UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE MATEMÁTICA
CURSO DE LICENCIATURA EM MATEMÁTICA

FELIPE DE ALMEIDA DUARTE

# A ÁLGEBRA NA CRIPTOGRAFIA

TRABALHO DE CONCLUSÃO DE CURSO

**CORNÉLIO PROCÓPIO**

**2015**

FELIPE DE ALMEIDA DUARTE

# A ÁLGEBRA NA CRIPTOGRAFIA

Trabalho de Conclusão de Curso apresentado ao Departamento Acadêmico de Matemática da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do título de "Licenciado em Matemática ".

Orientador: Prof. Dr. Josimar da Silva Rocha

**CORNÉLIO PROCÓPIO**

**2015**

# TERMO DE APROVAÇÃO

FELIPE DE ALMEIDA DUARTE

## A ÁLGEBRA NA CRIPTOGRAFIA

Trabalho de Conclusão de Curso apresentada ao Curso de Licenciatura em Matemática da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do grau de "Licenciado em Matemática em Matemática" – Área de Concentração: .

**BANCA EXAMINADORA**

_____

Profº. Drº. Josimar da Silva Rocha
Universidade Tecnológica Federal do Paraná

_____

Profº. Drº. Anderson Paião dos Santos
Universidade Tecnológica Federal do Paraná

_____

Profº. Me. Tiago Henrique dos Reis
Universidade Tecnológica Federal do Paraná

Cornélio procópio, 03 de Junho de 2015.

A Folha de Aprovação assinada encontra-se na Coordenação do Curso.

Dedico este trabalho primeiramente a Deus, por seus inacreditáveis feitos em minha vida, e a todos os alunos de graduação em Matemática que por ventura, se sentirem motivados com tal obra.

## AGRADECIMENTOS

É pouco o espaço que tenho para descrever a minha gratidão porém, deixo aqui meus agradecimentos especiais aos grandes mestres que acompanharam minha jornada durante a gradução e que logo poderei chamá-los de companheiros de trabalho, eles que me mostraram os prazeres e as dificuldades de tal profissão e me ajudaram a chegar até aqui, os meus eternos agradecimentos.

Gostaria de agradecer ainda à minha família, que mesmo na distância me apoiaram e me motivaram a prosseguir com o curso; em especial aos meus pais que, mesmo não compreendendo toda a minha fala, me ouviam pacientemente, a todo sacrifício que fizeram por mim os meus agradecimentos.

Não poderia encerrar o texto sem agradecer àqueles que se fizeram minha família durante o curso (em especial a Família 14 BIS, no qual sempre estarão no meu coração), os meus amigos. Estes, que por muitas vezes tiveram que aguentar minhas reclamações, que me faziam ficar com o pé no chão quando estava desesperado e que dividiram minhas alegrias, tristezas, festas, almoços, entre outros; meus agradecimentos de coração.

*“Às vezes, são as pessoas que ninguém espera nada que fazem as coisas que ninguém consegue imaginar.”*
(O Jogo da Imitação, 2014)

# RESUMO

DUARTE, Felipe de Almeida. A ÁLGEBRA NA CRIPTOGRAFIA. 61 f. Trabalho de Conclusão de Curso – Curso de Licenciatura em Matemática, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2015.

A Álgebra é um ramo da matemática que se destaca devido às suas aplicabilidades que auxiliam na compreensão e no aperfeiçoamento do mundo moderno. Uma das áreas de grande atuação é a Criptografia, que ganhou destaque após o surgimento do computador e do desenvolvimento de métodos que possibilitem troca de informações de forma cada vez mais segura e rápida. O presente trabalho procura destacar como a Álgebra auxiliou e auxilia os métodos criptográficos de forma a aperfeiçoá-los e, ao mesmo tempo, oferecer maior segurança para tais métodos.

**Palavras-chave:** álgebra, criptografia, conjuntos algébricos

## ABSTRACT

DUARTE, Felipe de Almeida. ALGEBRA IN ENCRYPTIONS. 61 f. Trabalho de Conclusão de Curso – Curso de Licenciatura em Matemática, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2015.

Algebra is a branch of mathematics that stands out because of its applicability to aid in the understanding and improvement of the modern world. One area of great acting is encryption, which gained prominence after the emergence of the computer and the development of methods that enables information exchange increasingly safe and fast ways. This paper seeks to highlight how the Algebra helped and helps cryptographic methods to improve them and at the same time providing greater security for such methods.



**Keywords:** algebra, cryptography, algebric sets

## LISTA DE FIGURAS

## LISTA DE TABELAS

**SUMÁRIO**

# 1 INTRODUÇÃO

A criptografia surgiu com a necessidade de trocar mensagens secretas entre aliados durante as guerras sem que os inimigos soubessem sobre o conteúdo contido nas mesmas. Com o avanço da tecnologia e das aplicações matemáticas, visando a facilidade de comunicação e a segurança de informação, muitas cifras foram desenvolvidas para confundir os inimigos. A partir da matemática pode-se compreender como alguns desses métodos funcionam, destacando fórmulas computacionalmente complexas, o que auxilia na segurança e complexidade de cálculos de novos sistemas criptográficos.

Analisar tais processos criptográficos do ponto de vista matemático, permite avaliar o quão complexo podem parecer os cálculos sob algumas estruturas algébricas. As estruturas algébricas fornecem maior segurança nos processos criptográficos devido às operações definidas sobre as mesmas; tais operações podem ser fáceis nos conjuntos numéricos usuais mas, em determinados conjuntos, estas operações se demonstram mais complexas e, às vezes, não estão claramente definidas, o que ajuda na complexidade dos cálculos das cifras.

Este trabalho tem como objetivo discutir como elementos da álgebra moderna contribuem para a segurança nos processos criptográficos. Para isso, se faz necessário definir os principais elementos algébricos presentes nas cifras aqui apresentadas; definir criptografia, apresentar algumas cifras e mostrar alguns exemplos; e por fim, discutir como os elementos algébricos utilizados em tais processos estabelecem maior segurança para os mesmos.

Este trabalho está organizado da seguinte forma: No Capítulo 2 será apresentado definições de elementos algébricos que serão necessários para a compreensão dos processos criptográficos citados neste trabalho. No Capítulo 3, será definido criptografia e apresentado algumas cifras, consideradas por este autor, mais interessantes de serem estudadas. No Capítulo 4 será apresentado a força que os elementos algébricos fornecem nos métodos apresentados, destacando algumas considerações do autor e; no Capítulo 5 se faz presente as considerações finais e conclusões do autor.

# 2 UM POUCO DE ÁLGEBRA MODERNA

Determinadas estruturas algébricas ganharam força nos processos criptográficos devido às suas propriedades algébricas e à sua praticidade nas aplicações em determinados algoritmos computacionais. Na cifra Rijndael (atual criptografia AES), por exemplo, utiliza-se uma estrutura algébrica que, devido às suas propriedades, é determinada como um corpo finito de $2^8$ elementos e é representado pelo símbolo $GF(2^8)$ (GF = *Galois Field* ou Corpo de Galois, em homenagem ao matemático francês Évarist Galois (1811-1832)). Neste capítulo será apresentado algumas definições e teoremas fundamentais sobre algumas estruturas algébricas e suas propriedades. Maiores informações sobre definições, teoremas e demonstrações presentes neste capítulo encontram-se nas obras de Lidl e Niederreiter (1997) e de Domingues e Iezzi (2003).

## 2.1 DIVISIBILIDADE E NÚMEROS PRIMOS

Esta seção se dedica exclusivamente aos números primos devido à importância que os mesmos apresentam na maioria das cifras. Maiores informações sobre o conteúdo desta seção podem ser encontradas na obra de Coutinho (2003).

**Definição 1.** Diz-se que o número inteiro $a$ é **divisor** do número inteiro $b$ (ou divide $b$), ou que o número $b$ é divisível por $a$ se é possível encontrar $c \in \mathbb{Z}$ tal que $b = ac$. Neste caso, pode-se dizer também que $b$ é **múltiplo** de $a$. Para indicar que $a$ divide $b$, denotaremos por $a|b$.

A relação entre os elementos de $\mathbb{Z}$, definida por $x|y$, goza das seguintes proriedades:

1. $a|a$;
2. Se $a|b$ e $b|c$, então $a|c$.
3. Se $a|b$ e $c|d$, então $ac|bd$.
4. Se $a|b$ e $a|c$, então $a|(b+c)$.

5. Se $a|b$, então para todo $m \in \mathbb{Z}$,tem-se que $a|mb$.

6. Se $a|b$ e $a|c$, então $a|(bx+cy)$ para quaisquer $x, y \in \mathbb{Z}$.

As demonstrações das propriedades acima podem ser encontradas na obra de Coelho e Milies (2006).

**Definição 2.** (Números Primos) Um número inteiro $p > 1$ é um **número primo** se os únicos divisores positivos de $p$ são 1 e $p$.

**Exemplo 1.** Os números 2, 3, 5, 7, 11, 13, 17, 19, 23 e 29 são os dez primeiros números primos, em ordem crescente. Outros exemplos são os números: 311, 509, 978491 e 27795571.

**Definição 3.** (Números Compostos) Um número inteiro $q$ é dito **composto** se possuir mais de dois divisores positivos.

**Exemplo 2.** Números pares em geral (com exceção do número 2) são números compostos: $2n = 2 \cdot n$; o número $15 = 5 \cdot 3$; $2310 = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11$.

**Teorema 1.** (Teorema Fundamental da Aritmética) Todo inteiro maior do que 1 pode ser representado de maneira única (a menos da ordem) como um produto de fatores primos.

*Demonstração.* Se $n$ é primo não há nada a ser demonstrado. Supondo que $n$ é composto. Seja $p_1$ $(p_1 > 1)$ o menor dos divisores de $n$. $p_1$ é primo pois, caso contrário, existiria $p$, $1 < p < p_1$ com $p|n$, contradizendo a escolha de $p_1$. Logo, $n = p_1 n_1$. Se $n_1$ for primo a demonstração se completa. Caso contrário, toma-se $p_2$ como o menor fator de $n_1$. Pelo argumento anterior, $p_2$ é primo e temos que $n = p_1 p_2 n_2$.

Repetindo este procedimento, obtemos uma sequência decrescente de inteiros positivos $n_1,\ n_2\ ,...,n_r$. Como todos eles são inteiros maiores do que 1, este processo deve terminar. Como os primos na sequência $p_1,\ p_2\ ,...,\ p_k$ não são, necessariamente, distintos, $n$ terá, em geral, a forma:

$$n = p_1^{a_1} p_2^{a_2} ... p_k^{a_k}.$$

Para mostrarmos a unicidade usamos indução em $n$. Para $n = 2$ a afirmação é verdadeira. Assumimos, então, que ela se verifica para todos os inteiros maiores do que 1 e menores do que $n$. Vamos provar que ela também é verdadeira para $n$. Se $n$ é primo, não há o que mostrar. Vamos supor, então, que $n$ seja composto e que tenha duas fatorações, isto é,

$$n = p_1 p_2 ... p_s = q_1 q_2 ... q_r$$

Vamos provar que $s = r$ e que cada $p_i$ é igual a algum $q_j$. Como $p_1$ divide o produto $q_1q_2...q_r$ ele divide pelo menos um dos fatores $q_j$. Sem perda de generalidade podemos supor que $p_1|q_1$. Como ambos são primos, isto implica que $p_1 = q_1$. Logo $n/p_1 = p_2...p_s = q_2...q_r$. Como $1 < n/p_1 < n$, a hipótese de indução nos diz que as duas fatorações são idênticas, isto é, $s = r$ e, a menos da ordem, as fatorações $p_1p_2...p_s$ e $q_1q_2...q_s$ são iguais. □

**Definição 4.** (Máximo Divisor Comum - MDC) Sejam $a$ e $b$ dois números inteiros com $b \neq 0$. Um número inteiro positivo $d$ é o **máximo divisor comum** entre $a$ e $b$ se:

**i)** $d|a$ e $d|b$;

**ii)** se $d'$ é um inteiro tal que $d'|a$ e $d'|b$, então $d'|d$.

**Exemplo 3.** 2 é o máximo divisor comum entre 8 e 6, ou seja, $mdc(8,6) = 2$, assim como, $mdc(301,54) = 1$, $mdc(18,6) = 6$, e $mdc(68,36) = 4$.

**Definição 5.** (Números Coprimos) Dois números inteiros $a$ e $b$ são **coprimos** quando ambos não compartilham de fatores primos comuns, isto é, quando $mdc(a,b) = 1$

**Exemplo 4.** No exemplo 3, os números 301 e 54 são coprimos, bem como todo par de números primos.

**Teorema 2.** (Algoritmo da Divisão) Dados dois inteiros $a$ e $b$, $b > 0$, existe um único par de inteiros $q$ e $r$ tais que

$$a = qb + r, \; com \; 0 \leq r < b \; (r = 0 \Leftrightarrow b|a)$$

*Demonstração.* Para a demonstração, verifique na obra de Santos (2014). □

**Teorema 3.** (Algoritmo de Euclides) Sejam $r_0 = a$ e $r_1 = b$ inteiros não-negativos com $b \neq 0$. Se o algoritmo da divisão for aplicado sucessivamente para obter

$$r_j = q_{j+1}r_{j+1} + r_{j+2}, \; 0 \leq r_{j+2} < r_{j+1}$$

para $j = 0,1,2,...,n-1$ e $r_{n+1} = 0$ então $(a,b) = r_n$, o último resto não-nulo.

*Demonstração.* Vamos, inicialmente, aplicar o teorema anterior para dividir $r_0 = a$ por $r_1 = b$ obtendo $r_0 = q_1r_1 + r_2$, em seguida dividimos $r_1$ por $r_2$ obtendo $r_1 = q_2r_2 + r_3$ e assim, sucessivamente, até a obtenção do resto resto $r_{n+1} = 0$. Como, a cada passo o resto é sempre menor do que o anterior, e estamos lidando com números inteiros positivos, é claro que após um número finito de aplicações do Teorema 2, teremos resto nulo.

Temos, pois, aseguinte sequência de equações:

$$\begin{array}{rcll} r_0 & = & q_1 r_1 + r_2 & 0 < r_2 < r_1 \\ r_1 & = & q_2 r_2 + r_3 & 0 < r_3 < r_2 \\ \vdots & & \vdots & \vdots \\ r_{n-2} & = & q_{n-1} r_{n-1} + r_n & 0 < r_n < r_{n-1} \\ r_{n-1} & = & q_n r_n + 0 & \end{array}$$

A última destas equações nos diz, que o máximo divisor comum de $r_n$ e $r_{n-1}$ é $r_n$ (vide Santos (2014)). A penúltima, que este número é igual ao $mdc(r_{n-1}, r_{n-2})$ e, proseguindo dessa maneira teremos:

$$r_n = mdc(r_{n-1}, r_n) = mdc(r_{n-2}, r_{n-1}) = ... = mdc(r_1, r_2) = mdc(r_0, r_1) = mdc(a, b).$$

Portanto o máximo divisor comum de $a$ e $b$ é o último resto não nulo da sequência de divisões descrtitas. □

**Exemplo 5.** Considerando os números inteiros $-387$ e $-99$ têm-se que, $-387 = (-99) \cdot 4 + 9$. Assim, o par (4,9) é único e, têm-se que $0 \leq 9 < |-99|$.

**Teorema 4.** Sejam $a$ e $b$ números inteiros positivos tais que $a = bq + r$, com $q, r \in \mathbb{Z}$. Então $mdc\ (a,b) = mdc\ (b,r)$.

*Demonstração.* Supondo que

$$d_1 = mdc(a,b) \text{ e } d_2 = mdc(b,r),$$

Precisa-se mostrar que $d_1 = d_2$. Têm-se que:

$$d_1|a \text{ e } d_1|b \Rightarrow d_1|(a - bq) \text{ e } d_1|b \Rightarrow d_1|d_2 \tag{1}$$

$$d_2|b \text{ e } d_2|r \Rightarrow d_2|b \text{ e } d_2|(bq + r) \Rightarrow d_2|d_1 \tag{2}$$

Por (1) e (2) têm-se que $d_1 = d_2$. □

**Exemplo 6.** Do exemplo 5 têm-se que o $mdc(-387, -99) = 9 = mdc(-99, 9) = 9$.

**Algoritmo da Fatoração:** O algoritmo deve buscar um número inteiro que divida algum $n \in \mathbb{Z}$, começando de 2 e avançando até $\sqrt{n}$. Se $n$ for composto, encontra-se o menor fator de $n$ por este método, e este fator é necessariamente um número primo. Se nenhum dos números pesquisados é fator de $n$, isto significa que $n$ é primo.

**Exemplo 7.** Seja o número 527, pelo algoritmo da fatoração, deve-se verificar se existe um inteiro entre 2 e $\sqrt{527} \cong 22,9564$ que divida 527. Têm-se, por este processo, que o número 527 não é primo pois $13|527$. O leitor poderá verificar que tal algoritmo pode ser reduzido para uma verificação entre os números primos presentes entre 2 e $\sqrt{n}$.

**Números de Mersenne:** Martin Mersenne foi um frade e matemático amador do século XVII, que foi correspondente de muitos matemáticos famosos da época. Os números da forma $2^n - 1$ receberam o nome de Números de Mersenne devido à uma afirmação que o mesmo fez e que teve grande repercussão. Segundo ele, os números da forma $M(n) = 2^n - 1$ seriam primos quando $n = 2,3,5,7,13,17,19,31,67,127,257$ e composto para os demais primos menores que 257. A primeira coisa que deve-se observar é que os expoentes de tal lista são números primos. De fato, se $n = rs$, então,

$$M(n) = 2^n - 1 = 2^{rs} - 1 = (2^r - 1)(2^{r(s-1)} + 2^{r(s-2)} + ... + 2^r + 1)$$

Portanto, se $r$ divide $n$, então $M(r)$ divide $M(n)$. O seguinte fato a se observar é que a recíproca é falsa, ou seja, se $n$ for primo não significa que $M(n)$ tem que ser primo, como por exemplo, $M(11)$. Porém alguns erros foram encontrados em tal lista, como por exemplo, $M(61)$, $M(89)$ e $M(107)$ são primos e inclui os compostos $M(67)$ e $M(257)$.

**Números de Fermat:** Em uma carta de 1640 a um matemático amador, Fermat enumerou os números da forma $F(n) = 2^{2^n} + 1$, para os valores inteiros de $n$ entre 0 e 6. Tais números são: 3, 5, 17, 257, 65537, 4294967297 e 18446744073709551617. Em seguida conjecturou que todos os números dessa forma são primos. Porém, assim como os Números de Mersenne, existe uma falha que Euler descobriu ao encontrar o menor fator primo de $F(5)$. Na verdade, até hoje não se descobriu nenhum número $F(n)$ primo com $n \geq 5$.

**Fórmulas Fatoriais:** Definindo uma função semelhante ao fatorial só que esta só aceita valores $p$ primos de entrada e, a multiplicação se dá apenas com os números primos menores ou iguais a $p$. Tal função será denotada por $p^{\#}$. Observe que se $q < p$ são primos sucessivos temos $p^{\#} = q^{\#}p$. Assim, podemos notar que tal função gera muitos números primos ao se somar 1 ao resultado de $p^{\#}$:

| $p$ | $p^{\#}$ | $p^{\#} + 1$ |
|---|---|---|
| 2 | 2 | 3 |
| 3 | 6 | 7 |
| 5 | 30 | 31 |

Porém, pode-se notar que se $p = 13$, o resultado de $p^{\#} + 1$ será um número composto. Pode-se demonstrar que $p^{\#} + 1$ não tem nenhum fator primo menor ou igual a $p$ por redução ao absurdo (para detalhes de tal demonstração consulte a obra de Coutinho, p. 59, 2003).

**Crivo de Eratóstenes:** É um dos métodos mais antigos para encontrar números primos. Ele funciona como uma "peneira" e trabalha com números inteiros de 2 a $n$ onde, em cada etapa do método, elimina-se os múltiplos do menor número primo que ainda não foi utilizado no processo. Por exemplo, tomando os números de 2 a 50; na primeira rodada, eliminamos todos os múltiplos de 2, na segunda todos os de 3, na terceira todos os de 5, e assim sucessivamente até não sobrar múltiplos de nenhum dos números restantes da lista.

| | 2 | 3 | ~~4~~ | 5 | ~~6~~ | 7 | ~~8~~ | ~~9~~ | ~~10~~ |
|---|---|---|---|---|---|---|---|---|---|
| 11 | ~~12~~ | 13 | ~~14~~ | ~~15~~ | ~~16~~ | 17 | ~~18~~ | 19 | ~~20~~ |
| ~~21~~ | ~~22~~ | 23 | ~~24~~ | ~~25~~ | ~~26~~ | ~~27~~ | ~~28~~ | 29 | ~~30~~ |
| 31 | ~~32~~ | ~~33~~ | ~~34~~ | ~~35~~ | ~~36~~ | 37 | ~~38~~ | ~~39~~ | ~~40~~ |
| 41 | ~~42~~ | 43 | ~~44~~ | ~~45~~ | ~~46~~ | 47 | ~~48~~ | ~~49~~ | ~~50~~ |

**Tabela 1: Crivo de Eratóstenes - De 2 a 50**

## 2.2 ARITMÉTICA MODULAR

A aritmética modular está ligada aos processos de cifragem e decifragem de muitos algoritmos como no RSA, no AES, no ElGamal, entre outros. Nesta seção será apresentada as principais propriedades da aritmética modular.

**Definição 6.** (Conjunto $\mathbb{Z}_m$) $\mathbb{Z}_m$ é o conjunto formado pelos números inteiros $\{0, 1, ..., m-1\}$. Com as operações de adição e multiplicação definidas por $a + b = c$, onde $c$ é o resto da divisão da adição usual dos números inteiros $a$ e $b$ por por $m$ e $a \cdot b = d$, onde $d$ é o resto da divisão do produto usual dos números inteiros $a$ e $b$ por $m$.

**Exemplo 8.** Dado o conjunto das classes de restos $\mathbb{Z}_5 = \{0, 1, 2, 3, 4\}$, onde $3 + 2 = 0$ e $3 \cdot 2 = 1$.

**Definição 7.** (Congruência dos Inteiros) Sejam $a$, $b$ números inteiros e $m$ um inteiro positivo. Diz-se que $a$ é congruente a $b$ módulo $m$ se $m \mid (a - b)$, isto é, se $a - b = mq$ para algum $q$ inteiro. Para indicar que $a$ é congruente a $b$ módulo $m$, usa-se a notação:

$$a \equiv b \ (mod \ m). \tag{3}$$

**Exemplo 9.** Todos os números pares são, dois a dois, congruentes módulo 2, ou seja, $2k \equiv 2q \ (mod \ 2)$ para quaisquer $k, q \in \mathbb{Z}$ não necessariamente distintos.

Na sequência listamos algumas das propriedades básicas de congruência de inteiros cuja demonstração pode ser encontrada em Domingues e Iezzi (2003). Sejam $a, b, c \in \mathbb{Z}$.

**1) Reflexividade:** $a \equiv a \ (mod\ m)$.

**2) Simetria:** Se $a \equiv b \ (mod\ m)$, então $b \equiv a \ (mod\ m)$.

**3) Transitividade:** Se $a \equiv b \ (mod\ m)$ e $b \equiv c \ (mod\ m)$, então $a \equiv c \ (mod\ m)$.

**4)** Se $a, b \in \mathbb{Z}$ com $0 \leq b < m$, então $a \equiv b \ (mod\ m)$ se, e somente se, $b$ é o resto da divisão euclidiana de $a$ por $m$.

**5)** $a \equiv b \ (mod\ m)$ se, e somente se, $a$ e $b$ dão o mesmo resto na divisão euclidiana por $m$.

**6)** $a \equiv b \ (mod\ m)$ se, e somente se, $a \pm c \equiv b \pm c \ (mod\ m)$.

**7)** $a \equiv b \ (mod\ m)$ e $c \equiv d \ (mod\ m)$, então $a + c \equiv b + d \ (mod\ m)$

**8)** Se $a \equiv b \ (mod\ m)$, então $ac \equiv bc \ (mod\ m)$

**9)** Se $a \equiv b \ (mod\ m)$ e $c \equiv d \ (mod\ m)$, então $ac \equiv bd \ (mod\ m)$

**10)** Se $ca \equiv cb \ (mod\ m)$ e $mdc(c, m) = d > 0$, então $a \equiv b \ (mod\ m/d)$

**Definição 8.** Um número inteiro $a$ possui inverso módulo $n$ se existe um inteiro $b$ tal que $a \cdot b \equiv 1 \ (mod\ n)$.

**Teorema 5.** Seja $n$ um número inteiro positivo. Um número inteiro $a$ possui inverso módulo $n$ se, e somente se, $a$ e $n$ forem primos entre si, isto é, quando $mdc(n, a) = 1$.

*Demonstração.* Supondo que $a$ possui inverso módulo $n$. Então, existe um $\alpha \in \mathbb{Z}$ tal que $\alpha a \equiv 1 \ (mod\ n)$. Isto é equivalente a dizer que $\alpha a - 1$ é múltiplo de $n$, isto é, $\alpha a - 1 = nt$, para algum $t \in \mathbb{Z}$. Seja $d = mdc(a, n)$. Então, $d$ divide $a$ e $d$ divide $n$, isto é, $a = da'$, para algum $a' \in \mathbb{Z}$ e $n = dn'$, para algum $n' \in \mathbb{Z}$. Podemos reescrever a igualdade $\alpha a - 1 = nt$ como $d\alpha a' - 1 = dn't$, que é equivalente a $d(\alpha a' - n't) = 1$. Esta última igualdade implica que $d$ divide 1, ou seja, d = 1.

Suponha que $mdc(a, n) = 1$. Utilizando o Método de Divisões Sucessivas, obtemos a igualdade $\alpha a + \beta n = 1$, que pode ser escrita como $\alpha a - 1 = (-\beta)n$. Esta igualdade é equivalente a dizer que $\alpha a \equiv 1 \ (mod\ n)$. Logo $\alpha$ é o inverso multiplicativo de $a$ módulo $n$. □

**Exemplo 10.** Considerando o número 257 e o conjunto $\mathbb{Z}_7$ temos, pelo Algoritmo da Fatoração, que 257 é primo portanto, $mdc(257, 7) = 1$, ou seja, 257 possui inverso em $\mathbb{Z}_7$ que é 3.

## 2.3 GRUPOS, ANÉIS E CORPOS

Nesta seção será apresentada alguns conceitos algébricos sobre a teoria de anéis e corpos. Tais conceitos são importantes para entender algumas terminologias e propriedades destacadas no decorrer deste texto. Para maiores informações sobre o conteúdo deta seção, consulte Domingues e Iezzi (2003).

**Definição 9.** (Grupo) Uma estrutura algébrica formada por um conjunto não vazio $G$ e uma operação fechada $(x,y) \mapsto x \bullet y$ sobre $G$ é um **grupo**, e denota-se $(G,\bullet)$, se essa operação $\bullet$ satisfaz aos seguintes axiomas:

**A1) Associatividade:** $a \bullet (b \bullet c) = (a \bullet b) \bullet c$, para todos $a$, $b$ e $c$ em $G$;

**A2) Elemento Neutro:** Existe um elemento $e \in G$, que satisfaz $a \bullet e = e \bullet a = a$, para todo $a \in G$. Este elemento é chamado de elemento neutro do grupo $G$;

**A3) Simétrico:** Para todo $a \in G$ existe um elemento $a' \in G$ tal que $a \bullet a' = a' \bullet a = e$, chamado de simétrico de $a$.

Um grupo $G$ é um **grupo finito** se possui uma quantidade finita de elementos e, no caso de $G$ ter infinitos elementos é denominado **grupo infinito**. Um grupo é chamado **abeliano**, se satisfazer a seguinte propriedade:

**A4) Comutatividade:** $a \bullet b = b \bullet a$, para todos $a$, $b$ em $G$.

**Exemplo 11.** O conjunto das permutações dos elementos de um conjunto E, denotado por S(E), é um grupo não comutativo se possuir três ou mais elementos (DOMINGUES, 2003, p.145). Já o conjunto $\mathbb{Z}_p$, com $p$ primo, forma um grupo finito e abeliano sobre a operação de multiplicação.

**Definição 10.** (Subgrupo) Seja $(G,\bullet)$ um grupo. Diz-se que um subconjunto não vazio $H \subset G$ é um **subgrupo** de $G$ se:

**i)** $H$ é fechado para a operação $\bullet$ (isto é, se $a,b \in H$ então $a \bullet b \in H$);

**ii)** $(H,\bullet)$ também é um grupo.

**Exemplo 12.** Têm-se que $m\mathbb{Z}$ (conjunto dos inteiros múltiplos de $m$) forma um subgrupo de $\mathbb{Z}$ sob a operação de adição.

**Definição 11.** (Classes Laterais) Seja $(G,\bullet)$ um grupo, $H \subset G$ e $a \in G$. O conjunto $a \bullet H = \{a \bullet h | h \in H\}$ é chamado **classe lateral à esquerda** de $H$ em $G$ contendo o elemento $a \in G$. Analogamente, o conjunto $H \bullet a = \{h \bullet a | h \in H\}$ é chamado **classe lateral à direita** de $H$ em $G$ contendo o elemento $a \in G$.

**Definição 12.** (Subgrupo Normal) Um subgrupo $(N,\bullet)$ de um grupo $(G,\bullet)$ é chamado **subgrupo normal** (ou invariante) se, para todo $x \in G$, se verifica a igualdade:

$$x \bullet N = N \bullet x,$$

ou seja, se as classes laterais à esquerda e à direita de $N$ em $G$ contendo $x \in G$ são iguais.

**Exemplo 13.** Todo subgrupo de um grupo abeliano é um subgrupo normal.

**Definição 13.** (Grupo Quociente) Sejam $(G,\bullet)$ um grupo e $N$ um subgrupo normal de $G$. Nessas condições, o **grupo quociente** de $G$ por $N$ é definido pelo conjunto $G/N = \{a \bullet N | a \in G\}$.

**Exemplo 14.** Sejam $G = \mathbb{Z}_6 = \{0,1,2,3,4,5\}$ e $H = \{0,3\}$. As classes laterais neste caso são: $0+H = H, 1+H = \{1,4\}, 2+H = \{2,5\}, 3+H = H$, já que essas três englobam todos os elementos de $\mathbb{Z}_6$.

**Definição 14.** (Grupos Cíclicos) Um grupo $G$ será chamado de **grupo cíclico** se, para algum elemento $a \in G$, se verificar a igualdade $G = \{a^m = \underbrace{a \bullet a \bullet ... \bullet a}_{m\ fatores} | m \in \mathbb{Z}\}$. Nessas condições, o elemento $a$ é chamado de **gerador** do grupo $G$.

**Exemplo 15.** O grupo aditivo $\mathbb{Z}$ é cíclico, pois todos os seus elementos são gerados por 1 ou -1. Na verdade, os únicos geradores deste grupo são os números 1 e -1.

**Definição 15.** (Problema do Logaritmo Discreto) Seja $G$ um grupo finito com operação $\bullet$ e $A \in G$ um gerador do subgrupo $H \subseteq G$, isto é, $H = \{A^j | j > 0\}$. Dados $A \in G$ e $B \in H$, calcular um inteiro $S$, onde $1 \leq S \leq |H| - 1$, tal que $A^S = \underbrace{A \bullet A \bullet ... \bullet A}_{S\ vezes} = B$

Note que $G$ não precisa ser comutativo mas $H$ sim, pois é cíclico. Este problema é computacionalmente inviável para certos grupos $G$ e subgrupos $H$, o que desperta certo interesse do ponto de vista criptográfico, pois está sujeito a ser mais resistente à ataques criptoanalíticos.

**Definição 16.** (Anéis) Um sistema algébrico constituído de um conjunto não vazio $A$ e de operações de adição $(x,y) \mapsto x+y$ e de multiplicação $(x,y) \mapsto x \cdot y$ é um anel se:

**A1 ao A4)** A é um grupo abeliano com respeito a adição;

**M1) Associatividade na Multiplicação:** $a \cdot (b \cdot c) = (a \cdot b) \cdot c$ para todos $a, b, c \in A$;

**M2) Distributividade:** $a \cdot (b+c) = ab + ac$ ou $(a+b) \cdot c = ac + bc$ para todos $a, b, c \in A$.

**Definição 17.** (Subanel) Um subconjunto não vazio $S$ de um anel $R$ é um **subanel** de $R$ se $S$ for fechado para as operações de adição e multiplicação de $R$, e $S$ com estas operações é um anel.

**Exemplo 16.** O conjunto dos números inteiros $(\mathbb{Z})$ é um subnel do anel $(\mathbb{R}, +, \cdot)$.

**Definição 18. i)** Um anel $R$ é um anel com identidade (ou com unidade) se existir um elemento $e \in R$ que satisfaz $a \cdot e = e \cdot a = a$ para todo $a \in R$. Este elemento $e$ é chamado de unidade (ou identidade) do anel $R$ e, usualmente denota-se tal elemento pelo número 1.

**ii)** Um anel $R$ é um **anel comutativo** se a operação de multiplicação for comutativa, ou seja, se $a \cdot b = b \cdot a$ para todos $a, b \in R$.

**iii)** Um anel $R$ é um **domínio de integridade** se $R$ for um anel comutativo com identidade $e \neq 0$ e se, para todo $a, b \in R$, $a \cdot b = 0$, então $a = 0$ ou $b = 0$).

**iv)** Um anel é um **anel de divisão** se o conjunto dos elementos não nulos de $R$ é um grupo multiplicativo.

**v)** Um anel de divisão comutativo é um **corpo**.

**Exemplo 17.** O conjunto das matrizes com entradas em $\mathbb{R}$ com as operações usuais de adição e multiplicação de matrizes é um anel não comutativo com unidade, onde a matriz identidade é o elemento identidade. Os conjuntos $\mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}$ com as operações usuais de adição e de multiplicação são exemplos de anéis comutativos com unidade. Os conjuntos $\mathbb{Q}, \mathbb{R}$ *e* $\mathbb{C}$ são corpos.

**Teorema 6.** Todo domínio de integridade finito é um corpo.

*Demonstração.* Seja $R$ um domínio de integridade finito com $n$ elementos $\{a_1, a_2, ..., a_n\}$. Tomando um elemento não nulo $a \in R$, com $a \neq 0$, considerando os produtos $aa_1, aa_2, ..., aa_n$ tem-se que tais produtos são distintos, pois se $i \neq j$ e $aa_i = aa_j$, então $a(a_i - a_j) = 0$, e como $a \neq 0$, deve-se ter $a_i - a_j = 0$, ou $a_i = a_j$. Então, cada um dos elementos de $R$ é da forma $aa_i$, em particular, $e = aa_i$ para algum $1 \leq i \leq n$, onde $e$ é o elemento identidade de $R$. Como $R$ é comutativo, têm-se que $a_i a = e$, e então, $a_i$ é o inverso multiplicativo de $a$. Então, os elementos não nulos de $R$ formam um grupo multiplicativo, ou seja, $R$ é um corpo. □

**Definição 19.** (Ideal) Seja $A$ um anel comutativo. Um subconjunto $I \subset A, I \neq \varnothing$, será chamado de **ideal** em $A$ se, para quaisquer $x, y \in I$ e para qualquer $a \in A$, verificarem-se as seguintes relações: $x - y \in I$ e $ax \in I$.

**Definição 20.** (Ideal Gerado) Seja $A$ um anel comutativo com unidade e $S \subseteq A$. Se $S = \{x_1, ..., x_k\}$, então $I = \{a_1x_1 + ... + a_kx_k | a_1, ..., a_k \in A\}$ é o **ideal gerado** por $S$.
Notação: $I =< x_1, ..., x_k >=< S >$

**Definição 21.** (Ideal Principal) Um ideal $I$ de um anel $R$ é chamado **principal** se existe um elemento $a \in R$ tal que:

$$I = aR = \{ar : r \in R\}.$$

Em outras palavras, o ideal gerado pelo elemento $a$ denomina-se de **ideal principal**.

**Definição 22.** (Ideal Maximal) Seja $M$ um ideal num anel comutativo $A$. Diz-se que $M$ é um **ideal maximal** se $M \neq A$ e se os únicos ideais de $A$ que contém $M$ são o próprio $M$ e $A$.

**Exemplo 18.** $2\mathbb{Z}$ é um ideal maximal em $\mathbb{Z}$. De fato, se $I$ é um ideal em $\mathbb{Z}$ que contém $2\mathbb{Z}$ propriamente, então $I$ possui um número ímpar $2t+1$. Mas, como $2t \in I$, pois $2t$ pertence a $2\mathbb{Z}$ e $I \supset 2\mathbb{Z}$, então $(2t+1) - (2t) = 1 \in I$. De onde, $I = \mathbb{Z}$.

**Teorema 7.** (Anel Quociente) Seja $I$ um ideal em um anel comutativo com unidade $A$. Considerando-se $I$ como subgrupo normal de $A$, e definindo-se a multiplicação em $A/I$ da seguinte forma:

$$(a+J)(b+J) = (ab+J),$$

temos que este cojunto quociente $(A/I)$ com esta operação é um anel chamado de **anel quociente** de $A$ pelo ideal $I$.

*Demonstração.* Primeiro, é necessário provar que esta multiplicação está bem definida, ou seja, que não depende dos elementos de $A$ usados na representação das classes. Para isso, supondo $a_1 + I = a_2 + I$ e $b_1 + I = b_2 + I$, será mostrado que $(a_1b_1) + I = (a_2b_2) + I$.

De $a_1 + I = a_2 + I$ segue que $a_1 - a_2 \in I$ e, analogamente, $b_1 - b_2 \in I$. Portanto, levando-se em conta que $I$ é um ideal em $A$:

$$b_1(a_1 - a_2) \in I \text{ e } a_2(b_1 - b_2) \in I.$$

Logo,

$$[b_1(a_1 - a_2) + a_2(b_1 - b_2)] = (a_1b_1 - a_2b_2) \in I.$$

O que significa que $a_1b_1 + I = a_2b_2 + I$.

Falta provar as propriedades da multiplicação necessárias para completar a estrutura de anel comutativo em $A/I$. Dado que o raciocínio é o mesmo sempre, nesta demonstração será

mostrado a distributividade da multiplicaçao em relação à adição. Para isso considere $a,b,c \in A$. Então:

$$\begin{aligned}(a+I)[(b+I)+(c+I)] &= (a+I)[(b+c)+I] \\ &= [a(b+c)]+I \\ &= (ab+ac)+I \\ &= (ab+I)+(ac+I) \\ &= (a+I)(b+I)+(a+I)(b+I).\end{aligned}$$

O que encerra a demonstração. □

**Teorema 8.** Seja $A$ um anel comutativo com unidade e $M$ um ideal maximal de $A$, então $A/M$ é um corpo.

*Demonstração.* Como $A$ é um anel comutativo com unidade, então $A/M$ também é um anel comutativo com unidade. Seja $a+M \in (A/M)-\{0+M\}$, assim, $a \notin M$. Seja $J = \{ca+m | c \in A \text{ e } m \in M\}$, logo $J$ é um ideal de $A$ com $M \subsetneq J \subset A$. Por hipótese, têm-se que $M$ é maximal, então $J = A$.

Em particular, $1 = c_0a + m_0$, para algum $c_0 \in A$ e $m_0 \in M$. Assim,

$$1+M = c_0a+m_0+M \Rightarrow 1+M = c_0a+M \Rightarrow 1+M = (c_0+M)(a+M) \Rightarrow (a+M)^{-1} = (c_0+M)$$

Portanto, $A/M$ é um corpo, como esperado. □

**Teorema 9.** $\mathbb{Z}/ < p >$, o anel quociente de $\mathbb{Z}$ pelo ideal gerado pelo primo $p$, é um corpo.

*Demonstração.* Pelo Teorema 6 é suficiente mostrar que $\mathbb{Z}/ < p >$ é um domínio de integridade. Têm-se que 1 é a identidade de $\mathbb{Z}/ < p >$, e $ab = 0$ se, e somente se, $ab = kp$ para algum inteiro $k$. Mas como $p$ é primo, $p$ divide $ab$ se, e somente se, $p$ divide ao menos um dos fatores. Portanto, ou $a = pc = 0$ ou $b = pc = 0$, para algum $c \in \mathbb{Z}$, de modo que $\mathbb{Z}/ < p >$ não contém divisores de zero. □

**Observação 1.** Trabalhar com $\mathbb{Z}_p$ é o mesmo que trabalhar com congruências dos inteiros módulo $n$ portanto, todas as propriedades vistas com congruências módulo $n$ podem ser utilizadas sobre o anel $\mathbb{Z}_p$.

## 2.4 ALGUNS CONCEITOS COMPUTACIONAIS

Esta seção é destinada a definições de elementos computacionais no qual será muito utilizada para descrever o processo de cifragem e decifragem do AES, para mais detalhes

consulte Mizrhi (2008).

### 2.4.1 SISTEMA BINÁRIO

Um computador só lê informações de duas maneiras distintas, ligado ou desligado, que usualmente é representado por 1 ou 0, respectivamente. Ou seja, um computador só compreende dados de $\mathbb{Z}_2$. Essas entradas (1 e 0) são chamados de dígitos binários ou *bits*. O computador numera os *bits* de uma variável da direita para a esquerda. Seja $a_i$ um *bit* temos $(a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0)$ uma variável de oito *bits* sendo $a_0$ denominado de *bit* menos significativo e $a_7$ de *bit* mais significativo.

Como um *bit* só pode assumir dois valores distintos, 1 ou 0, um valor armazenado em 8 *bits* pode ter $2^8 = 256$ combinações diferentes. Para realizar a conversão do sistema binário para o decimal deve-se levar em conta que o valor numérico decimal de um *bit* está ligado à 2 elevado à potência igual sua posição.

**Exemplo 19.** Seja o número em binário 01100110; têm-se que tal valor representa o número 102 pois

$$102 = 2^7 \cdot 0 + 2^6 \cdot 1 + 2^5 \cdot 1 + 2^4 \cdot 0 + 2^3 \cdot 0 + 2^2 \cdot 1 + 2^1 \cdot 1 + 2^0 \cdot 0 = 0 + 64 + 32 + 0 + 0 + 4 + 2 + 0$$

Para a conversão de um número decimal em binário, realiza-se divisões sucessivas por 2 e anota-se seus restos, que serão 0 ou 1. A leitura, em binário, se faz tomando do último resto até o primeiro.

**Exemplo 20.** Tomando o número 102, sabe-se do exemplo anterior que seu valor em binário é 01100110.

$$\begin{aligned} 102 &= 2 \cdot 51 + \mathbf{0} \\ 51 &= 2 \cdot 25 + \mathbf{1} \\ 25 &= 2 \cdot 12 + \mathbf{1} \\ 12 &= 2 \cdot 6 + \mathbf{0} \\ 6 &= 2 \cdot 3 + \mathbf{0} \\ 3 &= 2 \cdot 1 + \mathbf{1} \\ 1 &= 2 \cdot 0 + \mathbf{1} \\ 0 &= 2 \cdot 0 + \mathbf{0} \end{aligned}$$

Têm-se ainda que um *byte* ou um octeto corresponde a 8 *bits* e, um *word* corresponde a 4 *bytes*.

**Definição 23.** (Operação XOR) XOR (também denominado **ou exclusivo** e denotado por $\oplus$) é um operador lógico computacional que trabalha sob dois ou mais valores lógicos, no qual produz valor verdadeiro apenas se a quantidade de valores verdadeiros for ímpar. Sua tabela verdade e sua respectiva interpretação computacional são descritas abaixo:

| p | q | $p \oplus q$ |
|---|---|---|
| F | F | F |
| F | V | V |
| V | F | V |
| F | F | F |

**Tabela 2: Valores Lógicos do XOR**

| p | q | $p \oplus q$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**Tabela 3: Valores Binários do XOR**

### 2.4.2 SISTEMA HEXADECIMAL

No sistema hexadecimal cada dígito representa quatro *bits*, nos quais podem representar 16 números distintos. Assim, o sistema usa números de 0 a 9 e letras de A a F. Tais letras se fazem necessárias para não confundir na leitura de tal sistema. A tabela para leitura e conversões do sistema hexadecimal para o sistema binário e o decimal encontra-se abaixo.

## 2.5 POLINÔMIOS

A aritmética dos polinômios sobre um conjunto finito fornece meios de cifrar e decifrar grandes partes de mensagens através de cifras de bloco como o AES. Abaixo, será definido alguns conceitos importantes de polinômios que serão base para compreender algumas cifras. Detalhes das demonstrações desta seção encontram-se na obra de Lidl e Niederreiter (1987).

**Definição 24.** (Polinômios) Seja $R$ um anel. Um polinômio sobre $R$ é uma soma formal do tipo

$$f(x) = \sum_{i=0}^{\infty} a_i x^i, \tag{4}$$

onde $a_i \in R$, $\forall i \in \mathbb{N}$ e $a_j \neq 0$ para uma quantidade finita de índices $j \in \mathbb{N}$ ou $a_j = 0$ para todo índice $j \in \mathbb{N}$, caso este em que o polinômio $f(x) = 0$ é chamado de polinômio nulo ou polinômio

| Hexadecimal | Binário | Decimal |
|---|---|---|
| 0 | 0000 | 0 |
| 1 | 0001 | 1 |
| 2 | 0010 | 2 |
| 3 | 0011 | 3 |
| 4 | 0100 | 4 |
| 5 | 0101 | 5 |
| 6 | 0110 | 6 |
| 7 | 0111 | 7 |
| 8 | 1000 | 8 |
| 9 | 1001 | 9 |
| A | 1010 | 10 |
| B | 1011 | 11 |
| C | 1100 | 12 |
| D | 1101 | 13 |
| E | 1110 | 14 |
| F | 1111 | 15 |

**Tabela 4: Respectivos Valores em Hexadecimal, Binário e Decimal**

zero. Os $a_i$'s são chamados de coeficientes do polinômio $f(x)$. O maior índice $k$ tal que $a_k$ é diferente de zero é o grau do polinômio $f(x)$, se k for diferente de zero. Neste caso o coeficiente $a_k$ é chamado de coeficiente líder do polinômio $f(x)$ e denota-se por $k = gr(f(x)) = deg(f(x))$. Por convenção, tem-se que $deg(0) = -\infty$.

**Exemplo 21.** $f(x) = 3x^3 + 5x^2 + 0,2x + \sqrt{2}$ é um polinômio sobre o anel dos números reais ($\mathbb{R}$).

**Definição 25.** (Operações sobre os Polinômios) Dados dois polinômios, $f(x) = \sum_{i=0}^{n} a_i x^i$ e $g(x) = \sum_{j=0}^{m} b_j x^j$, sobre um anel $R$, tem-se que as operações de adição e multiplicação são definidas, respectivamente, da seguinte forma:

**i)** $f(x) + g(x) = \sum_{i=0, j=0}^{max(n,m)} (a_i + b_j)x^i$

**ii)** $f(x) \cdot g(x) = \sum_{k=0}^{m+n} c_k x^k$, onde $c_k = \sum_{\substack{i+j=k \\ 0 \le i \le n, 0 \le j \le m}} a_i b_j$

**Exemplo 22.** Dados os polinômios $f(x) = x^5 + 3x^3 + 2x^2 + 1$ e $g(x) = 2x + 3$ sobre $\mathbb{Z}$. Têm-se que $f(x) + g(x) = x^5 + 3x^3 + 2x^2 + 2x + 4$ e $f(x) \cdot g(x) = 2x^6 + 3x^5 + 6x^4 + 13x^3 + 6x^2 + 2x + 3$

**Definição 26.** (Polinômio Mônico) Um polinômio $f(x)$ é mônico se $R$ tem elemento unidade 1 e se o coeficiente líder de $f(x)$ for 1.

**Exemplo 23.** Têm-se que o polinômio $g(x)$ dado no exemplo anterior é mônico.

**Definição 27.** (Anel Polinomial) O anel formado pelos polinômios sobre $R$ com as operações descritas na Definição 24 é chamado de **anel polinomial** sobre $R$ e é denotado por $R[x]$.

**Teorema 10.** Seja $R$ um anel. Então,

**(i)** $R[x]$ é comutativo se, e somente se, $R$ é comutativo.

**(ii)** $R[x]$ é um anel com identidade se, e somente se, $R$ tem identidade.

**(iii)** $R[x]$ é um domínio de integridade se, e somente se, $R$ é um domínio de integridade.

A partir deste ponto, será admitido que $F$ é um corpo.

**Teorema 11.** (Algoritmo da Divisão) Sejam $f, g \in F[x]$ com $g \neq 0$. Então, existe um único par $(q, r)$ de polinômios tais que:

$$f = qg + r, \ onde \ deg(r) < deg(g) \tag{5}$$

**Definição 28.** Um polinômio $p \in F[x]$ é dito ser irredutível sobre $F$ (ou irredutível em $F[x]$, ou primo em $F[x]$) se $p$ tem grau positivo e $p = bc$ com $b, c \in F[x]$ implicar em $b$ ou $c$ ser um polinômio constante. Um polinômio $f(x)$ é dito constante quando $f(x) = k$ para algum $k \in F$.

**Exemplo 24.** O polinômio $m(x) = x^8 + x^4 + x^3 + x + 1$ é irredutível sobre $\mathbb{Z}_2$.

**Teorema 12.** (Fatoração Única em $F[x]$) Qualquer polinômio $f \in F[x]$ de grau positivo pode ser escrito na forma

$$f = a p_1^{e_1} ... p_k^{e_k}, \tag{6}$$

onde $a \in F$, $p_1, ..., p_k$ são polinômios mônicos irredutíveis distintos em $F[x]$, e $e_1, ..., e_k$ são inteiros positivos. Além disso, esta fatoração é única a menos da ordem em que os fatores são encontrados no produto.

**Teorema 13.** Todo ideal de $F[x]$ é principal.

*Demonstração.* Seja $I$ um ideal de $F[x]$ e seja $f(x)$ um polinômio não nulo de menor grau em $I$. Se $g(x) \in F[x]$ então, pelo algoritmo de Euclides, existe um único par $(q(x), r(x)) \in F[x] \times F[x]$ tal que

$$g(x) = q(x)f(x) + r(x), \text{ onde } r(x) = 0 \text{ ou } 0 \leq deg(r(x)) < deg(f(x)).$$

Assim, como $r(x) = g(x) - q(x)f(x) \in I$ então, pela primalidade de $f(x)$, segue que $r(x) = 0$ e, $g(x) = q(x)f(x) \in < f >$. Logo $< f >= I$. □

**Teorema 14.** Se $f(x) \in F[x]$ é irredutível e $F$ é um corpo, então $I =< f(x) >$ (ideal gerado por $f(x)$) é maximal.

*Demonstração.* Seja $L$ um ideal de $F[x]$ tal que $I \subsetneq L \subset F[x]$. Como $I \subsetneq L$, então existe $g(x) \in L - I$. Assim, como $g(x) \notin I =< f(x) >$, então $J =< g(x), f(x) >=< h(x) >$, para algum $h(x) \in F[x]$, pelo Teorema 13. Portanto, $h(x)|g(x)$ e $h(x)|f(x)$ e, como $f(x)$ é irredutível e $g(x) \notin < f(x) >$, temos que $h(x) = 1$. Logo $F[x] = F[x] \cdot 1 = J = L$. □

### 2.5.1 ARITMÉTICA POLINOMIAL MODULAR E O $GF(2^8)$

Considere o conjunto $\mathbb{Z}_p^{(n)}[x]$ de todos os polinômios de grau $n-1$ ou menor sobre o corpo $\mathbb{Z}_p$. Assim, cada polinômio tem a forma

$$f(x) = a_{n-1}x^{n-1} + ... + a_1x + a_0 = \sum_{i=0}^{n-1} a_i x^i, \tag{7}$$

onde cada $a_i$ assume um valor no conjunto $\{0, 1, ..., p-1\}$. Existe um total de $p^n$ polinômios diferentes em $\mathbb{Z}_p^{(n)}[x]$.

Para $\mathbb{Z}_p^{(n)}[x]$ ser um corpo finito, o conjunto deve atender às seguintes propriedades:

1. A aritmética segue as regras da aritmética polinomial.
2. A aritmética sobre os coeficientes é realizada sobre o corpo $\mathbb{Z}_p$.
3. Se a soma ou o produto resultar em um polinômio de grau maior que $n-1$, então este será reduzido módulo algum polinômio irredutível $m(x)$ de grau $n$; ou seja, $r(x) \equiv f(x)\ (mod\ m(x))$ onde $0 \leq deg(r(x)) < n$.

**Exemplo 25.** Seja $\mathbb{Z}_2[x]$ o conjunto dos polinômios em $\mathbb{Z}_2$. Tome o polinômio $f(x) = x^{10} + x^5 + x^2 + x + 1$ e o polinômio irredutível $m(x) = x^8 + x^4 + x^3 + x + 1$. Temos que $x^{10} + x^5 + x^2 + x + 1 \equiv x^6 + x^3 + x + 1\ (mod\ x^8 + x^4 + x^3 + x + 1)$.

Tem-se que $\mathbb{Z}_p[x]/f(x)$ será um corpo se, e somente se, $f(x)$ for irredutível. O inverso de qualquer elemento do corpo $\mathbb{Z}_p[x]/f(x)$, com exceção do zero, existe e é calculável. Em criptografia, o caso particular do corpo $\mathbb{Z}_2[x]/f(x)$ é muito importante e é chamado de Corpo Finito de Galois e denotado por $GF(2^m)$ onde $m = deg(f)$. Na criptografia AES, trabalha-se com o polinômio irredutível $m(x) = x^8 + x^4 + x^3 + x + 1$ como será visto no capítulo 3.

Como os coeficientes dos polinômios nestes corpos são 0 ou 1, pode-se definir a soma como um XOR dos coeficientes dos polinômios.

**Exemplo 26.** Considere os seguintes polinômios em $GF(2^8)$: $f(x) = x^7 + x^5 + x$ e $g(x) = x^7 + x^6 + x^5 + x^2 + x + 1$. A soma, ou seja, $f(x) + g(x)$ é dada por:

- $(x^7 + x^5 + x) + (x^7 + x^6 + x^5 + x^2 + x + 1) = x^6 + x^2 + 1$ (Notação Polinomial)
- $(10100010) \oplus (11100111) = (01000101)$ (Notação Binária)
- $\{A2\} \oplus \{E7\} = \{45\}$ (Notação Hexadecimal)

A multiplicação é mais complexa do que a soma porém, existe uma técnica razoavelmente simples. Nos exemplos abaixo será trabalhado com o polinômio irredutível $m(x) = x^8 + x^4 + x^3 + x + 1$, que é o utilizado no AES. Primeiramente, observa-se que:

$$x^8 \ (mod\ m(x)) = x^4 + x^3 + x + 1$$

Agora, considere um polinômio em $GF(2^8)$ que tem a forma $f(x) = b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0$. Ao multiplicar por $x$ obtem-se:

$$x \cdot f(x) = b_7x^8 + b_6x^7 + b_5x^6 + b_4x^5 + b_3x^4 + b_2x^3 + b_1x^2 + b_0x \ (mod\ m(x))$$

Se $b_7 = 0$, então o resultado já está em sua forma reduzida, caso contrário, a redução módulo $m(x)$ é obtida usando $x^4 + x^3 + x + 1$:

$$x \cdot f(x) = (b_7x^8 + b_6x^7 + b_5x^6 + b_4x^5 + b_3x^4 + b_2x^3 + b_1x^2 + b_0x) + (x^4 + x^3 + x + 1)$$

Segue que a multiplicação por $x$, isto é, a multiplicação por (00000010), pode ser repensada como um deslocamento de um *bit* para a esquerda seguido de um XOR *bit* a *bit* com (00011011), ou seja:

$$x \cdot f(x) = \begin{cases} (b_6b_5b_4b_3b_2b_1b_00) & se\ b_7 = 0 \\ (b_6b_5b_4b_3b_2b_1b_00) \oplus (00011011) & se\ b_7 = 1 \end{cases}.$$

**Exemplo 27.** Seja $f(x) = x^7 + x^5 + x^2 + x$ e $g(x) = x^6 + x^4 + x^3 + 1$, deseja-se calcular $f(x) \cdot g(x)\ (mod\ m(x))$, ou seja, deseja-se fazer $(10100110) \cdot (01011001)$ para isso, precisa-se dos resultados da multiplicação por potências de $x$:

$$(10100110) \cdot (00000010) = (01001100) \oplus (00011011) = (01010111)$$

$$(10100110) \cdot (00000100) = (10101110)$$

$$(10100110) \cdot (00001000) = (01011100) \oplus (00011011) = (01000111)$$

$(10100110) \cdot (00010000) = (10001110)$

$(10100110) \cdot (00100000) = (00011100) \oplus (00011011) = (00000111)$

$(10100110) \cdot (01000000) = (00001110)$

$(10100110) \cdot (10000000) = (00011100)$

Assim, tem-se que:

$(10100110) \cdot (01011001) = (10100110) \cdot [(01000000) \oplus (00010000) \oplus (00001000) \oplus (00000001)] = (00001110) \oplus (10001110) \oplus (01000111) \oplus (10100110) = (01100001)$

que é equivalente a $x^6 + x^5 + 1$, portanto $f(x) \cdot g(x) = x^6 + x^5 + 1 \ (mod\ m(x))$.

## 2.6 CURVAS ELÍPTICAS

Curvas elípticas têm grande atuação na criptografia devido à sua segurança contra criptoanálise por conta do Problema do Logaritmo Discreto aplicado nelas e pelo tamanho da chave, que é menor do que a chave utilizada na criptografia RSA porém, transfere uma segurança equivalente.

**Definição 29.** (Curva Elíptica) Seja um primo $p > 3$ e sejam $a, b \in \mathbb{Z}_p$ tais que $4a^3 + 27b^2 \neq 0 \ (mod\ p)$ (esta última condição equivale a exigir que $x^3 + ax + b$ não contenha algum fator repetido.) **Curva elíptica** é o conjunto união de um ponto infinito $\mathfrak{D}$ e as soluções $(x, y)$ de $y^2 = (x^3 + ax + b) \ (mod\ p)$.

**Definição 30.** Soma de dois pontos $P = (x_P, y_P)$, $Q = (x_Q, y_Q)$ de uma curva elíptica é definida como $R = (x_R, y_R)$ tal que:

1. se $x_P = x_Q$, $y_P = -y_Q$, então $R = \mathfrak{D}$;
2. senão, $x_R = t^2 - x_P - x_Q$, $y_R = t(x_P - x_R) - y_P$ onde $t = \begin{cases} \dfrac{y_Q - y_P}{x_Q - x_P} & se\ P \neq Q \\ \dfrac{3(x_P)^2 + a}{2y_p} & se\ P = Q \end{cases}$
3. $P + \mathfrak{D} = \mathfrak{D} + P = P$

Nota-se que a soma, assim definida, mostra que os pontos de uma curva elíptica formam um grupo comutativo com $\mathfrak{D}$ como identidade.

**Definição 31.** O inverso de um ponto $P = (x, y)$ de uma curva elíptica é o ponto $(x, -y)$ que é denotado por $-P = -(x, y)$.

**Exemplo 28.** A figura 1 demonstra a representação geométrica da soma nas curvas elípticas, tomando para o exemplo a curva $y^2 = x^3 - 2x + 3$. Os pontos $P$ e $Q$ pertencem a curva e sua soma é obtida trançando-se uma reta sobre eles e obtendo o ponto $R' = -R$. O resultado da soma é dado pelo ponto $R$ simétrico, pelo eixo $x$, ao ponto $R'$.

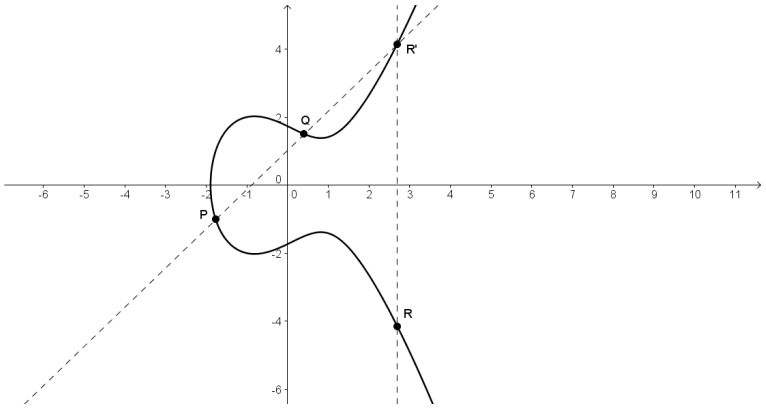


**Figura 1: Soma de dois pontos distintos em uma curva elíptica**

**Exemplo 29.** Para representar a duplicação de um ponto, ou seja, $P + P = 2P$, traça-se uma reta tangente a $P$ onde, a interseção $R'$ é o inverso do resultado de tal duplicação. A figura 2 demonstra tal conceito utilizando a curva elíptica de $y^2 = x^3 + 3x$.

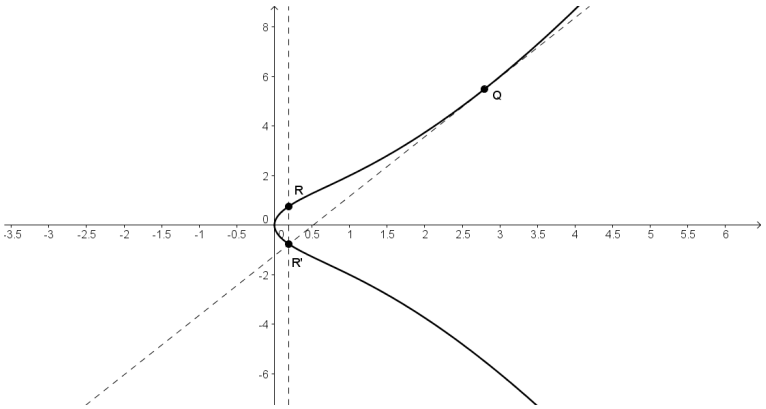


**Figura 2: Duplicação de um ponto em curvas elípticas**

As curvas elípticas sobre um conjunto $\mathbb{Z}_p$ trabalha com os resíduos quadráticos módulo $p$ que será definido a seguir.

**Definição 32.** (Resíduos Quadráticos *mod n*) Seja $a \in \mathbb{Z}_n^*$; $a$ é um resíduo quadrático módulo $n$ se existir um $x \in \mathbb{Z}_n^*$ tal que $x^2 = a \ (mod\ n)$. Seja $g \in \mathbb{Z}_p^*$ um gerador de $\mathbb{Z}_p^*$ no caso particular de $p$ ser um primo ímpar. Neste caso, $a \in \mathbb{Z}_p^*$ é um resíduo quadrático se, e somente se, $a = g^i \ (mod\ p)$ para um $i$ inteiro par.

Observe que $0 \notin \mathbb{Z}_p^*$, portanto também não pertencerá ao conjunto dos resíduos quadráticos.

**Exemplo 30.** Sendo $g = 2$ um gerador de $\mathbb{Z}_{13}^*$, tem-se:

| i | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $2^i$ mod 13 | 1 | 2 | 4 | 8 | 3 | 6 | 12 | 11 | 9 | 5 | 10 | 7 |

Portanto, o conjunto dos resíduos quadráticos de $\mathbb{Z}_{13}^*$ é $\{1,3,4,9,10,12\}$.

**Exemplo 31.** Considere a curva elíptica $E_1 : y^2 = x^3 + 3x + 6$ sobre $\mathbb{Z}_{13}$. Primeiro deve-se verificar para quais valores de $x \in \mathbb{Z}_{13}$ o valor $x^3 + 3x + 6$ é um resíduo quadrático. Tem-se pelo exemplo anterior que o conjunto dos resíduos quadráticos de $\mathbb{Z}_{13}^*$ é $\{1,3,4,9,10,12\}$. Na tabela 5 é apresentada todas as raízes dos valores que são resíduos quadráticos:

| **x** | $(x^3 + 3x + 6)$ *mod* 13 | **y** |
|---|---|---|
| 0 | 6 | - |
| 1 | 10 | 6 e 7 |
| 2 | 7 | - |
| 3 | 3 | 4 e 9 |
| 4 | 4 | 2 e 11 |
| 5 | 3 | 4 e 9 |
| 6 | 6 | - |
| 7 | 6 | - |
| 8 | 9 | 3 e 10 |
| 9 | 8 | - |
| 10 | 9 | 3 e 10 |
| 11 | 5 | - |
| 12 | 2 | - |

**Tabela 5: Raízes dos Resíduos Quadráticos**

Para cada resíduo quadráticos tem-se duas raízes, ou seja, cada valor de $x$ que resulta em um resíduo quadrático corresponde a dois valores de $y$ na curva $E_1$, ou seja, um total de 12 pares $(x,y)$ na curva. Considerando o ponto infinito $\mathfrak{D}$ tem-se um total de 13 pontos, isto é, a ordem do grupo é 13.

Qualquer grupo de ordem igual a um primo é cíclico. Logo, os pontos de $E_1$ são isomorfos a $\mathbb{Z}_{17}$ e ainda, qualquer ponto distinto de $\mathfrak{D}$ é gerador dos pontos de $E_1$. Tomando

como exemplo o ponto $(3,4)$ têm-se, pela Definição 24, que a seguinte tabela descreve os pontos gerados por somas sucessivas de $(3,4)$. (Considere o ponto $(x_P, y_P) = (3,4)$; o ponto $(x_Q, y_Q)$ são pontos $(x_R, y_R)$ gerados na linha anterior e que as operações dadas abaixo estão sobre módulo 13).

| $j \times (3,4)$ | $t = \frac{y_Q - y_P}{x_Q - x_P}$ | $x_R = t^2 - x_P - x_Q$ | $y_R = t(x_P - x_R) - y_R$ |
|---|---|---|---|
| $2 \times (3,4)$ | 7 | 4 | 2 |
| $3 \times (3,4)$ | 11 | 10 | 10 |
| $4 \times (3,4)$ | 12 | 1 | 7 |
| $5 \times (3,4)$ | 5 | 8 | 10 |
| $6 \times (3,4)$ | 9 | 5 | 4 |
| $7 \times (3,4)$ | 0 | 5 | 9 |
| $8 \times (3,4)$ | 9 | 8 | 3 |
| $9 \times (3,4)$ | 5 | 1 | 6 |
| $10 \times (3,4)$ | 12 | 10 | 3 |
| $11 \times (3,4)$ | 11 | 4 | 11 |
| $12 \times (3,4)$ | 7 | 3 | 9 |

**Tabela 6: Pontos Gerados por (3,4)**

### 2.6.1 PROBLEMA DO LOGARITMO DISCRETO SOBRE CURVAS ELÍPTICAS

Como já visto, no problema do logaritmo discreto, pode-se considerar qualquer grupo finito $G$ com operação $\bullet$ como se vê a seguir: seja $A \in G$ um gerador do grupo $H \subseteq G$, isto é, $H = \{A^j | j \geq 0\}$; e dados $A \in G$ e $B \in H$, calcular um inteiro $s$ tal que, $1 < S \leq |H| - 1$ tal que $A^S = B$, onde $A^S = \underbrace{A \bullet A \bullet ... \bullet A}_{S \text{ vezes}}$.

No caso de $G$ ser o conjunto dos pontos de uma curva elíptica, a operação $\bullet$ é a mesma de dois pontos. Este problema sobre uma curva elíptica $E : y^2 = x^3 + ax + b$ em $\mathbb{Z}_p$ consiste em, dados dois pontos $Q, P$ em $E$, calcular $S$ tal que $Q = SP$. A segurança do sistema é garantido pela dificuldade de se resolver tal problema, sendo comparado com a dificuldade em se fatorar números primos devido à demora dos algoritmos computacionais que procuram soluções dos mesmos.

## 2.7 OUTROS CONCEITOS IMPORTANTES

**Método das Divisões Sucessivas:** O Método das Divisões Sucessivas opera através de divisões inteiras sucessivas. O objetivo é calcular o máximo divisor comum entre os inteiros positivos $a$ e $b$, inicialmente realiza-se a divisão inteira de $a$ por $b$, obtendo um quociente $q_1$ e um resto $r_1$. Se este resto for nulo, então $mdc(a,b) = b$ e o algoritmo termina. Caso contrário, o divisor da última divisão será usado como dividendo de uma nova divisão e o resto da última divisão será usado como divisor desta nova divisão. Isto é, divide-se $b$ por $r_1$, obtendo um quociente $q_2$ e um resto $r_2$. Caso o resto $r_2$ não seja nulo, repetimos o processo descrito acima, dividindo $r_1$ por $r_2$ e assim por diante. O algoritmo termina quando obtivermos um resto zero. Teremos então que o máximo divisor comum entre $a$ e $b$ será o último resto diferente de zero nesta sequência de divisões. Pode-se reduzir tal processo nas equações abaixo:

$$\begin{array}{ccc}
a = bq_1 + r_1 & 0 < r_1 < b & \alpha_1 a + \beta_1 b = r_1 \\
b = r_1 q_2 + r_2 & 0 < r_2 < r_1 & \alpha_2 a + \beta_2 b = r_2 \\
r_1 = r_2 q_3 + r_3 & 0 < r_3 < r_2 & \alpha_3 a + \beta_3 b = r_3 \\
\vdots & \vdots & \vdots \\
r_{n-3} = r_{n-2} q_{n-1} + r_{n-1} & 0 < r_{n-1} < r_{n-2} & \alpha_{n-1} a + \beta_{n-1} b = r_{n-1} \\
r_{n-2} = r_{n-1} q_n + r_n & r_n = 0 &
\end{array}$$

Onde $\alpha_i$ e $\beta_i$ são constantes inteiras de forma que, através de combinações lineares entre $a$ e $b$, reescrevem os restos das divisões sucessivas obtidas pelo processo.

**Função Phi de Euler:** A função Phi de Euler ($\Phi(n)$) é uma função que calcula todos os números coprimos de $n$ entre 0 e um inteiro $n-1$. Certamente, $\Phi(1) = 1$ e, a partir de $n > 1$, $\Phi(n) < n$. Quando $n$ é primo, $\Phi(n) = n-1$ pois todos os inteiros positivos menores do que $n$ são seus coprimos. Quando $n$ é composto, ou seja, $n = p_1^{e_1} p_2^{e_2} ... p_k^{e_k}$ e $p_1, p_2, ..., p_k$ são primos distintos, então:

$$\begin{aligned}
\Phi(n) &= (p_1^{e_1} - p_1^{e_1-1})(p_2^{e_2} - p_2^{e_2-1}) ... (p_k^{e_k} - p_k^{e_k-1}) \\
&= n\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right) ... \left(1 - \frac{1}{p_k}\right)
\end{aligned}$$

**Teorema 15.** (Teorema de Fermat) Seja $p$ um número primo e seja $r \equiv s \ (mod\ p-1)$.

1. Se o número inteiro $b$ não for divisível por $p$, então $b^{p-1} \equiv 1 \ (mod\ p) \Rightarrow b^r \equiv b^s \ (mod\ p)$.

2. Se $r$ e $s$ forem números inteiros positivos então, para todo número inteiro $b$ vale $b^r \equiv b^s \ (mod\ p)$.

*Demonstração.* 1. Considere o conjunto $\mathbb{Z}_p^* = \{1,2,...,p-1\}$ dos números em $\mathbb{Z}_p$ que são coprimos com $p$. Quando $x \in \mathbb{Z}_p^*$, então $bx \ (mod\ p)$ também pertence a $\mathbb{Z}_p^*$ e a função $f : \mathbb{Z}_p^* \to \mathbb{Z}_p^*$ definida por $f(x) = bx \ (mod\ p)$ é injetora. Isto implica que o conjunto $\{f(1), f(2), ..., f(p-1)\}$ é um rearranjo de $\mathbb{Z}_p^*$ e $f(1) \cdot f(2) \cdot ... \cdot f(p-1) = 1 \cdot 2 \cdot ... \cdot (p-1)$ e:

$$1b \cdot 2b \cdot ... \cdot (p-1)b \ (mod\ p) = 1 \cdot 2 \cdot ... \cdot (p-1) \ (mod\ p).$$

Multiplicando os dois lados pelos inversos de $2, ..., (p-1)$ módulo $p$, obtemos:

$$b^{p-1} \ (mod\ p) = 1$$

o que prova a primeira congruência.

Sendo $r \equiv s \ (mod\ p-1)$, existe um inteiro $k$ tal que $r = k(p-1)+s$ e $b^r \equiv b^{k(p-1)+s} \equiv (b^{(p-1)})^k b^s \equiv b^s \ (mod\ p)$ o que prova a segunda congruência.

2. Quando $b$ for divisível por $p$, as potências negativas de $b \ (mod\ p)$ não existem. Entretanto, quando $r$ e $s$ forem positivos, tanto $b^r \ (mod\ p)$ quanto $b^s \ (mod\ p)$ são iguais a zero, provando que $b^r \equiv b^s \ (mod\ p)$. □

**Teorema 16.** (Teorema de Euler) Seja $n \geq 2$ um número inteiro, $r$ e $s$ côngruos módulo $\Phi(n)$.

1. Se $b$, um número inteiro, e $n$ forem primos entre si, $b^{\Phi(n)} \equiv 1 \ (mod\ n) \Rightarrow b^r \equiv b^s \ (mod\ n)$.
2. Se $n$ for o produto de primos distintos, $r > 0$ e $s > 0$, então $b^r \equiv b^s \ (mod\ n)$, para todo inteiro $b$.

*Demonstração.* 1. O conjunto dos elementos de $\mathbb{Z}_n$ que posssue inverso multiplicativo módulo $n$ é denotado por $\mathbb{Z}_n^*$. A função $f : \mathbb{Z}_n^* \to \mathbb{Z}_n^*$, definida por $f(x) = bx \ (mod\ n)$ é injetora. Por este motivo, $ax \ (mod\ n) : x \in \mathbb{Z}_n^*$ é um rearranjo de $\mathbb{Z}_n^*$ o que acarreta nas igualdades:

$$\prod_{x \in \mathbb{Z}_n^*} ax \ (mod\ n) = a^{\Phi(n)} \prod_{x \in \mathbb{Z}_n^*} x \ (mod\ n) = \prod_{x \in \mathbb{Z}_n^*} x \ (mod\ n)$$

Multiplicando os dois lados da igualdade por $x^{-1} \ (mod\ n)$, para todo $x$ em $\mathbb{Z}_n^*$, obtemos:

$$b^{\Phi(n)} \ (mod\ n) = 1$$

2. Se $n = p_1 p_2 ... p_k$, onde todos os fatores primos são distintos sabe-se que:

$$\Phi(n) = \Phi(p_1)\Phi(p_2)...\Phi(p_k) = (p_1-1)(p_2-1)...(p_k-1)$$

Sendo $r \equiv s \ (mod\ n)$, então $r \equiv s \ (mod\ p_i)$, para $i = 1, 2, ..., k$. Pelo Teorema de Fermat, $b^r \equiv b^s \ (mod\ p_i)$ quando $b$ não for divisível por $p_i$, pois os dois lados são côngruos a zero. Por este motivo, quando $b$ é divisível por $p_i$, os expoentes devem ser positivos pois, se $r$ ou $s$ for nulo, em lugar de zero obtemos 1 e, se $r$ ou $s$ for negativo, não existe o inverso de $b$ módulo $p_i$. Assim, para $i = 1, 2, ..., k$; $p_i$ divide $b^r - b^s$ que, portanto, é divisível por $n$, provando que $b^r \equiv b^s \ (mod\ n)$. □

# 3 CRIPTOGRAFIA

Criptografia, do grego *kryptós* (secreto, escondido) e *gráphein* (escrita), é a arte de elaborar métodos de comunicação de forma que somente remetente e destinatário consigam ler as informações presentes no mesmo. Estes métodos são denominados **cifras** ou **cifrários**. O texto antes de ser submetido às mudanças de um cifrário é o **texto claro** e, após as mudanças, **texto cifrado**. O cifrário precisa ser um método seguro de embaralhamento da mensagem pois, caso exista algum interceptador no meio da comunicação que estes estejam utilizando para a troca de informações, ele poderá facilmente acessar as informações presentes nas mensagens.

O estudo para se desenvolver tais ataques é denominada **criptoanálise** e, assim como a criptografia, é uma ramificação da criptologia. Quando desenvolvem alguma técnica de cifragem nova, esta passa por testes criptoanalíticos já conhecidos para verificar a segurança do novo método.

Na história, tem-se que o primeiro cifrário foi utilizado pelo imperador César para se comunicar com suas tropas durante os combates na Europa. Na época, o cifrário utilizado por César era viável para seus objetivos porém, como mostraremos abaixo, o seu método era muito frágil para ataques. Este capítulo será destinado a explicitar alguns cifrários já criados pela humanidade. As informações retiradas para a escrita deste capítulo foram baseadas nas obras que estão discriminadas nas referências.

## 3.1 CIFRÁRIO DE CÉSAR

Como mencionado acima, César desenvolveu tal cifrário para se comunicar, de forma segura, com seus generais durante a guerra. O Cifrário de César consiste em uma substituição simples pelos valores “rotacionados” três unidades do alfabeto original. Por exemplo, como podemos observar na Tabela 7, a letra A passa a receber o valor da letra D, B recebe E, e assim sucessivamente.

| A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ |
| D | E | F | G | H | I | J | K | L | M | N | O | P |
| N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ |
| Q | R | S | T | U | V | W | X | Y | Z | A | B | C |

**Tabela 7: Substituição por Letras**

Em termos matemáticos, o Cifrário de César trabalha com congruências modulares sob módulo 26. Primeiro, atribui-se valores inteiros para cada letra do alfabeto entre 0 e 25, como na Tabela 8, depois aplica-se o seguinte algoritmo:

$$d+3 \equiv c \ (mod\ 26) \tag{8}$$

onde $d$ é o valor de entrada do algoritmo e $c$ é o valor de saída, gerando assim, o texto cifrado.

| A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ |
| 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

**Tabela 8: Substituição por Valores**

Assim, por exemplo, se César ordenasse aos seus generais um ataque com a seguinte mensagem: ATACAR A BASE SUL. Pela Tabela 8 temos que a mensagem ficaria da seguinte maneira: DWDFDU D EDVH VXO. Mas o que os seus generais fariam ao receber esta estranha mensagem? Simples, pelo fato de ser apenas uma simples substituição de letras, os generais de César só precisam realizar a substituição inversa de cada letra encontrando o texto claro. O algoritmo matemático para decifrar seria:

$$c-3 \equiv d \ (mod\ 26). \tag{9}$$

Mas como pode-se perceber, a única segurança que César tinha em seu método era que seus adversários não sabiam nada sobre criptoanálise, porém, hoje em dia qualquer aluno de ensino médio com um pouco de bom senso consegue traduzir os textos que foram embaralhados com o Cifrário de César e sem se esforçar muito.

Qualquer um que queira traduzir os textos cifrados advindo de métodos de substituições simples de letras, só precisa saber contar e, talvez, do contexto sobre o assunto que trata a mensagem. É de conhecimento público que as vogais são as letras que mais aparecem na língua portuguesa e a letra A é a que mais aparece dentre as vogais; se tiver algum monossílabo de uma única letra, essa letra é uma vogal; são poucas as palavras que contenha em sua escrita três ou mais consoantes em sequência; valores repetidos em sequência, na maioria das vezes, pode simbolizar as letras R e S, entre outros padrões na língua podem auxiliar no processo de criptoanálise.

## 3.2 CIFRA DE HILL

A Cifra de Hill realiza substituições em mais de uma letra de uma vez (o que a classifica como uma cifra multiletras) e foi elaborada por Lester Hill em 1929. O algoritmo toma $n$ letras do texto claro e as substitui por $n$ letras do texto cifrado. Tal substituição é determinada por tranformações lineares, onde cada letra do alfabeto recebe um valor inteiro entre 0 e 25 e são cifrados da seguinte forma:

$$\begin{aligned} c_1 &= k_{11}p_1 + k_{12}p_2 + ... + k_{1n}p_n \ (mod\ 26) \\ c_2 &= k_{21}p_1 + k_{22}p_2 + ... + k_{2n}p_n \ (mod\ 26) \\ \vdots & \qquad\qquad \vdots \\ c_n &= k_{n1}p_1 + k_{n2}p_2 + ... + k_{nn}p_n \ (mod\ 26) \end{aligned}$$

O que também pode ser expresso como vetores colunas e matrizes:

$$\begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} k_{11} & k_{12} & ... & k_{1n} \\ k_{21} & k_{22} & ... & k_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ k_{n1} & k_{n2} & ... & k_{nn} \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{pmatrix} \ (mod\ 26) \tag{10}$$

ou,

$$C = KP \ (mod\ 26) \tag{11}$$

onde $C$ e $P$ são vetores colunas de tamanho $n$ representando, respectivamente, o texto cifrado e o texto claro; $K$ é uma matriz $n \times n$ inversível módulo 26, que representa a chave de criptografia do sistema. Todas as operações são dadas sobre a aritmética modular (módulo 26).

**Exemplo 32.** Tomando o texto MANTENHA A POSICAO, os valores das letras dadas na Tabela 8 e a seguinte chave:

$$K = \begin{pmatrix} 24 & 15 & 3 & 1 \\ 25 & 9 & 17 & 7 \\ 10 & 8 & 5 & 16 \\ 21 & 19 & 2 & 4 \end{pmatrix},$$

assim, têm-se que $n = 4$ e o texto, após trocar pelos seus valores correspondentes, 12-0-13-19-4-13-7-0-0-15-14-18-8-2-0-14. Assim, as quatro primeiras letras do texto são criptografadas da seguinte maneira:

$$\begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{pmatrix} = \begin{pmatrix} 24 & 15 & 3 & 1 \\ 25 & 9 & 17 & 7 \\ 10 & 8 & 5 & 16 \\ 21 & 19 & 2 & 4 \end{pmatrix} \begin{pmatrix} 12 \\ 0 \\ 13 \\ 19 \end{pmatrix} \ (mod\ 26) =$$

$$= \begin{pmatrix} 346 \\ 654 \\ 489 \\ 354 \end{pmatrix} \ (mod\ 26) = \begin{pmatrix} 8 \\ 4 \\ 21 \\ 16 \end{pmatrix} \ (mod 26).$$

Continuando com os cálculos, chegaremos no seguinte texto cifrado: 8-4-21-16-0-24-23-7-25-5-10-21-2-4-8-2 = YEVQAYXHZFKVCEIC.

Para decriptografar, é necessário encontrar uma matriz $K^{-1}$ tal que $KK^{-1} = K^{-1}K = I$, onde $I$ é a matriz identidade e $K^{-1}$ é a matriz inversa de $K$. É fácil verificar que, para se conseguir traduzir o texto cifrado, basta aplicar a matriz inversa $K^{-1}$ ao texto cifrado. No exemplo trabalhado nesta seção, temos que a matriz inversa de $K$ módulo 26 é:

$$K^{-1} = \begin{pmatrix} 21 & 5 & 14 & 8 \\ 25 & 7 & 22 & 17 \\ 16 & 4 & 5 & 8 \\ 10 & 23 & 21 & 23 \end{pmatrix}.$$

Aplicando tal matriz nas quatro primeiras letras do texto cifrado, obtêm-se as primeiras quatro letras do texto original do qual partimos:

$$\begin{pmatrix} 21 & 5 & 14 & 8 \\ 25 & 7 & 22 & 17 \\ 16 & 4 & 5 & 8 \\ 10 & 23 & 21 & 23 \end{pmatrix} \begin{pmatrix} 8 \\ 4 \\ 21 \\ 16 \end{pmatrix} (mod\ 26) = \begin{pmatrix} 610 \\ 962 \\ 377 \\ 981 \end{pmatrix} (mod\ 26) = \begin{pmatrix} 12 \\ 0 \\ 13 \\ 19 \end{pmatrix} (mod\ 26).$$

## 3.3 CRIPTOGRAFIA RSA

Em 1976, Diffie e Hellman, criptógrafos estadunidenses, publicaram o artigo *New Directions in Cryptography* no qual foi proposto um modelo em que cada usuário teria duas chaves distintas; uma delas de conhecimento público, e a outra guardada secretamente. Tais chaves se correlacionam de forma que:

**i)** as aplicações dessas chaves no algoritmo sejam inversas entre si;

**ii)** o cálculo das duas seja computacionalmente fácil;

**iii)** o cálculo da privada seja computacionalmente difícil a partir do conhecimento da pública;

**iv)** o cálculo das aplicações sejam fáceis para quem conhece as chaves;

**v)** seja computacionalmente difícil calcular a aplicação da chave privada sem conhecimento da chave pública.

Assim, não existe dificuldades em distribuição de chaves como nas cifras vistas anteriormente já que remetente e destinatário devem manter em segredo a chave privada e, através de um meio público, compartilhar a chave pública.

Dois anos após a proposta de Diffie-Hellman, Ron Rivest, Adi Shamir e Len Adleman publicaram o artigo *A method for obtaining digital signatures and public key cryptosystems* contendo um algoritmo que explora a dificuldade computacional de fatorar um número inteiro em primos. Tal algoritmo segue as seguintes etapas para cifrar e decifrar uma mensagem:

1. Pré-codificação;
2. Escolha de dois números primos e Decomposição em blocos;
3. Cálculo da Chave de Codificação e Codificação;
4. Cálculo da Chave de Decodificação e Decodificação.

**Etapa 1:** Na pré-codificação será feita uma substituição de letras por valores parecida com o que foi feito na Tabela 8, porém, para evitar ambiguidade nos cálculo, os valores serão maiores e, os espaços da mensagem será associada ao valor 99. Assim, as letras receberão os seguintes valores:

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | B | C | D | E | F | G | H | I | J | K | L | M |
| ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ |
| 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 |

**Tabela 9: Substituição por Valores (RSA)**

**Exemplo 33.** Tomando a frase "OS NÚMEROS GOVERNAM O MUNDO". Esta é convertida em:

$$242899233022142724289916243114272310229924992230231324.$$

Através de tal conversão, pode-se perceber a importância de se utilizar valores maiores para cada letra pois, por exemplo, utilizando a tabela 8 o par de algarismos 24 poderia representar o par de letras CE, ou então, somente a letra Y.

**Etapa 2:** A escolha de dois números primos $(p,q)$ é de extrema importância para a Criptografia RSA. Para continuar com os cálculos do exemplo 33, tomando $p = 13$ e $q = 31$, calcula-se então $n = pq = 13 \cdot 31 = 403$. Após a escolha dos números primos, separa-se a mensagem numérica que se obtém na etapa 1 em blocos $b$ tal que, cada bloco $b$ seja menor que $n$ e de forma que nenhum bloco comece com o algarismo 0 (vale ressaltar que a separação em blocos não é única, varia de acordo com a pessoa que esteja trabalhando com o algoritmo).

Assim, a mensagem do exemplo 33 poderá ficar da forma:

$$242\text{-}89\text{-}92\text{-}330\text{-}221\text{-}42\text{-}72\text{-}42\text{-}89\text{-}91\text{-}62\text{-}43\text{-}114\text{-}272\text{-}310\text{-}229\text{-}92\text{-}49\text{-}92\text{-}230\text{-}231\text{-}324.$$

**Etapa 3:** Para codificar de fato a mensagem obtida acima, é necessário calcular um inteiro positivo $e$ que seja inversível módulo $\Phi(n)$, ou seja, encontrar um $e$ de modo que $mdc(e,\Phi(n)) = 1$. Note que $\Phi(n)$ pode ser calculado facilmente quando $p$ e $q$ são conhecidos já que $\Phi(n) = (p-1)(q-1) = 12 \cdot 30 = 360$. Escolhendo $e = 107$ então $mdc(107,360) = 1$.

O par obtido $(e,n)$ é a chave de codificação (que é a chave pública) que será utilizada para codificar cada bloco da etapa 2 em um novo bloco seguindo o seguinte algoritmo:

$$C(b) \equiv b^e \ (mod\ n) \tag{12}$$

onde $b$ é um bloco pré-codificado, $e$ é um inteiro inversível módulo $\Phi(n)$ e, $n$ é o produto dos dois números primos $p$ e $q$. Desta forma, o primeiro bloco será codificado da seguinte maneira:

$$242^{107} \equiv (242^2)^{53} \cdot 242 \equiv 129^{53} \cdot 242 \equiv (129^4)^{13} \cdot 129 \cdot 242 \equiv 222^{13} \cdot 187 \equiv$$
$$\equiv (222^3)^4 \cdot 222 \cdot 187 \equiv 1^4 \cdot 5 \equiv 5 \ (mod\ 403).$$

Assim, repetindo este procedimento para os demais blocos, a mensagem codificada fica:

5-201-92-174-78-282-379-201-182-186-166-303-168-279-135-92-355-92-172-287-103.

**Etapa 4:** Para decodificar a mensagem acima, é necessário conhecer a chave de decodificação $(n,d)$ (que é a chave secreta) onde $n$ é o número obtido pelo produto entre os primos $p$ e $q$ na etapa 1 e $d$ é o inverso multiplicativo de $e$ módulo $\Phi(n)$. O algoritmo da decodificação é:

$$D(a) \equiv a^d \ (mod\ n), \tag{13}$$

onde $a$ é um bloco da mensagem codificada. Para encontrar tal inverso $d$, utiliza-se do algoritmo da divisão de Euclides várias vezes até encontrar o resultado desejado. Aplicando o Algoritmo de Euclides nos dados obtidos no exemplo desta seção temos:

$$\begin{aligned}
360 &= 3 \cdot 107 + 39 \\
39 &= 360 - 3 \cdot 107 \\
107 &= 2 \cdot 39 + 29 \\
29 &= 7 \cdot 107 - 2 \cdot 360 \\
39 &= 29 + 10 \\
10 &= 3 \cdot 360 - 10 \cdot 107 \\
29 &= 2 \cdot 10 + 9 \\
9 &= 27 \cdot 107 - 8 \cdot 360 \\
10 &= 9 + 1 \\
1 &= 11 \cdot 360 - 37 \cdot 107.
\end{aligned}$$

Portanto, -37 é o inverso multiplicativo de 107 módulo 360. Como $d$ será um expoente, é viável que este seja positivo, ou seja, é necessário calcular $-37 \equiv 323 \ (mod\ 360)$. Assim,

$d = 323$ e $n = 403$ será a chave de decodificação do exemplo 28. Decodificando o primeiro bloco da mensagem codificada utilizando a equação (13), obtêm-se:

$$5^{323} \equiv (5^{17})^{19} \equiv (5^{10})^{19} \cdot (5^7)^{19} \equiv 129^{19} \cdot 346^{19} \equiv 304^{19} \equiv (304^3)^5 \cdot 304^4 \equiv$$
$$\equiv 125^5 \cdot 118 \equiv 125^4 \cdot 242 \equiv 404 \cdot 242 \equiv 242 \ (mod\ 403).$$

Repetindo o processo para todos os demais blocos da mensagem cifrada, encontra-se a mensagem pré-codificada da primeira etapa.

### 3.3.1 SEGURANÇA E O PORQUÊ O RSA FUNCIONA

A segurança da criptografia RSA se deve principalmente pela escolha dos primos $p$ e $q$ pois se o valor $n$ no par $(e,n)$ (que é a chave pública do sistema) for de fácil decomposição, é fácil encontrar o valor de $d$ (que é o inverso multiplicativo de $e$ módulo $\Phi(n)$) pois seria fácil calcular o valor da função $\Phi(n)$. Quando os valores $p$ e $q$ são suficientemente grandes e distantes, obtém-se um $n$ que seja muito difícil de se fatorar. Um outro pensamento que possa surgir para decodificar uma mensagem através do conhecimento de $(e,n)$ (a chave pública) seria tentar descobrir $b$ a partir da forma reduzida de $b^e$ módulo $n$, porém, tal maneira, sem o conhecimento de $d$, não foi descoberta até o momento, o que torna o método muito seguro.

O método funciona pois, se $b$ é um bloco da mensagem pré-codificada, $C(b)$ é o processo de codificação dada pela equação (12), $D(a)$ é o processo de decodificação dada pela equação (13) e $a$ um bloco da mensagem codificada por $C(b)$, então:

$$D(a) \equiv D(C(b)) \equiv D(b^e) \equiv (b^e)^d \equiv b^{ed} \ (mod\ n). \tag{14}$$

Como $d$ é o inverso multiplicativo de $e$ módulo $\Phi(n)$, logo $ed = 1 + k\Phi(n)$, para algum inteiro $k$. Observe que, como $e$ e $d$ são inteiros maiores que 2 e $\Phi(n) > 0$, então $k > 0$. Substituindo na equação (14):

$$b^{ed} \equiv b^{1+k\Phi(n)} \equiv b \cdot (b^{\Phi(n)})^k \ (mod\ n). \tag{15}$$

Pelo Teorema de Euler temos que $b^{\Phi(n)} \equiv 1 \ (mod\ n)$, portanto:

$$b \cdot (b^{\Phi(n)})^k \equiv b \cdot 1^k \equiv b \ (mod\ n) \tag{16}$$

o que completa a demonstração.

## 3.4 CRIPTOGRAFIA AES

O AES é uma cifra de blocos simétrica que foi publicado pelo NIST em 2001. “Comparado a cifras de chave púbica, como o RSA, a estrutura do AES, e da maioria das cifras simétricas, é muito complexa e não pode ser explicada tão facilmente quanto o RSA e os algoritmos semelhantes” (STALLINGS, 2008). O atual AES surgiu como uma proposta realizada pelo NIST, em 1997, da criação de um padrão criptográfico que tivesse um grau de segurança igual ou superior aos sistemas utilizados na época, com eficiência melhorada e, exigiu que a nova cifra deveria trabalhar com blocos de 128 *bits* e suporte para chaves de 128, 192 e 256 *bits*.

Assim, das propostas sugeridas por vários cientistas internacionais, o NIST avaliou e testou todas até escolher, em 2001, o Rijndael como o algoritmo AES. O Rijndael foi desenvolvido por dois cientistas belgas: o Dr. Joan Daemen e Dr. Vincent Rijmen. Para conhecer os métodos de avaliação que o NIST utilizou para selecionar o algoritmo AES, consulte a seção 5.1 Critérios de Avaliação para o AES, na página 92, do livro de Stalling (2008).

Atendendo as especificações do NIST, a proposta do Rijndael para o AES sofre alterações de acordo com o tamanho da chave utilizada, como podemos ver na tabela abaixo:

| **Tamanho da chave (*words/bytes/bits*)** | 4/16/128 | 6/24/192 | 8/32/256 |
|---|---|---|---|
| **Tamanho do bloco de texto claro (*words/bytes/bits*)** | 4/16/128 | 4/16/128 | 4/16/128 |
| **Número de rodadas** | 10 | 12 | 14 |
| **Tamanho da chave da rodada (*words/bytes/bits*)** | 4/16/128 | 4/16/128 | 4/16/128 |
| **Tamanho da chave expandida (*words/bytes/bits*)** | 44/176 | 56/208 | 60/240 |

**Tabela 10: Parâmetros do AES**

A figura 3 apresenta a estrutura de cifragem e decifragem do algoritmo AES para uma chave de 128 *bits*.

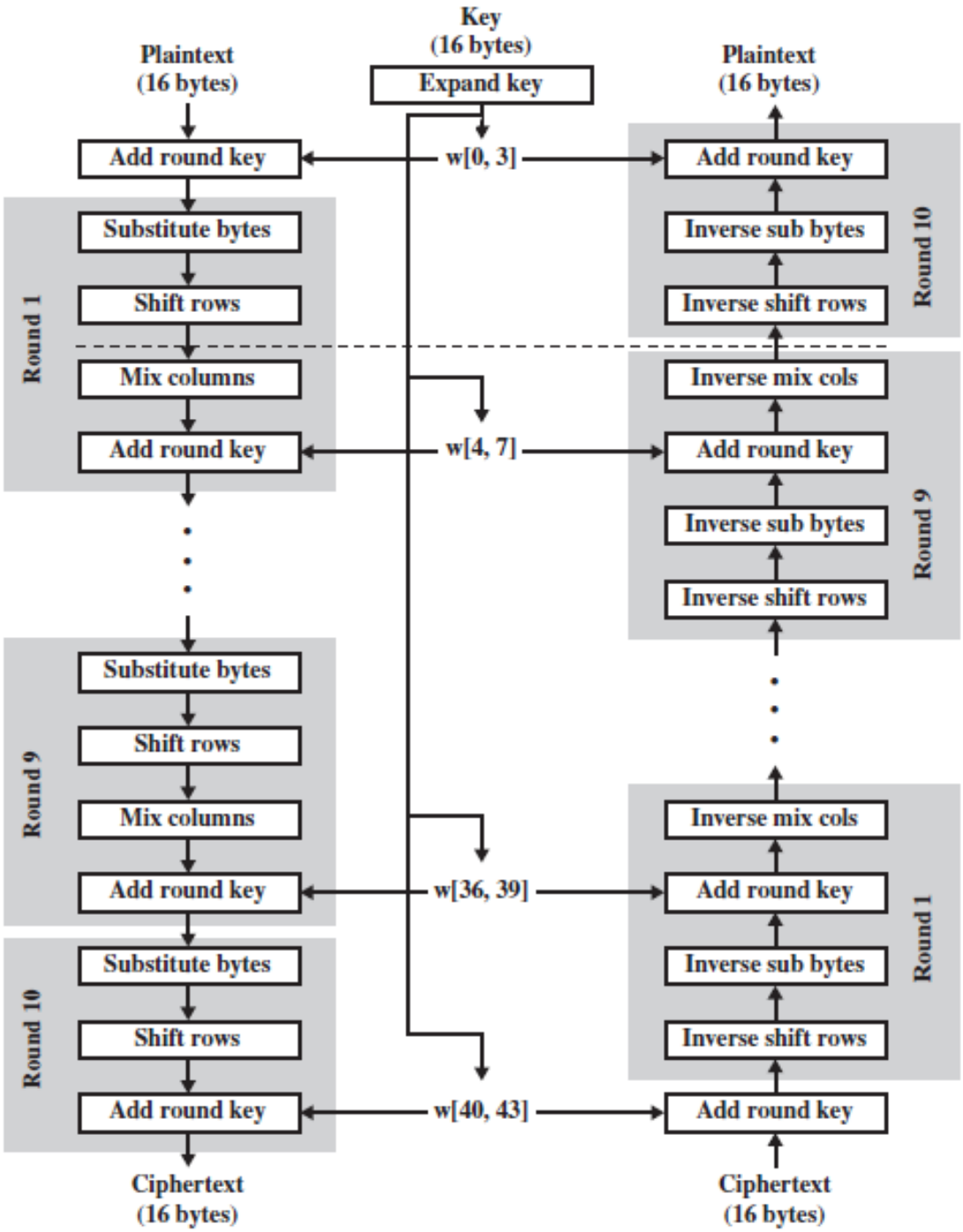


**Figura 3: Etapas AES**

Pode-se perceber que cada rodada é composta pelas seguintes etapas:

- ***Sub Bytes*:** Utiliza a caixa-S para realizar uma substituição *byte* a *byte* do bloco;
- ***Shift Rows*:** Permutação simples de linhas;
- ***Mix Columns*:** Combinação linear que utiliza aritmética sobre $GF(2^8)$;
- ***Add Round Key*:** Um XOR *bit* a *bit* simples do bloco atual com uma parte da chave expandida.

As próximas subseções são dedicadas para a explicação de cada uma das etapas de cada rodada da criptografia AES.

### 3.4.1 *SUB BYTES* E *SUB BYTES* INVERSO

Como pode-se notar pela tabela 10, a entrada dos algoritmos de criptografia é um único bloco de 128 *bits*; este bloco é representado por uma matriz quadrada de $4 \times 4$ *bytes* e é copiado para o vetor *State*, que é modificado a cada etapa da cifra. A transformação de *bytes*, chamada de *Sub Bytes*, é uma busca simples, como um jogo de batalha naval, onde se estabelece uma matriz $16 \times 16$ de valores de *byte* com uma permutação de todos os 256 valores possíveis de 8 *bits* chamada de caixa-S.

Cada *byte* do *State* é mapeado a um novo *byte* de forma que os 4 *bits* mais a esquerda represente um valor de entrada para linha ($x$) e os 4 *bits* a direita representa um valor de entrada para a coluna ($y$) da caixa-S. A caixa-S é construída da seguinte forma:

1. Escreva os elementos da caixa-S em forma crescente, linha por linha. A primeira linha contendo $\{00\}$, $\{01\}$, $\{02\}$, ..., $\{0F\}$; a segunda linha contendo $\{10\}$,$\{11\}$, ..., $\{1F\}$; e assim sucessivamente de forma que o valor de *byte* na linha $x$ e coluna $y$ seja o *byte* $\{xy\}$.
2. Mapeie cada *byte* da caixa-S para seu inverso multiplicativo no corpo finito GF($2^8$); o valor $\{00\}$ é mapeado para si mesmo.
3. Observe que cada *byte* na caixa-S é composto por 8 *bits* rotulados da seguinte maneira $(b_7, b_6, b_5, b_4, b_4, b_3, b_2, b_1, b_0)$ e, aplique a seguinte transformação:

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \tag{17}$$

Existe uma diferença na multiplicação de matrizes dada na expressão acima. Na multiplicação usual de matrizes, cada elemento da matriz resultante é a soma dos produtos dos elementos de uma linha e uma coluna. Para a construção da caixa-S, cada elemento da matriz resultante é um *XOR bit* a *bit* dos produtos dos elementos de uma linha e uma coluna.

| | | *y* | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| *x* | 0 | 63 | 7C | 77 | 7B | F2 | 6B | 6F | C5 | 30 | 01 | 67 | 2B | FE | D7 | AB | 76 |
| | 1 | CA | 82 | C9 | 7D | FA | 59 | 47 | F0 | AD | D4 | A2 | AF | 9C | A4 | 72 | C0 |
| | 2 | B7 | FD | 93 | 26 | 36 | 3F | F7 | CC | 34 | A5 | E5 | F1 | 71 | D8 | 31 | 15 |
| | 3 | 04 | C7 | 23 | C3 | 18 | 96 | 05 | 9A | 07 | 12 | 80 | E2 | EB | 27 | B2 | 75 |
| | 4 | 09 | 83 | 2C | 1A | 1B | 6E | 5A | A0 | 52 | 3B | D6 | B3 | 29 | E3 | 2F | 84 |
| | 5 | 53 | D1 | 00 | ED | 20 | FC | B1 | 5B | 6A | CB | BE | 39 | 4A | 4C | 58 | CF |
| | 6 | D0 | EF | AA | FB | 43 | 4D | 33 | 85 | 45 | F9 | 02 | 7F | 50 | 3C | 9F | A8 |
| | 7 | 51 | A3 | 40 | 8F | 92 | 9D | 38 | F5 | BC | B6 | DA | 21 | 10 | FF | F3 | D2 |
| | 8 | CD | 0C | 13 | EC | 5F | 97 | 44 | 17 | C4 | A7 | 7E | 3D | 64 | 5D | 19 | 73 |
| | 9 | 60 | 81 | 4F | DC | 22 | 2A | 90 | 88 | 46 | EE | B8 | 14 | DE | 5E | 0B | DB |
| | A | E0 | 32 | 3A | 0A | 49 | 06 | 24 | 5C | C2 | D3 | AC | 62 | 91 | 95 | E4 | 79 |
| | B | E7 | C8 | 37 | 6D | 8D | D5 | 4E | A9 | 6C | 56 | F4 | EA | 65 | 7A | AE | 08 |
| | C | BA | 78 | 25 | 2E | 1C | A6 | B4 | C6 | E8 | DD | 74 | 1F | 4B | BD | 8B | 8A |
| | D | 70 | 3E | B5 | 66 | 48 | 03 | F6 | 0E | 61 | 35 | 57 | B9 | 86 | C1 | 1D | 9E |
| | E | E1 | F8 | 98 | 11 | 69 | D9 | 8E | 94 | 9B | 1E | 87 | E9 | CE | 55 | 28 | DF |
| | F | 8C | A1 | 89 | 0D | BF | E6 | 42 | 68 | 41 | 99 | 2D | 0F | B0 | 54 | BB | 16 |

**Figura 4: Caixa-S**

O *Sub Bytes* Inverso é dada por uma caixa-S inversa que é construída através da aplicação inversa da equação 17 e pelo inverso multiplicativo em GF($2^8$). A transformação inversa é dada por:

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{18}$$

Para verificar que o *Sub Byte* Inverso realmente é a operação inversa do *Sub Byte* basta substituir a operação *Sub Byte* na operação inversa, o resultado de tal aplicação deverá ser o vetor de entrada da transormção de *Sub Byte*, ou seja, $(b_7, b_6, b_5, b_4, b_4, b_3, b_2, b_1, b_0)$.

| | | *y* | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| *x* | 0 | 52 | 09 | 6A | D5 | 30 | 36 | A5 | 38 | BF | 40 | A3 | 9E | 81 | F3 | D7 | FB |
| | 1 | 7C | E3 | 39 | 82 | 9B | 2F | FF | 87 | 34 | 8E | 43 | 44 | C4 | DE | E9 | CB |
| | 2 | 54 | 7B | 94 | 32 | A6 | C2 | 23 | 3D | EE | 4C | 95 | 0B | 42 | FA | C3 | 4E |
| | 3 | 08 | 2E | A1 | 66 | 28 | D9 | 24 | B2 | 76 | 5B | A2 | 49 | 6D | 8B | D1 | 25 |
| | 4 | 72 | F8 | F6 | 64 | 86 | 68 | 98 | 16 | D4 | A4 | 5C | CC | 5D | 65 | B6 | 92 |
| | 5 | 6C | 70 | 48 | 50 | FD | ED | B9 | DA | 5E | 15 | 46 | 57 | A7 | 8D | 9D | 84 |
| | 6 | 90 | D8 | AB | 00 | 8C | BC | D3 | 0A | F7 | E4 | 58 | 05 | B8 | B3 | 45 | 06 |
| | 7 | D0 | 2C | 1E | 8F | CA | 3F | 0F | 02 | C1 | AF | BD | 03 | 01 | 13 | 8A | 6B |
| | 8 | 3A | 91 | 11 | 41 | 4F | 67 | DC | EA | 97 | F2 | CF | CE | F0 | B4 | E6 | 73 |
| | 9 | 96 | AC | 74 | 22 | E7 | AD | 35 | 85 | E2 | F9 | 37 | E8 | 1C | 75 | DF | 6E |
| | A | 47 | F1 | 1A | 71 | 1D | 29 | C5 | 89 | 6F | B7 | 62 | 0E | AA | 18 | BE | 1B |
| | B | FC | 56 | 3E | 4B | C6 | D2 | 79 | 20 | 9A | DB | C0 | FE | 78 | CD | 5A | F4 |
| | C | 1F | DD | A8 | 33 | 88 | 07 | C7 | 31 | B1 | 12 | 10 | 59 | 27 | 80 | EC | 5F |
| | D | 60 | 51 | 7F | A9 | 19 | B5 | 4A | 0D | 2D | E5 | 7A | 9F | 93 | C9 | 9C | EF |
| | E | A0 | E0 | 3B | 4D | AE | 2A | F5 | B0 | C8 | EB | BB | 3C | 83 | 53 | 99 | 61 |
| | F | 17 | 2B | 04 | 7E | BA | 77 | D6 | 26 | E1 | 69 | 14 | 63 | 55 | 21 | 0C | 7D |

**Figura 5: Caixa-S Inversa**

### 3.4.2 *SHIFT ROWS* E *SHIFT ROWS* INVERSO

A transformação *ShiftRows* é, literalmente, uma operação de "mudança" nas linhas do *State*, como mostra a figura 6. A primeira linha do *State* segue inalterada; para a segunda, realiza-se um deslocamento "circular" de 1 *byte* para a esquerda; a terceira linha é realizada um deslocamento de 2 *bytes* para a esquerda; e a quarta linha, um deslocamento de 3 *bytes* para a esquerda.

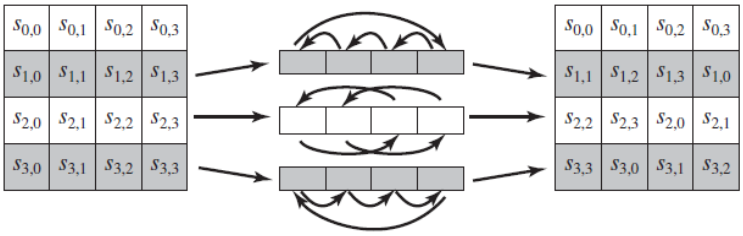


**Figura 6: Operação Shift Rows**

A tranformação inversa do *Shift Rows* segue com um deloscamento inverso de linhas, ou seja, um deslocamento circular em mesma quantidade para direita.

### 3.4.3 *MIX COLUMNS* E *MIX COLUMNS* INVERSO

A transformação *Mix Coluns* é uma operação que trabalha com as colunas do vetor *State*. Cada *byte* é mapeado a um novo valor que é uma combinação linear dos quatro *bytes* daquela coluna. A transformação pode ser resumida na seguinte multiplicação de matrizes:

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}. \quad (19)$$

Os elementos da matriz produto são gerados através de operações de adição e multiplicação realizadas em GF($2^8$).

A transformação inversa é dada pela seguinte multiplicação de matrizes:

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}. \quad (20)$$

Para conferir se a equação 20 é a inversa da equação 19, basta aplicar a 19 na equação 20, ou então, mostrar a seguinte igualdade:

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{21}$$

Vale a pena lembrar, que as operações realizadas com tais matrizes são feitas sobre o corpo do GF($2^8$).

### 3.4.4 *ADD ROUND KEY* E *ADD ROUND KEY* INVERSO

A adição de chaves da rodada é o processo em que os 128 *bits* do *State* passam por um *XOR bit* a *bit* com os 128 *bits* da chave da rodada. O inverso é a mesma operação, já que a operação *XOR* é o seu próprio inverso. Um exemplo de tal operação é dado abaixo:

| | | | |
|---|---|---|---|
| 47 | 40 | A3 | 4C |
| 37 | D4 | 70 | 9F |
| 94 | E4 | 3A | 42 |
| ED | A5 | A6 | BC |

$\oplus$

| | | | |
|---|---|---|---|
| AC | 19 | 28 | 57 |
| 77 | FA | D1 | 5C |
| 66 | DC | 29 | 00 |
| F3 | 21 | 41 | 6A |

$=$

| | | | |
|---|---|---|---|
| EB | 59 | 8B | 1B |
| 40 | 2E | A1 | C3 |
| F2 | 38 | 13 | 42 |
| 1E | 84 | E7 | D6 |

**Figura 7: Operação *Add Round Key***

A segurança garantida por este processo se encontra no processo de expansão de chaves das rodadas explicada na subseção abaixo.

Nota-se que, no processo de cifragem com uma chave de 16 *bytes*, antes de iniciar as etapas, o bloco inicial passa por uma operação *Add Round Key* seguido por nove etapas seguindo as quatro operações descritas acima e, na última etapa, pula a operação *Mix Columns*. O processo de decifragem segue o mesmo padrão da cifragem; no início da decifragem aplica-se a operação de *Add Round Key* e segue nove etapas com a seguinte ordem de operações: *Shift Rows* Inverso, *Sub Bytes* Inverso, *Add Round Key* Inverso e *Mix Columns* Inverso; na última etapa, pula-se a etapa *Mix Columns* Inverso.

### 3.4.5 EXPANSÃO DE CHAVES DO AES

Basicamente, a expansão de chaves toma uma chave de 4 *words* (16 *bytes*) e produz um vetor linear de 44 *words* (176 *bytes*), o que é suficiente para o estágio *Add Round Key* inicial e para todas as próximas 10 rodadas subsequentes.

A chave inicial é copiada para as quatro primeiras *words*. Cada *word* depende da *word*

imediatamente anterior e da *word* quatro posições atrás. Em três a cada quatro casos é utilizado uma simples operação *XOR*; para as *words* que estão nas posições múltiplas de 4 segue uma operação específica designada por:

1. ***Rot Word:*** realiza um deslocamento circular de um *byte* para à esquerda em um word.
2. ***Sub Word:*** substituição de *bytes* usando a caixa-S.
3. Os resultados das etapas acima passam por um *XOR* com a constante da rodada, Rcon[j].

Tal constante é uma *word* no qual os três *bytes* mais a direita são sempre 0, asssim, Rcon[j]=(RC[j],0,0,0), onde os valores em hexadecimal de RC[j] para cada rodada são descritos na tabela abaixo.

| j | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| RC[j] | 01 | 02 | 04 | 08 | 10 | 20 | 40 | 80 | 1B | 36 |

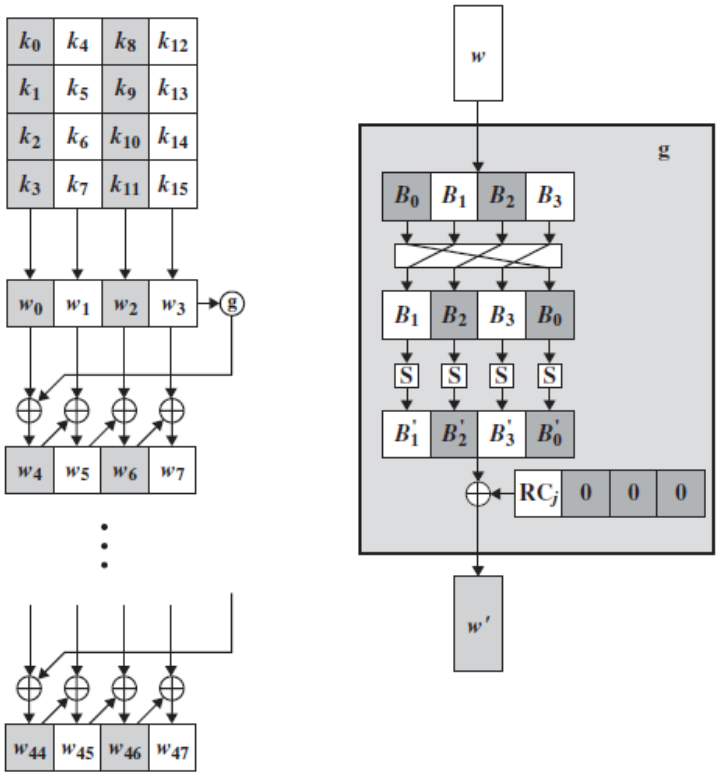


**Figura 8: Expansão de chave da rodada**

### 3.4.6 EXEMPLO DE AES

Tomando como exemplo o seguinte vetor *State*:

$$\begin{bmatrix} 56 & A0 & C3 & 23 \\ 5F & 62 & 91 & EF \\ 02 & 61 & 07 & AC \\ B3 & AC & 45 & 20 \end{bmatrix}$$

e a chave a seguir:

$$\left[\begin{array}{c|c|c|c} D4 & A6 & 01 & 09 \\ C4 & 45 & 87 & 01 \\ 13 & BC & 4D & E1 \\ 0A & 25 & FE & D7 \end{array}\right].$$

Primeiro, expande-se a chave para as demais rodadas seguindo o algorítmo descrito na subseção 3.5.4. Assim, os *words* $w_0, w_1, w_2, w_3$ são dados pela chave inicial e, para o $w_4$ tem-se:

1. **Rot Word**: rotação dos *bytes* do *word* $w_3$, ou seja, (01, E1, D7, 09);
2. **Sub Word**: substituição de bytes seguindo a caixa-S. Assim temos: (7C, F8, 0E, 01)
3. XOR com a primeira constante: $(7C, F8, 0E, 01) \bigoplus (01, 00, 00, 00) = (7D, F8, 0E, 01)$

Após o processo acima, toma-se $w_0$ e aplica-se um XOR com o *word* obtido acima: $(D4, C4, 13, 0A) \bigoplus (7D, F8, 0E, 01) = (A9, 3C, 1D, 0B) = w_4$. Os próximos três *words* são dados por um XOR do *word* anterior com um *word* quatro posições atrás. Assim

- $w_5 = w_4 \bigoplus w_1 = (A9, 3C, 1D, 0B) \bigoplus (A6, 45, BC, 25) = (0F, 79, A1, 2E)$;
- $w_6 = w_5 \bigoplus w_2 = (0F, 79, A1, 2E) \bigoplus (01, 87, 4D, FE) = (00, FE, EC, D0)$
- $w_7 = w_6 \bigoplus w_3 = (00, FE, EC, D0) \bigoplus (09, 01, E1, D7) = (09, FF, 0D, 07)$

Repete-se o processo acima até obter chave para todas as dez rodadas da cifra.

Pela figura 3, antes de começar com as rodadas da cifra, toma-se o vetor *State* e aplica-se um *Add Round Key* antes da primeira rodada com a chave inicial, obtendo a seguinte matriz:

$$\begin{bmatrix} 56 & A0 & C3 & 23 \\ 5F & 62 & 91 & EF \\ 02 & 61 & 07 & AC \\ B3 & AC & 45 & 20 \end{bmatrix} \bigoplus \left[\begin{array}{c|c|c|c} D4 & A6 & 01 & 09 \\ C4 & 45 & 87 & 01 \\ 13 & BC & 4D & E1 \\ 0A & 25 & FE & D7 \end{array}\right] = \begin{bmatrix} 82 & 06 & C2 & 2A \\ 9B & 27 & 16 & E0 \\ 11 & DD & 4A & 4D \\ B9 & 89 & BB & F7 \end{bmatrix}. \tag{22}$$

Assim, esta nova matriz passa por todas as etapas de cada rodada. Para o exemplo aqui apresentado, será realizada apenas a primeira rodada, afim de ilustrar o procedimento.

1. ***Sub Bytes:*** Substituição simples, *byte* a *byte* utilizando a caixa-S.

$$\begin{bmatrix} 82 & 06 & C2 & 2A \\ 9B & 27 & 16 & E0 \\ 11 & DD & 4A & 4D \\ B9 & 89 & BB & F7 \end{bmatrix} \rightarrow \begin{bmatrix} 13 & 6F & 25 & E5 \\ 7D & CC & 47 & E1 \\ 82 & C1 & D6 & E3 \\ 56 & A7 & EA & 68 \end{bmatrix}$$

2. ***Shift Rows:*** Permutação simples das linhas.

$$\begin{bmatrix} 13 & 6F & 25 & E5 \\ 7D & CC & 47 & E1 \\ 82 & C1 & D6 & E3 \\ 56 & A7 & EA & 68 \end{bmatrix} \rightarrow \begin{bmatrix} 13 & 6F & 25 & E5 \\ CC & 47 & E1 & 7D \\ D6 & E3 & 82 & C1 \\ 68 & 56 & A7 & EA \end{bmatrix}.$$

3. ***Mix Colums:*** Combinação linear realizada com as colunas da matriz obtida na rodada anterior realizada em $GF(2^8)$. Assim, pela equação 19 têm-se:

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} 13 & 6F & 25 & E5 \\ CC & 47 & E1 & 7D \\ D6 & E3 & 82 & C1 \\ 68 & 56 & A7 & EA \end{bmatrix} = \begin{bmatrix} D7 & A2 & 57 & 7C \\ 99 & 89 & C2 & AD \\ D0 & 0F & 29 & 24 \\ DF & B9 & 59 & 47 \end{bmatrix}. \tag{23}$$

As operações realizadas neste produto de matrizes são realizadas sobre $GF(2^8)$ onde o produto dos elemento é descrito na seção (2.5.1) e a soma é uma operação de XOR *bit* a *bit* de seus elementos. Tome como exemplo o primeiro elemento da operação. Tem-se que $\{02\}\{13\} \bigoplus \{03\}\{CC\} \bigoplus \{D6\} \bigoplus \{68\} = \{D7\}$; mais precisamente, como $\{02\}\{13\} = (00100110)$ e $\{03\}\{CC\} = (11001100) \bigoplus [(10011000) \bigoplus (00011011)] = (01001111)$:

$$\begin{aligned}
\{02\}\{13\} &= (00100110) \\
\{03\}\{CC\} &= (01001111) \\
\{D6\} &= (11010110) \\
\{68\} &= \underline{(01101000)} \\
& \quad (11010111) = \{D7\}
\end{aligned}$$

4. ***Add Round Key:*** XOR *bit* a *bit* dos elementos da matriz obtida na rodada anterior com parte da chave expandida, finalizando a rodada.

$$\begin{bmatrix} D7 & A2 & 57 & 7C \\ 99 & 89 & C2 & AD \\ D0 & 0F & 29 & 24 \\ DF & B9 & 59 & 47 \end{bmatrix} \oplus \left[\begin{array}{c|c|c|c} A9 & 0F & 00 & 09 \\ 3C & 79 & FE & FF \\ 1D & A1 & EC & 0D \\ 0B & 2E & D0 & 07 \end{array}\right] = \begin{bmatrix} 7E & AD & 57 & 75 \\ A5 & F0 & 3C & 52 \\ CD & AE & C5 & 29 \\ D4 & 97 & 89 & 40 \end{bmatrix}$$

Tais rodadas repetem-se dez vezes e, na décima rodada, pula-se a rodada de *Mix Columns*, como descrito na figura 3.

## 3.5 ELGAMAL

No Crypto'84, Taher ElGamal apresentou uma cifra criptográfica de chave pública que explora o problema do logaritmo discreto. Para um primo $p > 2$ relativamente longo, considera-se um gerador $g$ do conjunto $\mathbb{Z}_p^*$ dos inteiros relativamente primos a $p$. Inicialmente escolhe-se um número inteiro $S$ tal que $1 \leq S \leq p-2$. A chave de criptografia é dada pelo conjunto $K = \{(p,g,S,T) | T = g^S \ mod \ p\}$; note que $S$ é difícil de ser calculado mesmo tendo conhecimento da chave pública $(p,g,T)$ devido ao problema do logaritmo discreto.

Para cada $x \in \mathbb{Z}_p^*$ que queira cifrar, escolhe-se um número aleatório $k \in \mathbb{Z}_{p-1}$. Este número $k$ é denominado *NONCE-Number used ONCE*, ou seja, número usado uma vez. Calcula-se $y = g^k \ mod \ p$ e $z = xT^k \ mod \ p$. O texto $(y,z)$ é enviado para o destinatário que, para decifrar a mensagem, calcula $z(y^S)^{-1} \ mod \ p$, ou seja,

$$\begin{aligned}
z(y^S)^{-1} \ mod \ p &= (xT^k \ mod \ p)[(g^k \ mod \ p)^S]^{-1} \ mod \ p \\
&= [x(g^S \ mod \ p)^k \ mod \ p][g^{Sk} \ mod \ p]^{-1} \\
&= [x(g^{Sk}) \ mod \ p][g^{Sk} \ mod \ p]^{-1} \\
&= x
\end{aligned}$$

.

O algoritmo do ElGamal pode ser generalizado para qualquer grupo finito $G$ com uma operação $\bullet$; tal que o conjunto de chaves $K$ é dada por $K = \{(G,A,S,B)|B = A^S\}$, onde $A \in G$ um gerador do subgrupo $H \subseteq G$ e $S$ um inteiro menor que a ordem de $H$ e maior que 1.

**Exemplo 34.** Considere, para este exemplo, que alguém chamado Paulo queira mandar a frase do exemplo 33 (o exemplo da cifra RSA) para Maria utilizando o algoritmo ElGamal. Maria então escolhe $p = 661$, $g = 40$ e um $S$ tal que $1 < S < 559$, como por exemplo 300. Assim, calcula-se o parâmetro $T$:

$$T = g^S \ mod \ p = 40^{300} \ mod \ 661 = 220$$

e então publica a chave $(p,g,T) = (661,40,220)$ em um meio público de comunicação.

Paulo toma então a mensagem da seguinte forma:

$$242899233022142724289916243114272310229924992230231324$$

e divide em blocos de forma que cada bloco da mensagem seja um número $b < 661$. Para cada bloco, Paulo escolherá um número $k \in \mathbb{Z}_{p-1}$ distinto. Como a divisão dos blocos depende muito do usuário, tomemos a seguinte divisão para tal exemplo:

$$242\text{-}89\text{-}92\text{-}330\text{-}221\text{-}427\text{-}242\text{-}89\text{-}91\text{-}624\text{-}311\text{-}427\text{-}23\text{-}102\text{-}299\text{-}249\text{-}92\text{-}230\text{-}231\text{-}324$$

Paulo escolhe para o primeiro bloco $k = 7$. Assim, ele deverá calcular $y = g^k \ mod \ p$ e $z = xT^k \ mod \ p$ e enviar o par $(y,z)$ para Maria. Vale destacar que o par $(y,z)$ será a mensagem já codificada. Neste caso, o par $(252,433)$ são enviados para Maria.

Maria, com posse do primeiro par do texto cifrado já pode começar a decifrar o texto realizando os seguintes cálculos:

$$\begin{aligned} z(y^S)^{-1} \ mod \ p &= 433(252^{300})^{-1} \ (mod \ 661) \\ &= 433 \cdot 81^{-1} \ (mod \ 661) \\ &= 433 \cdot 457 \ (mod \ 661) \\ &= 242 \end{aligned}$$

Vale lembrar que para cada bloco deve-se tomar um $k \in \mathbb{Z}_{p-1}$ distinto, pois, caso contrário, se tiver dois pares de textos cifrados $(y_1,z_1)$ e $(y_2,z_2)$ calculado com o mesmo $k$, a seguinte igualdade é válida $z_1/z_2 = x_1/x_2$, o que facilitaria o cálculo de $x_2$ se $x_1$ for conhecido.

### 3.5.1 CRIPTOSSISTEMA ELGAMAL SOBRE CURVAS ELÍPTICAS

Para obter um criptossistema com força equivalente ao ElGamal, deve-se procurar um subgrupo cíclico de $E$ no qual o Problema do Logarítmo Discreto seja computacionalmente difícil.

**Exemplo 35.** Considere agora a curva elíptica $E_1 : y^2 = x^3 + 3x + 6$ sobre $Z_{13}$. Maria então escolhe sua chave secreta $S = 6$ e envia à Paulo a chave pública $\{(3,4);(5,4)\}$. Para enviar um texto criptografado para Maria, Paulo escolhe um $0 \leq k \leq 12$, e calcula $\{y = k(3,4); z = x + k(5,4)\}$. Para decriptografar, Maria calcula $z - S \cdot y$. De fato, pois $z - S \cdot y = x + k(5,4) - 6k(3,4) = x + k(5,4) - k(5,4) = x$, pois $6(3,4) = (5,4)$ como denota na tabela 6.

# 4 DA CONTRIBUIÇÃO DOS ELEMENTOS ALGÉBRICOS

Das cifras apresentadas neste trabalho, têm-se a Cifra de César e a Cifra de Hill que trabalham com o conjunto $\mathbb{Z}_{26}$, ou seja, utilizam da aritmética de congruências modulares dos inteiros como base para a cifragem e decifragem de um texto claro dado. Têm-se pelo teorema 4, que os algarismos que são múltiplos de 2 e 13 não tem inverso multiplicativo em $\mathbb{Z}_{26}$, o que faz com que tal conjunto seja classificado como um anel.

Na Cifra de César pode-se observar, que o algoritmo é dado somente no grupo aditivo de $\mathbb{Z}_{26}$, o que o torna muito frágil devido à facilidade de se trabalhar com a operação de soma. Por isso que, em métodos mais recentes, observa-se que o uso de grupos para definir um algoritmo criptográfico é raro, se este for trabalhado individualmente. A fragilidade do método de César também é dada pela quantidade de chaves que podem ser utilizadas (26) para o processo de cifragem (para maiores detalhes, vide a obra de Faleiros, 2011).

Um bom uso de grupos aditivos encontra-se na Cifra de Hill onde se trabalha com o anel $\mathbb{Z}_{26}$ sobre as operações de adição e multiplicação. O uso de propriedades de tal anel nos permite trabalhar com transformações lineares facilmente, de forma que a segurança do algoritmo seja maior devido à complexidade dos cálculos sob tal conjunto algébrico. Apesar do método ser mais seguro, temos (como já foi apresentado) que tal cifra conserva algumas propriedades da linguagem utilizada, por ser um método de substituição termo-a-termo, não sendo útil a utilização de tal cifra em textos extensos.

Superando tal falha presente na Cifra de Hill, a Criptografia RSA embaralha o texto claro de forma que seja impossível obter padrões linguísticos em seus textos cifrados. O RSA ao trabalhar com o anel $\mathbb{Z}_n$, oferta em seu processo de cifragem dados que não são encontrados no alfabeto original. Têm-se também, que a segurança do método é dada pela dificuldade em fatorar o número $n$ e, se não, também pelo fato de se tentar encontrar o valor de cada bloco $b$ através da forma reduzida de $b^e$ no módulo $n$ sem tentar encontrar $d$ o que, segundo Coutinho (2003), ninguém conseguiu desenvolver um método para tal façanha. Assim, quebrar a cifra RSA é sinônimo de fatorar o número $n$, o que é impraticável para $p$ e $q$ suficientemente grandes

e distantes.

O ElGamal trabalha de forma semelhante à criptografia RSA e explora a dificuldade computacional dada pelo Problema do Logaritmo Discreto. O fato de se trabalhar com um par ilegível de texto, dificulta ainda mais a criptoanálise sobre textos cifrados utilizando tal método. Porém, o fato de ter, obrigatoriamente, que trabalhar com subconjuntos cíclicos de um dado conjunto pode restringir o tamanho do texto no qual quer cifrar, sendo necessário operar com conjuntos muito grandes ou até mesmo, conjunto infinitos.

Já o AES, explora as propriedades bem definidas do conjunto $GF(2^n)$ e as operações com *bits/bytes*. A complexidade do algoritmo se dá exclusivamente pela forma em que o texto é trabalhado em cada uma das operações e, devido à exaustiva aplicação de várias rodadas repetidas, tornanando praticamente imune aos ataques criptoanalíticos. Apesar de não existirem ataques de segurança conhecidos ao *Rinjdael*, existe uma certa crítica à sua estrutura matemática "simples" que pode ser sucetível à ataques (Stallings, 2008).

# 5 CONCLUSÕES

Como pode-se notar neste trabalho, a Álgebra, uma das áreas mais abstratas da Matemática, tem importante papel na interpretação e estudo dos processos criptográficos. Com o passar do tempo, as cifras se tornaram cada vez mais complexas devido a quantidade de algoritmos matemáticos utilizados nos mesmos. Como destacado neste trabalho, pode-se tomar a Cifra de César e a *Rinjdael* para ilustrar tal mudança; a primeira trabalhava com substituições simples que poderiam ser interpretadas apenas como um processo de congruência modular; a segunda já trabalha com vários algoritmos matemáticos e computacionais para garantir melhor segurança na troca de informações.

O leitor pôde notar que os processos criptográficos, em maior parte, não trabalham com conteúdos matemáticos avançados e sim, com conteúdos básicos aplicados de maneira inteligente. Atualmente, o uso inteligente da matemática básica tem auxiliado o homem a compreender e desenvolver mais formas de garantir segurança na troca de informações pois, com o atual avanço tecnológico, se faz necessário garanatir a segurança de dados dispostos em rede pública de comunicação como, por exemplo, a internet.

Tal trabalho não apresenta todas as cifras conhecidas pelo autor como, por exemplo, as cifras mecânicas dadas pelas máquinas Enigma e Colossus utilizadas na segunda guerra mundial ou então a cifra DES (*Data Encryption Standard*), para que o mesmo não fique muito extenso porém, caso o leitor tenha curiosidade pode-se consultar a bibliografia utilizada para tal fim e/ou assistir documentários ou filmes que abordam tal conteúdo, como o recente filme ”*The Imitation Game*”que conta a hitória de Allan Turing e da criação da Colossus.

# REFERÊNCIAS

COELHO, Sônia P.; MILIES, Francisco C. P. **Números:** Uma Introdução à Matemática. 3 ed. 2 reimpr. São Paulo: Editora da Universidade de São Paulo, 2006.

COUTINHO, Severino C. **Números Inteiros e Criptografia RSA**. Rio de Janeiro: IMPA, 2003.

DOMINGUES, Hygino H.; IEZZI, Gelson.**Álgebra Moderna:** volume único. 4 ed. reform. São Paulo: Atual, 2003.

FALEIROS, Antonio C. **Criptografia**. Notas em Matemática Aplicada; v. 52. São Carlos (SP): SBMAC, 2011.

LIDL, Rudolf; NIEDERREITER, Harald. **Finite Fields:** Encyclopedia of Mathematics and its Applications. vol. 20. Cambrige: Cambrige University press, 1997.

MIZRAHI, Victorine V. **Treinamento em Linguagem C**. São Paulo: Pearson Prentice Hall, 2008.

SANTOS, José P. O. **Introdução à Teoria de Números**. 3 ed. Rio de Janeiro: IMPA, 2014.

STALLINGS, William. **Criptografia e Segurança de Redes:** Princípios e Práticas. 4 ed. São Paulo: Pearson Prentice Hall, 2008.

TERADA, Routo. **Segurança de Dados:** criptografia em redes de computadores. 2ª edição revisada e ampliada. São Paulo: Blucher, 2008.