advanced Encryption Standard* (AES), em respeito ao padrão FIPS 197 (STANDARDS; TECHNOLOGY, 2001). O objetivo desta proposta é fornecer evidência de um melhor desempenho até que as funções padronizadas de resumo criptográfico possuam suporte em *hardware* comumente disponível.

Expansão de vetores. A expansão de vetores (cujos métodos relacionados são ExpandA, ExpandS e ExpandMask) são operações que amostram uma série de polinômios pseudo-aleatórios pertencentes ao anel $R_q = \mathbb{Z}_q/\langle x^n + 1\rangle$, em coerência às propriedades de cada estrutura. Em um método de expansão arbitrário, dada uma semente $s \in \{0,1\}^*$ de tamanho fixo, gera-se uniformemente uma sequência de vetores de polinômios, ou diversas sequências de vetores de polinômios, que serão utilizadas pelos algoritmos do esquema. Assim, estes métodos de expansão acabam sendo responsáveis pela descompactação da chave pública e privada.

Dessa forma, considere o termo *empacotado* para representar a menor codificação que uma estrutura não descartada permite ser representada. Observe que na versão apresentada, ao armazenar a semente ρ tal que $A = \text{ExpandA}(\rho)$, compacta-se as duas chaves do par, pois

Algoritmo 10 HighBits_q

Entrada: Um coeficiente $r \in \mathbb{Z}_q$ e o parâmetro de centralização $\alpha \in \mathbb{Z}$.

Saída: O valor alto da decomposição $r_1 \in \{0, 1, 2, \cdots, (q-1)/\alpha\}$.

1▸ $(r_1, r_0) := \text{Decompose}_q(r, \alpha)$
2▸ **retorne** r_1

Algoritmo 11 Power2Round_q

Entrada: Coeficiente r e a potência d do parâmetro de centralização.

Saída: Realiza a decomposição de r com o parâmetro de centralização 2^d.

1▸ $r := r \bmod q$
2▸ $r_0 := r \operatorname{cmod} 2^d$
3▸ **retorne** $((r - r_0)/2^d, r_0)$

Algoritmo 12 Decompose_q

Entrada: Coeficiente $r \in \mathbb{Z}_q$ de um vetor e parâmetro de centralização $\alpha \in \mathbb{Z}$.

Saída: Decompõe $r \in \mathbb{Z}_q$ em $(r_1, r_0) \in \mathbb{Z}_q \times \mathbb{Z}$ tal que $r = r_1 \cdot \alpha + r_0$

1▸ $r := r \bmod q$
2▸ $r_0 := r \operatorname{cmod} \alpha$
3▸ **se** $r - r_0 = q - 1$ **então**
4▸ $\quad r_1 := 0$
5▸ $\quad r_0 := r_0 - 1$
6▸ **senão**
7▸ $\quad r_1 := (r - r_0)/\alpha$
8▸ **fim se**
9▸ **retorne** (r_1, r_0)

Algoritmo 13 LowBits_q

Entrada: Um coeficiente $r \in \mathbb{Z}_q$ e o parâmetro de centralização $\alpha \in \mathbb{Z}$.

Saída: O valor baixo da decomposição $r_0 \in [-\alpha/2, \alpha/2)$.

1▸ $(r_1, r_0) := \text{Decompose}_q(r, \alpha)$
2▸ **retorne** r_0

Algoritmo 14 MakeHint_q

Entrada: Dois coeficientes $z, r \in \mathbb{Z}_q$ e o parâmetro de centralização $\alpha \in \mathbb{Z}$.

Saída: Retorna o valor lógico (0 ou 1) indicando que, ao somar z em r, modifica-se o valor alto do coeficiente.

1▸ $r_1 := \text{HighBits}_q(r, \alpha)$
2▸ $v_1 := \text{HighBits}_q(r + z, \alpha)$
3▸ **retorne** $r_1 \neq v_1$

Algoritmo 15 UseHint_q

Entrada: Uma dica $h \in \{0, 1\}$, um coeficiente $r \in \mathbb{Z}_q$ e o parâmetro de centralização $\alpha \in \mathbb{Z}$.

Saída: Recupera o valor alto r_1 obtido com auxílio da dica.

1▸ $m := (q-1)/\alpha$
2▸ $(r_1, r_0) := \text{Decompose}(r, \alpha)$
3▸ **se** $h = 1$ e $r_0 > 0$ **então**
4▸ $\quad$ **retorne** $(r_1 + 1) \bmod m$
5▸ **senão se** $h = 1$ e $r_0 \leq 0$ **então**
6▸ $\quad$ **retorne** $(r_1 - 1) \bmod m$
7▸ **fim se**
8▸ **retorne** r_1

a semente é significativamente menor que a matriz de polinômios A. Deste modo, o método ExpandA recebe na entrada a semente $\rho \in \{0,1\}^{256}$, empacotada em 32 bytes, e expande os polinômios atribuídos à matriz $A \in (R_q)^{k \times l}$, empacotada em $(k \cdot l \cdot n \cdot \log_2 q)/8$ bytes. O método ExpandS recebe na entrada uma semente $\rho' \in \{0,1\}^{512}$, empacotada em 64 bytes, onde os primeiros l polinômios expandidos são atribuídos ao vetor $\boldsymbol{s}_1 \in (R_q)^l)$, empacotado em $(l \cdot n \cdot \log_2 2\eta)/8$ bytes, e os seguinte k polinômios são atribuídos ao vetor $\boldsymbol{s}_2$, empacotado em $(k \cdot n \cdot \log_2 2\eta)/8$ bytes. Por fim, o método ExpandMask recebe uma semente $\rho' \in \{0,1\}^{512}$, que não é a mesma da anterior pois pertence a outro algoritmo, mas que também é empacotada

em 64 bytes. Este método concatena esta semente ao *nonce* κ, incrementado após cada falha na construção da assinatura, e utiliza este resultado para expandir o vetor $\boldsymbol{y}$. O vetor $\boldsymbol{y}$ nunca será empacotado, porque sua função é proteger o segredo da chave privada na assinatura e é descartado após a construção da assinatura.

As três sequências de bytes ρ, ρ' e K obtidas no início do algoritmo de geração de chaves são utilizadas para gerar todos os demais elementos da chave privada pela expansão de vetores e outros métodos computacionais. A origem destas três sequências é a semente uniforme e aleatória ζ, que é entregue como entrada a uma função de resumo criptográfico em que a saída amostrada é atribuída às sequências ρ, ρ' e K. Por este motivo, uma das variações do Dilithium permite armazenar somente ζ na chave privada, computar e expandir os elementos restantes. Assim, reduz-se o tamanho da chave privada e em troca aumenta-se o esforço computacional para construir assinaturas.

Reduções modulares. Existem duas reduções modulares utilizadas no esquema. Uma delas é a operação de módulo convencional da matemática (representado pelos símbolos *mod*) e a outra é operação *centered modular reduction* (representado pelos símbolos *cmod*). A operação de módulo convencional, dado um divisor a e um dividendo q, computa o resto $r \in \mathbb{Z}_q$ da divisão euclideana de a por q. O resultado final desta operação pode ser atingido por uma implementação clássica, com uma única instrução fornecida pela arquitetura, pelo método de redução de Montgomery ou pelo método de redução de Barrett. Os métodos de redução de Montgomery ou Barrett exigem diversas instruções de *hardware*, contudo, permitem implementar o módulo matemático em tempo constante, bastante importante para sistemas criptográficos. O *centered modular reduction*, dado um operando a e um parâmetro de centralização α, centraliza o operando de forma que o resultado $r \in (-\alpha/2, \alpha/2]$. Uma explicação detalhada sobre implementações da operação de centralização pode ser encontrado na publicação de Ducas et al. (2018).

Decomposição de valores. Dado um parâmetro de centralização $\alpha \in \mathbb{N}$, a decomposição de um coeficiente $r \in \mathbb{Z}_q$ (a decomposição de valores é implementada no algoritmo Decompose_q e utilizada nos algoritmos Power2Round_q, HighBits_q e LowBits_q) obtém um par de números (r_1, r_0) tais que $r = r_1 \cdot \alpha + r_0$. O fator r_1 é nomeado valor alto (*high bits*) e $r_1 \cdot \alpha \in \mathbb{Z}_q$. Portanto, temos que $r_1 \in \{0, 1, 2, \cdots, (q-1)/\alpha - 1\}$. O fator r_0 é nomeado valor baixo e é o resultado de um *centered modular reduction* pelo parâmetro de centralização α. Ou seja, $r_0 \in (-\alpha/2, \alpha/2]$.

Norma de polinômios e vetores de polinômios. Considere as seguintes definições de norma de polinômio e norma de um vetor de polinômios.

Definição 4.3.1. Seja $n \in \mathbb{N}$ a dimensão do vetor de polinômios $\boldsymbol{w} \in (\mathbb{Z}_q[x])^n$. A norma ℓ_2 de $\boldsymbol{w}$, denotada por $\|\boldsymbol{w}\|_2$, é definida por $\sqrt{\|w_0\|_2^2 + \cdots + \|w_{n-1}\|_2^2}$, onde $w_i \in \mathbb{Z}_q[x]$ são os termos do vetor.

Definição 4.3.2. Seja $n \in \mathbb{N}$ o grau do polinômio $w \in \mathbb{Z}_q[x]$. A norma ℓ_2 de w, denotada por

$\|w\|_2$, é definida por $\sqrt{\|w_0\|_\infty^2 + \cdots + \|w_n\|_\infty^2}$ onde $w_i \in \mathbb{Z}_q$ são os coeficientes do polinômio.

Definição 4.3.3. Seja $n \in \mathbb{N}$ a dimensão do vetor de polinômios $\boldsymbol{w} \in (\mathbb{Z}_q[x])^n$. A norma ℓ_∞ de $\boldsymbol{w}$, denotada por $\|\boldsymbol{w}\|_\infty$, é definida por $\max_{i=0}^{n} \|w_i\|_\infty$ onde $w_i \in \mathbb{Z}_q$ são os termos do vetor.

Definição 4.3.4. Seja $n \in \mathbb{N}$ o grau do polinômio $w \in \mathbb{Z}_q[x]$. A norma ℓ_∞ de w, denotada por $\|w\|_\infty$, é definida por $\max_{i=0}^{n} \|w_i\|_\infty$ onde $w_i \in \mathbb{Z}_q$ onde $w_i \in \mathbb{Z}_q$ são os coeficientes do polinômio.

Definição 4.3.5. Seja $w \in \mathbb{Z}_q$. No esquema Dilithium, a norma ℓ_∞ de w, denotada por $\|w\|_\infty$, é definida pela expressão $w \bmod q$. Isto é, a *centered modular reduction* de w com o parâmetro de centralização q.

Construção de desafio. O desafio $c \in B$ é um polinômio gerado no algoritmo de construção de assinatura, tal que B é o conjunto de todos os polinômios que contêm τ coeficientes iguais a ± 1 e os demais coeficientes iguais a zero. O algoritmo relacionado a construção do desafio é o SampleInBall. Neste método, utiliza-se a sequência de bytes $\tilde{c}$ presente na assinatura ou obtida pelo resumo criptográfico μ concatenado com o vetor $\boldsymbol{w}_1$ empacotado. Assim, $\tilde{c}$ alimenta a geração de valores pseudo-aleatórios e, se a assinatura não foi modificada, permite que a execução do algoritmo SampleInBall obtenha o mesmo desafio c tanto no algoritmos de construção da assinatura quanto no algoritmo de verificação. O desafio é utilizado junto com as dicas no momento da verificação para construir a expressão que permite reconstruir o vetor $\boldsymbol{w}_1$. Mais detalhes sobre a relação matemática que verifica a assinatura serão apresentados na Seção 4.5.

Algoritmo 16 SampleInBall

Entrada: Uma amostra $\tilde{c}$ uniformemente aleatória que alimenta H para amostrar os valores de j e s.
Saída: Um vetor $c \in B$, onde B é o conjunto de polinômios contendo exatamente τ coeficientes igual a ± 1 e os coeficientes restantes igual a zero.

1. $c := c_0 \cdot x^0 + c_1 \cdot x^1 + \cdots + c_{n-1} \cdot x^{n-1}$; tal que $\forall_{0 \le i < n}, c_i = 0$
2. **para** $i := n - \tau$ **até** n **faça**
3. $\quad j \leftarrow \{0, 1, \cdots, i\}$
4. $\quad s \leftarrow \{0, 1\}$
5. $\quad c_i := c_j$
6. $\quad c_j := (-1)^s$
7. **fim para**

Empacotamento de chaves e assinatura. O processo de empacotamento de chaves e assinaturas consiste em codificar as estruturas, composta de vetores, utilizando a menor quantidade de bits possível. Assim, minimizando o número de bits desperdiçados na codificação. Para quase todos os vetores, o número de bits que será utilizado para representá-lo origina-se do limite inferior e limite superior do intervalo discreto que seus coeficientes pertencem. Independente do método de armazenamento, o limite discreto permite ditar a quantidade de valores possíveis que

o coeficiente pode assumir. Dessa forma, o coeficiente de um polinômio é representado pelo logaritmo do número de valores possíveis encontrado, em bits. Assim, descreve-se na Tabela 3 o número de bytes para empacotar as estruturas do esquemas.

Nível de Segurança do NIST:	2	3	5
Comprimento dos empacotamentos (em bytes)			
s_1 $[l \cdot \frac{n}{8} \cdot \log_2(2\eta)]$	384	640	672
s_2 $[k \cdot \frac{n}{8} \cdot \log_2(2\eta)]$	384	768	768
t_0 $[k \cdot \frac{n}{8} \cdot \log_2(2^d)]$	1664	2496	3328
t_1 $[k \cdot \frac{n}{8} \cdot \lceil \log_2(\frac{q-1}{2^d}) \rceil]$	1280	1920	2560
w_1 $[k \cdot \frac{n}{8} \cdot \log_2(\frac{q-1}{2\gamma_2})]$	512	1152	1536
z $[l \cdot \frac{n}{8} \cdot \log_2(\gamma_1 + \beta)]$	2304	3200	4480
h $[\omega + k]$	84	61	83

Tabela 3 – Comprimento dos vetores empacotados, conforme os parâmetros de cada nível de segurança.

Veja pela descrição do esquema que $\boldsymbol{s}_1$ e $\boldsymbol{s}_2$ são obtidos de maneira uniformemente aleatória cuja norma ℓ_∞ de seus coeficientes é menor ou igual a η. Portanto, o número de valores possíveis diferentes que cada um dos coeficientes dos polinômios presentes nestas estruturas pode assumir é $2\eta + 1$. Agora, veja que o número de coeficientes de cada polinômio é n e os números de polinômios são, respectivamente, l e k. Então, como exemplo, sabe-se que $l \cdot n \cdot \log_2(2\eta + 1)$ resulta no número de bits necessários para representar $\boldsymbol{s}_1$. Ao dividir esse valor por 8, obtém-se o resultado em bytes, como na Tabela 3. Este processo é realizado para cada vetor, com exceção do vetor $\boldsymbol{h}$, que por sua característica, pode ser representado com menos bytes de uma outra forma. Na maioria dos vetores restantes que compõem o par de chaves, o número de possibilidades está definido pela operação de decomposição. Os coeficientes dos vetores $\boldsymbol{t}_0$ e $\boldsymbol{t}_1$ são, respectivamente, os valores altos e os baixos dos coeficientes do vetor $\boldsymbol{t}$ com o parâmetro de centralização 2^d. Logo, $\boldsymbol{t} = \boldsymbol{t}_1 \cdot 2^d + \boldsymbol{t}_0$. Os coeficientes de $\boldsymbol{t}_1$ pertencem a $\{0, 1, \cdots, (q-1)/2^d - 1\}$ e existem $(q-1)/2^d$ diferentes valores possíveis para $\boldsymbol{t}_1$. Desta mesma decomposição, os coeficientes de $\boldsymbol{t}_0$ são resultados do *centered modular reduction* com parâmetro 2^d e pertencem ao intervalo $(-2^{d-1}, 2^{d-1}]$. Assim, existem 2^d diferentes valores possíveis para os coeficientes de $\boldsymbol{t}_0$. Para os coeficientes de $\boldsymbol{w}_1$ existem $(q-1)/2\gamma_2$ diferentes valores possíveis, pela decomposição de w.

Diferente dos últimos três vetores discutidos, o vetor $\boldsymbol{z}$ não é resultado de uma decomposição, mas, assume o valor resultante de expressão $\boldsymbol{y} + c\boldsymbol{s}_1$. Dessa forma, os coeficientes de $\boldsymbol{z}$ assumem um limite superior pertencente ao intervalo

$$\|\boldsymbol{y}\|_\infty + \|c\boldsymbol{s}_1\|_\infty \leq \|\boldsymbol{y}\|_\infty + \|c\|_1 \cdot \|\boldsymbol{s}_1\|_\infty \leq \gamma_1 + \tau \cdot \eta \leq \gamma_1 + \beta. \tag{4.1}$$

Contudo, devido a condição de *abort* da heurística de *Fiat-Shamir*, aplicado no contexto de reticulados, uma assinatura será considerada válida se, e somente se, a norma $\|\boldsymbol{z}\|_\infty < \gamma_1 - \beta$. Deste modo, existem $2 \cdot (\gamma_1 - \beta) - 1$ diferentes valores possíveis para os coeficientes de $\boldsymbol{z}$ e para

os níveis de segurança 2, 3, 5 estes coeficientes são representados, respectivamente, por 18, 20 e 20 bits.

O último vetor que será empacotado é o vetor $\boldsymbol{h}$. Este vetor é composto por polinômios com coeficientes 1's e 0's. Além disso, conforme a definição do esquema, o número máximo de 1's que o vetor de polinômios $\boldsymbol{h}$ pode conter é ω. Só que ω é um número bem menor do que a quantidade total de coeficientes no vetor. Portanto, se o esquema utilizasse o mesmo método de codificação dos outros vetores, considerando que cada coeficiente pode ser representado por um bit, resultaria em $(k \cdot n)/8$ bytes. Contudo, pode-se armazenar somente a posição dos ω coeficientes que são diferentes de 0, identificando qual polinômio o coeficiente pertence e seu índice. Os índices dos ω coeficientes são armazenados em ω bytes e a qual polinômio o coeficiente pertence pode ser identificado pela quantidade de coeficientes 1's que estão em cada polinômio. Isto requer k bytes adicionais. Assim, necessita-se somente de $\omega + k$ (nº máximo de coeficientes diferente de zero + nº de polinômios no vetor) bytes para empacotar o vetor.

Com isso, consegue-se calcular o tamanho da chave pública, chave privada e assinatura. Considere que as sementes e sequência de bits amostrados presentes nas chaves e assinaturas $\rho, K, tr, \tilde{c} \in \{0,1\}^{256}$ são todos representados com 32 bytes. Considere que o tamanho de uma estrutura seja descrita por uma tripla tal que seus elementos, em ordem, apresenta o tamanho da estrutura em bytes para os níveis de segurança 2, 3 e 5, respectivamente. Assim, considere uma chave pública $pk = (\rho, \boldsymbol{t}_1)$, uma chave privada $sk = (\rho, K, tr, \boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{t}_0)$ e uma assinatura $\sigma = (\tilde{c}, \boldsymbol{z}, \boldsymbol{h})$. Desta forma, pk é empacotado com $(1312, 1952, 2592)$ bytes; sk é empacotado com $(2528, 4000, 4864)$ bytes; e, finalmente, σ é empacotado com $(2420, 3293, 4595)$ bytes.

4.4 VAZAMENTO DE INFORMAÇÕES

Devido ao esquema Dilithium ser baseado na heurística de *Fiat-Shamir with Aborts*, mas, utilizando reticulados modulares, os mesmos problemas discutidos previamente prevalecem. Assim, a construção da assinatura $\boldsymbol{z} := \boldsymbol{y} + c\boldsymbol{s}_1$ requer que o vetor $\boldsymbol{y}$ faça papel de máscara e, assim, mantenha a assinatura desconexa do desafio c. Se o $\boldsymbol{y}$ escolhido for obtido de forma uniforme e aleatória, não pode-se assumir informações sobre o resultado de $c\boldsymbol{s}_1$ (onde $\boldsymbol{s}_1$ pertence a chave privada e c pode ser computado pela assinatura). Contudo, em esquemas similares, a obtenção de $\boldsymbol{y}$ de maneira uniforme e aleatória, conforme encontra-se teoricamente requisitado, pode não ser realizável ou ocasiona em situações não favoráveis de serem trabalhadas. O exemplo fornecido na Seção 3.3 foi do esquema de identificação de Girault. Este esquema requer $y \in \mathbb{Z}$, que é um conjunto infinito e não é computacionalmente realizável. No esquema do Diltihium, a razão por trás da amostragem de $\boldsymbol{y}$ não seguir a fundamentação teórica é pelas conjecturas atuais dos problemas de reticulados. Caso o esquema de reticulado amostrasse $\boldsymbol{y}$ no intervalo que não resulta no vazamento de informações, as provas de segurança dependeriam de conjecturas bem mais fracas do que as que são utilizadas atualmente. Assim, optou-se por construir um esquema dependente de conjecturas de reticulados mais bem estabelecidas e evi-

tar o vazamento da chave privada de outras formas. Logo, necessita-se abortar o processo de assinatura e isto ocorre em média quatro vezes, até que a saída produzida seja uma assinatura válida e que não vaze informações da chave privada (linhas 14 e 15 do Algoritmo 9), conforme Ducas et al. (2021).

A técnica de *abort* tem o objetivo de abortar o processo de assinatura ao identificar o vazamento de certas informações da chave privada. Por consequência, permite o esquema manter a propriedade de *witness-indistinguishable* (Definição 2.4.6). O processo de *abort* também pode ser chamado de *Rejection Sampling*, pois resulta na rejeição de assinaturas construídas até que seja amostrado uma assinatura válida e segura. Neste processo, ainda há o vazamento de certas informações da chave privada do assinante, mas, o vazamento é controlado. E assim, se mantém impraticável que um adversário consiga identificar qual a chave privada foi utilizada para a construção da assinatura entre todas as outras possíveis chaves (chaves possíveis $\neq$ chaves equivalentes). Existem duas condições que resultam em um *abort*. A primeira condição é $\|\boldsymbol{z}\|_\infty \geq \gamma_1 - \beta$ e a segunda condição é $\|\text{LowBits}_q(\boldsymbol{w} - c\boldsymbol{s}_2, 2\gamma_2)\|_\infty \geq \gamma_2 - \beta$. A matemática por trás destes valores permitem que qualquer par $(\boldsymbol{z}, c) \in (S_{\gamma_1-\beta-1})^l \times B_\tau$ em uma assinatura seja igualmente provável de ser obtido. A primeira condição está relacionada com o vazamento de informações, enquanto a segunda está relacionada com o vazamento de informações e a corretude do esquema.

O conceito de *abort* introduz diversas dificuldades na definição do esquema, que é a seleção dos parâmetros que serão utilizados. Estes valores são bastante delicados. O parâmetro γ_1 deve ser escolhido estrategicamente. Quando maior for γ_1 mais difícil será para um adversário encontrar a chave privada, contudo, facilita-se que o adversário forje uma assinatura. O inverso também é verdade, quando menor for γ_1 mais difícil será que uma assinatura seja forjada por um adversário e mais fácil será que a chave privada do assinante seja encontrada pelo adversário. As propriedades da segurança que envolvem o esquema são discutidas em detalhes na Seção 4.6.

4.5 VALIDADE DAS ASSINATURAS DILITHIUM

A verificação de assinaturas consiste em verificar três valores lógicos computados da assinatura, mensagem e chave pública munidos pela operação lógica *AND*. O primeiro valor lógico é a primeira condição da linha 5 do Algoritmo 8 e impõe regras de vazamento de informações da chave privada à assinatura. Esta verificação equivale a observar a norma dos polinômios no vetor $\boldsymbol{z}$. Como uma assinatura que vaze informações da chave privada nunca seria construída pelo esquema, se esta condição for desrespeitada, a assinatura é, por consequência, considerada inválida. Como resultado, assinaturas forjadas devem também respeitar esta condição. O segundo valor lógico corresponde diretamente à verificação das propriedades de integridade da assinatura, autenticação e não-repúdio do assinante, segunda condição da linha 5 do Algoritmo 8. Isto faz desta condição a mais interessante e, portanto, será aprofundada mais adiante. A terceira verificação lógica corresponde a observar se o número de dicas fornecidas pela assi-

natura ultrapassa o parâmetro ω do esquema. O valor atribuído a ω, nos respectivos níveis de segurança, restringe soluções do problema de forjar uma assinatura e, portanto, contribui para a evidência de segurança.

Dito isso, a segunda operação lógica, relacionada às principais propriedades de segurança, valida a assinatura com base em modificações (ou a falta delas) nos bits altos de coeficientes de vetores polinomiais. Veja que, ao verificar a assinatura, $\tilde{c} = \mathrm{H}(\mu \| \boldsymbol{w}_1)$ (construído na criação da assinatura) é comparado com $\mathrm{H}(\mu' \| \boldsymbol{w}'_1)$ (computado no momento da verificação). Devido a resistência à colisão de funções de resumo criptográfico, necessita-se que $\mu \| \boldsymbol{w}_1$ seja estritamente igual a $\mu' \| \boldsymbol{w}'_1$. Consegue-se facilmente observar que, se a mensagem e a chave pública, ambas corretas, forem entregues ao algoritmo de verificação, μ' será idêntico a μ. Contudo, no momento da assinatura, os coeficientes do vetor $\boldsymbol{w}_1$ são atribuídos com os valores altos (*high bits*) do vetor resultante de $A\boldsymbol{y}$. Enquanto isso, os coeficientes de $\boldsymbol{w}'_1$, de forma distinta, são atribuídos *aproximadamente* com os valores altos do vetor resultante de $A\boldsymbol{z} - c\boldsymbol{t}_1 \cdot 2^d$. Logo, exige-se uma explicação desta equivalência que permite verificar assinaturas com êxito neste esquema criptográfico.

A *aproximação* mencionada no trecho anterior vem do método $UseHint_q$. Neste método entrega-se como entrada o vetor de polinômios resultante da expressão $A\boldsymbol{z} - c\boldsymbol{t}_1 \cdot 2^d$. Dessa forma, o método realiza correções nos valores altos deste vetor com auxílio de dicas computadas após a construção de $\boldsymbol{w}_1$. Uma dica permite ao esquema identificar que os valores altos estão errôneos em até uma unidade. Ou seja, com base na dica, um valor alto obtido é mantido, somados em 1 unidade ou subtraído em 1 unidade. Assim, apesar da expressão que verifica a integridade da assinatura ser distinta da expressão $A\boldsymbol{y}$ presente na construção da assinatura, como será mostrado, a diferença dos valores altos das duas expressões nunca será maior do que 1. E as correções de erros com dicas contidas na assinatura são suficientes para obter a igualdade da segunda operação lógica que valida a assinatura.

Para mostrar a diferença dos valores altos das duas expressões, primeiro será apresentado como elas estão relacionadas. Isto é, como são distintas uma das outras. Veja que,

$$
\begin{aligned}
&A\boldsymbol{z} - c\boldsymbol{t}_1 \cdot 2^d = A\boldsymbol{z} - c(\boldsymbol{t} - \boldsymbol{t}_0), && \text{pela decomposição de } \boldsymbol{t};\\
&A\boldsymbol{z} - c(\boldsymbol{t} - \boldsymbol{t}_0) = A\boldsymbol{z} - c\boldsymbol{t} + c\boldsymbol{t}_0, && \text{por distribuição;}\\
&A\boldsymbol{z} - c\boldsymbol{t} + c\boldsymbol{t}_0 = A\boldsymbol{z} - c(A\boldsymbol{s}_1 + \boldsymbol{s}_2) + c\boldsymbol{t}_0, && \text{pela definição de } \boldsymbol{t};\\
&A\boldsymbol{z} - c(A\boldsymbol{s}_1 + \boldsymbol{s}_2) + c\boldsymbol{t}_0 = A\boldsymbol{z} - Ac\boldsymbol{s}_1 - c\boldsymbol{s}_2 + c\boldsymbol{t}_0, && \text{por distribuição;}\\
&A\boldsymbol{z} - Ac\boldsymbol{s}_1 - c\boldsymbol{s}_2 + c\boldsymbol{t}_0 = A(\boldsymbol{y} + c\boldsymbol{s}_1) - Ac\boldsymbol{s}_1 - c\boldsymbol{s}_2 + c\boldsymbol{t}_0, && \text{pela definição de } \boldsymbol{z};\\
&A(\boldsymbol{y} + c\boldsymbol{s}_1) - Ac\boldsymbol{s}_1 - c\boldsymbol{s}_2 + c\boldsymbol{t}_0 = A\boldsymbol{y} + Ac\boldsymbol{s}_1 - Ac\boldsymbol{s}_1 - c\boldsymbol{s}_2 + c\boldsymbol{t}_0, && \text{por distribuição;}\\
&A\boldsymbol{y} + Ac\boldsymbol{s}_1 - Ac\boldsymbol{s}_1 - c\boldsymbol{s}_2 + c\boldsymbol{t}_0 = A\boldsymbol{y} + c(-\boldsymbol{s}_2 + \boldsymbol{t}_0);\\
&A\boldsymbol{y} + c(-\boldsymbol{s}_2 + \boldsymbol{t}_0) = A\boldsymbol{y} + c(\boldsymbol{t}_0 - \boldsymbol{s}_2).
\end{aligned}
$$

Portanto, $A\boldsymbol{z} - c\boldsymbol{t}_1 \cdot 2^d = A\boldsymbol{y} + c(\boldsymbol{t}_0 - \boldsymbol{s}_2)$ e a diferença desta expressão com a expressão $A\boldsymbol{y}$

computada ao assinar a mensagem é $c(\boldsymbol{t}_0 - \boldsymbol{s}_2)$.

Conforme a definição de c, cada um de seus coeficientes pode assumir os valores -1, 0 e 1. Portanto, nos casos em que um coeficiente de $c \neq 0$, podemos enxergar a expressão $A\boldsymbol{y} + c(\boldsymbol{t}_0 - \boldsymbol{s}_2)$ não como uma adição do produtos de coeficientes, mas, como uma adição ou subtração de um coeficiente de $A\boldsymbol{y}$ com um coeficiente de $\boldsymbol{t}_0 - \boldsymbol{s}_2$. Com isso, para ver que os valores altos nunca serão editados em uma quantidade não corrigível pelas dicas fornecidas na assinatura, deve-se mostrar que para qualquer coeficiente de $A\boldsymbol{y}$ e para qualquer coeficiente de $\boldsymbol{t}_0 - \boldsymbol{s}_2$, os valores dos coeficientes de $\boldsymbol{t}_0 - \boldsymbol{s}_2$ nunca serão grandes o suficiente para modificar os valores altos dos coeficientes de $A\boldsymbol{y}$ em duas ou mais unidades.

Para demonstrar isso, será apresentado o caso em que o coeficiente de $A\boldsymbol{y}$ encontra-se na borda superior e qualquer adição resulta em uma modificação no atual valor alto. Porém, uma demostração análoga pode ser feita com a borda inferior. Considere dois vetores $\boldsymbol{w}_1 = \text{HighBits}_q(A\boldsymbol{y}, 2\gamma_2)$ e $A\boldsymbol{y}$. Pela definição de decomposição, para todo coeficiente r_1 de $\boldsymbol{w}_1$ e para todo coeficiente r de $A\boldsymbol{y}$ temos que $r = r_1 \cdot 2\gamma_2 + r_0$, onde $r \in \mathbb{Z}_q$, $r_1 \in \{0, 1, \cdots, (q-1)/2\gamma_2\}$ é um número inteiro e $r_0 \in (-\gamma_2, \gamma_2]$ também é um número inteiro. A Figura 7 demonstra o comportamento da decomposição por $\alpha = 2\gamma_2$.

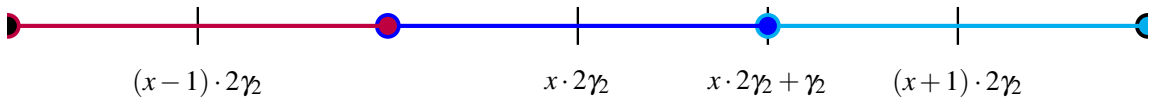

Figura 7 – Decomposição de valores com parâmetro de centralização $\alpha = 2\gamma_2$. A região em vermelho representa o intervalo onde o valor alto é $x-1$; a região em azul representa o intervalo onde o valor alto é x; finalmente a região ciana representa o intervalo onde o valor alto é $x+1$.

Considere um coeficiente $r = x \cdot 2\gamma_2 + \gamma_2$ de Ay. Veja que, ao somar 1 e r, temos que $(r+1) = x \cdot 2\gamma_2 + (\gamma_2 + 1)$. Mas, pela definição de decomposição, esta não é uma possível decomposição de $r+1$. O valor baixo deve estar contido no intervalo $(-\gamma_2, \gamma_2]$. Portanto, a decomposição obtida é $(r+1) = (x+1) \cdot 2\gamma_2 + (-\gamma_2 + 1)$ e ocorre um incremento no valor alto x. Um incremento é aceitável conforme a descrição do esquema, pois pode ser posteriormente corrigido pelas dicas. Agora, veja que ao adicionar $2\gamma_2$ a $r+1$, ocorrerá uma nova alteração no valor alto decomposto. Ou seja, $(r+1+2\gamma_2) = (x+1) \cdot 2\gamma_2 + (\gamma_2 + 1)$ não é uma possível decomposição de $r+1+2\gamma_2$. (Também pela restrição de intervalo.) Logo, a decomposição correta é $(r+1+2\gamma_2) = (x+2) \cdot 2\gamma_2 + (-\gamma_2 + 1)$, o que resulta em uma nova modificação no valor alto do coeficiente. Como consequência, encontramos que, se o resultado de $t_0 - s_2$ possuir um valor igual ou maior que $2\gamma_2 + 1$, existe possibilidade do valor alto de um coeficiente ser modificado em mais de uma unidade. Ao observar os parâmetros descritos nos níveis de segurança do esquema Dilithium, a expressão $\boldsymbol{t}_0 - \boldsymbol{s}_2$, cujo maior valor possível é $\gamma_2 + \eta$, nunca possuirá um valor desta magnitude. Assim, a comparação lógica verifica corretamente a assinatura.

Ainda mais, $\boldsymbol{t}_0 - \boldsymbol{s}_2$ nunca passará γ_2. Se o valor alto de um coeficiente for incrementado em 1 ao ser adicionado o respectivo coeficiente de $\boldsymbol{t}_0 - \boldsymbol{s}_2$ a $A\boldsymbol{y}$, então o valor baixo

resultante será negativo ou zero. Da mesma forma, se o valor alto foi decrementado em 1 pela mesma adição, então o novo valor baixo será positivo. Isto permite que a dica armazene somente a necessidade de uma correção. Mesmo que a dica não sinalize um incremento ou um decremento nos valores altos de um coeficiente, a correção pode ser inferida pelos valores baixos de $A\boldsymbol{y} + c(\boldsymbol{t}_0 - \boldsymbol{s}_2)$.

4.6 SEGURANÇA DO ESQUEMA

Nesta seção serão descritos os problemas que, se solucionados, quebram o esquema de assinatura estudado e permitem a construção da assinatura válida de uma mensagem que não tenha sido assinada pelo proprietário da chave privada. Abaixo são listados os principais ataques em esquemas de assinatura digital.

Substituição de mensagem. Estes ataques consistem em utilizar uma assinatura criada pelo proprietário da chave privada e substituir por uma nova mensagem com a mesma assinatura. Assim, veja que o $\tilde{c} := \mathrm{H}(\mu \| \boldsymbol{w}_1)$ é comparado com $\mathrm{H}(\mu' \| \boldsymbol{w}_1')$. Neste caso, $\mu := \mathrm{H}(\mathrm{H}(\rho \| \boldsymbol{t}_1) \| M)$ para ρ e $\boldsymbol{t}_1$ da chave pública e M é a mensagem assinada. Portanto, assumindo que a respectiva chave pública seja utilizada na verificação, modificar a mensagem assinada e manter o resultado válido de verificação equivale a encontrar uma mensagem M' tal que $\tilde{c} = \mathrm{H}(\mu' \| \boldsymbol{w}_1')$. Onde $\mu' := \mathrm{H}(\mathrm{H}(\rho \| \boldsymbol{t}_1) \| M')$ e $\boldsymbol{w}_1' = \boldsymbol{w}_1$ é o vetor construído no momento de verificação. Em outras palavras, um ataque de substituição de mensagem equivale a encontrar uma colisão em uma função de resumo criptográfico resistente à colisão. Em uma última nota, é importante perceber que μ é consequência do resumo da chave pública utilizada na verificação. Portanto, nos ataques subsequentes, assume-se que os parâmetros originados pela chave pública são fixos. Isto ocorre pois, uma vez que a chave pública é modificada, o ataque ter sucesso implica que foi encontrado uma colisão de resumo criptográfico em H.

Recuperação da chave privada. Uma segunda característica de esquemas de assinatura envolve ser seguro contra ataques de recuperação da chave privada. Neste ataque, o atacante utiliza informações da chave pública e assinaturas previamente geradas para obter a chave privada do assinante. Em certos esquemas de assinatura, ataques como este são bem sucedidos devido à forte relação matemática entre a chave privada, a assinatura e a mensagem. Contudo, a heurística *Fiat Shamir with Aborts*, o qual o Dilithium se baseia, utiliza da técnica de *abort* para obter assinaturas desconexas da mensagem. Isto implica que ao olhar para a assinatura e a mensagem assinada, não obtém-se nenhuma informação útil sobre a chave privada, mantendo o esquema seguro. As duas condições lógicas foram discutidas na Seção 4.4.

Apesar disso, o esquema ainda requer evidência de que, sem uma mensagem, a chave pública não expõe segredo da chave privada. Veja que a relação matemática da chave pública e chave privada é dada pela expressão $\boldsymbol{t} := A\boldsymbol{s}_1 + \boldsymbol{s}_2$. Assim, esta relação é, por definição, o próprio problema de reticulados MLWE. Neste caso, o par $(A, \boldsymbol{t})$ representa a entrada, e a saída da função definida por $\boldsymbol{s}_1$ e $\boldsymbol{s}_2$ representa o vetor de erro. Assim, encontrar a chave privada e

resolver o problema MLWE consiste em encontrar $(\boldsymbol{s}_1, \boldsymbol{s}_2)$ que melhor mapeia a entrada à saída. Uma vez que existem fortes conjecturas sobre a dificuldade de solucionar o MLWE, existem também fortes conjecturas sobre a dificuldade de recuperar a chave privada deste esquema. Vale notar que o vetor $\boldsymbol{t}_0$ da chave privada não necessita ser secreto. Caso o adversário conheça o par $(\boldsymbol{t}_1, \boldsymbol{t}_0)$ e compute $\boldsymbol{t}$, ele ainda precisaria resolver o problema MLWE para encontrar $\boldsymbol{s}_1$ e $\boldsymbol{s}_2$. O motivo da decomposição de $\boldsymbol{t}$ é comprimir ainda mais a chave pública do esquema.

Com isso, obtém-se a propriedade de *witness-indistinguishable*. A propriedade informa que, dado duas chaves privadas possíveis s e s', ambas chaves são igualmente prováveis de serem a chave privada do assinante. Devido a esta propriedade, um adversário que queira descobrir a chave privada por meio de assinaturas construídas, como comentado, é incapaz de identificar com exatidão qual das possíveis chaves realmente foi utilizada para construir a assinatura.

Construção indevida de assinaturas digitais. Os últimos ataques a serem discutidos são ataques que constroem assinaturas válidas de maneira indevida. Nestes ataques, o atacante, sem ter conhecimento da chave privada, consegue montar uma assinatura considerada válida ao ser verificado pela correspondente chave pública do usuário. Um ataque bem sucedido equivale a solucionar uma instância difícil do problema *SelfTargetMSIS* definido abaixo. Além disso, pode-se mostrar que, se existir um algoritmo probabilístico que solucione o *SelfTargetMSIS* em tempo praticável, então existe um algoritmo probabilístico que soluciona os piores casos do problema MSIS (Definição 2.2.22) em tempo praticável. Portanto, devido a credibilidade da dificuldade de MSIS, conjectura-se ser computacionalmente impraticável construir uma assinatura de maneira indevida no Dilithium. Isto é demonstrado ao reduzir o MSIS ao *SelfTargetMSIS* utilizando o *forking lemma* e somente é aprofundada na seção seguinte.

Definição 4.6.1 (*SelfTargetMSIS*)**.** Dada uma função de resumo criptográfico H e um vetor x, o problema *SelfTargetMSIS* consiste em encontrar um par $z' := (z, c)$ tal que $c = \mathrm{H}(x \| f(z'))$.

O *SelfTargetMSIS* foi definido de forma genérica. Quando se trata do esquema Dilithium, deve-se representar os termos de construção e verificação de assinaturas. Primeiro, considere $\boldsymbol{v} = A\boldsymbol{y} - (A\boldsymbol{z} - c\boldsymbol{t}_1 \cdot 2^d) - \mathrm{LowerBits}_q(A\boldsymbol{y}, 2\gamma_2)$. O resultado de $\boldsymbol{v}$ corrige a diferença de $\boldsymbol{w}_1$ obtido no momento da assinatura com $\boldsymbol{w}_1'$ construído na verificação. Ainda, permite isolar os $\mathrm{HighBits}_q(A\boldsymbol{y}, 2\gamma_2)$ multiplicados pelo parâmetro de centralização $2\gamma_2$ pela definição de decomposição. Ou seja,

$$2\gamma_2 \cdot \mathrm{HighBits}_q(A\boldsymbol{y}, 2\gamma_2) = A\boldsymbol{z} - c\boldsymbol{t}_1 \cdot 2\gamma_2 + \boldsymbol{v}.$$

Adiante, podemos representar a mesma relação de validade de assinatura como

$$
\begin{aligned}
A\boldsymbol{z} - c\boldsymbol{t}_1 \cdot 2\gamma_2 + \boldsymbol{v} &= A\boldsymbol{z} - c(\boldsymbol{t} - \boldsymbol{t}_0) + \boldsymbol{v}, && \text{pela decomposição de } \boldsymbol{t};\\
A\boldsymbol{z} - c(\boldsymbol{t} - \boldsymbol{t}_0) + \boldsymbol{v} &= A\boldsymbol{z} - c\boldsymbol{t} + (c\boldsymbol{t}_0 + \boldsymbol{v}), && \text{por distribuição;}\\
A\boldsymbol{z} - c\boldsymbol{t} + (c\boldsymbol{t}_0 + \boldsymbol{v}) &= A\boldsymbol{z} - c\boldsymbol{t} + \boldsymbol{u}, && \text{ao considerar } \boldsymbol{u} = c\boldsymbol{t}_0 + \boldsymbol{v}.
\end{aligned}
$$

Da expressão acima, como apresentado por Ducas et al. (2021), tem-se que

$$
\|\boldsymbol{u}\|_\infty \leq \|c\boldsymbol{t}_0\|_\infty + \|\boldsymbol{v}\|_\infty \leq \|c\|_1 \cdot \|\boldsymbol{t}_0\|_\infty + \|\boldsymbol{v}\|_\infty \leq \tau \cdot 2^{d-1} + 2\gamma_2 + 1. \tag{4.2}
$$

Assim, ao considerar um resumo criptográfico de uma mensagem μ, uma chave púbica $pk = (\rho, \boldsymbol{t}_1)$, de onde pode-se obter $A = \text{ExpandA}(\rho)$ e que a função em evidência seja $f(\boldsymbol{z}, c, \boldsymbol{u}) = A\boldsymbol{z} - c\boldsymbol{t} + \boldsymbol{u}$; o problema de forjar uma assinatura equivale a solucionar o *SelfTargetMSIS* no formato

$$
\text{H}\left(\mu \| \tfrac{1}{2\gamma 2} \cdot [\, A \mid -\boldsymbol{t} \mid I \,] \cdot \begin{bmatrix} \boldsymbol{z} \\ c \\ \boldsymbol{u} \end{bmatrix} \right) = c. \tag{4.3}
$$

Como (A, t) são obtidos de maneira aleatória e uniforme, a definição de *SelfTargetMSIS* se mantém correta. Uma vez que o problema é solucionado, encontrando algum $z \in (R_q)^l$, $c \in B$ e $u \in (R_q)^k$, a assinatura será válida se e somente se

- $\|\boldsymbol{z}\|_\infty < \gamma_1 - \beta$;
- $\text{H}(\mu \| \text{UseHint}_q(\boldsymbol{h}, A\boldsymbol{z} - c\boldsymbol{t}_1 \cdot 2^d, 2\gamma 2)) = c$;
- *A quantidade de 1's em c é menor que* ω;
- $\|c\|_\infty = 1$;
- $\|\boldsymbol{u}\| \leq \tau \cdot 2^{d-1} + 2\gamma_2 + 1$.

A redução pelo *forking lemma* é utilizada em esquemas que seguem a heurística de *Fiat-Shamir* há diversas décadas. Assim, tem-se conhecimento de que a redução apresentada não é rigorosa (*non-tight reduction*). Ainda, estes esquemas têm desconsiderado a falta de rigorosidade de reduções pelo *forking lemma*. Porém, mesmo assim mantém-se credibilidade sobre a segurança destes esquemas devido a dificuldade de problemas análogos à Equação 4.3. Para obter um esquema como o Dilithium que não dependa de reduções não rigorosas, o trabalho de Kiltz, Lyubashevsky e Schaffner (2018) pode ser consultado. Este trabalho providencia parâmetros que permitem construir uma versão do Dilithium que é *information-theoretically hard* e depende somente das fortes conjecturas sobre a dificuldade do problema LWE. Porém, estes parâmetros resultam em chaves públicas 5 vezes maiores e assinaturas duas vezes maiores.

Existem outros métodos de ataques que tentam demonstrar insegurança no esquema quanto a construção de assinaturas de maneira indevida. Um exemplo disso são ataques que

exploram características algébricas dos reticulados para forjar assinaturas válidas. Contudo, exemplos como este não fornecem evidências de serem aplicáveis a reticulados modulares, utilizados no esquema Dilithium.

4.7 REDUÇÃO DE MSIS PARA SELFTARGETMSIS

Esta seção exibirá a relação entre o problema MSIS e o problema SelfTargetMSIS. Devido a redução não rigorosa de um problema a outro, tem-se que a dificuldade de forjar uma assinatura, que equivale a resolver o problema SelfTargetMSIS, é de complexidade igual ou superior à complexidade do problema MSIS. Assim, o esquema encontra-se seguro contra ataques de construção indevida de assinaturas devido a fortes conjecturas sobre problemas de reticulados. A relação do *SelfTargetMSIS* com o problema MSIS se dá por uma redução que utiliza o *forking lemma*.

Definição 4.7.1 (*Forking Lemma*)**.** Considere $k \in \mathbb{Z}$. Considere um adversário e um oráculo, ambos equivalentes a uma máquina de Turing determinística que executa um algoritmo probabilístico. Portanto, estas máquinas possuem duas fitas, uma fita contendo a entrada da execução e outra fita somente leitura contendo valores aleatórios que orientam a execução probabilística do algoritmo. Dessa forma, considere τ^i_{Adv} e τ^i_{Orc} estados da fita somente leitura presente no adversário e no oráculo, anterior à execução da requisição i, para algum $1 \leq i \leq k$. Considere que o adversário com a fita τ_{Adv} realize k requisições para o oráculo com a fita τ_{Orc} e eventualmente obtém uma propriedade p sobre uma requisição i, para algum $1 \leq i \leq k$. Se esse adversário retornar para o estado i e reconfigurar a fita somente leitura do oráculo para uma nova fita aleatória e retomar as requisições, como foi feito anteriormente, então existe uma probabilidade não negligenciável de que este adversário obtenha a mesma propriedade sobre a requisição i novamente.

O *forking lemma* expressa que, visto que o adversário não editou sua fita somente leitura, mesmo após retornar ao estado anterior, então existe uma probabilidade não negligenciável de que ao replicar as execuções daquele ponto, mesmo que o oráculo tenha uma fita modificada, a propriedade será obtida novamente sobre a requisição i. Nesta seção a redução que utiliza o *forking lemma* será demonstrada de maneira informal, permitindo enxergar a relação entre os dois problemas mencionados. Sejam $k, l \in \mathbb{Z}$ dois números inteiros positivos, e $A \in (R_q)^{k \times l}$ uma sequência de vetores. Considere um oráculo equivalente a uma máquina de Turing probabilística que, dados $\mu \in \{0,1\}^*$ e $\boldsymbol{w} \in (R_q)^k$, computa $\mathrm{H}(\mu \| \boldsymbol{w})$ de maneira aleatória para alguma configuração. Assim, um adversário que objetiva encontrar duas entradas que geram colisão de resumo criptográfico em $f_{[\,I\,|\,A\,]} \in \mathcal{H}$ requisitará do oráculo uma sequência de resumos criptográficos

$$\mathrm{H}(\mu_1 \| \boldsymbol{w}_1), \mathrm{H}(\mu_2 \| \boldsymbol{w}_2), \ \ldots, \mathrm{H}(\mu_k \| \boldsymbol{w}_k) = c_1, c_2, \ \ldots, c_k.$$

Em algum momento o adversário que soluciona o problema *SelfTargetMSIS* obtém um par $(\mu_i, \boldsymbol{y})$ tal que

$$\boldsymbol{y} = \begin{bmatrix} r \\ c_i \end{bmatrix} \text{ e } c_i = \mathrm{H}(\mu_i \| \boldsymbol{w}_i) = \mathrm{H}(\mu_i \| [I \mid A] \cdot \boldsymbol{y})).$$

Devido ao *forking lemma*, o adversário volta atrás para o momento que a consulta $c_i = \mathrm{H}(\mu_i \| \boldsymbol{w}_i)$ foi realizada e reconfigura o oráculo com uma nova fita, que passará a retornar resultados diferentes ao computar H para as entradas. Após o adversário retomar as requisições como anteriormente, existe uma chance não negligenciável de que o adversário construa um novo par $(\mu_i, \boldsymbol{y}')$ tal que

$$\boldsymbol{y}' \neq \boldsymbol{y},\ \boldsymbol{y}' = \begin{bmatrix} r' \\ c_i' \end{bmatrix} \text{ e } c_i' = \mathrm{H}(\mu_i \| \boldsymbol{w}_i) = \mathrm{H}(\mu_i \parallel [I \mid A] \cdot \boldsymbol{y}').$$

O fato da máquina ter sido reconfigurada implica que $c_i \neq c_i'$ e $\boldsymbol{y} \neq \boldsymbol{y}'$. Além disso, como a função H possui a propriedade de resistência à colisão, então $c_i = \mathrm{H}(\mu_i \| \boldsymbol{w}_i) = \mathrm{H}(\mu_i \parallel [\, I \mid A \,] \cdot \boldsymbol{y})$ implica que $\boldsymbol{w}_i = [\, I \mid A \,] \cdot \boldsymbol{y}$. Da mesma forma, pela propriedade de resistência à colisão, $c_i' = \mathrm{H}(\mu_i \| \boldsymbol{w}_i) = \mathrm{H}(\mu_i \parallel [\, I \mid A \,] \cdot \boldsymbol{y}')$ implica que $\boldsymbol{w}_i = [\, I \mid A \,] \cdot \boldsymbol{y}'$. Como consequência, temos que $[I \mid A] \cdot (\boldsymbol{y} - \boldsymbol{y}') = [I \mid A] \cdot \boldsymbol{y} - [I \mid A] \cdot \boldsymbol{y}' = w_i - w_i = 0$ e $\boldsymbol{y} - \boldsymbol{y}'$ é uma solução para uma instância difícil do problema MSIS cuja matriz é $[\, I \mid A \,]$. Como A é escolhido de forma uniforme e aleatória, isto demonstra um redução do MSIS para o *SelfTargetMSIS*.

4.8 VARIANTES DO DILITHIUM

Existem variações para os algoritmos do esquema Dilthium que foram descritos neste trabalho. Algumas destas variações já foram brevemente mencionadas. Uma das variações consiste em reduzir o tamanho da chave privada e, em troca, incrementar o esforço computacional necessário para assinar mensagens. Isto é obtido ao descrever a chave privada $sk = \zeta$ ao invés de $sk = (\rho, K, tr, \boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{t}_0)$. A semente ζ permite gerar novamente toda a chave privada no momento da assinatura. De ζ, obtêm-se as sequências de bytes ρ, ρ' e K. A sequência ρ permite expandir A, enquanto a sequência ρ' permite expandir $\boldsymbol{s}_1$ e $\boldsymbol{s}_2$. Com estes valores, consegue-se computar $\boldsymbol{t} = A\boldsymbol{s}_1 + \boldsymbol{s}_2$ e, por consequência, computar $(\boldsymbol{t}_1, \boldsymbol{t}_0) := \mathrm{Decompose}_q(\boldsymbol{t}, 2\gamma_2)$. Ainda, pode-se computar o resumo criptográfico $tr = \mathrm{H}(\rho \| \boldsymbol{t}_1)$. Nesta variação, a chave privada possui o tamanho de somente 32 bytes, em comparação com o mínimo de 1312 bytes derivados dos níveis de segurança.

Uma outra variação do esquema computa $\rho' \leftarrow \{0,1\}^{512}$ de maneira uniforme e aleatória ao invés de $\rho' = \mathrm{H}(K \| \mu)$ na construção da assinatura. A sequência de bytes ρ' é utilizada para expandir y, portanto, utilizar um ρ' aleatório gera uma versão não-determinística do esquema. Esta variação fornece segurança contra certos ataques de *side-channel*, que tentam obter informações sensíveis da chave privada considerando propriedades subsequentes de um comportamento determinístico do algoritmo de assinatura. Um outro benefício da execução

não-determinística do esquema é a construção de assinaturas sem o vazamento de informações da mensagem assinada. Apesar do processo de abortar e reiniciar a operação de construção de assinaturas não acarretar em vazamento de informações da chave privada por análises de tempo, na versão determinística descrita, ainda ocorre o vazamento de informações da mensagem assinada. Isto não é algo preocupante em esquemas de assinatura pois o esquema não fornece a propriedade de sigilo da mensagem. Porém, um signatário com intenção de assinar uma mensagem e mantê-la em segredo por um tempo indeterminado deve utilizar a versão não-determinística do esquema. Isto é, utilizar a versão em que assinaturas não vazam informações da mensagem assinada pois o número de *aborts* ocorridos no algoritmos não deriva da mensagem assinada.

No esquema pós-quântico Dilithium, funções de resumo criptográfico são utilizadas tanto para obter valores uniformemente pseudo-aleatórios quanto para computar o resumo criptográfico de estruturas. Percebe-se isto na geração de par de chaves, onde ζ é entregue à função de resumo criptográfico H para obter ρ, ρ' e K uniformemente pseudo-aleatórios. Nesta mesma função, H é utilizado para computar o resumo criptográfico da chave pública, contida na chave privada. Na descrição fornecida, a função de resumo criptográfico, que difere em cada situação, é o SHAKE-128 ou SHAKE-256. Porém, apesar da padronização destas funções, elas possuem um grande custo computacional. Assim, na proposta do Dilithium submetida à terceira fase do processo de padronização do NIST, é fornecida uma variação que utiliza o procedimento de expansão pseudo-aleatório via AES ao invés das funções de SHAKE. Como mencionado, o objetivo desta variação foi fornecer evidências de que o desempenho computacional do esquema será aprimorado uma vez que o suporte de *hardware* para as funções do SHAKE for mais acessível, como já ocorre no AES.

Um último detalhe a ser mencionado talvez não deva ser caracterizado como uma variação do esquema. Contudo, o Dilithium fornece novos níveis de segurança diferente dos níveis 2, 3 e 5 mencionados anteriormente. Estes novos níveis são nomeados *1–*, *1-*, *5+* e *5++*. Estas versões são inferiores ou superiores aos níveis de segurança requisitados pelo NIST. Assim, as versões *1-* e *1–* devem funcionar como um ambiente de criptoanálise e operam como um mecanismo de alerta. Caso algum destes níveis inferiores tenha sua segurança quebrada, conclui-se que os problemas MSIS e MLWE acabaram sendo mais fáceis do que se foi conjecturado. Como mencionado por Ducas et al. (2021), o nível de segurança *1–* é concebível de ser quebrado, o que provavelmente ocorrerá daqui alguns anos. Mas, caso o nível de segurança *1-* seja quebrado, realmente demonstra-se um sinal de que o nível 2 proposto ao NIST encontra-se em risco de não ser mais seguro. Os níveis superiores *5+* e *5++* incrementam a segurança do esquema sobre ataques com dificuldade relacionado a problemas de reticulado. Ataques de substituição de mensagem, por exemplo, não tornam-se mais difíceis nestes níveis superiores. Os níveis superiores fornecem uma direção para qual a dificuldade do esquema deve seguir, se o nível de segurança 3 correr risco de não ser mais seguro. A tabela 4 apresenta os parâmetros para os níveis *1–*, *1-*, *5+* e *5++*.

Nível de Segurança do NIST:	1–	1-	5+	5++
Parâmetros				
n [*quantidade de coeficientes*]	256	256	256	256
q [módulo]	8380417	8380417	8380417	8380417
d [bits ignorados de t]	10	13	13	13
τ [*quantidade de* ± 1*'s no vetor* c]	24	30	60	60
γ_1 [intervalo de distribuição]	2^{17}	2^{17}	2^{19}	2^{19}
γ_2 [parâmetro de centralização]	$(q-1)/128$	$(q-1)/128$	$(q-1)/32$	$(q-1)/32$
(k,l) [comprimentos]	$(2,2)$	$(3,3)$	$(9,8)$	$(10,9)$
η [intervalo de distribuição]	6	3	2	2
β $[\tau \cdot \eta]$	144	90	120	120
ω [*quantidade de 1's na dica*]	10	80	85	90

Tabela 4 – Parâmetros para níveis secundários de segurança do NIST.

4.9 CONTRIBUIÇÃO DE IMPLEMENTAÇÃO

Implementações do esquema de assinatura digital Dilithium continuam a ser aprimoradas e estudadas no âmbito da ciência da computação. Apesar de esquemas criptográficos possuírem provas abstratas de matemática, que fornecem evidências de segurança, implementações ingênuas destes esquemas ainda podem ser inseguras contra ataques de *side-channel*. Nestes ataques, analisa-se propriedades arquiteturais de uma ou várias execuções para obter dados sensíveis do esquema criptográfico. Nos esquemas de assinatura digital, ataques de *side-channel* observam características como tempo de execução e/ou gastos energéticos e tentam obter informações secretas. Por este motivo, os requisitos para padronização de um esquema, no processo pós-quântico do NIST, exigem a existência de implementações seguras tanto por software quanto por hardware.

A implementação completa do Dilithium realizada neste trabalho (BIAGE, 2022) não é segura contra ataques desta natureza. Para obter uma implementação que seja segura contra ataques de análise temporal, por exemplo, requer-se que a execução dos algoritmos ocorra em tempo constante (*constant-time*). Isto é, operações como a multiplicação de dois números inteiros arbitrários devem levar sempre o mesmo número de ciclos para serem concluídas. Outras operações regulares neste esquema criptográfico são operações modulares e divisões aritméticas. Por este motivo, operações triviais são implementadas de uma forma mais complexa do que com o uso de um símbolo singular de multiplicação, módulo ou divisão presente em linguagens de programação. Logo, o código de implementação segura deste esquema distingue-se de uma implementação de alto nível ou de um pseudo-código que o descreve.

Um segundo problema que dificulta a compreensão de esquemas criptográficos é a ausência de uma fundamentação teórica detalhada. O trabalho de Ducas et al. (2018), que apresenta o esquema Dilithium, sofreu diversas modificações após as contribuições de profissionais da área, desde sua primeira publicação. Assim, o trabalho submetido na etapa mais recente do processo de padronização do NIST encontra-se mais atualizado com o estado da arte. Contudo,

no processo de atualização, informações básicas pertencentes ao esquema foram omitidas. Um exemplo disso é a explicação de um método para obter o *centered modular reduction*, que não encontra-se na versão mais recentemente publicada.

A contribuição fornecida pela implementação realizada do esquema, apesar de insegura, explica por uma demonstração prática o funcionamento do esquema criptográfico estudado nesta monografia. Deste modo, perceba pela Figura 8 e pela Figura 9 os detalhes previamente comentados ao ver a diferença entre método *Decompose* presente em ambas as implementações. Entre os principais benefícios da implementação fornecida neste trabalho, quando comparada com a implementação tornada pública pelos autores do Dilithium (SEILER, 2017), encontra-se que:

(i) os métodos implementados são assemelhantes ao pseudo-código apresentado e explicações fornecidas por Ducas et al. (2021);

(ii) a implementação do *Number Theoretic Transform* se assemelha à explicação do Teorema Chinês do Resto, fornecidas neste e em outros trabalhos;

(iii) a multiplicação de vetores, equivalente a transformação à forma NTT, multiplicação ponto-a-ponto e respectiva transformação à forma inversa são explicitamente indicados;

(iv) o comprimento de vetores, o grau de polinômios e o divisor dos módulos são explicitamente referenciados nas variáveis;

(v) percebe-se com clareza a estrutura algébrica utilizada.

Portanto, acredita-se que esta implementação se comporta melhor aos objetivos do trabalhos, de fornecer de uma forma a melhorar a compreensão um estudo sobre esquemas de assinatura digital baseados em reticulados.

Para evidenciar o funcionamento do código implementado, os testes realizados compilam também a implementação do Dilithium dos autores do esquema. Assim, entrega a ambas as implementações a mesma semente, gera milhares de chaves, constrói e verifica milhares de assinaturas, obtendo resultados *idênticos* nos processos. O objetivo é demonstrar que o comportamento inteiro do esquema é o mesmo, apesar da implementação ser de alto nível e utilizar certos componentes da orientação a objetos do C++. O motivo dos algoritmos pertencentes ao esquema serem executados muitas vezes é, principalmente, para garantir que o comportamento de *abort* esteja correto na implementação fornecida. Percebeu-se empiricamente que necessita-se de assinar centenas de assinaturas para que certas condições de *abort* ocorram. Assim, acredita-se que a construção de milhares de assinaturas se torna o suficiente para evidenciar o funcionamento do código implementado.

```
int32_t decompose(int32_t *a0, int32_t a) {
  int32_t a1;
  a1  = (a + 127) >> 7;

  a1  = (a + 127) >> 7;
#if GAMMA2 == (Q-1)/32
  a1  = (a1*1025 + (1 << 21)) >> 22;
  a1 &= 15;
#elif GAMMA2 == (Q-1)/88
  a1  = (a1*11275 + (1 << 23)) >> 24;
  a1 ^= ((43 - a1) >> 31) & a1;
#endif

  *a0  = a - a1*2*GAMMA2;
  *a0 -= (((Q-1)/2 - *a0) >> 31) & Q;
  return a1;
}
```

Figura 8 – Método de suporte *Decompose* tornado público pelos autores do esquema CRYSTALS-Dilithium.

```
template <unsigned int Q>
int32_t cmod(int32_t r, int32_t alpha) {
    /* "Centered modular reduction" explicado na primera publicação do Dilithium. */
    r = r % alpha;
    r = r - (alpha/2 + 1);
    int32_t shift = (r >> 31);
    r = r + (shift & alpha);
    r = r - (alpha/2 - 1);
    return r;
}

template <unsigned int Q>
std::pair<int32_t, int32_t> decompose(int32_t w, int32_t alpha) {
    w = ((int64_t) w + Q) % Q;
    int32_t w0, w1;
    w0 = cmod<Q>(w, alpha);
    if (w - w0 == (Q - 1)) {
        w1 = 0;
        w0 = w0 - 1;
    } else {
        w1 = (w - w0) / alpha;
    }
    return std::make_pair(w1, w0);
}
```

Figura 9 – Método de suporte *Decompose* implementado neste trabalho (com pequenas edições).

5 CONCLUSÃO

Problemas de reticulados têm sido estudados há diversas décadas. Três dos mais importantes problemas de reticulados são o *Shortest Vector Problem* (SVP), o *Approximate Shortest Vector Problem* (SVP_γ) e o *Closest Vector Problem* (CVP). A dificuldade de solucionar o CVP é bem definida para qualquer norma ℓ_p (MICCIANCIO; GOLDWASSER, 2002). Enquanto isso, os problemas envolvendo o SVP são estudados somente em cenários específicos. Dessa forma, existem provas da dificuldade do SVP dentro da norma ℓ_∞ (DINUR, 2002) e conjectura-se que o problema se mantém complexo nas demais normas (EMDE-BOAS, 1981) (MICCIANCIO, 2001). Com isso, Ajtai (1996) demonstrou a primeira primitiva criptográfica baseada em reticulados com evidência de segurança. Desde então, reticulados se tornaram atrativos para a construção de primitivas criptográficas. Trabalhos posteriores, como (GOLDREICH; GOLDWASSER; HALEVI, 2011) e (REGEV, 2005), relacionaram os problemas *Short Integer Solution* (SIS) e *Learning with Errors* (LWE) aos três problemas fundamentais de reticulados e às primitivas criptográficas baseadas em reticulados existentes. Como consequência, os problemas SIS e LWE se transformaram em ótimas ferramentas para construir e demonstrar segurança de esquemas criptográficos.

Um problema ainda pertinente em esquemas com segurança baseada no SIS e LWE é o desempenho. Para contornar isso, Lyubashevsky, Peikert e Regev (2010), Lyubashevsky e Micciancio (2006), e Peikert e Rosen (2006b) demonstram evidência da dificuldade de solucionar o RSIS e RLWE, envolvendo reticulados ideais. Como reticulados modulares são uma generalização de reticulados ideais, acredita-se que os problemas MSIS e MLWE, envolvendo reticulados modulares, sejam tão ou mais complexos do que o RSIS e RLWE. Os reticulados ideais e modulares são considerados reticulados estruturados. Devido a suas estruturas, a presença de um isomorfismo contendo o anel $R = \mathbb{Z}_q[x]/\langle x^n + 1\rangle$ permite que técnicas como a *Number Theoretic Transform* (NTT) realizem de maneira eficiente diversas multiplicações sobres seus elementos. Assim, consegue-se obter esquemas com bom desempenho sem sacrificar a segurança baseada em problemas difíceis de reticulados

Neste contexto, este trabalho atingiu o objetivo de introduzir, explicar e aprofundar a fundamentação teórica desta área, ausente em diversos cursos de ciência da computação. Foi selecionado um esquema de assinatura promissor e baseado em reticulados, o Dilithium, selecionado para padronização no processo pós-quântico do NIST, para servir de ferramenta educacional. *Fiat-Shamir with Aborts* (LYUBASHEVSKY, 2009) (FIAT; SHAMIR, 1987), na qual o Dilithium foi inspirado, é uma heurística que permite construir um esquema de assinatura em cima de um esquema de identificação, dentro do contexto de reticulados. Esta heurística é desenvolvida com um padrão observado em diversos esquemas de identificação. Por este motivo, existe uma sequência de reduções que relacionam o esquema de assinatura resultante aos problemas de reticulados de maneira organizada. Logo, acredita-se que o Dilithium seja a melhor ferramenta para introduzir a criptografia baseada em reticulados a novos estudantes da área, entre os esquemas promissores participantes do processo de padronização do NIST.

Quando se trata de compreender um esquema, diversos obstáculos são encontrados. A maioria das implementações de esquemas criptográficos têm o objetivo de serem seguras contra ataques de canal lateral (*side-channel*). Por este motivo, operações triviais tornam-se bem mais complexas e a programação utilizada deixa de ser alto nível. Só que algumas técnicas utilizadas no esquema, como exemplo a NTT, são melhor absorvidas por observação de código. Assim, este trabalho fornece uma implementação do esquema que permita relacionar os trechos de código com a fundamentação teórica apresentada. Além do conteúdo apresentado, o código se torna um outro ponto de partida para a compreensão dos algoritmos que compõem o esquema de assinatura. Para que a implementação fornecida por este trabalho esteja de acordo com os objetivos previamente estabelecidos, o código utiliza aspectos como a orientação a objetos para ser familiar a novos leitores. E para demonstrar corretude da implementação, utiliza-se uma comparação direta com os resultados da implementação fornecida pelos autores do esquema.

Do esquema Dilithium, pode-se obter uma versão simplificada bastante similar ao *Fiat-Shamir with Aborts*. Porém, a versão do esquema apresentada neste trabalho é a completa. Embora mais complexa, esta versão segue os mesmos princípios da anterior. A diferença está no uso de técnicas avançadas para compressão da chave privada e chave pública; e técnicas de convolução para aprimorar o desempenho da multiplicação de polinômios. Por isso, observa-se menos similaridades com os algoritmos de *Fiat-Shamir with Aborts* apresentados. Porém, o uso destas técnicas mais avançadas permite que o esquema de assinatura possua ótimos resultados de desempenho; isto é, tamanhos reduzidos de assinaturas, chaves e menor tempo de execução. Ainda, no esquema, a segurança contra ataques de construção indevida de assinaturas é baseada na dificuldade do problema MSIS. Enquanto isso, a segurança contra a recuperação da chave privada pela chave pública é computacionalmente impraticável pela dificuldade de solucionar o MLWE. A versão do MLWE utilizada no Dilithium também contém fortes evidências de segurança, mas, os erros são obtidos de uma distribuição uniforme. Portanto, o esquema pode ser implementado com mais facilidade pela simplicidade desta distribuição. Ao contrário do Dilithium, o esquema de assinatura FALCON, que possui segurança baseada no MLWE, depende de diversas distribuições gaussianas, operações com árvores e ponto flutuante. Assim, pela maior simplicidade do esquema, apesar de conter resultados um pouco piores que o FALCON, o Dilithium foi julgado ser um esquema que melhor se encaixa nos objetivos do trabalho.

Isto não foi algo observado somente na seleção do tópico desta monografia. A partir de julho de 2022, o Dilithium encontra-se como a recomendação do NIST para a utilização de esquemas de assinaturas pós-quânticos e será um dos poucos esquemas de assinatura digital padronizados (GORJAN et al., 2022). Portanto, tem-se esperança que este trabalho introduza a criptografia baseada em reticulados e incentive novas pesquisas envolvendo principalmente o Dilithium.

REFERÊNCIAS

AJTAI, M. Generating hard instances of lattice problems (extended abstract). In: **Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing**. New York, NY, USA: ACM, 1996. (STOC '96), p. 99–108. ISBN 0-89791-785-5. Disponível em: http://doi.acm.org/10.1145/237814.237838.

BABAI, L. On lovász'lattice reduction and the nearest lattice point problem. **Combinatorica**, Springer, v. 6, n. 1, p. 1–13, 1986.

BENNETT, H.; PEIKERT, C. Hardness of bounded distance decoding on lattices in ℓ_p norms. **CoRR**, abs/2003.07903, 2020. Disponível em: https://arxiv.org/abs/2003.07903.

BIAGE, G. de C. **Dilithium Implementation**. [S.l.]: GitHub, 2022. Https://github.com/gustavobiage/dilithium-implementation.

CABARCAS, D.; GÖPFERT, F.; WEIDEN, P. Provably secure lwe encryption with smallish uniform noise and secret. **ASIAPKC 2014 - Proceedings of the 2nd ACM Workshop on ASIA Public-Key Cryptography**, 06 2014.

CRAMER, R. et al. Recovering short generators of principal ideals in cyclotomic rings. In: FISCHLIN, M.; CORON, J.-S. (Ed.). **Advances in Cryptology – EUROCRYPT 2016**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016. p. 559–585. ISBN 978-3-662-49896-5.

CRAMER, R.; DUCAS, L.; WESOLOWSKI, B. **Short Stickelberger Class Relations and application to Ideal-SVP**. 2016. Cryptology ePrint Archive, Paper 2016/885. https://eprint.iacr.org/2016/885. Disponível em: https://eprint.iacr.org/2016/885.

DINUR, I. Approximating svp∞ to within almost-polynomial factors is np-hard. **Theoretical Computer Science**, v. 285, n. 1, p. 55–71, 2002. ISSN 0304-3975. Algorithms and Complexity. Disponível em: https://www.sciencedirect.com/science/article/pii/S0304397501002900.

DUCAS, L. et al. Crystals-dilithium: A lattice-based digital signature scheme. **IACR Transactions on Cryptographic Hardware and Embedded Systems**, p. 238–268, 2018.

DUCAS, L. et al. Crystals-dilithium: A lattice-based digital signature scheme. **IACR Transactions on Cryptographic Hardware and Embedded Systems**, 2021.

DUMMIT, D. S.; FOOTE, R. M. **Abstract algebra**. [S.l.]: Wiley Hoboken, 2004. v. 3.

EMDE-BOAS, P. van. **Another NP-complete partition problem and the complexity of computing short vectors in a lattice**. Department, Univ., 1981. (Report. Department of Mathematics. University of Amsterdam). Disponível em: https://books.google.com.br/books?id=tCQiHQAACAAJ.

FIAT, A.; SHAMIR, A. How to prove yourself: Practical solutions to identification and signature problems. In: ODLYZKO, A. M. (Ed.). **Advances in Cryptology — CRYPTO' 86**. Berlin, Heidelberg: Springer Berlin Heidelberg, 1987. p. 186–194. ISBN 978-3-540-47721-1.

GIRAULT, M. An identity-based identification scheme based on discrete logarithms modulo a composite number. In: DAMGÅRD, I. B. (Ed.). **Advances in Cryptology — EUROCRYPT '90**. Berlin, Heidelberg: Springer Berlin Heidelberg, 1991. p. 481–486. ISBN 978-3-540-46877-6.

GOLDREICH, O.; GOLDWASSER, S.; HALEVI, S. Collision-free hashing from lattice problems. In: ____. **Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation: In Collaboration with Lidor Avigad, Mihir Bellare, Zvika Brakerski, Shafi Goldwasser, Shai Halevi, Tali Kaufman, Leonid Levin, Noam Nisan, Dana Ron, Madhu Sudan, Luca Trevisan, Salil Vadhan, Avi Wigderson, David Zuckerman**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. p. 30–39. ISBN 978-3-642-22670-0. Disponível em: https://doi.org/10.1007/978-3-642-22670-0_5.

GORJAN, A. et al. Status report on the third round of the nist post-quantum cryptography standardization process. In: . [S.l.]: NIST, 2022.

GROVER, L. K. A fast quantum mechanical algorithm for database search. In: **STOC '96**. [S.l.: s.n.], 1996.

GUILLOU, L. C.; QUISQUATER, J.-J. A "paradoxical" indentity-based signature scheme resulting from zero-knowledge. In: GOLDWASSER, S. (Ed.). **Advances in Cryptology — CRYPTO' 88**. New York, NY: Springer New York, 1990. p. 216–231. ISBN 978-0-387-34799-8.

HEROLD, G.; KIRSHANOVA, E.; MAY, A. On the asymptotic complexity of solving lwe. **Designs, Codes and Cryptography**, v. 86, n. 1, p. 55–83, Jan 2018. ISSN 1573-7586. Disponível em: https://doi.org/10.1007/s10623-016-0326-0.

HOFFSTEIN, J.; PIPHER, J.; SILVERMAN, J. H. Ntru: A ring-based public key cryptosystem. In: BUHLER, J. P. (Ed.). **Algorithmic Number Theory**. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998. p. 267–288. ISBN 978-3-540-69113-6.

KHOT, S. Hardness of approximating the shortest vector problem in lattices. In: **45th Annual IEEE Symposium on Foundations of Computer Science**. [S.l.: s.n.], 2004. p. 126–135.

KILTZ, E.; LYUBASHEVSKY, V.; SCHAFFNER, C. A concrete treatment of fiat-shamir signatures in the quantum random-oracle model. In: SPRINGER. **Annual International Conference on the Theory and Applications of Cryptographic Techniques**. [S.l.], 2018. p. 552–586.

LAARHOVEN, T.; MOSCA, M.; POL, J. van de. Finding shortest lattice vectors faster using quantum search. **Designs, Codes and Cryptography**, v. 77, n. 2, p. 375–400, Dec 2015. ISSN 1573-7586. Disponível em: https://doi.org/10.1007/s10623-015-0067-5.

LANGLOIS, A.; STEHLÉ, D. Worst-case to average-case reductions for module lattices. **Designs, Codes and Cryptography**, v. 75, n. 3, p. 565–599, Jun 2015. ISSN 1573-7586. Disponível em: https://doi.org/10.1007/s10623-014-9938-4.

LENSTRA, A.; LENSTRA, H.; LOVÁSZ, L. Factoring polynomials with rational coefficients. **Mathematische Annalen**, v. 261, 12 1982.

LEURENT, G.; PEYRIN, T. From collisions to chosen-prefix collisions application to full sha-1. In: SPRINGER. **Annual International Conference on the Theory and Applications of Cryptographic Techniques**. [S.l.], 2019. p. 527–555.

LEURENT, G.; PEYRIN, T. Sha-1 is a shambles: First chosen-prefix collision on sha-1 and application to the pgp web of trust. In: **29th USENIX Security Symposium (USENIX Security 20)**. [S.l.: s.n.], 2020. p. 1839–1856.

LIU, F.; LIU, Y. Avalanche of md5. **Energy Procedia**, n. 13, p. 237–246, 2011.

LIU, Y.-K.; LYUBASHEVSKY, V.; MICCIANCIO, D. On bounded distance decoding for general lattices. In: DÍAZ, J. et al. (Ed.). **Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. p. 450–461. ISBN 978-3-540-38045-0.

LYUBASHEVSKY, V. Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In: MATSUI, M. (Ed.). **Advances in Cryptology – ASIACRYPT 2009**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. p. 598–616. ISBN 978-3-642-10366-7.

LYUBASHEVSKY, V.; MICCIANCIO, D. Generalized compact knapsacks are collision resistant. In: BUGLIESI, M. et al. (Ed.). **Automata, Languages and Programming**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. p. 144–155. ISBN 978-3-540-35908-1.

LYUBASHEVSKY, V.; MICCIANCIO, D. Asymptotically efficient lattice-based digital signatures. In: CANETTI, R. (Ed.). **Theory of Cryptography**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. p. 37–54. ISBN 978-3-540-78524-8.

LYUBASHEVSKY, V.; MICCIANCIO, D. On bounded distance decoding, unique shortest vectors, and the minimum distance problem. In: . [S.l.: s.n.], 2009. p. 577–594. ISBN 978-3-642-03355-1.

LYUBASHEVSKY, V.; PEIKERT, C.; REGEV, O. On ideal lattices and learning with errors over rings. In: GILBERT, H. (Ed.). **Advances in Cryptology – EUROCRYPT 2010**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 1–23. ISBN 978-3-642-13190-5.

MICCIANCIO, D. The shortest vector problem is NP-hard to approximate to within some constant. **SIAM Journal on Computing**, v. 30, n. 6, p. 2008–2035, mar. 2001. Preliminary version in FOCS 1998.

MICCIANCIO, D. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions from worst-case complexity assumptions. In: **The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.** [S.l.: s.n.], 2002. p. 356–365.

MICCIANCIO, D.; GOLDWASSER, S. **Complexity of lattice problems: a cryptographic perspective**. [S.l.]: Springer Science & Business Media, 2002. v. 671.

MOODY, D. **The Ship has Sailed: The NIST Post-Quantum Cryptography "Competition"(Presentation)**. Asiacrypt, December 3-7, 2017 (Hong Kong, China), 2017. Disponível em: https://csrc.nist.gov/Presentations/2017/The-Ship-has-Sailed-The-NIST-Post-Quantum-Cryptog.

MOODY, D. **Round 2 of the NIST PQC "Competition- What was NIST Thinking? (Presentation)**. PQCrypto 2019 in Chongqing, China, 2019. Disponível em: https://csrc.nist.gov/Presentations/2019/Round-2-of-the-NIST-PQC-Competition-What-was-NIST.

OKAMOTO, T. Provably secure and practical identification schemes and corresponding signature schemes. In: BRICKELL, E. F. (Ed.). **Advances in Cryptology — CRYPTO' 92**. Berlin, Heidelberg: Springer Berlin Heidelberg, 1993. p. 31–53. ISBN 978-3-540-48071-6.

PEIKERT, C.; ROSEN, A. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In: SPRINGER. **Theory of Cryptography Conference**. [S.l.], 2006. p. 145–166.

PEIKERT, C.; ROSEN, A. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In: HALEVI, S.; RABIN, T. (Ed.). **Theory of Cryptography**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. p. 145–166. ISBN 978-3-540-32732-5.

REGEV, O. On lattices, learning with errors, random linear codes, and cryptography. In: . [S.l.: s.n.], 2005. v. 56, p. 84–93.

REGEV, O. The learning with errors problem (invited survey). In: **2010 IEEE 25th Annual Conference on Computational Complexity**. [S.l.: s.n.], 2010. p. 191–204.

SCHNORR, C.-P. Efficient signature generation by smart cards. **Journal of cryptology**, Springer, v. 4, n. 3, p. 161–174, 1991.

SEILER, G. **Dilithium**. [S.l.]: GitHub, 2017. Https://github.com/pq-crystals/dilithium.

SHOR, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. **SIAM Review**, v. 41, n. 2, p. 303–332, 1999. Disponível em: https://doi.org/10.1137/S0036144598347011.

STANDARDS, N. I. of; TECHNOLOGY. Advanced encryption standard (aes). In: . [S.l.]: NIST, 2001.

STANDARDS, N. I. of; TECHNOLOGY. Sha-3 standard: Permutation-based hash and extendable-output functions. In: . [S.l.]: NIST, 2015.

STANDARDS, N. I. of; TECHNOLOGY. **Post-Quantum Cryptography**. 2022. Disponível em: https://csrc.nist.gov/projects/post-quantum-cryptography.

ZHANDRY, M. How to record quantum queries, and applications to quantum indifferentiability. In: SPRINGER. **Annual International Cryptology Conference**. [S.l.], 2019. p. 239–268.

A ARTIGO DO TCC

Estudo de Esquema de Assinatura Digital Dilithium

Gustavo C. Biage[1]

[1]Departamento de Informática e Estatística (INE)
Universidade Federal de Santa Catarina (UFSC)
CEP: 88040-370 – Florianópolis – SC – Brazil

Abstract. *In cryptography, the most common digital signature schemes base their security on the hardness of factoring large integers or computing discrete logarithmic. These schemes may be broken by powerful quantum computers with the help of Shor's algorithm. Therefore, post-quantum cryptography schemes search for new ways to obtain security against such computers. Schemes based on the complexity of certain lattices problems have maintained the interest of researchers for a few decades. Two signature schemes based on lattices have been selected to be standardized after the third round of NIST's post-quantum process. One of these schemes, Dilithium, make use of the Fiat-Shamir with aborts heuristic to obtain a signature scheme from an identification scheme. The aborting technique allows to avoid the leakage of information of the private key, keeping the scheme secure. This work study the fundaments of cryptography schemes based on the algebraic structure of lattices, presenting the computational theory and mathematic behind, and uses Dilithium as learning tool. As a result, we offer a material that allows an easy comprehension of the scheme, incentivizing the use and implementation of post-quatum cryptography algorithms.*

Keywords: *Post-quantum Cryptography. Lattice Based Scheme. Public Key Cryptography.*

Resumo. *[Resumo] Em criptografia, os esquemas de assinatura digital mais utilizados atualmente baseiam sua segurança na dificuldade de fatorar inteiros grandes ou computar logaritmo discreto. Estes esquemas podem ser quebrados por computadores quânticos poderosos com o algoritmo de Shor. Portanto, esquemas pós-quânticos procuram novas abordagens que forneçam segurança contra tais computadores. Esquemas baseados na complexidade de problemas de reticulados (lattices) são foco de pesquisas há algumas décadas. Duas propostas de esquemas de assinatura digital com esta abordagem foram selecionados para a padronização após a terceira etapa do processo de padronização do Instituto Nacional de Padrões e Tecnologia dos EUA (NIST). Um desses esquemas, o Dilithium, é baseado em problemas que envolvem estruturas algébricas de reticulados. O esquema faz uso da heurística de Fiat-Shamir with aborts, que rejeita assinaturas que levam ao vazamento de informações privadas. Este trabalho realiza um estudo dos fundamentos de esquemas baseados em reticulados, como também, apresenta a teoria e matemática computacional utilizada. Como resultado, oferece um material que permite a compreensão e implementação do esquema, incentivando o uso da criptografia pós-quântica em assinaturas digitais.*

Palavras-chave: *Criptografia pós-quântica. Esquema Baseado em Reticulado. Criptografia de Chave Pública.*

1. Introdução

Em criptografia, os esquemas de assinatura digital mais utilizados atualmente baseiam sua segurança na dificuldade de fatorar inteiros grandes ou computar logaritmo discreto. Como exemplo disso encontra-se o RSA (Rivest–Shamir–Adleman) e do ECDSA (*Elliptic Curve Digital Signature Algorithm*). Estes esquemas podem ser quebrados em tempo polinomial por computadores quânticos poderosos com o uso do algoritmo de [Shor 1999]. Assim, a partir do avanço da era pós-quântica, existe a necessidade de desenvolver e ter conhecimento da segurança de esquemas criptográficos capazes de garantir autenticidade, integridade e não-repúdio por todo futuro previsto. Historicamente, levou-se quase duas décadas para a implantação de esquemas criptográficos modernos (NIST, 2017). Portanto, o Instituto Nacional de Padrões e Tecnologia dos EUA (NIST) lançou um esforço de padronização pós-quântico, em que a terceira rodada terminou recentemente, com o objetivo de normatizar um ou mais esquemas de cifragem de chave pública (*public key encryption*), assinatura digital e acordo de chaves (*key exchange*).

Conforme apresentado por [Moody 2017] e [Moody 2019], na primeira rodada do incentivo criado pelo NIST houve a submissão de 23 esquemas de assinatura digital. Os esquemas participantes são geralmente separados em suas respectivas famílias. As famílias mencionadas são conjuntos de esquemas que utilizam princípios de segurança similares. Neste conjunto de famílias encontramos: criptografia baseada em sistemas de equações quadráticas multivariadas, criptografia baseada em reticulados, criptografia baseada em resumos criptográficos e criptografia baseada em códigos de correção de erros. Em cada rodada do esforço, os requisitos e critérios de avaliação para padronização analisam (i) a possibilidade de implementação de cada esquema em diversas plataformas; (ii) a performance do esquema; (iii) a publicação do esquemas e disponibilização gratuita do material publicado; (iv) a presença de evidência teórica e empírica, acompanhado de indicações de segurança contra ataques convencionais e pós-quânticos; por fim, (v) a descrição de parâmetros que caracterizam diversos níveis de segurança.

Na terceira rodada haviam como finalistas dois esquemas baseados em reticulados (FALCON e Dilithium) e um esquema baseado em sistemas de equações quadráticas multivariadas (Rainbow). Após a terceira rodada, foi selecionado estes dois esquemas de assinatura baseados em reticulados para serem normatizado [Gorjan et al. 2022], como esperado pelo NIST [Alagic et al. 2020]. Além destes, um esquema, não finalista, baseado em resumos criptográficos será padronizado (SPHINCS+).

Esquemas baseados em reticulados são construídos com base em conjecturas sobre a dificuldade de problemas de reticulados, como por exemplo os problemas LWE (*Learning with Errors*) e SIS (*Short Integer Solution*), ou suas variações. Estas conjecturas, de acordo com [Regev 2005], são suportadas pela ausência de uma solução quântica de tempo polinomial encontrada para os problemas. Um exemplo de esquema construído dessa forma é o Dilithium [Ducas et al. 2021], que faz uso da heurística *Fiat-Shamir with aborts* para obter um esquema de assinatura digital. Esta heurística se comporta de forma similar à heurística de construção de esquemas de assinatura Fiat-Shamir, porém, transferida para o contexto de reticulados. Neste contexto, existe a possibilidade de abortar-se

o processo de assinatura ao identificar o vazamento de informações da chave privada.

Em relação ao seu *concorrente*, o FALCON apresenta melhores resultados em geral, relacionado a tamanho de chave pública, chave privada e assinatura. Contudo, o esquema possui diversas dificuldades de implementação. Um exemplo disso está relacionado com a técnica de geração de números aleatórios utilizada. Por este motivo, na terceira rodada o Dilithium propõe a padronização de um esquema mais fácil de ser implementado e que ainda apresenta bons resultados de desempenho, tamanho de chave e assinatura; e fornece evidências de segurança ainda não questionadas. Por estes motivos o Dilithium foi escolhido como tópico de estudo deste trabalho e como a principal recomendação para o uso de algoritmo de assinatura pós-quânticos pelo NIST [Gorjan et al. 2022].

Devido ao Dilithium ser um esquema promissor e selecionado para estandardização no esforço criptográfico do NIST, o trabalho tem como objetivo específico mostrar a aplicação dos princípios modernos da criptografia pós-quântica baseada em reticulados na construção deste esquema. Ou seja, apresentar a matemática computacional por trás da geração de um par de chaves assimétricas, o processo de assinar uma mensagem com uma chave privada, o processo de verificar uma assinatura com uma chave pública. Dessa forma, por meio das primitivas criptográficas apresentadas, procura-se também esclarecer o motivo por trás da credibilidade do esquema ser seguro, isto significa, ser computacionalmente impossível descobrir a chave privada a partir da chave pública e assinaturas; e ser computacionalmente impossível construir uma assinatura válida sem a chave privada. No trabalho, a teoria do esquema de assinatura será acompanhada por uma implementação que tornará concreto os conceitos teóricos.

2. Dilithium

Dilithium é um esquema de assinatura digital da família CRYSTALS e sua segurança é baseada na dificuldade de solucionar problemas difíceis de reticulados. O esquema segue a heurística *Fiat-Shamir with Aborts*. Portanto, ao computar a assinatura de uma mensagem com a chave privada, a mesma ideia de *abort* é adotada. Uma vez que identifica-se o vazamento de informações sensíveis da chave privada, o processo de construção de assinatura é abortado e reiniciado. Assim, ataques de recuperação da chave privada se mantém computacionalmente impraticáveis de serem realizados. Outros dois ataques que um esquema de assinatura encontra-se sujeito são ataques de substituição de mensagem e construção indevida de assinaturas digitais. Todos estes três ataques se resumem a problemas difíceis de serem solucionados tanto por computadores clássicos quanto por computadores quânticos. Assim, estes problemas são problemas difíceis de reticulados ou, em certos casos, o problema de encontrar uma colisão de resumo criptográfico em uma função de resumo criptográfico resistente a colisão.

A credibilidade de que os problemas relacionados ao esquema sejam realmente difíceis de serem solucionados origina-se de conjecturas sobre o *Short Integer Solution* e *Learning With Errors* com reticulados modulares (nomeados, respectivamente, MSIS e MLWE). Os reticulados estruturados permitem que a álgebra presente no esquema Dilithium, que envolve a estrutura algébrica de anéis quocientes polinomiais, seja isomórfica aos reticulados modulares. Assim, devido a teorias de convolução de vetores e o teorema chinês do resto, obtém-se um esquema de assinatura que realiza multiplicações rápidas de polinômios (com mais baixa complexidade assintótica), apresenta bons re-

sultados de performance e com uma segurança baseada na dificuldade de problemas de reticulados. Onde boa performance é, tempo de execução e espaço de memória ocupado pelos pares de chaves e pelas assinaturas.

A Tabela 1 apresenta os parâmetros utilizados no Dilithium junto com seus respectivos valores. O significado de cada parâmetro será melhor aprofundado ao longo da explicação do Dilithium e os algoritmos que o compõem. Os parâmetros q e n correspondem ao anel quociente polinomial $R_q = \mathbb{Z}_q[x]/\langle x^n + 1\rangle$ utilizado no esquema.

Nível de Segurança do NIST:	2	4	5
Parâmetros			
n [*grau do polinômio no anel quociente*]	256	256	256
q [módulo]	8380417	8380417	8380417
d [bits ignorados de t]	13	13	13
τ [*quantidade de ± 1's no vetor c*]	39	49	60
γ_1 [intervalo para distribuição de y]	2^{17}	2^{19}	2^{19}
γ_2 [parâmetro de centralização]	$(q-1)/88$	$(q-1)/32$	$(q-1)/32$
(k, l) [dimensões de matrizes ou vetores]	$(4, 4)$	$(6, 5)$	$(8, 7)$
η [intervalo para distribuição de s_1 e s_2]	2	4	2
β [$\tau \cdot \eta$]	78	196	120
ω [*quantidade de 1's nas dicas*]	80	55	75

Table 1. Parâmetros para níveis de segurança do NIST.

Nas próximas seções será definido o esquema de assinatura Dilithium com detalhes sobre a geração de chaves, geração e construção de assinaturas. Serão apresentados as propriedades do esquema, permitindo enxergar o dimensionamento das estruturas algébricas, vazamento de informações, etc. E será clarificado quais são os problemas que, se solucionados, levam um adversário a invalidar as propriedades de autenticidade, não-repúdio e integridade asseguradas pelos esquemas criptográficos de assinatura digital. Apresentando no processo as principais evidências de segurança.

3. Definição do esquema

Segue abaixo a descrição dos algoritmos de geração de chaves, construção e verificação de assinaturas que compõe o esquema Dilithium. Considere que certas explicações fornecidas aqui, sobre estes processos, são parciais e que servirão somente como uma introdução ao conceitos do esquema que permitem construir a discussão projetada.

Geração de chaves. As principais características a serem notadas em respeito da geração de chaves são: a compressão do par de chaves, a relação matemática entre a chave privada e a chave pública; e, por último, o objetivo de cada termo presente nas chaves do par. A razão por trás da compressão de chaves é bem simples de se perceber. A matriz $A \in (R_q)^{k \times l}$ é composta por $k \times l$ polinômios com n coeficientes cada. Esta matriz é armazenada tanto na chave pública quanto na chave privada, pois ela é utilizada na geração e verificação de assinaturas. A matriz A ocupar um espaço majoritário dentro das duas chaves. Dessa forma, os métodos do esquema relacionados à expansão de vetores utiliza uma semente ρ para expandir a matriz A. Com isto, consegue-se compactar ambas as chaves ao não armazenar A, mas, ao armazenar uma única semente ρ de 32 bytes.

A relação matemática entre a chave privada e a chave pública se dá pela construção do vetor de polinômios $\boldsymbol{t} := A\boldsymbol{s_1} + \boldsymbol{s_2}$ e permite que uma assinatura seja verificada pela chave pública do assinante. Para perceber que esta relação não expõe os segredos da chave privada assuma que um adversário tome conhecimento de $\boldsymbol{t}$. Isto não é errado de se fazer, pois $\boldsymbol{t}$ é uma informação da chave pública em versões simplificadas do Dilithium e parcialmente conhecido na versão demonstrada ($\boldsymbol{t_1}$ pertence à chave pública). Assim, o segredo da chave privada representado pelo par $(\boldsymbol{s_1}, \boldsymbol{s_2})$ se mantém protegido pela dificuldade de solucionar instâncias difíceis do MLWE. De maneira breve perceba que, ao tomar o vetor $\boldsymbol{s_2}$ como os vetores de erros uniformemente aleatórios desconhecidos e ao tomar o vetor $\boldsymbol{t}$ como a solução da expressão cuja entrada é a matriz A, descreve-se o problema de reticulado MLWE. Logo, este problema consiste em encontrar a função de maior probabilidade, neste caso definida por $\boldsymbol{s_2}$, que mapeie a entrada à saída.

O último elemento de maior importância da chave a ser comentado é a sequência de bytes tr. Os bytes de tr representam o resumo criptográfico da chave pública e, a menos que seja encontrado colisões de resumo criptográfico, garantem que no momento da verificação realmente é a chave pública do assinante que está verificando a assinatura.

Algoritmo 1 Geração de chaves (Dilithium$_{n,q,d,\tau,\gamma_1,\gamma_2,k,l,\eta,\beta,\omega}$)

Saída: Um par de chaves (pk, sk).

1. $\zeta \leftarrow \{0,1\}^{256}$
2. $(\rho, \rho^{'}, K) \in \{0,1\}^{256} \times \{0,1\}^{512} \times \{0,1\}^{256} := \mathrm{H}(\zeta)$
3. $A \in (R_q)^{k \times l} :=$ ExpandA(ρ) ▷ *Assuma que a matriz A é obtida em sua forma NTT*
4. $(\boldsymbol{s_1}, \boldsymbol{s_2}) \in (S_\eta)^l \times (S_\eta)^k :=$ ExpandS(ρ') ▷ *Onde $S_\eta \subset R_q$ é o conjunto de todos os polinômios cuja a norma ℓ_∞ é menor ou igual a η*
5. $\boldsymbol{t} := A\boldsymbol{s_1} + \boldsymbol{s_2}$
6. $(\boldsymbol{t_1}, \boldsymbol{t_0}) :=$ Power2Round$_q(t, d)$
7. $tr \in \{0,1\} := \mathrm{H}(\rho \| \boldsymbol{t_1})$
8. **retorne** $(pk = (\rho, \boldsymbol{t_1}), sk = (\rho, K, tr, \boldsymbol{s_1}, \boldsymbol{s_2}, \boldsymbol{t_0}))$

Construção e verificação de assinaturas. A assinatura digital deve criar um relacionamento matemático que envolve a chave privada, a chave pública e a mensagem assinada. Com isto têm-se o objetivo de permitir que uma assinatura construída pela chave privada seja verificada unicamente pela chave pública do assinante. Não só isso, deseja-se que uma assinatura verificada pela chave pública do assinante somente seja capaz de ter sido criada pela respectiva chave privada. Assim, estabelece-se autenticidade e não-repúdio, que são duas propriedades de segurança fornecidas por esquemas de assinatura.

A terceira propriedade de segurança, integridade da mensagem, é obtida pelo forte vínculo da assinatura com a mensagem assinada. Para enxergar o relacionamento no Dilithium que obtém estes resultados, considere que M e $\boldsymbol{w_1} :=$ HighBits$_q(A\boldsymbol{y}, 2\gamma_2)$, linha 9 do Algoritmo 3, sejam a mensagem assinada e o vetor produzido no processo de construção da assinatura, respectivamente. Considere que M' e $\boldsymbol{w_1'} :=$ UseHint$_q(h, A\boldsymbol{z} - c\boldsymbol{t_1} \cdot 2^d, 2\gamma_2)$, linha 4 do Algoritmo 2, sejam a mensagem em verificação e o vetor produzido com a chave pública mais a assinatura no processo de verificação, respectivamente. Assim, se a mensagem assinada não for modificada e a respectiva chave pública do assinante for fornecida à verificação, então têm-se que o resumo da chave pública e a mensagem resultam, ao serem entregues a função de resumo criptográfico H, na

mesma saída que foi obtida dentro do processo de criação da assinatura. Isto é, $\mu' :=$ H(*resumo da chave pública* $\|$ M') é idêntico a $\mu :=$ H($tr \parallel M$) utilizado para assinar a mensagem. Portanto, a menos que seja encontrado colisões de resumo criptográficos, considerado impraticável de se realizar, quaisquer mudanças na mensagem ou na chave pública resulta em uma assinatura inválida.

Além disso, a relação matemática de ambas as chaves, definida por t, e os dados $(\boldsymbol{z}, \boldsymbol{h})$ inclusos na assinatura permitem computar $\boldsymbol{w_1}$, linha 9 da construção de assinatura, e $\boldsymbol{w_1'}$, linha 4 da verificação de assinaturas, tal que $\boldsymbol{w_1'} = \boldsymbol{w_1}$. A explicação da igualdade destas duas expressões é obtida pelo comportamento de UseHint_q e HighBits_q. A matemática envolvida permite que uma assinatura construída com a chave pública seja verificada. Uma vez que os dois vetores sejam idênticos, a comparação $\text{H}(\mu' \| \boldsymbol{w_1'}) = \text{H}(\mu \| \boldsymbol{w_1})$ identifica a assinatura como válida. Contudo, ainda deseja-se que a assinatura somente possa ter sido construída com o uso da chave privada. O esquema fornece esta segurança, pois um adversário que tente construir uma assinatura indevidamente, sem a chave privada, encontra-se buscando uma tripla $(\tilde{c}, \boldsymbol{z}, \boldsymbol{h})$ que satisfaça os requisitos do esquema e solucione instâncias difíceis do problema SelfTargetMSIS. O problema SelfTargetMSIS, cuja definição encontra-se na Seção 5, é demonstrado ser de complexidade igual ou superior ao problema MSIS de reticulado por uma redução não-rigorosa, mas, que ainda fornece evidências de segurança. Logo, o esquema encontra-se seguro contra a construção indevida de assinaturas com base em fortes conjecturas sobre os problemas de reticulados.

Algoritmo 2 Verificação de assinatura ($\text{Dilithium}_{n,q,d,\tau,\gamma_1,\gamma_2,k,l,\eta,\beta,\omega}$)

Entrada: Uma chave pública $pk := (\rho, t_1)$, uma mensagem $M \in \{0,1\}*$ e uma assinatura $\sigma := (\tilde{c}, \boldsymbol{z}, \boldsymbol{h})$

Saída: Um valor lógico que corresponde à validade da assinatura.

1. $A \in (R_q)^{k \times l} :=$ ExpandA(ρ) ▷ *Assuma que a matriz A é obtida em sua forma NTT*
2. $\mu \in \{0,1\}^{512} :=$ H(H($\rho \| t_1$)$\|M$)
3. $c :=$ SampleInBall($\tilde{c}$) ▷ $\hat{c} := \boldsymbol{NTT}(c)$
4. $\boldsymbol{w_1'} :=$ $\text{UseHint}_q(\boldsymbol{h}, A\boldsymbol{z} - c\boldsymbol{t_1} \cdot 2^d, 2\gamma_2)$ ▷ $A\boldsymbol{z} - c\boldsymbol{t_1} := \boldsymbol{NTT}^{-1}(A * \boldsymbol{NTT}(\boldsymbol{z}) - \hat{c} * \boldsymbol{NTT}(\boldsymbol{t_1}))$
5. **retorne** $\|\boldsymbol{z}\|_\infty < \gamma_1 - \beta$ e $\tilde{c} = \text{H}(\mu \| \boldsymbol{w_1'})$ e *"a quantidade de 1's em $\boldsymbol{h}$ é menor ou igual a ω"*

Geração de valores pseudo-aleatórios. A geração de valores pseudo-aleatórios (relacionados ao uso da função de resumo criptográfico H) está presente em diversas partes dos algoritmos de geração de chaves, construção de assinaturas e verificação de assinaturas. A versão do esquema Dilithium que foi descrita utiliza os algoritmos SHAKE-128 e SHAKE-256, pertencentes ao padrão de funções de resumo criptográficos FIPS 202 [of Standards and Technology 2015]. Conforme as propriedades das funções de resumo criptográfico, pode-se utilizar a saída destas funções para obter valores imprevisíveis e uniformemente pseudo-aleatórios.

Expansão de vetores. A expansão de vetores (cujos métodos relacionados são ExpandA, ExpandS e ExpandMask) são operações que amostram uma série de polinômios pseudo-aleatórios pertencentes ao anel $R_q = \mathbb{Z}_q / \langle x^n + 1 \rangle$, em coerência às propriedades de cada estrutura. Em um método de expansão arbitrário, dado uma semente $s \in \{0,1\}^*$

Algoritmo 3 Construção de assinatura (Dilithium$_{n,q,d,\tau,\gamma_1,\gamma_2,k,l,\eta,\beta,\omega}$)

Entrada: Uma chave privada $sk := (\rho, K, tr, \boldsymbol{s_1}, \boldsymbol{s_2}, \boldsymbol{t_0})$ e uma mensagem $M \in \{0,1\}^*$.
Saída: Uma assinatura $\sigma = (\tilde{c}, \boldsymbol{z}, \boldsymbol{h})$

$A \in (R_q)^{k \times l} :=$ ExpandA(ρ) ▷ *Assuma que a matriz A é obtida em sua forma NTT*
$\mu \in \{0,1\}^{512} :=$ H$(tr \| M)$
$\kappa := 0$
$(\boldsymbol{z}, \boldsymbol{h}) :=\perp$
$\rho' \in \{0,1\}^{512} :=$ H$(K \| \mu)$
enquanto $(\boldsymbol{z}, \boldsymbol{h}) =\perp$ **faça**
    $\boldsymbol{y} \in (\tilde{S}_{\gamma_1})^l :=$ ExpandMask(ρ', κ) ▷ *Onde $\tilde{S}_{\gamma_1} \subset R_q$ é o conjunto de polinômios resultados de centered modular reduction com γ_1*
    $\boldsymbol{w} := A\boldsymbol{y}$ ▷ $\boldsymbol{w} := \boldsymbol{NTT}^{-1}(\hat{A} * \boldsymbol{NTT}(y))$
    $\boldsymbol{w_1} :=$ HighBits$_q(\boldsymbol{w}, 2\gamma_2)$
    $\tilde{c} \in \{0,1\}^{256} :=$ H$(\mu \| \boldsymbol{w_1})$
    $c \in B :=$ SampleInBall$(\tilde{c})$ ▷ $\hat{c} := \boldsymbol{NTT}(c)$
    $\boldsymbol{z} := \boldsymbol{y} + c\boldsymbol{s_1}$ ▷ $c\boldsymbol{s_1} := \boldsymbol{NTT}^{-1}(\hat{c} * \boldsymbol{NTT}(\boldsymbol{s_1}))$
    $\boldsymbol{r_0} :=$ LowBits$_q(\boldsymbol{w} - c\boldsymbol{s_2}, 2\gamma_2)$ ▷ $c\boldsymbol{s_2} := \boldsymbol{NTT}^{-1}(\hat{c} * \boldsymbol{NTT}(\boldsymbol{s_2}))$
    se $\|\boldsymbol{z}\|_\infty \geq \gamma_1 - \beta$ ou $\|\boldsymbol{r_0}\|_\infty \geq \gamma 2 - \beta$ **então**
        $(\boldsymbol{z}, \boldsymbol{h}) :=\perp$
    senão
        $\boldsymbol{h} :=$ MakeHint$_q(-c\boldsymbol{t_0}, \boldsymbol{w} - c\boldsymbol{s_2} + c\boldsymbol{t_0}, 2\gamma_2)$ ▷ $c\boldsymbol{t_0} := \boldsymbol{NTT}^{-1}(\hat{c} * \boldsymbol{NTT}(\boldsymbol{t_0}))$
        se $\|c\boldsymbol{t_0}\|_\infty \geq \gamma_2$ ou *"a quantidade de 1's em h é maior que ω"* **então**
            $(\boldsymbol{z}, \boldsymbol{h}) :=\perp$
        fim se
    fim se
    $\kappa := \kappa + l$
fim enquanto
retorne $\sigma := (\tilde{c}, \boldsymbol{z}, \boldsymbol{h})$

de tamanho fixo, gera-se uniformemente uma sequência de vetores de polinômios, ou diversas sequências de vetores de polinômios, que serão utilizados pelos algoritmos do esquema. Assim, estes métodos de expansão acabam sendo responsáveis pela descompactação da chave pública e privada. As três sequências de bytes ρ, ρ' e K obtidas no início do algoritmos de geração de chaves são utilizadas para gerar todos os demais elementos da chave privada pela expansão de vetores e outros métodos computacionais.

Reduções modulares. Existem duas reduções modulares utilizadas no esquema. Uma delas é a operação de módulo convencional da matemática (representado pelos símbolos *mod*) e a outra é operação *centered modular reduction* (representado pelos símbolos *cmod*). O *centered modular reduction*, dado um operando a e um parâmetro de centralização α, centraliza o operando de forma que o resultado $r \in (-\alpha/2, \alpha/2]$.

Algoritmo 4 HighBits$_q$

Entrada: Um coeficiente $r \in \mathbb{Z}_q$ e o parâmetro de centralização $\alpha \in \mathbb{Z}$.

Saída: O valor alto da decomposição $r_1 \in \{0, 1, 2, \cdots, (q-1)/\alpha\}$.

1. $(r_1, r_0) :=$ Decompose$_q(r, \alpha)$
2. **retorne** r_1

Algoritmo 5 Power2Round$_q$

Entrada: Coeficiente r e a potência d do parâmetro de centralização.

Saída: Realiza a decomposição de r com o parâmetro de centralização 2^d.

1. $r := r \bmod q$
2. $r_0 := r \operatorname{cmod} 2^d$
3. **retorne** $((r - r_0)/2^d, r_0)$

Algoritmo 6 Decompose$_q$

Entrada: Coeficiente $r \in \mathbb{Z}_q$ de um vetor e parâmetro de centralização $\alpha \in \mathbb{Z}$.

Saída: Decompõe $r \in \mathbb{Z}_q$ em $(r_1, r_0) \in \mathbb{Z}_q \times \mathbb{Z}$ tal que $r = r_1 \cdot \alpha + r_0$

1. $r := r \bmod q$
2. $r_0 := r \operatorname{cmod} \alpha$
3. **se** $r - r_0 = q - 1$ **então**
4. $\quad r_1 := 0$
5. $\quad r_0 := r_0 - 1$
6. **senão**
7. $\quad r_1 := (r - r_0)/\alpha$
8. **fim se**
9. **retorne** (r_1, r_0)

Algoritmo 7 LowBits$_q$

Entrada: Um coeficiente $r \in \mathbb{Z}_q$ e o parâmetro de centralização $\alpha \in \mathbb{Z}$.

Saída: O valor baixo da decomposição $r_0 \in [-\alpha/2, \alpha/2)$.

1. $(r_1, r_0) :=$ Decompose$_q(r, \alpha)$
2. **retorne** r_0

Algoritmo 8 MakeHint$_q$

Entrada: Dois coeficientes $z, r \in \mathbb{Z}_q$ e o parâmetro de centralização $\alpha \in \mathbb{Z}$.

Saída: Retorna o valor lógico (0 ou 1) indicando que, ao somar z em r, modifica-se o valor alto do coeficiente.

1. $r_1 :=$ HighBits$_q(r, \alpha)$
2. $v_1 :=$ HighBits$_q(r + z, \alpha)$
3. **retorne** $r_1 \neq v_1$

Algoritmo 9 UseHint$_q$

Entrada: Uma dica $h \in \{0, 1\}$, um coeficiente $r \in \mathbb{Z}_q$ e o parâmetro de centralização $\alpha \in \mathbb{Z}$.

Saída: Recupera o valor alto r_1 obtido com auxílio da dica.

1. $m := (q - 1)/\alpha$
2. $(r_1, r_0) :=$ Decompose(r, α)
3. **se** $h = 1$ e $r_0 > 0$ **então**
4. $\quad$ **retorne** $(r_1 + 1) \bmod m$
5. **senão se** $h = 1$ e $r_0 \leq 0$ **então**
6. $\quad$ **retorne** $(r_1 - 1) \bmod m$
7. **fim se**
8. **retorne** r_1

Decomposição de valores. Dado um parâmetro de centralização $\alpha \in \mathbb{N}$, a decomposição de um coeficiente $r \in \mathbb{Z}_q$ (a decomposição de valores é implementada no algoritmo Decompose$_q$ e utilizada nos algoritmos Power2Round$_q$, HighBits$_q$ e LowBits$_q$) obtém um par de números (r_1, r_0) tais que $r = r_1 \cdot \alpha + r_0$. O fator r_1 é nomeado valor alto (*high bits*) e $r_1 \cdot \alpha \in \mathbb{Z}_q$. Portanto, temos que $r_1 \in \{0, 1, 2, \cdots, (q-1)/\alpha - 1\}$. O fator r_0 é

nomeado valor baixo e é o resultado de um *centered modular reduction* pelo parâmetro de centralização α. Ou seja, $r_0 \in (-\alpha/2, \alpha/2]$.

Norma de polinômios e vetores de polinômios. Considere as seguintes definições de norma de polinômio e norma de um vetor de polinômios.

Definição 1. Seja $n \in \mathbb{N}$ a dimensão do vetor de polinômios $\boldsymbol{w} \in (\mathbb{Z}_q[x])^n$. A norma ℓ_2 de $\boldsymbol{w}$, denotado por $\|\boldsymbol{w}\|_2$, é definida por $\sqrt{\|w_0\|_2^2 + \cdots + \|w_{n-1}\|_2^2}$, onde $w_i \in \mathbb{Z}_q[x]$ são os termos do vetor.

Definição 2. Seja $n \in \mathbb{N}$ o grau do polinômio $w \in \mathbb{Z}_q[x]$. A norma ℓ_2 de w, denotado por $\|w\|_2$, é definida por $\sqrt{\|w_0\|_\infty^2 + \cdots + \|w_n\|_\infty^2}$ onde $w_i \in \mathbb{Z}_q$ são os coeficientes do polinômio.

Definição 3. Seja $n \in \mathbb{N}$ a dimensão do vetor de polinômios $\boldsymbol{w} \in (\mathbb{Z}_q[x])^n$. A norma ℓ_∞ de $\boldsymbol{w}$, denotado por $\|\boldsymbol{w}\|_\infty$, é definida por $\max_{i=0}^n \|w_i\|_\infty$ onde $w_i \in \mathbb{Z}_q$ são os termos do vetor.

Definição 4. Seja $n \in \mathbb{N}$ o grau do polinômio $w \in \mathbb{Z}_q[x]$. A norma ℓ_∞ de w, denotado por $\|w\|_\infty$, é definida por $\max_{i=0}^n \|w_i\|_\infty$ onde $w_i \in \mathbb{Z}_q$ onde $w_i \in \mathbb{Z}_q$ são os coeficientes do polinômio.

Definição 5. Seja $w \in \mathbb{Z}_q$. No esquema Dilithium, a norma ℓ_∞ de w, denotada por $\|w\|_\infty$, é definida pela expressão $w \bmod q$. Isto é, a *centered modular reduction* de w com o parâmetro de centralização q.

Construção de desafio. O desafio $c \in B$ é um polinômio gerado no algoritmo de construção de assinatura, tal que B é o conjunto de todos os polinômios que contém τ coeficientes iguais a ± 1 e os demais coeficientes iguais a zero. O algoritmo relacionado a construção do desafio é o SampleInBall. Neste método, utiliza-se a sequência de bytes $\tilde{c}$ presente na assinatura ou obtida pelo resumo criptográfico μ concatenado com o vetor $\boldsymbol{w_1}$ empacotado. Assim, $\tilde{c}$ alimenta a geração de valores pseudo-aleatórios e, se a assinatura não foi modificada, permite que a execução do algoritmo SampleInBall obtenha o mesmo desafio c tanto no algoritmos de construção da assinatura quanto no algoritmo de verificação. O desafio é utilizado junto com as dicas no momento da verificação para construir a expressão que permite reconstruir o vetor $\boldsymbol{w_1}$.

4. Vazamento de informações

Devido ao esquema Dilithium ser baseado na heurística de *Fiat-Shamir with Abort*, mas, utilizando reticulados modulares, os mesmos problemas discutidos previamente prevalecem. Assim, a construção da assinatura $\boldsymbol{z} := \boldsymbol{y} + c\boldsymbol{s_1}$ requer que o vetor $\boldsymbol{y}$ faça papel de máscara e, assim, mantenha a assinatura desconexa do desafio c. Se o $\boldsymbol{y}$ escolhido for obtido de forma uniforme e aleatória, não pode-se assumir informações sobre o resultado de $c\boldsymbol{s_1}$ (onde $\boldsymbol{s_1}$ pertence a chave privada e c pode ser computado pela assinatura). Contudo, em esquemas similares, a obtenção de $\boldsymbol{y}$ de maneira uniforme e aleatória, conforme encontra-se teoricamente requisitado, pode não ser realizável ou ocasionam em situações não favoráveis de serem trabalhadas. Um exemplo disso ocorre no esquema de identificação de Girault. Este esquema requer que seja amostrado $y \in \mathbb{Z}$, que é um conjunto infinito e não é computacionalmente realizável. No esquema do Diltihium, a razão por trás da amostragem de $\boldsymbol{y}$ não seguir a fundamentação teórica é pelas conjecturas atuais dos problemas de reticulados. Caso o esquema de reticulado amostrasse $\boldsymbol{y}$ no intervalo que não resulta no vazamento de informações, as provas de segurança dependeriam de

conjecturas bem mais fracas do que as que são utilizadas atualmente. Assim, optou-se por construir um esquema dependente de conjecturas de reticulados mais bem estabelecidas e evitar o vazamento da chave privada de outras formas. Logo, necessita-se de abortar o processo de assinatura e isto ocorre aproximadamente quatro vezes, até que a saída produzida seja uma assinatura válida e que não vaze informações da chave privada (linhas 14 e 15 do Algoritmo 3), conforme constado no [Ducas et al. 2021]

A técnica de *abort* tem o objetivo de abortar o processo de assinatura ao identificar o vazamento de certas informações da chave privada. Por consequência, permite o esquema manter a propriedade de *witness-indistinguishable*. O processo de *abort* também pode ser chamado de *Rejection Sampling*, pois resulta na rejeição de assinaturas construídas até que seja amostrado uma assinatura válida e segura. Neste processo, ainda há o vazamento de certas informações da chave privada do assinante, mas, o vazamento é controlado. E assim, se mantém impraticável que um adversário consiga identificar qual a chave privada foi utilizada para a construção da assinatura entre todas as outras possíveis chaves (chaves possíveis $\neq$ chaves equivalentes). Existem duas condições que resultam em um *abort*. A primeira condição é $\|\boldsymbol{z}\|_\infty \geq \gamma_1 - \beta$ e a segunda condição é $\|\text{LowBits}_q(\boldsymbol{w} - c\boldsymbol{s_2}, 2\gamma_2)\|_\infty \geq \gamma_2 - \beta$. A matemática por trás destes valores permitem que todo par $(\boldsymbol{z}, c) \in (S_{\gamma_1-\beta-1})^l \times B_\tau$ em uma assinatura seja igualmente prováveis de ser obtido. A primeira condição está relacionada com o vazamento de informações, enquanto que a segunda está relacionada com o vazamento de informações e a corretude do esquema.

O conceito de *abort* introduz diversas dificuldades na definição do esquema, que é a seleção dos parâmetros que serão utilizados. Estes valores são bastante delicados. O parâmetro γ_1 deve ser escolhido estrategicamente. Quando maior for γ_1 mais difícil será que um adversário encontre a chave privada, contudo, facilita-se que o adversário forje uma assinatura. O inverso também é verdade, quando menor for γ_1 mais difícil será que uma assinatura seja forjada por um adversário e mais fácil será que a chave privada do assinante seja encontrada pelo adversário. As propriedades da segurança que envolvem o esquema são bem discutidos na próximo seção.

5. Segurança do esquema

Nesta seção serão descritos os problemas que, se solucionados, quebram o esquema de assinatura estudado e permitem a construção da assinatura válida de uma mensagem que não tenha sido assinada pelo proprietário da chave privada. Abaixo são listados os principais ataques em esquemas de assinatura digital.

Substituição de mensagem. Estes ataques consistem em utilizar uma assinatura criada pelo proprietário da chave privada e substituir por uma nova mensagem que é considerada assinada pela mesma assinatura. Assim, veja que o $\tilde{c} := \text{H}(\mu \| \boldsymbol{w_1})$ é comparado com $\text{H}(\mu' \| \boldsymbol{w_1'})$. Neste caso, $\mu := \text{H}(\text{H}(\rho \| \boldsymbol{t_1}) \| M)$ para ρ e $\boldsymbol{t_1}$ da chave pública e M é a mensagem assinada. Portanto, assumindo que a respectiva chave pública seja utilizada na verificação, modificar a mensagem assinada e manter o resultado válido de verificação equivale a encontrar uma mensagem M' tal que $\tilde{c} = \text{H}(\mu' \| \boldsymbol{w_1'})$. Onde $\mu' := \text{H}(\text{H}(\rho \| \boldsymbol{t_1}) \| M')$ e $\boldsymbol{w_1'} = \boldsymbol{w_1}$ é o vetor construído no momento de verificação. Em outras palavras, um ataque de substituição de mensagem equivale à encontrar uma colisão em uma função de resumo criptográfico resistente à colisão. Em uma última nota, é importante perceber que μ é

consequência do resumo da chave pública utilizada na verificação. Portanto, nos ataques subsequentes, assume-se que os parâmetros originados pela chave pública são fixos. Isto ocorre pois, uma vez que a chave pública é modificada, o ataque ter sucesso implica que foi encontrado uma colisão de resumo criptográfico em H.

Recuperação da chave privada. Uma segunda característica de esquemas de assinatura envolve ser seguro contra ataques de recuperação da chave privada. Neste ataque, o atacante utiliza informações da chave pública e assinaturas previamente geradas para obter a chave privada do assinante. Em certos esquemas de assinatura, ataques como este são bem sucedidos devido à forte relação matemática entre a chave privada, a assinatura e a mensagem. Contudo, a heurística *Fiat Shamir with Abort*, o qual o Dilithium se baseia, utiliza-se a técnica de *abort* para obter assinaturas desconexas da mensagem. Isto implica que ao olhar para a assinatura e a mensagem assinada, não obtém-se nenhuma informação útil sobre a chave privada, mantendo o esquema seguro. As duas condições lógicas foram discutidas na Seção 4. Com isso, obtém-se a propriedades de *witness-indistinguishable*. A propriedade informa que, dado duas chaves privadas possíveis s e s', ambas chaves são igualmente prováveis de serem a chave privada do assinante. Devido a esta propriedade, um adversário que queira descobrir a chave privada por meio de assinaturas construídas, como comentado, é incapaz de identificar com exatidão qual das possíveis chaves realmente foi utilizada para construir a assinatura.

Apesar disso, o esquema ainda requer evidência de que, sem uma mensagem, a chave pública não expõe segredo da chave privada. Veja que a relação matemática da chave pública e chave privada é dada pela expressão $\boldsymbol{t} := A\boldsymbol{s_1} + \boldsymbol{s_2}$. Assim, esta relação é, por definição, o próprio problema de reticulados MLWE. Neste caso, o par $(A, \boldsymbol{t})$ representa a entrada e a saída da função definida por $\boldsymbol{s_1}$ e $\boldsymbol{s_2}$ representa o vetor de erro. Assim, encontrar a chave privada e resolver o problema MLWE consiste em encontrar $(\boldsymbol{s_1}, \boldsymbol{s_2})$ que melhor mapeia a entrada à saída. Uma vez que existem fortes conjecturas sobre a dificuldade de solucionar o MLWE, existem também fortes conjecturas sobre a dificuldade de recuperar a chave privada neste esquema. Vale notar que o vetor $\boldsymbol{t_0}$ da chave privada não necessita ser secreto. Caso o adversário conheça o par $(\boldsymbol{t_1}, \boldsymbol{t_0})$ e compute $\boldsymbol{t}$, ele ainda precisaria resolver o problema MLWE para encontrar $\boldsymbol{s_1}$ e $\boldsymbol{s_2}$. O motivo da decomposição de $\boldsymbol{t}$ é comprimir ainda mais a chave pública do esquema.

Construção indevida de assinaturas digitais. Os últimos ataques a serem discutidos são ataques que constroem assinaturas válidas de maneira indevida. Nestes ataques, o atacante, sem ter conhecimento da chave privada, consegue montar uma assinatura considerada válida ao ser verificado pela correspondente chave pública do usuário. Um ataque bem sucedido equivale a solucionar uma instância difícil do problema *SelfTargetMSIS* definido abaixo. Além disso, pode-se mostrar que, se existir um algoritmo probabilístico que solucione o *SelfTargetMSIS* em tempo praticável, então existe um algoritmo probabilístico que soluciona os piores casos do problema MSIS em tempo praticável. Portanto, devido a credibilidade da dificuldade de MSIS, conjectura-se ser computacionalmente impraticável construir uma assinatura de maneira indevida no Dilithium. Isto é demonstrado ao reduzir o MSIS ao *SelfTargetMSIS* utilizando o *forking lemma* e somente é aprofundada na Seção seguinte.

Definição 6 (*SelfTargetMSIS*)**.** Dado uma função de resumo criptográfico H, um vetor x, o problema *SelfTargetMSIS* consiste em encontrar um par $z' := (z, c)$ tal que $c =$

$\mathrm{H}(x \| f(z'))$.

O *SelfTargetMSIS* foi definido de forma genérica. Quando se trata do esquema Dilithium, deve-se representar os termos de construção e verificação de assinaturas. Considere $\boldsymbol{v} = A\boldsymbol{y} - (A\boldsymbol{z} - c\boldsymbol{t_1} \cdot 2^d) - \mathrm{LowerBits}_q(A\boldsymbol{y}, 2\gamma_2)$. O resultado de $\boldsymbol{v}$ corrige a diferença de $\boldsymbol{w_1}$ obtido no momento da assinatura com $\boldsymbol{w_1'}$ construído na verificação, ainda, permite isolar os $\mathrm{HighBits}_q(A\boldsymbol{y}, 2\gamma_2)$ multiplicado pelo parâmetro de centralização $2\gamma_2$ pela definição de decomposição. Ou seja,

$$2\gamma_2 \cdot \mathrm{HighBits}_q(A\boldsymbol{y}, 2\gamma_2) = A\boldsymbol{z} - c\boldsymbol{t_1} \cdot 2\gamma_2 + \boldsymbol{v}. \tag{1}$$

Adiante, podemos representar a mesma relação de validade de assinatura como

$$\begin{aligned}
&A\boldsymbol{z} - c\boldsymbol{t_1} \cdot 2\gamma_2 + \boldsymbol{v} = A\boldsymbol{z} - c(\boldsymbol{t} - \boldsymbol{t_0}) + \boldsymbol{v}, && \text{, pela decomposição de } \boldsymbol{t} \\
&A\boldsymbol{z} - c(\boldsymbol{t} - \boldsymbol{t_0}) + \boldsymbol{v} = A\boldsymbol{z} - c\boldsymbol{t} + (c\boldsymbol{t_0} + \boldsymbol{v}), && \text{, por distribuição;} \\
&A\boldsymbol{z} - c\boldsymbol{t} + (c\boldsymbol{t_0} + \boldsymbol{v}) = A\boldsymbol{z} - c\boldsymbol{t} + \boldsymbol{u}, && \text{, ao considerar } \boldsymbol{u} = c\boldsymbol{t_0} + \boldsymbol{v}.
\end{aligned}$$

Da expressão acima, como apresentado em [Ducas et al. 2021], têm-se que

$$\|\boldsymbol{u}\|_\infty \leq \|c\boldsymbol{t_0}\|_\infty + \|\boldsymbol{v}\|_\infty \leq \|c\|_1 \cdot \|\boldsymbol{t_0}\|_\infty + \|\boldsymbol{v}\|_\infty \leq \tau \cdot 2^{d-1} + 2\gamma_2 + 1. \tag{2}$$

Assim, ao considerar um resumo criptográfico de uma mensagem μ, uma chave púbica $pk = (\rho, \boldsymbol{t_1})$, de onde pode-se obter $A = \mathrm{ExpandA}(\rho)$ e que a função em evidência seja $f(\boldsymbol{z}, c, \boldsymbol{u}) = A\boldsymbol{z} - c\boldsymbol{t} + \boldsymbol{u}$; o problema de forjar uma assinatura equivale a solucionar o *SelfTargetMSIS* no formato

$$\mathrm{H}\left(\mu \| \tfrac{1}{2\gamma^2} \cdot [\, A \mid -\boldsymbol{t} \mid I \,] \cdot \begin{bmatrix} \boldsymbol{z} \\ c \\ \boldsymbol{u} \end{bmatrix}\right) = c. \tag{3}$$

Como (A, t) são obtidos de maneira aleatória e uniforme, a definição de *SelfTargetMSIS* se mantém correta. Uma vez que o problema é solucionado, encontrando algum $z \in (R_q)^l$, $c \in B$ e $u \in (R_q)^k$, a assinatura será válida se e somente se

- $\|\boldsymbol{z}\|_\infty < \gamma_1 - \beta$;
- $\mathrm{H}(\mu \| \mathrm{UseHint}_q(\boldsymbol{h}, A\boldsymbol{z} - c\boldsymbol{t_1} \cdot 2^d, 2\gamma 2)) = c$;
- *A quantidade de 1's em* c *é menor que* ω;
- $\|c\|_\infty = 1$;
- $\|\boldsymbol{u}\| \leq \tau \cdot 2^{d-1} + 2\gamma_2 + 1$.

A redução pelo *forking lemma* é utilizada em esquemas que seguem a heurística de *Fiat-Shamir* a diversas décadas. Assim, têm-se conhecimento de que a redução

apresentada não é rigorosa (*non-tightness reduction*). Ainda, estes esquemas têm desconsiderado a falta de rigorosidade de reduções pelo *forking lemma*. Porém, mesmo assim mantêm-se credibilidade sobre a segurança destes esquemas devido a dificuldade de problemas análogos à Equação 3. Para obter um esquema como o Dilithium que não dependa de reduções não rigorosas, pode-se inspirar no trabalho de [Kiltz et al. 2018]. Este trabalho providencia parâmetros que permitem construir uma versão do Dilithium que seja *information-theoretically hard* e dependa somente das fortes conjecturas sobre a dificuldade do problema LWE. Porém, estes parâmetros resultam em chaves públicas 5 vezes maiores e assinaturas duas vezes maiores.

Existem outros métodos de ataques que tentam demonstrar insegurança no esquema quanto a construção de assinaturas de maneira indevida. Um exemplo disso são ataques que exploram características algébricas dos reticulados para forjar assinaturas válidas. Contudo, exemplos como este não fornecem evidências de serem aplicáveis a reticulados modulares, utilizados no esquema Dilithium.

6. Redução de MSIS para SelfTargetMSIS

Está seção exibirá a relação entre o problema MSIS e o problema SelfTargetMSIS. Devido a redução não rigorosa de um problema à outro, tem-se que a dificuldade de forjar uma assinatura, que equivale a resolver o problema SelfTargetMSIS, é de complexidade igual ou superior à complexidade do problema MSIS. Assim, o esquema encontra-se seguro contra ataques de construção indevida de assinaturas devido a fortes conjecturas sobre problemas de reticulados. A relação do *SelfTargetMSIS* com o problema MSIS se dá por uma redução que utiliza o *forking lemma*.

Definição 7 (*Forking Lemma*)**.** Considere $k \in \mathbb{Z}$. Considere um adversário e um oráculo, ambos equivalentes a uma máquina de turing determinística que executa um algoritmo probabilístico. Portanto, estas máquinas possuem duas fitas, uma fita contendo a entrada da execução e outra fita de somente leitura contendo valores aleatórios que orienta a execução probabilística do algoritmo. Dessa forma, considere τ^i_{Adv} e τ^i_{Orc} estados da fita de somente leitura presente no adversário e no oráculo, anterior a execução da requisição i, para algum $1 \leq i \leq k$. Considere que o adversário com a fita τ_{Adv} realize k requisições para o oráculo com a fita τ_{Orc} e eventualmente obtém uma propriedade p sobre uma requisição i, para algum $1 \leq i \leq k$. Se esse adversário retornar para o estado i e reconfigurar a fita de somente leitura do oráculo para uma nova fita aleatória e retomar as requisições, como foi feito anteriormente, então existe uma probabilidade não negligenciável de que este adversário obtenha a mesma propriedade sobre a requisição i novamente.

O *forking lemma* expressa que: porque o adversário não editou sua fita de somente leitura, mesmo após retornar ao estado anterior, então existe uma probabilidade não negligenciável de que ao replicar as execuções daquele ponto, mesmo que o oráculo tenha uma fita editada, a propriedade será obtida novamente sobre a requisição i. Nesta seção será demonstrado de maneira informal a redução que utiliza o *forking lemma*, permitindo enxergar a relação entre os dois problemas mencionados. Seja $k, l \in \mathbb{Z}$ dois números inteiros positivos, $A \in (R_q)^{k \times l}$ uma sequência de vetores. Considere um oráculo equivalente a uma máquina de turing probabilística que, dado $\mu \in \{0,1\}^*$ e $\boldsymbol{w} \in (R_q)^k$, computa $\mathrm{H}(\mu \| \boldsymbol{w})$ de maneira aleatória para alguma configuração. Assim, um adversário cujo objetivo é encontrar duas entradas que geram colisão de resumo criptográfico em $f_{[\ I\ |\ A\ ]} \in \mathcal{H}$ requisitará do oráculo uma sequência de resumos criptográficos

$$\mathrm{H}(\mu_1 \| \boldsymbol{w_1}), \mathrm{H}(\mu_2 \| \boldsymbol{w_2}), \ \cdots, \mathrm{H}(\mu_k \| \boldsymbol{w_k}) = c_1, c_2, \ \cdots, c_k. \tag{4}$$

Em algum momento o adversário que soluciona o problema *SelfTargetMSIS* obtém um par $(\mu_i, \boldsymbol{y})$ tal que

$$\boldsymbol{y} = \begin{bmatrix} r \\ c_i \end{bmatrix} \ \text{ e } \ c_i = \mathrm{H}(\mu_i \| \boldsymbol{w_i}) = \mathrm{H}(\mu_i \| \ [I \mid A] \cdot \boldsymbol{y})). \tag{5}$$

Devido ao *forking lemma*, o adversário volta atrás para o momento que a consulta $c_i = \mathrm{H}(\mu_i \| \boldsymbol{w_i})$ foi realizada e reconfigura o oráculo com uma nova fita, que passará a retornar resultados diferentes ao computar H para as entradas. Após o adversário retomar as requisições como anteriormente, existe uma chance não negligenciável de que o adversário construa um novo par $(\mu_i, \boldsymbol{y'})$ tal que

$$\boldsymbol{y'} \neq \boldsymbol{y}, \ \boldsymbol{y'} = \begin{bmatrix} r' \\ c'_i \end{bmatrix} \ \text{ e } \ c'_i = \mathrm{H}(\mu_i \| \boldsymbol{w_i}) = \mathrm{H}(\mu_i \parallel [I \mid A] \cdot \boldsymbol{y'}). \tag{6}$$

O fato da máquina ter sido reconfigurada implica que $c_i \neq c'_i$ e $\boldsymbol{y} \neq \boldsymbol{y'}$. Além disso, como a função H possuir a propriedade de resistência à colisão, então $c_i = \mathrm{H}(\mu_i \| \boldsymbol{w_i}) = \mathrm{H}(\mu_i \parallel [\ I \mid A] \cdot \boldsymbol{y})$ implica que $\boldsymbol{w_i} = [\ I \mid A\ ] \cdot \boldsymbol{y}$. Da mesma forma, pela propriedade de resistência a colisão, $c'_i = \mathrm{H}(\mu_i \| \boldsymbol{w_i}) = \mathrm{H}(\mu_i \parallel [\ I \mid A] \cdot \boldsymbol{y'})$ implica que $\boldsymbol{w_i} = [\ I \mid A\ ] \cdot \boldsymbol{y'}$. Como consequência, temos que $[I \mid A] \cdot (\boldsymbol{y} - \boldsymbol{y'}) = [I \mid A] \cdot \boldsymbol{y} - [I \mid A] \cdot \boldsymbol{y'} = w_i - w_i = 0$ e $\boldsymbol{y} - \boldsymbol{y'}$ é uma solução para uma instância difícil do problema MSIS cuja a matriz é $[\ I \mid A\ ]$. Como A é escolhido de forma uniforme e aleatória, isto demonstra um redução do MSIS para o *SelfTargetMSIS*.

7. Contribuição de implementação

Implementações do esquema de assinatura digital Dilithium continuam a serem aprimoradas e estudadas no âmbito da ciência em computação. Apesar de esquemas criptográficos possuírem provas abstratas de matemática, que fornecem evidências de segurança, implementações ingênuas destes esquemas ainda podem ser inseguras contra ataques de canal-lateral (*side-channel*). Nestes ataques, analisa-se propriedades arquiteturais em uma execução para obter dados sensíveis do esquema criptográfico. Nos esquemas de assinatura digital, ataques de canal-lateral observam características como tempo de execução e/ou gastos energéticos e tentam obter informações secretas da chave privada. Por este motivo, os requisitos para padronização de um esquema, no processo pós-quântico do NIST, exigem a existência de implementações seguras tanto por Software quanto por Hardware.

A implementação completa do Dilithium realizada neste trabalho não é segura contra ataques desta natureza [de Castro Biage 2022]. Para obter uma implementação que seja segura contra ataques de análise temporal, por exemplo, requer-se que a execução dos algoritmos ocorra em tempo constante (*constant-time*). Isto é, operações como a multiplicação de dois números inteiros arbitrários devem levar sempre o mesmo número de ciclos de *clock* para serem concluídas. Outras operações regulares neste esquema criptográfico são operações modulares e divisões aritméticas. Por este motivo, operações triviais são

implementadas de uma forma mais complexa do que com o uso de um símbolo singular de multiplicação, módulo ou divisão presente em linguagens de programação. Logo, o código de implementação segura deste esquema se distingue de uma implementação de alto nível ou de um pseudo-código que o descreve.

Um segundo problema que dificulta a compreensão de esquemas criptográficos é a ausência de uma fundamentação teórica detalhada. O trabalho de [Ducas et al. 2018] (que apresenta o esquema Dilithium) sofreu diversas modificações após as contribuições de profissionais da área, desde sua primeira publicação. Assim, o trabalho submetido na etapa mais recente do processo de padronização do NIST encontra-se mais atualizado com o estado da arte. Contudo, no processo de atualização, informações básicas pertencentes ao esquema foram omitidas. Um exemplo disso é a explicação de um método para obter o *centered modular reduction*, que não encontra-se na versão mais recentemente.

A contribuição fornecida pela implementação realizada do esquema, apesar de insegura, explica por uma demonstração prática e simples o funcionamento do esquema de assinatura Dilithium. Entre os principais benefícios da implementação fornecida neste trabalho, quando comparado com a implementação tornada pública pelos autores do Dilithium [Seiler 2017], encontra-se que: (i) os métodos implementados são assemelhantes ao pseudo-código apresentado e explicações fornecida em [Ducas et al. 2021]; (ii) a implementação do *Number Theoretic Transform* se assemelha à explicação do Teorema Chinês do Resto; (iii) a multiplicação de vetores, equivalente a transformação à forma NTT, multiplicação ponto-a-ponto e respectiva transformação à forma inversa são explicitamente indicados; (iv) o comprimento de vetores, o grau de polinômios e o divisor dos módulos são explicitamente referenciados nas variáveis; (v) percebe-se com clareza a estrutura algébrica utilizada. Portanto, acredita-se que esta implementação se comporta melhor aos objetivos do trabalhos, de fornecer de uma forma a melhorar a compreensão um estudo sobre esquemas de assinatura digital baseados em reticulados.

Para evidenciar o funcionamento do código implementado, os testes realizados compilam também a implementação do Dilithium dos autores do esquema. Assim, entrega a ambas as implementações a mesma semente, gera milhares de chaves, constrói e verifica milhares de assinaturas; obtendo resultados idênticos nos processos. O objetivo é demonstrar que o comportamento inteiro do esquema é o mesmo, apesar da implementação ser de alto nível e utilizar certos componentes da orientação à objetos do C++. O motivo dos algoritmos pertencentes ao esquema serem executados muitas vezes é, principalmente, para garantir que o comportamento de *abort* esteja correto na implementação fornecida. Percebeu-se empiricamente que necessita-se de assinar centenas de assinaturas para que certas condições de *abort* ocorram. Assim, acredita-se que a construção de milhares de assinaturas se torna o suficiente para evidenciar o funcionamento do código implementado.

8. Conclusão

Problemas de reticulados têm sido estudados à diversas décadas. Três dos mais importantes problemas de reticulados são o *Shortest Vector Problem* (SVP), o *Approximate Shortest Vector Problem* (SVP_γ) e o *Closest Vector Problem* (CVP). A dificuldade de solucionar o CVP é bem definida para qualquer norma ℓ_p [Micciancio and Goldwasser 2002]. Enquanto isso, os problemas envolvendo o SVP são estudados somente em cenários es-

pecíficos. Dessa forma, existem provas da dificuldade do SVP dentro da norma ℓ_∞ [Dinur 2002] e conjectura-se que o problema se mantém complexo nas demais normas [van Emde-Boas 1981] [Micciancio 2001]. Com isso, o trabalho [Ajtai 1996] demonstrou a primeira primitiva criptográfica baseada em reticulados com evidência de segurança. Desde então, reticulados se tornaram atrativos para a construção primitivas criptográficas. Trabalhos posteriores, como [Goldreich et al. 2011] e [Regev 2005], relacionaram os problemas *Short Integer Solution* (SIS) e *Learning with Errors* (LWE) aos três problemas fundamentais de reticulados e às primitivas criptográficas baseadas em reticulados existentes. Como consequência, os problemas SIS e LWE se transformaram em ótimas ferramentas para construir e demonstrar segurança de esquemas criptográficos.

Um problema ainda pertinente em esquemas com segurança baseada no SIS e LWE é o desempenho. Para contornar isso, trabalhos como [Lyubashevsky et al. 2010], [Lyubashevsky and Micciancio 2006] e [Peikert and Rosen 2006] demonstram evidência da dificuldade de solucionar o RSIS e RLWE, envolvendo reticulados ideais. Como reticulados modulares são uma generalização de reticulados ideais, acredita-se que os problemas MSIS e MLWE, envolvendo reticulados modulares, sejam tão ou mais complexos do que o RSIS e RLWE. Os reticulados ideais e modulares são considerados reticulados estruturados. Devido a suas estruturas, a presença de um isomorfismo contendo o anel $R = \mathbb{Z}_q[x]/\langle x^n + 1\rangle$ permite que técnicas como a *Number Theoretic Transform* (NTT) realizem de maneira eficiente diversas multiplicações sobres seus elementos. Assim, consegue-se obter esquemas com bom desempenho sem sacrificar a segurança baseada em problemas difíceis de reticulados

Neste contexto, este trabalho atingiu o objetivo de introduzir, explicar e aprofundar a fundamentação teórica desta área, ausente em diversos cursos de ciência da computação. Foi selecionado um esquema de assinatura promissor e baseado em reticulados, que foi selecionado para padronização no processo pós-quântico do NIST, para servir de ferramenta educacional. Portanto, o esquema selecionado foi o Dilithium. O *Fiat-Shamir with Aborts* [Lyubashevsky 2009] [Fiat and Shamir 1987], o qual o Dilithium foi inspirado, é uma heurística que permite construir um esquema de assinatura em cima de um esquema de identificação, dentro do contexto de reticulados. Esta heurística é desenvolvida com um padrão observado em diversos esquemas de identificação. Por este motivo, existe uma sequência de reduções que relacionam o esquema de assinatura resultante aos problemas de reticulados de maneira organizada. Logo, acredita-se que o Dilithium seja a melhor ferramenta para introduzir a criptografia baseada em reticulados a novos estudantes da área, entre os esquemas promissores participantes do processo de padronização do NIST.

Quando se trata de compreender um esquema, diversos obstáculos são encontrados. A maioria das implementações de esquemas criptográficos têm o objetivo de serem seguras contra ataques de canal lateral (*side-channel*). Por este motivo, operações triviais tornam-se bem mais complexas e a programação utilizada deixa de ser alto nível. Só que algumas técnicas utilizadas no esquema, como exemplo a NTT, são melhor absorvidas por observação de código. Assim, este trabalho fornece uma implementação do esquema que permita relacionar os trechos de código com a fundamentação teórica apresentada. Além do conteúdo apresentado, o código se torna um outro ponto de partida para a compreensão dos algoritmos que compõem o esquema de assinatura. Para que a implementação fornecida por este trabalho esteja de acordo com os objetivos previamente estabelecidos,

o código utiliza aspectos como a orientação a objetos para ser familiar a novos leitores. E para demonstrar corretude da implementação, utiliza-se uma comparação direta com os resultados da implementação fornecida pelos autores do esquema.

Do esquema Dilithium, pode-se obter uma versão simplificada bastante similar ao *Fiat-Shamir with Aborts*. Porém, a versão apresentada neste trabalho é a completa do esquema. Embora mais complexa, esta versão segue os mesmos princípios da anterior. A diferença está no uso de técnicas avançadas para compressão da chave privada e chave pública; e técnicas de convolução para aprimorar o desempenho da multiplicação de polinômios. Por isso, observa-se menos similaridades com os algoritmos do *Fiat-Shamir with Aborts* apresentados. Porém, o uso destas técnicas mais avançadas permite que o esquema de assinatura possua ótimos resultados de desempenho; isto é, tamanhos reduzidos de assinaturas, chaves e menor tempo de execução. Ainda, no esquema, a segurança contra ataques de construção indevida de assinaturas é baseada na dificuldade do problema MSIS. Enquanto isso, a segurança contra a recuperação da chave privada pela chave pública é computacionalmente impraticável pela dificuldade de solucionar o MLWE. A versão do MLWE utilizada no Dilithium também contém fortes evidências de segurança, mas, os erros são obtidos de uma distribuição uniforme. Portanto, o esquema pode ser implementado com mais facilidade pela simplicidade desta distribuição. Ao contrário do Dilithium, o esquema de assinatura Falcon, que possui segurança baseada no MLWE, depende de diversas distribuições gaussianas, operações com árvores e ponto flutuante. Assim, pela maior simplicidade do esquema, apesar de conter resultados um pouco piores que o Falcon, foi julgado ser um esquema que melhor se encaixa nos objetivos do trabalho.

Isto não foi algo observado somente na seleção do tópico desta monografia. Apesar de ser mais didático, a partir de julho de 2022, o Dilithium encontra-se como a recomendação do NIST para a utilização de esquemas de assinaturas pós-quânticos e será um dos poucos esquemas padronizados [Gorjan et al. 2022]. Portanto, tem-se esperança que este trabalho introduza a criptografia baseada em reticulados e incentive novas pesquisas envolvendo principalmente o Dilithium.

References

Ajtai, M. (1996). Generating hard instances of lattice problems (extended abstract). In *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, STOC '96, pages 99–108, New York, NY, USA. ACM.

Alagic, G., Alperin-Sheriff, J., Apon, D., Cooper, D., Dang, Q., Kelsey, J., Liu, Y.-K., Miller, C., Moody, D., Peralta, R., Perlner, R., Robinson, A., and Smith-Tone, D. (2020). Status report on the second round of the nist post-quantum cryptography standardization process. NIST.

de Castro Biage, G. (2022). Dilithium implementation. https://github.com/gustavobiage/dilithium-implementation.

Dinur, I. (2002). Approximating svp∞ to within almost-polynomial factors is np-hard. *Theoretical Computer Science*, 285(1):55–71. Algorithms and Complexity.

Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., and Stehlé, D. (2018). Crystals-dilithium: A lattice-based digital signature scheme. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 238–268.

Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., and Stehlé, D. (2021). Crystals-dilithium: A lattice-based digital signature scheme. *IACR Transactions on Cryptographic Hardware and Embedded Systems*.

Fiat, A. and Shamir, A. (1987). How to prove yourself: Practical solutions to identification and signature problems. In Odlyzko, A. M., editor, *Advances in Cryptology — CRYPTO' 86*, pages 186–194, Berlin, Heidelberg. Springer Berlin Heidelberg.

Goldreich, O., Goldwasser, S., and Halevi, S. (2011). *Collision-Free Hashing from Lattice Problems*, pages 30–39. Springer Berlin Heidelberg, Berlin, Heidelberg.

Gorjan, A., Apon, D., David, C., Dang, Q., Thinh, D., Kelsey, J., Jacob, L., Miller, C., Dustin, M., Peralta, R., Ray, P., Robinson, A., Daniel, S.-T., and Liu, Y.-K. (2022). Status report on the third round of the nist post-quantum cryptography standardization process. NIST.

Kiltz, E., Lyubashevsky, V., and Schaffner, C. (2018). A concrete treatment of fiat-shamir signatures in the quantum random-oracle model. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 552–586. Springer.

Lyubashevsky, V. (2009). Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In Matsui, M., editor, *Advances in Cryptology – ASIACRYPT 2009*, pages 598–616, Berlin, Heidelberg. Springer Berlin Heidelberg.

Lyubashevsky, V. and Micciancio, D. (2006). Generalized compact knapsacks are collision resistant. In Bugliesi, M., Preneel, B., Sassone, V., and Wegener, I., editors, *Automata, Languages and Programming*, pages 144–155, Berlin, Heidelberg. Springer Berlin Heidelberg.

Lyubashevsky, V., Peikert, C., and Regev, O. (2010). On ideal lattices and learning with errors over rings. In Gilbert, H., editor, *Advances in Cryptology – EUROCRYPT 2010*, pages 1–23, Berlin, Heidelberg. Springer Berlin Heidelberg.

Micciancio, D. (2001). The shortest vector problem is NP-hard to approximate to within some constant. *SIAM Journal on Computing*, 30(6):2008–2035. Preliminary version in FOCS 1998.

Micciancio, D. and Goldwasser, S. (2002). *Complexity of lattice problems: a cryptographic perspective*, volume 671. Springer Science & Business Media.

Moody, D. (2017). The ship has sailed: The nist post-quantum cryptography "competition" (presentation).

Moody, D. (2019). Round 2 of the nist pqc "competition" - what was nist thinking? (presentation).

of Standards, N. I. and Technology (2015). Sha-3 standard: Permutation-based hash and extendable-output functions. NIST.

Peikert, C. and Rosen, A. (2006). Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In Halevi, S. and Rabin, T., editors, *Theory of Cryptography*, pages 145–166, Berlin, Heidelberg. Springer Berlin Heidelberg.

Regev, O. (2005). On lattices, learning with errors, random linear codes, and cryptography. volume 56, pages 84–93.

Seiler, G. (2017). Dilithium. https://github.com/pq-crystals/dilithium.

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

van Emde-Boas, P. (1981). *Another NP-complete partition problem and the complexity of computing short vectors in a lattice*. Report. Department of Mathematics. University of Amsterdam. Department, Univ.