Lattices: Hard Problems and Cryptographic Techniques

Marta Paes Moreira

2 de Janeiro de 2013

Resumo

RESUMO

Conteúdo

1 Intoduction

1.1 Lattices

jjajjajajajajja

2 Problemas Hard em Reticulados

2.1 Vectores Curtos

No sentido de compreender a dimensão do problema dos vectores curtos, torna-se necessário abordar, previamente, duas temáticas que constituem a sua base: os Mínimos Sucessivos de Minkowski (2.1.1) e a Constante de Hermite (2.1.2).

2.1.1 Mínimos Sucessivos de Minkowski

Lattices' inherent discreteness property, mentioned in Section 1.1, causes any lattice L of rank at least 1 to have a nonzero lattice vector that is close to the origin, whose norm represents the shortest distance possible between any two distinct lattice vectors - the *first successive minimum* $\lambda_1(L)$. Figure 2.1.1 illustrates the concept for a lattice *L(a,b)*, being vector x the first successive minimum with respect to the Euclidian norm [1, 2, 3].

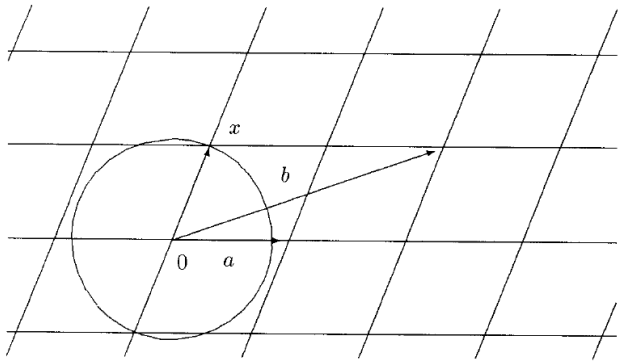

Figura 1: First successive minimum $\lambda_1(L(a,b))$[1].

The application of Minkowski's concept to other successive minima is generalized as follows.

Definição 1. *For i=1,...,d the ith successive minimum of the lattice L of rank d is*

$$\lambda_i(L)=min\{max\|x_1\|,...,\|x_i\|\},\ |x_1,...x_i| \in L \text{ are linearly independent.}$$

Following these assumptions, one could assume that any lattice should have a basis of which the norms are precisely the successive minima. However, despite the fact that there always exist linearly independent lattice vectors $\mathbf{v_1}, ..., \mathbf{v_d}$ reaching simultaneously the minima — that is, $\|\mathbf{v_i}\|=\lambda_i(L)$ for all i —, as soons as $\dim(L) \geqslant 4$, such vectors do not necessarily form a lattice basis [2]. As an example, consider the four-dimensional lattice L defined as the set of all $(x_1, x_2, x_3, x_4) \in \mathbb{Z}^4$ such that $\sum_{i=1}^{4} x_i$ is even. A possible basis of

L can be represented by Matrix A. Matrix B, on the other hand, isn't a suitable basis: although composed of linearly independent row vectors which also reach all the minima, its determinant is equal to 4.

$$A = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} \qquad B = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix}$$

This concept can be further reinforced by multiplying both matrices, as it has been proven that two basis give the same lattice if and only if their matrices are related by a unimodular matrix [4].

$$A \times B = \begin{pmatrix} 0 & -2 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix}$$

2.1.2 Constante de Hermite

A prova de que a quantidade $\lambda_1(L)/\mathrm{vol}(L)^{1/d}$ pode ser limitada, para todo o reticulado L de *rank d* foi dada, pela primeira vez, por Hermite, e dita que:

Definição 2. *O supremo de $\lambda_1(L)^2/vol(L)^{2/d}$ para todo o reticulado L de rank d é representado por γ_d e designado por Constante de Hermite de dimensão d.*

Embora seja assente que γ_d é atingido, o processo de cálculo do seu valor exacto constitui um problema árduo, sendo que este é apenas conhecido para os casos em que $1 \leqslant d \leqslant 8$ e $d = 24$.

Tabela 1: Valores actualmente conhecidos para a Constante de Hermite [2]

d	2	3	4	5	6	7	8	24
γ_d	$2/\sqrt{3}$	$2^{1/3}$	$\sqrt{2}$	$8^{1/5}$	$(64/3)^{1/6}$	$64^{1/7}$	2	4
Aproximação	1.1547	1.2599	1.4142	1.5157	1.6654	1.8114	2	4

Para todas as dimensões $d \geqslant 1$, existe um reticulado L de *rank d* que verifica os valores da Constante de Hermite explicitados na Tabela 1 — denominado de reticulado crítico. Da mesma forma, todos os reticulados críticos para cada uma das dimensões mencionadas são conhecidos [2].

2.1.3 Shortest Vector Problem

Com base na definição de mínimo postulada em 2.1.1 — que pode ser generalizada como a distância mínima entre dois pontos de um reticulado —, torna-se possível definir a primeira problemática no âmbito dos *hard problems*: a de encontrar um vector não nulo que atinja este valor — `Shortest Vector Problem (SVP)` (Figura 2) [5, 6]. Releve-se que o vector mais curto de um reticulado não é único, dado que, para além da possibilidade de existirem outros vectores de norma semelhante, há que considerar que $\|Bx\| = \|-Bx\|$ [3]. O número de vectores curtos compreendido num reticulado L denomina-se de *kissing number* e é superiormente delimitado [2].

Definição 3. *Dado um reticulado L(B), encontrar um vector não nulo Bx, $x \in \mathbb{Z}^k$, de comprimento (no máximo) $\|Bx\| \leqslant \lambda_1$.*

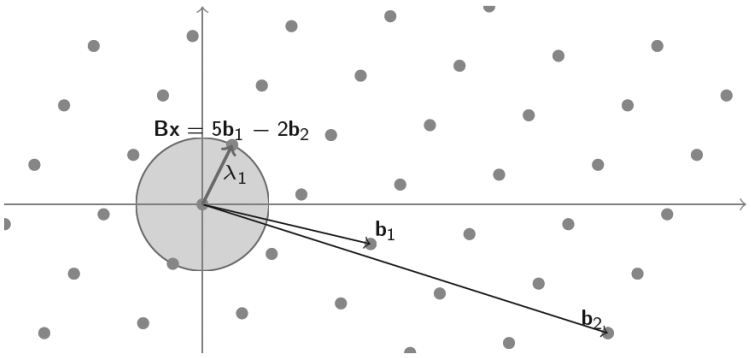

Figura 2: Representação gráfica do `Shortest Vector Problem` (SVP).

NP HARD, ETC!!!!!!!!!!!!!!!! [49 vpol]

2.1.4 Approximate Shortest Vector Problem

Por vezes, as aplicações não requerem um vector que seja efectivamente "o mais curto", sendo suficiente encontrar um vector que seja "suficientemente curto"— `Approximate Shortest Vector Problem (SVP`γ`)` (Figura 3) [5, 6].

Definição 4. *Dado um reticulado L(B) e um factor de aproximação $\gamma \geqslant 1$, encontrar um vector não nulo Bx, $x \in \mathbb{Z}^k$, de comprimento (no máximo) $\|Bx\| \leqslant \gamma\lambda_1$.*

2.1.5 Shortest Independent Vectors Problem

No caso particular em que se torna necessário encontrar não um, mas um conjunto de n vectores linearmente independentes — isto é, uma base de L — o problema denomina-se de `Shortest Independent Vectors Problem (SIVP)` (Figura 4) [5, 7].

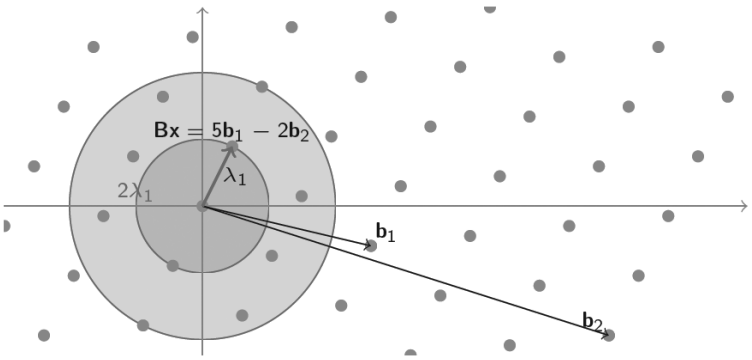

Figura 3: Representação gráfica do `Approximate Shortest Vector Problem` (SVPγ).

Definição 5. *Dado um reticulado L(B), encontrar um conjunto de n vectores linearmente independentes* $Bx_1, ...Bx_n$ *de comprimento (no máximo)* $max_i\|Bx_i\| \leqslant \lambda_n$.

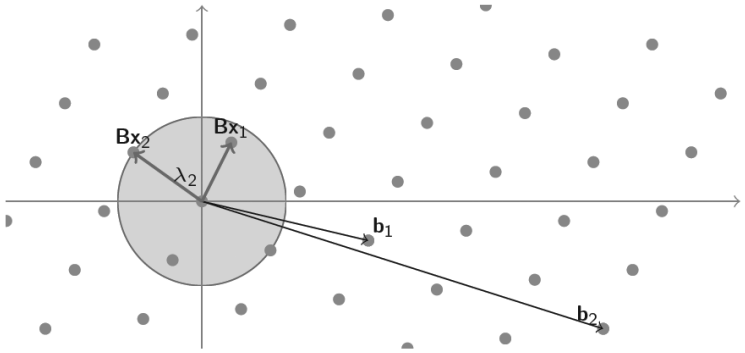

Figura 4: Representação gráfica do `Shortest Independent Vectors Problem` (SIVP).

2.1.6 Approximate Shortest Independent Vectors Problem

De forma análoga à variante do `Shortest Vector Problem` exposta na Sub-Secção 2.1.4, também o `Shortest Independent Vectors Problem (SIVP`γ`)` está associado a um problema que visa encontrar não o conjunto de vectores mais curtos, mas um conjunto de vectores "suficientemente curtos"(Figura 5) [5, 7].

Definição 6. *Dado um reticulado L(B), encontrar um conjunto de n vectores linearmente independentes* $Bx_1, ...Bx_n$ *de comprimento (no máximo)* $max_i\|Bx_i\| \leqslant \gamma\lambda_n$.

2.2 Vectores Próximos

sgsgshshshshshhs

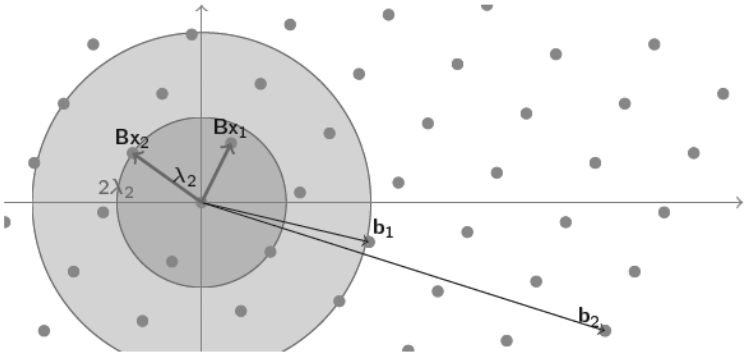

Figura 5: Representação gráfica do `Approximate Shortest Independent Vectors Problem (SIVP`γ`)`.

2.2.1 Closest Vector Problem

Um outro problema *hard* que surge no âmbito dos reticulados diz respeito à noção de vector mais próximo: dado um determinado ponto-alvo $t \in \mathbb{R}^d$ — não necessariamente pertencente ao reticulado *L(B)* —, o vector Bx mais próximo corresponde ao ponto de $L(B)$ que minimiza a distância entre os dois (Figura 6) [6, 5, 7].

Definição 7. *Dados um reticulado L(B) e um ponto-alvo t, encontrar um ponto x do reticulado que minimize a distância* $d(t-x) = \|t-x\|$*, isto é, o vector Bx à distância* $\|Bx-t\| \leqslant \mu$ *de t.*

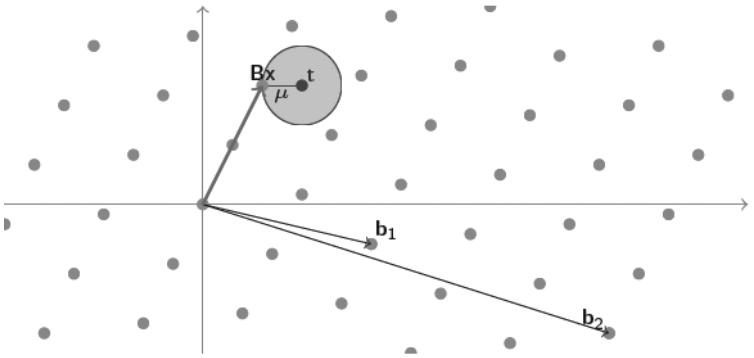

Figura 6: Representação gráfica do `Closest Vector Problem (CVP)`.

INTERCONVERSÃO COM SVP!!!
NP HARD, ETC!!!!!!!!!!!!!!!! [6,14 vpol]

2.2.2 Approximate Closest Vector Problem

Uma relaxação possível para o problema anterior consiste, igualmente, em de considerar um factor de aproximação γ que permita obter não o vector mais próximo, mas um vector

"suficientemente próximo"(Figura 7) [6, 5, 7].

Definição 8. *Dados um reticulado L(B) e um ponto-alvo t, encontrar um ponto x do reticulado que minimize a distância* $d(t-x) = \|t-x\|$*, isto é, o vector Bx à distância* $\|Bx-t\| \leqslant \gamma\mu$ *de t.*

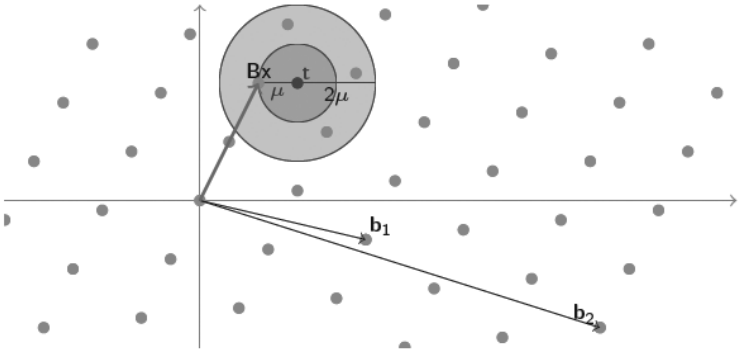

Figura 7: Representação gráfica do `Approximate Closest Vector Problem (CVP`γ`)`.

3 Resolução de Problemas Hard em Reticulados

sssssssssss

3.1 Enumeração

A resolução do `SVP` por intermédio da técnica de enumeração baseia-se no teste de todas as combinações possíveis de vectores de uma base, destacando, entre todas as possibilidades, o vector mais curto. Naturalmente, testar todas as combinações possíveis implicaria uma quantidade infinita de vectores, pelo que se torna necessário equacionar e explorar um método associado à enumeração que veicule um limite a esta quantidade — a ortagonalização de `Gram Schmidt` [5, 6].

3.1.1 Ortogonalização de Gram Schmidt

A existência de uma base ortonormal facilita a procura dos coeficientes necessários à descrição de um vector enquanto combinação linear dos vectores $v_1, ..., v_n$, pertencentes à base de um espaço vectorial.

$$v = \frac{(v, v_1)}{\|v_1\|^2} v_1 + ... + \frac{(v, v_n)}{\|v_n\|^2} v_n \tag{1}$$

Neste sentido, o processo de `Gram Schmidt` surge como um método de ortonormalização de bases, através do escalonamento de um vector da base de forma a que este tenha comprimento igual a um. Os vectores seguintes são iterativamente projectados no complemento ortogonal do *span* de vectores anteriores e igualmente escalonados. Quando os vectores não são escalonados no decorrer deste procedimento, a base resultante será ortogonal em vez de ortonormal, isto é, embora o produto interno entre todos os vectores da base seja $u.v = 0$, estes não são vectores unitários.

1 Ortogonalização de Gram Schimdt

Dada uma base $B = b_1, ..., b_d$ de um reticulado L, calcular, para cada $i = 1, ..., d$:

1. Para cada $j < i$, calcular $\mu_{i,j} = s_j^{-1}(b_i \cdot b_i^*$, sendo $s_j = b_j^* \cdot b_j^*$;
2. $b_j^* = b_1 - \sum_{j<i} \mu_{i,j} b_j^*$, onde $\mu_{i,j} = \frac{\langle b_i, b_j^* \rangle}{\|b_j^*\|^2}$, para todo $1 \leqslant j < i \leqslant d$.

No que concerne a aplicação deste método no âmbito dos reticulados, não existe qualquer garantia de que a projecção de um vector do reticulado no complemento ortogonal de um outro vector do reticulado se encontre necessariamente no reticulado, nem tão pouco é sempre possível escalonar um vector nas condições anteriormente referidas, pelo que nem sempre é

possível ortogonalizar as bases. Não obstante, o processo de `Gram Schmidt` desempenha um papel fundamental nos métodos de redução e enumeração aplicados a bases de reticulados. COMPLETAR!!!!

Uma forma simplificada de representar o algoritmo de enumeração é por intermédio de uma árvore de pesquisa (Figura 8), cujos nós correspondem a um qualquer vector. Considerando que a raiz se encontra no nível 0, o nível i da árvore é composto por todos os vectores de π_{d-i+1}, para $0 \leqslant i \leqslant d$. Sendo v um nó do nível i, isto é, $v \in \pi_{d-i+1}$, a respectiva descendência consiste no conjunto de todos os vectores $u \in \pi_{d-i+2}(L)$ projectados em v: $v = \pi_{d-i+1}(u)$. No seguimento deste raciocínio, infere-se que a raiz consiste do vector π_{d+1}, o primeiro nível no conjunto de vectores pertencentes a $\pi_d(L) = L(b_d^*)$ de norma, no máximo, R, e assim sucessivamente, até ao nível d, que contém todos os vectores de $\pi_1(L)$ de norma, no máximo, R.

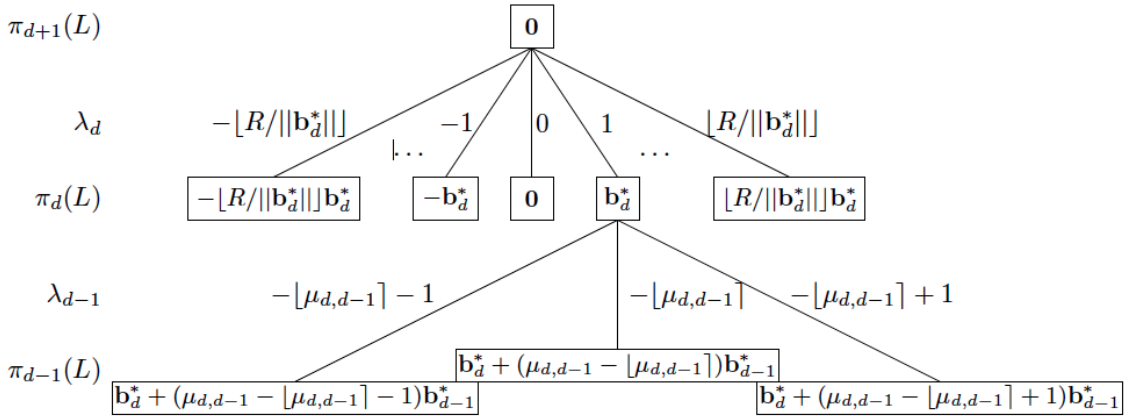

Figura 8: Esquematização dos dois primeiros níveis de uma árvore de enumeração. [6]

Uma outra forma de descrever a enumeração de forma simplificada é através de um algortimo (Algoritmo 1), recorrendo à noção de árvore supracitada:

Dependendo do grau de precisão do delimitador R, as árvores de enumeração tendem a desenvolver-se exponencialmente, pelo que, pese embora correspondam a um método capaz de retornar uma solução exacta do `SVP`, o tempo de execução não é polinomial e, por essa mesma razão, estes algoritmos são normalmente precedidos de algoritmos de redução. Em alternativa, Schnorr e Hörner propuseram uma técnica de enumeração melhorada, designada de `pruning` [8], recentemente elevada a outro patamar por Gama, Nguyen e Regev, num conceito denominado de `extreme pruning` [9].

Algoritmo 1 Enumeração

Requer: uma base reduzida $\{b_1, ..., b_d\}$ de L e os respectivos coeficientes de Gram-Schmidt $\mu_{i,j}$ e normas $\|b_i^*\|$
Garante: o vector de *output* do reticulado $\sum_i u_i b_i \in L$ é um vector curto de L

1: **repetir**
2: **se** a norma do nó actual for menor do que o limite, **então**
3: descer um nível, até ao descendente de norma mínima
4: **else**
5: subir um nível, até ao vizinho não visitado do ascendente de menor norma
6: **end if**
7: **até** que todos os nós tenham sido percorridos

3.2 Sieving

Em contraposição — ou como alternativa — à superexponencialidade da enumeração, surgem os algoritmos de `sieving`, que revelam tempos de execução exponenciais. Porém, mesmo a variante prática mais rápida deste algoritmo conhecida é ultrapassada, em termos de performance, pelo algoritmo de enumeração de Schnorr-Hörner, até $n \approx 50$) (dimensão acima da qual este se torna impraticável) [3, 10]. Abaixo, apresenta-se um possível algoritmo de `sieving` — o algoritmo de Micciancio-Voulgaris (Algoritmo 2).

Sumariamente, o algoritmo procura exaurir o espaço de vectores curtos do reticulado, adicionando sucessivamente vectores curtos a uma lista Q até que os vectores $\mathbf{v_i}$ e $\mathbf{v_j}$ em Q estejam afastados, no máximo, à distância μ ou se tenham usado demasiadas amostras. Caso se encontrem dois vectores $\mathbf{v_1}, \mathbf{v_j} \in L$ nas condições rederidas, então, o vector $\mathbf{v} = \mathbf{v_i} - \mathbf{v_j}$ também pertence ao reticulado e tem um comprimento, no máximo, de μ, e a solução retornada pelo algoritmo é $\mathbf{v}$. Em caso contrário, novos vectores são adicionados à lista — cujo tamanho é limitado por uma constante fixa. No cômputo geral, é possível garantir, com elevada probabilidade, que, a dada altura, se abandona o ciclo representado no Algoritmo 2, ao encontrar um vector curto do reticulado [3].

Actualmente, para além de os próprios Micciancio e Voulgaris terem proposto uma melhoria ao seu algoritmo de `sieving` — o denominado `Algoritmo de GaussSieve` —, Ajtai et al. [**?**] propuseram uma abordagem distinta, que vai ao encontro do conceito de `sieving` de forma mais explícita: em vez de exaurir o espaço de vectores curtos numa lista de vectores de dimensão substancialmente elevada, começa-se por uma lista de vectores longos, que vai sendo reduzida em tamanho, paralelamente à redução das normas dos vectores nela contidos, através de sucessivas iterações. Na mesma linha de construção, inserem-se as propostas de Nguyen e Vidick [**?**], e Wang et al. [**?**], a título de exemplo, que não serão exploradas no presente trabalho.

Algoritmo 2 Algoritmo de `sieving` Micciancio-Voulgaris

Requer: uma base reduzida $\{b_1, ..., b_d\}$ de L e um valor $\mu \in \mathbb{R}$

Garante: se $\mu > \lambda_1(L)$, então, com grande probabilidade, o algoritmo encontra um vector $v \in L$ com $\|v\| \leqslant \mu$

$Q \leftarrow \{0\}$
$\xi \leftarrow 0.685$
$N \leftarrow poly(d) \cdot 2^{3.199d}$
for $i = 1$ to N **do**
    $\mathbf{e_i} \in_R \mathbf{B_d}(\mathbf{0}, \xi, \mu)$
    $\mathbf{r_i} \leftarrow \mathbf{e_i} \bmod \mathbf{B}$
    while $\exists \mathbf{v_j} \in Q : \|\mathbf{r_i} - \mathbf{v_j}\| \leqslant (1 - \frac{1}{d}\|\mathbf{v_i}\|$ **do**
        $\mathbf{r_i} \leftarrow \mathbf{r_i} - \mathbf{v_j}$
    end while
    $\mathbf{v_i} \leftarrow \mathbf{r_i} - \mathbf{e_i}$
    se $\mathbf{v_i} \notin Q$ **then**
        se $\exists \mathbf{v_j} \in Q : \|\mathbf{v_i} - \mathbf{v_j}\| < \mu$ **then**
            return $\mathbf{v_i} - \mathbf{v_j}$
        end if
        $\mathbf{Q} \leftarrow \mathbf{Q} \cup \{\mathbf{v_i}\}$
    end if
end for
return $\perp$

4 Cryptographic Techniques

4.1 Hash Functions

4.2 Public Key Encryption Schemes

4.3 Digital Signature Schemes

5 Cryptographic Attacks

hhhhh

6 Conclusion

jjjjjjjj

Referências

[1] C. Dwork. Lattices and their application to cryptography. Lecture Notes, 1998. `http://theory.stanford.edu/~csilvers/cs359/`.

[2] P. Q. Nguyen. *Hermite's Constant and Lattice Algorithms*. Information Security and Cryptography. Springer, 2009.

[3] B. Weger T. Laarhoven, J. Pol. Solving hard lattice problems and the security of lattice-based cryptosystems. Cryptology ePrint Archive, Report 2012/533, 2012. `http://eprint.iacr.org/`.

[4] Geometry of numbers: Determinant of the lattice and the fundamental parallelepiped. Number Theory Reading Group, 2008. `http://numbertheoryreadinggroup.wordpress.com/2008/04/24/`.

[5] Department of Computer Science and Engineering, University of California. *The Geometry of Lattice Cryptography*, 2011.

[6] J. van de Pol. Lattice-based criptography. Master's thesis, Department of Mathematics and Computer Science, Eindhoven University of Technology, 2011.

[7] J. Valença. *Introdução à Segurança da Informação, Códigos e Criptografia*. Departamento de Informática, Universidade do Minho, 2012.

[8] C. P. Schnorr, H.H. Hörner, Johann Wolfgang, and Goethe universitat Frankfurt. Attacking the chor-rivest cryptosystem by improved lattice reduction. Springer-Verlag, 1995.

[9] N. Gama, P. Q. Nguyen, and O. Regev. Lattice enumeration using extreme pruning. In *Advances in Cryptology - Proceedings of EUROCRYPT '10*, volume 6110 of *LNCS*. Springer, 2010.

[10] D. Micciancio and P. Voulgaris. Faster exponential time algorithms for the shortest vector problem. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '10. Society for Industrial and Applied Mathematics, 2010.

[11] M. Ajtai, R. Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *In STOC*. ACM, 2001.

[12] P. Q. Nguyen and T. Vidick. Sieve algorithms for the shortest vector problem are practical. *J. of Mathematical Cryptology*, 2(2), 2008.

[13] Xiaoyun Wang, Mingjie Liu, Chengliang Tian, and Jingguo Bi. Improved nguyen-vidick heuristic sieve algorithm for shortest vector problem. In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*. ACM, 2011.