UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE CIÊNCIAS EXATAS E DA NATUREZA
DEPARTAMENTO DE MATEMÁTICA

Teoria dos Números e Criptografia

Relatório Final

José Laudelino de Menezes Neto

Antônio de Andrade e Silva
Orientador

agosto de 2006

Sumário

Capítulo 1

Introdução

Visando dar continuidade ao nosso estudo sobre criptografia, neste trabalho iremos nos aprofundar na análise do Logaritmo Discreto e do Problema da Mochila, estudando os cripto-sistemas relacionados a estas teorias.

Um outro assunto a ser tratado é o algoritmo AKS, que informa se um determinado número inteiro é primo ou composto, pois critérios de primalidade são de grande importância para criptografia, como, por exemplo, no cripto-sistema RSA.

Iniciaremos o estudo do cripto-sistema com chave pública que utiliza uma seqüência recursiva linear de ordem 3 sobre o Corpo Finito GF_q ou o anel dos inteiros módulo n, $\mathbb{Z}_n$.

Tendo em vista de que os cripto-sistemas são processos algorítmicos, iniciaremos este trabalho, fazendo um breve estudo de quando um algoritmo é de tempo polinomial ou não.

Capítulo 2

Objetivo

Estudar a complexidade e o tempo dos cálculos computacionais necessários para realização de uma determinada operação. Revisar conceitos relacionados a anéis de polinômios, os quais serão utilizados no estudo do Logaritmo Discreto. Entender a idéia básica do algoritmo AKS e os conceitos por trás do mesmo.

Relacionado ao problema do logaritmo discreto e o problema da mochila, construiremos alguns cripto-sistemas de chave-pública que são baseados na dificuldade de resolvê-los.

O Algoritmo AKS utiliza vários resultados da Teoria dos Números, que serão revistos, se necessário, visando suas aplicações neste algoritmo.

Estudar seqüências recursivas lineares de ordem 3 sobre o corpo finito GF_q ou o anel $\mathbb{Z}_n$ para, em seguida, examinar o cripto-sistema que utiliza estas estruturas.

O estudo desses sistemas, Logaritmo Discreto, Problema da Mochila, Algoritmo AKS etc, irá requerer o uso de várias ferramentas matemáticas tais como: *Teoria dos Números, Corpos Finitos, Anéis de Polinômio* e *Problemas de Primalidade*. Estes assuntos, juntamente com os sistemas vistos, servirão de base ao aluno para que este possa iniciar seus estudos na pós-graduação em Matemática Pura, Matemática Aplicada, Engenharia Elétrica, Ciência da Computação ou qualquer outra área que utilize os conceitos aqui estudados.

Capítulo 3

Complexidade dos Cálculos

Estimar o número de operações binárias é de grande utilidade para verificarmos se um deteminado algoritmo é de tempo Polinomial ou não, isto é, verificarmos o custo de um algoritmo.

Começaremos com um simples problema aritmético, a adição de dois inteiros binários. Por exemplo,

$$\begin{array}{rr} & {\scriptstyle 1}\,{\scriptstyle 1}\,{\scriptstyle 1}\,{\scriptstyle 1}\,{\scriptstyle 1}\,{\scriptstyle 1}\,{\scriptstyle 0}\,{\scriptstyle 0} \\ & 1111000 \\ + & 0011110 \\ \hline & 10010110 \end{array}$$

Suponha que os números têm ambos k dígitos binários, se um dos inteiros tem menos dígitos do que o outro, então, preenchemos o que falta com zeros a esquerda, como foi feito no exemplo, para fazer com que tenham a mesma quantidade de dígitos.

Analisemos o que está por trás desta adição. Basicamente, devemos repetir as seguintes etapas k vezes:

1. Observamos o dígito binário de cima e o de baixo e também se há algum "vai um" sobre o dígito binário de cima.
2. Se ambos os dígitos são 0 e não há "vai um", então escrevemos 0 como resultado e continuamos.
3. Se (a) ambos os dígitos são 0 e há "vai um", ou (b) um dos dígitos é 0, o outro é 1 e não há "vai um", então, escrevemos 1 como resultados e continuamos.
4. Se (a) um dos dígitos é 0, o outro é 1 e há "vai um", ou (b) ambos os dígitos são 1 e não há "vai um", então, escrevemos 0 como resultado, vai um na próxima coluna e continuamos.
5. Se ambos os dígitos são 1 e há "vai um", então, escrevemos 1 como resultado, vai um na próxima coluna e continuamos.

Todo este procedimento é chamado operação binária. Fazer a adição de dois números com k dígitos binários requer k operações binárias. Cálculos mais complicados podem ser reduzidos em operações binárias. A quantidade de tempo necessária para um computador fazer um cálculo é essencialmente proporcional ao número de

operações binárias e a constante de proporcionalidade depende da configuração do computador. Nestas nossas estimativas não contamos com as operações do sistema, passos lógicos, armazenamento de dados na memória etc.

Vejamos o processo de multiplicação de um inteiro binário com k dígitos e outro com ℓ dígitos.

$$\begin{array}{r} 11101 \\ 1101 \\ \hline 11101 \\ 111010 \\ 11101 \\ \hline 101111001 \end{array}$$

Suponhamos que utilizamos este procedimento já conhecido para multiplicar um inteiro n com k dígitos binários por um inteiro m com ℓ dígitos binários. Obteremos no máximo ℓ linhas (uma linha é diminuida para cada dígito 0 em m), onde cada linha é uma cópia de n transladada para esquerda a uma certa distância, isto é, com zeros escritos no final. Suponhamos que existam ℓ' linhas, $\ell' \leq \ell$. Como queremos quebrar todos os cálculos em operações binárias, não podemos somar todas as linhas simultaneamente. Logo, movemos da 2.ª linha até a linha de posição ℓ', somando cada nova linha com a soma parcial das anteriores. Em cada estágio, notemos quantas vezes transladamos para esquerda o inteiro n para escrevermos a nova linha. Escrevemos no resultado os dígitos binários mais a direita da soma parcial e, então, somamos com n o inteiro formado com o restante da soma parcial – como explicado acima, isto levará k operações binárias. No exemplo acima, 11101×1101, após somarmos as duas primeiras linhas, obtemos 10010001, escrevemos os três últimos dígitos 001 e somamos o resto 10010 com $n = 11101$. Finalmente, fazemos a soma $10010+11101 = 101111$ e acrescentarmos 001 para obtermos o resultado 1011110001, a soma de $\ell' = 3$ linhas.

Esta descrição nos mostra que o processo de multiplicação pode ser quebrado em $\ell' - 1$ somas, cada uma levando k operações binárias. Como $\ell' - 1 \leq \ell' \leq \ell$, temos a cota superior

$$\text{Tempo(multiplicar um inteiro com } k \text{ dígitos por outro com } \ell \text{ dígitos)} < k\ell,$$

ou seja, o tempo para multiplicar um inteiro com k dígitos binários por outro com ℓ dígitos binários é menor do que k vezes ℓ operações binárias. Em particular,

$$\text{Tempo(elevar ao quadrado um inteiro com } k \text{ dígitos)} < k^2.$$

Definiremos uma nova notação para facilitar quando escrevermos as estimativas. Sejam $f, g : \mathbb{N} \longrightarrow \mathbb{R}_+$. Dizemos que

$$f(n) = O(g(n)) \text{ ou } f = O(g)$$

se existe uma constante K tal que

$$f(n) < Kg(n).$$

Neste caso, dizemos que g é uma cota superior de f. De modo geral, f e g podem ser funções de várias variáveis e estaremos interessados em analisar os casos em que n é suficientemente grande, ou seja, $n \to \infty$.

Com esta notação reescrevemos

$$\text{Tempo(multiplicar um inteiro com } k \text{ dígitos por outro com } \ell \text{ dígitos)} = O(k\ell)$$

e

$$\text{Tempo(elevar ao quadrado um inteiro com } k \text{ dígitos)} = O(k^2).$$

Sabemos que o número de dígitos binários de um inteiro n é dado por

$$k = \lfloor \log_2 n \rfloor + 1 \leq \frac{\ln n}{\ln 2}, \text{ onde } \lfloor \cdot \rfloor \text{ é a função maior inteiro.}$$

Assim,

$$k = O(\ln n).$$

Sejam $m, n \in \mathbb{Z}$, logo

$$\text{Tempo(multiplicar } m \text{ por } n) = O(\ln m \cdot \ln n).$$

Dizemos que um algoritmo para realizar determinada operação, envolvendo inteiros de no máximo k dígitos binários, é de tempo (custo) polinomial, se existe $d \in \mathbb{Z}$ tal que o número de operações binárias requeridas no algoritmo é $O(k^d)$.

Capítulo 4

Polinômios

Neste capítulo faremos uma breve revisão sobre anéis de polinômios. Salvo menção explícita em contrário, a palavra ANEL siginifica anel comutativo com identidade.

Sejam R anel e R^{Seq} o conjunto de todas as sequências formais

$$f = (a_i)_{i \in \mathbb{Z}_+},$$

onde os $a_i \in R$ e $a_i \neq 0$ somente para um número finito de índices.

Dados $f = (a_i), g = (b_j) \in R^{\text{Seq}}$. Dizemos que f é igual a g, em símbolos $f = g$, se, e somente se, $a_i = b_i, \forall i \in \mathbb{Z}$.

O conjunto R^{Seq} pode ser equipado com uma estrutura de anel do seguinte modo:

$$f + g = (a_i + b_i) \text{ e } f * g = (c_k),$$

onde $c_k = \sum_{i+j=k} a_i b_j = \sum_{j=0}^{k} a_{k-j} b_j$. Note que c_k é finito, pois $0 \leq i, j \leq k$.

Estas operações estão bem definidas em R^{Seq}. De fato, sejam $m, n \in \mathbb{Z}_+$ tais que

$$a_i = 0, \forall i > m, \text{ e } b_j = 0, \forall j > n.$$

Escolhendo $k \geq \max\{n, m\}$ temos que

$$a_i + b_i = 0, \forall i > k.$$

Logo, $f + g \in R^{\text{Seq}}$. Escolhendo $k = m + n + 1$, temos que

$$c_k = \sum_{j=0}^{n} a_{k-j} b_j + \sum_{j=n+1}^{k} a_{k-j} b_j = 0,$$

pois

$$k - n = m + n + 1 - n = m + 1 > n \text{ e } n + t > n, t = 1, 2, \ldots, m + 1.$$

Logo, $f * g \in R^{\text{Seq}}$.

Teorema 4.1 *$(R^{Seq}, +, *)$ é um anel.*

A função $\varphi : R \longrightarrow R^{\text{Seq}}$ definida por

$$\varphi(a) = (a, 0, 0, \ldots)$$

é um homomorfismo de aneis injetor. Logo,

$$R \cong \text{Im}\varphi \leq R^{\text{Seq}}.$$

Assim, podemos identificar os elementos de R com as sequências $(a, 0, 0, \ldots), a \in R$. Note que $x = (0, 1, 0, 0, \ldots) \in R^{\text{Seq}}$ e que

$$\begin{aligned} x^2 &= (0, 0, 1, 0, 0, \ldots) \\ &\vdots \\ x^n &= (0, 0, \ldots, 0, 1, 0, 0, \ldots). \end{aligned}$$

Assim,

$$\begin{aligned} ax &= (0, a, 0, 0, \ldots) \\ &\vdots \\ ax^n &= (0, 0, \ldots, 0, a, 0, 0, \ldots), \forall a \in R \text{ e } n \in \mathbb{Z}_+. \end{aligned}$$

Dado $f \in R^{\text{Seq}}$, digamos $f = (a_0, a_1, \ldots, a_n, 0, 0, \ldots)$. Então,

$$f = a_0 + a_1 x + \ldots + a_n x^n.$$

Denotamos R^{Seq} por $R[x] = \langle R \cup \{x\} \rangle$ e chamamos $R[X]$ de anel dos polinômios na variável x. Os elementos de R são chamados de polinômios constantes.

Seja $f = \sum_{i=0}^{n} a_i x^i \in R[x]$. Se $a_n \neq 0$, dizemos que f tem grau n e denotamos por $\partial(f) = n$. Neste caso, o termo $a_n x^n$ é chamado de termo lider e a_n de coeficiente lider. Em particular, se $a_n = 1$, dizemos que f é um polinômio mônico (unitário).

Exemplo 4.2 *1. $\mathbb{Z}[x], \mathbb{Q}[x], \mathbb{R}[x]$ e $\mathbb{C}[x]$ são anéis de polinômios.*

2. $f(x) = 3x^2 + 5x + 1$ é um polinômio em $\mathbb{Z}[x]$.

3. $g(x) = x^5 + \sqrt{4}x^2$ é um polinômio mônico em $\mathbb{R}[x]$.

Proposição 4.3 *Seja R um Domínio de Integridade (DI). Então,*

1. $R[x]$ é um DI.

2. $\partial(f + g) \leq max\{\partial(f), \partial(g)\}, \forall f, g \in R[x] - 0$ com $f + g \neq 0$.

3. $\partial(fg) = \partial(f) + \partial(g), \forall f, g \in R[x] - 0$.

4. $U(R[x]) = U(R)$.

Seja R um anel. Dizemos que um anel S é uma extensão de R se R for um subanel de S. Seja s um elemento arbitrário de S. Então, para cada $f = \sum a_i x^i \in R[x]$, podemos definir $f(s) \in S$ por

$$f(s) = a_0 + a_1 s + \ldots + a_n s^n,$$

onde a adição e a multiplicação usada na definição de $f(s)$ são as de S e não as de $R[x]$. A função

$$\begin{aligned} E_s : R[x] &\longrightarrow S \\ f &\longrightarrow f(s) \end{aligned}$$

é um homo de anéis. Note que $\mathrm{Im}E_s = \{f(s); f \in R[x]\}$ é um subanel de S e será denotado por $\mathrm{Im}E_s = R[s] = \langle R \cup \{s\}\rangle$.

Sejam S uma extensão de R, $\alpha \in S$ e $f \in R[x]$. Dizemos que α é uma raiz de f se $f(\alpha) = 0$.

4.1 Algoritmo da Divisão

O algoritmo da divisão considerado em $\mathbb{Z}$ pode ser extendido para um anel de polinômios $R[x]$.

Teorema 4.4 (Algoritmo da Divisão) *Sejam $f, g \in R[x]$ com o coeficiente líder de g uma unidade. Então, existem únicos $q, r \in R[x]$ tais que*

$$f = qg + r, r = 0 \text{ ou } \partial(r) < \partial(g).$$

Observação 4.5 *1. Se g é mônico ou R é um corpo, então não é necessário supor que o coeficiente líder de g seja uma unidade.*

2. Se $R = \mathbb{Z}_4$, então

$$x = (2x^2 + x + 2)(2x + 1) + 2 \text{ e } x = (2x^2 + x)(2x + 1).$$

Em $\mathbb{Z}_4$, $2 \notin U(\mathbb{Z}_4)$. Não garantindo a unicidade de q e r.

Sejam $f, g \in R[x]$ com $g \neq 0$. Dizemos que g divide f, em símbolos $g \mid f$, se existe $h \in R[x]$ tal que $f = hg$. Caso contrário, dizemos que g não divide f, em símbolos $g \nmid f$. Dizemos que f e g são associados se $f \mid g$ e $g \mid f$, isto é, quando existe $u \in U(R)$ tal que $f = ug$.

4.2 Máximo Divisor Comum (MDC)

Sejam $f_1, \ldots, f_k \in R[x]^*$. Um elemento $d \in R[x]$ é um máximo divisor comum (MDC) de $f_1, \ldots, f_k$ se as seguintes condições são satisfeitas:

1. $d \mid f_i, i = 1, 2, \ldots, k$
2. $d' \mid f_i, i = 1, 2, \ldots, k \Rightarrow d' \mid d$.

Notação: $d = \mathrm{mdc}(f_1, \ldots, f_k)$.

Teorema 4.6 *Sejam $d = mdc(f_1, \ldots, f_k)$ em $R[x]$ e $a \in U(R)$. Então, $ad = mdc(f_1, \ldots, f_k)$.*

Teorema 4.7 (Algoritmo Euclidiano) *Existe um algoritmo para calcular o MDC.*

Demonstração: Sejam K um corpo e $f, g \in K[x]^*$. Então, aplicando sucessivamente o AD, temos que

$$\begin{aligned}
f &= q_0 g + r_1, r_1 = 0 \text{ ou } \partial(r_1) < \partial(g) \\
g &= q_1 r_1 + r_2, r_2 = 0 \text{ ou } \partial(r_2) < \partial(r_1) \\
&\vdots \\
r_n &= q_{n+1} r_{n+1} + r_{n+2}, r_{n+2} = 0 \text{ ou } \partial(r_{n+2}) < \partial(r_{n+1})
\end{aligned}$$

Como $\partial(g) > \partial(r_1) > \ldots > \partial(r_{n+2}) \geq 0$ temos que existe um primeiro índice k tal que $r_{k+2} = 0$. Afirmação: $d = r_{k+1} = \text{mdc}(f, g)$. De fato, é claro que $d = \text{mdc}(r_{k-2}, r_{k-1})$ e, portanto, $d = \text{mdc}(f, g)$. ■

De forma indutiva, podemos generalizar o resultado para $f_1, \ldots, f_k \in K[x]^*$.

Corolário 4.8 *Sejam K um corpo, $f, g \in K[x]^*$ e $d = mdc(f, g)$. Então, existem $r, s \in K[x]$ tais que*

$$d = rf + sg.$$

Demonstração: Basta aplicar o Teorema 5.7 de trás para frente. ■

Sejam K um corpo e $f_1, \ldots, f_k \in K[x]^*$. Dizemos que $f_1, \ldots, f_k$ são relativamente primos se $\text{mdc}(f_1, \ldots, f_k) = 1$. Neste caso, existem $g_1, \ldots, g_k \in K[x]$ tais que

$$g_1 f_1 + \ldots + g_k f_k = 1.$$

4.3 Polinômios Irredutíveis e Fatoração Única

Sejam R um DI e $f \in R[x]^*$. Dizemos que f é irredutível sobre R se as seguintes condições são satisfeitas:

1. $f \notin U(R)$
2. Se $f = gh$, então $g = a \in U(R)$ ou $h = b \in U(R)$.

Caso contrário, dizemos que f é redutível.

Exemplo 4.9 *1. Todo polinômio linear $f = a_0 + a_1 x$, $a_1 \neq 0$, é irredutível sobre K, onde K é um corpo. Isto significa que todo polinômio redutível tem grau pelo menos dois em um corpo qualquer.*

2. O polinômio $f = x^2 - 2$ é irredutível sobre $\mathbb{Q}$.

3. O polinômio $f = x^4 + 4$ é redutível em $\mathbb{R}[x]$.

Solução do Exemplo 5.9 item 2: Suponhamos, por absurdo, que f seja redutível sobre $\mathbb{Q}$. Logo, $f = (a + bx)(c + dx), a, b, c, d \in \mathbb{Q}, b \neq 0, d \neq 0$. Assim,

$$ac = -2, bd = 1 \text{ e } ad + bc = 0.$$

Então,

$$abd + b^2 c = 0 \text{ e } a^2 bd + b^2 ac = 0 \Rightarrow a^2 - 2b^2 = 0 \Rightarrow (\frac{a}{b})^2 = 2 \Rightarrow \sqrt{2} \in \mathbb{Q}.$$

O que é absurdo. Note que, embora f seja irredutível sobre $\mathbb{Q}$, f é redutível sobre $\mathbb{R}$, pois $f = (x - \sqrt{2})(x + \sqrt{2})$.

Solução do Exemplo 5.9 item 3: Basta notarmos que,

$$x^4 + 4 = x^4 + 4x^2 + 4 - 4x^2 = (x^2 + 2)^2 - (2x)^2 = (x^2 - 2x + 2)(x^2 + 2x + 2).$$

Teorema 4.10 *Sejam K corpo e $p \in K[x]^*$. Então, as seguintes condições são equivalentes:*

1. *p é irredutível sobre K*
2. *$M = \langle p \rangle$ é um ideal maximal de $K[x]$*
3. *$\dfrac{K[x]}{M}$ é um corpo.*

Exemplo 4.11 *É fácil verificar que x^2+1 é irredutíve em $\mathbb{R}[x]$. Então, $\dfrac{\mathbb{R}[x]}{M} \cong \mathbb{C}$, onde $M = \langle x^2+1 \rangle$.*

Solução: Dado $f \in \mathbb{R}[x]$, existem únicos $q, r \in \mathbb{R}[x]$ tais que $f = q(x^2+1) + r, r = 0$ ou $\partial(r) < 2$. Logo, $r = a + bx$, com $a, b \in \mathbb{R}$. Assim,

$$\begin{aligned} \frac{\mathbb{R}[x]}{M} &= \{f + M; f \in \mathbb{R}[x]\} \\ &= \{a + bx + M; a, b \in \mathbb{R}\}. \end{aligned}$$

Vamos definir $\varphi : \dfrac{\mathbb{R}[x]}{M} \longrightarrow \mathbb{C}$ por $\varphi(a + bx + M) = a + bi, i^2 = -1$. É fácil verificar que φ é um homomorfismo de corpos bijetor.

Teorema 4.12 (Fatoração Única) *Seja K um corpo. Então,*

1. *Todo polinômio em $K[x]^*$ pode ser escrito como um produto finito de fatores irredutíveis.*
2. *Se $f = up_1 \dots p_m = u'q_1 \dots q_n$ são duas fatorações de f com $u, u' \in K^*$, então $m = n$ e existe uma permutação σ em $\{1, \dots, m\}$ tal que p_i e $q_{\sigma(i)}$ sejam associados.*

Demonstração: (1) Vamos usar indução em $\partial(f) = k$.

(i) Se $k = 0$, nada há para ser provado, pois $f = a \in K^*$.

(ii) Suponhamos que o resultado seja válido para todo polinômio de grau menor que k. Se f é irredutível, acabou. Vamos supor que f seja redutível. Então, $f = gh$, onde $1 \leq \partial(g), \partial(h) < k$. Assim, pela hipótese de indução,

$$g = ap_1 \dots p_r \text{ e } h = bp_{r+1} \dots p_m,$$

onde $a, b \in K^*$ e os p_i são polinômios irredutíveis sobre K. Portanto, $f = up_1 \dots p_m$, onde $u = ab \in K^*$.

(2) Vamos usar indução sobre m.

(i) Se $m = 0$, então, $u = u'q_1 \dots q_n$. Assim, se $n > 0$, então $q_j \mid u \Rightarrow q_j \mid 1$, para algum $1 \leq j \leq n$. Logo, $q_j \in K^*$, o que é impossível. Portanto, $n = 0$ se $m = 0$.

(ii) Suponhamos que o resultado seja válido para todo ℓ com $0 < \ell < m$. Como $p_m \mid u'q_1 \dots q_n$ e $p_m \nmid u'$ temos que $n > 0$ e $p_m \mid p_j$, para algum $1 \leq j \leq n$. Reenumerando, se necessário, podemos supor $p_m \mid p_n$. Logo, q_n é associado a p_m e $p_m = vq_n, v \in K^*$. Assim,

$$uvp_1 \dots p_{m-1} = u'q_1 \dots q_{n-1}.$$

Pela hipotese de indução, temos que $m-1 = n-1$ e p_i é associado a $q_{\sigma(i)}$, para alguma permutação σ de $\{1, \dots, m-1\}$. Logo, $m = n$ e p_i é associado a $q_{\tilde{\sigma}(i)}$, onde $\tilde{\sigma}$ é uma permutação de $\{1, \dots, m\}$ definida por

$$\tilde{\sigma}(i) = \begin{cases} \sigma(i) & , i = 1, \dots, m-1 \\ n & , i = m \end{cases}$$ ■

Capítulo 5

Logaritmo Discreto

Nos dois cripto-sistemas a seguir a associação de cada símbolo do nosso alfabeto $\mathbb{A} = \{A, B, C, \ldots, X, Y, Z\}$ com um inteiro será dada por

$$A = 0$$

$$B = 1$$

$$\vdots$$

$$Z = 25$$

e consideramos fixado um corpo finito $\mathbb{Z}_q$, o qual é conhecido publicamente e o número de elementos de $\mathbb{Z}_q$ é maior ou igual ao número de símbolos do alfabeto escolhido.

Um usuário que saiba resolver o problema do logaritmo discreto em $\mathbb{Z}_q$ pode quebrar os cripto-sistemas que serão descritos a seguir.

5.1 Cripto-sistema Massey-Omura

Cada usuário u_i escolhe um inteiro qualquer e_i tal que $0 < e_i \leq q-1$ e

$$mdc(e_i, q-1) = 1.$$

Calcula o inverso de e_i em $\mathbb{Z}_{q-1}$, $(e_i)^{-1} = d_i$, isto é

$$e_i d_i \equiv 1(\mod q-1) \Rightarrow e_i d_i = h(q-1)+1, h \in \mathbb{Z}$$

e não divulga estes dois números.

Um usuário u_i que deseja enviar uma mensagem $\mathbf{o}$ para o usuário u_j, primero calcula $\mathbf{o}^{e_i}$ e envia a mensagem criptografada para u_j. Ao receber a mensagem, u_j calcula $\mathbf{o}^{e_i e_j}$ e manda a mensagem de volta a u_i. Assim, u_i aplica a chave de decodificação d_i obtendo apenas $\mathbf{o}^{e_j}$, porque

$$\mathbf{o}^{(e_i e_j)d_i} = \mathbf{o}^{e_j(e_i d_i)} = \mathbf{o}^{e_j[h(q-1)+1]} = (\mathbf{o}^{[h(q-1)+1]})^{e_j} = \mathbf{o}^{e_j}. \tag{5.1}$$

Desta forma, u_i manda o resultado obtido para u_j e este calcula $\mathbf{o}^{e_j d_j}$ para obter a mensagem original $\mathbf{o}$.

Observação 5.1 *Note que estamos trabalhando em $\mathbb{Z}_q$, pois, neste caso, dado $a \in \mathbb{Z}_q$ temos*

$$a^{|\mathbb{Z}_q|} = a^{q-1} = 1,$$

por este motivo vale a equação (6.1).

Além de ser um cripto-sistema muito trabalhoso, é muito importante ter um bom sistema de assinatura. Caso contrário, qualquer usuário u_k, que não pode saber a mensagem $\mathbf{o}$, pode se passar por u_j, retornando para u_i $\mathbf{o}^{e_i e_k}$ e u_i, sem saber que é um intruso, poderá dar continuidade ao processo, fazendo com que u_k leia a mensagem $\mathbf{o}$.

5.2 Cripto-sistema ElGamal

Fixamos um elemento $g \in \mathbb{Z}_q^*$. Neste cripto-sistema $\mathbb{Z}_q$ não precisa ser um corpo e o elemento fixado g não precisa ser um gerador de $\mathbb{Z}_q^*$.

Cada usuário u_i escolhe um inteiro $a_i \in \mathbb{Z}_q^*$, que será a chave secreta de decodificação. A chave pública de codificação será o elemenro $g^{a_i} \in \mathbb{Z}_q$.

Um usuário u_j que deseja enviar uma mensagem $\mathbf{o}$ para o usuário u_i, escolhe um inteiro qualquer k e envia o seguinte par de elementos

$$(g^k, \mathbf{o}g^{a_i k}) \in \mathbb{Z}_q \times \mathbb{Z}_q.$$

Para decodificar a mensagem, o usuário u_i calcula

$$d \equiv (g^k)^{q-1-a_i} (\mod q)$$

e, em seguida,

$$d(\mathbf{o}g^{a_i k}) \equiv \mathbf{o} (\mod q)$$

recuperando a mensagem $\mathbf{o}$.

Observação 5.2

$$g^{k(q-1-a_i)} \equiv g^{k(q-1)-a_i k} \equiv g^{k(q-1)} g^{-a_i k} \equiv (g^{(q-1)})^k g^{-a_i k} \equiv 1^k g^{-a_i k} \equiv g^{-a_i k} (\mod q)$$

Capítulo 6

Problema da Mochila

Relembraremos a seguir o caso especial do problema da mochila, já abordado em nosso último relatório, e, em seguida, faremos a construção do cripto-sistema Merkle-Hellman.

Sabemos que sempre é possível resolver o problema da mochila quando os pesos são arrajandos na ordem crescente, tendo a seguinte propriedade

$$\sum_{j=0}^{i-1} s_j < s_i, i = 1, 2, \ldots, k-1,$$

ou seja, cada s_i é maior do que a soma dos s_i anteriores. Para resolver tal problema, utilizamos o seguinte algoritmo, o qual chamaremos de Algoritmo Crescente.

1. Ajuste W igual a V e tome $j = k$.
2. Começando com ϵ_{j-1} e diminuindo o índice de ϵ, escolha todos os $\epsilon_i = 0$ até chegar ao primeiro i – chamando-o de i_0 – tal que $s_{i_0} \leq W$. Ajuste $\epsilon_{i_0} = 1$.
3. Troque W por $W - s_{i_0}$, ajuste $j = i_0$, e, se $W > 0$ volte ao passo 2.
4. Se $W = 0$, então está terminado. Se $W > 0$ e todos os s_i restantes são maiores do que W, então, sabemos que não existe

   $$n = (\epsilon_{k-1}\epsilon_{k-2}\ldots\epsilon_1\epsilon_0)_2$$

   solução do problema.

A solução (se existe) é única.

6.1 Cripto-sistema Merkle-Hellman

Agora, estamos aptos para construirmos o cripto-sistema Merkle-Hellman. Iremos fazer o caso particular em que os símbolos do nosso alfabeto tem equivalência numérica com um inteiro de cinco dígitos binários. A associação de cada símbolo do nosso alfabeto $\mathbb{A} = \{A, B, C, \ldots, X, Y, Z\}$ com um inteiro será dada por

$$A = 0 = (00000)_2$$

$$B = 1 = (00001)_2$$

$$\vdots$$

$$Z = 25 = (11001)_2.$$

Em seguida, cada usuário u_j escolhe um vetor peso

$$\vec{v} = (v_0, v_1, v_2, v_3, v_4),$$

um inteiro m maior do que $\sum_{i=0}^{4} v_i$ e um inteiro a tal que $mdc(a, m) = 1$ e $0 < a < m$. Feito isso, calcula-se o inteiro b que é o inverso de a módulo m, isto é,

$$ab \equiv 1(\mod m),$$

e também o vetor $\vec{w} = (w_0, w_1, w_2, w_3, w_4)$ definido por

$$w_i \equiv av_i(\mod m).$$

Assim, a chave de codificação pública é

$$k_{c,j} = (w_0, \ldots, w_4) = \vec{w}$$

e a chave de decodificação, que é mantida secreta, é

$$k_{d,j} = (b, m).$$

Um usuário u_i que deseja enviar uma mensagem do texto-original $\mathbf{o} = (\epsilon_4\epsilon_3\epsilon_2\epsilon_1\epsilon_0)$ para o usuário u_j com chave de codificação $k_{c,j} = (w_0, w_1, w_2, w_3, w_4)$ calcula

$$\mathbf{c} = \sum_{i=0}^{4} \epsilon_i w_i$$

e transmite o inteiro encontrado $\mathbf{c}$.

O usuário u_j ao receber o inteiro $\mathbf{c}$ calcula

$$b\mathbf{c} \equiv b\sum_{i=0}^{4} \epsilon_i w_i \equiv \sum_{i=0}^{4} \epsilon_i bav_i \equiv \sum_{i=0}^{4} \epsilon_i v_i \equiv V(\mod m).$$

Agora, o usuário u_j utiliza o Algoritmo Crescente no inteiro V com o vetor peso $\vec{v}$ para recuperar a mensagem do texto-original

$$\mathbf{o} = (\epsilon_4\epsilon_3\epsilon_2\epsilon_1\epsilon_0).$$

A dificuldade para um usuário não autorizado desvendar a mensagem codificada sem saber a chave de decodificação é que mascaramos um Problema da Mochila de fácil resolução em um outro de difícil resolução quando multiplicamos v_i por a módulo m. Assim, o Algoritmo Crescente não pode ser utilizado neste Problema da Mochila "mascarado".

Exemplo 6.1 *Enviar e receber a mensagem "FIM" utilizando o cripto-sistema Merkle-Hellman.*

Solução: Escolhemos o vetor peso $\vec{v}= (2, 5, 8, 17, 35)$, $m = 71$ e $a = 21$, então $b = 44$. Desta forma temos a chave de codificação dada por

$$k_c = \vec{w}= (42, 34, 26, 2, 25)$$

e a chave de decodificação

$$k_d = (44, 71).$$

Para enviar a mensagem "FIM" o correspondente deve utilizar a chave de codificação para calcular

$$F = 5 = (00101)_2 \to 1 \cdot 42 + 0 \cdot 34 + 1 \cdot 26 + 0 \cdot 2 + 0 \cdot 25 = 68$$

$$I = 8 = (01000)_2 \to 0 \cdot 42 + 0 \cdot 34 + 0 \cdot 26 + 1 \cdot 2 + 0 \cdot 25 = 2$$

$$M = 12 = (01100)_2 \to 0 \cdot 42 + 0 \cdot 34 + 1 \cdot 26 + 1 \cdot 2 + 0 \cdot 25 = 28$$

e nos mandar a mensagem codificada $68, 2, 28$. Logo, quando recebermos a mensagem codificada, primeiro multiplicamos cada número por 44 módulo 71 para obtermos

$$44 \cdot 68 \equiv 10(\mod 71)$$

$$44 \cdot 2 \equiv 17(\mod 71)$$

$$44 \cdot 28 \equiv 25(\mod 71)$$

e utilizamos o Algoritmo Crescente para $V = 10$, $V = 17$ e $V = 25$ com o vetor peso $\vec{v}= (2, 5, 8, 17, 35)$ para recuperarmos o texto-original

$$(00101)_2 = 5 \to F$$

$$(01000)_2 = 8 \to I$$

$$(01100)_2 = 12 \to M.$$

Capítulo 7

Algoritmo AKS

Existem testes de primalidade que são probabilísticos, que fornecem com precisão de até 99,9% se um determinado número é primo ou não. Um exemplo de teste probabilístico é o Teste de Monte Carlo. Os testes deterministicos de probalibilidade nos diz com total certeza quando um número é primo. O Crivo de Erastótenes é um teste deterministico, entretanto é de tempo exponencial. O algoritmo AKS, que foi idealizado por três indianos, Manindra **A**grawal, Neeraj **K**ayal e Nitin **S**axena, vislumbrou muitos matemáticos por ser o primeiro teste deterministico de tempo polinomial e por sua simplicidade ao resolver este problema matemático que perdurava durante muitos anos.

O algoritmo AKS é baseado em um resultado de identidade para números primos, o qual é uma generalização do Pequeno Teorema de Fermat. Antes de enunciarmos esse resultado, vejamos quatro Lemas.

Lema 7.1 *Sejam $p, k \in \mathbb{N}$ com $p \geq 2$ e $k < p$, então $\binom{p}{k}$ é inteiro*

Demonstração: Provaremos este resultado utilizando indução sobre p. Para $p = 2$ e $k = 1 < p$ temos

$$\binom{2}{1} = \frac{2!}{1!(2-1)!} = \frac{2}{1 \cdot 1} = 2 \in \mathbb{N}.$$

Suponhamos que o resultado seja válido para $p > 2$. Como

$$\binom{p}{k} + \binom{p}{k+1} = \binom{p+1}{k+1}$$

e, pela hipótese de indução,

$$\binom{p}{k}, \binom{p}{k+1} \in \mathbb{N} \Rightarrow \binom{p+1}{k+1} \in \mathbb{N}. \ \blacksquare$$

Lema 7.2 *Sejam $p \in \mathbb{Z}$ um primo e k um inteiro qualquer com $1 \leq k < p$. Então,*

$$\binom{p}{k} \equiv 0(\mod p).$$

Lema 7.3 *Se p é primo, então*

$$(x+y)^p \equiv x^p + y^p(\mod p), \forall x, y \in \mathbb{Z}.$$

Demonstração: Observamos que

$$(x+y)^p = x^p + \binom{p}{p-1}x^{p-1}y + \ldots + \binom{p}{1}xy^{p-1} + y^p$$

e, pelo Lema 8.2, temos

$$(x+y)^p \equiv x^p + y^p (\mod p). \blacksquare$$

Lema 7.4 (Pequeno Teorema de Fermat) *Sejam $p \in \mathbb{Z}$ um primo e $a \in \mathbb{Z}$ tal que $mdc(a,p) = 1$. Então,*

$$a^p \equiv a (\mod p).$$

Demonstração: Sabemos que $\left|\frac{\mathbb{Z}}{p\mathbb{Z}}\right| = p-1$ e, portanto,

$$a^{p-1} \equiv 1 (\mod p) \Rightarrow a^p \equiv a (\mod p). \blacksquare$$

Teorema 7.5 *Seja $a \in \mathbb{Z}$, $n \in \mathbb{N}$, $n \geq 2$ e $mdc(a,1) = 1$. Então, n é primo se, somente se,*

$$(x+a)^n \equiv x^n + a (\mod n).$$

Demonstração: ($\Rightarrow$) Consequência imediata dos quatro Lemas anteriores.
($\Leftarrow$) Consultar referência [6] ou [7]. $\blacksquare$

A seguir, a primeira versão do algoritmo AKS,

```
        Input:  integer n > 1

 1.     if ( n is of the form a^b, b > 1 ) output COMPOSITE;
 2.     r = 2;
 3.     while (r < n) {
 4.        if ( gcd(n, r) ≠ 1 ) output COMPOSITE;
 5.        if ( r is prime )
 6.           let q be the largest prime factor of r − 1;
 7.           if (q ≥ 4√r log n) and (n^((r−1)/q) ≢ 1 ( mod r) )
 8.             break;
 9.        r ← r + 1;
10.     }
11.     for a = 1 to 2√r log n
12.        if ( (x − a)^n ≢ (x^n − a) ( mod x^r − 1, n) ) output COMPOSITE;
13.     output PRIME;
```

Teorema 7.6 *O algoritmo acima retorna PRIME se, e somente se, n é primo.*

Vejamos uma breve análise do algoritmo,

- As linhas de 1 a 10 funcionam como um filtro para descartar muitos valores que não influenciam na decisão de primalidade.
- Na linha 1, deve-se utilizar um algoritmo que detecte potências perfeitas.

- Na linha 4, verifica-se se o $mdc(n, r)$ é diferente de um. Se n é primo, então $mdc(n, r) = 1, \forall r \leq n$.
- Na linha 5, utiliza-se um teste não probabilístico, por exemplo o Crivo de Erastótenes, para verificar se r é primo. Na linha 6, sendo r primo, determina-se o maior fator primo de $r - 1$, no caso q. Se q satisfaz a condição da linha 7, então se passa à linha 11. Caso contrário, repete-se o processo da linha 4 para $r + 1$.
- Na linha 12 é utilizado o Teorema 8.5 com uma alteração na congruência, que passa para módulo $x^r - 1$ e, desta forma, o teste é feito para os r últimos termos de $(x - a)^n$.

O Algoritmo AKS tem apenas relevância matemática, pois, na prática, leva mais tempo para retornar primo do que os testes probabilísticos.

Capítulo 8

Sequências Recursivas Lineares de Ordem 3

Uma sequência recursiva linear de ordem 3 em $\mathbb{Z}_q = \frac{\mathbb{Z}}{q\mathbb{Z}}$ é uma sequência $(s_n) \subset \mathbb{Z}_q$ onde existem $a_0, a_1, a_2 \in \mathbb{Z}_q$ tais que

$$s_{n+3} = a_0 s_{n+2} + a_1 s_{n+1} + a_2 s_n, \forall n = 0, 1, 2, \ldots \tag{8.1}$$

Os termos s_0, s_1, s_2 são chamados valores iniciais e a Equação 9.1 é denominada relação de recorrência linear de ordem 3.

O vetor $\vec{s}_n = (s_{n+2}, s_{n+1}, s_n)$ é o n-ésimo vetor de estado e $\vec{s}_0 = (s_2, s_1, s_0)$ o vetor de estado inicial.

Todas as sequências em $\mathbb{Z}_q$ são periódicas. A sequência $(s_n) \subset \mathbb{Z}_q$ é de maior período se existirem $r, n_0 \in \mathbb{N}$ tais que

$$s_{n+r} = s_n, \forall n \geq n_0.$$

Denominamos n_0 de pré-período e r de período da sequência. O menor de todos os possíveis períodos é chamado de menor período.

Proposição 8.1 *Seja (s_n) uma sequência recursiva linear de ordem 3 de maior período em $\mathbb{Z}_q$. Então, todo período é divisível pelo menor período.*

Uma sequência de maior período com menor período com menor período r é chamada periódica se

$$s_{n+r} = s_n, \forall n = 0, 1, 2, \ldots$$

Proposição 8.2 *A sequência (s_n) é periódica se, e somente se, existir um inteiro $r > 0$ tal que $s_{n+r} = s_n$ para todo $n = 0, 1, 2, \ldots$*

Teorema 8.3 *Toda sequência recursiva linear de ordem 3 $(s_n) \subset \mathbb{Z}_q$ é de maior período com menor período $r \leq q^3 - 1$.*

Teorema 8.4 *Se (s_n) é uma sequência recursiva linear em $\mathbb{Z}_q$ satisfazendo a equação (9.1) com $a_0 \neq 0$, então a sequência é periódica.*

Podemos associar a sequência recursiva linear de ordem 3 $(s_n) \subset \mathbb{Z}_q$ satisfazendo a equação (9.1) com uma matriz 3×3 da seguinte forma

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & a_0 \\ 1 & 0 & a_1 \\ 0 & 1 & a_2 \end{bmatrix}.$$

A matriz $\mathbf{A}$ é chamada matriz companheira da sequência recursiva linear de ordem 3.

Teorema 8.5 *Sejam (s_n) uma sequência recursiva linear de ordem 3 em $\mathbb{Z}_q$ satisfazendo a equação (9.1) e $\boldsymbol{A}$ sua matriz companheira. Então, o n-ésimo vetor da sequência satisfaz*

$$\vec{s}_n = \vec{s}_0 \boldsymbol{A}^n, \forall n = 0, 1, 2, \ldots$$

Seja (s_n) uma sequência recursiva linear de ordem 3 em $\mathbb{Z}_q$. Então, o polinômio

$$f = x^3 - a_2x^2 - a_1x - a_0 \in \mathbb{Z}_q[x]$$

é chamado polinômio característico da sequência (s_n). Definimos

$$f^{\perp} = -x^3 f(\frac{1}{x}) = -1 + a_2x + a_1x^2 + a_0x^3 \in \mathbb{Z}_q$$

como sendo o polinômio característico recíproco da sequência (s_n).

Lema 8.6 *Seja $f \in \mathbb{Z}_q[x]$ um polinômio de grau $m \geq 1$ com $f(0) \neq 0$. Então, existe um inteiro positivo $k \leq q^m - 1$ tal que f divide $x^k - 1$.*

Definição 8.7 *Seja $f \in \mathbb{Z}_q[x]$, $f \neq 0$. Se $f(0) \neq 0$, então o menor inteiro positivo k para o qual f divide $x^k - 1$ é chamado a ordem de f e denotada por ord(f). Se $f(0) = 0$, então $f(x) = x^h g(x)$, onde $h \in \mathbb{N}$ e $g \in \mathbb{Z}_q[x]$, com $g(0) \neq 0$, são unicamente determinados, logo ord(f) é dada por ord(g).*

Teorema 8.8 *Seja (s_n) uma sequência recursiva linear de ordem 3 em $\mathbb{Z}_q$ com matriz companheira $\boldsymbol{A}$. Então, a matriz $\boldsymbol{A}$ tem como polinômio minimal o polinômio característico da sequência (s_n), isto é,*

$det(\boldsymbol{I}x - \boldsymbol{A}) = f = x^3 - a_2x^2 - a_1x - a_0 \in \mathbb{Z}_q[x]$, *onde $\boldsymbol{I}$ é a 3×3 matriz identidade.*

Restringiremos o estudo de sequência à sequência recursiva linear homogênea de ordem 3 em $\mathbb{Z}_q$

$$s_k = as_{k-1} - bs_{k-2} + s_{k-3}, a, b \in \mathbb{Z}_q \text{ e } \forall k = 3, 4, 5, \ldots \tag{8.2}$$

com polinômio característico

$$f = x^3 - ax^2 + bx - 1 \in \mathbb{Z}_q[x]$$

e valores iniciais

$$s_0 = 3, s_1 = a \text{ e } s_2 = a^2 - 2b.$$

Denotaremos s_k por $s_k(a, b)$ ou $s_k(f)$ e $\vec{s}$ como $s(a, b)$ ou $s(f)$.

Teorema 8.9 *Sejam (s_k) uma sequência recursiva linear homogênea de ordem 3, f seu polinômio característico e $\alpha_1, \alpha_2, \alpha_3$ suas raízes no corpo de decomposição de f sobre $\mathbb{Z}_q$. Então,*

1. $s_k(a,b) = \alpha_1^k + \alpha_2^k + \alpha_3^k, \forall k = 3, 4, \ldots$
2. $s_{-k} = bs_{-k+1} - as_{-k+2} + s_{-k+3}, \forall k = 3, 4, \ldots$ *é uma sequência recursiva linear homogênea de ordem 3 com polinômio característico $f^{\perp}$.*
3. $s_{-k}(a,b) = \alpha_1^{-k} + \alpha_2^{-k} + \alpha_3^{-k} = s_k(b,a), \forall k = 3, 4, 5, \ldots$

Teorema 8.10 *Sejam (s_n) uma sequência recursiva linear de ordem 3 em $\mathbb{Z}_q$ satisfazendo a equação (9.2). Então,*

$$s_k(s_l(a,b), s_{-l}(a,b)) = s_{kl}(a,b), \forall k, l = 0, 1, 2, \ldots$$

Denotaremos por

$$Q = p^2 + p + 1$$

a ordem (ou período) de um polinômio irredutível sobre $\mathbb{Z}_q$.

Um inteiro positivo k é chamado um líder de classe módulo Q se k é o menor inteiro no conjunto

$$\{lp^i(\mod Q); i = 0, 1, 2 \text{ e } l \in \mathbb{Z}_+\}.$$

Teorema 8.11 *Sejam $f = x^3 - ax^2 + bx - 1$ um polinômio irredutível sobre $\mathbb{Z}_q$ de ordem $Q = p^2 + p + 1$ e $s(f)$ uma sequência associada a f. Consideremos k e k' diferentes líderes de classe módulo Q com $mdc(k, Q) = mdc(k', Q) = 1$. Então,*

$$(s_k, s_{-k}) \neq (s_{k'}, s_{-k'}).$$

Exemplo 8.12 *Seja $(s_n) \subset \mathbb{Z}_3$ uma sequencia recursiva linear de ordem 3 com $s_0 = 1, s_1 = 2$ e $s_2 = 2$ e*

$$s_{n+3} = s_n + s_{n+1} + 2s_{n+2}.$$

Então,

$$\vec{s_0} = (1, 2, 2), \vec{s_n} = (s_n, s_{n+1}, 2s_{n+2})$$

e

$$a_0 = 1, a_1 = 1, a_2 = 2.$$

Por inspeção, temos que

$$s_{n+6} = s_n, \forall n = 0, 1, 2, \ldots$$

A matriz companheira de (s_n) é

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

com polinômio característico $f = x^3 - 2x^2 - x - 1$, o qual, de fato, é o polinômio característco da sequência.

Observamos também que

$$\vec{s_3} = \vec{s_0}\boldsymbol{A}^3 = (1, 0, 0).$$

Portanto, $s_3 = 1, s_4 = 0$ e $s_5 = 0$.

Exemplo 8.13 *Consideremos a seguinte sequência recursiva linear homogênea de ordem 3 em* $\mathbb{Z}_5$

$$s_k = 3s_{k-1} - s_{k-2} + s_{k-3}, \forall k = 3, 4, \ldots$$

com $s_0 = 3, s_1 = 3$ *e* $s_2 = 2$ *e polinômio característico*

$$f = x^3 - 3x^2 + x - 1$$

de período $Q = 31$*. Então, observamos que*

$$s_3 = 1, s_5 = 3, s_{-3} = s_3(1, 3) = 0 \text{ e } s_{-5} = s_5(1, 3) = 1.$$

Notemos também que

$$s_{15} = s_3(s_5, s_{-5}) = s_5(s_3, s_{-3}) = 1$$

e

$$s_{-15} = s_{-3}(s_5, s_{-5}) = s_{-5}(s_3, s_{-3}) = 0.$$

8.1 Cripto-sistema GH-PKD

Apresentaremos o sistema de distribuição com chave pública GH-PKD, o qual foi proposto por Gong e Harn. Este sistema utiliza os conceitos de sequências recursivas lineares de ordem 3.

O cripto-sistema GH-PKD tem como parâmetros públicos um número primo p, um polinômio

$$f = x^3 - ax^2 + bx - 1$$

irredutível sobre $\mathbb{Z}_p$ com período $Q = p^2 + p + 1$ e uma sequência recursiva linear homogênea de ordem 3 associada a este polinômio $s_k(a, b)$.

Neste sistema, cada usuário u_i escolhe aleatoriamente um inteiro r_i tal que

$$0 < r_i < Q \text{ e } mdc(r_i, Q) = 1.$$

A seguir, u_i calcula (s_{r_i}, s_{-r_i}) e torna público sua chave de codificação

$$k_{c,i} = (s_{r_i}, s_{-r_i})$$

e mantem secreta a chave de decodificação $k_{d,i} = r_i$.

Um outro usuário u_j terá suas chaves

$$k_{c,j} = (s_{r_j}, s_{-r_j}) \text{ e } k_{d,j} = r_j.$$

Exemplo 8.14 *Sejam* $p = 5, f = x^3 - 3x + x - 1$ *polinômio irredutível sobre* $\mathbb{Z}_5$ *de período* $Q = 5^2 + 5 + 1 = 31$ *e* $r_i = 3$ *e* $r_j = 5$ *as chaves de decodificação dos usuários* u_i *e* u_j *respectivamente. Então, pelo Exemplo 9.13,*

$$k_{c,i} = (s_3, s_{-3}) = (1, 0) \text{ e } k_{c,j} = (s_5, s_{-5}) = (3, 1).$$

Como

$$s_3(s_5, s_{-5}) = s_5(s_3, s_{-3}) = s_{15} = 1$$

e

$$s_{-3}(s_5, s_{-5}) = s_{-5}(s_3, s_{-3}) = s_{-15} = 0,$$

então $k_{ij} = (1, 0)$.

A correspondência entre o alfabeto $\mathbb{A}$ *e os números inteiros é a mesma exibida no início do Capítulo 6. Assim, para o usuário* u_i *enviar para* u_j *o texto-original*

$$\text{“}OI\text{”}$$

com correspondência

$$\begin{array}{cc} O & I \\ \updownarrow & \updownarrow \\ 14 & 8 \end{array}$$

o usuário u_i *calcula* $k_{ij} = (1, 0)$ *e depois*

$$\boldsymbol{c}_1 = (s_{r_i}, s_{-r_i}) = (1, 0) \text{ e } \boldsymbol{c}_2 = k_{ij} - \boldsymbol{u} = (1, 0) + (14, 8) = (15, 8).$$

Logo, o texto-cifrado é o par

$$\boldsymbol{c} = (c_1, c_2).$$

Portanto, o usuário u_i *envia para* u_j *a mensagem*

$$\text{“}BAPI\text{”}.$$

Para decodificar o texto-cifrado com correspondência numérica

$$\begin{array}{cccc} B & A & P & I \\ \updownarrow & \updownarrow & \updownarrow & \updownarrow \\ 1 & 0 & 15 & 8 \end{array}$$

o usuário u_j *recupera* $\boldsymbol{c}_1 = (1, 0)$ *e* $\boldsymbol{c}_2 = (15, 8)$ *e, em seguida, calcula* $s_5(1, 0) = 1$ *e* $s_{-5}(1, 0) = 0$ *para obter* $k_{ij} = (1, 0)$*, calcular*

$$\boldsymbol{u} = \boldsymbol{c}_2 - k_{ij} = (15, 8) - (1, 0) = (14, 8)$$

e obter o texto-orignal

$$\begin{array}{cc} O & I \\ \updownarrow & \updownarrow \\ 14 & 8 \end{array}$$

Capítulo 9

Conclusão

Nesses 2 anos de iniciação científica nos defrontamos com vários cripto-sistemas e pudemos constatar que, de fato, o mais eficiente e seguro é o RSA. O Merkle-Hellman, na época de sua divulgação, foi creditado como o mais seguro e eficaz, entretanto Shamir, o S do RSA, desenvolveu um algoritmo que quebra o Merkle-Hellman. O sistema de distribuição de chave pública GH-PKD, descrito neste trabalho, não é muito seguro, entretanto há uma combinação dele com o RSA mais sofisticada e mais segura. Para finalizar este assunto de cripto-sistemas, reafirmamos que a Aritmética Modular é essencial para o aprendizado de criptografia, pois em todos os cripto-sistemas estudados foi necessário a utilização de tal teoria.

Quanto ao Algoritmo AKS, constatamos que, apesar de toda a sua simplicidade, um estudo mais aprofundado do mesmo requer mais tempo e outros pré-requisitos.

Referências Bibliográficas

[1] Silva, A. A. *Números, Relações e Criptografia*. Departamento de Matemática - UFPB.

[2] Sidki, S. *Introdução à Teoria dos Números*. 10.º Colóquio Brasileiro de Matemática, IMPA, 1975.

[3] Koblitz, *A Course in Number Theory and Cryptography*, 2nd ed., Springer, 1994.

[4] Koblitz, *Algebraic Aspects of Cryptography*, Volume 3. Springer, 1999.

[5] Cameron, P. J. *Notes on Cryptography.*
http://www.maths.qmw.ac.uk/%7Epjc/notes/crypt.pdf

[6] Agrawal, Kayal e Saxena, *PRIMES in P*, Preprint, 2002.

[7] Coutinho, S. C. *Primalidade em tempo polinomial: Uma introdução ao Algoritmo AKS*, SBM, 2004.

[8] Cavalcanti, A. C. F. *Cripto-sistemas com chave pública baseado em extensões cúbicas*, Departamento de Matemática - UFPB, 2002.