UMA IMPLEMENTAÇÃO EM SOFTWARE DO ALGORITMO FALCON NA PLATAFORMA ARM

Caio Teixeira
Prof. Dr. Julio López César Hernández

16 de Junho de 2023

Esquema de Assinatura Digital

Um **esquema de assinatura digital** é um sistema criptográfico que provê a *autenticidade* e *integridade* de mensagens e documentos.

Estrutura geral:

- **Geração de chaves:** Gera um par de chaves interligadas, uma *privada*, e outra *pública*.
- **Geração de Assinatura:** Dado um documento, usa a chave privada para gerar uma *assinatura*.
- **Verificação de Assinatura:** Dado um documento, sua assinatura e a chave pública do assinante, verifica que a assinatura corresponde ao documento e a chave privada do assinante.

Esquema de Assinatura Digital

Um **esquema de assinatura digital** é um sistema criptográfico que provê a *autenticidade* e *integridade* de mensagens e documentos.

Estrutura geral:

- **Geração de chaves:** Gera um par de chaves interligadas, uma *privada*, e outra *pública*.
- **Geração de Assinatura:** Dado um documento, usa a chave privada para gerar uma *assinatura*.
- **Verificação de Assinatura:** Dado um documento, sua assinatura e a chave pública do assinante, verifica que a assinatura corresponde ao documento e a chave privada do assinante.

Esquema de Assinatura Digital

A segurança (garantia de autenticidade e integridade) do esquema é provida pela seguinte relação:

Recuperar a chave privada ou forjar uma assinatura $\Longrightarrow$ Resolver um problema computacionalmente difícil

Problemas Clássicos e Esquemas Padronizados

- **RSA:** Fatoração de Grandes Inteiros
- **DSA:** Logaritmo Discreto
- **ECDSA:** Logaritmo Discreto sobre Curvas Elípticas

Esquema de Assinatura Digital

A segurança (garantia de autenticidade e integridade) do esquema é provida pela seguinte relação:

$$\text{Recuperar a chave privada ou forjar uma assinatura} \implies \text{Resolver um problema computacionalmente difícil}$$

Problemas Clássicos e Esquemas Padronizados

- **RSA:** Fatoração de Grandes Inteiros
- **DSA:** Logaritmo Discreto
- **ECDSA:** Logaritmo Discreto sobre Curvas Elípticas

Criptografia Pós-Quântica

(Shor, 1997):[1] Propõe um algoritmo quântico que resolve tanto a fatoração de grandes inteiros quanto o problema do logaritmo discreto em *tempo polinomial*

[1] Peter W. Shor. *Polynomial-time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*. SIAM Journal on Computing, 26:1484–1509, 10 1997.

Criptografia Pós-Quântica

- Esquemas baseados em problemas para os quais não se espera que um computador quântico consiga resolver em tempo polinomial.
- **Classes de problemas:** Reticulados, códigos corretores de erro, funções de *hash*, isogenias entre curvas elípticas, e outros.

Criptografia Pós-Quântica

- Esquemas baseados em problemas para os quais não se espera que um computador quântico consiga resolver em tempo polinomial.
- **Classes de problemas:** Reticulados, códigos corretores de erro, funções de *hash*, isogenias entre curvas elípticas, e outros.

FALCON - Visão Geral

FALCON é um esquema de assinatura digital pós-quântico, recentemente padronizado pelo órgão de padronização norte-americano NIST[2]. Sua segurança é baseada em problemas difíceis em reticulados, e ele é projetado para **minimizar o custo de comunicação** – isto é, a *soma dos tamanho da chave pública e das assinaturas*. Para alcançar essa propriedade, FALCON instancia o Framework GPV sobre reticulados NTRU.

[2] *National Institute of Standards and Technology.*

Reticulados

Subgrupo aditivo discreto de $\mathbb{R}^m$. Seja $\mathcal{B}$ um conjunto de vetores linearmente independentes

$$\mathcal{B} = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}, \mathbf{b}_i \in \mathbb{R}^m,$$

o *reticulado* Λ é definido como o conjunto de todas combinações lineares inteiras de vetores em $\mathcal{B}$. Também podemos denotar $\mathcal{B}$ (a *base*) como uma matriz $\mathbf{B}^{m \times n}$ com os vetores de $\mathcal{B}$ nas colunas. Descrevemos Λ em termos de $\mathbf{B}$ como o conjunto

$$\Lambda = \left\{ \mathbf{B} \cdot \mathbf{z}^t \ : \ \mathbf{z} \in \mathbb{Z}^n \right\}.$$

Fundamentos

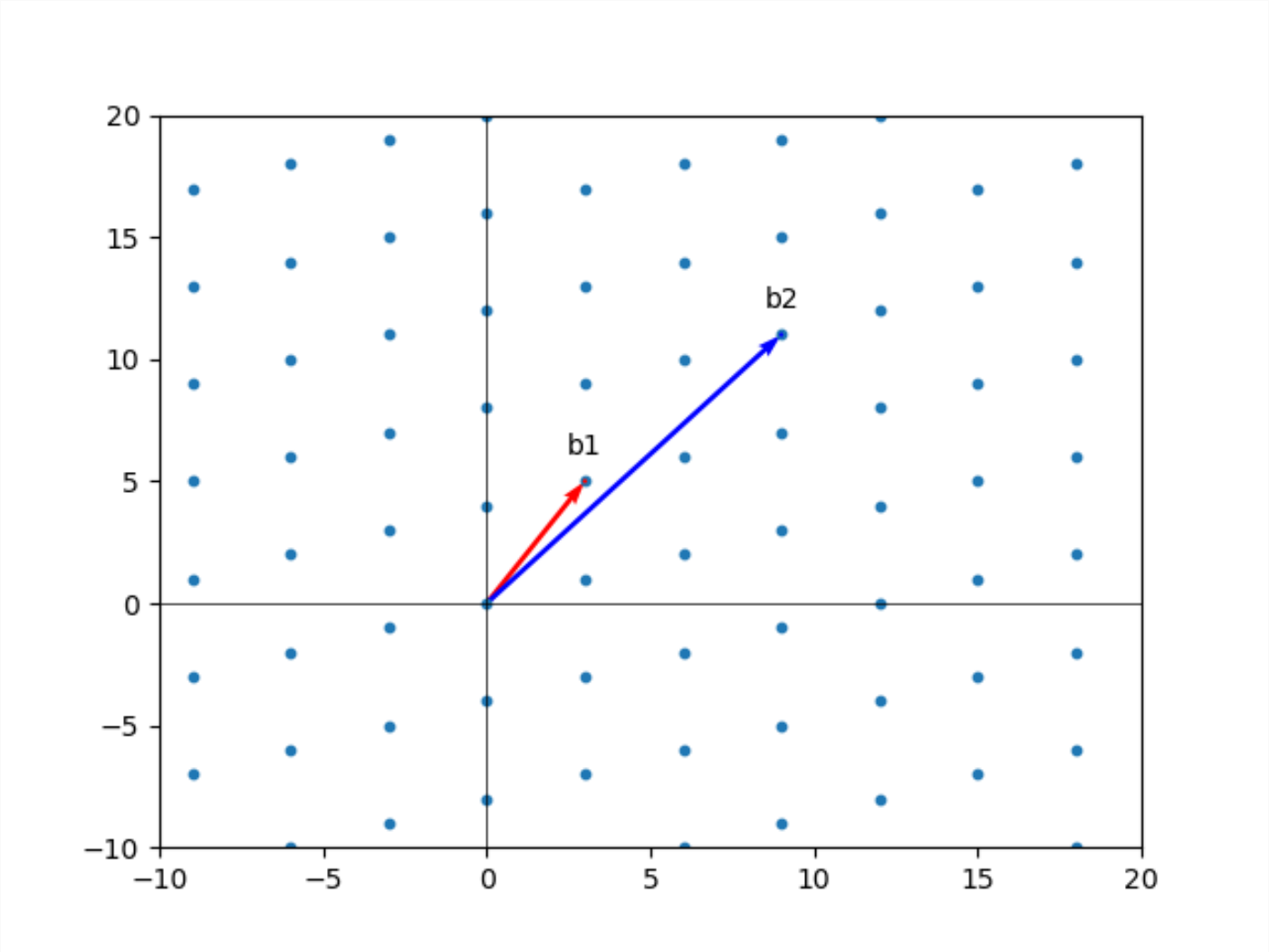


Figura: Examplo de reticulado com base $\beta = \{(3, 5), (9, 11)\}$

Problemas difíceis em reticulados[3]

- **Shortest Vector Problem (SVP):** Dado um reticulado Λ, encontre o vetor não-nulo $\mathbf{v} \in \Lambda$ tal que $\|\mathbf{v}\|_2 = \min_{\mathbf{x} \in \Lambda} \|\mathbf{x}\|_2$.
- **Closest Vector Problem (CVP):** Dado um reticulado Λ e um ponto $\mathbf{c} \in \mathbb{R}^n$, encontre o ponto no reticulado mais próximo de $\mathbf{c}$; ou seja, encontre $\mathbf{v} \in \Lambda$ que minimize $\|\mathbf{c} - \mathbf{v}\|_2$.

[3] M. Ajtai. Generating hard instances of lattice problems (extended abstract). Proceedings of the twenty-eighth annual ACM symposium on Theory of computing - STOC'96, pages 99–108, 1996

Problemas difíceis em reticulados[3]

- **Shortest Vector Problem (SVP):** Dado um reticulado Λ, encontre o vetor não-nulo $\mathbf{v} \in \Lambda$ tal que $\|\mathbf{v}\|_2 = \min_{\mathbf{x} \in \Lambda} \|\mathbf{x}\|_2$.
- **Closest Vector Problem (CVP):** Dado um reticulado Λ e um ponto $\mathbf{c} \in \mathbb{R}^n$, encontre o ponto no reticulado mais próximo de $\mathbf{c}$; ou seja, encontre $\mathbf{v} \in \Lambda$ que minimize $\|\mathbf{c} - \mathbf{v}\|_2$.

[3] M. Ajtai. Generating hard instances of lattice problems (extended abstract). Proceedings of the twenty-eighth annual ACM symposium on Theory of computing - STOC'96, pages 99–108, 1996

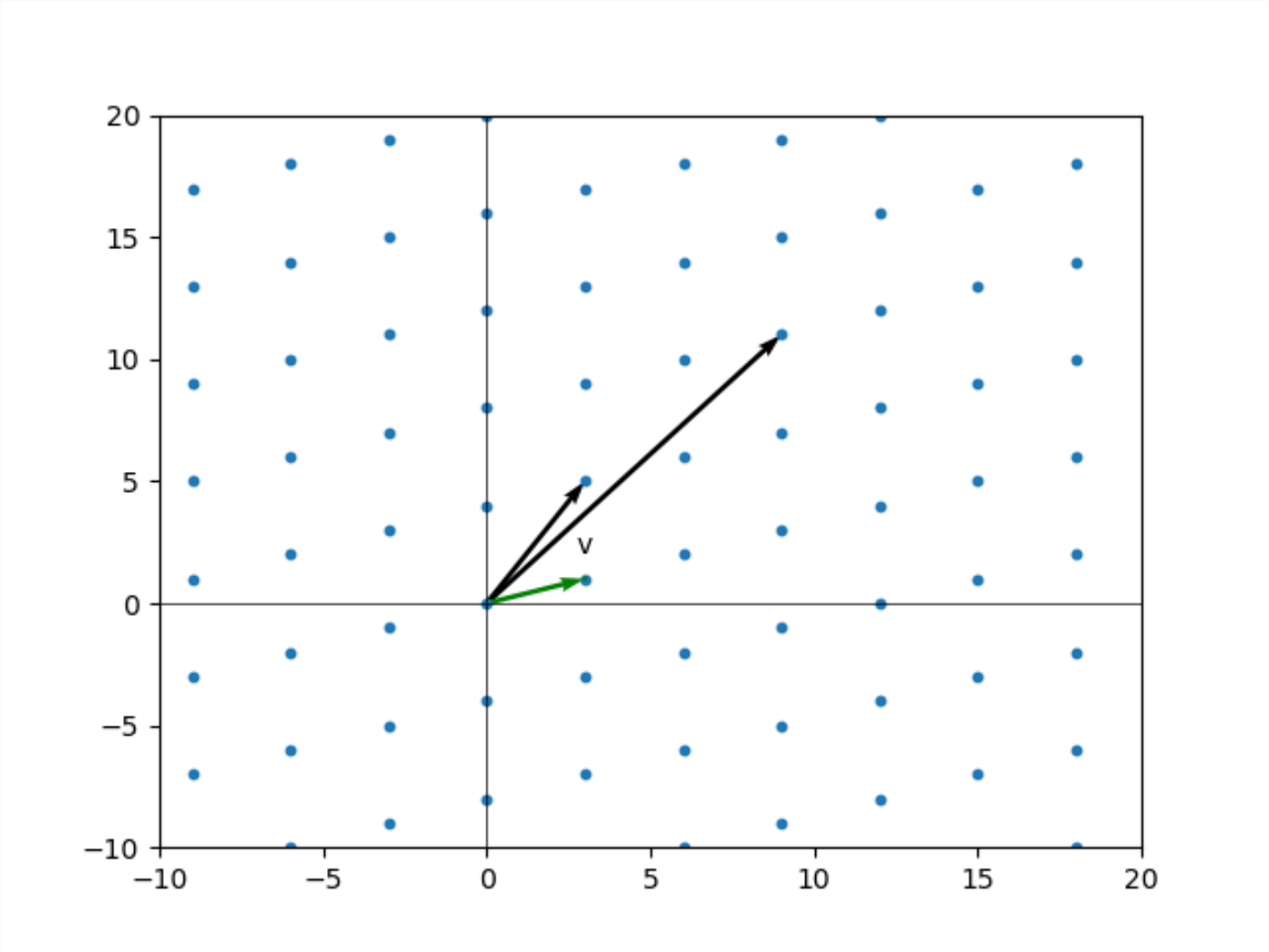


Figura: Exemplo de uma instância do SVP

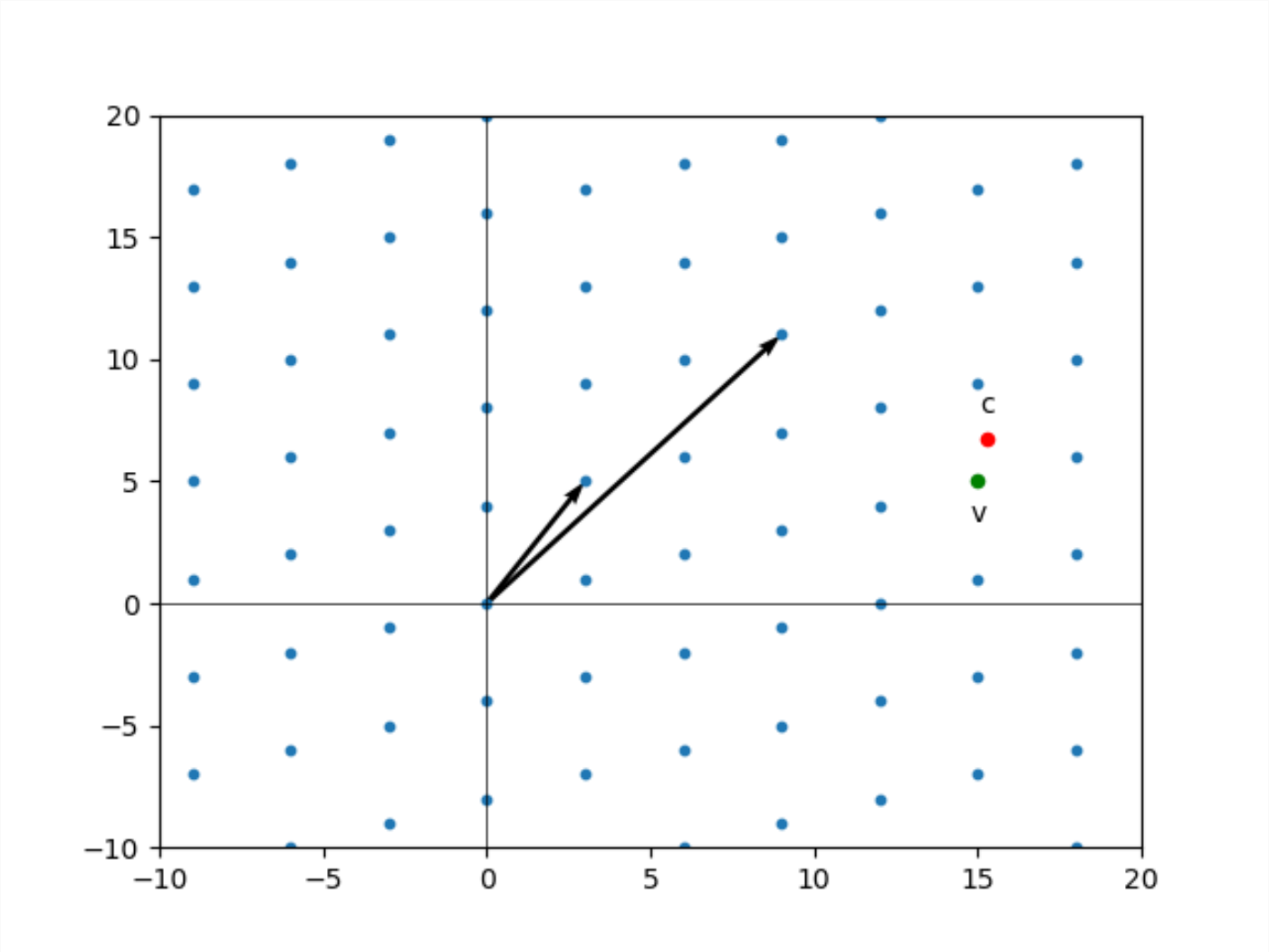


Figura: Exemplo de uma instância do CVP

Fundamentos

No entanto, o SVP e o CVP são difíceis apenas no *pior caso*.

A criptografia baseada em reticulados moderna usa os problemas *Learning With Errors* (LWE) e *Short Integer Solution* (SIS), que são difíceis no **caso médio**, e

$$\text{SVP} \succ \text{SIS}^{4} \quad \text{and} \quad \text{CVP} \succ \text{LWE}^{5}.$$

Por brevidade, apresentamos apenas o problema SIS, utilizado pelo FALCON.

---

[4] M. Ajtai. *Generating hard instances of lattice problems (extended abstract).* Proceedings of the twenty-eighth annual ACM symposium on Theory of computing - STOC'96, pages 99–108, 1996.

[5] Oded Regev. 2005. *On lattices, learning with errors, random linear codes, and cryptography.* In Proceedings of the thirty-seventh annual ACM symposium on Theory of computing (STOC '05). Association for Computing Machinery, New York, NY, USA, 84–93.

Short Integer Solution (SIS)

Dados n vetores uniformemente aleatórios $\mathbf{a}_i \in \mathbb{Z}_q^m$, dispostos nas colunas de uma matriz $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, e um limitante superior de norma β, encontre um vetor inteiro $\mathbf{z} \in \mathbb{Z}^n$ tal que

$$\|\mathbf{z}\|_2 \leq \beta, \text{ e}$$

$$\mathbf{A} \cdot \mathbf{z}^t = \sum_i \mathbf{a}_i \cdot z_i = \mathbf{0} \in \mathbb{Z}_q^m.$$

Framework GPV

O **Framework GPV**[6] é um framework teórico para projetar esquemas de assinatura digital cuja segurança é baseada no problema SIS.

[6] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. *Trapdoors for Hard Lattices and New Cryptographic Constructions*. In Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing, STOC '08, page 197–206, New York, NY, USA, 2008. Association for Computing Machinery.

Framework GPV

Geração de chaves

A chave pública é uma matriz $\mathbf{A} \in \mathbb{Z}_q^{n \times n}$, que gera um reticulado q-ário Λ_q. A chave privada é a matriz $\mathbf{B} \in \mathbb{Z}_q^{n \times n}$ que gera o reticulado *dual* (ou *ortogonal*) a Λ_q, denotado por $\Lambda_q^{\perp}$.
Ou seja, para qualquer $\mathbf{x} \in \Lambda_q$ e $\mathbf{y} \in \Lambda_q^{\perp}$,

$$\langle \mathbf{x}, \mathbf{y} \rangle = 0 \bmod q.$$

Framework GPV

Geração de assinatura

Dada uma mensagem m, calculamos seu resumo (*hash*) $H(m)$ e, então, calculamos deterministicamente $\mathbf{c}_0 \in \mathbb{Z}_q^m$ que satisfaça $\mathbf{c}_0\mathbf{A}^t = H(m)$. Então, amostramos aleatoriamente $\mathbf{v} \in \Lambda_q^\perp$ *próximo* a $\mathbf{c}_0$ utilizando a chave privada $\mathbf{B}$. A assinatura $\mathbf{s}$ é, então,

$$\mathbf{s} = \mathbf{c}_0 - \mathbf{v},$$

que é um vetor *pequeno* (norma abaixo de um limitante superior β).

Verificação de assinatura

Dada uma mensagem m, sua assinatura $\mathbf{s}$ e a chave pública $\mathbf{A}$, simplesmente verificamos que $\mathbf{s}$ é curto o suficiente (ou seja, $\|\mathbf{s}\| \leq \beta$) e que $\mathbf{s}\mathbf{A}^t = H(m)$, que deve ser satisfeito pois

$$\mathbf{s}\mathbf{A}^t = (\mathbf{c}_0 - \mathbf{v})\mathbf{A}^t = \mathbf{c}_0\mathbf{A}^t - \mathbf{v}\mathbf{A}^t = H(m) - \mathbf{0} = H(m)$$

Framework GPV

Geração de assinatura

Dada uma mensagem m, calculamos seu resumo (*hash*) $H(m)$ e, então, calculamos deterministicamente $\mathbf{c}_0 \in \mathbb{Z}_q^m$ que satisfaça $\mathbf{c}_0\mathbf{A}^t = H(m)$. Então, amostramos aleatoriamente $\mathbf{v} \in \Lambda_q^\perp$ *próximo* a $\mathbf{c}_0$ utilizando a chave privada $\mathbf{B}$. A assinatura $\mathbf{s}$ é, então,

$$\mathbf{s} = \mathbf{c}_0 - \mathbf{v},$$

que é um vetor *pequeno* (norma abaixo de um limitante superior β).

Verificação de assinatura

Dada uma mensagem m, sua assinatura $\mathbf{s}$ e a chave pública $\mathbf{A}$, simplesmente verificamos que $\mathbf{s}$ é curto o suficiente (ou seja, $\|\mathbf{s}\| \leq \beta$) e que $\mathbf{s}\mathbf{A}^t = H(m)$, que deve ser satisfeito pois

$$\mathbf{s}\mathbf{A}^t = (\mathbf{c}_0 - \mathbf{v})\mathbf{A}^t = \mathbf{c}_0\mathbf{A}^t - \mathbf{v}\mathbf{A}^t = H(m) - \mathbf{0} = H(m)$$

Matriz anti-circulante

Seja $f \in \mathbb{Z}[x]/(x^n + 1)$, com coeficientes f_i tal que

$$f = f_0 + f_1 x + \cdots + f_{n-1} x^{n-1}.$$

A *matriz anti-circulante* de f, denotada por $\mathcal{A}(f)$, é definida como

$$\mathcal{A}(f) = \begin{bmatrix} f_0 & f_1 & f_2 & \cdots & f_{n-1} \\ -f_{n-1} & f_0 & f_1 & \cdots & f_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -f_1 & -f_2 & -f_3 & \cdots & f_0 \end{bmatrix} = \begin{bmatrix} \mathcal{C}(f) \\ \mathcal{C}(x \cdot f) \\ \vdots \\ \mathcal{C}(x^{n-1} \cdot f) \end{bmatrix},$$

onde a multiplicação é feita módulo $x^n + 1$ e $\mathcal{C}(f)$ denota o vetor de coeficientes de f.

Reticulados NTRU

FALCON minimiza o custo de comunicação instanciando o framework GPV sobre **reticulados NTRU**.

Reticulados NTRU

Dados dois polinômios $f, g \in \mathbb{Z}[x]/(x^n + 1)$, podemos gerar **duas** bases para o **mesmo reticulado** computando:

$$h = g \cdot f^{-1} \bmod q, \qquad f \cdot G - g \cdot F = q \bmod (x^n + 1),$$

que define a base

$$\mathbf{A}_{h,q}^{2n \times 2n} = \begin{bmatrix} -\mathcal{A}(h) & \mathbf{I}_n \\ q \cdot \mathbf{I}_n & \mathbf{0}_{n \times n} \end{bmatrix},$$

que define a base

$$\mathbf{B}_{f,g}^{2n \times 2n} = \begin{bmatrix} \mathcal{A}(g) & -\mathcal{A}(f) \\ \mathcal{A}(G) & -\mathcal{A}(F) \end{bmatrix}.$$

Computar o ortogonal de $\mathbf{A}_{h,q}$ dado apenas h é simples (e define a chave pública), e ambas bases $2n \times 2n$ podem ser representadas simplesmente por seus polinômios.

Reticulados NTRU

FALCON minimiza o custo de comunicação instanciando o framework GPV sobre **reticulados NTRU**.

Reticulados NTRU

Dados dois polinômios $f, g \in \mathbb{Z}[x]/(x^n + 1)$, podemos gerar **duas** bases para o **mesmo reticulado** computando:

$$h = g \cdot f^{-1} \bmod q, \qquad f \cdot G - g \cdot F = q \bmod (x^n + 1),$$

que define a base

$$\mathbf{A}_{h,q}^{2n \times 2n} = \begin{bmatrix} -\mathcal{A}(h) & \mathbf{I}_n \\ q \cdot \mathbf{I}_n & \mathbf{0}_{n \times n} \end{bmatrix},$$

que define a base

$$\mathbf{B}_{f,g}^{2n \times 2n} = \begin{bmatrix} \mathcal{A}(g) & -\mathcal{A}(f) \\ \mathcal{A}(G) & -\mathcal{A}(F) \end{bmatrix}.$$

Computar o ortogonal de $\mathbf{A}_{h,q}$ dado apenas h é simples (e define a chave pública), e ambas bases $2n \times 2n$ podem ser representadas simplesmente por seus polinômios.

Reticulados NTRU

FALCON minimiza o custo de comunicação instanciando o framework GPV sobre **reticulados NTRU**.

Reticulados NTRU

Dados dois polinômios $f, g \in \mathbb{Z}[x]/(x^n + 1)$, podemos gerar **duas** bases para o **mesmo reticulado** computando:

$$h = g \cdot f^{-1} \bmod q, \qquad f \cdot G - g \cdot F = q \bmod (x^n + 1),$$

que define a base

$$\mathbf{A}_{h,q}^{2n \times 2n} = \begin{bmatrix} -\mathcal{A}(h) & \mathbf{I}_n \\ q \cdot \mathbf{I}_n & \mathbf{0}_{n \times n} \end{bmatrix},$$

que define a base

$$\mathbf{B}_{f,g}^{2n \times 2n} = \begin{bmatrix} \mathcal{A}(g) & -\mathcal{A}(f) \\ \mathcal{A}(G) & -\mathcal{A}(F) \end{bmatrix}.$$

Computar o ortogonal de $\mathbf{A}_{h,q}$ dado apenas h é simples (e define a chave pública), e ambas bases $2n \times 2n$ podem ser representadas simplesmente por seus polinômios.

Otimizações

Plataforma: ARMv8-A

Foco: Assinatura e verificação

Algoritmos principais:

- Aritmética polinomial
- Amostragem de inteiros
- Função de resumo (*hash*)

Técnicas utilizadas:

- Vetorização de operações
- Uso de instruções especializadas
- Mudança de gerador de bits pseudoaleatório

Resultados Experimentais

Falcon512

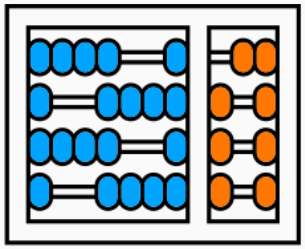
Algoritmo	Cortex-A57	Cortex-X2	Apple M1
Ger. Chaves	8%	3%	**8%**
Ger. Ass.	44%	56%	**79%**
Verificação	44%	33%	**61%**

Falcon1024

Algoritmo	Cortex-A57	Cortex-X2	Apple M1
Ger. Chaves	**7%**	3%	6%
Ger. Ass.	36%	56%	**78%**
Verificação	49%	32%	**58%**

Tabela: Resumo dos ganhos da nossa implementação quando comparada à implementação de referência.

Agradecimentos



Instituto de Computação

UNIVERSIDADE ESTADUAL DE CAMPINAS