unesp

UNIVERSIDADE ESTADUAL PAULISTA

PROGRAMA DE PÓS-GRADUAÇÃO EM MATEMÁTICA

Números Inteiros e Criptografia RSA

Marco Antonio Fávaro Teixeira

INSTITUTO DE GEOCIÊNCIAS E CIÊNCIAS EXATAS

RIO CLARO – SP
2020

Universidade Estadual Paulista "Júlio de Mesquita Filho"
Instituto de Geociências e Ciências Exatas
Câmpus de Rio Claro

Números Inteiros e Criptografia RSA

Marco Antonio Fávaro Teixeira

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Matemática, junto ao Programa de Pós-Graduação em Matemática, mestrado profissional, do Instituto de Geociências e Ciências Exatas da Universidade Estadual Paulista "Júlio de Mesquita Filho", Câmpus de Rio Claro.

Orientadora
Profa. Dra. Thaís Fernanda Mendes Monis

Rio Claro
2020

T266n

Teixeira, Marco Antonio Fávaro
Números inteiros e criptografia RSA / Marco Antonio Fávaro Teixeira. -- Rio Claro, 2020
72 p. : il., tabs.

Dissertação (mestrado) - Universidade Estadual Paulista (Unesp), Instituto de Geociências e Ciências Exatas, Rio Claro
Orientadora: Thaís Fernanda Mendes Monis

1. Teoria dos Números. 2. Números Primos. 3. Teorema de Fermat. 4. Criptografia RSA. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca do Instituto de Geociências e Ciências Exatas, Rio Claro. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

TERMO DE APROVAÇÃO

Marco Antonio Fávaro Teixeira

Números Inteiros e Criptografia RSA

Dissertação APROVADA como requisito parcial para a obtenção do grau de Mestre no Curso de Pós-Graduação em Matemática, mestrado profissional, do Instituto de Geociências e Ciências Exatas da Universidade Estadual Paulista "Júlio de Mesquita Filho", pela seguinte banca examinadora:

Profa. Dra. Thaís Fernanda Mendes Monis
Orientadora

Prof. Dr. Marcio Andrey Teixeira
IFSP / Catanduva-SP

Profa. Dra. Carina Alves Severo
IGCE / UNESP / Rio Claro-SP

Rio Claro, 7 de dezembro de 2020

Dedico à minha esposa Wanessa e meu filho Luiz Felipe.
Aos meus pais, Waine e Tereza.
Meus irmãos Junior e Chrys.

AGRADECIMENTOS

A Deus pela proteção de cada dia. A minha esposa Wanessa pela compreensão e paciência durante o tempo em que estive ausente e ao meu filho Luiz Felipe que sempre ficava acordado até tarde da noite aguardando o meu retorno das viagens de Rio Claro. Aos meus pais e irmãos pelo incentivo para que eu pudesse realizar este trabalho. À minha orientadora, professora Dra. Thaís Fernanda Mendes Monis, que aceitou a orientação deste trabalho e pelo apoio em compartilhar conhecimentos. Aos membros da Banca de Qualificação, professor Dr. Marcio Andrey Teixeira e a professora Dra. Carina Alves, a minha profunda gratidão pelas contribuições para a melhoria deste trabalho. Agradeço ao Instituto Federal de São Paulo, Câmpus Catanduva, pelo apoio à realização desse mestrado e uma melhor qualificação profissional. Enfim, a todos os amigos e amigas que de alguma forma participaram direta ou indiretamente de todo o processo deste trabalho.

A Álgebra é generosa... ela, frequentemente, dá mais do que aquilo que lhe é pedido.
Jean D'Alembert

Resumo

Este trabalho tem por objetivo apresentar a Criptografia RSA, uma das ferramentas mais utilizadas para transmitir informações seguras pela internet. Iniciamos com uma breve história da criptografia desde sua origem até o surgimento do método RSA. Na sequência, apresentamos a matemática na qual o método se fundamenta, ou seja, a teoria dos números atribuída aos antigos gregos e as contribuições dos matemáticos Fermat, Euler e Gauss. Descrevemos através de um exemplo o sistema de criptografia de chave pública RSA, provando o porquê do método funcionar, a questão da segurança, bem como os impactos provenientes com o surgimento dos computadores quânticos. Por fim, mostramos a assinatura digital, uma das aplicações oriundas dos algoritmos de criptografia de chave pública e os certificados digitais.

Palavras-chave: Teoria dos Números, Números Primos, Teorema de Fermat, Criptografia RSA.

Abstract

This work aims to present RSA Cryptography, one of the most used tools to transmit safe information over the internet. We begin with a brief history of cryptography from its origin to the emergence of the RSA method. In the sequence, we present the Mathematics on which the method is based, that is, the number theory attributed to the ancient Greeks and the contributions of the mathematicians Fermat, Euler and Gauss. We describe through an example the RSA public-key cryptography system, proving why the method works, the security issue as well as the impacts arising from the emergence of quantum computers. Finally, we show the digital signature, one of the applications coming from public-key cryptography algorithms and digital certificates.

Keywords: Number Theory, Prime Numbers, Fermat's Theorem, RSA Cryptography.

Lista de Figuras

Lista de Tabelas

Lista de Siglas

AC	Autoridade Certificadora
AES	Advanced Encryption Standard
AR	Autoridade de Registro
CRL	Certificate Revocation List
DES	Data Encryption Standard
DSA	Digital Signature Algorithm
DSS	Digital Signature Standard
ICP	Infraestrutura de Chaves Públicas
ICP-Brasil	Infraestrutura de Chaves Públicas Brasileira
ITI	Instituto Nacional de Tecnologia da Informação
ITU	International Telecommunications Union
LCR	Lista de Certificados Revogados
MD	Message Digest
MDC	Máximo Divisor Comum
MIT	Massachussets Institute of Technology
NIST	National Institute of Standards and Technology
NSA	National Security Agency
PKI	Public Key Infrastructure
RSA	Rivest-Shamir-Addleman
SHA	Secure Hash Algorithm

Sumário

Introdução

Nos dias atuais, vivemos constantemente ligados à tecnologia, seja por computadores ou até mesmo pelos celulares, o que torna imprescindível garantir a segurança das informações nesse processo da comunicação digital.

Há relatos históricos de que o homem desde a antiguidade buscava manter o sigilo e a segurança de diversos tipos de informações como mensagens particulares ou informações de segurança nacional, de modo que estas estivessem a salvo de mãos inimigas. Mais tarde, com o advento dos computadores, surge também formas diferentes de armazenamento e comunicação das informações deixando evidente a necessidade de protegê-los [1].

Nesse sentido, a criptografia ganha mais importância para que o fluxo das transações comerciais, bancárias, militares ou pessoais pudessem ser mantidas em sigilo de modo que nenhum interceptor tivesse condições de interpretar informações sigilosas ou até mesmo alterá-las, caso os dados referentes ao fluxo de transações fossem capturados.

A matemática é a principal ferramenta no processo de codificação para que as mensagens possam ser encaminhadas em meios eletrônicos. A partir da década de 60, a Teoria dos Números, uma das áreas mais antigas da matemática, muito utilizada nas aplicações à criptografia, passou a desempenhar papel fundamental num específico sistema de criptografia assimétrica chamado RSA (Rivest-Shamir-Addleman) [2].

Desse modo, com a evolução da criptografia assimétrica, foi possível resolver questões como a distribuição de chaves até então um dos principais problemas associados a criptografia simétrica, o que possibilitou a criação das assinaturas digitais, sendo um dos mecanismos mais utilizados no meio computacional com a finalidade de garantir a autenticidade nas transações eletrônicas por meio da certificação digital.

A fim de proporcionar uma visão geral dos estudos relacionados a este trabalho, apresentamos de forma sucinta os assuntos que iremos abordar em cada capítulo. O presente trabalho é composto por seis capítulos.

No capítulo 1, *Breve História da Criptografia*, abordamos uma breve discussão sobre a história da criptografia desde a sua origem, com os diferentes tipos de criptossistemas. Destacamos a criptografia assimétrica que possibilitou o surgimento do mais famoso sistema de criptografia assimétrica, o RSA.

No capítulo 2, *Aritmética dos Números Inteiros*, abordamos a matemática necessária para o entendimento e a compreensão dos métodos de criptografias e as assinatura digitais. Toma-se como base os estudos fundamentados na teoria dos números, com destaque para o Pequeno Teorema de Fermat, que serviu de base para a demonstração do funcionamento do método RSA e também para a criptografia de ElGamal e a Assinatura Digital DSA *(Digital Signature Algorithm)*.

No capítulo 3, *Sistemas de Criptografia*, apresentamos os detalhes do funcionamento

dos sistemas de criptografia simétrica e assimétrica bem como suas vantagens e desvantagens na sua aplicabilidade. Apresentamos também o primeiro algoritmo de chave pública, o protocolo Diffie-Hellman, que serviu de base para a Criptografia ElGamal no qual é muito utilizado nos sistemas de assinaturas digitais.

No capítulo 4, *A Criptografia RSA*, descrevemos todos os passos necessários para a compreensão desse método. Apresentamos os fundamentos da técnica, evidenciando a sua segurança. Apresentamos também os impactos da Criptografia Quântica diante dos sistemas criptográficos que tem por base a intratabilidade da fatoração de números inteiros em produto de números primos e o problema do logaritmo discreto.

No capítulo 5, *Assinatura e Certificado Digital*, mostramos que os certificados digitais além de estarem associados às assinaturas digitais, são mecanismos criptográficos indispensáveis para segurança das informações e fundamentais diante do isolamento social em que empresas e pessoas, durante o período de pandemia causado pelo novo coronavírus (Sars-CoV-2), estão vivenciando na pandemia mundial.

Por fim, o capítulo 6, *Considerações Finais*, é dedicado à conclusão do trabalho e previsão para novos métodos referentes à criptografia pós-quântica.

1 Breve História da Criptografia

Etimologicamente, o termo "criptografia" se originou a partir das raízes gregas kryptos, que significa secreto e graphos, que significa escrita. Portanto, criptografia é a área do conhecimento que desenvolve métodos para codificar mensagens [1].

Há muitos séculos atrás, reis, rainhas e generais tentavam de alguma forma manter segura a troca de informações, a fim de que pudessem governar seus países e comandar seus exércitos de maneira segura, de modo que ninguém fosse capaz de interceptá-las [3].

Nesse contexto, surge a necessidade de se inventar códigos e cifras, ou seja, técnicas para disfarçar uma mensagem de modo que somente o destinatário pudesse lê-la. Nasce, assim, a criptografia.

Nesse processo de desenvolvimento de códigos secretos, surgia paralelamente à criptografia, a criptoanálise, uma área do conhecimento que trata do ato de decifrar ou "quebrar" o sistema criptográfico. As pessoas que desvendavam as mensagens cifradas ou codificadas eram chamadas de "quebra-códigos", que tinham o objetivo de descobrir o significado da mensagem ou até mesmo modificá-la antes que seu destinatário pudesse recebê-la. Juntas, a criptografia e a criptoanálise, formam a criptologia [1].

Alguns dos primeiros relatos de escrita secreta datam da época de Heródoto - "o pai da história". Segundo Heródoto, um grego precisava transmitir uma mensagem secretamente. Para isso, ele raspa o cabelo do mensageiro, tatua a mensagem na cabeça dele e espera o cabelo crescer. Quando chegasse ao destinatário, a mensagem seria revelada raspando novamente a cabeça do mensageiro. Um outro exemplo aconteceu com um grego que vivia em solo persa e, sabendo de um possível ataque do rei Xerses para conquistar a Grécia, avisou os gregos através de uma escrita em tábuas de madeira e cobertas em cera, de modo que os guardas não pudessem interceptá-la, salvando assim a Grécia de ser conquistada pelos persas [3].

Este tipo de técnica é conhecida como Esteganografia. Observa-se, portanto, que a mensagem que deveria ser transmitida precisou ser camuflada, escondida para que ninguém soubesse seu verdadeiro conteúdo.

A história mostra que a esteganografria apresentava certo grau de segurança, todavia, se a mensagem fosse descoberta, todo o conteúdo poderia ser revelado [3].

Portanto, observa-se que a criptografia e a esteganografia buscam ocultar a mensagem de maneiras diferentes, ou seja, enquanto a primeira oculta o significado da mensagem, a segunda oculta a existência dessa.

Com o passar dos anos, os estudos referentes às técnicas criptográficas foram se tornando cada vez mais eficientes. Em 50 a.C, o general romano Júlio César enviou mensagens cifradas para se comunicar com seus generais nas batalhas, utilizando sua famosa cifra de substituição representada pela Tabela 1.1 a seguir:

Tabela 1.1: Cifra de César

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

Fonte: Elaborada pelo autor.

O método da cifra de substituição fazia com que uma letra do alfabeto fosse trocada por uma outra a três posições a sua frente. Apesar da sua simplicidade este parece ter sido o primeiro exemplo de um código secreto de que se tem notícia [4].

Estudos na área da criptografia passaram a ficar cada vez mais importantes, o que fez com que no mundo todo uma grande quantidade de pesquisadores das mais diversas áreas como a matemática, a engenheira, a ciência da computação, entre outras, pudessem se lançar na descoberta de novas técnicas criptográficas.

Diante da busca por novas tecnologias, o matemático britânico Alan Turing passa a desempenhar um papel fundamental na história da criptografia graças ao surgimento dos primeiros computadores. Conhecido também por ser um dos fundadores da ciência da computação, ele foi o principal responsável por decifrar as mensagens secretas do exército alemão durante a 2ª Guerra Mundial [4].

Mesmo sabendo que o mundo cada vez mais tornava-se refém dos avanços tecnológicos, a proliferação do uso dos computadores trouxe consigo uma preocupação ainda maior com a segurança dos dados e informações, principalmente a partir do surgimento da Internet.

O grande destaque de todo o desenvolvimento da criptografia ocorreu em 1976, quando Diffie e Hellman publicaram o artigo *New Directions in Cryptography*, introduzindo um novo conceito de criptografia de chave pública. Até então, a criptografia era de chave simétrica, ou seja, os processos de codificação e decodificação eram feitos com uma única chave de modo que tanto o remetente quanto o destinatário partilhavam da mesma chave, diferentemente da assimétrica onde são usadas duas chaves: uma pública, que é de conhecimento de todos, e uma privada. Um exemplo de chave simétrica é o Código de César, no qual saber codificar implica em saber decodificar. Na criptografia de chave pública, decodificar uma mensagem codificada não é uma tarefa trivial. Embora os autores não tivessem apresentado uma solução para esse novo conceito de criptografia, o artigo despertou na comunidade criptográfica um grande interesse e, em 1978, Rivest, Shamir e Adleman, que na época trabalhavam no Massachussets Institute of Technology (MIT), descobriram um esquema prático de chave pública, que ficou conhecido como RSA [4], [5] e [6]. As letras RSA correspondem as iniciais dos sobrenomes dos seus inventores, conforme mencionado na Introdução.

Antes de apresentarmos os tipos de sistemas de criptografia e o funcionamento do método RSA, iremos desenvolver no próximo capítulo a matemática necessária para que possamos compreendê-los.

2 Aritmética dos Números Inteiros

Neste capítulo, estudaremos as principais propriedades do conjunto dos números inteiros que servirão de base não somente para o entendimento dos sistemas de criptografia Diffie-Hellman, ElGamal e RSA, como também para a compreensão dos métodos de assinaturas digitais ElGamal e DSA.

2.1 Divisibilidade em $\mathbb{Z}$

Definição 2.1.1. [7] Se a e b são inteiros, dizemos que a divide b, denotando por $a \mid b$, se existir um inteiro c tal que $b = ac$. Nesse caso, diremos também que a é um divisor ou fator de b; ou ainda, que b é um múltiplo de a ou que b é divisível por a. Se a não divide b, escrevemos $a \nmid b$.

Proposição 2.1.1. [7] Se a, b e c são inteiros, $a \mid b$ e $b \mid c$, então $a \mid c$.

Demonstração. De fato, por hipótese temos que $a \mid b$ e $b \mid c$. Pela Definição 2.1.1, existem inteiros k_1 e k_2 tais que $b = k_1 a$ e $c = k_2 b$. Substituindo o valor de b na equação $c = k_2 b$, teremos $c = k_2(k_1 a) = (k_2 k_1)a$. Como $(k_2 k_1)$ é produto de dois números inteiros, então podemos denotá-lo por $k_3 \in \mathbb{Z}$, ou seja, $c = k_3 a$, com $k_3 \in \mathbb{Z}$, o que implica que $a \mid c$. □

Proposição 2.1.2. [7] Se a, b, c, m e n são inteiros, $a \mid b$ e $a \mid c$, então $a \mid (bm + cn)$.

Demonstração. Por hipótese, $a \mid b$ e $a \mid c$ então $b = ak_1$ e $c = ak_2$, para algum inteiro k_1, k_2. Multiplicando-se estas duas equações, respectivamente por m e n, teremos $bm = ak_1 m$ e $cn = ak_2 n$. Somando-se membro a membro, obtemos $bm + cn = ak_1 m + ak_2 n = (k_1 m + k_2 n)a$, o que nos diz que $a \mid (bm + cn)$. □

Antes de introduzirmos o algoritmo da divisão, vamos enunciar o chamado Teorema de Eudoxius [7]: dados a e b inteiros com $b \neq 0$ então a é um múltiplo de b ou se encontra entre dois múltiplos consecutivos de b, isto é, correspondendo a cada par de inteiros a e $b \neq 0$ existe um inteiro q tal que, para $b > 0$,

$$bq \leq a < b(q + 1)$$

e para $b < 0$,

$$bq \leq a < b(q - 1).$$

Teorema 2.1.1. (Algoritmo da Divisão)[7] Dados dois inteiros a e b, $b > 0$, existe um único par de inteiros q (quociente) e r (resto) tal que

$$a = bq + r, \text{ com } 0 \leq r < b. \quad (r = 0 \Leftrightarrow b \mid a)$$

Demonstração. Dividiremos a demonstração em duas partes:

1. Existência: Por hipótese, b é um número inteiro estritamente positivo. Consideremos a um inteiro com as seguintes possibilidades:

   (i) a é múltiplo de b e, portanto, $a = bq$ para um conveniente inteiro q.

   (ii) a não é um múltiplo de b. Nesse caso, pelo teorema de Eudoxius, a está situado entre dois múltiplos consecutivos de b, isto é, existe um inteiro q tal que $bq \leq a < b(q+1)$. Somando $-bq$ na desigualdade anterior obtemos $0 \leq a - bq < b$. Desta forma, se definirmos $r = a - bq$, o que equivale a $a = bq + r$, em que $0 \leq r < b$, teremos, garantida, a existência de q e r.

   Evidentemente, $r = 0$ corresponde ao caso em que a é múltiplo de b.

2. Unicidade: Vamos supor que fosse possível determinar outro par de inteiros, q_1 e r_1, tais que $a = bq_1 + r_1$, com $0 \leq r_1 < b$. Mostraremos que $q = q_1$ e $r = r_1$. Lembrando que r e r_1 são inteiros, logo um deles é maior ou igual ao outro. Para isso, iremos considerar que $r \geq r_1$. Como $a = bq + r$ e $a = bq_1 + r_1$, subtraindo uma da outra
$$r - r_1 = (a - bq) - (a - bq_1) = b(q_1 - q).$$
Por outro lado, tanto r quanto r_1 são menores que b. Como estamos supondo que $r \geq r_1$, obtemos $0 \leqslant r - r_1 < b$. Comparando com $r - r_1 = b(q_1 - q)$ concluímos que
$$0 \leqslant b(q_1 - q) < b.$$
Mas b é um inteiro positivo e pode ser cancelado, logo $0 \leqslant q_1 - q < 1$. Como $q_1 - q$ é número inteiro, essa desigualdade só se verifica quando $q_1 - q = 0$, ou seja, $q = q_1$. Disto segue imediatamente que $r = r_1$.

□

2.2 Máximo Divisor Comum e Números Primos

Definição 2.2.1. [8] Sejam a e b dois números inteiros. Um elemento $d \in \mathbb{Z}$ se diz máximo divisor comum de a e b se cumpre as seguintes condições:

1. $d \geq 0$.
2. $d \mid a$ e $d \mid b$.
3. Se d' é um inteiro tal que $d' \mid a$ e $d' \mid b$, então $d' \mid d$ (ou seja, todo divisor comum a a e b também é divisor de d).

Notação: $MDC(a, b) = d$.

Teorema 2.2.1. (Identidade de Bézout)[7] Para quaisquer inteiros a e b, existem inteiros n_0 e m_0 tais que $d = an_0 + bm_0$, onde d é o máximo divisor comum de a e b.

Demonstração. Considere o conjunto $B = \{na + mb \,|\, n, m \in \mathbb{Z}\}$. Note que B é o conjunto de todas as combinações lineares inteiras de a e de b, e contém números negativos, positivos e também o zero. Vamos escolher n_0 e m_0 tais que $c = n_0a + m_0b$ seja o menor inteiro positivo pertencente ao conjunto B. Vamos provar que $c \mid a$ e $c \mid b$. Como as demonstrações são similares, mostraremos apenas que $c \mid a$. A prova é por contradição. Suponhamos que $c \nmid a$. Neste caso, pelo Teorema 2.1.1, existem q e r tais que $a = qc + r$ com $0 < r < c$. Portanto, $r = a - qc = a - q(n_0a + m_0b) = (1 - qn_0)a + (-qm_0)b$. Isto mostra que $r \in B$, pois $(1 - qn_0)$ e $(-qm_0)$ são inteiros, o que é uma contradição, uma vez que $0 < r < c$ e c é o menor elemento positivo de B. Logo $c \mid a$ e de forma análoga se prova que $c \mid b$.

Como d é um divisor comum de a e b, existem inteiros k_1 e k_2 tais que $a = k_1d$ e $b = k_2d$ e, portanto, $c = n_0a + m_0b = n_0k_1d + m_0k_2d = d(n_0k_1 + m_0k_2)$ o que implica $d \mid c$. Temos que $d \leqslant c$ (ambos são positivos) e como $d < c$ não é possível, uma vez que d é o máximo divisor comum, concluímos que $d = n_0a + m_0b$. □

Lema 2.2.2. (Lema de Euclides)[2] Se a e b são inteiros e $a = bq + r$ onde q e r são inteiros, então $MDC(a, b) = MDC(b, r)$.

Demonstração. Considere $d = MDC(a, b)$ e $c = MDC(b, r)$. Queremos provar que $d = c$. Provaremos isto em duas etapas, verificando que $d \leqslant c$ e em seguida que $c \leqslant d$. Destas duas desigualdades segue a igualdade desejada. Provaremos que $d \leqslant c$ e a outra desigualdade é provada de maneira análoga. Como $d = MDC(a, b)$, então $d \mid a$ e $d \mid b$, ou seja, existem inteiros k_1 e k_2 tais que $a = dk_1$ e $b = dk_2$. Substituindo em $a = bq + r$, obteremos $dk_1 = dk_2q + r$, ou seja

$$r = dk_1 - dk_2q = d(k_1 - k_2q).$$

Mas isto significa que $d \mid r$. Como $d = MDC(a, b)$, temos em particular que $d \mid b$. Mostramos que $d \mid r$. Portanto, d é um divisor comum entre b e r. Mas c é o maior divisor comum entre b e r. Logo $d \leqslant c$. □

Teorema 2.2.3. (O Algoritmo de Euclides)[2] Sejam a e b números inteiros não negativos. Se o algoritmo da divisão for aplicado diversas vezes, então o último resto não nulo será o $MDC(a, b)$.

Demonstração. Aplicando o algoritmo da divisão a a e b e supondo que o resto nulo ocorre após n divisões, temos

$$\begin{array}{ccc}
a = bq_1 + r_1 & \text{e} & 0 \leqslant r_1 < b \\
b = r_1q_2 + r_2 & \text{e} & 0 \leqslant r_2 < r_1 \\
r_1 = r_2q_3 + r_3 & \text{e} & 0 \leqslant r_3 < r_2 \\
r_2 = r_3q_4 + r_4 & \text{e} & 0 \leqslant r_4 < r_3 \\
\vdots & & \vdots \\
r_{n-4} = r_{n-3}q_{n-2} + r_{n-2} & \text{e} & 0 \leqslant r_{n-2} < r_{n-3} \\
r_{n-3} = r_{n-2}q_{n-1} + r_{n-1} & \text{e} & 0 \leqslant r_{n-1} < r_{n-2} \\
r_{n-2} = r_{n-1}q_n & \text{e} & r_n = 0.
\end{array}$$

Da última divisão temos que $r_{n-1} \mid r_{n-2}$. Logo o maior divisor comum entre os dois é o próprio r_{n-1}. Portanto, $MDC(r_{n-2}, r_{n-1}) = r_{n-1}$. Pelo Lema 2.2.2, aplicando-o à penúltima divisão, concluímos que

$$MDC(r_{n-3}, r_{n-2}) = MDC(r_{n-2}, r_{n-1}) = r_{n-1}.$$

Continuando assim, coluna acima, chegamos finalmente a $MDC(a,b) = r_{n-1}$. □

O dispositivo prático para expressar o Algoritmo de Euclides de tal modo que o MDC entre a e b seja representado como uma combinação linear desses dois números é denominado de **Algoritmo de Euclides Estendido** [2]. Portanto, é possível encontrar n_0 e m_0 tal que

$$a \cdot n_0 + b \cdot m_0 = d.$$

Nesse caso, considere a seguinte sequência de divisões após o cálculo do $MDC(a,b)$:

$$\begin{array}{ccc} a = bq_1 + r_1 & \text{e} & r_1 = ax_1 + by_1 \\ b = r_1q_2 + r_2 & \text{e} & r_2 = ax_2 + by_2 \\ r_1 = r_2q_3 + r_3 & \text{e} & r_3 = ax_3 + by_3 \\ r_2 = r_3q_4 + r_4 & \text{e} & r_4 = ax_4 + by_4 \\ \vdots & & \vdots \\ r_{n-3} = r_{n-2}q_{n-1} + r_{n-1} & \text{e} & r_{n-1} = ax_{n-1} + by_{n-1} \\ r_{n-2} = r_{n-1}q_n & \text{e} & r_n = 0. \end{array} \tag{2.1}$$

Os números $x_1, \ldots, x_{n-1}$ e $y_1, \ldots, y_{n-1}$ são inteiros que serão determinados. Desse modo, as informações contidas em (2.1) serão organizadas através da Tabela 2.1:

Tabela 2.1: Algoritmo de Euclides Estendido.

Resto	Quociente	x	y
a	$*$	x_{-1}	y_{-1}
b	$*$	x_0	y_0
r_1	q_1	x_1	y_1
r_2	q_2	x_2	y_2
r_3	q_3	x_3	y_3
$\vdots$	$\vdots$	$\vdots$	$\vdots$
r_{n-2}	q_{n-2}	x_{n-2}	y_{n-2}
r_{n-1}	q_{n-1}	x_{n-1}	y_{n-1}

Fonte: Elaborada pelo autor.

Considerendo a $j-$ésima linha, vamos fazer a divisão de r_{j-2} por r_{j-1} para encontrarmos r_j e q_j, de forma que $r_{j-2} = r_{j-1}q_j + r_j$ e $0 \leqslant r_j < r_{j-1}$. Assim

$$r_j = r_{j-2} - r_{j-1}q_j. \tag{2.2}$$

Observando os valores de x_{j-2}, x_{j-1}, y_{j-2} e y_{j-1} nas linhas $j-1$ e $j-2$, é possível escrever

$$r_{j-2} = ax_{j-2} + by_{j-2} \quad \text{e} \quad r_{j-1} = ax_{j-1} + by_{j-1}.$$

Substituindo estes valores em (2.2), obtém-se

$$\begin{aligned} r_j &= (ax_{j-2} + by_{j-2}) - (ax_{j-1} + by_{j-1})q_j \\ &= a(x_{j-2} - q_j x_{j-1}) + b(y_{j-2} - q_j y_{j-1}). \end{aligned}$$

Portanto,

$$x_j = x_{j-2} - q_j x_{j-1} \quad \text{e} \quad y_j = y_{j-2} - q_j y_{j-1}.$$

Através de um processo recursivo, para calcular x_j e y_j, usam-se os valores contidos nas duas linhas imediatamente anteriores à linha j, desde que as duas linhas que a precedem sejam conhecidas. Então, para determinar os valores de x e y, deve-se fazer

$$a = ax_{-1} + by_{-1} \quad \text{e} \quad b = ax_0 + by_0;$$

que sugere escolher

$$x_{-1} = 1, y_{-1} = 0, x_0 = 0 \text{ e } y_0 = 1.$$

Assim, pode-se dar início ao processo recursivo executando o algoritmo e descobrindo que o máximo divisor comum corresponde ao resto r_{n-1}, ou seja,

$$d = r_{n-1} = ax_{n-1} + by_{n-1}.$$

Logo,

$$n_0 = x_{n-1} \text{ e } m_0 = y_{n-1}.$$

Exemplo 2.2.1. Determine o $MDC(1248, 696)$ e os números inteiros n_0 e m_0 que satisfazem a seguinte equação:

$$1248 \cdot n_0 + 696 \cdot m_0 = MDC(1248, 696).$$

Utilizando a Tabela 2.1, o preenchimento conforme os cálculos de cada divisão fica da seguinte forma:

Resto	Quociente	x	y
1248	$*$	1	0
696	$*$	0	1
552	1	$1 - 1 \cdot 0 = 1$	$0 - 1 \cdot 1 = -1$
144	1	$0 - 1 \cdot 1 = -1$	$1 - 1 \cdot (-1) = 2$
120	3	$1 - 3 \cdot (-1) = 4$	$-1 - 3 \cdot 2 = -7$
24	1	$-1 - 1 \cdot 4 = -5$	$2 - 1 \cdot (-7) = 9$
0	5	$*$	$*$

Com isso, observa-se de acordo com a tabela que o $MDC(1248, 696) = 24$, $n_0 = -5$ e $m_0 = 9$. Portanto,

$$1248 \cdot (-5) + 696 \cdot 9 = 24.$$

Definição 2.2.2. [2] Um número inteiro p é primo se $p \neq \pm 1$ e os únicos divisores de p são ± 1 e $\pm p$. Um número inteiro, diferente de ± 1, que não é primo é chamado de composto.

Definição 2.2.3. [7] Os inteiros a e b são relativamente primos entre si (ou coprimos) se o $MDC(a, b) = 1$.

Teorema 2.2.4. [8] Sejam a, b e c inteiros positivos e suponhamos que a e b são primos entre si, ou seja, $MDC(a, b) = 1$.

(1) Se $a \mid bc$ então $a \mid c$.

(2) Se $a \mid c$ e $b \mid c$, então $ab \mid c$.

Demonstração.

(1) De fato, por hipótese o $MDC(a, b) = 1$. Logo, pelo Teorema 2.2.1 existem números inteiros n e m tais que $na + mb = 1$. Multiplicando-se os dois lados desta igualdade por c temos: $n(ac) + m(bc) = c$. Como $a \mid a$, então $a \mid n(ac)$; e, por hipótese, $a \mid bc$ então $a \mid m(bc)$. Logo, $a \mid [n(ac) + m(bc)]$. Ou seja, $a \mid c$, como queríamos provar.

(2) Consideremos uma identidade de Bézout para a e b: $ax_0 + by_0 = 1$. Multiplicando-se ambos os membros dessa igualdade por c temos: $(ac)x_0 + (bc)y_0 = c$. Como $a \mid a$ e $b \mid c$, então $ab \mid ac$ e, portanto, $ab \mid (ac)x_0$; de maneira análoga, demonstra-se que $ab \mid (bc)y_0$. Logo, $ab \mid [(ac)x_0 + (bc)y_0]$, ou seja, $ab \mid c$. □

Proposição 2.2.1. [9] Seja p um número primo. Se $p \mid ab$ então $p \mid a$ ou $p \mid b$.

Demonstração. Vamos supor que $a \neq 0$ e $b \neq 0$ (o caso $a = 0$ ou $b = 0$ é imediato). Consideremos que $p \nmid a$ e provaremos que o $MDC(a, p) = 1$. De fato, se $c \mid a$ e $c \mid p$, então $c = 1$ ou $c = p$ (pois p é primo). Como $p \nmid a$, então $c = 1$, o que implica, pelo Teorema 2.2.4 que $p \mid b$. □

Definição 2.2.4. [8] Seja a um número inteiro não nulo e diferente de ± 1, então o mínimo do conjunto $S = \{x \in \mathbb{Z} \mid x > 1 \text{ e } x \mid a\}$ é um número primo.

Demonstração. O conjunto S é diferente do vazio, pois $a \in S$. Seja p o mínimo de S. Se p não fosse primo, então existiria um divisor não trivial q de p. Mas $(-q)$ também é um divisor de p. Então p teria um divisor q_1 tal que $1 < q_1 < p$ ($q_1 = q$ ou $q_1 = -q$). Juntando as conclusões temos que $p \mid a$ e $q_1 \mid p$, do que segue que $q_1 \mid a$ e, portanto, $q_1 \in S$. Absurdo, pois p é o mínimo de S e $1 < q_1 < p$. □

2.3 Teorema Fundamental da Aritmética

Teorema 2.3.1. (Teorema Fundamental da Aritmética)[8] Dado um número inteiro $a > 1$, existem números primos $p_1, \ldots, p_r$ ($r \geqslant 1$), de maneira que $a = p_1 p_2 \ldots p_r$. Além disso, se também $a = q_1 q_2 \ldots q_s$ ($s \geqslant 1$), onde os q_j são igualmente primos, então $r = s$ e cada p_i é igual a algum dos q_j.

Demonstração. A demonstração será feita em duas partes:

(i) Devido a Proposição 2.2.4, a tem um divisor primo positivo p_1. Logo, $a = p_1q_1$, para um conveniente $q_1 \in \mathbb{Z}$. Como $a > 0$ e $p_1 > 0$, o mesmo acontece com q_1, que é menor que a (é um fator positivo de a). Se $q_1 = 1$, a demonstração está concluída: $a = p_1$ é primo positivo. Se $q_1 > 1$, repete-se o raciocínio com esse número: toma-se um divisor primo p_2 de q_1, o que é garantido pela Proposição 2.2.4, e, portanto, $q_1 = p_2q_2$, para um conveniente inteiro positivo $q_2(q_2 < q_1)$. Nesta altura: $a = p_1p_2q_2$, em que p_1 e p_2 são primos e $q_2 \geqslant 1$. Agora, repete-se o raciocínio com q_2, e assim por diante. Como $a > q_1 > q_2 > \ldots \geqslant 1$, em alguma etapa desse procedimento se terá $q_r = 1$ e, então, $a = p_1p_2 \ldots p_r$.

(ii) Agora vamos supor que $p_1.p_2 \ldots p_r = q_1.q_2 \ldots q_s$, então $p_1 \mid q_1q_2 \ldots q_s$ o que implica $p_1 \mid q_j$, para algum $1 \leqslant j \leqslant s$ devido a Proposição 2.2.1. Sendo apenas 1 e q_1 os divisores de q_1 e sendo $p_1 \neq 1$, então $p_1 = q_1$. Cancelando p_1 na igualdade inicial, obtemos $p_2.p_3 \ldots p_r = q_2.q_3 \ldots q_s$. Repetindo essa argumentação o quanto for necessário, chegaremos à unicidade conforme o enunciado.

□

2.4 Congruência

Definição 2.4.1. [7] Se a e b são inteiros dizemos que a é congruente a b módulo m $(m > 0)$ se $m \mid (a - b)$. Denotamos isto por $a \equiv b \ (mod\ m)$. Se $m \nmid (a - b)$ dizemos que a é incongruente a b módulo m e denotamos $a \not\equiv b \ (mod\ m)$.

Proposição 2.4.1. [7] Se a e b são inteiros, temos que $a \equiv b \ (mod\ m)$ se, e somente se, existir um inteiro k tal que $a = b + km$.

Demonstração. Se $a \equiv b \ (mod\ m)$, então $m \mid (a - b)$ o que implica na existência de um inteiro k tal que $a - b = km$, isto é, $a = b + km$. A recíproca, pela hipótese, nos garante a existência de um k satisfazendo $a = b + km$. Esta equação pode ser reescrita como $km = a - b$, ou seja, que $m \mid (a - b)$, isto é, $a \equiv b \ (mod\ m)$. □

A próxima proposição nos diz que a relação de congruência, definida no conjunto dos inteiros, é uma relação de equivalência, pois satisfaz as seguintes propriedades: reflexiva, simétrica e transitiva.

Proposição 2.4.2. [7] Se a, b, c e m são inteiros, $m > 0$, as seguintes sentenças são verdadeiras:

1. $a \equiv a \ (mod\ m)$. (Reflexiva)

   Demonstração. Como $m \mid 0$, ou seja, $0 = km$ com k inteiro. Considere $a - a = 0$. Então $a - a = km$. Pela Proposição 2.4.1 temos que $a \equiv a \ (mod\ m)$. □

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

   Demonstração. Por hipótese, temos que $m \mid (a - b)$, ou seja, $a - b = km$ com k inteiro. Multiplicando a equação por (-1) obtemos $-a + b = -km$, que equivale a $b - a = (-k)m$ com $(-k)$ inteiro. Logo, pela Proposição 2.4.1 temos que $b \equiv a \ (mod\ m)$. □

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

   Demonstração. Por hipótese, temos que $a \equiv b \ (mod\ m)$ e que $b \equiv c \ (mod\ m)$, ou seja, $m \mid (a-b)$, que implica $a-b = k_1 m$ com k_1 inteiro e $m \mid (b-c)$, que implica $b-c = k_2 m$ com k_2 inteiro. Somando-se as duas equações, obtemos o seguinte resultado: $a-b+b-c = k_1 m + k_2 m$ que implica $a-c = (k_1+k_2)m$. Como k_1+k_2 é a soma de dois números inteiros, vamos representar essa soma por k, ou seja, $a-c = km$. Logo, pela Proposição 2.4.1, temos que $a \equiv c \ (mod\ m)$. □

2.5 Conjunto $\mathbb{Z}_m$

O conjunto quociente de $\mathbb{Z}$, denotado por $\mathbb{Z}_m$, possui um nome especial: *conjunto dos inteiros módulo m* que representa um papel fundamental para os estudos na área da criptografia. Os elementos de $\mathbb{Z}_m$ são subconjuntos de $\mathbb{Z}$ que são chamados de *classes de equivalência da congruência módulo m* [2].

Seja $a \in \mathbb{Z}$ então a classe de a é representada da seguinte forma:

$$\overline{a} = \{a + km : k \in \mathbb{Z}\}.$$

Se $a \in \mathbb{Z}$, então podemos dividi-lo por m, obtendo q e r inteiros tais que

$$a = mq + r \text{ e } 0 \leqslant r \leqslant m-1.$$

Logo $a - r = mq$ é um múltiplo de m, ou seja, $a \equiv r \ (mod\ m)$. Desta maneira, um inteiro qualquer é congruente módulo m a um inteiro no intervalo que vai de 0 a $m-1$. Resumindo

$$\mathbb{Z}_m = \left\{\overline{0}, \overline{1}, \ldots, \overline{m-1}\right\}.$$

2.6 Divisões em $\mathbb{Z}_m$

Digamos que $\overline{a} \in \mathbb{Z}_m$. Diremos que a classe $\overline{a}' \in \mathbb{Z}_m$ é o *inverso* de $\overline{a}$ se a equação $\overline{a}.\overline{a}' = \overline{1}$ é verificada em $\mathbb{Z}_m$. É claro que se $\overline{a} = \overline{0}$ então $\overline{a}$ não tem inverso. Em $\mathbb{Z}_m$, além da classe $\overline{0}$ pode haver outros elementos sem inverso. Nesse sentido, o teorema seguinte nos diz quando uma classe possui inverso em $\mathbb{Z}_m$.

Teorema 2.6.1. [2] A classe $\overline{a}$ possui inverso em $\mathbb{Z}_m$ se, e somente se, $MDC(a,m) = 1$.

Demonstração. A demonstração será feita em duas partes:

(i) ($\Rightarrow$) Por hipótese, $\overline{a} \in \mathbb{Z}_m$ é um elemento inversível. Logo, existe $\overline{a}' \in \mathbb{Z}_m$ tal que $\overline{a}.\overline{a}' = \overline{a.a'} = \overline{1}$. Daí, temos que

$$\overline{a.a'} = \overline{1} \Rightarrow a.a' \equiv 1 \ (mod\ m) \Rightarrow m \mid (aa'-1) \Rightarrow aa'-1 = mq, q \in \mathbb{Z}.$$

Reescrevendo a equação temos: $aa' - mq = 1$. De acordo com o Teorema 2.2.1, provamos que o $MDC(a,m) = 1$.

(ii) ($\Leftarrow$) Se o $MDC(a,m) = 1$, novamente pelo Teorema 2.2.1, existem inteiros x_0 e y_0 tais que $ax_0 + my_0 = 1$.

$$ax_0 - 1 = m(-y_0) \Rightarrow ax_0 \equiv 1 \ (mod\ m) \Rightarrow \overline{ax_0} \equiv \overline{1} \Rightarrow \overline{a}.\overline{x_0} = \overline{1}.$$

Portanto, $\overline{a}$ é inversível e $\overline{x_0}$ é seu inverso.

□

Uma maneira muito eficiente para encontrar o inverso multiplicativo de uma classe $\overline{a}$ no conjunto quociente de $\mathbb{Z}$, é utilizarmos o Algoritmo de Euclides Estendido.

Exemplo 2.6.1. Sabendo que o $MDC(2978, 55) = 1$, determine o inverso 55 em $\mathbb{Z}_{2978}$. Para isso, vamos primeiramente recorrer à Tabela 2.1 Algoritmo de Euclides Estendido.

Resto	Quociente	x	y
2978	$*$	1	0
55	$*$	0	1
8	54	$1 - 54 \cdot 0 = 1$	$0 - 54 \cdot 1 = -54$
7	6	$0 - 6 \cdot 1 = -6$	$1 - 6 \cdot (-54) = 325$
1	1	$1 - 1 \cdot (-6) = 7$	$-54 - 1 \cdot 325 = -379$
0	7	$*$	$*$

Com isso, observamos pela tabela que o $MDC(2978, 55) = 1$, $n_0 = 7$ e $m_0 = -379$. Portanto,

$$2978 \cdot (7) + 55 \cdot (-379) = 1. \tag{2.3}$$

Pela Proposição 2.4.1, podemos reescrever (2.3) da seguinte forma:

$$55 \cdot (-379) \equiv 1 \ (mod\ 2978).$$

Pelo Teorema 2.6.1, concluímos que o inverso de 55 em $\mathbb{Z}_{2978}$ é -379.

2.7 Teorema de Euler e Fermat

Nesta seção, veremos dois teoremas que desempenham papéis importantes na criptografia de chave pública que são o Teorema de Euler e o Teorema de Fermat, também conhecido como *Pequeno Teorema de Fermat*.

Antes de enunciarmos o próximo teorema, vamos definir a função ϕ de Euler e duas propriedades importantes.

Definição 2.7.1. (Função Totiente de Euler)[7] Se m é um inteiro positivo, a função totiente de Euler, também chamada de função ϕ de Euler, denotada por $\phi(m)$, é definida como sendo o número de inteiros positivos menores do que ou iguais a m que são relativamente primos com m. Por convenção, $\phi(1) = 1$. Além disso, esta função apresenta duas propriedades importantes:

1. Se m é um número primo então $\phi(m) = m - 1$.
2. A função ϕ de Euler é multiplicativa, isto é, se m e n são inteiros positivos relativamente primos entre si, então $\phi(m \cdot n) = \phi(m) \cdot \phi(n)$.

Lema 2.7.1. [7] Se p é um número primo e a um número inteiro positivo temos

$$\phi(p^a) = p^a - p^{a-1}.$$

Demonstração. Pela definição de $\phi(m)$, sabe-se que $\phi(p^a)$ é o número de inteiros positivos não-superiores de p^a e relativamente primos com p^a. Porém, os únicos números não primos com p^a e menores do que ou iguais a p^a são aqueles divisíveis por p. Como os múltiplos de p não-superiores a p^a são, em número, p^{a-1}, segue o resultado que $\phi(p^a) = p^a - p^{a-1}$. □

Teorema 2.7.2. [7] Para $m = p_1^{a_1} p_2^{a_2} \dots p_r^{a_r}$, temos

$$\phi(m) = m\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right)\cdots\left(1 - \frac{1}{p_r}\right).$$

Demonstração. De acordo com o Lema 2.7.1 temos

$$\phi(p_i^{ai}) = p_i^{a_i} - p_i^{a_i - 1} = p_i^{a_i}\left(1 - \frac{1}{p_i}\right).$$

Portanto, pelo fato da função ϕ ser multiplicativa isso garante que

$$\begin{aligned}
\phi(p_1^{a_1} p_2^{a_2} \dots p_r^{a_r}) &= p_1^{a_1}\left(1 - \frac{1}{p_1}\right) p_2^{a_2}\left(1 - \frac{1}{p_2}\right)\cdots p_r^{a_r}\left(1 - \frac{1}{p_r}\right) \\
&= p_1^{a_1} p_2^{a_2}\cdots p_r^{a_r}\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right)\cdots\left(1 - \frac{1}{p_r}\right) \\
&= m\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right)\cdots\left(1 - \frac{1}{p_r}\right).
\end{aligned}$$

□

Teorema 2.7.3. (Teorema de Euler)[9] Se m é um inteiro positivo e a é um inteiro com $MDC(a, m) = 1$, então $a^{\phi(m)} \equiv 1 \ (mod\ m)$.

Demonstração. Sejam $s_1, s_2, \dots, s_k$ os inteiros de 1 a m, inclusive os extremos, que são primos com m (logo $k = \phi(m)$). Dividamos cada as_i por m:

$$as_i = mq_i + r_i \ (0 \leqslant r_i < m).$$

Se existisse um primo p tal que $p \mid m$ e $p \mid r_i$, dessa igualdade decorreria que $p \mid as_i$. Mas então $p \mid a$ e $p \mid s_i$, o que seria impossível já que $MDC(a, m) = 1$ (hipótese) e ainda $MDC(m, s_i) = 1$, devido à escolha dos s_i. Donde m e r_i são primos entre si, para todo i, $1 \leqslant i \leqslant k$.

Mostremos agora que na sequência de restos $r_1, r_2, \dots, r_k$ não há elementos repetidos. Suponhamos que $r_i = r_j$ $(1 \leqslant i, j \leqslant k; i \neq j)$. Então $as_i - mq_i = as_j - mq_j$ e daí $a(s_i - s_j) = m(q_i - q_j)$. Como $MDC(a, m) = 1$, então $m \mid (s_i - s_j)$. Como $1 \leqslant s_i, s_j \leqslant m$, então teríamos que ter $s_i = s_j$, o que não é possível, posto que $i \neq j$. Disso tudo decorre então que $s_1, s_2, \dots, s_k = r_1, r_2, \dots, r_k$. Assim, se multiplicarmos as congruências $as_i \equiv r_j (mod\ m)$ decorrentes de $as_i = mq_i + r_i$ $(1 \leqslant i \leqslant k)$:

$$a^k s_1 s_2 \dots s_k \equiv (as_1)(as_2)\dots(as_k) \equiv r_1 r_2 \dots r_k (mod\ m),$$

os produtos $s_1, s_2, \ldots, s_k$ e $r_1, r_2, \ldots, r_k$ que nela aparecem são iguais. Como m é primo com cada r_i(ou s_i) e, portanto, com o produto $r_1.r_2 \ldots r_k$, então esse produto pode ser cancelado na última congruência, resultando a tese:

$$a^{\phi(m)} \equiv 1 \ (mod\ m),$$

pois $k = \phi(m)$. □

Teorema 2.7.4. (Pequeno Teorema de Fermat)[9] Sejam p um número primo e a um inteiro que não é divisível por p. Então $a^{p-1} \equiv 1 \ (mod\ p)$.

Demonstração. Basta aplicar a primeira propriedade da Definição 2.7.1 no Teorema 2.7.3. Por hipótese p é primo, então $\phi(p) = p - 1$. Do Teorema 2.7.3 segue que $a^{\phi(p)} \equiv 1 \ (mod\ p)$, ou seja, $a^{p-1} \equiv 1 \ (mod\ p)$. □

Corolário 2.7.1. [9] Se p é um primo e a é um inteiro positivo, então $a^p \equiv a \ (mod\ p)$.

Demonstração. Se $MDC(a, p) = 1$, então, multiplicando a congruência do Pequeno Teorema de Fermat por a temos que $a^p \equiv p \ (mod\ p)$. Se $MDC(a, p) \neq 1$, então $a \equiv 0 \ (mod\ p)$, pois p é primo. Donde:

$$a^p \equiv 0 \equiv a(mod\ p).$$

□

2.8 Raízes Primitivas

A importância dos resultados introduzidos nesta seção apresenta uma caracterização dos números que possuem raízes primitivas e que servem de fundamento para o entedimento de alguns algoritmos criptográficos de chave pública como Diffie-Hellman e ElGamal.

De acordo com o Teorema 2.7.3, se m é um inteiro positivo e a é um inteiro com $MDC(a, m) = 1$, então $a^{\phi(m)} \equiv 1 \ (mod\ m)$. Considerando-se as potências de a, isto é, $a, a^2, a^3, \ldots a^i, \ldots$ existirá um k de menor expoente tal que $a^k \equiv 1 \ (mod\ m)$. Nessas condições, k é chamado *ordem de a módulo m* e é denotado por $ord_m a$, [7].

Exemplo 2.8.1. A ordem de 12 módulo 19 é igual a 6. Com efeito,

$$\begin{aligned} 12^1 &\equiv 12 \ (mod\ 19) \\ 12^2 &\equiv 11 \ (mod\ 19) \\ 12^3 &\equiv 18 \ (mod\ 19) \\ 12^4 &\equiv 7 \ (mod\ 19) \\ 12^5 &\equiv 8 \ (mod\ 19) \\ 12^6 &\equiv 1 \ (mod\ 19). \end{aligned}$$

Percebemos que ao calcular $12^6 \equiv 1 \ (mod\ 19)$ nenhuma outra potência de 12 menor que 6 é congruente a 1 módulo 19. Logo,

$$ord_{19}(12) = 6.$$

Teorema 2.8.1. [7] Se $k = ord_m a$ e $a^h \equiv 1\ (mod\ m)$, então $k \mid h$.

Demonstração. Pelo Teorema 2.1.1, dados dois inteiros h e k, $k > 0$, existe um único par de inteiros q e r, $0 \leq r < k$ tal que $h = kq + r$. Assim, temos

$$a^h = a^{kq+r} = (a^k)^q a^r \equiv a^r (mod\ m).$$

Por definição $a^k \equiv 1\ (mod\ m)$, pois $k = ord_m a$. Logo, como $a^h \equiv 1\ (mod\ m)$, acabamos de mostrar que $a^r \equiv 1\ (mod\ m)$. Como $r < k$, r deve ser zero pois k é o menor inteiro positivo para o qual $a^k \equiv 1\ (mod\ m)$. Portanto, $h = qk$, ou seja, $k \mid h$. □

Corolário 2.8.1. [7] $ord_m(a) \mid \phi(m)$.

Demonstração. O Teorema de Euler garante que $a^{\phi(m)} \equiv 1(mod\ m)$ desde que o $MDC(a, m) = 1$. Portanto, pelo Teorema 2.8.1, a $ord_m a \mid \phi(m)$. □

Definição 2.8.1. [7] Se $ord_m(a) = \phi(m)$ dizemos que a é uma raiz primitiva módulo m.

Exemplo 2.8.2. Para verificar se o número 3 é uma raiz primitiva módulo 14, considere as seguintes etapas:

1. É preciso encontrar o valor de $\phi(14)$:

$$\begin{aligned} \phi(14) &= 2 \cdot 7\left(1 - \frac{1}{2}\right)\left(1 - \frac{1}{7}\right) \\ \phi(14) &= 2 \cdot 7\left(\frac{1}{2}\right)\left(\frac{6}{7}\right) \\ \phi(14) &= 6. \end{aligned}$$

2. Em seguida, calculamos a ordem de 3 módulo 14. Veja que não é necessário computar todas as potências do número 3, pois o Corolário 2.8.1 garante testar somente as potências cujos expoentes são divisores de $\phi(14) = 6$:

$$\begin{aligned} 3^1 &\equiv 3\ (mod\ 14) \\ 3^2 &\equiv 9\ (mod\ 14) \\ 3^3 &\equiv 13\ (mod\ 14) \\ 3^6 &\equiv 1\ (mod\ 14). \end{aligned}$$

   Portanto, a $ord_{14}(3) = 6$.

Sendo assim, o número 3 é uma raiz primitiva módulo 14, pois a $ord_{14}(3) = \phi(14)$.

Teorema 2.8.2. [7] Considere $k = ord_m a$, então $a^t \equiv a^h\ (mod\ m)$ se, e somente se, $t \equiv h\ (mod\ k)$.

Demonstração. ($\Rightarrow$) Por hipótese $a^t \equiv a^h \ (mod\ m)$. Sem perda de generalidade podemos supor $t \geq h$. Como $a^t = a^h a^{t-h}$ e $a^t \equiv a^h \ (mod\, m)$ temos que $a^h a^{t-h} \equiv a^h \ (mod\, m)$. Como $MDC(a, m) = 1$ implica $MDC(a^h, m) = 1$, podemos cancelar a^h, nesta última congruência, obtendo

$$1 \equiv a^{t-h} \ (mod\ m).$$

Pelo Teorema 2.8.1, $k \mid (t - h)$ o que equivale a dizer que $t \equiv h \ (mod\ k)$.

($\Leftarrow$) Se $t \equiv h \ (mod\ k)$ então existe um inteiro n tal que $t = h + nk$. Logo,

$$a^t = a^{h+nk} = a^h (a^k)^n \equiv a^h \ (mod\ m),$$

pois k é a ordem de a módulo m, o que conclui a demonstração. □

2.9 Logaritmos Discretos

Nesta seção, apresentamos uma visão geral dos logaritmos discretos que são fundamentais nos estudos da área da criptografia. Além de estarem associados com as raízes primitivas, eles são utilizados no esquema de troca de chaves Diffie-Hellman, no sistema de criptografia ElGamal e também no algoritmo de assinatura digital (DSA) [10].

Da mesma forma que os logaritmos com números reais são o inverso da exponenciação, na aritmética modular existe uma função que também preserva essa semelhança. Considere a uma raiz primitiva módulo p com p primo. Então, sabe-se que as potências de a de 1 até $(p - 1)$ produzem cada inteiro de 1 até $(p - 1)$ exatamente uma vez.

Pela definição da aritmética modular, tem-se que para qualquer inteiro b satisfaz

$$b \equiv r \ (mod\ p) \text{ para algum } r, \text{ onde } 0 \leqslant r \leqslant p - 1.$$

Assim, para qualquer inteiro b e a uma raiz primitiva módulo p, podemos encontrar um expoente exclusivo i, tal que

$$b \equiv a^i \ (mod\ p) \text{ onde } 0 \leqslant i \leqslant p - 1.$$

O expoente i é definido como **logaritmo discreto** do número b para a base $a (mod\, p)$ e é denotado por $dlog_{a,p}(b)$.

Exemplo 2.9.1. As potências de 7 com expoentes positivos módulo 13 são:

$$\begin{aligned}
7^1 &\equiv 7 \ (mod\ 13)\\
7^2 &\equiv 10 \ (mod\ 13)\\
7^3 &\equiv 5 \ (mod\ 13)\\
7^4 &\equiv 9 \ (mod\ 13)\\
7^5 &\equiv 11 \ (mod\ 13)\\
7^6 &\equiv 12 \ (mod\ 13)\\
7^7 &\equiv 6 \ (mod\ 13)\\
7^8 &\equiv 3 \ (mod\ 13)\\
7^9 &\equiv 8 \ (mod\ 13)\\
7^{10} &\equiv 4 \ (mod\ 13)\\
7^{11} &\equiv 2 \ (mod\ 13)\\
7^{12} &\equiv 1 \ (mod\ 13).
\end{aligned}$$

Note que a $ord_{13}(7) = 12$ e $\phi(13) = 13 - 1 = 12$. Logo, 7 é uma raiz primitiva módulo 13. Dessa forma, com os resultados obtidos é possível determinar o índice de um inteiro x na base 7 módulo 13. Por exemplo, $dlog_{7,13}(8) = 9$, pois $7^9 \equiv 8 \ (mod\ 13)$. Com isso, podemos obter a Tabela 2.2:

Tabela 2.2: Logaritmos discretos com base 7, módulo 13.

x	1	2	3	4	5	6	7	8	9	10	11	12
$dlog_{7,13}(x)$	12	11	8	10	3	7	1	9	4	2	5	6

Fonte: Elaborada pelo autor.

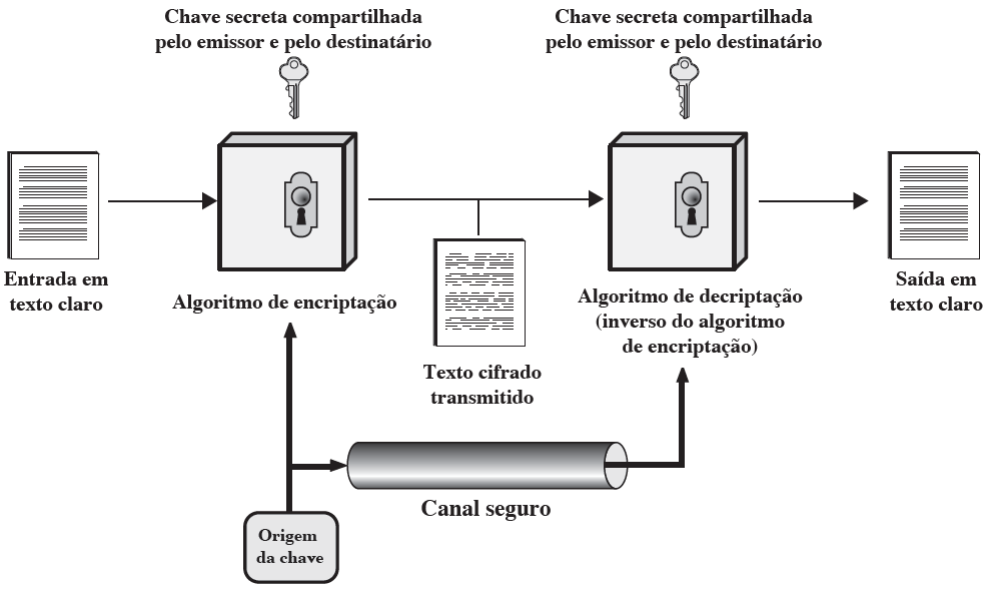
3 Sistemas de Criptografia

Diante dos avanços da tecnologia foi possível também o desenvolvimento da criptografia que possibilitou o surgimento de dois sistemas diferentes, a criptografia simétrica e a criptografia assimétrica. Até a década de 1970, a criptografia simétrica era o único mecanismo criptográfico utilizado para garantir a segurança das informações, até que Diffie e Hellman remodelaram a área da criptografia introduzindo um novo método chamado de criptografia assimétrica de chave pública que possibilitou solucionar o problema do compartilhamento das chaves. Neste capítulo, apresentaremos as principais vantagens e desvantagens na utilização dos sistemas de criptografia com base nas referências [6], [10], [11] e [12].

3.1 Criptografia Simétrica

A criptografia simétrica, também conhecida como *criptografia de chave secreta*, *encriptação convencional* ou *encriptação de chave única*, é um processo que utiliza apenas uma única chave para codificação e decodificação, ou seja, o emissor e o destinatário ambos compartilham da mesma chave. Neste caso a Figura 3.1 a seguir exemplifica esse método:

Figura 3.1: Criptografia simétrica.



Fonte: Adaptado de [10](p. 21).

Observa-se que o processo ilustrado na Figura 3.1 é constituído pelas seguintes etapas:

1. O texto claro é a mensagem que serve como entrada do algoritmo de encriptação (codificação).

2. O algoritmo de encriptação realiza várias substituições e transformações no texto claro.

3. A chave secreta é inserida no algoritmo de encriptação e seu valor não depende do texto claro e nem do algoritmo. O algoritmo produzirá uma saída diferente dependendo da chave que está sendo usada no momento. As substituições e transformações exatas praticadas pelo algoritmo dependem da chave.

4. O texto cifrado é a mensagem embaralhada (ininteligível) produzida como saída que depende do texto claro e da chave secreta. Para uma determinada mensagem, duas chaves diferentes produzirão dois textos cifrados diferentes.

5. Algoritmo de decriptação (decodificação) é essencialmente o algoritmo de encriptação executado ao contrário, pois transforma o texto cifrado no texto claro que deu origem.

Os principais algoritmos de cifras simétricas mais utilizados são, conforme [10]:

- **DES:** O *Data Encryption Standard (DES)*, criado em 1977 pelo National Bureau of Standards, hoje National Institute of Standards and Technology (NIST), foi um dos primeiros algoritmos criptográficos mais utilizados no mundo e que posteriormente foi substituído por uma versão mais aperfeiçoada conhecida como 3DES (que basicamente repete três vezes o algoritmo DES).

- **AES:** O *Advanced Encryption Standard (AES)* foi publicado em 2001 pelo NIST. O AES é um algoritmo criptográfico simétrico efetivamente mais rápido e mais seguro que o DES, com a finalidade de superar as fragilidades do seu antecessor.

Outros exemplos de algoritmos simétricos são o IDEA, Blowfish, Twofish, Serpent, CAST-128 e RC4.

3.2 Criptografia Assimétrica

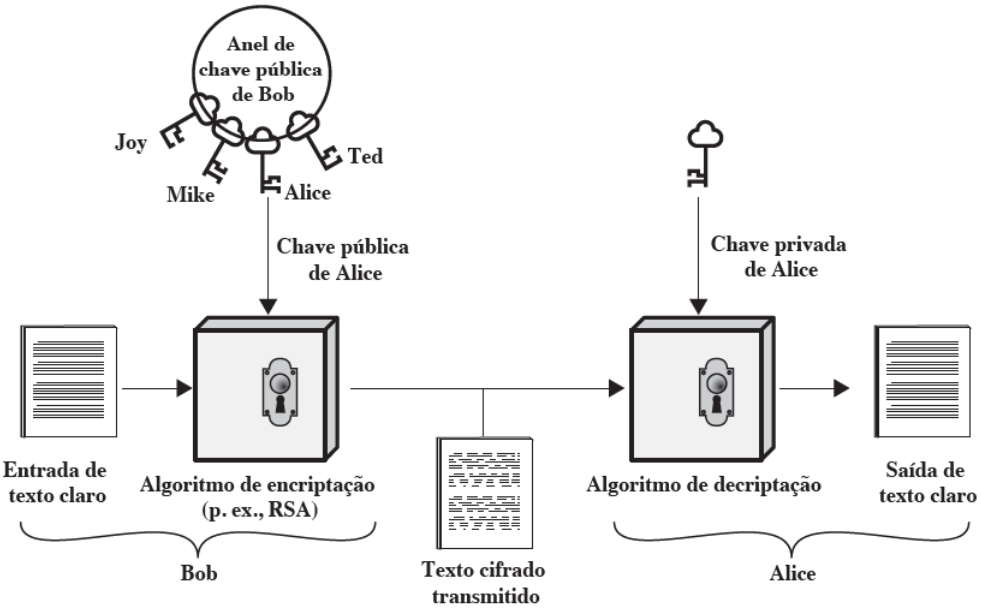
O sistema de criptografia assimétrica ou *criptografia de chave pública* é constituído por duas chaves distintas e que são relacionadas matematicamente: a chave pública *(chave de codificação ou encriptação)* e chave privada *(chave de decodificação ou decriptação)*.

A chave pública pode ser distribuída para qualquer pessoa ao contrário da chave privada que deverá ser mantida em segurança pelo criador do par das chaves. Uma característica desse sistema é que qualquer uma das chaves pode ser usada para codificar, com a outra sendo voltada para decodificar, permitindo assim duas situações: a codificação de mensagens *(confidencialidade)* ou assinaturas digitais *(autenticidade)*.

Se a finalidade for utilizar a criptografia assimétrica para codificação de mensagens impedindo que terceiros tenham acesso ao conteúdo de modo que somente o destinatário

legítimo seja capaz de interpretar a mensagem, então a chave pública será usada para codificar e a chave privada para decodificar. A Figura 3.2 a seguir exemplifica esse método:

Figura 3.2: Criptografia assimétrica para assegurar confidencialidade.



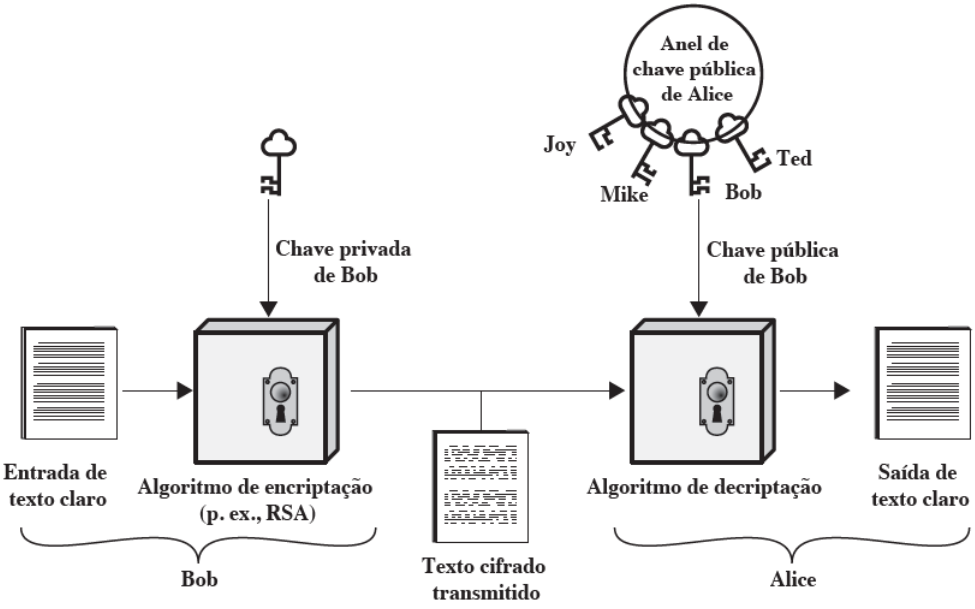
Fonte: Adaptado de [10](p. 202).

Nesse caso, o processo é constituído pelas seguintes etapas conforme ilustrado na Figura 3.2:

1. Os usuários geram um par de chaves para codificar e decodificar a mensagem.
2. Os usuários disponibilizam publicamente suas respectivas chaves públicas e armazenam num local seguro as chaves privadas correspondentes.
3. Cada usuário possui uma coleção de chaves públicas de outros usuários.
4. Bob envia uma mensagem secreta para Alice codificando com a chave pública de Alice.
5. Alice, ao receber a mensagem, decodifica usando sua chave privada, retornando a mensagem original.

No entanto, se a criptografia assimétrica tem por finalidade garantir autenticidade do autor da mensagem como é o caso das assinaturas digitais, então a Figura 3.3 a seguir ilustra esta situação:

Figura 3.3: Criptografia assimétrica para assegurar autenticidade.



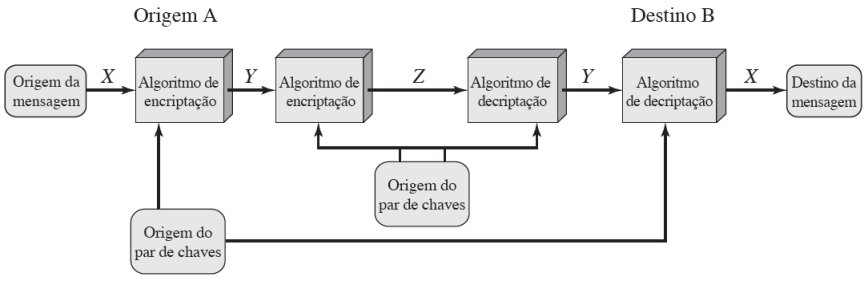
Fonte: Adaptado de [10](p. 202).

Observa-se através da Figura 3.3 que a chave privada é usada para codificar e a chave pública para decodificar, por exemplo: Bob envia uma mensagem para Alice e a encripta usando sua chave privada antes de transmiti-la. Alice decripta a mensagem com a chave pública de Bob. Como a mensagem foi encriptada com a chave privada de Bob, somente Bob poderia ter preparado a mensagem. Dessa maneira, a mensagem encriptada inteira serve como uma assinatura digital, sendo impossível alterar a mensagem sem acesso à chave privada de Bob, o que garante sua autenticidade em termos da origem e da integridade dos dados.

Nesse processo não é possível garantir confidencialidade da mensagem, uma vez que qualquer um que tiver a chave pública de Bob conseguirá ter acesso ao conteúdo das informações, ou seja, a mensagem é segura contra qualquer modificação em seu conteúdo, mas não contra à observação não autorizada.

Apesar disso, é possível solucionar a questão da confidencialidade conforme exposto na Figura 3.4:

Figura 3.4: Criptografia assimétrica para assegurar confidencialidade e autenticidade.



Fonte: Adaptado de [10](p. 205).

De acordo com a Figura 3.4, a partir da mensagem X, tem-se o início do processo de encriptação com a chave privada do emissor oferecendo portanto a assinatura digital, resultando em Y. Em seguida, repetimos novamente o processo de encriptação em Y usando a chave pública do receptor, obtendo Z. Por fim, a decriptação de Z só é possível pelo receptor intencionado, pois é o único que possui a chave privada equivalente, de modo a garantir a confidencialidade da mensagem.

Apesar de solucionar a questão da confidencialidade, é importante enfatizar que essa técnica apresenta a seguinte desvantagem, o algoritmo de criptografia assimétrica, que é complexo, é executado quatro vezes ou invés de duas, tornando o processo ainda mais lento.

Os principais algoritmos de chave pública são: Diffie-Hellman, RSA, ElGamal, Curvas Elípticas e DSA.

3.3 Comparação entre os Sistemas de Criptografia

Com o surgimento da criptografia assimétrica foi possível diagnosticar uma série de problemas apontados na criptografia simétrica, principalmente aqueles voltados à questão do gerenciamento e distribuição de chaves tornando o processo mais simples e seguro. A Tabela 3.1 apresenta alguns pontos positivos e negativos entre a criptografia simétrica e a criptografia assimétrica.

Tabela 3.1: Comparação entre os sistemas de criptografia.

	Criptografia Simétrica	**Criptografia Assimétrica**
Processamento	Rápido	Lento
Distribuição das chaves	Complexo	Simples
Assinatura Digital	Não oferece	Oferece

Fonte: Adaptado de [6](p. 39).

Analisando o conteúdo exposto da Tabela 3.1, percebe-se que os algoritmos que utilizam a criptografia simétrica são rápidos quando comparados aos algoritmos com criptografia assimétrica que tendem a ser mais seguros e lentos, tornando-os mais dependentes da capacidade do processamento das máquinas.

Apesar disso, a criptografia assimétrica possibilita o gerenciamento e a distribuição das chaves de forma simples sem a necessidade do emissor e o receptor se reunirem antecipadamente para o compartilhamento das chaves permitindo não só a confidencialidade de dados como também a implementação das assinaturas digitais.

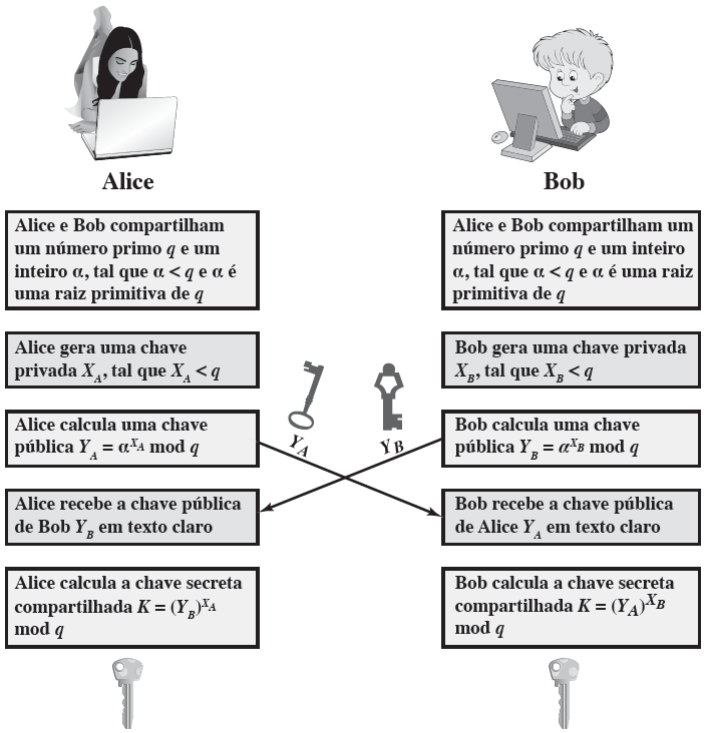
Em virtude das diversas características apresentadas pelos dois sistemas, foi desenvolvido um modelo que fosse capaz de utilizar as vantagens de cada um, aproveitando assim a rapidez do algoritmo simétrico para o ciframento de mensagens em si e o assimétrico para a distribuição de chaves e assinaturas digitais. Esse tipo de modelo é chamado de *modelo híbrido*, no qual é possível combinar três tipos de mecanismos criptográficos: o ciframento, a assinatura digital e o *hashing* [6]. No Capítulo 5, será descrito e analisado os conceitos de assinatura digital e *hashing*.

Os algoritmos que operam com sistemas criptográficos híbridos são, conforme [6]: IPSec, SSL/TLS, PGP, S/MIME, SET e X.509.

3.4 A Criptografia Diffie-Hellman

Whitfield Diffie e Martin Hellman introduziram em 1976 o conceito de criptografia de chave pública, conhecido também como troca de chaves Diffie-Hellman e que serve de base para o algoritmo ElGamal, descrito na Seção 3.5. A principal vantagem do protocolo Diffie-Hellman é permitir que o emissor e o receptor possam trocar as chaves de maneira segura em canais inseguros, conforme ilustrado na Figura 3.5 [10].

Figura 3.5: Algoritmo de troca de chaves Diffie-Hellman.



Fonte: [10].

Alice e Bob pretendem utilizar um sistema de codificação criptográfica simétrica para troca de mensagens através de um canal inseguro. A princípio, eles precisam compartilhar a chave secreta comum e, conforme a Figura 3.5, o protocolo Diffie-Hellman faz com que esse compartilhamento seja possível.

Para esse esquema, observa-se que Alice e Bob combinam um mesmo número primo q e um inteiro α que é uma raiz primitiva módulo q, ambos conhecidos publicamente. Em seguida, Alice escolhe um inteiro aleatório $X_a < q$ e calcula

$$Y_a \equiv \alpha^{X_a} \ (mod\ q)$$

e envia o resultado Y_a para Bob mantendo em segredo o expoente X_a. Da mesma maneira, Bob escolhe independentemente um inteiro aleatório $X_b < q$ e calcula

$$Y_b \equiv \alpha^{X_b} \ (mod\ q)$$

e envia o resultado para Alice, também mantendo em segredo seu expoente X_b. Portanto, X_a é a chave privada de Alice e Y_a é a chave pública correspondente, enquanto que para Bob a chave privada é X_b e a chave pública correspondente é Y_b.

A chave secreta comum é obtida da seguinte maneira:

- Alice calcula: $Y_b^{X_a}(mod\ q) \equiv \alpha^{X_a X_b}\ (mod\ q)$.
- Bob calcula: $Y_a^{X_b}(mod\ q) \equiv \alpha^{X_a X_b}\ (mod\ q)$.

Logo, a chave secreta comum é

$$k \equiv \alpha^{X_a X_b}\ (mod\ q).$$

Exemplo 3.4.1. Suponha que Alice e Bob desejam realizar a troca de chaves utilizando o algoritmo de Diffie-Hellman:

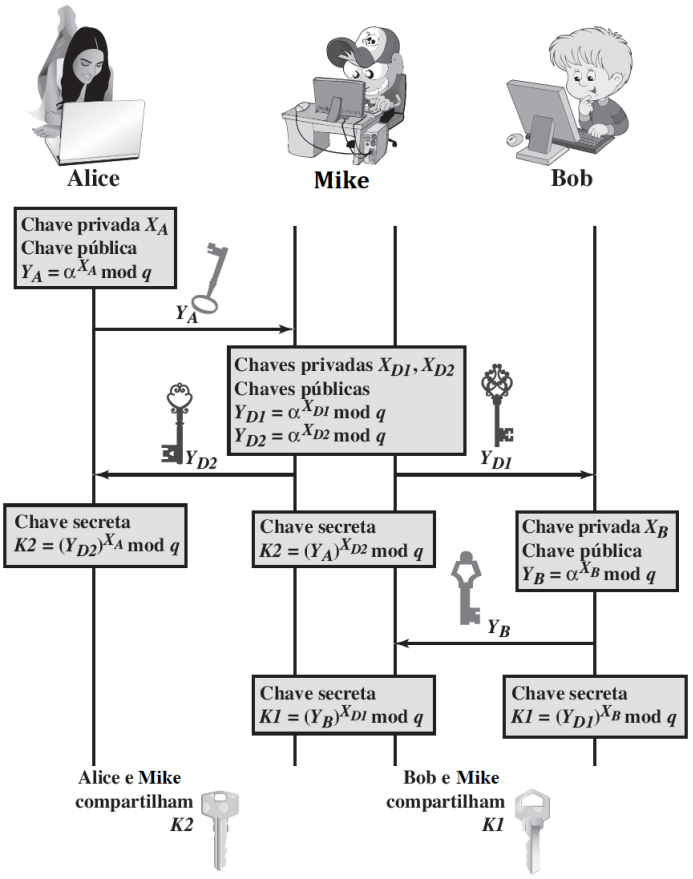
- Alice e Bob definem um número primo $q = 13$ e uma raiz primitiva de 13, neste caso, $\alpha = 7$, conforme Exemplo 2.9.1.
- Alice escolhe seu número secreto $X_a = 10$ e calcula $Y_a \equiv 7^{10}\ (mod\ 13)$ e envia o resultado $Y_a = 4$ para Bob.
- Bob escolhe seu número secreto $X_b = 5$ e calcula $Y_b \equiv 7^5\ (mod\ 13)$ e envia o resultado $Y_b = 11$ para Alice.
- Alice calcula $k \equiv 11^{10}(mod\ 13) \Rightarrow k = 10$.
- Bob calcula $k \equiv 4^5(mod\ 13) \Rightarrow k = 10$.
- Portanto, a chave secreta comum é $k = 10$.

Considere que o espião, Mike, tivesse conhecimento dos números inteiros $q = 13$, $\alpha = 7$, $Y_a = 4$ e $Y_b = 11$, mas não do logaritmo discreto X_a de Alice e X_b de Bob na base α. No Exemplo 3.4.1, por meio do método de força bruta computacional é possível determinar a chave secreta devido ao tamanho dos números escolhidos. Dessa maneira, o espião facilmente encontraria uma solução para a equação $7^{X_a}(mod\ 13) \equiv 4$, ou seja, $dlog_{7,13}(4) = 10$. Portanto, $X_a = 11^{10}(mod\ 13)$, ou seja, $k = 10$.

Apesar de ser praticamente fácil resolver funções exponenciais módulo um número primo, a segurança da troca de chaves Diffie-Hellman se sustenta no problema do logaritmo discreto, o que o torna impraticável para se obter soluções em tempo hábil, pois quanto maior for o tamanho do número primo, mais tempo necessário levará um programa de computador para realizar o processamento dos cálculos.

No entanto, o método de força bruta para determinar uma solução possível para a chave secreta do protocolo Diffie-Hellman não é a única forma de ataque possível. Existe uma outra maneira de ataque chamado "*man-in-the-middle*" (ataque do homem do meio) quando uma pessoa espiã deseja interceptar mensagens compartilhadas entre duas pessoas com a intenção de replicar a mensagem interceptada ou alterar seu conteúdo de modo que pensem que estão falando um com o outro, quando na verdade estão se comunicando com um espião intermediário. A Figura 3.6 a seguir ilustra essa situação:

Figura 3.6: Ataque man-in-the-middle.



Fonte: [10].

Na Figura 3.6, pode-se observar as seguintes etapas:

1. Alice cria a chave privada X_A e a chave pública correspondente Y_A.
2. Alice transmite Y_a para Bob e Mike intercepta.
3. Nesse momento, Mike assume as identidades de Alice e Bob, gerando duas chaves privadas aleatórias X_{D1} e X_{D2} com suas respectivas chaves públicas correspondentes Y_{D1} e Y_{D2}.
4. Mike transmite Y_{D2} para Alice onde ambos irão compartilhar a mesma chave $K_2 = {Y_{D2}}^{X_A}(mod\ q) = {Y_A}^{X_{D2}}(mod\ q)$ e também transmite Y_{D1} para Bob criar a chave pública Y_B.
5. Bob calcula $K_1 = {Y_{D1}}^{X_B}(mod\ q)$ que equivale a $K_1 = {Y_B}^{X_{D1}}(mod\ q)$ de Mike.

Quando Bob transmite uma mensagem criptografada para Alice, ele utiliza a chave que trocou previamente com Mike pensando ser a chave para se comunicar com Alice. Mike quando intercepta a mensagem e a decifra, ele simplesmente pode replicá-la como alterá-la. Com isso, Mike explora o fato de que Alice não é capaz de ter a garantia de que as mensagens recebidas sejam realmente de Bob - da mesma forma para Bob, prejudicando toda comunicação entre eles. Para contornar essa situação, de maneira

que seja possível autenticar as mensagens encaminhadas por Alice e Bob, pode-se fazer o uso das assinaturas digitais que serão descritas no Capítulo 5.

3.5 A Criptografia ElGamal

Influenciado diretamente pelo protocolo Diffie-Hellman, Taher Elgamal desenvolveu em 1984 um algoritmo de chave pública conhecido como algoritmo ElGamal. Embora esteja relacionado com a troca de chaves Diffie-Hellman que utiliza chave simétrica, o algoritmo ElGamal se baseia no uso de chaves assimétricas, tornando-o muito utilizado nos sistemas de assinaturas digitais.

Os componentes centrais nesse processo, assim como Diffie-Hellman, são o número primo q e α uma raiz primitiva módulo q.

Geração de chave

Alice gera um par de chaves privada/pública da seguinte maneira:

1. Alice escolhe um número primo q e uma raiz primitiva α módulo q.
2. Alice gera um inteiro aleatório m, tal que $1 < m < q-1$ e calcula $B \equiv \alpha^m (mod\, q)$.
3. A chave pública de Alice é (q, α, B) e sua chave privada é m.

Codificação criptográfica

Para criptografar um texto comum Bob realiza as seguintes etapas:

1. Bob obtém a chave pública de Alice (q, α, B).
2. Bob converte o texto comum em um inteiro M, tal que $0 < M < q-1$. Textos maiores são divididos em blocos representados por inteiros menores que q.
3. Bob escolhe aleatoriamente um expoente x, tal que $1 < x < q-1$ e calcula: $c \equiv \alpha^x (mod\ q)$ e $d \equiv MB^x (mod\ q)$.
4. O texto cifrado ElGamal enviado de Bob para Alice é o par (c, d).

Decodificação criptográfica

Para recompor o texto normal M, Alice deve aplicar os seguintes passos:

1. Alice calcula $(B^x)^{-1} \equiv c^{q-1-m} (mod\ q)$. De fato,

$$c^{q-1-m} \equiv c^{q-1}c^{-m} \equiv \underbrace{1 \cdot (\alpha^x)^{-m}}_{\text{Teorema: 2.7.4}} \equiv (\alpha^m)^{-x} \equiv (B^x)^{-1} (mod\ q).$$

2. Alice recupera o texto normal calculando $M \equiv d(B^x)^{-1} (mod\ q)$.

Exemplo 3.5.1. Bob deseja enviar a palavra OLÁ para Alice utilizando o sistema criptográfico ElGamal. Para fins de exemplo, considere $q = 29$ e uma raiz primitiva $\alpha = 3$, pois a $ord_{29}(3) = \phi(29)$. Alice escolhe um inteiro $m = 10$, tal que $0 < m < 28$, e calcula $B \equiv \alpha^m (mod\ q)$:

$$B \equiv 3^{10} (mod\ 29) \Rightarrow B = 5.$$

Portanto, a chave pública de Alice é $(29, 3, 5)$ e sua chave privada é 10. Antes que Bob criptografe a mensagem com a chave pública de Alice, é necessário converter a palavra OLÁ em números. Com auxílio da Tabela 4.1, obtém-se a seguinte sequência: 242110. Os blocos são divididos da seguinte forma:

$$24 - 21 - 10.$$

Considere $M_1 = 24, M_2 = 21$ e $M_3 = 10$. Bob escolhe aleatoriamente $x = 6$ e para cada M_i realiza os seguintes cálculos:

- $c \equiv \alpha^x (mod\ q) = 3^6 (mod\ 29) \Rightarrow c = 4$
- $d_1 \equiv M_1 B^x (mod\ q) = 24 \cdot 5^6 (mod\ 29) \Rightarrow d_1 = 1$
- $d_2 \equiv M_2 B^x (mod\ q) = 21 \cdot 5^6 (mod\ 29) \Rightarrow d_2 = 19$
- $d_3 \equiv M_3 B^x (mod\ q) = 10 \cdot 5^6 (mod\ 29) \Rightarrow d_3 = 27.$

A palavra cifrada é representada pelos pares $(4, 1), (4, 19)$ e $(4, 27)$, que serão enviados para Alice. Para decriptar a palavra, Alice calcula

$$(B^x)^{-1} \equiv 4^{29-1-10} \equiv 24 (mod\ 29)$$

e na sequência, calcula

- $M_1 \equiv d_1 (B^x)^{-1} (mod\ q) \equiv 1 \cdot 24 \equiv 24 (mod\ 29) \Rightarrow M_1 = 24$
- $M_2 \equiv d_2 (B^x)^{-1} (mod\ q) \equiv 19 \cdot 24 \equiv 21 (mod\ 29) \Rightarrow M_2 = 21$
- $M_3 \equiv d_3 (B^x)^{-1} (mod\ q) \equiv 27 \cdot 24 \equiv 10 (mod\ 29) \Rightarrow M_3 = 10$

recuperando a mensagem original:

$$24 - 21 - 10 \Rightarrow \text{OLÁ}.$$

Com isso, apresentamos as principais características dos sistemas de criptografia simétrica e assimétrica bem como alguns de seus respectivos algoritmos que servem de grande utilidade para manter a confidencialidade e autenticidade das comunicações. Whitfield Diffie e Martin Hellman buscaram a solução para resolver a questão da distribuição de chaves com a tecnologia de chave pública. Inspirado nesse método, o criptógrafo Taher Elgamal desenvolveu o algoritmo Elgamal com base nos mesmos conceitos matemáticos de Diffie-Hellman. Deste modo, a segurança do protocolo Diffie-Hellman e ElGamal tem como base o problema do logaritmo discreto, pois torna-se inviável calcular o logaritmo discreto módulo um número primo muito grande, em um tempo computacionalmente razoável.

4 A Criptografia RSA

Neste capítulo será apresentado o método de criptografia de chave pública RSA baseado em [2], onde todo o conhecimento visto no Capítulo 2 servirá de base para a realização dos cálculos, o porquê que o método funciona e o motivo pelo qual ele é seguro. É importante destacar que o processo de Criptografia RSA é constituído por duas chaves distintas relacionadas matematicamente: a chave pública *(chave de codificação)* e chave privada *(chave de decodificação)*. A primeira é utilizada para cifrar a mensagem, podendo ser revelada para qualquer pessoa e a segunda para decifrar a mensagem, devendo ser mantida em segredo pelo seu criador. Passamos agora a descrever o método.

4.1 Pré-Codificação

O primeiro passo do método RSA é a *pré-codificação* onde será feita a conversão de cada palavra do texto normal a ser codificado em uma sequência de números. Para isso, utiliza-se a Tabela 4.1:

Tabela 4.1: Conversão de letra para número.

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

Fonte: Elaborada pelo autor.

O número 99 é usado para substituir o espaço entre as palavras do texto a ser codificado. Por exemplo, *Mestrado em Matemática* fica convertido no seguinte número:

$$22142829271013249914229922102914221029181210.$$

A principal vantagem de representar cada letra por números com dois algarismos é evitar ambiguidades, por exemplo, se iniciar a sequência da Tabela 4.1 pelo número 1 e juntar as letras A (representada pelo número 1) e B (representada pelo número 2) o resultado será o número 12 que também representa a letra L, tornando assim um problema para decidir qual a escolha correta.

Antes dar passarmos para a próxima etapa, que é *a codificação*, precisamos determinar os parâmetros do sistema RSA que são representados por dois números primos

distintos, que vamos denotar por p e q. Em seguida, iremos calcular o produto desses dois números primos e o resultado chamaremos de n, ou seja, $n = p \cdot q$. Para finalizar esta etapa do processo de pré-codificação, iremos dividir o número produzido anteriormente em blocos de modo que cada bloco seja menor do que n. Para facilitar nossas contas, vamos escolher da seguinte maneira: $p = 11$ e $q = 17$. Portanto, $n = 187$. Logo, a sequência numérica fica dividida nos seguintes blocos:

22–142–82–9–27–101–32–49–91–42–29–92–2–102–91–42–2–102–91–8–12–10.

Observa-se que existem várias maneiras de quebrar em blocos a sequência gerada na conversão, porém, é imprescindível que não haja blocos formados com 0 no primeiro dígito, uma vez que não é possível diferenciar os blocos 029 e 29, razão pela qual traria problema na montagem da sequência recebida.

4.2 Codificação

Na etapa de *codificação* precisamos do valor n, que é o produto dos números primos p e q, e de um inteiro positivo e que seja inversível módulo $\phi(n)$, ou seja, o $MDC(e, \phi(n)) = 1$. Vimos na Seção 2.7 que $\phi(n)$ é a *função Totiente de Euler* e que se n é um número primo então $\phi(n) = n - 1$. Como $n = p \cdot q$, temos que:

$$\phi(n) = \phi(p \cdot q) = \phi(p) \cdot \phi(q) = (p - 1) \cdot (q - 1).$$

Dessa maneira, a *chave de codificação* será denotada pelo par (n, e) e na sequência codificaremos cada bloco separadamente. Denotaremos cada bloco por b e o bloco codificado por $C(b)$, que será o resto da divisão de b^e por n. Em termos de aritmética modular $C(b)$, é a forma reduzida de b^e módulo n:

$$C(b) \equiv b^e \ (mod\ n).$$

Voltando para o nosso exemplo, onde $p = 11$ e $q = 17$, logo $n = 187$ e $\phi(187) = 160$. Ainda precisamos encontrar o valor de e. Sabemos que $MDC(e, \phi(n)) = 1$. Portanto, $e = 3$, já que 3 é o menor primo que não divide 160.

Utilizando as ferramentas da aritmética modular e fazendo o uso de calculadora eletrônica, o bloco 22 é codificado conforme segue:

$$C(22) \equiv 22^3 \equiv 22^2 \cdot 22 \equiv 484 \cdot 22 \equiv 110 \cdot 22 \equiv 2420 \equiv 176 \ (mod\ 187).$$

Aplicando o mesmo raciocínio para os demais blocos obtemos a seguinte sequência:

176–131–92–168–48–118–43–26–148–36–79–20–8–170–148–36–8–170–148–138–45–65.

4.3 Decodificação

Para decodificar a mensagem serão necessários dois números: n e d. O número n já é nosso conhecido, pois utilizamos no processo de codificação, ou seja, $n = 187$. Portanto, resta achar d que nada mais é que o inverso multiplicativo de e módulo $\phi(n)$. O par (n, d) será chamado de *chave de decodificação*. Seja a um bloco da mensagem codificada, então $D(a)$ será o resultado do processo de decodificação, ou seja, é o resto

da divisão de a^d por n. Em termos de aritmética modular, $D(a)$ é a forma reduzida de a^d módulo n. Portanto,

$$D(a) \equiv a^d \ (mod\ n).$$

Observe que, se b é um bloco da mensagem original, então esperamos que $D(C(b)) = b$, ou seja, decodificando um bloco da mensagem codificada, presumimos encontrar o bloco correspondente da mensagem original.

Como $\phi(n)$ e e são conhecidos, basta aplicarmos o Algoritmo de Euclides Estendido para encontrar d. Então, de acordo com a Tabela 2.1, temos:

Resto	Quociente	x	y
160	$*$	1	0
3	$*$	0	1
1	53	$1 - 53 \cdot 0 = 1$	$0 - 53 \cdot 1 = -53$
0	3	$*$	$*$

$$160 \cdot 1 + 3 \cdot (-53) = 1.$$

Logo, o inverso de 3 módulo 160 é -53. Como vamos usar d como expoente de potências, é necessário que ele seja positivo. Portanto, $d = 160 - 53 = 107$.

Assim, para decodificar o bloco 176 da mensagem codificada basta calcularmos, através da computação algébrica, a forma reduzida de 176^{107} módulo 187:

$$D(176) \equiv 176^{107} \equiv 22 \ (mod\ 187).$$

Por fim, verifica-se que o processo de codificação de fato levou à mensagem original pré-codificada:

22–142–82–9–27–101–32–49–91–42–29–92–2–102–91–42–2–102–91–8–12–10.

4.4 Por que o RSA Funciona?

Para podermos mostrar que o método acima funciona precisamos garantir que decodificando um bloco codificado, obtemos de volta o bloco correspondente da mensagem original, ou seja, é suficiente provarmos o seguinte resultado:

$$DC(b) \equiv b \ (mod\ n).$$

Vamos mostrar que este resultado é válido utilizando como ferramenta na demonstração o Pequeno Teorema de Fermat.

Assim como foi definido nas Seções 4.2 e 4.3, $C(b) \equiv b^e \ (mod\ n)$ e $D(a) \equiv a^d \ (mod\ n)$ são respectivamente, as representações do bloco codificado b e do bloco decodificado a. Portanto, para representarmos a decodificação do bloco codificado temos que

$$DC(b) \equiv (b^e)^d \equiv b^{ed} (mod\ n). \tag{4.1}$$

Porém, d é o inverso 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 em 4.1

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

Como $n = pq$, onde p e q são primos distintos, calculamos a forma reduzida de b^{ed} módulo p e módulo q. O cálculo para ambos os primos é análogo. Desta forma, queremos achar a forma reduzida b^{ed} módulo p. Já vimos que

$$ed = 1 + k\phi(n) = 1 + k(p-1)(q-1),$$

logo

$$b^{ed} \equiv b \cdot (b^{p-1})^{k(q-1)} \ (mod\ p).$$

Vamos supor que p não divide b. Pelo Teorema 2.7.4, temos que $b^{p-1} \equiv 1 \ (mod\ p)$, e obtemos $b^{ed} \equiv b \ (mod\ p)$. Agora vamos supor que se p divide b, o fato de p ser primo nos leva a concluir que $b \equiv 0 \ (mod\ p)$ e a congruência é imediatamente verificada. Assim, $b^{ed} \equiv b \ (mod\ p)$ para qualquer valor de b e analogamente é possível provar que $b^{ed} \equiv b \ (mod\ q)$. Em outras palavras, $b^{ed} - b$ é divisível por p e q. Como p e q são primos distintos, temos que o $MDC(p, q) = 1$, donde pq divide $b^{ed} - b$, pelo Teorema 2.2.4. Como $n = pq$, concluímos que $b^{ed} \equiv b \ (mod\ n)$.

4.5 A Segurança da Criptografia RSA

Vimos que o sistema de criptografia RSA é o método de chave pública que se baseia na combinação de duas chaves, uma pública e outra privada. A primeira é usada para codificar a mensagem e pode ser divulgada para qualquer pessoa, enquanto que a segunda, é usada para decodificar a mensagem e mantida em segredo.

Como a chave pública é denotada pelo par (n, e), sendo n o resultado do produto dos números primos p e q, podemos afirmar que a confiabilidade do sistema de Criptografia RSA, em termos de segurança, está justamente direcionada na escolha do números primos p e q. Na prática, é essencial que esses números primos sejam suficientemente grandes a ponto que um supercomputador não seja capaz de conseguir fatorar n num pequeno intervalo de tempo [2].

Em 2010, um grupo internacional de cientistas conseguiram bater um recorde na fatoração de números primos. Utilizando uma rede interligada por centenas de computadores, eles fatoraram um número de 232 algarismos no produto de dois números primos, sinalizando naquela época a fragilidade em chaves criptográficas de 768 bits [13].

Recentemente, em fevereiro de 2020, um novo recorde para a criptografia RSA foi estabelecido por uma equipe internacional de cientistas da computação que fatoraram um número de 250 dígitos (829 bits) no produto de dois números primos, cada um com 125 dígitos. A mesma equipe havia estabelecido o recorde anterior em dezembro de 2019, fatorando o RSA-240 (795 bits). Segundo Nadia Heninger, professora de ciência da computação da Universidade da California em San Diego e membro da equipe, "é necessário obter registros computacionais para atualizar a segurança de parâmetros criptográficos e recomendações de tamanho de chave" [14].

Diante disso e com toda tecnologia disponível, recomenda-se que hoje, para garantir um bom nível de segurança para um futuro não muito distante, seja necessário chaves criptográficas com pelo menos 2048 bits, o equivalente a 617 algarismos.

4.6 A Criptografia Quântica

Com relação ao que foi apresentado na seção anterior, a questão da segurança do sistema RSA está diretamente relacionada com o tempo de fatoração que um computador leva para processar a decomposição de um número muito grande em fatores primos.

A partir da década de 1990, pesquisas relacionadas à computação quântica tiveram um papel determinante na busca pelo desenvolvimento da criação de um computador que tivesse um poder de processamento extremamente mais rápido do que os atuais.

Os computadores clássicos operam através do sistema numérico binário baseado em 2 dígitos que são representados pelos *bits* podendo conter apenas um valor, ou seja, (0 ou 1). Na computação quântica, o componente central é *qubit* (conhecido como bit quântico) com características atípicas em relação à computação clássica, pois os valores (0 e 1) além de poderem assumir o estado (0 ou 1), também podem assumir os estados (0 e 1) ao mesmo tempo. Com isso, os computadores atuais operam com processamento sequencial, enquanto que na computação quântica esse processamento é simultâneo [15].

Dessa forma, os sistemas de criptografias tradicionais utilizam técnicas apoiadas nos estudos da matemática para garantir a segurança, ao passo que a criptografia quântica se sustenta nas bases das propriedades das leis da física e da mecânica quântica.

Um artigo publicado em 1994 pelo matemático Peter Shor, conhecido como algoritmo de Shor, é sem dúvida o principal resultado que impulsionou a pesquisa em computação e criptografia quântica.

De acordo com a Tabela 4.2 a seguir, é possível comparar a diferença de tempo entre o algoritmo clássico com o algoritmo de Shor em relação ao tamanho do comprimento do número a ser fatorado [16].

Tabela 4.2: Comparação do tempo de fatoração entre o algoritmo clássico e o algoritmo de Shor.

Comprimento do número a ser fatorado em bits	**Tempo de fatoração por algoritmo clássico**	**Tempo de fatoração com o algoritmo de Shor**
512	4 dias	34 segundos
1024	100 mil anos	4,5 minutos
2048	100 mil bilhões de anos	36 minutos
4096	100 bilhões de quatrilhões de anos	4,8 horas

Fonte: Adaptado de [16].

É possível observar na Tabela 4.2 que o algoritmo quântico é capaz de fatorar números suficientemente grandes em fatores primos num intervalo de tempo muito menor em comparação com o algoritmo clássico (por exemplo o RSA) levaria para fatorar.

Transpondo esses números apresentados para a nossa realidade, mesmo hoje, com toda tecnologia dos processadores atuais não seria muito diferente.

Diante desses dados, Artur Ekert, professor de física quântica do Instituto de Matemática da Universidade de Oxford, e demais colegas da área da criptografia quântica, observaram que os sistemas criptográficos contemporâneos possuem suas vulnerabilidades, e assim que o primeiro computador quântico for ligado, momentos depois nenhuma mensagem criptografada deixará de ser secreta, excluindo de vez os métodos criptográficos tradicionais [16].

Nesse contexto, se por um lado a computação quântica representa um novo degrau nos avanços das pesquisas científicas com base na física quântica, por outro os impactos provenientes dessa nova tecnologia podem ameaçar de vez toda segurança dos sistemas criptográficos atualmente utilizados, fazendo que qualquer sistema baseado em RSA, Diffie-Hellman, ElGamal, possa ser quebrado em tempo hábil.

5 Assinatura e Certificado Digital

Neste capítulo, apresentaremos a assinatura digital sendo um dos serviços mais utilizados no mundo eletrônico capaz de substituir a tradicional assinatura de próprio punho, garantindo assim a sustentabilidade no consumo de papel e a autenticidade das partes envolvidas nas transações realizadas por meio de documentos eletrônicos. Nesse processo, destacamos a importância da função hash para garantir a integridade das mensagens e a certificação digital que permite comprovar a autenticidade das assinaturas digitais no meio tecnológico.

5.1 Assinatura Digital

À medida que a tecnologia avança pessoas e empresas buscam cada vez mais os recursos disponíveis na internet principalmente aqueles voltados para agilizar os trâmites das informações contidas em documentos eletrônicos que trafegam pela internet. Por essa razão, as transações utilizadas por meios de papéis são deixadas de lado para dar lugar aos documentos eletrônicos gerando menos gastos e outros tipos de despesas.

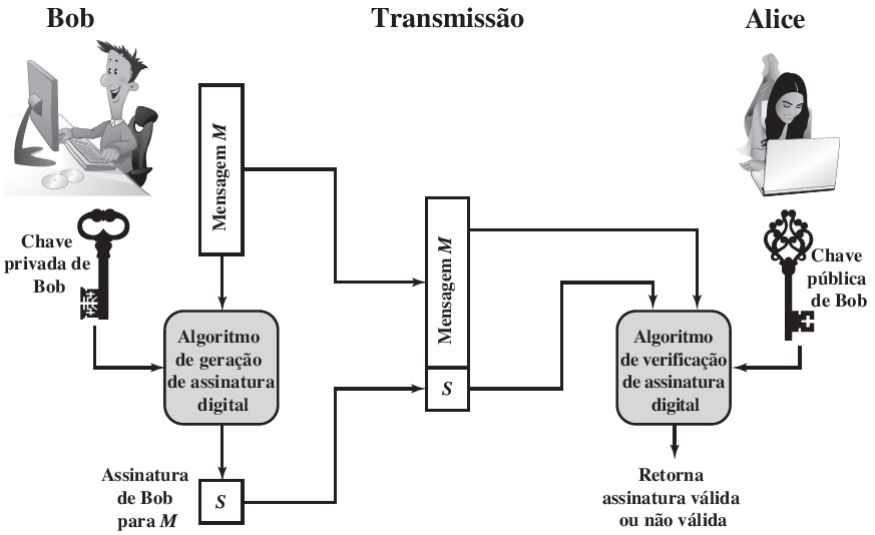
Diante dessa burocracia excessiva, surge a assinatura digital como um mecanismo que utiliza algoritmos criptográficos capazes de garantir a autenticidade nos meios de comunicação permitindo que o criador da mensagem seja capaz de anexar um código funcionando como uma assinatura [10].

A criptografia de chave pública que tem como característica fundamental a utilização de duas chaves distintas, oferece uma mudança radical nos sistemas de criptografia contornando assim, dois dos problemas mais difíceis associados à criptografia simétrica: a distribuição de chaves e as assinaturas digitais [10].

> Se o uso da criptografia tivesse que se tornar comum, não apenas nas situações militares, mas para fins comerciais e particulares, então as mensagens e documentos eletrônicos precisariam do equivalente das assinaturas usadas nos documentos em papel. Ou seja, poderia ser criado um método para estipular, para a satisfação de todas as partes, que uma mensagem digital foi enviada por determinada pessoa? ([10], p. 201).

Nesse ponto, o RSA é um dos algoritmos criptográficos mais utilizados para gerar assinaturas digitais, pois além da sua utilização no processo normal para codificação com a chave pública e decodificação com a chave privada, o método permite que codificando-se com a chave privada, o processo resulte na recuperação da mensagem ao decodificar com a chave pública [11]. Na Figura 5.1 ilustra-se esse procedimento:

Figura 5.1: Modelo genérico de assinatura digital utilizando algoritmo RSA.



Fonte: Adaptado de [10].

Observamos na Figura 5.1 que Bob utiliza um algoritmo de geração de assinatura digital para assinar a mensagem M com sua chave privada. Em seguida, Bob envia para Alice a mensagem M acompanhada de uma assinatura digital S. Ao chegar no destino, Alice verifica a assinatura S usando um algoritmo de verificação utilizando a chave pública de Bob. A assinatura digital será válida, ou seja, autêntica, se Alice ao decodificar a assinatura de Bob obter um resultado satisfatório, do contrário será não válida, indicando que não foi Bob quem assinou a mensagem ou que durante a transmissão da mensagem ocorreu algum tipo de adulteração no conteúdo.

5.2 Funções de Hash

Devido a criptografia de chave pública ser considerada lenta, conforme a Tabela 3.1, isso faz com que não seja viável codificar uma mensagem inteira para usar como assinatura digital conforme a Figura 5.1. Neste caso, utilizamos uma função *hash* para codificar apenas um representante da mensagem, isto é, um resumo de mensagem *(message digest)* [12].

A função hash recebe como entrada uma mensagem de tamanho variável M e produz um valor de tamanho fixo, que representa o conteúdo dessa mensagem. Com o valor fixo gerado pela função hash é possível assegurar a integridade de qualquer mensagem, pois o resumo gerado pelo destinatário por uma função hash terá que ser igual ao resumo gerado pelo remetente [10].

Dois dos algoritmos de função hash fortemente recomendados pela comunidade criptográfica e com aplicação comercial são:

- **MD:** O *Message Digest (MD)* foi desenvolvido por Ron Rivest na década de 90. As versões mais comuns da família MD são representadas pelas funções MD2, MD4 e MD5, que recebem mensagens de tamanho variável produzindo um hash fixo de 16 bytes (ou 128 bits) [12].

- **SHA:** O *Secure Hash Algorithm (SHA)* foi desenvolvido pelo National Institute of Standards and Technology (NIST) em 1993. Com base na função hash MD4, o SHA possui algumas variações e tamanhos de valores hash diferentes produzindo resumos mais longos comparados com os algoritmos MD, como SHA-1 (160 bits), SHA-224 (224 bits), SHA-256 (256 bits) e o SHA-512 (512 bits) [10].

Vamos considerar duas situações em que Bob transmite duas mensagens para Alice utilizando o mesmo algoritmo de função hash, SHA-1:

1. Mensagem: Alice, compre 1 kg de arroz.
   SHA-1: b9 6e d3 21 21 a3 b2 dd ee 8b 20 e0 ba 48 17 6c bf 9c 48 e9

2. Mensagem: Alice, compre 2 kg de arroz.
   SHA-1: f5 28 cf 5e 8a 19 36 6d 7d d2 83 58 c2 72 a8 87 05 7c 1c 46

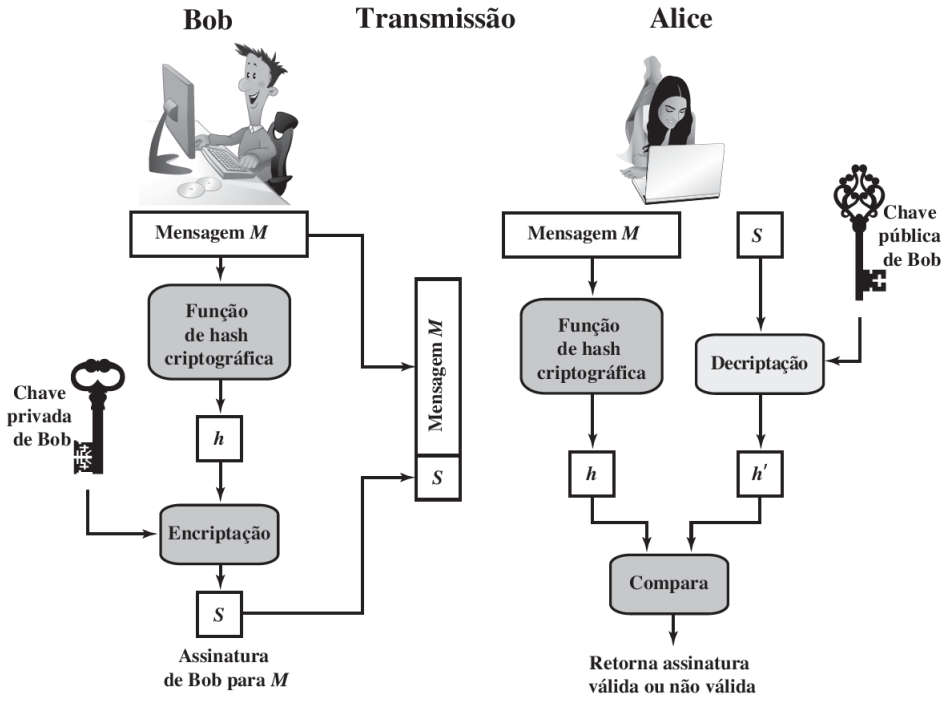
Observe que as mensagens possuem a mesma quantidade de caracteres iguais a 28 bytes (224 bits) de comprimento e seus resumos possuem 20 bytes (160 bits). Desse modo, independentemente do tamanho da mensagem ser maior ou menor do que 20 bytes, o resultado da função SHA-1 será 20 bytes. Uma outra característica da função hash é que embora as mensagens sejam quase idênticas, a menos de um caractere, o resultado do hash será totalmente diferente um do outro.

Outras propriedades importantes da função hash são, conforme [6]:

- Simples (eficiente, rápida) de gerar o hash de qualquer mensagem;
- Impraticável determinar a mensagem conhecendo o hash de entrada (funções de sentido único);
- Impraticável encontrar duas mensagens com o mesmo valor hash (o termo técnico que descreve essa situação é denominado de *colisão*).

Para solucionar a dificuldade apresentada na Figura 5.1, introduzimos a função hash no processo de assinatura digital conforme ilustrado na Figura 5.2 a seguir:

Figura 5.2: Assinatura digital com função hash.



Fonte: Adaptado de [10](p. 311).

Observando a Figura 5.2, Bob resume a mensagem M utilizando uma função hash e então codifica o resumo h com sua chave privada gerando a sua assinatura digital S. Bob transmite a mensagem assinada digitalmente para Alice que separa em duas partes. Alice utiliza a mesma função hash para resumir a mensagem M e obtém h. Em seguida, Alice decodifica o resumo S utilizando a chave pública de Bob produzindo h'. Por fim, Alice compara os dois resumos, h e h', se forem iguais, a integridade da mensagem transmitida é fidedigna.

Desta forma, é possível identificarmos as principais características da aplicação da assinatura digital na criptografia de chave pública, conforme [12]:

- Quando Alice verifica, certificando-se que de fato a mensagem veio de Bob e não de outra pessoa fingindo ser ele, chamamos de **autenticação**.
- Quando Alice realiza o processo de verificação comparando os resumos h e h', chamamos de **verificação de integridade de dados**, de que a mensagem não sofreu nenhum tipo de alteração durante a transmissão.
- E por fim, Bob não pode negar que foi ele o autor da mensagem codificada, isso é chamado de **não repúdio**.

5.3 Assinatura Digital ElGamal

Nesta seção, vamos apresentar o esquema de assinatura digital proposto por Taher Elgamal, publicado em um artigo em 1985, semelhante ao criptossistema ElGamal visto na Seção 3.5. Os elementos centrais também são o número primo q e α uma raiz

primitiva módulo q. Na próxima seção, veremos o algoritmo de assinatura digital DSA, uma variação da assinatura digital ElGamal adotada pelo NIST. Para compreender o método, vamos utilizar a referência [10].

Geração de chave

Bob gera um par de chaves privada/pública da seguinte maneira:

1. Bob gera um inteiro aleatório m, tal que $1 < m < q-1$ e calcula $B \equiv \alpha^m (mod\ q)$.
2. A chave pública de Bob é (q, α, B) e sua chave privada é m.

Assinatura da mensagem

Para assinar a mensagem M Bob calcula o hash γ, tal que $0 \leq \gamma \leq q-1$ e realiza as seguintes etapas:

1. Bob escolhe um inteiro aleatório K tal que $1 < K < q-1$ e $MDC(K, q-1) = 1$.
2. Bob calcula $S_1 \equiv \alpha^K (mod\ q)$ e o inverso de K módulo $q-1$.
3. Bob calcula $S_2 \equiv K^{-1}(\gamma - mS_1)\ mod\ (q-1)$.
4. O par (S_1, S_2) representa a assinatura.

Verificação da assinatura

Alice pode verificar a assinatura de Bob da seguinte maneira:

1. Calcula $V_1 \equiv \alpha^\gamma (mod\ q)$.
2. Calcula $V_2 \equiv B^{S_1} {S_1}^{S_2} (mod\ q)$.

Nesse caso, a assinatura digital de Bob será válida se $V_1 = V_2$. Portanto, vamos mostrar que vale a igualdade.

$$\begin{aligned} V_1 &= V_2 \\ \alpha^\gamma (mod\ q) &= B^{S_1} {S_1}^{S_2} (mod\ q) \\ \alpha^\gamma (mod\ q) &= (\alpha^m)^{S_1} (\alpha^K)^{S_2} (mod\ q) \\ \alpha^\gamma (mod\ q) &= \alpha^{mS_1 + KS_2} (mod\ q). \end{aligned}$$

Como α é uma raiz primitiva módulo q, o Teorema 2.8.2 implica

$$\gamma \equiv (mS_1 + KS_2)\ mod\ (q-1).$$

Substituindo S_2 obtemos:

$$\begin{aligned} \gamma &\equiv mS_1 + K \cdot [K^{-1}(\gamma - mS_1)]\ mod\ (q-1) \\ \gamma &\equiv mS_1 + K \cdot K^{-1}(\gamma - mS_1)\ mod\ (q-1) \\ \gamma &\equiv mS_1 + \underbrace{K \cdot K^{-1}}_{1}(\gamma - mS_1)\ mod\ (q-1) \\ \gamma &\equiv mS_1 + \gamma - mS_1\ mod\ (q-1) \\ \gamma &\equiv \gamma\ mod\ (q-1). \end{aligned}$$

Portanto, a assinatura digital de Bob está verificada.

Exemplo 5.3.1. Como no Exemplo 3.5.1, Bob escolhe $q = 29$ e uma raiz primitiva $\alpha = 3$ gerando um par de chaves da seguinte maneira. Bob escolhe $m = 18$ e calcula $B \equiv \alpha^m (mod\ q)$:

$$B \equiv 3^{18} (mod\ 29) \Rightarrow B = 6.$$

Portanto, a chave pública de Bob é $(29, 3, 6)$ e sua chave privada é 18. Vamos supor que Bob queira assinar um documento M, que possui valor hash $\gamma = 8$. Ele escolhe um inteiro aleatório $K = 9$, de modo que o $MDC(9, 28) = 1$, e calcula $S_1 \equiv \alpha^K (mod\ q)$:

$$S_1 \equiv 3^9 (mod\ 29) \Rightarrow S_1 = 21.$$

Bob calcula o inverso de K módulo $q - 1$ e obtém $K^{-1} = 25$. Em seguida calcula $S_2 \equiv K^{-1}(\gamma - mS_1)\ mod\ (q - 1)$:

$$S_2 \equiv 25(8 - 18 \cdot 21)\ (mod\ 28) \Rightarrow S_2 = 18.$$

Alice deseja verificar a assinatura de Bob e calcula $V_1 \equiv \alpha^\gamma (mod\, q)$ e $V_2 \equiv B^{S_1} {S_1}^{S_2} (mod\, q)$:

$$\begin{aligned} V_1 \equiv 3^8 (mod\ 29) &\Rightarrow V_1 = 7; \\ V_2 \equiv 6^{21} 21^{18} (mod\ 29) &\Rightarrow V_2 = 7. \end{aligned}$$

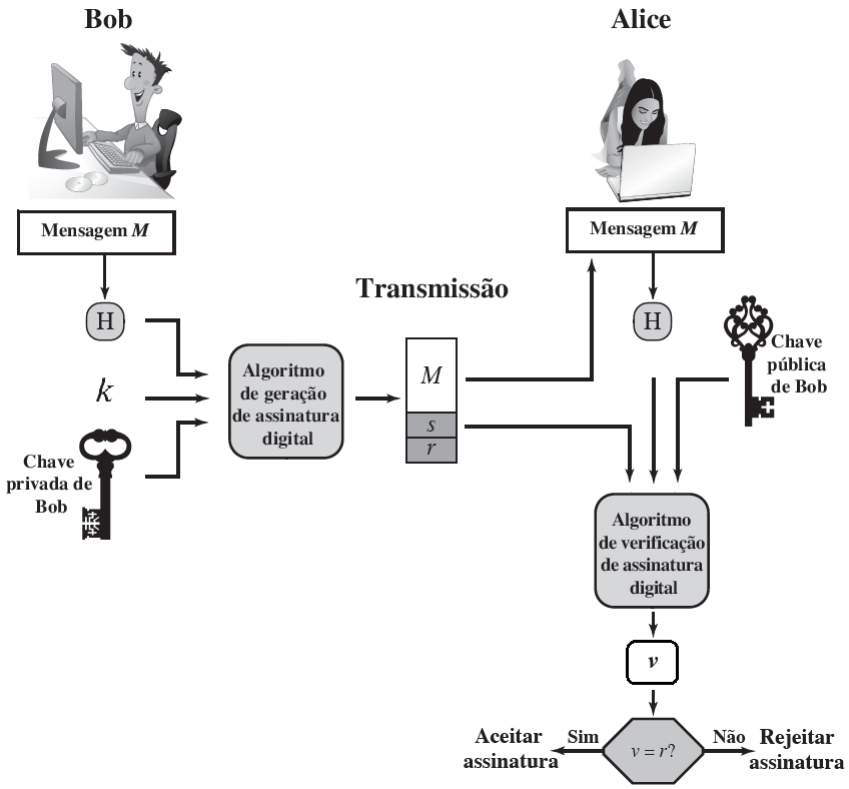
Portanto, a assinatura de Bob é válida.

5.4 Assinatura Digital DSA

Apresentado pela primeira vez em 1991, o *Digital Signature Algorithm (DSA)* é um algoritmo de chave pública com base no chamado *Digital Signature Standard (DSS)*, proposto pelo NIST e tornou-se o método de assinatura digital padrão aceito pelo governo americano. David Kravitz, que trabalhava para a *National Security Agency (NSA)*, desenvolveu o DSA com base nos trabalhos de Taher Elgamal e Claus Schnorr. A segurança do método deriva da complexidade computacional de obter logaritmos discretos [12]. Nesta seção, serão utilizadas as referências [10], [11], [12] e [17].

O mecanismo de assinatura digital DSA, embora não seja projetado para criptografar dados, tem como base a técnica de chave pública e sua principal função é oferecer a assinatura digital. Para descrevermos os passos necessários vamos considerar a Figura 5.3 a seguir:

Figura 5.3: Assinatura digital DSA



Fonte: [10].

Conforme a Figura 5.3, Bob gera um hash (H) da mensagem (M). O DSA utiliza o *Secure Hash Algorithm (SHA)* que serve como entrada para o algoritmo de geração da assinatura digital, juntamente com um número aleatório k e a chave privada de Bob, resultando na assinatura representada pelos componentes s e r. Do outro lado, Alice recebe a mensagem (M) e os componentes s e r, e gera o código hash (H) de (M). O código hash gerado pela Alice juntamente com o componente s e a chave pública de Bob, são processadas pelo algoritmo de verificação digital, gerando um novo componente v. Se v for igual ao componente r, comprova-se a validade da assinatura de Bob, do contrário indica que houve adulteração da mensagem original rejeitando a assinatura do emissor.

Agora, vejamos os detalhes do algoritmo.

Geração de chave

1. Bob escolhe um número primo p de tamanho L bits, onde $2^{L-1} < p < 2^L$ e q um módulo primo de tamanho N bits, onde $2^{N-1} < q < 2^N$, de modo que o problema do logaritmo discreto em $\mathbb{Z}_p$ seja considerado computacionalmente intratável. Além disso, $q \mid (p-1)$.

2. Então, Bob escolhe uma raiz primitiva h módulo p e calcula

$$g \equiv h^{(p-1)/q} (mod\ p), \text{ com } g > 1.$$

3. Bob escolhe um inteiro aleatório x com $0 < x < q$ e calcula

$$y \equiv g^x (mod\ p).$$

4. Finalmente, a chave pública de Bob é (p, q, g, y) e sua chave privada é x.

Geração de assinatura

1. A partir da mensagem M de Bob é gerado um hash $H(M)$, o qual passa pelo algoritmo de geração de assinatura digital, que requer a chave privada de Bob e um número aleatório k com $0 < k < q$.

2. Na saída do algoritmo, obtém-se um par de números r e s que são calculados de acordo com as seguintes equações:

$$r \equiv [g^k (mod\ p)](mod\ q) \quad \text{e} \quad s \equiv k^{-1}[H(M) + xr](mod\ q).$$

3. k^{-1} é inverso de k módulo q.

4. Portanto, a assinatura da mensagem M de Bob é (r, s).

Verificação

1. Quando Alice recebe a mensagem M de Bob junto com a assinatura (r, s), ela aplica a função hash $H(M)$ a qual será submetida ao algoritmo de verificação conjuntamente com a chave pública y de Bob e o componente s.

2. Desse processo, origina-se um novo componente v, calculado da seguinte forma:

   - $w \equiv s^{-1}(mod\ q) \Rightarrow s^{-1}$ é inverso de s módulo q.
   - $u_1 \equiv [H(M) \cdot w](mod\ q)$
   - $u_2 \equiv (r \cdot w)(mod\ q)$
   - $v \equiv [(g^{u_1} \cdot y^{u_2})\ mod\ p](mod\ q)$

3. Se $v = r$ então comprova-se a validade da assinatura de Bob.

O objetivo agora é fornecer uma prova de que se $v = r$ então a assinatura DSA é válida. Para isso, considere $g \equiv h^{\frac{(p-1)}{q}}(mod\ p)$ e h um inteiro positivo menor que p. Multiplicando-se ambos os lados da congruência por q, obtemos:

$$g^q \equiv h^{\frac{q(p-1)}{q}} \equiv h^{(p-1)} \equiv 1(mod\ p),$$

devido ao Teorema 2.7.4 (Pequeno Teorema de Fermat). Como q é um número primo e $g^q \equiv 1(mod\ p)$, então q é a ordem de g módulo p. Considerando que

$$s \equiv k^{-1}[H(M) + xr](mod\ q)$$

e multiplicando-se os dois lados da congruência por $(k \cdot s^{-1})$, temos

$$\begin{aligned} (k \cdot s^{-1})s &\equiv (k \cdot s^{-1})k^{-1}[H(M) + xr](mod\ q) \\ k &\equiv s^{-1}[H(M) + xr](mod\ q) \\ k &\equiv [s^{-1}H(M) + s^{-1}xr](mod\ q). \end{aligned}$$

Como $w = s^{-1}(mod\ q)$, basta substituirmos na última congruência e obtemos:

$$k \equiv [H(M)w + xrw](mod\ q).$$

Como a ordem de g módulo p é q, e $k \equiv [wH(M) + wxr](mod\ q)$, o Teorema 2.8.2 garante que

$$\begin{aligned} g^k &\equiv g^{[H(M)w+xrw]}(mod\ q) \\ g^k &\equiv g^{H(M)w} \cdot g^{xrw}(mod\ q) \\ g^k &\equiv (g^{H(M)w} \cdot y^{rw}\ mod\ p)\ (mod\ q) \\ g^k &\equiv (g^{u_1} \cdot y^{u_2} mod\ p)\ (mod\ q). \end{aligned}$$

Dessa última congruência provamos que $v = r$, pois $v = [(g^{u_1} \cdot y^{u_2})\ mod\ p](mod\ q)$ e $r = [g^k(mod\ p)](mod\ q)$.

Exemplo 5.4.1. Vamos supor que Bob queira assinar uma mensagem (M). Para isso, Bob escolhe dois números primos $p = 7$ e $q = 3$ de modo que $q \mid (p-1)$. Em seguida, Bob determina uma raiz primitiva $h = 5$ e calcula

$$g \equiv h^{(p-1)/q}(mod\ p) \equiv 5^{(7-1)/3}(mod\ 7) \Rightarrow g = 4.$$

Bob escolhe um inteiro aleatório $x = 5$ e calcula

$$y \equiv g^x(mod\ p) \equiv 4^5(mod\ 7) \Rightarrow y = 2.$$

Finalmente, a chave pública de Bob é $(7, 3, 4, 2)$ e sua chave privada é (5).
A partir da mensagem (M), Bob gera um hash $H(M) = 3$ e um número aleatório $k = 2$ com inverso multiplicativo $k^{-1} = 2$, os quais, juntamente com a chave privada, serão processados pelo algoritmo de geração de assinatura digital resultando nos componentes r e s, que são calculados da seguinte maneira:

$$\begin{aligned} r \equiv [g^k(mod\ p)](mod\ q) \equiv [4^2(mod\ 7)](mod\ 3) &\Rightarrow r = 2; \\ s \equiv K^{-1}[H(M) + xr](mod\ q) \equiv 2[3 + 5 \cdot 2](mod\ 3) &\Rightarrow s = 2. \end{aligned}$$

Portanto, a assinatura da mensagem (M) é $(r, s) = (2, 2)$. Para verificar a assinatura de Bob, Alice aplica a função hash na mensagem a qual será submetida ao algoritmo de verificação conjuntamente com a chave pública de Bob e o componente s. O resultado dará origem ao componente v e o cálculo é feito da seguinte maneira:

1. $w \equiv s^{-1}(mod\ q)$
   O inverso multiplicativo de s módulo q é $s^{-1} = 2$.
   $w \equiv 2(mod\ 3)$
   $w = 2$.

2. $u_1 \equiv [H(M) \cdot w](mod\ q)$
   $u_1 \equiv 3 \cdot 2(mod\ 3)$
   $u_1 = 0$

3. $u_2 \equiv (r \cdot w)(mod\ q)$
   $u_2 \equiv 2 \cdot 2(mod\ 3)$
   $u_2 = 1$

4. $v \equiv [(g^{u_1} \cdot y^{u_2})\ mod\ p](mod\ q)$
   $v \equiv [(4^0 \cdot 2^1)\ mod\ 7](mod\ 3)$
   $v = 2$

Como $v = r$, a assinatura de Bob é válida.

5.5 Certificado Digital

Vimos na Seção 3.2 que a chave pública pode ser livremente divulgada. Apesar disso, caso não seja possível constatar a quem de fato ela pertence, corre o risco de um impostor substituí-la por uma outra chave pública falsa comprometendo, assim, toda a comunicação. Um dos recursos que podemos usar para tentar impedir que isso aconteça é a utilização dos certificados digitais.

Nesta seção, apresentamos uma das mais importantes aplicações das assinaturas digitais, o *certificado digital* que foi sugerido pela primeira vez em 1978, por Kohnfelder, como uma tecnologia para autenticar as chaves públicas. Essencialmente, um certificado digital consiste em uma chave pública mais o nome do seu respectivo dono, com o bloco inteiro assinado por uma Autoridade Certificadora (AC) [10].

Além de funcionar como uma identidade digital e de servir como um meio seguro de divulgação da chave pública, com o certificado digital é possível saber se a chave pública pertence ou não a uma pessoa física ou jurídica, capaz de garantir a autenticidade, a confidencialidade, a integridade e não repúdio quanto às transações feitas pela internet ou qualquer outro meio de serviço digital [12].

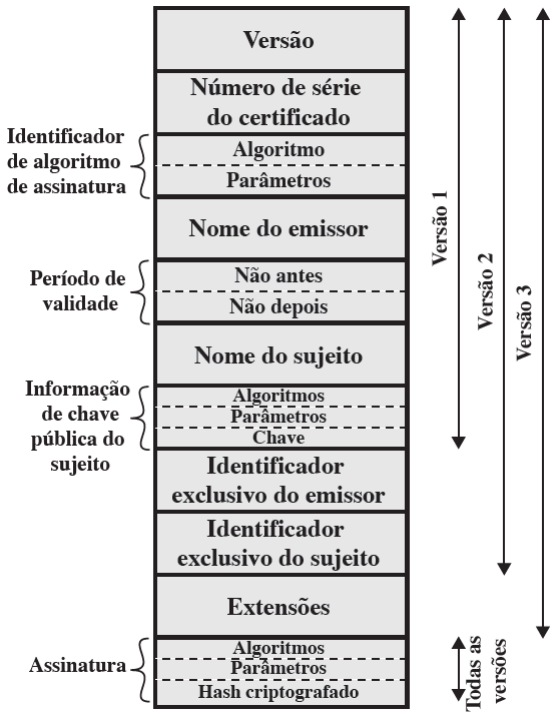
O certificado digital também proporciona segurança nas transações comerciais realizadas na internet, impedindo que pessoas não autorizadas tenham acesso, como por exemplo, a senhas de cartões de crédito, informações bancárias, números de documentos ou qualquer outro tipo de dados sigilosos, evitando assim aborrecimentos e problemas futuros.

Uma maneira de mostrar como os certificados digitais são utilizados é através das transações bancárias praticadas em sites de bancos chamados de *Internet Banking*. Quando o cliente de um banco acessa a sua conta virtual através de um navegador de internet, seja para fazer uma consulta de saldo ou, até mesmo, para realizar uma transferência de valores, o site do banco envia um certificado digital assegurando ao cliente que as operações serão todas criptografadas, proporcionando uma conexão segura.

Os certificados digitais são formados por um conjunto padronizado de informações e o mais aceito e utilizado é o padrão $X.509$ Versão 3 da *International Telecommunications Union (ITU)* que faz parte da série $X.500$. Publicado inicialmente em 1988, o padrão $X.509$ utiliza criptografia de chave pública e assinaturas digitais, além de ser amplamente usado em S/MIME, IPSec e SSL/TLS. Esse padrão recomenda o uso do algoritmo RSA e uma função *hash* no esquema de assinatura digital não especificando qual algoritmo de *hash* a ser utilizado [10].

A Figura 5.4 a seguir, ilustra o padrão de um certificado $X.509$, conforme [10]:

Figura 5.4: Estrutura do certificado X.509.



Fonte: [10].

Podemos observar que a Figura 5.4 contém uma série de informações de acordo com o tipo da versão do certificado, tais como:

- **Versão:** esse campo diferencia entre as versões sucessivas do formato do certificado, como Versão 1, Versão 2 e Versão 3.
- **Número de série:** é um número sequencial exclusivo para cada certificado emitido por uma AC.
- **Identificador do algoritmo de assinatura:** esse campo indica o algoritmo usado para assinar o certificado.
- **Nome do emissor:** é o nome da AC que criou e assinou o certificado.
- **Período de validade:** esse campo contém dois valores de data/hora do período de validade do certificado.
- **Nome do sujeito:** é o nome do usuário a quem esse certificado se refere.
- **Informação de chave pública do sujeito:** esse campo contém o algoritmo, os parâmetros e a chave pública do sujeito.
- **Identificador exclusivo do emissor:** esse campo é opcional e exclusivo da Versão 2 e da Versão 3. Serve para identificar exclusivamente a AC emissora caso o nome $X.500$ tenha sido reutilizado para entidades diferentes.

- **Identificador exclusivo do sujeito:** esse campo é opcional e exclusivo da Versão 2 e da Versão 3. Serve para identificar exclusivamente o sujeito caso o nome $X.500$ tenha sido reutilizado para entidades diferentes.

- **Extensões:** esse campo foi implementado na Versão 3 e contém um conjunto de uma ou mais informações de extensões.

- **Assinatura:** esse contém o código hash dos outros campos encriptados com a chave privada da AC juntamente com o identificador do algoritmo de assinatura.

Desse modo, para mantermos uma estrutura de emissão de certificados digitais confiáveis conforme os preceitos legais previstos em lei, é necessário a criação de um conjunto de regras e serviços que serão gerenciados por uma Infraestrutura de Chaves Públicas (ICP), que será apresentada na próxima seção.

5.6 Infraestrutura de Chave Pública

A Infraestrutura de Chaves Públicas (ICP) que, em inglês, significa *Public Key Infrastructure (PKI)*, corresponde a um conjunto de políticas e metodologias empregadas para gerenciar a autenticidade das assinaturas digitais. No Brasil, a (ICP-Brasil)-Infraestrutura de Chaves Públicas Brasileira, instituída pelo governo federal através da Medida Provisória nº 2.200-02, de 24 de agosto de 2001, é a norma responsável por regulamentar a validade jurídica dos certificados digitais em documentos eletrônicos no âmbito nacional. A ICP-Brasil é formada por um Comitê Gestor e Autoridades Certificadoras, conforme a seguinte hierarquia pública [18]:

1. **Autoridade Certificadora Raiz (AC Raiz)** é representada pelo Instituto Nacional de Tecnologia da Informação (ITI) sendo a primeira autoridade da cadeia de certificação. Além de supervisionar e realizar auditorias nos processos de certificação, a AC Raiz é responsável pelo credenciamento e descredenciamento dos demais participantes da cadeia [18].

2. **Autoridade Certificadora (AC)** é a entidade de confiança, ou seja, o "terceiro confiável" no processo de emissão dos certificados digitais para outras entidades, pessoas ou empresas, vinculando os pares das chaves criptográficas com o respectivo titular de acordo com as normas definidas pela ICP-Brasil [18].

3. **Autoridade de Registro (AR)** é responsável por atuar como órgão de apoio à Autoridade Certificadora, verificando as informações que são fornecidas pelos requisitantes do certificado, podendo exigir que estes compareçam pessoalmente à AR para assegurar autenticidade das informações [19].

Uma visão mais detalhada da estrutura hierárquica definida pela ICP-Brasil pode ser consultada em [20].

As ACs também disponibilizam um serviço conhecido como Diretório de Certificado ou localização central de armazenamento, que permite aos usuários realizarem consultas ou até mesmo navegarem em determinados diretórios de outros usuários sem que esses autorizem antecipadamente, auxiliando assim na administração e distribuição de certificados digitais [12].

Vimos na Figura 5.4 que os certificados digitais são criados de modo que sejam válidos e utilizáveis de acordo com o tempo previsto mencionado no campo Período de Validade. No entanto, certas condições fazem com que os certificados digitais ainda que não tenham expirados sejam invalidados por uma AC, seja porque a chave privada tenha sido danificada, ou porque tenha sido descoberto um erro por parte da AC, ou até mesmo pelo fato do funcionário de uma empresa não ser mais o detentor da chave.

O método mais comum para que as ACs publiquem os certificados digitais que são revogados ainda em vigor para as partes verificadoras é através do uso da Lista de Certificados Revogados (LCR) que, em inglês, significa *Certificate Revocation List - CRL*. A LCR é formada por uma estrutura de dados assinada, em geral, pela mesma AC que a emitiu, contendo uma lista de data/hora dos certificados revogados [12].

A ICP-Brasil classifica os certificados digitais de acordo com a forma de armazenamento e quanto ao prazo de validade através de duas classes distintas:

1. A classe **A** está voltada para fins de identificação e autenticação do usuário, sendo possível utilizá-la para assinar documentos eletrônicos e garantir a validade das transações em meio digital.

2. A classe **S** tem por finalidade assegurar a proteção de arquivos que sejam confidenciais para os usuários voltada para atividades de natureza sigilosa.

Além disso, essas classes são subdivididas respectivamente em quatro tipos: **A1, A2, A3, A4** e **S1, S2, S3, S4**. As principais características dessas categorias são apontadas por [21], a seguir:

- **A1** e **S1:** tipo de certificado em que as chaves são geradas com tamanho mínimo de 1024 bits através de um *software* em que o armazenamento pode ser feito por meio de dispositivos como HDs ou pendrives. O prazo de validade é de um ano;

- **A2** e **S2:** tipo de certificado em que as chaves são geradas com tamanho mínimo de 1024 bits através de um *software* em que o armazenamento pode ser feito por meio de cartão inteligente com *chip* ou *token* (dispositivo semelhante a um pendrive). O prazo de validade é de dois anos;

- **A3** e **S3:** tipo de certificado em que as chaves são geradas com tamanho mínimo de 1024 bits através de um *hardware* em que o armazenamento pode ser feito por meio de cartão inteligente com *chip* ou *token* (dispositivo semelhante a um pendrive). O prazo de validade é de cinco anos;

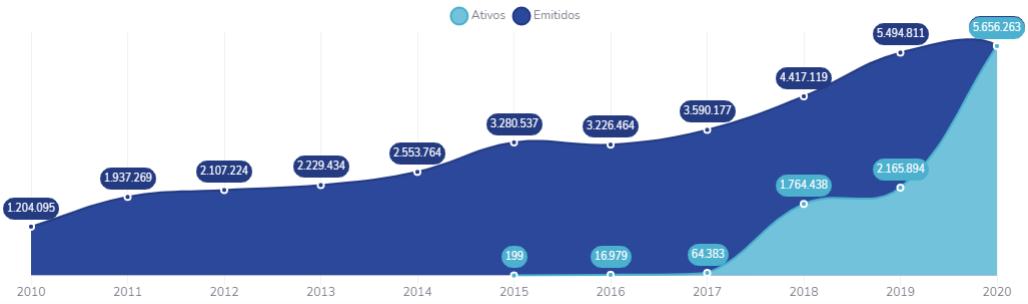
- **A4** e **S4:** tipo de certificado em que as chaves são geradas com tamanho mínimo de 2048 bits através de um *hardware* em que o armazenamento pode ser feito por meio de cartão inteligente com *chip* ou *token* (dispositivo semelhante a um pendrive). O prazo de validade é de seis anos;

Os tipos de certificados digitais mais vendidos pelas ACs correspondem aos modelos A1 e A3, sendo que o primeiro pode ser armazenado no computador do usuário através do navegador de internet, enquanto que o segundo é armazenado em cartões inteligentes *smartcards* ou *tokens* protegidos por senha [22].

Diante do cenário que estamos vivendo com a pandemia do novo coronavírus que teve início aqui no Brasil em março de 2020, transformando drasticamente a rotina dos brasileiros principalmente pelo isolamento social, diversos setores tiveram suas

portas fechadas e outros tiveram que se reinventar fazendo uso de tecnologias digitais para manter seus negócios abertos, por exemplo, realizando compras ou efetuando vendas *on-line*. Nesse período, alguns setores tiveram crescimento expressivo, fato que repercutiu também no aumento da emissão dos certificados digitais, conforme mostra o gráfico da Figura 5.5.

Figura 5.5: Histórico anual de emissão de certificados.



Fonte: [23].

Observa-se no gráfico da Figura 5.5 que, mesmo antes da pandemia, o histórico da emissão dos certificados digitais no Brasil já tinha um crescimento surpreendente, atingindo uma média de 30% ao ano. No entanto, foi durante a pandemia que os números tiveram um salto maior ainda, como mostra a notícia publicada no site do ITI (2020) através do gráfico representado pela Figura 5.6.

Figura 5.6: Recorde de emissões dos certificados digitais.

Fonte: [24].

Analisando o gráfico da Figura 5.6, percebe-se a comparação no mês de setembro em relação aos anos anteriores, em que o aumento foi de 90 mil certificados digitais, evidenciando o crescimento de 19,4%. Esse aumento destaca a importância desta tecnologia diante do cenário atípico do ano de 2020.

6 Considerações Finais

Neste trabalho, vimos a importância da matemática voltada especialmente para o estudo da Aritmética dos Números Inteiros, tornando-se uma ferramenta indispensável diante dos avanços da criptografia no mundo atual, principalmente no processo de desenvolvimento dos códigos secretos. Nesse contexto, destacamos a relevância do Pequeno Teorema de Fermat que possibilitou não somente provar o funcionamento da Criptografia RSA, mas também auxiliar na prova do método da Assinatura Digital DSA.

O método de Criptografia RSA continua sendo um dos algoritmos mais seguros e utilizados atualmente, que tem por base a segurança orientada pelo nível de dificuldade da fatoração de números inteiros com mais de 250 algarismos. Diante disso, estudos voltados na área de criptografia quântica têm chamado atenção não só para as consequências voltadas na questão da segurança do método RSA, como também no problema do logaritmo discreto apresentado no protocolo Diffie-Hellman e ElGamal.

Vimos também que a criptografia além de prover o sigilo das informações a fim de torná-las incompreensíveis, ainda proporciona um processo lógico-matemático sobre a mensagem, permitindo assim o uso das assinaturas digitais na qual a confidencialidade é ratificada através do certificado digital gerado por meio de uma Autoridade Certificadora, assegurando a autenticidade dos documentos assinados virtualmente e a sustentabilidade nos serviços prestados na internet.

Considerando a evolução tecnológica vivenciada nos últimos anos, a exemplo da computação quântica, deixando em risco sistemas de criptografia como o RSA e outros algoritmos similares como Diffie-Hellman, ElGamal e DSA, busca-se novos métodos e atualizações na área da criptografia pós-quântica como, por exemplo, a criptografia baseada em reticulados a qual se destaca como sendo uma forte candidata a garantir proteção em um universo pós-quântico [25].

Referências

[1] COSTA, C. **Introdução à criptografia**. Rio de Janeiro: Centro de Estudos de Pessoal, 2010. v. 1.

[2] COUTINHO, S. C. **Número inteiros e criptografia RSA**. 2. ed. Rio de Janeiro: IMPA, 2014.

[3] SINGH, S. **The code book**: the secret history of codes and code-breaking. United Kingdom: Fourth Estate, 1999.

[4] COUTINHO, S. C. **Criptografia**. Rio de Janeiro: IMPA, 2016.

[5] MENEZES, A. M.; OORSCHOT, P. C. V.; VANSTONE, S. A. **Handbook of Applied Cryptography**. Boca Raton: CRC Press, 1997.

[6] MORENO, E. D.; PEREIRA, F. D.; CHIARAMONTE, R. B. **Criptografia em software e hardware**. São Paulo: Novatec, 2005.

[7] SANTOS, J. P. de O. **Introdução à teoria dos números**. 3. ed. Rio de Janeiro: IMPA, 2007.

[8] IEZZI, G.; DOMINGUES, H. H. **Álgebra moderna**. 4. ed. São Paulo: Atual, 2003.

[9] DOMINGUES, H. H. **Fundamentos de Aritmética**. São Paulo: Atual, 1991.

[10] STALLINGS, W. **Criptografia e segurança de redes**. 6. ed. São Paulo: Pearson Education do Brasil, 2015.

[11] BUCHMANN, J. **Introduction to cryptography**. 2th ed. New York: Springer-Verlag, 2004.

[12] BURNETT, S.; PAINE, S. **Criptografia e segurança**: o guia oficial RSA. Rio de Janeiro: Campus, 2002.

[13] KLEINJUNG, T. Factorization of a 768-bit rsa modulus. **Advances in cryptology- CRYPTO 2010**. Berlin Heildeberg: Springer Veerlag, v. 6223, p. 333-350, ago. 2010. Trabalho apresentado na 30th Annual Cryptology Conference, Santa Barbara, CA, USA, 2010.

[14] PATRINGENARU, I. New record set for cryptographic challenge. **PHYS. ORG**. San Diego: Universtity of California, 2020. Disponível em: https://phys.org/news/2020-03-cryptographic.html. Acesso em: 30 jun. 2020.

[15] FALBRIARD, C.; BROSSO, I. **Computação quântica**: a realidade de uma nova era. Rio de Janeiro: Alta Books, 2020.

[16] OLIVEIRA, I. S. Computação quântica. In: Física. **Ciência Hoje**. Rio de Janeiro, v. 33, p. 22-29, 2003.

[17] FIGUEIREDO, L. M. **Introdução à criptografia**. Rio de Janeiro: Centro de Estudos de Pessoal, 2010. v. 2.

[18] BRASIL. Medida provisória n. 2.200-2, de 24 de agosto de 2001. Institui a Infra-Estrutura de Chaves Públicas Brasileira - ICP-Brasil, transforma o Instituto Nacional de Tecnologia da Informação em autarquia, e dá outras providências. **[Diário Oficial da União]**. Brasília, DF, 2001. Disponível em: http://www.planalto.gov.br/ccivil_03/MPV/Antigas_2001/2200-2.htm. Acesso em: 30 ago. 2020.

[19] MONTEIRO, E. S.; MIGNONI, M. E. **Certificados digitais**: conceitos e práticas. Rio de Janeiro: Brasport, 2007.

[20] INSTITUTO NACIONAL DE TECNOLOGIA DA INFORMAÇÃO. **Estrutura da ICP-Brasil**. Disponível em: https: //estrutura.iti.gov.br. Acesso em: 20 set. 2020.

[21] ALECRIM, E. **Entendendo a certificação digital**. Disponível em: https://www.gov.br/iti/pt-br/assuntos/noticias/iti-na-midia/entendendo-a-certificacao-digital. Acesso em: 23 set. 2020.

[22] INSTITUTO NACIONAL DE TECNOLOGIA DA INFORMAÇÃO. **Certificado digital**: o que é, para que serve e como fazer um documento virtual. Disponível em: https://www.gov.br/iti/pt-br/assuntos/noticias/iti-na-midia/iti-na-midia-certificado-digital-o-que-e-para-que-serve-e-como-fazer-um-documento-virtual. Acesso em: 23 out. 2020.

[23] INSTITUTO NACIONAL DE TECNOLOGIA DA INFORMAÇÃO. **Histórico anual de emissão de certificados**. Disponível em: https://numeros.iti.gov.br/#/cert-visao-geral. Acesso em: 30 nov. 2020.

[24] INSTITUTO NACIONAL DE TECNOLOGIA DA INFORMAÇÃO. **ICP-Brasil apresenta mais um recorde de emissões**. Disponível em: https://www.gov.br/iti/pt-br/assuntos/noticias/indice-de-noticias/icp-brasil-apresenta-mais-um-recorde-de-emissoes. Acesso em: 30 nov. 2020.

[25] MENEGHETTI, F. C. C. **Reticulados**: um estudo de alguns parâmetros relevantes para aplicações em criptografia. Dissertação (Mestrado em Matemática) - Instituto de Matemática, Estatística e Computação Científica, Universidade Estadual de Campinas, 2020.