

# Reticulados Compactos e Decodificadores LWE

Ramon Ribeiro

Instituto de Computação - UNICAMP

01/10/2018

# Tópicos

- Reticulados Compactos
- Reticulados Criptograficamente Relevantes
- Decodificadores LWE Existentes
- Aplicação dos Decodificadores no Reticulado Barnes-Wall
- Referências

# Empacotamento

Se denota como  $\delta_n$  a maior densidade de empacotamento por esferas em  $R^n$ .

Sendo que um reticulado  $\mathcal{L}_n$  é um subgrupo de  $R^n$ , um reticulado perfeito seria aquele cuja densidade  $\delta$  fosse igual a  $\delta_n$ . Implicando em um base melhor para criptografia, visto que possuem as menores distancias possíveis entre vetores.

# Reticulados Perfeitos

dim.	Nr. of perfect lattices	Absolute maximum of $\gamma$ realized by
2	1 (Lagrange)	$A_{hex}$
3	1 (Gauss)	$A_3$
4	2 (Korkine & Zolotareff)	$D_4$
5	3 (Korkine & Zolotareff)	$D_5$
6	7 (Barnes)	$E_6$
7	33 (Jaquet)	$E_7$
8	10916 (Dutour, Schürmann & Vallentin)	$E_8$

# Reticulados com os Melhores Empacotamentos Conhecidos

Dim.	Simbolo	Nome
9	$\Lambda_9$	Laminated lattice
10	$K_{10}$	Coxeter Todd lattice
11	$\Lambda_{11}$	Laminated lattice
12	$K_{12}$	Coxeter Todd lattice
16	$BW_{16}$	Barnes-Wall lattice
24	$\Lambda_{24}$	Leech lattice

# Reticulados Criptograficamente Relevantes

Dim.	Simbolo	Nome
8	$E_8$	Perfect 8 dim. lattice
9	$\Lambda_9$	Laminated lattice
16	$BW_{16}$	Barnes-Wall lattice
24	$\Lambda_{24}$	Leech lattice

# O Problema LWE

Para  $n, q$  inteiros positivos,  $\chi$  uma distribuição de probabilidade em  $\mathbb{Z}$  e  $s$  um vetor secreto em  $\mathbb{Z}_q^n$ .

Denotamos, portanto,  $L_{s,\chi}$  como a distribuição de probabilidade em  $\mathbb{Z}_q^n \times \mathbb{Z}_q$ , ao escolher  $a \in \mathbb{Z}_q^n$  aleatória e uniformemente, escolhendo  $e \in \mathbb{Z}$  de acordo com  $\chi$  e considerando-o contido em  $\mathbb{Z}_q$ .

Obtendo:

$$(a, c) = (a, \langle a, s \rangle + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$$

# Decision-LWE e Search-LWE

## Decision-LWE

É o problema em decidir se os pares  $(a, c) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$  são obtidos de acordo com  $L_{s, \chi}$  ou a distribuição uniforme em  $\mathbb{Z}_q^n \times \mathbb{Z}_q$



# Decision-LWE e Search-LWE

## Decision-LWE

É o problema em decidir se os pares  $(a, c) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$  são obtidos de acordo com  $L_{s, \chi}$  ou a distribuição uniforme em  $\mathbb{Z}_q^n \times \mathbb{Z}_q$

## Search-LWE

É o problema em recuperar  $s$  de  $(a, c) = (a, \langle a, s \rangle + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ , obtido de  $L_{s, \chi}$

## SIS - Short Integer Solutions (Decision-LWE)

Pode-se tentar encontrar um vetor pequeno  $\mathbf{v}$  tal que  $\mathbf{v} \cdot \mathbf{a} = 0$ , a fim de distinguir se  $m$  casos  $(\mathbf{a}, \mathbf{c})$  seguem/pertencem a  $L_{S, \chi}$ , e, portanto, satisfazem  $c = \langle \mathbf{a}, \mathbf{s} \rangle + e$ , ou se  $\mathbf{c}$  é uniformemente aleatório.

## SIS - Perspectiva de reticulados

Dessa perspectiva, procura-se obter um vetor  $\mathbf{v}$  no reticulado dual escalado (por  $\mathbf{q}$ ) gerado por  $\mathbf{a}$ .

Considere  $\langle \mathbf{v}, \mathbf{c} \rangle$ , se  $\mathbf{c} = \mathbf{a}s + \mathbf{e}$  então  $\langle \mathbf{v}, \mathbf{c} \rangle$  seguindo a distribuição gaussiana em  $\mathbb{Z} \pmod{q}$ . Do outro lado, se  $\mathbf{c}$  for uniforme, então  $\langle \mathbf{v}, \mathbf{c} \rangle$  é uniforme em  $\mathbb{Z}_q$ .

Deve-se manter  $\|\mathbf{v}\|$  pequeno o suficiente para manter a distribuição gaussiana de  $\langle \mathbf{v}, \mathbf{e} \rangle$  boa para se distinguir de distribuições aleatórias.

## BDD - Bounded Distance Decoder (Search-LWE)

Dadas amostras de  $(a, c) = (a, as + e)$  dentro de  $L_{s, \chi}$ , pode-se observar que  $\mathbf{c}$  é próximo a uma combinação linear das colunas de  $\mathbf{a}$ . Além disso, o ruído é gaussiano, sendo que quase todo ele está contido em três vezes o desvio padrão  $(\frac{3\alpha q}{\sqrt{2\pi}})$ .

Logo, o problema seria achar o ponto  $w = as$  do qual  $\mathbf{c}$  está contido através do limitante. A partir disso recuperaremos  $\mathbf{s}$  com álgebra linear.

## Solving for $s$ (Search-LWE)

Usa uma estratégia similar a anterior, só que busca  $s$  diretamente, tal que  $\|as - c\|$  seja mínimo.

## BBD - Tipos de algoritmos (Search-LWE)

- Parallel Bounded Distance Decoding Algorithm - Micciancio e Nicolosi
- List-Decoding Algorithm - Grigorescu e Peikert†

## Reticulado de Barnes-Wall

Sendo  $\mathbb{G} = \mathbb{Z} + i\mathbb{Z}$  o grupo de inteiros gaussianos, definimos  $\phi = 1 + i$  como o inteiro gaussiano de menor norma. Dessas definições, escreve-se o reticulado assim:

$$BW^n = \left[ \begin{array}{cc} 1 & 1 \\ 0 & \phi \end{array} \right]^{\otimes n}.$$

Definindo  $N = 2^n$  como a dimensão do reticulado de Barnes-Wall, vê-se que ele possui distancia mínima  $d_{\min}(BW^n) = \sqrt{N}$ , volume  $V(BW^n) = \sqrt{N^N}$ , e ganho nominal de código  $\gamma_c(BW^n) = \sqrt{N}$ .

# Reticulado de Barnes-Wall

Definindo o produto de Kronecker deste modo:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix}$$

Temos que o reticulado pode ser representado assim:

$$BW^n = \begin{bmatrix} BW^{n-1} & BW^{n-1} \\ \mathbf{O} & \phi \cdot BW^{n-1} \end{bmatrix}$$



# Parallel Bounded Distance Decoding Algorithm

O tempo de uso do algoritmo (paralelo), medido em termos de operações aritméticas é  $O(N \log^2 N / \sqrt{p})$ .

# Parallel Bounded Distance Decoding Algorithm

---

**Algorithm 1** Parallel Bounded Distance Decoder (BDD) for Barnes-Wall Lattices

---

```

1: function PARBW( $p, s$ )
2:   if  $p < 4$  or  $s \in \mathbb{C}^1$  then
3:     return SEQBW(0,  $s$ )
4:   else
5:      $[s_0, s_1] \leftarrow s$ 
6:      $[s_-, s_+] = (\phi/2) \cdot [s_0 - s_1, s_0 + s_1]$ 
7:      $\begin{bmatrix} z_0 \\ z_1 \\ z_- \\ z_+ \end{bmatrix} \leftarrow \begin{bmatrix} \text{PARBW}(p/4, s_0) \\ \text{PARBW}(p/4, s_1) \\ \text{PARBW}(p/4, s_-) \\ \text{PARBW}(p/4, s_+) \end{bmatrix}$ 
8:      $z_0^- \leftarrow [z_0, z_0 - 2\phi^{-1} z_-]$ 
9:      $z_0^+ \leftarrow [z_0, 2\phi^{-1} z_+ - z_0]$ 
10:     $z_1^- \leftarrow [2\phi^{-1} z_- + z_1, z_1]$ 
11:     $z_1^+ \leftarrow [(2\phi^{-1} z_+ - z_1), z_1]$ 
12:     $z = \underset{z' \in \{z_0^-, z_0^+, z_1^-, z_1^+\}}{\text{argmin}} \{ \|s - z'\| \}$ 
13:    return  $z$ 
14:   end if
15: end function

```

▷ Run the sequential decoder from Section 3  
 ▷ Split  $s$  into two halves  
 ▷ Compute  $T(s)$   
 ▷ Execute recursive calls in parallel  
 ▷ Compute 4 candidate vectors  
 ▷ Select the candidate closest to  $s$

---

# Parallel BDD - Sequential BDD

---

**Algorithm 2** Sequential Bounded Distance Decoder for Barnes-Wall Lattices and Their Principal Sublattices

---

```

function SEQBW( $r, s$ )
  if  $s \in \mathbb{C}^N$  with  $N \leq 2^r$  then
    return  $\lceil s \rceil \in \mathbb{G}^N$  ▷ Round  $s$  component-wise to the closest Gaussian integer
  else
     $\mathbf{b} \leftarrow \lceil \Re(s) \rceil + \lceil \Im(s) \rceil \bmod 2$  ▷ Compute binary target component-wise
     $\rho = 1 - 2 \max(|\Re(s) - \lceil \Re(s) \rceil|, |\Im(s) - \lceil \Im(s) \rceil|)$  ▷ Compute the reliability information
     $\mathbf{t} \leftarrow (\mathbf{b}, \rho)$  ▷ Component-wise pairing, i.e.,  $t_j = (b_j, \rho_j)$ 
     $\psi(\mathbf{c}) \leftarrow \text{RMDEC}^\psi(r, \mathbf{t})$  ▷ Call the Reed-Muller soft-decision decoder
     $\mathbf{v} \leftarrow \text{SEQBW}(r+1, (s - \psi(\mathbf{c}))/\phi)$ 
    return  $\psi(\mathbf{c}) + \phi \mathbf{v}$ 
  end if
end function

```

---

## Parallel BDD - Definições

Cada vetor do reticulado  $BW^n$  pode ser unicamente representado assim:

$$v = \sum_{r=0}^{n-1} \phi^r \psi(c_r) + \phi^n c_n$$

em que  $c_n \in \mathbb{G}^N$  e  $c_r \in RM_r^n$ , para  $r=0,1,\dots,n-1$ .

Também tem-se que:

$$\begin{cases} \psi(0) = 0 \\ \psi(1) = 1 \\ \psi([u, u \oplus v]) = [\psi(u), \psi(u) + \psi(v)] \end{cases}$$

# Parallel BBD - Soft Decision

---

**Algorithm 3** Soft Decision Decoder for Reed-Muller Codes

---

```

function RMDECψ( $r, \mathbf{t}$ ) ▷ Input:  $r \geq 0, \mathbf{t} \in (\{0, 1\} \times \{0, 1\})^N$ 
  if  $r = 0$  then
    if  $\sum_{b_j=0} \rho_j > \sum_{b_j=1} \rho_j$  then
      return  $[0, \dots, 0]$ 
    else
      return  $[1, \dots, 1]$ 
    end if
  else if  $N = 2^r$  then
    return  $[b_1, \dots, b_N]$  ▷ where  $(b_j, \rho_j) = t_j$ 
  else
     $[\mathbf{t}^0, \mathbf{t}^1] \leftarrow \mathbf{t}$  ▷ Split  $\mathbf{t}$  into halves
    for  $j = 1, \dots, N/2$  do
       $t_j^+ \leftarrow (b_j^0 \oplus b_j^1, \min(\rho_j^0, \rho_j^1))$  ▷ where  $(b_j^0, \rho_j^0) = t_j^0$  and  $(b_j^1, \rho_j^1) = t_j^1$ 
    end for
     $\mathbf{v} \leftarrow \text{RMDEC}^\psi(r-1, \mathbf{t}^+)$ 
    for  $j = 1, \dots, n/2$  do
      if  $b_j^0 \oplus b_j^1 = v_j \bmod 2$  then
         $t_j^- \leftarrow (b_j^0, (\rho_j^0 + \rho_j^1)/2)$ 
      else
         $t_j^- \leftarrow (b_j^0 \oplus \text{EVAL}(\rho_j^0 < \rho_j^1), |\rho_j^0 - \rho_j^1|/2)$  ▷ where  $\text{EVAL}(\varphi) = 1$  iff formula  $\varphi$  holds
      end if
    end for
     $\mathbf{u} \leftarrow \text{RMDEC}^\psi(r, \mathbf{t}^-)$ 
    return  $[\mathbf{u}, \mathbf{u} + \mathbf{v}]$ 
  end if
end function
  
```

---

# Referências

- Sphere packings and lattice sphere packings - Mathieu Dutour Sikiric
- A Simple Construction for the Barnes-Wall Lattices
- Efficient Bounded Distance Decoders for Barnes-Wall Lattices  
- Daniele Micciancio and Antonio Nicolosi April 30, 2008
- List Decoding Barnes-Wall Lattices - Elena Grigorescu and Chris Peikert April 10, 2012

# Referências

- On the concrete hardness of Learning with Errors - Martin R. Albrecht, Rachel Player, and Sam Scott
- On Lattices, Learning with Errors, Random Linear Codes, and Cryptography - Oded Regev May 2, 2009
- Post-Quantum Cryptography - Daniel J. Bernstein and Johannes Buchmann (Lattice-based Cryptography Chapter)
- Lecture 3 - CVP Algorithm - Lecturer: Oded Regev