Criptografia e Reticulados

Caio Teixeira, Ramon Ribeiro e Tomás S. R. Silva

Instituto de Computação - UNICAMP

21/08/2018

O que é criptografia?

Criptografia: A ciência de ocultar mensagens, com o objetivo de esconder seu significado.

Criptoanálise: A ciência de *quebrar* sistemas criptográficos e/ou garantir sua segurança.

Por quê?

- Possibilitar a **comunicação de informação sensível** através de canais inseguros (potencialmente monitorados ou manipulados por adversários).
- Garantir **autenticidade** de documentos através de assinaturas digitais, tais que outras pessoas possam confirmar sua validade mas que ninguém consiga forjar a assinatura de outra pessoa.
- Garantir a **integridade** de um documento através da criação de um resumo (*hash*).

Por quê?

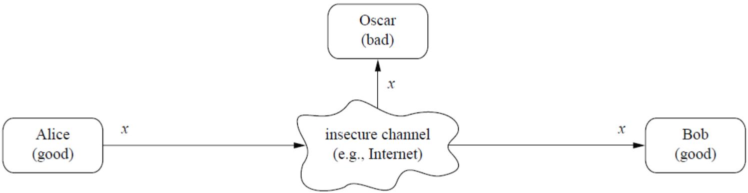


Figure 1: Representação de um canal inseguro

Como?

Precisamos de uma transformação da mensagem para uma sequência aparentemente aleatória de símbolos que não retenha **nenhuma informação** sobre a mensagem original.

Além disso, precisamos de algo que permita ao destinatário recuperar a mensagem a partir de tal sequência; ou seja, uma **chave**.

Como?

Funções de encriptação e decriptação

Seja $\mathcal{K}$ o espaço de possíveis chaves, $\mathcal{M}$ o espaço de possíveis mensagens em claro e $\mathcal{C}$ o espaço de possíveis mensagens cifradas. Queremos $e : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$ e $d : \mathcal{K} \times \mathcal{C} \to \mathcal{M}$ tais que

$$d(k, e(k, m)) = m, \quad \forall k \in \mathcal{K}, m \in \mathcal{M}$$

Outras notações para estas funções são $e_k : \mathcal{M} \to \mathcal{C}$ e $d_k : \mathcal{C} \to \mathcal{M}$.

Como?

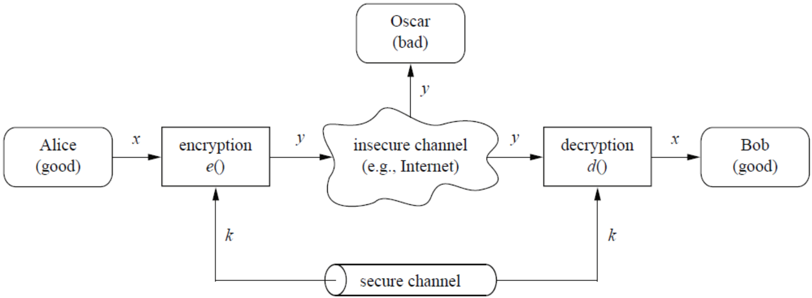
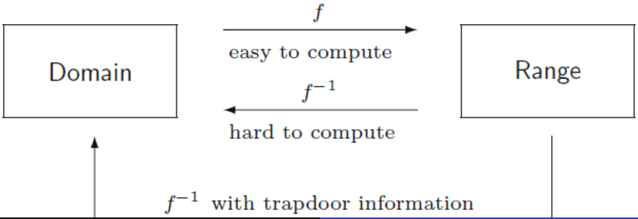


Figure 2: Representação do uso das funções criptográficas

Propriedades importantes

1. Para quaisquer pares (k,m) de chave $k \in \mathcal{K}$ e mensagem em claro $m \in \mathcal{M}$, deve ser fácil computar a mensagem cifrada $e_k(m)$.
2. Para quaisquer pares (k,c) de chave $k \in \mathcal{K}$ e mensagem cifrada $c \in \mathcal{C}$, deve ser fácil computar a mensagem em claro $d_k(c)$.
3. Dadas uma ou mais mensagens cifradas $c_1, c_2, ..., c_n \in \mathcal{C}$ encriptadas usando a chave $k \in \mathcal{K}$, deve ser **muito difícil** computar quaisquer mensagens em claro $d_k(c_1)$, $d_k(c_2)$, ..., $d_k(c_n)$ correspondentes sem conhecimento de k.



Propriedades importantes

4. Dados um ou mais pares de mensagens em claro e suas correspondentes mensagens cifradas, $(m_1, c_1), (m_2, c_2), ..., (m_n, c_n)$ deve ser **muito difícil** decriptar qualquer mensagem cifrada c que não esteja na lista sem o conhecimento de k.
5. Para qualquer lista de mensagens em claro $m_1, ..., m_n \in \mathcal{M}$ escolhidas pelo adversário, mesmo com o conhecimento de $e_k(m_1), ..., e_k(m_n)$, deve ser **muito difícil** decriptar qualquer mensagem cifrada c que não esteja na lista sem o conhecimento de k.

Dificuldade de computação

Função de complexidade

Define a quantidade de *unidades de tempo* gastas para a execução de um algoritmo relativa ao tamanho da entrada (n) no pior caso.

$f(n)$	$n = 20$	$n = 40$	$n = 60$	$n = 80$	$n = 100$
n	$2{,}0\times10^{-11}$seg	$4{,}0\times10^{-11}$seg	$6{,}0\times10^{-11}$seg	$8{,}0\times10^{-11}$seg	$1{,}0\times10^{-10}$seg
n^2	$4{,}0\times10^{-10}$seg	$1{,}6\times10^{-9}$seg	$3{,}6\times10^{-9}$seg	$6{,}4\times10^{-9}$seg	$1{,}0\times10^{-8}$seg
n^3	$8{,}0\times10^{-9}$seg	$6{,}4\times10^{-8}$seg	$2{,}2\times10^{-7}$seg	$5{,}1\times10^{-7}$seg	$1{,}0\times10^{-6}$seg
n^5	$2{,}2\times10^{-6}$seg	$1{,}0\times10^{-4}$seg	$7{,}8\times10^{-4}$seg	$3{,}3\times10^{-3}$seg	$1{,}0\times10^{-2}$seg
2^n	$1{,}0\times10^{-6}$seg	1,0seg	13,3dias	$1{,}3\times10^{5}$séc	$1{,}4\times10^{11}$séc
3^n	$3{,}4\times10^{-3}$seg	140,7dias	$1{,}3\times10^{7}$séc	$1{,}7\times10^{19}$séc	$5{,}9\times10^{28}$séc

Figure 3: Tempo de execução supondo um computador com velocidade de 1 Terahertz (mil vezes mais rápido que um computador de 1 Gigahertz).

(Cortesia dos professores Eduardo C. Xavier e Flávio K. Miyazawa)

Dificuldade de computação

$f(n)$	Computador atual	$100\times$mais rápido	$1000\times$mais rápido
n	N_1	$100N_1$	$1000N_1$
n^2	N_2	$10N_2$	$31.6N_2$
n^3	N_3	$4.64N_3$	$10N_3$
n^5	N_4	$2.5N_4$	$3.98N_4$
2^n	N_5	$N_5 + 6.64$	$N_5 + 9.97$
3^n	N_6	$N_6 + 4.19$	$N_6 + 6.29$

Figure 4: Fixando o tempo de execução.

(Cortesia dos professores Eduardo C. Xavier e Flávio K. Miyazawa)

Propriedades de sistemas criptográficos

Operações de Shannon

Confusão: operação de encriptação na qual o relacionamento entre chave e mensagem cifrada é obscurecido.

Difusão: operação de encriptação onde a influência de um símbolo da mensagem em claro é espalhada entre vários símbolos da mensagem cifrada, com o objetivo de esconder propriedades estatísticas da mensagem em claro.

Lei de Kerckhoff

"Um sistema criptográfico deve ser seguro mesmo se o atacante souber todos detalhes sobre o sistema, com a exceção da chave secreta."

Sistemas de chave pública (assimétricos)

"We stand today on the brink of a revolution in cryptography."
- W. Diffie e M. Hellman, *New Directions in Cryptography*, 1976

Cifras assimétricas

Dados os espaços de chave $\mathcal{K}$, de mensagens em claro $\mathcal{M}$ e de mensagens cifradas $\mathcal{C}$, um elemento k de $\mathcal{K}$ é da forma

$$k = (k_{priv}, k_{pub})$$

E as funções de encriptação e decriptação agora são definidas por

$$e_{k_{pub}} : \mathcal{M} \to \mathcal{C}, \text{e}$$

$$d_{k_{priv}} : \mathcal{C} \to \mathcal{M}$$

tais que

$$d_{k_{priv}}(e_{k_{pub}}(m)) = m, \quad \forall m \in \mathcal{M}$$

Sistemas de chave pública (assimétricos)

- Os sistemas de chave pública mais populares hoje em dia se baseiam nos problemas de encontrar um **logaritmo discreto módulo p** ou **fatorar inteiros em números primos**.
- Estes sistemas são mais lentos, e portanto são muito utilizados para acordos de chaves de sistemas simétricos, constituindo assim sistemas híbridos.
- Também são a base para assinaturas digitais.

Criptografia Pós-Quântica

Algoritmo de Shor

Em 1997, Peter Shor descreveu um algoritmo capaz de resolver ambos problemas de **fatoração de inteiros** e **logaritmos discretos** em tempo polinomial, para ser executado em computadores quânticos.

Sistemas pós-quânticos

São sistemas criptográficos baseados em problemas para os quais não se conhece algoritmos (quânticos ou não) que resolvam seu problema base em tempo polinomial.
Alguns deles são baseados em **problemas difíceis em reticulados.**

Problema da soma dos subconjuntos

- Primeira construção de modelo criptográfico baseada em problema NP-completo.
- Autoria de Merkle e Hellman - 70's

Definição do problema

Suponha que lhe dada uma lista de inteiros positivos $M = (m_1, m_2, ..., m_n)$ e uma constante inteira S. O problema consiste em encontrar um subconjunto de M cuja soma dos seus elementos seja S. (assumimos que há pelo menos um subconjunto que satisfaz a condição)

Isto é:

Para $M = (m_1, m_2, ..., m_n)$, queremos encontrar
$X = (x_1, x_2, ..., x_n) : x_i = 0 \vee x_i = 1, i = 1, ..., n$
tal que $S = \sum_{i=1}^{n} x_i m_i$

Se M é umas sequência de super crescimento ($m_{n+1} \geq 2m_n \rightarrow m_n > m_{n-1} + m_{n-2} + ... + m_1$), o problema da soma do subconjunto em (M,S) pode ser resolvido pelo seguinte algoritmo:

De i=n até i=1

  Se $S \geq M_i$, faça $x_i = 1$ e subtraia m_i de S

  Caso contrário, faça $x_i = 0$

Como usamos isso para criar um modelo criptográfico de chave pública?

Suponha que Alice quer enviar uma mensagem cifrada para Bob

1. Alice escolhe uma sequência de super crescimento $R = r_1, ..., r_n$, uma constante $B > 2r_n$ e uma constante $A : gdc(A, B) = 1$
2. Em seguida, Alice computa os termos $m_i \in M$ tal que $m_i = Ar_i \mod B$. M será a chave pública
3. Bob escolhe um texto original binário X. Usando a chave pública M, Bob computa o texto cifrado como sendo $S = X.M$ e envia S para Alice.
4. Por fim, Alice computa $S' = A^-1S \mod B$ e resolve da soma do subconjunto para S' usando a sequência R (o texto original satisfaz $XR = S'$)

Como utilizamos problemas NP em reticulados para construção de modelos criptográficos?

Primeiramente, vamos apresentar alguns problemas NP em reticulados:

Seja $\mathfrak{L} \subset \mathbb{R}^n$ um reticulado,temos os seguintes problemas:

- The Shortest Vector Problem (SVP) : Encontre o menor $v \in \mathfrak{L}$.
- The Closest Vector Problem (CVP) : Dado $w \in \mathbb{R}^n \wedge w \notin \mathfrak{L}$, encontre $v \in \mathfrak{L}$ tal que seja mínima $||w - v||$
- apprSVP : Encontre $v \in \mathfrak{L}$ tal que $||v|| \leq \psi(n)||V_{shortest}||$
- apprCVP : Dado $w \in \mathbb{R}^n \wedge w \notin \mathfrak{L}$, encontre $v \in \mathfrak{L}$ tal que $||w - v|| \leq \psi(n)||V_{closest}||$

- Computacionalmente, tanto o SVP quanto o CVP se tornam tão mais difíceis quanto maior a dimensão do reticulado em análise.
- O CVP é considerado NP-difícil.
- O SVP é considerado NP-difícil sobre certas "hipóteses de redução aleatórias"[a]

[a] Essas hipóteses significam que a classe de algoritmos de tempo polinomial é ampliada para incluir aqueles que não são determinísticos, mas mesmo assim, com alta probabilidade, terminam em tempo polinomial com um resultado correto.

Vale notar que:

Esquemas criptográficos baseados em problemas NP-difíceis ou NP-completos fazem uso de outra subclasse particular de problemas, seja para alcançar maior eficiência, seja para permitir a criação de um "alçapão".

Vejamos agora alguns limitantes interessantes para o SVP e CVP

- Vimos alguns problemas NP em reticulados que estão intimamente relacionados com a obtenção de distâncias mínimas...
- Dessa forma, vamos introduzir alguns limitantes de distância mínima que irão auxiliar o andamento da nossa análise.

Teorema de Minkowski

Seja $\mathfrak{L} \subset \mathbb{R}^n$ um reticulado de dimensão n e $S \subset \mathbb{R}^n$ um conjunto convexo e simétrico.

$$Vol(S) > 2^n \det(\mathfrak{L})$$

Teorema de Hermite

Seja $\mathfrak{L} \subset \mathbb{R}^n$ um reticulado de dimensão n e $v \in \mathfrak{L}$

$$||v|| \leq \sqrt{n} \det(\mathfrak{L})^{\frac{1}{n}}$$

Heurística Gaussiana

Seja $B_R(a)$ a bola de raio R centrada em a. Seu volume é dado por $\frac{\pi^{(n/2)} R^n}{\Gamma(1+(n/2))}$*. Assim, é plausível supor que o número de pontos do reticulado dentro de $B_R(0)$ é aproximadamente $\frac{Vol(B_R(0))}{Vol(\mathfrak{L})}$.
A distância mínima esperada pela heurística Gaussiana é

$$\sigma(\mathfrak{L}) = \sqrt{\frac{n}{2\pi e}} \det(\mathfrak{L})^{\frac{1}{n}}$$

O que implica que

$$(1-\xi)\sigma(\mathfrak{L}) \leq ||V_{shortest}|| \leq (1+\xi)\sigma(\mathfrak{L})$$

**Para valores grandes de n, podemos escrever* $Vol(B_R(a))^{\frac{1}{n}} \simeq \sqrt{\frac{2\pi e}{n}} R$

"Usando uma base boa para resolver o apprCVP"

- Se um reticulado $\mathfrak{L} \subset \mathbb{R}^n$ tem um base $v_1, v_2, ...v_n$ que consiste de vetores que são ortogonais entre si ($v_i v_j = 0 \ \ \forall i \neq j$), considera-se fácil resolver o CVP e o SVP.
- Vamos introduzir a seguir uma forma de mensurar o "grau de ortogonalidade" de uma base do reticulado.

Proporção de Hadamard

$$\mathfrak{H}(v_1, v_2, ..., v_n) = \left(\frac{\det(\mathfrak{L})}{||v_1||.||v_2||...||v_n||}\right)^{\frac{1}{n}}$$

- "Bases boas tem a proporção de Hadamard próxima de 1"
- "Bases ruins tem a proporção de Hadamard próxima de 0"

O algoritmo de Babai - "O vértice mais próximo"

Seja $\mathfrak{L} \subset \mathbb{R}^n$ um reticulado com base $v_1, v_2, ...v_n$ e seja $w \in \mathbb{R}^n$ um vetor arbitrário. Se as bases do reticulado são suficientemente ortogonais entre si, o seguinte algoritmo resolve o CVP:

1. Escreva $w = t_1v_1 + t_2v_2 + ... + t_nv_n,\ t_i \in \mathbb{R}, i = 1, 2, ..., n$
2. Faça $a_i = \lfloor t_i \rceil$ para $i = 1, 2, ..., n$
3. Retorne $v = a_1v_1 + a_2v_2 + ... + a_nv_n$

Exemplo de modelo criptográfico baseado em reticulados

- Vamos agora apresentar brevemente um esquema criptográfico baseado em problemas de reticulados
- GGH - Goldreich, Goldwasser, Halevi; 1997.

Para nos ajudar na explicação desse modelo, vamos chamar novamente nossos amigos Alice e Bob! Considere que Alice quer enviar uma mensagem segura para Bob; para isso, deve-se fazer:

1. Alice deve escolher uma base boa $G = \{g_1, ..., g_n\}$ e uma matriz de inteiros $U : \det(U) = \pm 1$. Essas serão a chave privada.
2. Alice computa uma base ruim $B = \{b_1, ..., b_n\}$, fazendo $B = UG$. Alice fornece B como sendo a chave pública do esquema.
3. Bob escolhe um vetor x como texto original e um vetor aleatório r.
4. Usando a chave pública fornecida por Alice, Bob computa o texto cifrado $c = x_1 b_1 + x_2 b_2 + ... + x_n b_n + r$ e o envia para Alice.
5. Alice usa o algoritmo de Babai para computar o vetor $v \in \mathfrak{L}$ mais próximo de c (CVP).
6. Por fim, Alice recupera x fazendo vW^{-1}

O LWE é um problema computacional que serve como a fundação de novos algoritmos criptográficos designados para nos proteger da criptoanalise quântica.

LWE

Dado que um polinomial geral tem formato
$a(x) = a_0 + a_1x + ...a_nx^n$.
Sendo s(x) um segredo, a(x) um polinomial público e e(x) um polinomial pequeno, $t(x) = (a(x).s(x)) + e(x)$ deve ser seguro.

RLWE

- Especializa-se em anéis polinomiais em campos finitos
- NP-difícil e possui chaves mais curtas do que os LWEs normais.

RLWE

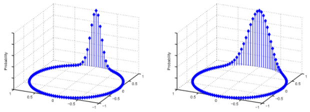


Figure 5: Representação de um anel gaussiano

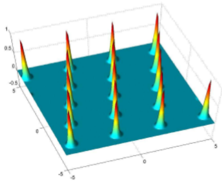


Figure 6: Resultado final

*Referência [4]

Existem agora vários sistemas criptográficos que resolvem esses problemas de aproximar SVP e CVP para reticulados.
Nessa parte vamos descrever o algoritmo LLL que resolve esses problemas com um fator de C^n, onde C é uma constante pequena e n é a dimensão do reticulado.

LLL

- Resolve para pequenas dimensões
- Mas não para grandes dimensões
- Portanto a segurança em sistemas de reticulados dependem da inabilidade do LLL (e outros algoritmos) em resolver de modo eficiente esses problemas

Bases para a redução LLL

Dado uma base $\{v_1, v_2, ..., v_n\}$ para um reticulado $\mathfrak{L}$, o nosso objetivo seria achar uma base boa.

Desigualdade de Hadamard

$$det(\mathfrak{L}) = Vol(F) \leq ||v_1|| ||v_2|| ... ||v_n||$$

Em que F é o domínio fundamental de $\mathfrak{L}$ e a desigualdade se aproxima da igualdade quando a base se aproxima de ser puramente ortogonal.

Bases para a redução LLL

Teorema de Gram-Schmidt

$$v_i^* = v_i - \sum_{j=1}^{i-1} \mu_{i,j} v_j^*, \text{ onde } \mu_{i.j} = \frac{v_i v_j^*}{||v_j^*||^2} \text{ para } 1 \leq j \leq i-1$$

Usando o teorema podemos chegar em uma coleção de vetores $B^* = \{v_1^*, v_2^*, ..., v_n^*\}$ que são puramente ortogonais, mas que não necessariamente são uma base para $\mathfrak{L}$.

Decorrência de Gram-Schmidt

$$det(\mathfrak{L}) = \prod_{i=1}^{n} ||v_i^*||$$

Bases reduzidas em LLL

Seja B uma base do reticulado $\mathfrak{L}$ e B^* a base associada Gram-Schmidt. Essa base B é dita reduzida se ela satisfaz essas duas condições:

- $|\mu_{i,j}| = \frac{|v_i . v_j^*|}{||v_j^*||^2} \leq \frac{1}{2}$, para todo $1 \leq j < i \leq n$. (**Tamanho**)
- $||v_i^*||^2 \geq (\frac{3}{4} - \mu_{i,i-1}^2)||v_{i-1}^*||^2$, para todo $1 < i \leq n$. (**Lovasz**)

Algoritmo de redução

Input a basis $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ for a lattice L
Set $k = 2$
Set $\mathbf{v}_1^* = \mathbf{v}_1$
Loop while $k \leq n$
    Loop $j = 1, 2, 3, \ldots, k-1$
        Set $\mathbf{v}_k = \mathbf{v}_k - \lfloor \mu_{k,j} \rceil \mathbf{v}_j^*$ [Size Reduction]
    End j Loop
    If $\|\mathbf{v}_k^*\|^2 \geq \left(\frac{3}{4} - \mu_{k,k-1}^2\right) \|\mathbf{v}_{k-1}^*\|^2$ [Lovász Condition]
        Set $k = k + 1$
    Else
        Swap $\mathbf{v}_{k-1}$ and $\mathbf{v}_k$ [Swap Step]
        Set $k = \max(k-1, 2)$
    End If
End k Loop
Return LLL reduced basis $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$

Figure 7: Algoritmo LLL

Solucionando CVP com LLL

Aplica-se o algoritmo LLL para atingir uma base boa e depois aplica-se um método como o Babai[1] para decriptar o mensagem. O LLL não tem dificuldades em quebrar sistemas com reticulados de pequena dimensão. Na pratica, sistemas seguros requerem reticulados de dimensão 500-1000, que levam a chaves de tamanhos as vezes impraticáveis.

[1] Normalmente, com o método de Babai resolver por plano mais próximo atinge um melhor resultado do que por vértice mais próximo.

Referências

1. J. Hoffstein, J. Pipher, J.H. Silverman, *An Introduction to Mathematical Cryptography*, 2nd edn. Undergraduate Texts in Mathematics (Springer-Verlag New York, 2014)

2. C. Paar, J. Pelzl, *Understanding Cryptography: A Textbook for Students and Practitioners* (Springer-Verlag Berlin Heidelberg, 2009)

3. D. Micciancio, S. Goldwasser, *Complexity of Lattice Problems: A Cryptographic Perspective*. The Kluwer International Series in Engineering and Computer Science, 671 (Kluwer Academic, Boston, 2002)

4. O. Regev,*On Lattices, Learning with Errors, Random Linear Codes, and Cryptography*, Department of Computer Science, Tel-Aviv University, Tel-Aviv, Israel