Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Matemática
Programa de Mestrado Profissional
em Matemática em Rede Nacional

Estudo sobre os Principais Aspectos da Criptografia Simétrica e Assimétrica ao longo da História

Thiago Marques Esteves Póvoa

Brasília

2019

Thiago Marques Esteves Póvoa

Estudo sobre os Principais Aspectos da Criptografia Simétrica e Assimétrica ao longo da História

Dissertação apresentada ao Departamento de Matemática da Universidade de Brasília, como parte dos requisitos do Programa de Mestrado Profissional em Matemática em Rede Nacional - PROFMAT, para obtenção do grau de Mestre.

Universidade de Brasília - UnB

Departamento de Matemática - MAT

PROFMAT - SBM

Orientador: Prof. Dr. Vinícius de Carvalho Ríspoli

Brasília

2019

Thiago Marques Esteves Póvoa
Estudo sobre os Principais Aspectos da Criptografia Simétrica e Assimétrica ao longo da História/ Thiago Marques Esteves Póvoa. – Brasília, 2019-
110 p.

Orientador: Prof. Dr. Vinícius de Carvalho Ríspoli

Dissertação de Mestrado – Universidade de Brasília - UnB
Departamento de Matemática - MAT
PROFMAT - SBM, 2019.

1. Palavra Chave 1: Criptografia 2. Palavra Chave 2: Criptografia Simétrica 3. Palavra Chave 3: Criptografia Assimétrica I. Nome do Orientador: Prof. Dr. Vinícius de Carvalho Ríspoli II. Título.

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Matemática

Estudo sobre os Principais Aspectos da Criptografia Simétrica e Assimétrica ao longo da História.

Por

THIAGO MARQUES ESTEVES PÓVOA

Dissertação apresentada ao Departamento de Matemática da Universidade de Brasília, como parte dos requisitos "Programa" de Mestrado Profissional em Matemática em Rede Nacional – PROFMAT, para obtenção do grau de

MESTRE EM MATEMÁTICA

Brasília, 08 de agosto de 2019.

Comissão Examinadora:

Prof. Vinicius de Carvalho Rispoli (Orientador)

Prof. Tiago Alves da Fonseca – FGA/UnB

Prof. Edson Alves da Costa Júnior – FGA/UnB

Dedico esta dissertação, primeiramente, à minha família, pelo incondicional apoio com o qual sempre pude contar, mesmo nas horas mais difíceis. Dedico também aos colegas de mestrado, por terem me ajudado em diversos momentos da nossa caminhada e por terem me proporcionado um imenso aprendizado, que jamais seria alcançado no estudo solitário. Dedico também aos professores, que na incansável missão de ensinar, nos proporcionam, em cada nova aula, uma nova visão sobre os conceitos matemáticos muitas vezes abstratos e de difícil compreensão. Por fim, dedico esta dissertação aos meus alunos, com os quais tenho o imenso prazer de resolver problemas diversos e aprender cada vez mais a cada dia em que tenho o prazer de lecionar.

Agradecimentos

Agradeço a todos que, de alguma forma, contribuem para tornar o mundo um lugar melhor. Sobretudo, àqueles que o fazem por meio da educação, essa ferramenta que possui o poder de transformar a sociedade. Espero poder retribuir um pouco de tudo que recebi de todos vocês.

"A mathematician, like a painter or a poet, is a maker of patterns. If his patterns are more permanent than theirs, it is because they are made with ideas "

G.H. Hardy

Resumo

A criptografia está no nosso dia a dia. Nós utilizamos criptografia quando realizamos uma troca de *e-mails* por um provedor, quando realizamos uma troca de mensagens pelo aplicativo *WhatsApp*, quando utilizamos um certificado digital para assinar documentos digitais ou atestar nossa identidade nos meios digitais, quando compramos um produto em um *site* usando nosso cartão de crédito, quando configuramos a segurança da rede *Wi-Fi* das nossas casas... Enfim, nossa vida está repleta de criptografia, e não há como lidar com todas essas questões sem um conhecimento mínimo sobre o funcionamento das cifras e dos algoritmos usados para garantir a nossa privacidade e segurança. Esta dissertação busca prover ao leitor uma noção geral sobre as ideias e sobre os processos utilizados na imensa atividade criptográfica que está diariamente à nossa volta. Inicialmente, é apresentado um resumo do panorama geral da criptografia ao longo da história, seguido por descrições de processos de criptografia simétrica e assimétrica, e de uma proposta de aplicação de criptografia em sala de aula. O ferramental matemático necessário para o bom entendimento dos tópicos deste trabalho consiste em conhecimentos sobre Álgebra e Teoria dos Números, basicamente. À medida que a complexidade das técnicas vai aumentando, a complexidade da matemática necessária também aumenta. Portanto, algumas cifras específicas são tratadas mais detalhadamente, enquanto outras cifras, mais simples, são descritas com um grau de detalhes consideravelmente menor. Os algoritmos criptográficos mais complexos abordados neste trabalho, baseados em propriedades algébricas de curvas elípticas, necessitam de alguns conhecimentos básicos sobre espaços projetivos e variedades projetivas, que são fornecidos no próprio texto. Por fim, após a descrição das cifras e algoritmos, são sugeridas ao professor propostas de utilização de criptografia no contexto da sala de aula do ensino fundamental II, com o intuito de tentar aproximar os estudantes dos conceitos básicos da criptografia.

Palavras-chaves: criptografia. criptografia simétrica. criptografia assimétrica.

Abstract

Cryptography is in our day to day. We use cryptography when performing an e-mail exchange by a provider, when we perform an exchange of messages by the WhatsApp application, when we use a digital certificate to sign digital documents or to attest our identity in digital media, when we buy a product on a site using our credit card, when we configure the security of the Wi-Fi network in our homes... Anyway, our life is full of cryptography, and there is no way to deal with all these issues without a minimum knowledge about the operation of cyphers and algorithms used to ensure our privacy and security. This dissertation seeks to provide the reader with a general notion about the ideas and processes used in the immense cryptographic activity that is daily around us. Initially, is presented a overview of cryptography throughout it's history, followed by descriptions of symmetric and asymmetric cryptography processes, and by an application proposal of cryptography in classroom. The mathematical knowledge necessary for a good understanding of the topics of this work consists of knowledge about Algebra and Number Theory, basically. As the complexity of the techniques increases, the complexity of the required mathematics also increases. Therefore, some specific cyphers are treated in more detail, while others, more simple, are described in a considerably lower degree of detail. The most complex cryptographic algorithms in this work, that are based on algebraic properties of elliptic curves, require some basic knowledge about projective spaces and projective varieties, which are provided in the text itself. Finally, after describing the cyphers and algorithms, proposes to use cryptography in the context of the elementary school classroom are suggested to the teacher, in order to try to bring students closer to the basic concepts of cryptography.

Key-words: cryptography. symmetric cryptography. asymmetric cryptography.

Lista de ilustrações

Lista de tabelas

Sumário

Introdução

A criptografia sempre esteve presente na história da humanidade, desde a nossa história antiga, passando pela era medieval, pelo renascimento, pelos períodos marcados pelas grandes guerras do século XX, culminando nos modernos processos de criptografia da atualidade, que permeiam toda a comunicação realizada atualmente na internet. A engenhosidade no desenvolvimento dos mais variados mecanismos para tentar ocultar uma mensagem dos olhos de um possível espião tornou a ciência criptográfica extremamente sofisticada. A história do desenvolvimento dessas cifras, cada vez mais seguras e complexas, é uma grande aventura pela própria história da humanidade. Ao longo dos séculos, sempre houve um embate velado entre aqueles que buscavam desenvolver as cifras seguras, denominados criptógrafos, e aqueles que buscavam métodos não autorizados para decifrar as mensagens, denominados criptoanalistas. A constante disputa entre a criptografia e a criptoanálise foi o grande motor do desenvolvimento das modernas tecnologias que possuímos hoje. A ciência que estuda tanto as técnicas de criptografia quanto as técnicas de criptoanálise é denominada criptologia (DOOLEY, 2018). Antes da invenção dos computadores, essa disputa estava restrita aos círculos diplomáticos e militares, sempre no âmbito governamental. Porém, com a utilização em larga escala dos computadores pessoais e, principalmente, da internet, a necessidade de se viabilizar um mecanismo de troca segura de mensagens entre todos que possuem uma conta de *e-mail*, por exemplo, trouxe novos e complexos desafios para os criptógrafos. Junto com esses desafios, vieram ideias e soluções inovadoras, que solucionaram problemas milenares como, por exemplo, o problema da distribuição segura de chaves, que será abordado neste trabalho. Portanto, a história da criptografia deve ser encarada como uma verdadeira história de desafio, competição e aprendizado. Uma batalha intelectual entre aqueles que buscavam ocultar os segredos, e aqueles que queriam conhecê-los. A cada nova geração de cifras, o desafio estava lançado aos criptoanalistas. E a cada vez que os criptoanalistas conseguiam desvendar os segredos daquela geração de cifras, tornando-as inseguras, o desafio estava lançado aos criptógrafos. Dessa forma, construiu-se a história da criptografia.

Objetivos

Os objetivos deste trabalho são, basicamente, prover ao leitor uma noção geral sobre as ideias e sobre os processos utilizados na criptografia que está diariamente à nossa volta, e propor uma utilização de criptografia em sala de aula, por meio de atividades direcionadas a alunos do ensino fundamental II. A divisão das atividades

por ano letivo escolar foi realizada com base na divisão de conteúdos estabelecida pela Base Nacional Comum Curricular (BRASIL, 2017). Como ainda são escassos os títulos sobre criptografia escritos em língua portuguesa, este trabalho busca contribuir para uma maior disseminação dos conhecimentos criptográficos ao leitor que não possui acesso aos títulos escritos, em sua grande maioria, em língua inglesa. Buscou-se, sempre que possível, a contextualização histórica das cifras e dos algoritmos abordados, na tentativa de tornar o fluxo de ideias que foram surgindo na história da criptografia o mais natural possível. Espera-se que o leitor possa se beneficiar da abordagem sempre pautada na busca pela clareza e simplicidade na descrição das ideias envolvidas nas diversas aplicações criptográficas da atualidade.

Estrutura do Trabalho

No Capítulo 1, serão apresentadas algumas definições e terminologia básica utilizadas no universo criptográfico, seguidas de um panorama histórico acerca do desenvolvimento da criptografia ao longo dos séculos, até o surgimento da criptografia de chave pública e da assinatura digital, no século XX. No Capítulo 2, será feita uma apresentação de duas cifras simétricas muito utilizadas na atualidade, a cifra DES e a cifra AES. No Capítulo 3, será apresentado o algoritmo de criptografia assimétrica RSA, e também serão abordados tópicos do ramo da criptografia assimétrica conhecido como *Elliptic Curve Cryptography - ECC*, por meio da apresentação de algoritmos de criptografia e assinatura digital baseados em propriedades de curvas elípticas. No Capítulo 4, serão apresentadas propostas de utilização de criptografia no contexto da sala de aula do ensino fundamental II, com o intuito de tentar aproximar os estudantes dos conceitos básicos de criptografia. Por fim, no Capítulo 5, serão apresentadas algumas conclusões e impressões gerais sobre a importância da criptografia na sociedade.

1 Terminologia e Aspectos Históricos

1.1 Terminologia

A palavra criptografia é derivada das palavras gregas *kriptos*, que significa "oculto" e *graphein*, que significa "escrever". A criptografia é um dos ramos da criptologia, e seu objetivo não é esconder a mensagem, mas sim ocultar seu significado. Para isso, devem ser utilizadas técnicas adequadas. As técnicas de criptografia que realizam a substituição de elementos da mensagem por outros elementos previamente combinados são denominadas técnicas de substituição, e podem ser divididas em 2 grandes grupos: os códigos e as cifras. Os códigos consistem em se transformar elementos da mensagem, de comprimento variável, mas que carregam algum significado, em elementos codificados, seguindo-se um padrão previamente combinado entre as partes. Por exemplo, pode-se adotar um código em que a palavra "sorte" seja sistematicamente substituída pelo símbolo *. Não há qualquer relação lógica entre a palavra original e o símbolo codificado, apenas há a necessidade de que essa relação tenha sido combinada previamente entre o emissor e o receptor da mensagem. As mensagens codificadas são difíceis de serem descobertas por um espião, mas possuem uma grande desvantagem: as relações entre o texto da mensagem e seus respectivos códigos precisam ser guardadas em uma espécie de "dicionário", e tanto o emissor quanto o receptor da mensagem precisam possuir dicionários idênticos. Esses dicionários costumam ser longos, pois precisam conter todas as expressões dotadas de significado que serão usadas na troca de mensagens, bem como seus respectivos códigos. Além disso, se um dos dicionários for roubado, todas as mensagens que já foram codificadas utilizando-se aquele padrão de código poderão ser decodificadas por um espião. Não há qualquer flexibilidade para a mudança do padrão de codificação, todas as mensagens serão codificadas e decodificadas com o mesmo padrão de dicionário. Daí a fragilidade da criptografia que se utiliza de códigos para esconder o significado da mensagem. Diferentemente dos códigos, o modo de operação das cifras consiste em se transformar elementos da mensagem, de comprimento fixo (letras, por exemplo), e que não carregam qualquer significado, em elementos cifrados, a partir de uma regra lógica, que pode ser aplicada em qualquer um dos elementos da mensagem original. Por exemplo, para cifrar a palavra "sorte", pode-se adotar a cifra que substitui cada letra pela letra seguinte do alfabeto, resultando na mensagem cifrada "TPSUF". A vantagem das cifras em relação aos códigos é que uma só regra precisa ser compartilhada entre emissor e receptor da mensagem, pois essa regra pode ser aplicada em qualquer elemento da mensagem. Além disso, essa regra carrega uma certa flexibilidade, conhecida como "chave" da

cifra, de forma que se possa mudar a regra sem grandes problemas, sempre que se desejar. Por exemplo, em um alfabeto com 26 letras, é necessário memorizar apenas 26 regras, de qual letra utilizar para se substituir cada uma das letras da mensagem original. Evidentemente, essa quantidade de informação a ser memorizada não precisa ser colocada em um "dicionário", o que impede que tal regra seja roubada por um espião. Também, as 26 regras podem ser construídas de forma que haja uma relação entre elas, como por exemplo, "substituir cada letra da mensagem original pela letra seguinte no alfabeto", de forma que memorizar essa instrução permite a construção de todas as 26 regras de cifragem das letras do alfabeto. Caso se deseje mudar a regra, basta mudar a instrução, como por exemplo, "substituir cada letra da mensagem original pela letra anterior no alfabeto". Essa flexibilidade da regra de cifração é uma enorme vantagem da cifra em relação ao código. Além disso, o processo de cifração é muito mais simples e rápido, pois não é necessário procurar no dicionário de códigos o correspondente para cada elemento dotado de significado do texto, basta aplicar a regra universal para cada um dos elementos semelhantes a serem cifrados (letras, por exemplo). Isso torna muito mais rápido o processo de cifração e decifração das cifras do que o processo de codificação e decodificação dos códigos.

Nesse ponto, é necessário fazer uma ressalva quanto à terminologia utilizada, e adotar algumas definições, que serão utilizadas ao longo de todo o trabalho. Primeiramente, o que se observa, tanto na literatura em língua portuguesa, quanto na literatura em língua inglesa, é que não há uma padronização na utilização dos termos cifrar, decifrar, codificar e decodificar. Em alguns autores, por exemplo (DOOLEY, 2018), (SINGH, 2011) e (KAHN, 1996), os termos cifrar e decifrar, bem como codificar e decodificar, são utilizados para designar as ações do emissor e do receptor da mensagem, não havendo uma terminologia específica para um possível espião, que está tentando "quebrar" a cifra ou o código. Em outros autores, por exemplo (COUTINHO, 2000), a utilização dos termos codificar e decodificar se aplica às ações realizadas pelo emissor e pelo receptor da mensagem, não importando se eles utilizaram uma cifra ou um código. Já o termo decifrar, neste mesmo autor, é utilizado para designar a ação de um espião, que está tentando "quebrar" a cifra ou o código. Dessa forma, neste trabalho optou-se pela nomenclatura adotada em (DOOLEY, 2018), (SINGH, 2011) e (KAHN, 1996), por ser mais específica e inequívoca, de forma que serão utilizados os termos cifrar e decifrar para a utilização geral de cifras, e os termos codificar e decodificar para a utilização geral de códigos.

É importante ressaltar também que, além de os códigos poderem ser utilizados para a transmissão de mensagens secretas, sua utilidade não se restringe a esse propósito. Os códigos podem ser utilizados como instrumento para permitir o processo de transmissão de informações em determinados canais. É assim que a informação trafega atualmente nos meios digitais, por meio de códigos, que transformam a linguagem hu-

mana em *bits*, que correspondem à linguagem dos computadores digitais. A teoria dos códigos e da informação, que estuda técnicas eficientes e seguras para a transmissão de informações por meio de códigos, é bastante recente na história da humanidade, tendo se iniciado com os trabalhos publicados por Claude Shannon, um matemático norte-americano. Shannon publicou os artigos *A mathematical theory of communication* (SHANNON, 1948) e *Communication Theory of Secrecy Systems* (SHANNON, 1949) em 1948 e 1949, respectivamente. Logo em seguida, vieram os trabalhos de Richard W. Hamming, publicados em 1950 (SHOKRANIAN, 2012). Ambos trabalhavam no laboratório da AT&T Bell, que atualmente é conhecido como laboratório Nokia Bell. A partir dos trabalhos desses pesquisadores, sobretudo de Shannon, nasceram duas novas teorias matemáticas, a Teoria da Informação e a Teoria dos Códigos Corretores de Erros. Este trabalho não abordará os tópicos estudados nessas áreas de conhecimento. Para uma abordagem mais detalhada sobre a Teoria dos Códigos Corretores de Erros, pode-se consultar (HEFEZ A.; VILLELA, 2008), e para uma abordagem histórica sobre surgimento da Teoria da Informação, pode-se consultar (GLEICK, 2012).

Conforme observado anteriormente, no universo da criptografia, faz-se necessário definir uma terminologia padrão a ser adotada, para que não haja ambiguidade ou confusão nos termos utilizados. Dessa forma, este trabalho utilizará uma terminologia criptográfica alinhada com a termilogogia utilizada nas obras sobre criptografia, em geral. Por exemplo, quando nos referirmos ao emissor de uma mensagem, é comum utilizarmos o nome Alice. Analogamente, quando nos referirmos ao receptor da mensagem, é comum utilizarmos o nome Bob. Por fim, quando nos referirmos a um espião que consegue interceptar a mensagem cifrada e deseja conhecer seu conteúdo, é comum utilizarmos o nome Eve. Também, quando nos referirmos à mensagem original, antes de passar pelo processo de cifração, é comum chamá-la de texto claro ou *plaintext*, e a mensagem nesse formato será sempre escrita em letras minúsculas. Por sua vez, quando nos referirmos à mensagem após ter passado pelo processo de cifração, é comum chamá-la de texto cifrado, ou *ciphertext*, e a mensagem nesse formato será sempre escrita em letras maiúsculas. Neste ponto, pela própria nomenclatura utilizada, fica claro que não serão abordados processos de codificação e decodificação neste trabalho, apenas processos de cifração e decifração. Por fim, quando nos referirmos ao segredo necessário para cifrar e para decifrar mensagens utilizando-se uma cifra conhecida pelo emissor e pelo receptor, é comum chamar esse segredo de chave ou *key*.

1.2 Histórico

Nesta seção, será apresentado um panorama geral do desenvolvimento da criptografia ao longo da história da humanidade. Sempre que possível, será dada prioridade à elucidação dos fatos históricos. Os aspectos matemáticos das cifras serão abordados

nos capítulos seguintes.

1.2.1 Surgimento das Mensagens Secretas

Os primeiros registros de utilização de mensagens secretas ao longo da história foram feitos pelo historiador grego Heródoto, em sua narrativa sobre a invasão da Grécia pelos persas, no século V a.C (SINGH, 2011). Na história narrada por Heródoto, consta que os gregos utilizaram uma técnica para esconder mensagens do conhecimento dos persas, que poderiam interceptá-las. Tratava-se de usar tábuas de madeira recobertas com cera (algo que, na época, era utilizado para a escrita) como instrumento para se esconder as mensagens. O que os gregos fizeram foi escrever as mensagens diretamente na madeira, e recobri-la com a cera, de forma que as mensagens não ficassem expostas. Dessa forma, conseguiam passar com as tábuas pelo controle dos persas, sem que eles identificassem qualquer mensagem nas tábuas. Chegando ao seu destino, os gregos derretiam ou raspavam a cera, de forma que a mensagem escondida se tornava então visível. Essa técnica, embora bastante engenhosa, não pertence ao ramo da criptografia, trata-se de esteganografia. A esteganografia, nome derivado das palavras gregas *steganos*, que significa "coberto" e *graphein*, que significa "escrever", consiste em um conjunto de técnicas utilizadas para se transmitir uma mensagem secreta de maneira a ocultá-la em algum dos elementos relacionados com o transporte da mensagem. Por exemplo, pode-se utilizar uma tinta invisível para escrever a mensagem sobre um tecido, que só se tornará visível quando o tecido for aquecido. Outro exemplo mais moderno de esteganografia consiste na utilização de uma imagem digital (um arquivo JPEG, por exemplo) para se esconder uma mensagem secreta. Isso pode ser feito utilizando-se a sequência binária de uns e zeros, que representam as cores de cada pixel da imagem, para se esconder uma mensagem secreta, sem que haja uma alteração significativa nas cores da imagem. A esteganografia se diferencia da criptografia porque, apesar de ambas as técnicas buscarem a mesma finalidade, que é impedir que um possível espião tenha acesso ao conteúdo da mensagem, elas o fazem de maneira distinta. Enquanto na esteganografia o objetivo é impedir que a mensagem seja descoberta, na criptografia o objetivo é impedir que a mensagem seja compreendida.

Pode-se, inclusive, utilizar-se uma combinação das técnicas de esteganografia, para esconder a mensagem dos olhos de espiões, com as técnicas de criptografia para, caso a mensagem seja descoberta, que não possa ser compreendida, aumentando-se assim ainda mais a segurança do processo de troca de mensagens. Essa combinação de criptografia com esteganografia foi utilizada pelos alemães durante a Segunda Guerra Mundial (KAHN, 1996). Eles, primeiramente, cifravam a mensagem que desejavam transmitir. Em seguida, reduziam fotograficamente o texto cifrado, até que seu tamanho fosse compatível com o tamanho de um ponto final de uma frase impressa em

tamanho natural. Dessa forma, eles inseriam o ponto final com a mensagem cifrada em correspondências aparentemente normais, de forma que o receptor da correspondência pudesse ampliar o ponto final com a mensagem cifrada e, conhecendo a cifra utilizada e sua chave, pudesse ler seu conteúdo. Entretanto, nem todas as mensagens secretas enviadas pelos alemães por meio de micropontos foram cifradas previamente e, quando essa técnica foi descoberta pelos agentes norte-americanos, várias mensagens alemãs puderam ser lidas. Esse exemplo mostra que, muito embora a esteganografia possa oferecer algum nível de segurança para a transmissão de mensagens secretas, sem o auxílio da criptografia pode-se colocar tudo a perder.

1.2.2 Tipos de Cifras

Dentro da criptografia, as cifras podem ser divididas em 2 grandes ramos, dependendo de como elas funcionam para esconder o conteúdo das mensagens: as cifras de transposição e as cifras de substituição.

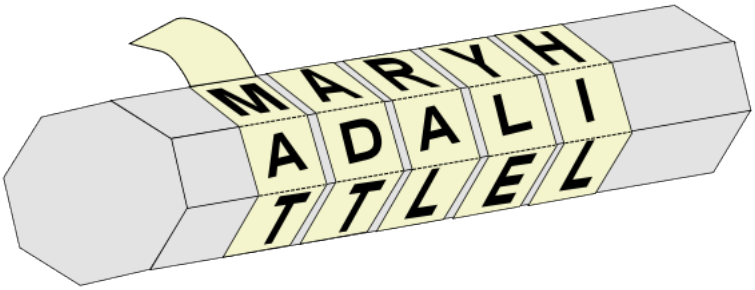
As cifras de transposição funcionam embaralhando as letras do texto claro, para gerar o texto cifrado. Em outras palavras, elas produzem anagramas do texto claro, por meio de uma regra de embaralhamento das letras, conhecida como chave da cifra. Quanto maior o número de letras na mensagem, maior o número de anagramas possíveis, e esse crescimento é bastante rápido. Portanto, para mensagens com uma grande quantidade de letras cifradas com uma cifra de transposição, sem se conhecer a regra utilizada para cifrá-la (chave), espera-se encontrar alguma dificuldade na tentativa de decifrar a mensagem, pois serão muitas possibilidades de anagramas para se testar qual deles corresponde a uma mensagem coerente. Esse tipo de abordagem, que consiste na tentativa de se obter o texto claro testando-se todas as possibilidades de chaves de uma cifra, é conhecido como método da força bruta. Para mensagens longas, o ataque pelo método da força bruta se mostra ineficiente nas cifras de transposição, desde que somente o emissor e o receptor da mensagem conheçam a regra utilizada (chave) para embaralhar as letras do texto claro. Há vários tipos de cifras de transposição, alguns mais simples, outros mais complexos. Um tipo bastante simples de cifra de transposição consiste em escrever a mensagem em 2 linhas, colocando-se cada letra em cada linha, de maneira alternada. Por fim, para obter o texto cifrado, basta escrever a primeira linha seguida da segunda linha. Por exemplo, vamos cifrar a frase "numeros sao abstratos" utilizando-se essa cifra, conforme se observa na figura 1.

Figura 1 – Cifra de Transposição com 2 Linhas

n m r s a a s r t s
u e o s o b t a o

A mensagem cifrada será então "NMRSAASRTSUEOSOBTAO". Para o receptor decifrá-la, basta aplicar o método contrário, isto é, escrever a mensagem em 2 linhas e ler, alternadamente, uma letra de cada linha. Esse tipo de cifra de transposição pode ser modificado aumentando-se o número de linhas em que se escrevem as letras de forma alternada, produzindo-se assim uma nova cifra. Um outro exemplo de cifra de transposição, utilizado desde o século V a.C., utiliza uma longa tira de tecido e um bastão cilíndrico, denominado *citale*. A cifra consiste em se enrolar a tira de tecido no bastão, de maneira helicoidal, e escrever o texto claro ao longo do comprimento do bastão, iniciando uma nova linha sempre que a mensagem chegar até o fim do comprimento do bastão. Após finalizada a escrita da mensagem, basta desenrolar a tira de tecido do bastão, e o que se observará ao longo da tira de tecido será o texto cifrado. Para o receptor decifrar a mensagem, basta que ele enrole a tira em outro bastão, de mesmo diâmetro do bastão utilizado para cifrar a mensagem, e leia a mensagem nas linhas ao longo do comprimento do bastão. Essa técnica foi utilizada pelos espartanos, no século V a.C., para transmitir mensagens secretas, de maneira que os persas não pudessem conhecer seu conteúdo (SINGH, 2011). Um exemplo de *citale* pode ser observado na figura 2.

Figura 2 – Exemplo de *citale* (Disponível em <commons.wikimedia.org>)



O outro tipo de cifra, diferente da cifra de transposição, é a cifra de substituição. Enquanto a cifra de transposição se preocupa em embaralhar as letras da mensagem, a cifra de substituição substitui cada letra do texto claro por uma outra letra, ou mesmo um símbolo, previamente combinados, obtendo-se assim o texto cifrado. Na prática, o que se faz para operacionalizar o processo de cifragem por substituição é escrever o alfabeto original e, embaixo dele, as letras ou símbolos pelas quais as letras do alfabeto original devem ser trocadas. Dessa forma, surge um segundo alfabeto, denominado alfabeto cifrado. O alinhamento vertical das letras do alfabeto original com as letras ou símbolos do alfabeto cifrado facilita o processo de cifragem e decifragem pela cifra de substituição. Um exemplo bastante conhecido de cifra de substituição é a cifra que foi utilizada por Júlio César, imperador romano, para se comunicar com seus generais em combate pela Europa, por volta do século I a.C. (COUTINHO, 2000). A cifra de César

consistia em se deslocar as letras do alfabeto em 3 posições, de forma que cada letra do alfabeto original no texto claro fosse substituída pela letra correspondente no alfabeto deslocado, dando origem assim ao texto cifrado. Esse tipo de cifra de substituição, em que ocorre apenas um deslocamento do alfabeto original, preservando-se a posição relativa entre as letras, é denominado cifra de deslocamento. Segue na figura 3 uma ilustração do alfabeto original e do alfabeto deslocado de 3 posições, que é utilizado na cifra de César.

Figura 3 – Alfabeto Original e Alfabeto Deslocado de 3 posições - Cifra de César

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Como exemplo, vamos utilizar a cifra de César para cifrar a mesma mensagem cifrada anteriormente com a cifra de transposição: “números são abstratos”. Essa mensagem, cifrada com a cifra de César, se torna: “QXPHURVVDRDEVWRDWRV”. De maneira geral, as cifras de substituição não precisam manter a posição relativa entre as letras no alfabeto original e do alfabeto cifrado. Isto é, as letras do alfabeto cifrado podem estar ordenadas de qualquer maneira. Esse fato faz com que o ataque às cifras de substituição utilizando-se o método da força bruta seja bastante difícil, pois se tem $(26! - 1)$ possibilidades distintas de alfabetos cifrados (chaves), e é impraticável testar cada um deles para obter a mensagem original. Além disso, é possível que, durante a tentativa de busca do alfabeto cifrado correto, algum dos possíveis alfabetos cifrados errados também produza uma mensagem coerente quando aplicado ao texto cifrado, uma espécie de “falso positivo”, causando assim dúvida no criptoanalista. Porém, caso a cifra de substituição seja uma simples cifra de deslocamento, o método da força bruta é suficiente para se descobrir o conteúdo da mensagem, pois há somente 25 possibilidades de deslocamento do alfabeto (25 chaves), dado que há 26 letras e uma das posições corresponde à posição do alfabeto original. Portanto, fica evidente que as cifras de deslocamento são bastante inseguras, não resistindo nem mesmo a um simples ataque por força bruta. Também, é possível notar que as cifras de transposição e de substituição possuem caráter complementar: enquanto a primeira se preocupa em trocar a posição das letras, sem alterar sua natureza, a segunda faz justamente o oposto, altera a natureza de cada uma das letras, mas preserva a sua posição.

Tanto as cifras de transposição quanto as de substituição utilizam um segredo comum, que deve ser partilhado entre o emissor e o receptor da mensagem. No caso da cifra de transposição, o segredo é a maneira como se deve embaralhar as letras da mensagem. No caso da cifra de substituição, o segredo consiste no alfabeto cifrado. Esse segredo, em ambos os casos, é denominado chave da cifra. Portanto, a partir de

uma mensagem cifrada, basta que se conheça o tipo de cifra e a chave utilizada, para que se possa decifrar a mensagem. Porém, caso um espião intercepte a mensagem cifrada, mesmo que ele conheça a cifra utilizada, ele não poderá decifrar a mensagem sem o conhecimento da chave. Portanto, a segurança da troca de mensagens secretas utilizando essas cifras depende, essencialmente, do caráter secreto da chave. Para que uma cifra seja forte contra o ataque por força bruta, ela deve possuir um número grande de possibilidades de chaves, que inviabilize a tentativa de todas elas. É por esse motivo que a cifra de deslocamento é insegura, pois só possui 25 chaves, sendo possível testar todas elas rapidamente.

1.2.3 Criptoanálise das Cifras de Substituição Monoalfabéticas

Primeiramente, faz-se necessário definir o conceito de cifra de substituição monoalfabética. Trata-se, simplesmente, de uma cifra de substituição em que apenas um alfabeto cifrado é utilizado no processo de cifração de toda a mensagem. Isto é, o alfabeto cifrado é fixo, e é utilizado para cifrar toda a mensagem. Todas as cifras de substituição abordadas anteriormente são cifras de substituição monoalfabéticas. Essas cifras padecem de uma fraqueza sutil, que permite a sua decifração sem que se conheça a chave secreta, e sem que seja necessário utilizar-se o método da força bruta: trata-se da análise da frequência média de ocorrência das letras em cada idioma. A criptoanálise das cifras de substituição monoalfabéticas pode ser realizada com base no conhecimento dessas frequências, caso o texto cifrado seja suficientemente longo para que se espere que as frequências de ocorrência das letras no idioma seja mantida no texto cifrado. Como cada letra do alfabeto original possui apenas uma letra correspondente no alfabeto cifrado, as frequências de ocorrências das letras em ambos os alfabetos serão as mesmas. Por exemplo, caso a letra "a" do alfabeto original corresponda à letra "P" do alfabeto cifrado, espera-se que a letra "P" no texto cifrado apresente a mesma frequência que a letra "a" no idioma em que a mensagem foi escrita. Isso permite que se possa comparar as frequências das letras do texto cifrado com as frequências das letras no idioma em que a mensagem foi escrita, estabelecendo-se assim correspondências. A primeira utilização documentada desse tipo de técnica de criptoanálise para quebra de cifras de substituição monoalfabéticas se deu no século IX, com os trabalhos de al-Kindi (SINGH, 2011), conhecido como "o filósofo dos árabes". Portanto, a partir da técnica de análise de frequências das letras em um determinado idioma, é possível "quebrar" as cifras de substituição monoalfabéticas, bastando para isso conhecer o idioma no qual o texto claro foi escrito, e possuir um texto cifrado suficientemente longo para preservar as frequências das letras naquele idioma.

1.2.4 Surgimento das Cifras de Substituição Polialfabéticas

Após o mundo árabe descobrir uma maneira eficaz para decifrar mensagens cifradas com cifras de substituição monoalfabéticas sem o conhecimento da chave secreta, o mundo enfrentou um período em que os criptoanalistas estavam ganhando a batalha, pois não havia cifra absolutamente segura. No início do século XV, grande parte dos criptoanalistas da Europa já conheciam a técnica de análise de frequência, de forma que se tornou necessário o desenvolvimento de cifras mais robustas, para que se garantisse a segurança no processo de comunicação por mensagens secretas. Nesse período, a cidade italiana de Florença estava repleta de pensadores, filósofos e cientistas renascentistas. Um deles se chamava Leon Battista Alberti, e ele foi o primeiro homem a propor um tipo de cifra mais robusta que as cifras de substituição monoalfabéticas conhecidas até então (SINGH, 2011). Alberti percebeu que a grande fragilidade das cifras de substituição monoalfabéticas, o que faz com que elas possam ser exploradas com a análise de frequências, consiste no fato de cada letra do texto claro possuir apenas um correspondente no texto cifrado, preservando assim a frequência de ocorrência das letras. A ideia que Alberti teve para contornar essa fragilidade foi utilizar 2 alfabetos cifrados diferentes para cifrar as letras da mensagem. Ele propôs a utilização alternada de cada um dos alfabetos cifrados na cifração em sequência das letras da mensagem, de forma que não houvesse mais a preservação das frequências de ocorrência das letras, pois a mesma letra do alfabeto comum poderia agora ser representada por 2 letras distintas dos alfabetos cifrados. Essa ideia de utilização de diferentes alfabetos cifrados na cifração de uma mensagem revolucionou o pensamento dos criptógrafos da Europa naquele período, de forma que surgiram algumas propostas de cifras que se utilizavam desse princípio para tornarem a cifra de substituição mais robusta. Essas cifras de substituição, que utilizam mais de um alfabeto cifrado na cifração da mensagem, são conhecidas como cifras de substituição polialfabéticas, e sua principal vantagem em relação às substituições monoalfabéticas é que as substituições polialfabéticas não preservam a frequência de ocorrência das letras, dificultando assim o seu ataque pelos criptoanalistas.

Coube então ao diplomata francês Blaise de Vigenère, já no século XVI, propor um modelo de cifra de substituição polialfabética que utilizava as ideias propostas por Alberti de uma forma bastante prática e eficiente. O que Vigenère propôs foi a utilização alternada de vários alfabetos cifrados, sendo essa quantidade de alfabetos cifrados um parâmetro que podia ser variado em sua cifra. Outra característica da cifra de Vigenère é que os vários alfabetos cifrados utilizados são fáceis de serem construídos, não necessitando assim que emissor e receptor memorizassem longas sequências de letras. Vejamos como funciona a cifra de Vigenère. Primeiramente, deve-se montar uma tábua, denominada Tábua de Vigenère, em que o alfabeto aparece escrito 27 vezes.

A primeira linha da tábua corresponde ao alfabeto escrito em sua forma convencional, e as outras 26 linhas correspondem aos alfabetos cifrados, escritos deslocando-se suas letras sempre em uma posição em relação ao alfabeto da linha anterior. Portanto, a segunda linha também corresponde ao alfabeto na forma convencional, a terceira linha corresponde ao alfabeto deslocado em uma posição, a quarta linha em 2 posições, e assim sucessivamente. A figura 4 representa uma Tábua de Vigenère.

Figura 4 – Tábua de Vigenère

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

Após a confecção da Tábua de Vigenère, deve-se combinar qual dos alfabetos cifrados será utilizado para cifrar cada letra da mensagem. Isso pode ser feito de maneira simples utilizando-se uma palavra-chave comum entre o emissor e o receptor: escreve-se a palavra-chave de forma a se alinhar verticalmente suas letras com as letras da mensagem e, para cada letra da mensagem a ser cifrada, utiliza-se o alfabeto da Tábua de Vigenère que se inicia com a letra correspondente da palavra-chave. Dessa forma, cada letra da palavra-chave determina qual alfabeto cifrado deve ser utilizado para cifrar cada letra do texto claro. Caso a mensagem possua mais letras que a palavra-chave, pode-se repetir a palavra-chave, seguidamente, quantas vezes forem necessárias, de forma que se tenha uma letra da palavra-chave correspondente para cada letra do texto claro, indicando assim qual alfabeto cifrado deve ser utilizado para cifrar aquela letra da mensagem. Portanto, essa cifra utiliza tantos alfabetos cifrados quantas letras distintas há na palavra-chave, de forma que não há a preservação da frequência de

ocorrência das letras no idioma do texto claro. Portanto, a cifra de Vigenère resiste ao ataque de frequência utilizado na criptoanálise das cifras de substituição monoalfabéticas. Como exemplo da utilização da cifra de Vigenère, vamos utilizá-la para cifrar a frase "retas paralelas nunca param", utilizando-se a palavra chave "GEOMETRIA". A letra "r" deve ser cifrada pelo alfabeto cifrado que se inicia pela letra "G", de forma que a letra correspondente ao "r" nesse alfabeto é a letra "X". Em seguida, a letra "e" deve ser cifrada pelo alfabeto cifrado que se inicia pela letra "E", de forma que a letra correspondente ao "e" nesse alfabeto é a letra "I". E assim sucessivamente, até que se obtém a mensagem completamente cifrada "XIHMWIRZARIZMWGLVCGTODEF". A figura 5 ilustra esse processo de cifração.

Figura 5 – Cifração Utilizando-se a Cifra de Vigenère

r	*e*	*t*	*a*	*s*	*p*	*a*	*r*	*a*	*l*	*e*	*l*	*a*	*s*	*n*	*u*	*n*	*c*	*a*	*p*	*a*	*r*	*a*	*m*
G	*E*	*O*	*M*	*E*	*T*	*R*	*I*	*A*	*G*	*E*	*O*	*M*	*E*	*T*	*R*	*I*	*A*	*G*	*E*	*O*	*M*	*E*	*T*
X	*I*	*H*	*M*	*W*	*I*	*R*	*Z*	*A*	*R*	*I*	*Z*	*M*	*W*	*G*	*L*	*V*	*C*	*G*	*T*	*O*	*D*	*E*	*F*

Pode-se observar que a letra "a" da palavra "retas" foi cifrada pela letra "M". Porém, na palavra "paralelas", cada uma das letras "a" foi cifrada por uma letra diferente, "R", "A" e "M", respectivamente. Portanto, fica claro que a mensagem cifrada não conserva a frequência de ocorrência de letras na língua portuguesa, idioma em que a mensagem foi escrita. Essa é a principal vantagem da cifra de Vigenère frente às cifras de substituição monoalfabéticas, conhecidas até então.

1.2.5 Criptoanálise da Cifra de Vigenère

O trabalho de Vigenère, que continha a descrição de sua cifra homônima, foi publicado em 1586. A cifra era considerada tão segura que foi apelidada pelos criptógrafos ao longo da história de *le chiffre indéchiffrable*. Portanto, pelos próximos 300 anos, os criptógrafos estariam ganhando a batalha contra os criptoanalistas. Porém, na metade do século XIX, começaram a surgir as ideias necessárias para se conseguir quebrar a cifra de Vigenère. Essas ideias surgiram nos trabalhos de Charles Babbage e Friedrich Kasiski. Babbage foi um cientista britânico, que ficou famoso por ter desenvolvido o projeto de uma máquina de calcular programável, que seria o precursor dos computadores modernos. Embora Babbage fosse um brilhante cientista, era comum que ele não concluísse seus trabalhos (SINGH, 2011). No caso da cifra de Vigenère, não foi diferente. Especula-se que ele conseguiu desenvolver a técnica para quebrar a cifra por volta de 1854, mas ele nunca a publicou. Sua técnica só veio à luz no século

XX, quando estudiosos examinaram postumamente as suas anotações. Por outro lado, a mesma técnica desenvolvida por Babbage foi descoberta, de maneira independente, pelo oficial da reserva do exército prussiano Friedrich Kasiski. Ele, ao contrário de Babbage, publicou sua técnica de criptoanálise para quebrar a cifra de Vigenère no ano de 1863, motivo pelo qual essa técnica ficou conhecida como Teste de Kasiski.

Essencialmente, as estratégias tanto de Babbage quanto de Kasiski eram a mesma. Elas consistiam em, primeiramente, tentar encontrar o tamanho da palavra-chave utilizada na cifra de Vigenère. Para isso, eles buscavam sequências de letras repetidas no texto cifrado. Essas sequências podem ter se formado ao acaso, mas, provavelmente, ocorreram devido ao fato de uma mesma sequência de letras do texto claro estar alinhada com as mesmas letras da palavra-chave, produzindo assim as mesmas letras no texto cifrado. Caso se procure por sequências de 3 letras repetidas, por exemplo, a probabilidade da repetição ter vindo do acaso já é bastante baixa. Portanto, inicialmente, pode-se procurar sequências repetidas de 3 letras no texto cifrado. Depois disso, verifica-se qual a quantidade de letras que há entre o início de uma das sequências de 3 letras e o início das outras ocorrências da sequência. Vamos denominar essa quantidade de letras como distância entre as repetições. Essa distância deverá ser um múltiplo do número de letras da palavra-chave. Portanto, caso haja mais de uma repetição da sequência de 3 letras no texto cifrado, o número de letras da palavra chave provavelmente será um divisor comum de todas as distâncias. Essa informação nos permite encontrar qual o número de letras da palavra-chave utilizada na cifração. Sabendo-se o número de letras da palavra-chave, sabe-se quantos alfabetos cifrados foram utilizados e sabe-se também a ordem em que eles foram utilizados, de forma que se pode separar quais letras do texto cifrado foram cifradas com determinado alfabeto cifrado. Isso permite que o problema de decifrar a cifra de Vigenère consista em decifrar vários problemas de substituição monoalfabética, cada problema relacionado com um dos alfabetos cifrados utilizados para cifrar a mensagem. Portanto, se a palavra-chave contiver n letras, será necessário decifrar n cifras de substituição monoalfabética, o que pode ser facilmente realizado por análise de frequência. Dessa forma, a cifra de Vigenère pode ser quebrada.

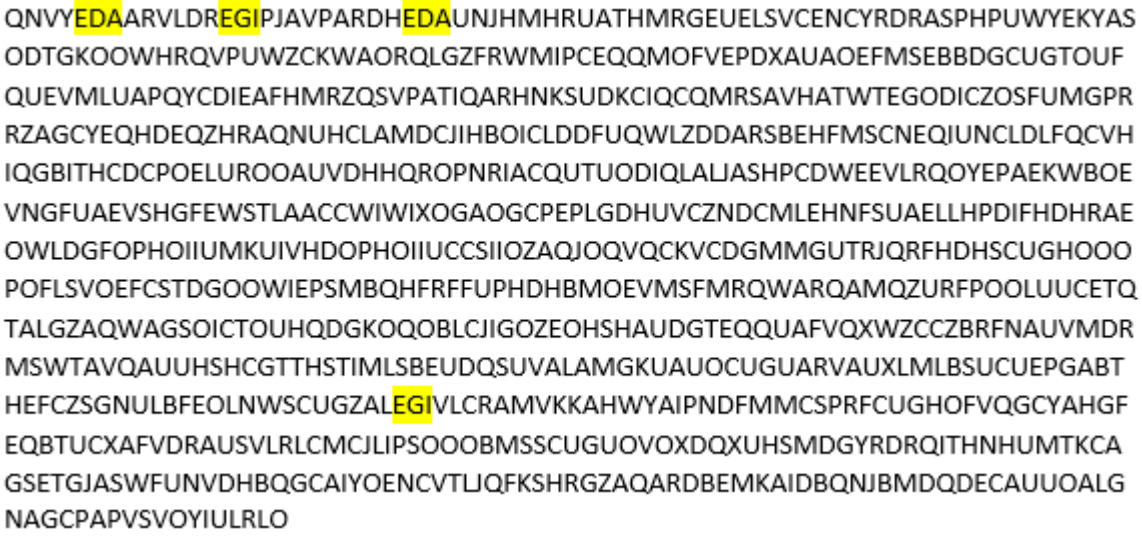
Como exemplo da aplicação do método exposto, considere que, em uma troca de mensagens cifradas utilizando-se a cifra de Vigenère, um espião intercepte a mensagem contida na figura 6.

Figura 6 – Texto Cifrado com Cifra de Vigenère

QNVYEDAARVLDREGIPJAVPARDHEDAUNJHMHRUATHMRGEUELSVCENCYRDRASPHPUWYEKYAS
ODTGKOOWHRQVPUWZCKWAORQLGZFRWMIPCEQQMOFVEPDXAUAOEFMSEBBDGCUGTOUF
QUEVMLUAPQYCDIEAFHMRZQSVPATIQARHNKSUDKCIQCQMRSAVHATWTEGODICZOSFUMGPR
RZAGCYEQHDEQZHRAQNUHCLAMDCJIHBOICLDDFUQWLZDDARSBEHFMSCNEQIUNCLDLFQCVH
IQGBITHCDCPOELUROOAUVDHHQROPNRIACQUTUODIQLALJASHPCDWEEVLRQOYEPAEKWBOE
VNGFUAEVSHGFEWSTLAACCWIWIXOGAOGCPEPLGDHUVCZNDCMLEHNFSUAELLHPDIFHDHRAE
OWLDGFOPHOIIUMKUIVHDOPHOIIUCCSIIOZAQJOQVQCKVCDGMMGUTRJQRFHDHSCUGHOOO
POFLSVOEFCSTDGOOWIEPSMBQHFRFFUPHDHBMOEVMSFMRQWARQAMQZURFPOOLUUCETQ
TALGZAQWAGSOICTOUHQDGKOQOBLCJIGOZEOHSHAUDGTEQQUAFVQXWZCCZBRFNAUVMDR
MSWTAVQAUUHSHCGTTHSTIMLSBEUDQSUVALAMGKUAUOCUGUARVAUXLMLBSUCUEPGABT
HEFCZSGNULBFEOLNWSCUGZALEGIVLCRAMVKKAHWYAIPNDFMMCSPRFCUGHOFVQGCYAHGF
EQBTUCXAFVDRAUSVLRLCMCJLIPSOOOBMSSCUGUOVOXDQXUHSMDGYRDRQITHNHUMTKCA
GSETGJASWFUNVDHBQGCAIYOENCVTLJQFKSHRGZAQARDBEMKAIDBQNJBMDQDECAUUOALG
NAGCPAPVSVOYIULRLO

Sabendo-se que se trata de uma cifra de Vigenère, ele pode buscar sequências de 3 letras repetidas para tentar descobrir qual o comprimento da palavra-chave utilizada na cifração. Essa busca pode ser facilmente realizada com o comando "Localizar", do software Microsoft® Word. Dessa forma, inicia-se buscando as sequências iniciais do texto cifrado: "QNV", "NVY", "VYE", e assim sucessivamente. Rapidamente, é possível encontrar a sequência "EDA", que se repete 2 vezes, com distância entre as repetições de 21 letras. Portanto, a palavra-chave provavelmente é um dos divisores não unitários de 21, que são 3, 7 e 21. Prossegue-se na busca por sequências de 3 letras repetidas, e encontra-se facilmente a sequência "EGI", que também se repete 2 vezes, com uma distância entre as repetições de 686 letras. Portanto, analogamente, a palavra-chave provavelmente é um dos divisores não unitários de 686, que são 2, 7, 14, 49, 98, 343 e 686. O único número comum nas 2 listas de divisores é o número 7. Portanto, provavelmente a palavra-chave possui 7 letras. As repetições das sequências "EDA" e "EGI" podem ser visualizadas na figura 7.

Figura 7 – Repetições das sequências “EDA” e “EGI”



Sabendo-se que a palavra-chave possui 7 letras, basta separar as letras da mensagem cifrada de acordo com sua posição em relação às letras da palavra-chave, e tentar quebrar 7 cifras de substituição monoalfabética. Iniciando-se pelas letras do texto cifrado que correspondem à primeira letra da palavra-chave, deve-se agrupar todas as letras que ocupam posições congruentes a 1 *modulo* 7, isto é, deve-se agrupar as letras das posições 1ª, 8ª, 15ª e assim sucessivamente, até o final da mensagem cifrada. Isso pode ser feito sem maiores dificuldades utilizando-se o software Microsoft® Excel. Pode-se utilizar o mesmo software para realizar uma contagem de letras e análise de frequência. O conjunto de letras correspondentes às posições congruentes a 1 *modulo* 7 da mensagem cifrada, bem como suas respectivas frequências, podem ser observadas na tabela 1.

Tabela 1 – Letras Cifradas com 1ª Letra da Palavra-Chave e Frequências

QAGAUUEEAYDHZQMMXMCQAEQQUQADUADQMOUAMUQBPOQADAEYBUFAXPUMUDAFUD
UZQMQCPEOMFMMAPEZOQBZUUZNMAGMQMCASAZFCGMYMCQFXUMOCXMQMEFQEQZEQ
DAPY

LETRA	FREQUÊNCIA	LETRA	FREQUÊNCIA
M	14%	B	2%
Q	14%	G	2%
A	13%	H	1%
U	11%	N	1%
E	7%	S	1%
D	5%	I	0%
Z	5%	J	0%
C	5%	K	0%
F	5%	L	0%
O	4%	R	0%
P	4%	T	0%
X	3%	V	0%
Y	3%	W	0%

As frequências de ocorrência das letras podem, enfim, ser comparadas com as frequências de ocorrência das letras no idioma em que a mensagem foi escrita pois, agora, trata-se de uma sequência de letras cifrada de maneira monoalfabética, isto é, com apenas um alfabeto cifrado. Como se sabe que o texto claro foi escrito em português, basta comparar suas frequências com as frequências de ocorrências das letras da língua portuguesa. Nesta dissertação, as frequências das letras em língua portuguesa foram obtidas a partir do trabalho de (QUARESMA, 2008), que realizou uma compilação de frequências a partir da contagem de letras em 141 obras de 47 autores consagrados da língua portuguesa. Essas frequências podem ser observadas na tabela 2.

Tabela 2 – Frequências das Letras na Língua Portuguesa (QUARESMA, 2008)

LETRA	FREQUÊNCIA	LETRA	FREQUÊNCIA	LETRA	FREQUÊNCIA
a	13,52192%	q	1,22637%	ú	0,04994%
e	12,24526%	g	1,21076%	ô	0,03999%
o	10,35746%	b	1,05156%	y	0,02784%
s	7,91774%	f	0,98032%	â	0,02731%
r	6,73638%	ã	0,75899%	k	0,00593%
i	5,73063%	z	0,46639%	w	0,00514%
d	5,09429%	ç	0,44653%	ü	0,00285%
n	4,90934%	á	0,41247%	è	0,00107%
m	4,66504%	é	0,40459%	ò	0,00048%
u	4,42039%	j	0,37186%	ñ	0,00017%
t	4,17410%	x	0,21527%	ì	0,00009%
c	3,24447%	í	0,16822%	î	0,00005%
l	3,05857%	ó	0,16582%	ù	0,00004%
p	2,42987%	ê	0,12496%	û	0,00003%
v	1,70539%	à	0,09055%		
h	1,46360%	õ	0,06997%		

Portanto, comparando-se as frequências das letras cifradas utilizando-se a 1ª letra da palavra-chave com as frequências da língua portuguesa, pode-se inferir algumas coisas sobre a palavra-chave. Primeiramente, analisando-se a letra de maior frequência nas letras cifradas, a letra "M", observa-se que, provavelmente, corresponda à letra "a" da mensagem clara, pois essa é a letra de maior frequência na língua portuguesa. Para que isso tenha ocorrido, a letra "a" deve ter sido cifrada utilizando-se o alfabeto cifrado que se inicia com a letra "M". Portanto, pela análise da letra de maior frequência, chega-se à possibilidade de a primeira letra da palavra-chave ser a letra "M". Isso pode ser confirmado analisando-se as frequências da segunda e da terceira letra mais frequentes nas letras cifradas. Como a letra "Q" é a segunda mais frequente nas letras cifradas, deve corresponder à letra "e" do texto claro, assim como a letra "A", que é a terceira mais frequente nas letras cifradas, deve corresponder à letra "o" do texto claro. Essa correspondência ocorre exatamente quando se utiliza o alfabeto cifrado que se inicia com a letra "M". Portanto, confirma-se a hipótese de que a primeira letra da palavra-chave é a letra "M".

Adotando-se o mesmo procedimento para cada um dos outros 6 grupos de letras, correspondentes às posições congruentes à 2, 3, 4, 5, 6 e 0 *modulo* 7, obtém-se cada uma das demais letras da palavra-chave. Fica de exercício para o leitor que se sentir desafiado encontrá-las. Para os demais leitores, segue na figura 8 a palavra-chave completa, bem como o texto claro, obtido pela decifração da mensagem utilizando-se essa palavra-chave.

Figura 8 – Palavra-Chave e Texto Claro

Palavra-Chave: MACHADO

entreamortedoquincasborbaeaminhamediaramossucessosnarradosnaprimeirapartedolivroopri
ncipaldelesfoiainvencaodoemplastobrascubasquemorreucomigoporcausadamolestiaqueapanh
eidivinoemplastotumedariasoprimeirologarentreoshomensacimadacienciaedariquezaporqueer
asagenuinaedirectainspiracaodoceuoacasodeterminouocontrarioeaivosficaiseternamentehipo
condriacosesteultimocapituloetododenegativasnaoalcanceiacelebridadedoemplastonaofuimini
stronaofuicalifanaoconheciocasamentoverdadeequeaoladodessasfaltascoubemeaboafortunad
enaocompraropaocomosuordomeurostomaisnaopadeciamortededonaplacidanemasemideme
nciadoquincasborbasomadasumascousaseoutrasqualquerpessoaimaginaraquenaohouvemingu
anemsobraeconseguintementequesaiquitecomavidaeimaginaramalporqueaochegaraesteoutro
ladodomisterioacheimecomumpequenosaldoqueeaderradeiranegativadestecapitulodenegativ
asnaotivefilhosnaotransmitianenhumacreaturaolegadodanossamiseria

Percebe-se, portanto, que o texto claro se trata do último capítulo da obra *Memórias Póstumas de Brás Cubas*, de Machado de Assis.

1.2.6 *One-Time Pad*, a Cifra Perfeita

Com a quebra da cifra de Vigenère, os criptógrafos estavam, mais uma vez, perdendo a batalha contra os criptoanalistas. Ninguém podia garantir que as mensagens secretas que circulavam cifradas com a cifra de Vigenère não seriam quebradas por alguém que conhecesse as estratégias desenvolvidas por Babbage e Kasiski. Portanto, era necessário e urgente que se desenvolvesse uma nova cifra, mais forte que a geração anterior. Os criptógrafos começaram analisando qual foi a fraqueza que permitiu a criptoanálise da cifra de Vigenère, e concluíram que sua grande vulnerabilidade era o fato de os alfabetos cifrados se repetirem com alguma periodicidade, igual ao comprimento da palavra-chave. Era essa periodicidade que permitia que se transformasse o problema de uma substituição polialfabética em várias substituições monoalfabéticas. Portanto, para sanar esse problema, bastaria que se utilizasse uma palavra-chave tão longa quanto a própria mensagem a ser cifrada, de forma que não houvesse mais a periodicidade dos alfabetos cifrados, observada com palavras-chave menores. Apesar da dificuldade de se utilizar uma palavra-chave tão longa, aparentemente a segurança que esta palavra-chave trazia à cifra compensava o esforço. Mas, com uma análise mais precisa, pode-se verificar que, mesmo com uma palavra-chave de comprimento tão grande quanto a própria mensagem, essa cifra ainda pode ser decifrada por um criptoanalista persistente. Ele pode proceder da seguinte forma: sabendo em qual idioma a mensagem foi cifrada, ele sabe que, provavelmente, algumas palavras mais frequentes naquele idioma estarão presentes na mensagem. Na língua portuguesa, por exemplo, de acordo com (QUARESMA, 2008), as palavras com 3 letras mais frequentes são “que”, “não”, “com” e “uma”, nesta ordem. Portanto, o criptoanalista pode supor que essas

palavras frequentes aparecerão ao longo do texto. Ele pode alinhar essas palavras em posições aleatórias do texto cifrado, e verificar qual deveria ser o "pedaço de palavra-chave" necessário para que aquele trecho fosse cifrado conforme a palavra comum da língua que ele alinhou naquele local. Se esse pedaço de palavra-chave obtido fizer algum sentido no idioma da mensagem, isso é uma pista de que, de fato, aquele é um pedaço autêntico da palavra-chave. Esse método é bastante exaustivo, pois exige muitas tentativas de se posicionar as palavras comuns do idioma no texto cifrado e obter pedaços de palavra-chave, a maioria deles sem qualquer sentido. Mas um bom criptoanalista, com bastante dedicação e um pouco de sorte, poderia quebrar essa cifra. Seria necessário um modelo de cifra ainda mais segura.

Novamente, analisando onde estava a fraqueza da cifra, os criptógrafos perceberam que, mesmo com palavras-chave de mesmo comprimento da mensagem original, a cifra poderia ser quebrada porque o criptoanalista poderia obter pistas sobre pequenos pedaços da palavra-chave alinhando palavras comuns no idioma aleatoriamente no texto. Portanto, a fraqueza da cifra estava no fato de o criptoanalista conseguir identificar que esses pequenos pedaços da palavra-chave faziam algum sentido naquele idioma, sendo assim bons candidatos para estarem corretos de fato. Portanto, para impedir esse tipo de ataque, bastaria que a palavra-chave não fizesse qualquer sentido em qualquer idioma. Indo mais além, caso a palavra-chave não contivesse nenhum padrão reconhecível em sua estrutura, o criptoanalista não teria pistas se estaria descobrindo a palavra-chave correta ou não. Seguindo por esse caminho, o major Joseph Mauborgne, na época diretor de pesquisa criptográfica do exército americano, introduziu o conceito de chave aleatória, que consistia em uma sequência de letras desprovida de qualquer padrão ou significado (SINGH, 2011). O major Mauborgne propunha o uso de um bloco de folhas de papel, com uma chave aleatória em cada página do bloco, e cada página deveria ser usada apenas uma vez para cifrar a mensagem, para não dar a oportunidade de o criptoanalista comparar 2 mensagens cifradas com a mesma página. Nascia aí o conceito de cifra *one-time pad*. Essa cifra, inicialmente descrita por Gilbert Vernam, em 1917, para a cifração de mensagens telegráficas, não utilizava letras, mas sim *bits* como carregadores de informação. Como ela foi concebida para ser utilizada com o telégrafo, antes de se cifrar a mensagem, era necessário codificá-la, de acordo com o padrão telegráfico, resultando em uma sequência de zeros e uns, que na terminologia atual são chamados de bits. Essa sequência de bits deveria ser "mesclada" com a sequência de bits correspondente à palavra-chave, obtendo-se assim o texto cifrado, também em bits. A maneira como se deveria mesclar a mensagem clara com a palavra-chave, ambas em bits, é a operação binária conhecida como *XOR* ou *exclusive or*, que opera da seguinte forma: a cada par de bits de entrada, a operação atribui um bit de saída de acordo com a paridade dos bits de entrada, isso é, caso os bits de entrada sejam iguais (ambos zeros ou ambos uns), a operação retorna o bit zero, do contrário, a operação retorna o

bit um. É imediato perceber que essa operação corresponde simplesmente à adição em $\mathbb{Z}_2$. O problema da cifra proposta por Vernam é que ela reutilizava chaves, tornando-a suscetível a ataques. Portanto, o verdadeiro conceito de *one-time pad*, com utilização única de cada chave, foi aquele proposto pelo major Mauborgne (SINGH, 2011). A cifra *one-time pad*, conforme foi provado por Shannon, era um exemplo de cifra que possuía a propriedade que ele denominou segredo perfeito (*perfect secrecy*), isto é, o texto cifrado não fornecia qualquer informação adicional a respeito da mensagem clara. Dessa forma, tratava-se de uma cifra perfeita no quesito segurança, ela era completamente imune aos ataques dos criptoanalistas.

Porém, a cifra *one-time pad*, apesar de ser imune aos ataques dos criptoanalistas, não resolvia todos os problemas da criptografia da época, pois sua utilização é difícil de ser implementada, por uma série de motivos. Primeiramente, porque é necessário que emissor e receptor tenham o mesmo bloco de palavras-chave a serem usadas uma única vez em cada mensagem trocada. A logística de distribuição desses blocos frequentemente é bastante complexa, o que gera muitos custos. Outro problema dessa cifra é a dificuldade de se gerar chaves realmente aleatórias em grande quantidade. A aleatoriedade das chaves não é simples de ser obtida, sendo frequentemente retirada de fenômenos naturais, como decaimento radioativo e ruído de diodos. Na prática, utilizam-se chaves denominadas pseudoaleatórias, que consistem em chaves produzidas a partir de chaves realmente aleatórias (denominadas *seeds* ou sementes), e que guardam a maior parte das propriedades estatísticas da aleatoriedade necessárias para a aplicação criptográfica segura. Para mais detalhes sobre geradores de chaves pseudoaleatórias, pode-se consultar (JUNIOR, 2014). Portanto, apesar de completamente segura, a cifra *one-time pad* é extremamente difícil de ser viabilizada na prática, devido à complexa logística de distribuição dos blocos de chaves e à dificuldade na geração de chaves verdadeiramente aleatórias. Devido a toda essa necessidade logística, o uso dessa cifra só se justifica em casos de comunicações ultrassecretas, que podem arcar com esse custo logístico. Um exemplo de comunicação que utiliza essa cifra é o chamado "telefone vermelho" entre os presidentes dos Estados Unidos e da Rússia, uma linha direta que permite que Washington se comunique com Moscou de maneira segura (SINGH, 2011).

1.2.7 Criptografia na Era dos Computadores

Com o surgimento dos computadores digitais, a maneira como a informação trafegava mudou, e isso exigiria que a criptografia responsável pela segurança dessas informações também mudasse. Não seria possível utilizar palavras-chave no universo dos computadores, porque eles lidam apenas com sequências de bits. Dessa forma, a primeira modificação necessária para que as cifras pudessem ser utilizadas em ambi-

entes de computador é a representação, tanto da mensagem clara quanto da chave da cifra, em uma linguagem adequada para a manipulação por meio de um computador digital, ou seja, uma sequência de bits. Para isso, é necessário que se estabeleça um código que faça a correspondência entre cada elemento da mensagem ou da chave em uma sequência de bits. Um protocolo muito utilizado para fazer essa correspondência é o *American Standard Code for Information Interchange*, ou simplesmente ASCII, como é mais conhecido. O padrão ASCII codifica as letras do alfabeto ocidental, maiúsculas e minúsculas, além de uma série de outros símbolos utilizados na escrita, em uma sequência de 7 bits. Porém, o ASCII apresenta uma desvantagem para os usuários da língua portuguesa, porque ele não permite a codificação de caracteres acentuados nem do caractere "cê-cedilha". Após a conversão da mensagem clara e da chave em uma sequência de bits utilizando-se um protocolo de codificação, por exemplo o ASCII, pode-se finalmente utilizar as cifras para cifrar a mensagem. As cifras aplicadas a sequências de bits funcionam da mesma maneira que aplicadas em sequências de letras, isto é, se baseiam nos princípios de transposição e substituição. Na transposição, ocorre um rearranjo dos bits em uma sequência distinta da sequência original, ao passo que na substituição ocorre a troca do valor dos bits. Qualquer cifra implementada em um computador, por mais complexa que seja, consiste em uma sequência de transposições e substituições de bits, aplicadas em alguma ordem específica. Uma característica interessante das transposições de sequências de bits é que pode haver transposição entre bits provenientes de letras diferentes, pois cada letra é representada por mais de um bit.

No universo das cifras que trabalham com sequências de bits, há 2 tipos básicos de cifras, que funcionam de maneiras distintas: as *Stream ciphers* e as *Block ciphers*. As Stream ciphers realizam a cifração de cada bit, individualmente, por meio de uma operação *XOR* realizada entre o bit da mensagem e o bit da *key stream*, que é uma espécie de chave que é gerada a partir da chave da cifra, ou a partir de uma combinação entre a chave da cifra e a própria sequência binária que está sendo gerada (*ciphertext*). Caso a *key stream* dependa apenas da chave da cifra, essa cifra é denominada síncrona, caso dependa também do *ciphertext* que está sendo gerado, essa cifra é denominada assíncrona. É possível observar que, como a *Stream cipher* opera bit a bit, ela não realiza transposições na mensagem, apenas substituições. Por outro lado, as denominadas *Block ciphers*, ou cifras de bloco, realizam a cifração de blocos inteiros de bits da mensagem clara, de tamanho fixo, obtendo-se assim blocos inteiros de bits de mensagem cifrada, com o mesmo tamanho dos blocos de bits da mensagem original. Nesse tipo de cifra, pode ocorrer tanto a transposição quanto a substituição de bits, promovendo assim uma maior segurança para a cifra. Neste trabalho não serão abordadas as *Stream ciphers* com mais detalhes. Para uma abordagem mais detalhada, podem ser consultados (PAAR C.; PELZL, 2010) ou (STINSON, 2006).

Com o avanço da tecnologia dos transistores, foi possível produzir computadores cada vez mais baratos e compactos, de forma que seu uso foi aumentando cada vez mais. Por volta da década de 1960, cada vez mais empresas utilizavam computadores no seu dia a dia, e buscavam métodos de cifração para proteger as suas informações mais sensíveis, como por exemplo informações sobre transações bancárias. Nesse momento, a utilização em larga escala da criptografia saiu dos círculos militares e governamentais, passando a fazer parte do dia a dia de segmentos da comunidade empresarial. Com essa enorme expansão da atividade computacional e criptográfica no mundo, houve também uma enorme demanda por cifras eficientes e seguras. Nessa época, surgiram várias cifras distintas, que eram utilizadas pelas empresas sem qualquer controle. Isso acarretou diversos problemas por não haver uma padronização das cifras, pois cada usuário tinha a liberdade de escolher a sua cifra, dificultando assim a troca de informações entre entes que utilizassem cifras diferentes. Para tentar sanar esse problema de padronização, em 1973, o *National Bureau of Standards - NBS* norte-americano, (atual *National Institute for Standards and Technology - NIST*) solicitou propostas de cifras a serem apresentadas pela comunidade de criptografia mundial, com o objetivo de escolher uma cifra e torná-la padrão da troca de mensagens criptografadas no âmbito federal dos Estados Unidos. Várias cifras foram apresentadas, mas a vencedora foi a cifra *Lucifer*, uma cifra de blocos de propriedade da empresa IBM. A cifra *Lucifer* foi desenvolvida pelo pesquisador Horst Feistel, quando ele trabalhava no laboratório Thomas J. Watson da IBM, nos arredores de Nova York. A versão original da cifra Lucifer trabalhava com blocos de 64 bits de texto claro e utilizava chaves de 128 bits. Porém, o NBS, aconselhado pela *Nacional Security Agency - NSA* norte-americana, solicitou que a cifra fosse modificada, de forma que o novo tamanho das chaves deveria ser de 56 bits. Essa modificação reduziu a segurança da cifra, de forma que, acredita-se, a própria NSA seria capaz de quebrar a cifra com esse novo comprimento de chave, motivo pelo qual foi solicitada a mudança na cifra original, que possuía chaves com 128 bits (SINGH, 2011). Após essa mudança, a cifra foi rebatizada de *Data Encryption Standard - DES*, e passou a ser usada como o padrão de toda a criptografia não sensível na esfera federal norte-americana, com a publicação da *Federal Information Processing Standard - FIPS 46*, em 1977 (NIST, 1999). O DES figurou como padrão de criptografia norte-americana por 22 anos, pois se mostrou uma cifra robusta e resistente aos diversos tipos de ataques disponíveis na época. Como se trata de uma cifra de bloco, sua força reside na combinação de operações de transposição e substituição de bits, de modo a proporcionar ao texto cifrado 2 propriedades fundamentais para que uma cifra resista aos ataques dos criptoanalistas: a difusão e a confusão. Esses 2 termos foram introduzidos na literatura criptográfica por Shannon, com seus estudos sobre criptografia e teoria da informação, e podem ser resumidos da seguinte forma: a difusão em uma cifra consiste em, a partir de uma pequena mudança no texto claro, por exemplo 1 bit, obter-se uma grande mudança

no texto cifrado, em vários bits. Essa propriedade é importante pois, a partir do texto cifrado, não se obtém informações significativas a respeito das propriedades estatísticas do texto claro. Quem proporciona a difusão é geralmente a operação de transposição de bits e suas variações. Por sua vez, a propriedade de confusão em uma cifra consiste em cada bit do texto cifrado depender de vários bits da chave, de forma a se ter uma relação obscura entre os bits do texto cifrado e os bits da chave. Quem proporciona a confusão é geralmente a operação de substituição de bits e suas variações. Shannon propôs que, para se ter uma cifra forte, deve-se concatenar várias operações que provoquem tanto difusão quanto confusão ao bloco de bits a ser cifrado. As cifras que promovem essas sequências de difusão e confusão são denominadas cifras produto, e o DES é uma delas, motivo pelo qual se mostrou uma cifra resistente por bastante tempo. Os detalhes a respeito do funcionamento da cifra DES serão apresentados no Capítulo 2, Seção 2.1.

No final da década de 90, o NIST reconheceu que a cifra DES já não era mais suficientemente segura para as aplicações de criptografia governamental da época. Em 1999, em um concurso promovido pelo *RSA Laboratories Inc.*, o DES foi quebrado no tempo recorde de 22 horas e 15 minutos, por um grupo de aproximadamente 100.000 computadores pessoais, conectados pela internet, que trabalhavam juntos na tentativa de quebrar a cifra por força bruta (SALOMON, 2006). Portanto, era necessário encontrar um substituto para o DES. Como uma medida paliativa, em 1999, o NIST recomendou o uso de uma cifra semelhante ao DES, porém adaptada, denominada triploDES, ou 3DES, que consistia em se aplicar a cifra DES 3 vezes consecutivas, cada uma delas com uma chave diferente (NIST, 1999). Essa cifra, apesar de segura, era muito mais lenta computacionalmente que o DES, e sua implementação em software não permitia muitas otimizações. Antes disso, em janeiro de 1997, o NIST lançou um novo concurso para escolher definitivamente uma cifra substituta para o DES. O processo de escolha durou 4 anos e contou com a participação de vários pesquisadores, que propuseram várias cifras, como por exemplo as cifras MARS, RC6, Serpent e Twofish, entre outras. Porém, a cifra vencedora do concurso foi a cifra Rijndael, desenvolvida pelos pesquisadores belgas Vincent Rijmen e Joan Daemen, daí o seu nome Rijndael, uma junção dos nomes de seus desenvolvedores. Assim, em outubro de 2000, o Rijndael foi escolhido pelo NIST como nova cifra padrão para a criptografia não sensível no âmbito federal dos Estados Unidos, e foi denominado *Advanced Encryption Standard - AES*, passando a ser realmente adotado como padrão criptográfico mundial a partir de novembro de 2001. Sua publicação em dezembro de 2001, por meio da FIPS 197, tornou pública a cifra AES (NIST, 2001). Trata-se de uma cifra de bloco, que funciona de maneira semelhante ao DES, promovendo várias rodadas que implementam tanto difusão quanto confusão no texto cifrado. O AES trabalha com blocos de 128 bits de texto claro, e existem 3 possibilidades de tamanho para a chave: podem-se ter chaves de 128 bits e, nesse caso, o algoritmo necessitará de 10 rodadas ou "rounds" para cifrar a mensagem,

podem-se ter chaves de 192 bits e, nesse caso, o algoritmo necessitará de 11 rodadas ou, por fim, podem-se ter chaves de 256 bits e, nesse caso, o algoritmo necessitará de 12 rodadas. Observa-se que, mesmo sendo o tamanho das chaves flexível, ainda assim o menor tamanho possível, 128 bits, é mais que o dobro dos 56 bits da chave da cifra DES. Isso dificulta o ataque ao AES por força bruta, porque o número de chaves que devem ser tentadas é absurdamente maior. Atualmente, o AES ainda é usado como padrão de criptografia mundial, dada a sua versatilidade aliada a sua robustez contra os ataques dos criptoanalistas. Os detalhes a respeito do funcionamento da cifra AES serão apresentados no Capítulo 2, Seção 2.2.

1.2.8 Dificuldades na Troca de Chaves e o Surgimento da Criptografia de Chave Pública

Na década de 70, embora a adoção da cifra DES resolvesse com segurança o problema de padronização das cifras utilizadas pelas empresas ao redor do mundo, ainda havia um problema que precisava ser resolvido para viabilizar uma comunicação global segura: a troca de chaves entre as partes. Tudo que se conhecia sobre criptografia até aquele momento, desde as cifras de deslocamento mais primitivas, até o moderno DES, necessitava de uma premissa crucial para o funcionamento: emissor e receptor deveriam conhecer a mesma chave secreta, para que o receptor pudesse decifrar a mensagem cifrada pelo emissor. Esse compartilhamento da chave nem sempre era possível de ser realizado por um meio seguro. Uma chamada telefônica, por exemplo, poderia ser interceptada, ou uma correspondência poderia ser violada. Portanto, muitas vezes a troca de chaves era realizada de maneira presencial, para que fosse garantida a segurança do processo. O problema é que esse método de distribuição de chaves é extremamente caro e ineficiente quando se deseja trocar mensagens cifradas com entidades espalhadas por todo o mundo. Havia a necessidade de se desenvolver alguma técnica que permitisse uma troca segura de chaves, mesmo que o trânsito da informação se desse por meio de um canal inseguro. Nem mesmo se sabia se era possível desenvolver tal técnica, sendo a ideia negligenciada por muitos criptógrafos da época, que acreditavam ser impossível existir algo do tipo (SINGH, 2011). Portanto, quem resolvesse o problema da distribuição de chaves de maneira segura e eficaz, estaria seguramente imortalizando seu nome na própria história da criptografia. Tratava-se da busca pelo santo graal da criptografia do século XX.

Essa busca chegou ao fim no ano de 1976, quando os pesquisadores Whitfield Diffie e Martin Hellman, da Universidade de Stanford, e, de maneira independente, Ralph Merkle, da Universidade da Califórnia, introduziram o conceito de criptografia de chave pública, também denominada criptografia assimétrica. Neste novo modelo criptográfico, não havia a necessidade de um compartilhamento prévio de chave entre

emissor e receptor, pois a técnica permitia justamente esse compartilhamento da chave de maneira segura, mesmo que a troca de informação se desse por meio de um canal inseguro. Em linhas gerais, a ideia desenvolvida pelos pesquisadores foi a seguinte: suponha que Alice deseja compartilhar uma chave secreta com Bob. Para isso, ela coloca a chave secreta em uma caixa, tranca a caixa com um cadeado do qual somente a própria Alice possui a chave, e envia a caixa para Bob. Esse envio pode ocorrer mesmo em um canal inseguro, pois somente Alice possui a chave do cadeado. Quando Bob recebe a caixa, ele não pode abri-la, pois não possui a chave do cadeado de Alice. Bob então coloca um novo cadeado na caixa, paralelamente ao cadeado de Alice, do qual somente o próprio Bob possui a chave, e envia a caixa de volta para Alice, com os 2 cadeados trancados. Esse envio é duplamente seguro, pois, para se abrir a caixa, são necessárias as chaves tanto de Bob como de Alice. Quando Alice recebe a caixa com os 2 cadeados, ela destranca o seu próprio cadeado, e reenvia a caixa para Bob, agora trancada somente com o cadeado do próprio Bob. Finalmente, quando Bob recebe a caixa, ele pode destrancá-la com a chave do seu próprio cadeado e ler o conteúdo da caixa, que corresponde à chave secreta escolhida por Alice. Portanto, agora ambos conhecem uma chave secreta e podem utilizá-la para trocar mensagens utilizando-se uma cifra qualquer. Observa-se que, ao longo de todos os envios da caixa com os cadeados, mesmo que um espião, Eve, conseguisse interceptar a caixa no processo de envio, ele nada poderia fazer para conhecer seu conteúdo, pois não dispunha nem da chave de Alice nem da chave de Bob. Estava resolvido, ao menos de maneira teórica, o problema da troca segura de chaves por meio de canais inseguros. Porém, era necessário encontrar uma maneira de implementar a ideia, de forma que ela pudesse ser aplicada ao universo criptográfico da época. Era necessário que fossem desenvolvidos "cadeados criptográficos", que permitissem a troca de informações na forma de sequências de bits. Diffie e Hellman encontraram uma maneira de operacionalizar essa ideia utilizando as operações de exponenciação em corpos finitos como "funções cadeado". Essas funções são adequadas para o propósito porque são funções denominadas funções de uma única via (*one-way functions*), isto é, dados 2 números naturais, denominados α e x, é relativamente fácil realizar computacionalmente a exponenciação $\alpha^x \equiv k \; mod(p)$ em um $\mathbb{Z}_p^*$ qualquer. Porém, dados k e α, é computacionalmente custoso encontrar o expoente $x = log_\alpha k$ em $\mathbb{Z}_p^*$. Esse problema é conhecido como problema do logaritmo discreto, e sua intratabilidade computacional é a base para o funcionamento do algoritmo de troca de chaves proposto por Diffie e Hellman. Esse algoritmo de troca de chaves por meio de um canal inseguro ficou conhecido na comunidade criptográfica como algoritmo de troca de chaves de Diffie-Hellman.

Além da divulgação do algoritmo inovador para troca segura de chaves, em 1975, antes mesmo da descoberta desse algoritmo, os pesquisadores publicaram um artigo em que propuseram as bases de algo também inovador para aquela época, eles

propuseram o conceito que hoje é conhecido como criptografia de chave pública ou criptografia assimétrica. Nesse novo conceito de criptografia, a troca de mensagens entre emissor e receptor não necessitaria mais de uma chave secreta conhecida por ambos. Para exemplificar o novo conceito, vamos supor que Alice desejasse mandar uma mensagem criptografada para Bob. Para isso, Alice utilizaria uma chave de Bob, conhecida como chave pública, que é amplamente divulgada por Bob. De posse da chave pública de Bob, Alice a utilizaria para cifrar a mensagem e a enviaria para Bob. Ele, ao receber a mensagem cifrada, utilizaria uma outra chave, conhecida como chave privada, para decifrar a mensagem e ler seu conteúdo. Para garantir a segurança do processo, somente Bob deve conhecer sua chave privada. A ideia inovadora deste tipo de criptografia está na utilização de funções específicas, as chamadas funções de uma única via, que, com algum engenho, permitem que somente o detentor da chave privada possa decifrar uma mensagem cifrada com a chave pública, e que a obtenção da chave privada a partir da chave pública seja um processo computacionalmente custoso. Portanto, mesmo que o espião Eve pudesse interceptar a mensagem cifrada com a chave pública de Bob, ele não poderia decifrá-la, pois não conhece a chave privada de Bob, e é difícil obtê-la conhecendo-se somente a chave pública de Bob. Portanto, esse tipo de criptografia garantia o segredo da mensagem enviada em um canal inseguro, sem que fosse necessária a utilização de uma chave secreta comum conhecida somente por Alice e Bob.

Após a divulgação das ideias novas sobre troca de chaves e criptografia segura por meio de canais inseguros, vários pesquisadores da comunidade criptográfica se debruçaram sobre o problema de encontrar funções adequadas para esse propósito, as chamadas funções de mão única, e esses cientistas buscavam desenvolver uma cifra realmente eficiente e aplicável à troca de informações criptografadas por meio de sequências de bits, que são a linguagem dos computadores. Em 1977, os pesquisadores do *Massachusetts Institute of Technology* Ron Rivest, Adi Shamir e Leonard Adleman desenvolveram uma cifra completamente nova, que utilizava a dificuldade de se fatorar computacionalmente números compostos muito grandes como função de uma única via, o que tornava a sua cifra forte. Eles deram o nome de RSA para a cifra, em referência às iniciais de seus nomes. Pela sua simplicidade e facilidade de implementação, o RSA foi imediatamente adotado pelos governos e empresas que necessitavam de cifras seguras para mensagens que circulariam em canais inseguros. Os 3 pesquisadores do MIT fundaram uma empresa para comercializar sua cifra, a *RSA Data Security Inc*. Apesar de ser uma cifra segura e simples de se implementar, o RSA exige uma capacidade de processamento computacional relativamente alta, pois trabalha com exponenciações de números muito grandes. Dessa forma, o RSA não foi utilizado de maneira ampla como uma cifra para troca de mensagens longas, pois a necessidade computacional se tornaria proibitiva. Alternativamente, utilizava-se o RSA para realizar a troca de pequenas

mensagens, frequentemente chaves, que pudessem ser utilizadas como chaves secretas na criptografia simétrica, por exemplo com a cifra AES, que é computacionalmente muito mais "leve" que o RSA. Essa estratégia de se utilizar uma cifra de criptografia assimétrica para trocar as chaves da criptografia simétrica é a base de funcionamento dos atuais protocolos SSL/TLS de comunicação segura na internet.

Apesar de a cifra RSA ser extremamente simples de se implementar, ela exige uma capacidade computacional relativamente grande para a troca de mensagens longas. Também, sua segurança reside na dificuldade computacional de se fatorar números compostos muito grandes. Portanto, à medida que a capacidade computacional avança, com a fabricação de processadores cada vez mais potentes, para se manter a segurança do RSA é necessário que sejam utilizados números cada vez maiores, para dificultar a fatoração. Essa é uma das grandes desvantagens do RSA, sua necessidade de chaves cada vez maiores ao longo dos anos, de forma que a necessidade de capacidade computacional para se trabalhar com essas chaves enormes é cada vez maior. Buscando contornar esse problema, em 1986, os pesquisadores Neal Koblitz e Victor Miller, independentemente, propuseram as bases do que se conhece atualmente como criptografia de curvas elípticas - ECC. Tratava-se de um tipo de criptografia assimétrica que utilizava uma função de uma única via bastante inusitada para conferir segurança à cifra: a segurança da ECC se baseava na intratabilidade computacional do problema do logaritmo discreto adaptado ao universo da operação de multiplicação de pontos de curvas elípticas definidas sobre corpos finitos por um número inteiro. Para o entendimento desse tipo de cifra assimétrica exigia-se um conhecimento matemático muito mais profundo do que para o pleno entendimento do RSA, motivo pelo qual a criptografia ECC demorou um pouco para ser aceita e utilizada de maneira mais ampla pela comunidade de criptografia mundial. A principal vantagem da criptografia ECC sobre o RSA é o reduzido tamanho das chaves necessárias para se garantir o mesmo nível de segurança. Enquanto as chaves recomendadas atualmente para se garantir segurança com o RSA possuem comprimento em torno de 4096 bits, as chaves recomendadas para cifras de ECC possuem comprimento em torno de 521 bits para prover o mesmo nível de segurança (ICP-BRASIL, 2018).

O surgimento da ideia de criptografia assimétrica revolucionou a utilização criptográfica conhecida até aquele momento, pois o conceito de segurança das cifras de criptografia assimétrica é bastante diferente do conceito de segurança das cifras de criptografia simétrica. Enquanto na criptografia simétrica a segurança da cifra reside na chave secreta, a força das cifras de criptografia assimétrica não está somente no caráter secreto da chave, mas sim nas limitações computacionais para se realizar certos tipos de operações matemáticas em um tempo reduzido. Isto é, na computação clássica, não se conhecem algoritmos de complexidade polinomial para o tratamento dos problemas matemáticos envolvidos nas cifras de criptografia assimétrica. Em li-

nhas gerais, a complexidade de um algoritmo é dita polinomial quando o seu tempo de execução é limitado superiormente por um polinômio $P(n)$ de grau finito, em que n é o comprimento da entrada do algoritmo. Com a capacidade dos computadores clássicos da atualidade, os problemas para os quais existem algoritmos de complexidade polinomial podem ser resolvidos computacionalmente, o que inviabiliza a sua utilização em criptografia. Portanto, para se garantir a segurança dos algoritmos de criptografia assimétrica, essas cifras devem ser baseadas em problemas matemáticos para os quais não se conheçam algoritmos clássicos de complexidade computacional polinomial. Também, a segurança desses algoritmos depende de outro fator crucial: a inexistência de compudadores quânticos na atualidade. Isso ocorre pois já se conhecem algoritmos passíveis de implementação em processadores quânticos e que resolvem, em tempo polinomial, problemas intratáveis da computação clássica. Por exemplo, em 1994, o pesquisador Peter Shor, do *AT&T Bell Laboratories*, desenvolveu um algoritmo quântico capaz de resolver em tempo polinomial problemas intratáveis da computação clássica. Para mais detalhes sobre o trabalho de Shor, podem ser consultados (SHOR, 1997) e (COUTINHO, 2000). Neste trabalho, serão abordados 2 tipos de algoritmo de criptografia assimétrica: a cifra RSA, que baseia a sua segurança no problema da fatoração de inteiros grandes, e a cifra baseada em curvas elípticas, que utiliza a dificuldade de se obterem certos parâmetros a partir de multiplicações de pontos de curvas elípticas por números inteiros, como base para a sua segurança. Os detalhes a respeito do funcionamento do RSA serão apresentados no Capítulo 3, Seção 3.1, e os detalhes a respeito do funcionamento das cifras baseadas em ECC serão apresentados no Capítulo 3, seção 3.2.

1.2.9 Necessidade da Assinatura Digital

Quando se faz uma análise mais criteriosa do modelo criptográfico de chave pública, surge um questionamento sobre a autenticidade do emissor da mensagem. Isso ocorre pois, como o receptor deve divulgar sua chave pública por meio de um canal aberto, é possível que uma entidade estranha utilize essa chave para criptografar e enviar uma mensagem ao receptor, como se se tratasse do emissor conhecido. Em outras palavras, uma entidade estranha qualquer pode "fingir" ser um emissor conhecido pelo receptor, e enviar mensagens se passando por esse emissor conhecido. Portanto, na troca de mensagens por meio de cifras de chave pública, é necessário também que se garanta a autenticidade da mensagem. Essa autenticidade pode ser assegurada utilizando-se um método bastante simples, semelhante a uma "assinatura" na mensagem, tal qual uma assinatura feita em um documento físico, para atestar sua autenticidade. Essa "assinatura" funciona como uma prova de que realmente se trata do emissor legítimo da mensagem, pois ela é realizada com a chave privada do emissor, e se supõe que nenhuma outra entidade estranha a conheça. Portanto, sem a

chave privada do emissor, torna-se impraticável uma "falsificação" da assinatura do emissor autêntico. Como a troca de mensagens ocorre com a informação codificada em sequências de bits, o processo ficou conhecido como assinatura digital. Em linhas gerais, o processo consiste em o emissor "assinar" a mensagem utilizando para isso uma característica que só ele conheça. Como no processo de criptografia assimétrica a chave privada é uma característica que somente seu próprio detentor conhece, o emissor pode aproveitar sua própria chave privada para assinar a mensagem. Dessa forma, o emissor utiliza a sua chave privada para assinar a mensagem, de forma que o correspondente matemático daquela chave privada, isto é, a sua chave pública, poderá ser utilizado para atestar a veracidade da assinatura. Dessa forma, o emissor pode enviar a mensagem cifrada com a chave pública do receptor e, adicionalmente, também a mensagem cifrada com sua própria chave privada, para garantir a sua autenticidade. O problema dessa estratégia é que, como o emissor utiliza a sua chave privada para assinar a mensagem, qualquer entidade com o conhecimento de sua chave pública poderia facilmente decifrar a assinatura e ler o conteúdo da mensagem. Portanto, o emissor não deve assinar a própria mensagem secreta com a sua chave privada, mas sim algum outro tipo de mensagem, que possa ser obtida facilmente a partir da mensagem secreta original, e cujo conteúdo possa ser tornado público sem prejuízo ao caráter secreto da mensagem. Em outras palavras, é necessário que se assine uma espécie de "resumo" da mensagem original, que, caso interceptado por uma entidade estranha, não possa ser lido sem o conhecimento da mensagem original completa. Também é necessário que esse resumo possa ser facilmente gerado pelo receptor a partir da mensagem completa, para que este possa comparar o resumo gerado por ele com o resumo enviado assinado pelo emissor, verificando assim a autenticidade da mensagem. Portanto, em um processo de troca de mensagens utilizando-se assinatura digital, o emissor envia ao receptor a mensagem cifrada utilizando a chave pública do receptor e envia também o "resumo" desta mensagem, cifrado utilizando a sua própria chave privada. O receptor, então, utiliza a sua chave privada para decifrar a mensagem e ler seu conteúdo. Porém, ainda resta dúvida a respeito da autenticidade do emissor. Então, o receptor utiliza a chave pública da entidade que ele supõe ser a emissora e decifra o "resumo". De posse da mensagem clara e do seu "resumo" o receptor pode concluir se o "resumo" foi obtido, de fato, a partir daquela mensagem clara. Caso o emissor seja autêntico, a chave pública utilizada pelo receptor para decifrar o "resumo" levará a uma coincidência com o que se observa na mensagem clara. Porém, caso o emissor não seja autêntico, a utilização da chave pública do suposto emissor para decifrar o "resumo" levará a um texto que não guarda relação com a mensagem clara, podendo assim ser constatada a fraude na autenticidade da mensagem. Dessa forma, garantem-se tanto a integridade da mensagem (pois, caso ela tenha sido alterada por uma entidade estranha, essa entidade não poderá assinar seu "resumo" como o emissor autêntico) quanto a autenticidade

do emissor, pois só ele poderia assinar o "resumo" com sua chave privada. Esse tipo de resumo especial utilizado em processos de assinatura digital é denominado *hash*, e há vários algoritmos conhecidos para a geração de *hash*. O *hash* deve transformar o texto claro da mensagem em um "texto resumido", mas não deve permitir a operação contrária, isto é, a obtenção do texto claro a partir do *hash*. O *hash* também não deve permitir que 2 mensagens diferentes originem o mesmo resumo, pois, nesse caso, poderia haver um falso positivo no processo de autenticação da mensagem. Há muitas outras propriedades desejáveis em um *hash* que não serão abordadas neste trabalho, tampouco serão abordados detalhes sobre o funcionamento dos algoritmos geradores de *hash* utilizados na atualidade. Na prática, muitas vezes não se utiliza o modelo de assinatura digital para garantir a integridade e a autenticidade das mensagens trocadas, mas sim os chamados *MACs - Message Authentication Codes* e *HMACs - Hashed Message Authentication Codes*, que são estruturas que permitem a conferência da autenticidade das mensagens em ambientes de criptografia de chave secreta. A grande vantagem dos MACs e HMACS frente aos protocolos de assinatura digital está no seu custo computacional, que é muito menor que o custo dos algoritmos de assinatura digital, pois estes utilizam cifras de chave pública, enquanto os MACS e HMACS utilizam cifras de chave secreta e funções geradoras de *hash*, que são computacionalmente muito menos custosas. O protocolo de segurança TLS, utilizado na troca de informações criptografadas entre um computador e um site, por exemplo, utiliza tanto assinaturas digitais quando HMACs no seu funcionamento (PAAR C.; PELZL, 2010). Para uma abordagem mais ampla sobre os protocolos de assinatura digital utilizando-se algoritmos geradores de *hash*, podem ser consultados (STINSON, 2006) e (TRAPPE W.; WASHINGTON, 2006), e para uma descrição mais detalhada dos MACs e HMACS pode ser consultado (PAAR C.; PELZL, 2010).

1.2.10 Autoridade Certificadora, Infraestrutura de Chaves Públicas e Certificados Digitais

À medida que surge a possibilidade de o emissor utilizar a sua chave privada para assinar o *hash* da mensagem a ser enviada, atestando assim a autenticidade da mensagem, surge também um problema nesse processo. Por exemplo, se Alice envia uma mensagem cifrada para Bob e também envia um *hash* da mensagem, assinado com a sua chave privada, Bob precisa conhecer a chave pública de Alice para que possa confrontar o *hash* assinado com o *hash* gerado após a decifração da mensagem. Por outro lado, Alice também precisa conhecer a chave pública de Bob, para que possa cifrar a mensagem e enviá-la a Bob. Portanto, é necessário que ambos conheçam as chaves públicas um do outro. Ora, mas como as chaves públicas são ostensivamente divulgadas, isso não deveria ser um problema. Porém, supondo que um espião, Eve,

deseja se passar por Alice, basta que ele divulgue sua própria chave pública como se fosse a chave pública de Alice. Como Bob poderia ter certeza de que aquela chave pública, divulgada na internet, por exemplo, realmente pertence a Alice? Para que essa dúvida não exista, é necessário que a divulgação da chave pública de Alice seja feita por uma entidade que seja de confiança para ambos, emissor e receptor. Essa entidade é conhecida como autoridade certificadora. Uma autoridade certificadora precisa garantir a identidade dos proprietários das chaves públicas que ela divulga. Ela deve realizar uma conferência criteriosa da identidade da pessoa cuja chave pública ela se propõe a divulgar. Por isso, a autoridade certificadora deve ser uma entidade com boa reputação em todo o mundo, para que os usuários da criptografia confiem nos dados por ela divulgados.

A ideia de uma autoridade certificadora resolve o problema da divulgação segura de chaves públicas, porém, surge aí um enorme problema logístico: como uma autoridade certificadora, localizada fisicamente em algum lugar do mundo, pode cadastrar e divulgar as chaves públicas de todos os usuários interessados em utilizar cifras de chave pública e assinatura digital? É bastante improvável que uma entidade consiga lidar sozinha com essa quantidade de informações e validações de identidade dos proprietários das chaves. Para contornar esse problema, as autoridades certificadoras acabam "delegando" a sua responsabilidade para outras entidades que também sejam confiáveis, para que o trabalho de cadastro e divulgação das chaves possa ser realizado de forma distribuída. Essa ideia é a base da estrutura que se conhece como infraestrutura de chaves públicas - ICP, ou *public key infrastructure - PKI*. Essa infraestrutura estabelece uma relação de hierarquia e cooperação entre diversos entes responsáveis pela coleta de dados e divulgação de chaves públicas fidedignas. A forma como essas entidades, denominadas autoridades certificadoras, divulgam as chaves públicas dos usuários, de maneira segura, é conhecida como certificado digital. Um certificado digital é um conjunto de informações padronizadas, que são divulgadas pela autoridade certificadora, contendo os dados do proprietário daquela chave pública que está no certificado, além de outras informações relevantes. Como a autoridade certificadora goza de uma reputação boa na comunidade de usuários, ela "assina" cada um dos certificados que emite, atestando que aquelas informações, de fato, são fidedignas. As autoridades certificadoras podem fornecer certificados para os mais diversos tipos de usuários, sejam pessoas, empresas, sites da internet e até mesmo equipamentos eletrônicos. A utilização de certificados digitais emitidos por autoridades certificadoras reconhecidas é a base dos protocolos de segurança da internet utilizados atualmente, que possibilitam, por exemplo, que se possa realizar compras com cartão de crédito de maneira segura na internet.

No Brasil, a autoridade certificadora de mais alta hierarquia, denominada autoridade certificadora raiz ou AC-raiz, é gerida pelo Instituto Nacional de Tecnologia

da Informação - ITI, uma autarquia federal vinculada à Presidência da República. É o ITI quem realiza a gestão da Infraestrutura de Chaves Públicas Brasileira - ICP-Brasil. A autoridade certificadora raiz dessa infraestrutura "certifica" as autoridades certificadoras de grau hierárquico mais baixo, para que elas possam emitir certificados digitais. Qualquer pessoa física ou jurídica pode solicitar um certificado digital às autoridades certificadoras, arcando com o custo do certificado. Com a modernização e digitalização de processos, tanto no setor público como no setor privado, o número de solicitações de certificados digitais vem crescendo a cada ano no Brasil. Para mais detalhes sobre a ICP-Brasil e seu funcionamento, podem ser consultados (ICP-BRASIL, 2018) e o *site* do ITI na internet.

2 Criptografia de Chave Privada ou Simétrica

Neste capítulo, serão apresentadas as cifras simétricas *Data Encryption Standard - DES* e *Advanced Encryption Standard - AES*.

2.1 *Data Encryption Standard - DES*

O DES é uma cifra bastante conhecida no universo da criptografia, sendo utilizada até hoje em aplicações de criptografia de chave privada, por meio da sua modificação conhecida como triploDES ou 3DES (NIST, 1999). A cifra foi desenvolvida na década de 70 pelo pesquisador da IBM Horst Feistel, por meio de uma modificação da sua cifra *Lucifer*, e foi utilizada como padrão de criptografia mundial por 3 décadas, até que, em 1999, foi anunciada a quebra do algoritmo no tempo recorde de 22 horas e 15 minutos, o que colocou em xeque a confiança na utilização do DES como cifra realmente segura. O DES é uma cifra de criptografia simétrica, isto é, há uma chave secreta compartilhada pelo emissor e pelo receptor, utilizada para cifrar e decifrar as mensagens. A cifra realiza uma sequência de transposições e substituições de bits, promovendo tanto a difusão quanto a confusão, conforme os preceitos estabelecidos por Shannon para se ter uma boa cifra. A cifra trabalha com blocos de 64 bits como dados de entrada, utiliza uma chave secreta de 56 bits, e gera um bloco de saída criptografado de 64 bits. Portanto, trata-se de uma cifra de bloco. Apesar da chave secreta utilizada pela cifra possuir 56 bits, frequentemente essa chave aparece com 64 bits. Porém, somente 56 desses bits serão efetivamente utilizados no processo de cifragem, os demais bits serão desprezados, conforme algumas regras que serão descritas mais à frente. O motivo de se utilizar 64 bits em vez de 56 bits na chave é para aproveitar os 8 bits sobressalentes para realizar detecção de erros na geração, armazenamento e distribuição de sub-chaves geradas ao longo do processo de funcionamento da cifra. Portanto, do ponto de vista de segurança da cifra, esses bits não têm qualquer importância.

2.1.1 Cifração com o DES

A cifra DES funciona de maneira dividida em 3 etapas: primeiramente, ocorre uma permutação inicial no bloco de entrada, denominada *IP*, seguida de um processo de substituição em rodadas ou *rounds*, utilizando-se a chave secreta, e finalizando com uma permutação final, denominada IP^{-1}. O primeiro passo, a IP, é um processo de simples transposição de bits. Ele atua no bloco de 64 bits de entrada e não utiliza a

chave secreta ainda. A maneira como deve ser realizada a permutação inicial dos bits está exemplificada na tabela 3.

Tabela 3 – Permutação Inicial DES

IP

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Essa tabela determina a nova ordem dos bits no bloco. O bit que ocupava 58ª posição, por exemplo, passará a ocupar a 1ª posição, já o bit que ocupava a 50ª posição passará a ocupar a 2ª posição, e assim sucessivamente. A próxima etapa do processo é um pouco mais complexa, e consiste em várias rodadas de substituição de bits, que utilizam uma função criptográfica e uma parte do bloco de bits proveniente da rodada anterior, além da chave secreta específica daquela rodada. Esse processo é composto por 16 rodadas, em que cada rodada utiliza uma chave de comprimento 48 bits, específica para aquela rodada, chamada de *round-key* K_i, e essas chaves são geradas a partir da chave original de 56 bits, por meio de um processo que será descrito um pouco mais à frente. O bloco de 64 bits proveniente da IP é dividido em 2 sub-blocos de 32 bits cada um, denominados L_0 e R_0, sendo L_0 correspondente aos 32 primeiros bits do bloco, e R_0 aos 32 bits finais do bloco. Ao longo do processo, os sub-blocos serão representados como L_i e R_i, em que o índice subscrito representa a rodada em que os sub-blocos se encontram. Em cada rodada, ocorre uma interação entre um dos sub-blocos e a chave da rodada, por meio de uma função criptográfica denominada função-cifra, ou simplesmente f . A função-cifra f possui como parâmetros de entrada o sub-bloco R_{i-1}, de 32 bits, e a round-key K_i, de 48 bits, com i variando conforme a rodada em que a função-cifra está atuando. Inicialmente, ocorre uma expansão do sub-bloco R_{i-1}, denominada E, de 32 bits para 48 bits, conforme indica a tabela 4.

Tabela 4 – Expansão dos Sub-Blocos R_{i-1}

E					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Esse novo sub-bloco expandido, agora com 48 bits, será somado *modulo* 2 com a respectiva round-key de 48 bits, obtida a partir da chave original, de 56 bits. Cada uma das round-keys é obtida a partir da chave original por um processo que será descrito no final desta seção. O resultado da soma *modulo* 2 (conhecida como *XOR*, e representada por $\oplus$) é uma sequência de 48 bits. Em seguida, esses 48 bits são divididos em 8 pequenos blocos de 6 bits, e cada um desses pequenos blocos é submetido a um conjunto de operações, consistindo em 8 substituições realizadas utilizando-se tabelas específicas da cifra DES, chamadas S-boxes. Cada uma das 8 S-boxes recebe um pequeno bloco de 6 bits e, por meio de algumas operações, retorna um bloco menor ainda, de 4 bits. A concatenação dos 8 blocos de 4 bits fornece um novo bloco de 32 bits. As S-boxes realizam uma operação de compressão de bits, pois sua entrada é uma sequência de 48 bits e sua saída uma sequência de 32 bits. O esquema de funcionamento das S-boxes é o seguinte: cada uma das 8 S-boxes, denominada S_j , com $j = 1, 2, \ldots, 8$, é uma matriz 4 x 16, que transforma a sua entrada, que é um dos 8 blocos de 6 bits, $B_j = b_1b_2b_3b_4b_5b_6$, com $j = 1, 2, \ldots, 8$, em um bloco de 4 bits na sua saída. As 4 linhas de S_j são numeradas de 0 a 3, e as 16 colunas de 0 a 15. Concatenando-se os bits b_1b_6 de B_j, obtém-se um número entre 0 e 3 na base 10, e esse número indicará qual linha de S_j será utilizada. Analogamente, concatenando-se os bits $b_2b_3b_4b_5$ de B_j, obtém-se um número entre 0 e 15 na base 10, e esse número indicará qual coluna de S_j será utilizada. Com as coordenadas da linha e da coluna da S-box obtidas anteriormente, identifica-se qual número ocupa a posição correspondente a tais coordenadas em S_j. Por fim, esse número sempre estará no intervalo entre 0 e 15 na base 10, o que corresponde a uma sequência de 4 bits. Essa será a saída da interação entre B_j e S_j. Concatenando-se as sequências de 4 bits obtidas nas 8 interações entre B_j e S_j, $j = 1, 2, \ldots, 8$, obtém-se um novo bloco de 32 bits. Por fim, a última operação realizada pela função-cifra é uma Permutação de bits, denominada P, que é realizada de acordo com a tabela 5.

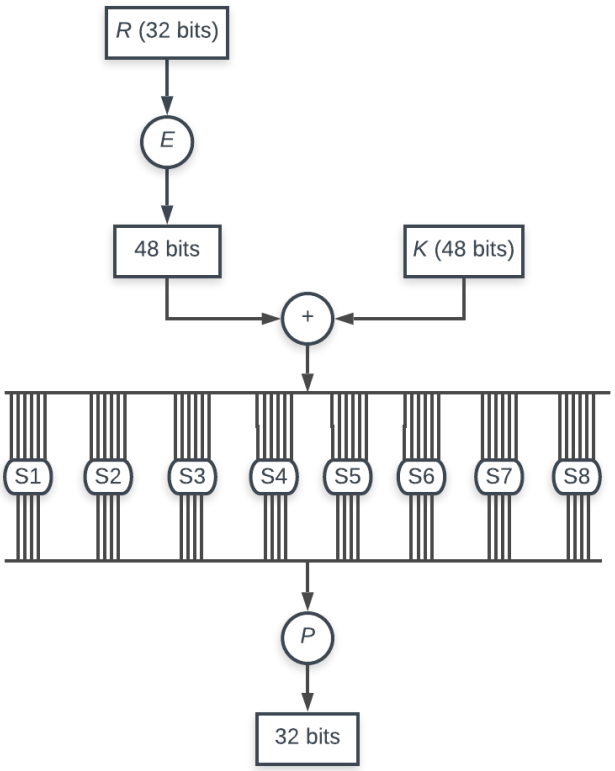
Tabela 5 – Permutação de Bits P da função-cifra

P

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

Portanto, a função-cifra gera um bloco de 32 bits a partir de R_{i-1}, que possui 32 bits, e da round-key K_i, que possui 48 bits. O funcionamento resumido da função-cifra está esquematizado na figura 9.

Figura 9 – Resumo função-cifra



Em cada rodada, o resultado de $f(R_{i-1}, K_i)$ é somado *modulo* 2 com o sub-bloco L_{i-1}, obtendo-se assim o sub-bloco R_i, da próxima rodada. Por sua vez, o sub-bloco L_i é obtido simplesmente atribuindo-se a ele a mesma sequência de bits do sub-bloco R_{i-1}, da rodada anterior. Em resumo, tem-se:

$$R_i = f(R_{i-1}, K_i) \oplus L_{i-1}$$
$$L_i = R_{i-1}$$

Esse processo é realizado em cada uma das 16 rodadas da cifra. Após a obtenção de R_{16} e L_{16}, cada um com 32 bits, realiza-se a concatenação de seus bits, sendo os primeiros 32 bits provenientes de R_{16}, e os 32 bits finais de L_{16}. Essa sequência de 64 bits obtida é denominada pre-output. Por fim, a última etapa da cifra consiste em se aplicar ao pre-output uma permutação final, denominada IP^{-1}, que consiste em uma transposição de bits exatamente inversa à da permutação inicial IP. A permutação IP^{-1} é realizada de acordo com a tabela 6, a figura 10 apresenta um resumo geral da cifra e a tabela 7 contém a 1ª das 8 S-boxes.

Tabela 6 – Permutação Final IP^{-1}

IP^{-1}							
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Figura 10 – Visão Geral da Cifra DES

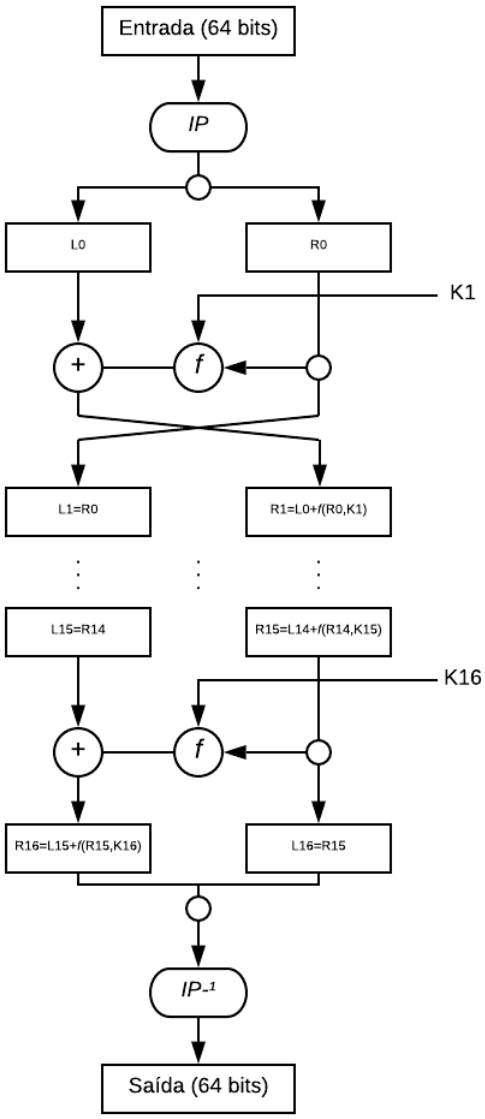


Tabela 7 – 1ª S-Box da Cifra DES

S_1																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	3	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	13	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Para se obter cada uma das *round-keys* K_i, procede-se da forma a seguir. Em alguns casos, a chave original é fornecida como uma sequência de 64 bits em vez de 56 bits, conforme mencionado anteriormente. Nesses casos, 8 bits dessa chave serão desprezados, por meio de uma tabela que despreza os bits que ocupam as posições 8ª,

16^a, 24^a ... 64^a, e permuta os demais bits, obtendo-se assim uma sequência de 56 bits, que corresponde efetivamente à chave. Essa primeira operação a ser realizada com os 64 bits da chave é denominada *permutation choice 1 - PC-1*. A operação $PC-1$ é realizada de acordo com a tabela 8, que apresenta tanto a supressão dos 8 bits quanto a permutação dos 56 bits restantes.

Tabela 8 – Tabela *PC-1*

PC-1

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

O resultado da operação *PC-1* é uma chave de 56 bits, que precisa ser transformada em 16 *round-keys* K_i, cada uma delas com 48 bits. Para se obter as K_i, inicialmente dividem-se os 56 bits da chave em 2 grupos C_0 e D_0, de 28 bits cada um, sendo C_0 composto pelos primeiros 28 bits da chave e D_0 pelos últimos 28 bits. O próximo passo é recursivo, e será realizado 16 vezes, para a obtenção das 16 *round-keys*. Para se obter C_i e D_i a partir de C_{i-1} e D_{i-1}, com $i = 1, 2, \ldots, 16$, devem-se realizar transposições dos bits de C_{i-1} e D_{i-1} de 1 ou 2 posições para a esquerda, dependendo do valor de i. Por exemplo, caso o 1^o bit seja deslocado de 1 posição à esquerda, ele ocupará a posição 28^o. Analogamente, caso seja deslocado de 2 posições à esquerda, ocupará a 27^o posição, e assim sucessivamente. O número de posições que devem ser utilizadas para se deslocarem os bits tanto de C_{i-1} quanto de D_{i-1}, para se obter C_i e D_i, dependendo de i, podem ser visualizados na tabela 9.

Tabela 9 – Número de Deslocamentos à Esquerda para se obter C_i e D_i

i	Deslocamentos à Esquerda
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

Após a realização das transposições e obtenção de C_i e D_i, esses 2 blocos de 28 bits são concatenados como C_iD_i, obtendo-se assim um bloco de 56 bits. Esse bloco passará por um outro processo, que consiste em nova supressão de 8 bits e em uma permutação dos bits restantes, denominado *permutation choice 2 - PC-2*. Após a realização da operação *PC-2*, obtém-se um bloco de 48 bits, que será a *round-key* K_i da rodada *i*. A operação *PC-2* é realizada de acordo com a tabela 10, que apresenta tanto a supressão dos 8 bits quanto a permutação dos 48 bits restantes.

Tabela 10 – Tabela *PC-2*

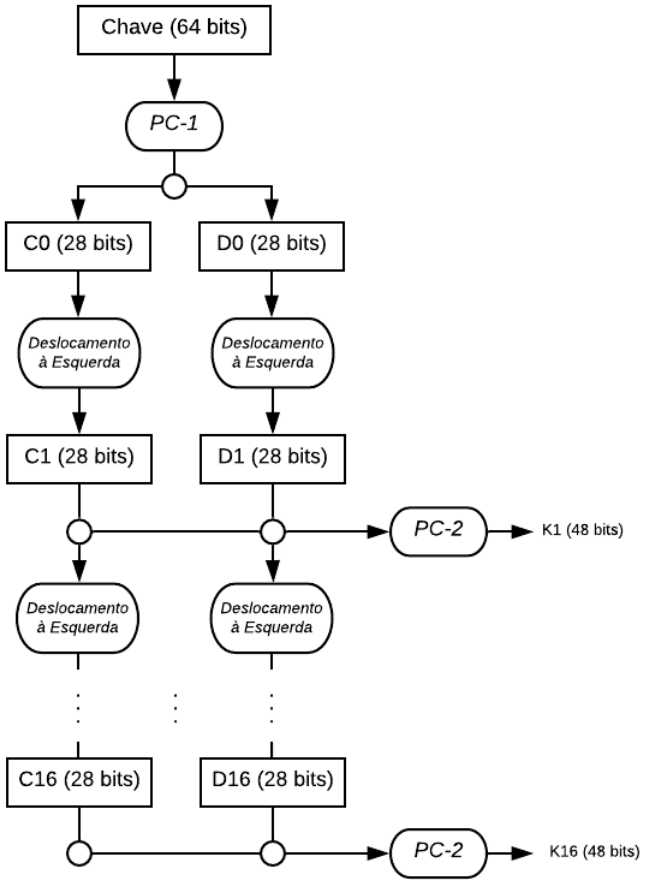
PC-2

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Procedendo-se dessa maneira para cada um dos 16 valores de *i*, obtêm-se as 16 *round-keys* utilizadas nas 16 etapas de cifração do DES. A figura 11 apresenta um

resumo geral do processo de obtenção das 16 *round-keys*.

Figura 11 – Visão Geral da Obtenção das *round-keys*



2.1.2 Decifração com o DES

A decifração de mensagens cifradas com a cifra DES não oferece grandes dificuldades para o receptor que conheça a chave secreta. Primeiramente, ele realiza o processo de se obter as mesmas 16 *round-keys*, de maneira análoga ao demonstrado na seção anterior. Em seguida, o receptor realiza o mesmo procedimento descrito no processo de cifração da seção anterior, com exceção das permutações IP e IP^{-1}, realiza todas as demais operações de maneira contrária, sem necessitar de mudanças na função-cifra. Inicialmente, o receptor realiza a permutação IP no bloco cifrado de 64 bits. Como a permutação IP é exatamente o inverso da permutação IP^{-1}, ele obtém o *pre-output* $R_{16}L_{16}$. Analogamente ao processo de cifração, ele faz $R_{15} = L_{16}$ e $L_{15} = R_{16} \oplus f(L_{16}, K_{16})$, obtendo-se assim R_{15} e L_{15}. De maneira geral, para obter R_{i-1} e L_{i-1} a partir de R_i e L_i, o

receptor realiza as seguintes operações:

$$R_{i-1} = L_i$$
$$L_{i-1} = R_i \oplus f(L_i, K_i)$$

Procedendo-se dessa maneira ao longo das 16 rodadas, obtém-se ao final o bloco de 64 bits L_0R_0. Para se recuperar a mensagem original, basta realizar a operação IP^{-1}, que consiste no inverso da operação IP, obtendo-se assim o bloco de 64 bits correspondente à mensagem clara, cifrada pelo emissor.

2.1.3 Comentários sobre o DES

Quando foi anunciado e divulgado o algoritmo para cifração utilizando o DES, inicialmente isso gerou uma certa suspeita na comunidade criptográfica ao redor do mundo, pois não se tinha certeza se, de fato, o DES não poderia ser quebrado por meio de uma *backdoor*, algo colocado na cifra intencionalmente para permitir a sua vulnerabilidade frente a um ataque de alguma entidade que conhecesse a *backdoor*. Essa insegurança surgiu pois, como as S-Boxes utilizadas na cifra são cruciais para promover confusão e trazer um componente não linear no processo de cifração, e a maneira como essas S-Boxes foram projetadas não foi divulgado pelo NIST na época, uma parcela da comunidade criptográfica suspeitou que esse segredo sobre o projeto das S-Boxes pudesse estar associado à inserção de uma *backdoor*, para que o governo dos EUA pudessem quebrar a cifra caso necessitassem (PAAR C.; PELZL, 2010). Entretanto, essa *backdoor* jamais foi encontrada. Inclusive, já nos anos 90, após a IBM ter finalmente revelado quais critérios usou no desenvolvimento das S-Boxes do DES, observou-se que esses critérios faziam com que a cifra fosse resistente a um tipo de ataque denominado criptoanálise diferencial, que não era conhecido pela comunidade criptográfica nos anos 70, na época em que o DES foi lançado. A própria IBM confirmou que já conhecia a criptoanálise diferencial em 1977, quando desenvolveu o DES, pelo menos 16 anos antes dessa técnica passar a ser conhecida e divulgada na comunidade criptográfica mundial.

Ainda sobre as S-Boxes, elas constituem um elemento fundamental de segurança da cifra DES, pois introduzem não linearidade ao processo de cifragem, o que torna a cifra resistente a um outro tipo de ataque, conhecido como ataque do texto-claro ou *known-plaintext attack*. Esse tipo de ataque necessita que o criptoanalista conheça ao menos um exemplo de texto claro e seu respectivo texto cifrado. Dessa forma, caso a cifra realize apenas processos lineares no texto claro para a obtenção do texto cifrado, como o criptoanalista conhece ambos, pode montar sistemas lineares que os relacionem por meio de dependência linear com a chave secreta. Ao resolver os sistemas, é obtida então a chave secreta utilizada na cifra. Como as S-Boxes atuam na mensagem de maneira não linear, elas protegem a cifra DES contra esse tipo de ataque. Para uma descrição

mais detalhada das técnicas de criptoanálise tanto do DES como de várias outras cifras, pode ser consultado (MENEZES; OORSCHOT; VANSTONE, 2001). Entretanto, apesar de o DES possuir resistência contra ataques criptoanalíticos mais sofisticados, ele possui uma fraqueza que vem sendo explorada pelos criptoanalistas ao longo de toda a história da criptografia: ele pode ser quebrado pelo método da força bruta. Isso ocorre devido ao tamanho reduzido da sua chave, 56 bits. Caso um criptoanalista deseje atacar o DES por força bruta, deve testar todas as chaves possíveis, isto é, 2^{56} possibilidades. Talvez não seja possível para um computador pessoal realizar todas essas tentativas de chave em um tempo razoável, mas os governos interessados em quebrar cifras inimigas possuem supercomputadores muito potentes, dedicados inteiramente a essa tarefa. Além disso, no final dos anos 90, foram desenvolvidas máquinas com *hardware* projetado especificamente para a tentativa exaustiva de chaves do DES, de forma que mesmo 2^{56} tentativas já poderiam ser realizadas em uma questão de dias, ou até mesmo de horas (PAAR C.; PELZL, 2010).

2.2 Advanced Encryption Standard - AES

Em janeiro de 1997, o NIST iniciou o processo para a escolha da cifra sucessora do DES. A cifra escolhida nesse processo passaria a ser o padrão mundial de criptografia simétrica, conhecido como *Advanced Encryption Standard - AES*. Dentre as várias cifras candidatas, por exemplo MARS, RC6, Serpent e Twofish, a vencedora do concurso foi a cifra Rijndael, desenvolvida pelos pesquisadores belgas Vincent Rijmen e Joan Daemen, motivo do nome Rijndael, uma junção dos nomes de seus desenvolvedores. Assim, em outubro de 2000, o Rijndael foi escolhido para ser o AES, passando a ser adotado como padrão criptográfico mundial a partir de novembro de 2001, e sua publicação em dezembro de 2001, na FIPS 197, tornou público o algoritmo. Essa é a cifra utilizada como padrão em criptografia simétrica na atualidade, sendo utilizada nos protocolos de segurança de internet IPsec, TLS, SSH, WPA, WPA2, dentre várias outras aplicações. Inclusive, o AES é recomendado pela Agencia Nacional de Segurança dos EUA - NSA, para a criptografia de documentos classificados como *SECRET* e *TOP SECRET* do governo dos EUA. Essa foi a primeira ocasião na história em que foi divulgada a cifra utilizada pelo governo dos EUA para criptografar documentos classificados (PAAR C.; PELZL, 2010). O AES é uma cifra de criptografia simétrica, isto é, há uma chave secreta compartilhada pelo emissor e pelo receptor, utilizada para cifrar e decifrar as mensagens. O AES funciona da seguinte maneira: a entrada da cifra é um bloco inicial de comprimento 128 bits, e existem 3 tamanhos possíveis de chaves a serem adotados: podem-se ter chaves de 128 bits, 192 bits ou 256 bits. Caso a chave possua 128 bits, então a cifra é desenvolvida em 10 rodadas ou *rounds*, caso a chave possua 192 bits, são 11 rodadas e, caso a chave possua 256 bits, o AES realiza 12 rodadas no processo de

cifração. Como o funcionamento da cifra é semelhante para os três tamanhos possíveis de chave, neste trabalho será apresentada apenas a descrição do processo para chaves de 128 bits. A cifra AES, quando utiliza chaves de 128 bits, é denominada AES-128.

2.2.1 Cifração com o AES-128

As entradas da cifra são o bloco de mensagem que se deseja criptografar, com comprimento de 128 bits, e a chave secreta, neste caso com comprimento de 128 bits também. A saída da cifra é um bloco criptografado de comprimento 128 bits. A cifra se divide em 4 operações básicas, denominadas *SubBytes*, *ShiftRows*, *MixColumns* e *AddRoundKey*. A cifração é dividida em 10 rodadas, também chamadas de estados ou *rounds*, sendo necessária a aplicação das 4 operações básicas em cada rodada. Somente na 10ª rodada não será aplicada a operação *MixColumns*. Antes do início da 1ª rodada, aplica-se a operação *AddRoundKey* na sequência de entrada de 128 bits, em uma rodada extra denominada "rodada zero", de forma que a 1ª rodada é iniciada após a conclusão dessa operação. Em cada uma das 11 rodadas, incluída a rodada zero, é utilizada uma chave na operação *AddRoundKey*, que será denominada K_i, com $i = 0, 1, 2, \ldots, 10$, sendo K_0 a chave secreta da cifra. As chaves K_i são obtidas a partir de K_0, por meio de um processo recursivo denominado *KeyExpansion*, que será descrito mais à frente.

Inicialmente, divide-se o bloco de 128 bits de entrada em 16 blocos de 8 bits cada um, sendo cada bloco de 8 bits denominado 1 *byte*. Essa divisão deve ser realizada na ordem em que os bits estão no bloco original, isto é, o 1º byte consiste na sequência do 1º ao 8º bit, o 2º byte consiste na sequência do 9º ao 16º bit, e assim sucessivamente, até o 16º byte, que consiste na sequência do 121º ao 128º bit. Esses 16 bytes devem ser agrupados em uma matriz 4 x 4, da seguinte forma: os primeiros 4 bytes são colocados na 1ª coluna da matriz, os próximos 4 bytes na 2ª coluna da matriz, e assim sucessivamente, até completar a matriz com os 16 bytes. Cada byte em sua posição na matriz será representado por S_{jk}, com $j = 0, 1, 2, 3$ e $k = 0, 1, 2, 3$, o primeiro índice indicando a linha da matriz e o segundo índice indicando a coluna, conforme a figura 12.

Figura 12 – Matriz de Bytes - AES

Essa maneira de montar uma matriz 4 x 4 de 16 bytes a partir de uma sequência de 128 bits será utilizada também para montar as matrizes das chaves K_i, que serão utilizadas em cada rodada da cifra. Cada matriz 4 x 4 formada por bytes das chaves K_i das rodadas será representada, de forma análoga à representação dos bytes do bloco, por K_{jk}, com $j = 0, 1, 2, 3$ e $k = 0, 1, 2, 3$, o primeiro índice indicando a linha da matriz e o segundo índice indicando a coluna.

Como forma de representar um byte de maneira mais sintética, é bastante comum representá-lo na base hexadecimal pois, em vez de representá-lo com 8 dígitos na base binária, na base hexadecimal bastam 2 dígitos para representá-lo. Essa representação dos bytes em base hexadecimal será utilizada nos próximos passos da descrição do AES. Além disso, para se realizarem algumas operações da cifra, será necessário estabelecer uma correspondência entre cada byte e um polinômio de grau 7, em que cada um dos bits do byte representa um coeficiente binário do polinômio. Por exemplo, seja $S_{jk} = a_7a_6a_5a_4a_3a_2a_1a_0$ um byte, sendo a_r cada um de seus bits, com $r = 0, 1, 2, \ldots, 7$. Esse byte pode ser tratado como o polinômio $a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0 = \sum_{r=0}^{7} a_rx^r$. Utilizando-se determinadas operações de soma e multiplicação, esses polinômios formam um corpo finito com 2^8 elementos, conhecido como $GF(2^8)$. As operações de soma e multiplicação desse corpo serão aplicadas aos bytes na cifra, e são definidas da seguinte forma: A soma de 2 elementos em $GF(2^8)$, isto é, a soma de 2 polinômios, é realizada de maneira análoga à operação XOR definida anteriormente, somando-se os coeficientes correspondentes de cada potência de x, e o resultado de cada soma é reduzido *modulo* 2. Essa operação entre bytes será representada pelo mesmo símbolo utilizado na operação XOR, $\oplus$. Por exemplo, para se realizar a soma dos bytes $a_7a_6a_5a_4a_3a_2a_1a_0$ e $b_7b_6b_5b_4b_3b_2b_1b_0$ em $GF(2^8)$, procede-se da seguinte forma:

$$a_7a_6a_5a_4a_3a_2a_1a_0 \oplus b_7b_6b_5b_4b_3b_2b_1b_0 = c_7c_6c_5c_4c_3c_2c_1c_0$$
$$\text{sendo cada } c_i = a_i \oplus b_1, \text{ com } i = 0, 1, 2, \ldots, 7$$

A multiplicação de 2 elementos em $GF(2^8)$, isto é, o produto de 2 polinômios, é realizada

efetuando-se o produto usual de polinômios, e reduzindo-se o resultado *modulo* um polinômio irredutível de 8º grau de $\mathbb{Z}_2[x]$. No caso da cifra AES, o polinômio escolhido foi $x^8 + x^4 + x^3 + x + 1$. Essa operação entre bytes será representada pelo símbolo $\odot$. Por exemplo, para se realizar o produto dos bytes $a_7a_6a_5a_4a_3a_2a_1a_0$ e $b_7b_6b_5b_4b_3b_2b_1b_0$ em $GF(2^8)$, com o polinômio irredutível $x^8 + x^4 + x^3 + x + 1$, procede-se da seguinte forma:

$$a_7a_6a_5a_4a_3a_2a_1a_0 \odot b_7b_6b_5b_4b_3b_2b_1b_0 = c_7c_6c_5c_4c_3c_2c_1c_0 \text{ , sendo:}$$

$$a(x) = a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$$

$$b(x) = b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0$$

$$c(x) = c_7x^7 + c_6x^6 + c_5x^5 + c_4x^4 + c_3x^3 + c_2x^2 + c_1x + c_0$$

$$c(x) \equiv a(x) \cdot b(x)\ mod(x^8 + x^4 + x^3 + x + 1)$$

A necessidade de se realizar essa correspondência entre bytes e polinômios em $GF(2^8)$ será melhor explicada nos próximos passos da cifra.

Após a separação dos 16 bytes de entrada em formato de matriz 4 x 4, a primeira operação realizada é a *AddRoundKey*. Essa operação é realizada em todas as rodadas, e é a única operação realizada na rodada zero. Portanto, será apresentada uma descrição geral da operação *AddRoundKey*, que vale para qualquer rodada da cifra. A operação consiste em, para uma determinada rodada i, com $i = 0, 1, 2, \ldots, 10$, realizar-se a soma entre cada byte do bloco e o byte correspondente da chave da rodada, que ocupem as mesmas posições da matriz 4 x 4. Basicamente, se S_{jk} é um byte do bloco da rodada i, e K_{jk} é um byte da chave da rodada i, o byte do bloco S_{jk} da nova rodada $i+1$ será igual a soma $S_{jk} \oplus K_{jk}$ da rodada i. Quando $i = 0$, a matriz de bytes S_{jk}, com $j = 0, 1, 2, 3$ e $k = 0, 1, 2, 3$ corresponde ao bloco de mensagem inicial da cifra, e a matriz de bytes K_{jk}, com $j = 0, 1, 2, 3$ e $k = 0, 1, 2, 3$ corresponde à chave secreta inicial.

Após a realização da operação *AddRoundKey* na rodada zero, todas as rodadas subsequentes se iniciam com a operação *SubBytes*. Essa operação consiste em um processo de substituição de bytes da matriz 4 x 4, em que cada byte da matriz da rodada é substituído por outro byte de uma tabela previamente construída, conhecida como S-Box da cifra AES. Ao contrário das S-Boxes da cifra DES, na cifra AES os parâmetros de construção da S-Box foram divulgados juntamente com a cifra, o que proporciona maior transparência ao processo. A maneira como a S-Box da cifra AES foi construída será apresentada em detalhes mais à frente. Todos os bytes da S-Box são dados em base hexadecimal. A troca de bytes ocorre da seguinte maneira: Seja $S_{jk} = a_7a_6a_5a_4a_3a_2a_1a_0$ um byte da matriz de bytes da rodada. Escreve-se S_{jk} na base hexadecimal, obtendo-se o número de 2 dígitos IJ. Então, utilizam-se a linha I e a coluna J da S-Box para definir

qual será o novo byte que substituirá S_{jk}. A tabela 11 contém a S-Box da cifra AES.

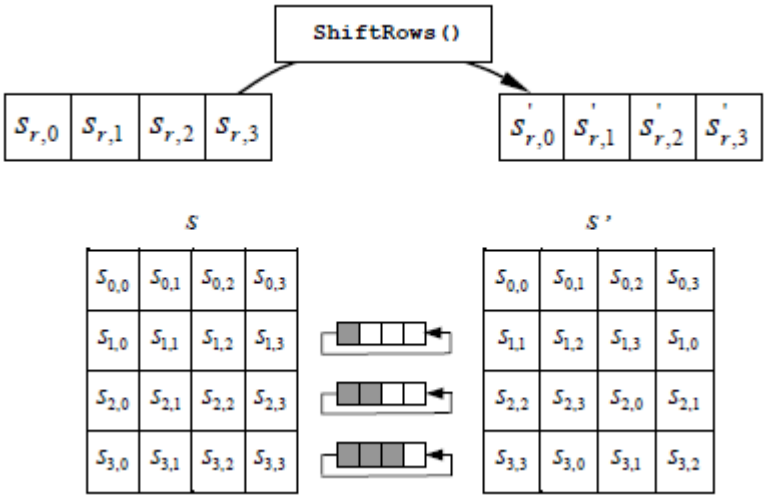
Tabela 11 – S-Box da Cifra AES

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

Por exemplo, o byte 01010011, que corresponde ao número 53 em base hexadecimal, deve ser substituído pelo byte que está na linha 5 e na coluna 3 da S-Box, portanto, pelo byte "ED", que em binário corresponde à 11101101.

Após a realização da operação *SubBytes* no início de cada uma das rodadas (com exceção da rodada zero), a cifra realiza a operação *ShiftRows*, que consiste em deslocar algumas posições para a esquerda determinadas linhas da matriz 4 x 4 obtida após a operação *SubBytes* da rodada. A 1ª linha da matriz é mantida inalterada, a 2ª linha é deslocada 1 posição para a esquerda, a 3ª linha é deslocada 2 posições para a esquerda e a 4ª linha é deslocada 3 posições para a esquerda. Esse processo de deslocamento pode ser melhor visualizado na figura 13.

Figura 13 – Operação *ShiftRows*



Após a realização da operação *ShiftRows*, realiza-se a operação *MixColumns*, que consiste em realizar uma alteração em cada coluna da matriz 4 x 4 dos bytes obtida após a operação *ShiftRows* da respectiva rodada. Nesta etapa, a alteração de cada coluna da matriz não influencia as demais colunas, assim como na operação *ShiftRows* a alteração de cada linha da matriz não influencia as demais linhas. Porém, a mudança de um byte em uma coluna influencia o resultado em todos os bytes daquela coluna. Os bytes da matriz da rodada são tratados como polinômios sobre o $GF(2^8)$, conforme descrito anteriormente. Nessa etapa, será necessário efetuar tanto somas quanto multiplicações de elementos neste corpo finito, utilizando-se as operações sobre $GF(2^8)$ definidas anteriormente. Pode-se representar a operação *MixColumns* da seguinte maneira: Para uma coluna c fixada da matriz 4 x 4, consideremos seus elementos S_{jc}, com $j = 0, 1, 2, 3$. Denotemos por U_{jc}, com $j = 0, 1, 2, 3$ os elementos da nova coluna c obtida, após a operação de *MixColumns* nessa coluna. Assim, os novos elementos U_{jc} de c são obtidos da seguinte maneira:

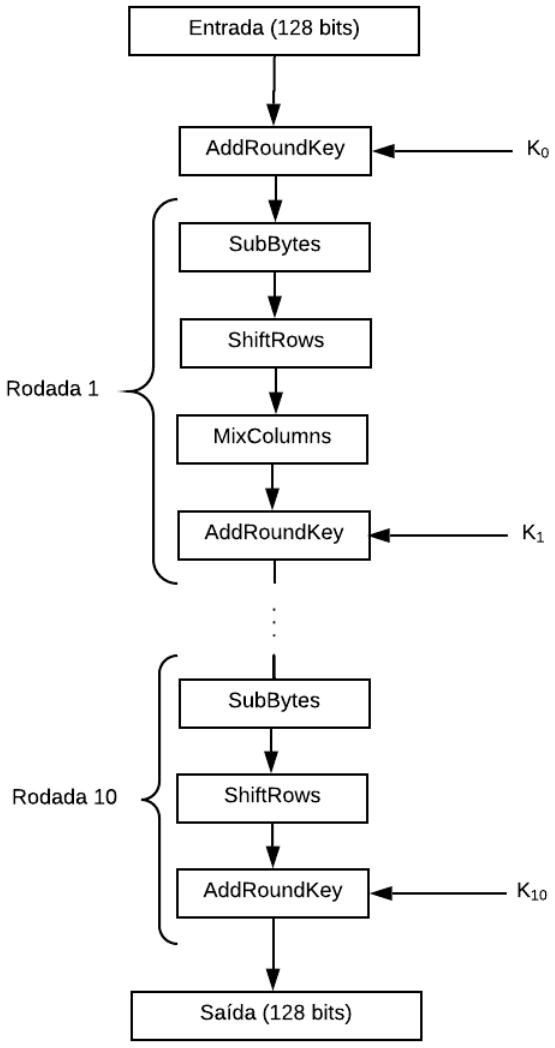
$$
\begin{aligned}
U_{0c} &= (00000010 \odot S_{0c}) \oplus (00000011 \odot S_{1c}) \oplus S_{2c} \oplus S_{3c} \\
U_{1c} &= S_{0c} \oplus (00000010 \odot S_{1c}) \oplus (00000011 \odot S_{2c}) \oplus S_{3c} \\
U_{2c} &= S_{0c} \oplus S_{1c} \oplus (00000010 \odot S_{2c}) \oplus (00000011 \odot S_{3c}) \\
U_{3c} &= (00000011 \odot S_{0c}) \oplus S_{1c} \oplus S_{2c} \oplus (00000010 \odot S_{3c})
\end{aligned}
$$

Portanto, aplicando-se o procedimento acima em cada uma das colunas c, $c = 0, 1, 2, 3$, obtém-se a nova matriz de bytes após a operação *MixColumns*. Essa operação será realizada em todas as rodadas da cifra, com exceção da última rodada.

Por fim, após a operação *MixColumns*, realiza-se a operação *AddRoundKey* em cada uma das rodadas da cifra, finalizando-se assim o processo de cifração do AES. Uma representação geral dos processos necessários para o funcionamento da cifra

encontra-se na figura 14.

Figura 14 – Visão Geral da Cifra AES



Para finalizar a descrição da cifra, ainda resta descrever como é construída a S-Box da cifra, e como são obtidas as chaves K_i, $i = 1, 2, \ldots, 10$ a partir de K_0. Primeiramente, será apresentado o processo de construção da S-Box. Seja $S_{jk} = a_7a_6a_5a_4a_3a_2a_1a_0$ um byte qualquer de entrada na operação *SubBytes*. Para construir a S-Box, basta verificar como obter o byte de saída para cada um dos 256 possíveis bytes de entrada S_{jk}, e colocá-los da forma de tabela. Vejamos como cada um desses bytes de saída é obtido. Conforme descrito anteriormente, cada um dos bytes $S_{jk} = a_7a_6a_5a_4a_3a_2a_1a_0$ pode ser tratado como um polinômio $a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$ de $\mathbb{Z}_2[x]$. Como o AES utiliza um polinômio irredutível de 8º grau na definição da sua multiplicação de bytes, tratados como polinômios, o anel $\mathbb{Z}_2[x]/(x^8+x^4+x^3+x+1)$ na verdade é um corpo, que possui $2^8 = 256$ elementos, e cada um desses elementos, com exceção do polinômio

nulo, possui um elemento inverso, que será denotado por $S_{jk}^{-1} = b_7b_6b_5b_4b_3b_2b_1b_0$, tal que $S_{jk} \odot S_{jk}^{-1} \equiv 1 \ mod(x^8 + x^4 + x^3 + x + 1)$. Caso $S_{jk} = 00000000$, faz-se $S_{jk}^{-1} = 00000000$ por definição, para que todos os 256 elementos de $\mathbb{Z}_2[x]/(x^8 + x^4 + x^3 + x + 1)$ possuam elemento inverso. Portanto, a partir de um byte $S_{jk} = a_7a_6a_5a_4a_3a_2a_1a_0$ qualquer, pode-se obter seu byte inverso $S_{jk}^{-1} = b_7b_6b_5b_4b_3b_2b_1b_0$. Por fim, de posse de S_{jk}^{-1}, aplica-se uma composição de 2 operações em cada um de seus bits, obtendo-se assim o byte de saída da S-Box $D_{jk} = b'_7b'_6b'_5b'_4b'_3b'_2b'_1b'_0$. A composição das 2 operações a serem realizadas nos bits do byte S_{jk}^{-1} para se obter os bits do byte de saída D_{jk} estão representadas na figura 15.

Figura 15 – Composição de Operações entre Bits para Construção da S-Box do AES

$$\begin{pmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} \ mod(2)$$

Portanto, realizando-se o procedimento descrito acima com todos os possíveis 256 bytes S_{jk} de entrada da S-Box, obtêm-se cada um dos 256 possíveis bytes D_{jk} de saída da S-Box, permitindo assim a construção completa da S-Box.

O processo de obtenção das chaves K_i, $i = 1, 2, \ldots, 10$ a partir de K_0, denominado *KeyExpansion*, será descrito a seguir. Novamente, é necessário fazer algumas considerações a respeito da notação utilizada. Um conjunto de 4 bytes, isto é, 32 bits, é denominado *palavra* e essa nomenclatura será utilizada na descrição da *KeyExpansion*. Quando a cifra AES utiliza chaves de 128 bits, ela trabalha com 10 rodadas, sendo então necessário que se obtenham 10 chaves diferentes a partir da chave inicial K_0, por meio do processo *KeyExpansion*. Como cada chave possui 128 bits, isto é, 4 *palavras*, ao todo serão necessárias 40 *palavras* para se gerar as 10 chaves, além das 4 *palavras* iniciais de K_0. Portanto, ao final do processo, são necessárias 44 *palavras* no total, que serão designadas por $w[0], w[1], w[2], \ldots, w[43]$, sendo $K_0 = w[0]w[1]w[2]w[3]$, $K_1 = w[4]w[5]w[6]w[7]$ e assim sucessivamente. Define-se também uma matriz M de dimensões 1 x 10, em que cada entrada m_{1i} da matriz, com $i = 1, 2, \ldots, 10$, representa uma *palavra* constante. Os valores de cada m_{1i}, com $i = 1, 2, \ldots, 10$, estão representados a seguir, em base

hexadecimal:

$$\begin{aligned} m_1 &= 0x01000000 & m_6 &= 0x20000000 \\ m_2 &= 0x02000000 & m_7 &= 0x40000000 \\ m_3 &= 0x04000000 & m_8 &= 0x80000000 \\ m_4 &= 0x08000000 & m_9 &= 0x1B000000 \\ m_5 &= 0x10000000 & m_{10} &= 0x36000000 \end{aligned}$$

Por último, define-se uma variável temporária *temp*, necessária apenas para armazenar *palavras* intermediárias ao longo do processo de *ExpansionKey*. Finalmente, segue a descrição do processo: A partir da chave secreta inicial $K_0 = w[0]w[1]w[2]w[3]$, serão realizadas operações recursivas com essas 4 *palavras*, obtendo-se assim novas *palavras*, e assim sucessivamente, até que haja 44 *palavras* no total. O algoritmo utilizado para expandir essas 4 *palavras* iniciais opera de 2 maneiras distintas, dependendo da posição da *palavra* na sequência de geração. Assim, dada uma *palavra* $w[i]$, se i for múltiplo de 4, realizam-se as seguintes operações:

1. Atribui-se à variável *temp* o valor de $w[i-1]$;
2. Realiza-se um deslocamento de uma posição para a esquerda nos bytes da variável *temp*, isto é, se $temp = B_0B_1B_2B_3$, então após o deslocamento ter-se-á $temp = B_1B_2B_3B_0$;
3. Após esse deslocamento, aplica-se o processo *SubBytes* em cada byte da variável *temp*. Nesta etapa, utiliza-se a mesma S-Box da cifra AES;
4. De posse da variável *temp* permutada e substituída, faz-se $w[i] = w[i-4] \oplus m_{\frac{i}{4}} \oplus temp$.

Caso i não seja múltiplo de 4, procede-se da seguinte forma:

1. Atribui-se à variável *temp* o valor de $w[i-1]$;
2. Faz-se $w[i] = w[i-4] \oplus temp$.

Aplicando-se esse algoritmo para $i = 4, 5, 6, ..., 43$, obtém-se a sequência de 40 *palavras* $w[4], w[5], w[6], \ldots, w[43]$. Finalmente, faz-se:

$$\begin{aligned} K_1 &= w[4]w[5]w[6]w[7] \\ K_2 &= w[8]w[9]w[10]w[11] \\ &\vdots \end{aligned}$$

Dessa forma, obtêm-se as 10 chaves K_i, com $i = 1, 2, \ldots, 10$, necessárias para a cifração utilizando o AES-128.

2.2.2 Decifração com o AES-128

Ao contrário do que ocorre na decifração da cifra DES, em que não há qualquer mudança na descrição das operações necessárias para cifração e decifração, na cifra AES, para se realizar a decifração, são necessárias algumas mudanças nas operações realizadas nas rodadas de cifração. Das 4 operações realizadas ao longo das rodadas no AES, apenas a operação *AddRoundKey* é a mesma tanto para a cifração quanto para a decifração, pois trata-se apenas de uma operação XOR entre bytes, e essa operação é idêntica à sua operação inversa, isto é, $(S \oplus K) \oplus K = S$, sendo S e K bytes. As demais operações, *SubBytes*, *ShiftRows* e *MixColumns*, precisam ser adaptadas para a decifração, e são designadas por *InvSubBytes*, *InvShiftRows* e *InvMixColumns*, respectivamente. Cada uma dessas novas operações descreve passos que devem ser aplicados aos bytes cifrados, em cada uma das rodadas, para que a decifração recupere a mensagem original. Este trabalho não descreverá as operações modificadas para a decifração do AES. O leitor interessado em uma descrição detalhada das operações *InvSubBytes*, *InvShiftRows* e *InvMixColumns* pode consultar (NIST, 2001). Também, é necessário realizar-se o mesmo processo para se obter as 10 chaves utilizadas nas rodadas do AES, a partir da chave secreta K_0. Porém, a utilização das chaves ocorre de maneira invertida em relação à cifração, isto é, a primeira chave utilizada na operação *AddRoundKey* da decifração é K_{10}, seguida por K_9, e assim sucessivamente, até que a última operação realizada na decifração é *AddRoundKey* utilizando-se K_0, recuperando-se assim a mensagem clara.

2.2.3 Comentários sobre o AES

O AES vem se mostrado por quase 20 anos, desde o seu lançamento como padrão de criptografia simétrica nos EUA, como uma cifra resistente aos métodos de criptoanálise modernos, e o principal tipo de ataque contra a cifra continua sendo a força bruta. Porém, como as chaves do AES são bem maiores que a chave do DES, a tentativa à exaustão de todas as possibilidades de chaves, como foi feito no ataque ao DES, não se mostra viável, nem mesmo utilizando-se *hardwares* específicos para tal tarefa. Devido à segurança proporcionada pelo AES, essa cifra é utilizada modernamente por diversos protocolos de transferência segura de arquivos na internet, como FTPS, HTTPS, grande parte das VPNs disponíveis no mercado, também protocolos de segurança de redes Wi-Fi, como WPA e WPA2. O AES é a cifra utilizada, inclusive, pelo aplicativo de troca de mensagens mais popular do Brasil, o WhatsApp. Todas as mensagens trocadas pelos usuários do WhatsApp são cifradas utilizando-se o AES com uma chave de 256 bits, trocada entre os usuários por meio de um algoritmo de criptografia de chave pública para a troca de chaves, denominado *Elliptic Curve Diffie Helman - ECDH*. Essa troca segura de chaves promove o que se conhece como criptografia de ponta a ponta, o que garante que nem mesmo o próprio WhatsApp conheça o conteúdo das mensagens trocadas,

pois a chave secreta do AES é compartilhada entre emissor e receptor utilizando-se um protocolo de compartilhamento de chaves por criptografia assimétrica. O processo de troca de mensagens criptografadas pelo WhatsApp possui vários detalhes técnicos adicionais, que podem ser consultados em (WHATSAPP, 2017).

3 Criptografia de Chave Pública ou Assimétrica

Neste capítulo, serão apresentados o algoritmo de criptografia assimétrica RSA e algoritmos de criptografia baseados em propriedades de curvas elípticas, conhecidos como *Elliptic Curve Cryptography - ECC*.

3.1 Algoritmo de Criptografia de Chave Pública RSA

No ano de 1976, Diffie e Hellman introduziram no universo criptográfico existente àquela época o conceito de criptografia de chave pública, também chamada de criptografia assimétrica. As ideias inovadoras divulgadas pela dupla se espalharam por todo o mundo, e, em 1977, três pesquisadores do *Massachusetts Institute of Technology - MIT* desenvolveram o algoritmo denominado RSA, cuja metodologia de funcionamento também se baseava no conceito de criptografia assimétrica. Esse algoritmo tornou-se mundialmente conhecido rapidamente, e começou a ser utilizado como padrão mundial para troca de informações, inclusive chaves, por meio de canais inseguros e em protocolos de assinatura digital, sendo amplamente utilizado principalmente em transações pela internet. Ainda hoje, grande parte da troca de informações sigilosas via internet se dá por meio da utilização de protocolos que se baseiam no algoritmo RSA. A sigla RSA é derivada dos nomes dos autores do algoritmo: R. L. Rivest, A. Shamir e L. Adleman. A força do algoritmo RSA reside na dificuldade computacional para se fatorar números compostos muito grandes, embora a multiplicação de números primos grandes seja relativamente simples do ponto de vista computacional. Até hoje, não se conhecem algoritmos de computação clássica eficientes o suficiente para que se possam fatorar esses números compostos grandes em um tempo reduzido, o que torna a quebra do RSA bastante improvável atualmente, desde que se utilizem chaves de tamanho adequado. No entanto, já se conhecem algoritmos de computação quântica que possibilitam a fatoração de números inteiros grandes em tempo polinomial, de forma que a segurança do RSA estará fortemente comprometida caso os computadores quânticos se tornem uma realidade.

3.1.1 Funcionamento do RSA

O RSA é um algoritmo de criptografia assimétrica, isto é, usa-se uma chave pública para cifrar a mensagem, e uma chave privada para decifrá-la. Portanto, para que haja troca de mensagens entre 2 entidades, é necessário que cada uma delas possua

um par de chaves, uma pública e outra privada. Primeiramente, é necessário entender o processo de geração desses pares de chaves pública e privada. Esse processo é realizado de maneira totalmente independente entre as 2 entidades. Cada uma delas escolhe, secretamente, dois números primos grandes p e q, e efetua a multiplicação $n = pq$. Em seguida, calcula-se o valor da função ϕ de Euler do número n, dada por $\phi(n) = (p-1)(q-1)$. Conhecendo-se o valor de $\phi(n)$, escolhe-se um número e, de forma que $1 < e < \phi(n)$, e também $MDC(e, \phi(n)) = 1$. A necessidade de e e $\phi(n)$ serem primos entre si está relacionada com o próximo passo, que consiste em encontrar o inverso multiplicativo de e *modulo* $\phi(n)$. Por meio do algoritmo de Euclides estendido, pode-se encontrar d, tal que $de \equiv 1 \ mod(\phi(n))$, isto é, $d \equiv e^{-1} \ mod(\phi(n))$. A chave pública de cada uma das entidades consiste no par de números (n, e), e a chave privada no par (n, d). Divulga-se a chave pública de maneira ostensiva em qualquer tipo de canal, seja ele seguro ou não, e utiliza-se a chave privada para decifrar as mensagens recebidas que foram cifradas utilizando a chave pública. Por exemplo, vamos analisar o caso em que Bob deseja enviar uma mensagem cifrada para Alice, utilizando-se o RSA. Primeiramente, Bob deve conhecer a chave pública de Alice, que consiste no par (n, e). Em seguida, é necessário converter o texto a ser cifrado em blocos numéricos $m_1, m_2, \ldots, m_i$, de forma que cada número não exceda o valor de n. A conversão do texto em blocos numéricos pode ser realizada, por exemplo, utilizando-se o padrão ASCII. Pode-se observar o padrão de conversão ASCII na tabela 12.

Tabela 12 – Tabela ASCII

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(	72	48	110	H	104	68	150	h
9	9	11		41	29	51	)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[	123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	|
29	1D	35		61	3D	75	=	93	5D	135	]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	

De posse de cada um dos blocos numéricos $m_1, m_2, ..., m_i$, Bob calcula $C_1 \equiv m_1^e \ mod(n)$, $C_2 \equiv m_2^e \ mod(n)$, ... ,$C_i \equiv m_i^e \ mod(n)$, e envia a sequência de blocos cifrados $C_1, C_2, ..., C_i$ para Alice. Para decifrar os blocos $C_1, C_2, ..., C_i$, Alice realiza a exponenciação *modulo n* de cada um dos blocos pelo componente d da sua chave privada, de forma que se obtém o seguinte:

$$C_1^d \equiv (m_1^e)^d \equiv m_1^{ed} \equiv m_1^{k\phi(n)+1} \equiv (m_1^{\phi(n)})^k m_1^1 \equiv (1)^k m_1 \equiv m_1 \ mod(n)$$

$$C_2^d \equiv (m_2^e)^d \equiv m_2^{ed} \equiv m_2^{k\phi(n)+1} \equiv (m_2^{\phi(n)})^k m_2^1 \equiv (1)^k m_2 \equiv m_2 \ mod(n)$$

$$\vdots$$

$$C_i^d \equiv (m_i^e)^d \equiv m_i^{ed} \equiv m_i^{k\phi(n)+1} \equiv (m_i^{\phi(n)})^k m_i^1 \equiv (1)^k m_i \equiv m_i \ mod(n)$$

Dessa forma, Alice obtém os blocos numéricos $m_1, m_2, ..., m_i$, que podem ser convertidos na mensagem de texto original utilizando-se o mesmo padrão ASCII. Para justificar a validade desse procedimento de decifração, o principal detalhe a ser observado é a utilização do teorema de Euler, que estabelece o seguinte:

$$a \text{ e } n \in \mathbb{Z}, \ n > 1 \text{ e } MDC(a, n) = 1 \implies a^{\phi(n)} \equiv 1 \ mod(n)$$

Portanto, para provar que, de fato, $(m_i^{\phi(n)})^k m_i \equiv (1)^k m_i \equiv m_i \ mod(n)$, é necessário analisar 2 casos distintos:

1. Caso $MDC(m_i, n) = 1$, então a prova decorre imediatamente do teorema de Euler.

2. Caso $MDC(m_i, n) \neq 1$, então m_i deve ser um múltiplo de p ou de q, pois $n = pq$. Sem perda de generalidade, vamos supor que m_i é um múltiplo de p, isto é, $m_i = rp$, para algum $r \in \mathbb{N}$ e $r < q$. Portanto, nesse caso, $MDC(m_i, q) = 1$, o que nos permite utilizar o teorema de Euler para concluir que:

   $$m_i^{\phi(q)} \equiv 1 \ mod(q)$$

   Portanto, podemos calcular $(m_i^{\phi(n)})^k \ mod(q)$ da seguinte forma:

   $$(m_i^{\phi(n)})^k \equiv (m_i^{(p-1)(q-1)})^k \equiv (m_i^{(q-1)})^{k(p-1)} \equiv (m_i^{\phi(q)})^{k(p-1)} \equiv (1)^{k(p-1)} \equiv 1 \ mod(q)$$

   Dessa forma, concluímos que $(m_i^{\phi(n)})^k = 1 + sq$, para algum $s \in \mathbb{N}$. Por fim, podemos aplicar esse resultado para realizar o cálculo de $(m_i^{\phi(n)})^k m_i \ mod(n)$ da seguinte forma:

   $$(m_i^{\phi(n)})^k m_i \equiv (1 + sq)m_i \equiv m_i + m_i sq \equiv m_i + (rp)sq \equiv m_i + nrs \equiv m_i \ mod(n)$$

   Portanto, fica provado que o processo de decifração descrito acima sempre permite a recuperação do bloco numérico m_i a partir de C_i, n e d.

Pode-se observar que a força do algoritmo reside no fato de que, caso o espião Eve intercepte a mensagem e queira decifrá-la, é necessário que ele conheça a chave privada (n, d) de Alice. Porém, a única informação disponível para Eve é a chave pública (n, e) de Alice, que foi divulgada por meio de um canal qualquer. Para se obter a chave privada (n, d) a partir da chave pública (n, e) é necessário calcular o valor de $\phi(n)$, conhecendo-se apenas o número composto n, sem que sejam conhecidos os seus fatores primos p e q. Portanto, como $\phi(n) = (p-1)(q-1)$, e não se conhece uma maneira melhor para carcular $\phi(n)$, seria necessário decompor n em seus fatores primos p e q. Porém, para fatores primos p e q suficientemente grandes, essa decomposição é computacionalmente inviável. Mesmo os melhores algoritmos de computação clássica para fatoração conhecidos na atualidade ainda se mostram ineficazes quando trabalham com inteiros suficientemente grandes, o que garante a segurança do RSA.

3.1.2 Exemplo Numérico de Aplicação do RSA

A seguir, será apresentado um exemplo numérico ilustrativo da utilização do algoritmo RSA. Todos os cálculos foram realizados utilizando-se o software *wxMaxima*. Supondo-se que Bob deseja enviar uma mensagem cifrada com o RSA para Alice, para isso ele precisa da chave pública de Alice. Portanto, primeiramente, é necessário que Alice execute o processo de geração do seu par de chaves pública e privada. Para isso, Alice escolhe dois números primos grandes p e q, e efetua a multiplicação $n = pq$. Como exemplo, serão tomados $p = 14386079$ e $q = 250368653$. Em seguida, Alice calcula:

$$\begin{aligned} n &= pq \\ &= (14386079)(250368653) \\ &= 3601823221181587 \end{aligned}$$

Vale lembrar que os números primos utilizados em aplicações práticas do algoritmo RSA são muito maiores que os números primos utilizados neste exemplo, que possui o intuito meramente ilustrativo do método. Nas implementações mais atuais do RSA, trabalha-se com chaves de 4096 bits, o que gera a necessidade de números primos muito maiores que os desse exemplo. O próximo passo para que Alice gere seu par de chaves pública e privada consiste em calcular o valor $\phi(n)$, obtendo-se:

$$\begin{aligned} \phi(n) &= (p-1)(q-1) \\ &= (14386078)(250368652) \\ &= 3601822956426856 \end{aligned}$$

Em seguida, Alice escolhe um número e, com $1 < e < \phi(n)$, e $MDC(e, \phi(n)) = 1$. Como exemplo, será tomado $e = 9166939$. Então, calcula-se d, o inverso multiplicativo de e $mod(n)$, obtendo-se $d = 1353380917365627$. Finalmente, Alice publica o par de números (n, e) como sua chave pública, e mantém secretamente o par de números (n, d) como

sua chave privada. Bob toma conhecimento da chave pública de Alice e deseja enviar a ela a mensagem $m = 64728264834628$ cifrada com o RSA. Bob então calcula:

$$\begin{aligned} C &\equiv m^e \ mod(n) \\ &\equiv (64728264834628)^{9166939} \ mod(n) \\ &\equiv 2062780619908712 \ mod(n) \end{aligned}$$

Essa é a mensagem cifrada. Em seguida, Bob envia C para Alice. Ao receber a mensagem cifrada, Alice utiliza sua chave privada para decifrá-la, realizando-se a seguinte operação:

$$\begin{aligned} m &\equiv C^d \ mod(n) \\ &\equiv (2062780619908712)^{1353380917365627} \ mod(n) \\ &\equiv 64728264834628 \ mod(n) \end{aligned}$$

Dessa forma, Alice obtém a mensagem original enviada por Bob, sem que houvesse a necessidade de um canal seguro para a comunicação entre emissor e receptor, exemplificando assim o funcionamento da criptografia assimétrica ou de chave pública.

3.1.3 Exemplo de Aplicação do RSA em Assinatura Digital

O algoritmo RSA, além de possuir larga aplicação na criptografia de mensagens, também pode ser utilizado no processo de assinatura digital. A seguir, será apresentado um exemplo numérico simplificado da utilização do algoritmo RSA em uma troca de mensagens com assinatura digital. Como se trata meramente de um exemplo ilustrativo, a assinatura será feita na própria mensagem, o que não ocorre na prática. Nas situações reais, a assinatura é feita no resumo ou *hash* da mensagem, para garantir a sua confidencialidade na transmissão por um canal inseguro. Analogamente ao exemplo anterior, vamos supor que Bob deseja enviar uma nova mensagem para Alice, mas agora Bob deseja assinar a mensagem utilizando o RSA. Inicialmente, é necessário que Bob execute o processo de geração de seu par de chaves pública e privada, de maneira análoga ao processo realizado por Alice. Esse processo não será descrito novamente, de forma que já serão apresentadas as chaves pública e privada de Bob:

chave pública de Bob = $(91610606752480252 6781, 7645321)$
chave privada de Bob = $(91610606752480252 6781, 497028085498237186681)$

Bob deseja enviar a mesma mensagem $m = 64728264834628$ cifrada e assinada para Alice. Conforme realizado no exemplo anterior, de posse da chave pública de Alice, Bob calcula $C = 20627806199087122 \ mod(3601823221181587)$. Essa é a mensagem cifrada, análoga àquela obtida no exemplo anterior. Em seguida, Bob deverá "assinar" a mensagem, utilizando para isso a sua própria chave privada, e enviar tanto a mensagem cifrada quanto a mensagem assinada para Alice, para que ela possa comparar as

duas e verificar a autenticidade do emissor da mensagem. Mais uma vez, vale ressaltar que, nas aplicações reais de assinatura digital, caso a mensagem a ser transmitida seja confidencial, não se assina a própria mensagem, mas sim seu *hash*, algo que não será feito nesse exemplo, devido ao seu caráter simplificado e meramente didático. Prosseguindo com o exemplo, Bob então calcula a mensagem "assinada" S da seguinte forma:

$$\begin{aligned} S &\equiv m^d \ mod(n) \\ &\equiv (64728264834628)^{49702808549823718 6681} \ mod(n) \\ &\equiv 171079776946897088586 \ mod(n) \end{aligned}$$

Essa é a mensagem assinada por Bob. Dessa forma, Bob envia a mensagem cifrada C e a mensagem assinada S para Alice. Ao receber as mensagens, Alice utiliza sua chave privada para decifrar a mensagem C, de maneira análoga ao processo realizado no exemplo anterior, obtendo-se assim $m = 64728264834628$. Além disso, Alice deseja verificar a autenticidade da mensagem. Ela, então, realiza a exponenciação de S por e, que faz parte da chave pública de Bob, obtendo-se:

$$\begin{aligned} m' &\equiv S^e \ mod(n) \\ &\equiv (171079776946897088586)^{7645321} \ mod(n) \\ &\equiv 64728264834628 \ mod(n) \end{aligned}$$

Como $m = m'$, Alice pode ter certeza de que Bob é o verdadeiro emissor da mensagem. Caso as mensagens cifrada e assinada apresentassem conteúdos diferentes após Alice decifrá-las, então se poderia afirmar que a entidade emissora das mensagens não se tratava de Bob, pois a chave pública de Bob não se mostrou adequada na decifração da mensagem assinada. Na prática, conforme já ressaltado, não se assina a própria mensagem, mas sim o seu *hash*. O algoritmo utilizado para a geração do *hash* deve ser conhecido tanto pelo emissor quanto pelo receptor da mensagem. Dessa forma, de maneira análoga ao exemplo numérico apresentado, o emissor envia dois blocos numéricos ao receptor: o *hash* assinado com a sua própria chave privada, e a mensagem original cifrada com a chave pública do receptor. Da mesma forma, o receptor utiliza a sua própria chave privada para decifrar a mensagem e, para comprovar a autenticidade do emissor, ele utiliza a chave pública do suposto emissor verdadeiro e decifra o *hash*. Em seguida, ele aplica a função geradora do *hash* à mensagem original decifrada, e compara o resultado com o *hash* assinado, que já havia sido decifrado. Se os dois *hashes* forem idênticos, está assegurada a autenticidade do emissor, pois somente o emissor autêntico poderia cifrar o *hash* utilizando a sua própria chave privada, de forma que a sua chave pública pudesse ser utilizada para decifrá-lo corretamente. Há vários motivos que justificam a utilização do *hash* no processo de assinatura digital, sendo o principal deles a proteção do caráter sigiloso da mensagem. Caso o emissor enviasse a própria mensagem assinada (como foi feito no exemplo numérico com Bob), qualquer entidade estranha, por exemplo o espião Eve, que pudesse interceptar a mensagem

assinada poderia facilmente descobrir o seu conteúdo, bastando para isso decifrá-la utilizando a chave pública do emissor, que é ostensivamente divulgada. Dessa forma, caso a assinatura digital fosse aplicada diretamente à mensagem, haveria um sério comprometimento do sigilo da mensagem, o que justifica a necessidade de se assinar o *hash* ao invés da própria mensagem. Para mais detalhes sobre funções geradoras de *hashes*, podem ser consultados (STINSON, 2006) e (PAAR C.; PELZL, 2010).

3.1.4 Comentários sobre o RSA

Ao se analisar com cuidado a metodologia de funcionamento do algoritmo RSA, surge uma questão muito importante: como obter o par de primos p e q grandes utilizados no RSA, se os algoritmos de fatoração conhecidos atualmente não conseguem determinar os fatores primos de números muito grandes? Essa aparente contradição não se verifica pelo seguinte fato: apesar de não se conhecerem algoritmos eficientes o suficiente para se fatorar números compostos grandes, são conhecidos algoritmos que determinam se um número é primo ou composto sem a necessidade de fatorá-lo. Há uma variedade de testes probabilísticos, com tempo de processamento polinomial, que podem ser aplicados em números grandes e verificar se são compostos ou provavelmente primos, sem que para isso seja necessário conhecer seus fatores. No ano de 2002, os pesquisadores indianos Manindra Agrawal, Neeraj Kayal e Nitin Saxena, do *Indian Institute of Technology Kanpur*, publicaram o artigo "*PRIME is in P*", no qual apresentaram, pela primeira vez na história, um algoritmo determinístico que determina, em tempo de processamento polinomial, se um número qualquer n é primo ou composto. Esse algoritmo ficou conhecido como *AKS Primality Test*, devido às iniciais dos sobrenomes de seus desenvolvedores. O funcionamento do AKS se baseia em uma generalização polinomial do pequeno teorema de Fermat, e seus autores receberam diversos prêmios pelo seu desenvolvimento, entre eles o Prêmio Gödel, no ano de 2006. Para maiores detalhes acerca do funcionamento do algoritmo AKS e de outros testes de primalidade, podem ser consultados (AGRAWAL; KAYAL; SAXENA, 2002), (SHOKRANIAN, 2008) e (COUTINHO, 2000).

Outro aspecto importante sobre o RSA diz respeito à sua segurança. Apesar da segurança da troca de mensagens utilizando o RSA se basear no caráter secreto da chave privada, o método para a obtenção da chave privada a partir da chave pública não é um segredo, mas sim um problema computacional difícil de ser resolvido em um tempo razoável, com a tecnologia da computação atual. Portanto, apesar da utilização do RSA ser considerada segura atualmente com chaves suficientemente grandes, o surgimento de novos algoritmos de fatoração mais eficientes pode colocar toda a segurança do RSA em risco. Há inúmeros pesquisadores em todo o mundo trabalhando no possível desenvolvimento de tais algoritmos. Outra possibilidade real de quebra do

RSA consiste no desenvolvimento dos chamados "computadores quânticos", que se baseiam na utilização de certas propriedades quânticas da matéria em seu funcionamento. Mesmo que a tecnologia dos chamados "processadores quânticos" ainda não seja aplicável comercialmente, já há uma enorme expectativa acerca dos impactos que seriam causados pelo desenvolvimento de um computador quântico. Por exemplo, em 1994, o pesquisador Peter Shor, do AT&T Bell Laboratories, desenvolveu um algoritmo que poderia ser implementado em um computador quântico, capaz de fatorar números inteiros enormes, e também resolver o problema do logaritmo discreto de maneira extremamente rápida. Peter Shor também foi ganhador do Prêmio Gödel, no ano de 1999, pela publicação desse seu artigo sobre fatoração de números inteiros em tempo polinomial em computadores quânticos. Dessa forma, caso o computador quântico seja realmente viabilizado, toda a utilização atual do algoritmo RSA se tornará insegura. Trata-se de um grande risco para a estabilidade dos serviços de troca de mensagens criptografadas na internet. Além disso, muitas infraestruturas de chaves públicas ao redor do mundo, inclusive a ICP Brasil, possuem processos de criptografia fortemente ancorados no RSA. A desestabilização do RSA causaria forte abalo nos principais gerenciadores de certificados digitais do mundo, o que traria grandes prejuízos para a sociedade. A segurança do RSA se mostra, portanto, um tema extremamente relevante no contexto global em que o algoritmo está inserido. Para mais detalhes sobre os algoritmos quânticos de fatoração de inteiros, podem ser consultados (SHOR, 1997) e (COUTINHO, 2000).

3.2 Algoritmo de Criptografia com Curvas Elípticas

O estudo das curvas elípticas e de suas propriedades teve início ainda no século XVIII, com trabalhos dos matemáticos Giulio Fagnano e Leonhard Euler sobre integrais elípticas. Ao contrário do que sugere seu nome, as curvas elípticas não são elipses. Elas receberam esse nome devido à sua relação com algumas integrais elípticas que surgem no cálculo do comprimento do arco de elipses. As primeiras aplicações de curvas elípticas em criptografia foram propostas no ano de 1985, de maneira independente, pelos pesquisadores Neal Koblitz e Victor S. Miller. A criptografia de curvas elípticas, conhecida como ECC devido à sigla em inglês para *Elliptic Curve Cryptography*, é aplicada à criptografia assimétrica, ou de chave pública, e também possui aplicabilidade na troca de chaves de cifras simétricas entre emissor e receptor. Quando utilizada para a troca de chaves, a criptografia de curvas elípticas recebe o nome de ECDH devido à sigla em para *Elliptic Curve Diffie–Hellman Key Exchange*. O algoritmo ECDH funciona de maneira semelhante ao algoritmo de troca de chaves proposto por Diffie e Hellman, porém, ao contrário deste, sua segurança não se baseia na intratabilidade computacional do problema do logaritmo discreto em um $\mathbb{Z}_p^*$, mas sim em uma versão análoga para

curvas elípticas, denominado problema do logaritmo discreto para curvas elípticas. Não apenas o algoritmo de troca de chaves ECDH, mas a própria troca de mensagens utilizando-se ECC se baseia na intratabilidade computacional desse problema. Mais à frente, serão apresentados mais detalhes acerca do problema do logaritmo discreto sobre $\mathbb{Z}_p^*$ e do problema do logaritmo discreto para curvas elípticas também sobre $\mathbb{Z}_p^*$.

3.2.1 Introdução às Curvas Elípticas

Antes de se apresentar a definição de uma curva elíptica, é necessário apresentar algumas outras definições importantes. Primeiramente, é preciso definir o que é um espaço projetivo: um espaço projetivo de dimensão n sobre um corpo K, representado por $\mathbb{P}_K^n$, é definido como o conjunto de todas as direções possíveis em K^{n+1}, isto é, cada ponto de $\mathbb{P}_K^n$ pode ser representado como um vetor não nulo $(a_0, a_1, ..., a_n) \in K^{n+1}$. Como 2 ou mais vetores proporcionais de K^{n+1} indicam a mesma direção, o conjunto desses vetores é tratado como apenas 1 ponto em $\mathbb{P}_K^n$, e essa é uma das grandes vantagens dos espaços projetivos. Para simplificar a notação, pode-se definir uma equivalência entre vetores de K^{n+1} da seguinte forma: os vetores não nulos $(a_0, a_1, ..., a_n)$ e $(b_0, b_1, ..., b_n)$ são equivalentes se existe um $\lambda \in K$ não nulo, tal que $a_i = \lambda b_i$, com $i = 0, 1, \ldots, n$. Dessa forma, cada classe de equivalência dos vetores de K^{n+1} representa 1 ponto em $\mathbb{P}_K^n$. Seja $(a_0, a_1, ..., a_n)$ um vetor de de K^{n+1}, sua classe de equivalência será representada por $(a_0 : a_1 : ... : a_n)$. Por exemplo, $\mathbb{P}_K^1$, que é conhecido como "reta projetiva", é formado pelas classes de equivalência não nulas $(a_0 : a_1)$, sendo $(a_0, a_1) \in K^2$. Ainda, podem-se separar as classes de equivalência em que $a_1 \neq 0$ daquelas em que $a_1 = 0$, da seguinte forma: caso $a_1 \neq 0$, a classe $(a_0 : a_1)$ é equivalente à classe $(\frac{a_0}{a_1} : 1)$, que também é equivalente à classe $(a_0 : 1)$; caso $a_1 = 0$, a classe $(a_0 : a_1)$ é equivalente à classe $(1 : 0)$. Portanto, $\mathbb{P}_K^1$ pode ser representado como a união de todas as classes da forma $(a_0 : 1)$, sendo $a_0 \in K$, com a classe $(1 : 0)$, sendo essa classe denominada "ponto no infinito" de $\mathbb{P}_K^1$. Em símbolos, tem-se:

$$\mathbb{P}_K^1 = \{(a_0 : 1), \text{ sendo } a_0 \in K\} \cup \{(1 : 0)\}$$

Analogamente, pode-se definir o "plano projetivo" $\mathbb{P}_K^2$ separando-se as classes da forma $(a_0 : a_1 : 1)$, com $(a_0, a_1) \in K^2$, das classes da forma $(a_0 : a_1 : 0)$, com a_0 e a_1 não ambos nulos, sendo essa classe denominada "reta no infinito" de $\mathbb{P}_K^2$. Em símbolos, tem-se:

$$\mathbb{P}_K^2 = \left\{(a_0 : a_1 : 1), \text{ sendo } (a_0, a_1) \in K^2\right\} \cup \{(a_0 : a_1 : 0), \text{ com } a_0 \text{ e } a_1 \text{ não ambos nulos}\}$$

Esse será o espaço projetivo utilizado na definição de curva elíptica, que será feita em breve.

A ideia por trás da utilização do espaço projetivo $\mathbb{P}_K^2$ é a definição de curvas projetivas nesse espaço. Por exemplo, em K^2, dado um polinômio qualquer $p(x, y) \in$

$K[x, y]$, pode-se definir uma curva C como sendo o conjunto dos pares ordenados (a, b), tais que $p(a, b) = 0$. Os elementos constituintes dessa curva são pares ordenados $(a, b) \in K^2$, que são chamados de pontos da curva. De maneira semelhante, pode-se definir uma curva C no espaço projetivo $\mathbb{P}^2_K$, em que os "pontos" da curva sejam classes de equivalência $(a : b : c)$ que sejam soluções de um $p(x, y, z) = 0$, com $p(x, y, z) \in K[x, y, z]$. Porém, como cada classe de equivalência representa infinitos vetores equivalentes $(a_0, a_1, a_2) \in K^3$, para que cada classe de equivalência seja um ponto da curva, é necessário que todos os vetores equivalentes dessa classe sejam solução de $p(x, y, z) = 0$. Isso pode ser facilmente satisfeito restringindo-se o polinômio $p(x, y, z)$ a um polinômio homogêneo, isto é, todos os seus monômios devem possuir o mesmo grau. Por exemplo, se $p(x, y, z)$ é homogêneo de grau r, então $p(\lambda x, \lambda y, \lambda z) = \lambda^r p(x, y, z)$. Dessa forma, se $p(a, b, c) = 0$, então $p(\lambda a, \lambda b, \lambda c) = \lambda^r p(a, b, c) = 0$, e conclui-se que todos os vetores dessa classe de equivalência serão soluções de $p(x, y, z) = 0$. Portanto, desde que se respeite a restrição de se utilizar um polinômio homogêneo, pode-se definir uma curva projetiva C como sendo o conjunto de todas as classes de equivalência $(a : b : c) \in \mathbb{P}^2_K$, tais que $p(a, b, c) = 0$.

Feita a elucidação dos conceitos de espaço projetivo e curva projetiva, finalmente pode-se definir uma curva elíptica. Para as aplicações criptográficas pretendidas neste trabalho, serão abordadas apenas curvas elípticas definidas sobre corpos de característica diferente de 2 e 3. Há uma definição mais abrangente de curvas elípticas, que engloba curvas definidas sobre corpos de qualquer característica. Essa definição não será abordada neste trabalho. Para mais detalhes sobre a definição mais abrangente de curvas elípticas, podem ser consultados (WASHINGTON, 2008) e (JUNIOR, 2003). Para as aplicações deste trabalho, define-se curva elíptica E sobre um corpo K como uma curva projetiva em $\mathbb{P}^2_K$ formada pelas classes de equivalência $(x : y : z)$, tais que $y^2z = x^3 + Axz^2 + Bz^3$, com constantes $A, B \in K$. Simbolicamente, tem-se:

$$E = \left\{(x : y : z) \in \mathbb{P}^2_K \mid y^2z = x^3 + Axz^2 + Bz^3, \text{ com constantes } A, B \in K\right\}$$

Conforme discutido anteriormente, essas classes de equivalência podem ser separadas em 2 grupos, aqueles em que $z \neq 0$ e aqueles em que $z = 0$. O grupo das classes em que $z \neq 0$ é equivalente a $(x : y : 1)$, de forma que a equação $y^2z = x^3 + Axz^2 + Bz^3$ se transforma em $y^2 = x^3 + Ax + B$. O grupo das classes em que $z = 0$ é equivalente a $(x : y : 0)$, de forma que a equação $y^2z = x^3 + Axz^2 + Bz^3$ se transforma em $0 = x^3$ e, portanto, $x = 0$. Nesse caso, as classes $(x : y : 0)$ que pertencem à curva elíptica são aquelas equivalentes a $(0 : y : 0)$, que é equivalente a $(0 : 1 : 0)$. Essa classe de equivalência é denominada "ponto no infinito" da curva elíptica, e denotada por O. Portanto, por uma questão de simplicidade, pode-se trabalhar com os pontos das curvas elípticas de maneira separada nas classes de equivalência em que $z \neq 0$ ou $z = 0$, como foi feito acima. Dessa forma, pode-se definir, simplificadamente, uma curva elíptica E

sobre um corpo K da seguinte forma:

$$E = \left\{(x, y) \in K^2 \mid y^2 = x^3 + Ax + B, \text{ com constantes } A, B \in K\right\} \cup \{\text{ponto no infinito } O\}$$

Essa definição é conhecida como forma reduzida de Weierstrass de uma curva elíptica. Apesar dessa definição simplificada não mostrar a verdadeira natureza das curvas elípticas, em que seus pontos são classes de equivalência de $\mathbb{P}^2_K$, ela é muito mais conveniente para se realizar operações entre pontos de curvas elípticas, e será adotada neste trabalho para as aplicações criptográficas. Caso necessário, pode-se voltar à definição projetiva sem grandes dificuldades. A partir deste ponto, somente a forma reduzida de Weierstrass será utilizada nas aplicações de curvas elípticas deste trabalho. As figuras 16 e 17 apresentam 2 exemplos de curvas elípticas definidas sobre $\mathbb{R}$, para uma melhor visualização dos seus "pontos". Vale salientar que os corpos K sobre os quais serão definidas as curvas elípticas para aplicações criptográficas são corpos finitos, de forma que a visualização de seus pontos não corresponde ao que se observa nas figuras 16 e 17, em que as curvas estão definidas sobre $\mathbb{R}$.

Figura 16 – Gráfico da Curva Elíptica $y^2 = x^3 - 4x$, definida sobre $\mathbb{R}$

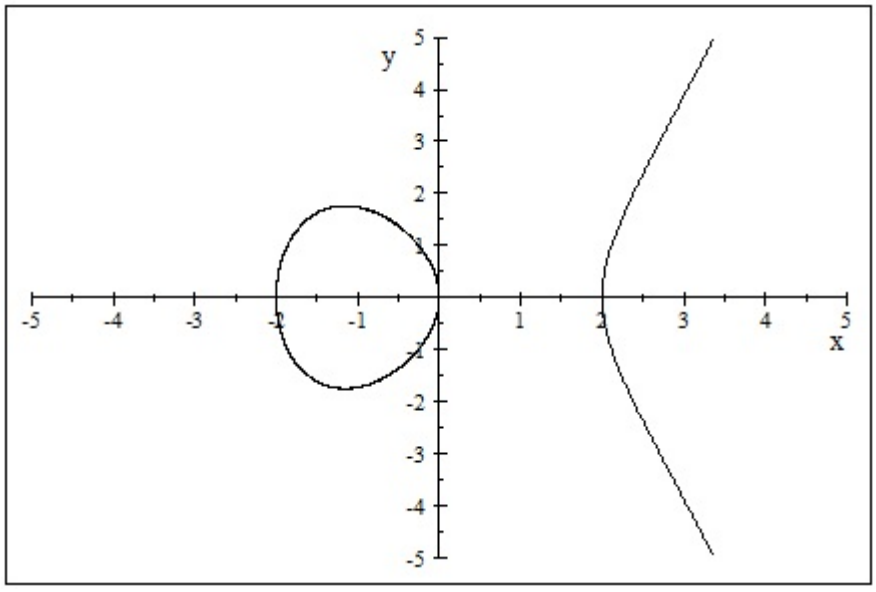
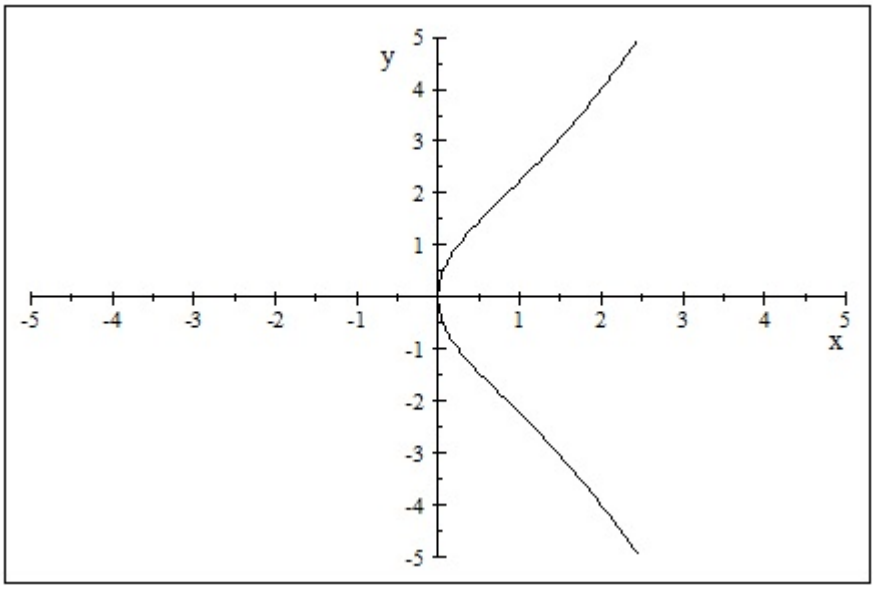


Figura 17 – Gráfico da Curva Elíptica $y^2 = x^3 + 4x$, definida sobre $\mathbb{R}$



O que permite a utilização de curvas elípticas em aplicações criptográficas é que se pode definir uma operação "'soma" entre pontos da curva, de forma que o conjunto de seus pontos, juntamente com a operação "soma" definida, adquirem uma estrutura de grupo abeliano. Para que essa operação esteja bem definida entre todos os pontos da curva elíptica, é necessário que a curva seja não singular. Uma curva elíptica E formada pelos pontos (x, y), tais que $p(x, y) = y^2 - x^3 - Ax - B = 0$, mais o ponto no infinito O, é dita não singular, ou suave, quando não possui pontos singulares. E um ponto (x, y) de E é dito singular quando $\frac{\partial p}{\partial x}(x, y) = \frac{\partial p}{\partial y}(x, y) = 0$. Portanto, para se garantir que a curva seja não singular, basta que as derivadas parciais de $p(x, y)$ não sejam nulas simultaneamente, isto é, deve-se evitar a seguinte situação:

$$\frac{\partial p}{\partial x}(x, y) = -3x^2 - A = 0$$

$$\frac{\partial p}{\partial x}(x, y) = 2y = 0$$

Portanto, não se deve permitir que ocorra simultaneamente $3x^2 + A = 0$ e $y = 0$. Porém, se $y = 0$, tem-se $x^3 + Ax + B = 0$. Logo, deve-se evitar a seguinte situação:

$$x^3 + Ax + B = 0$$

$$3x^2 + A = 0$$

Ora, mas para que não exista x que torne as igualdades acima verdadeiras simultaneamente, basta que o polinômio $q(x) = x^3 + Ax + B$ não possua raízes múltiplas. Por outro lado, sabe-se que o discriminante do polinômio cúbico $q(x)$, cujas raízes são x_1, x_2 e x_3, é dado por $((x_1 - x_2)(x_1 - x_3)(x_2 - x_3))^2 = -(4A^3 + 27B^2)$. Portanto, garantindo-se que $4A^3 + 27B^2 \neq 0$, garante-se também a inexistência de raízes múltiplas em $q(x)$,

garantindo-se assim também a não-singularidade da curva elíptica E. Portanto, para que a operação "soma" entre dois pontos da curva elíptica E sempre exista, basta que $4A^3 + 27B^2 \neq 0$.

3.2.2 Definição da Operação "soma" entre Pontos de Curvas Elípticas

Conforme citado anteriormente, o que possibilita a aplicação de curvas elípticas em criptografia é a estrutura de grupo abeliano que os pontos da curva adquirem quando é definida uma certa operação "soma" entre eles. Essa operação, denominada Lei de Grupo (WASHINGTON, 2008), (JUNIOR, 2003) ou Lei da Corda-Tangente (MARTINEZ et al., 2010), deve possuir as 5 propriedades fundamentais que definem um grupo abeliano, que são: fechamento, associatividade, elemento neutro, elemento inverso e comutatividade. O símbolo utilizado para se definir a operação "soma" de pontos de curvas elípticas é +, análogo ao símbolo da soma usual de números. Portanto, para que o conjunto de pontos de uma curva elíptica E, juntamente com a operação + formem um grupo abeliano, deve-se ter:

1. Fechamento: sejam P e Q pontos quaisquer de E. Então, $P + Q$ deve também ser um ponto de E;

2. Associatividade: sejam P, Q e R pontos quaisquer de E. Então, deve-se ter $(P + Q) + R = P + (Q + R)$;

3. Elemento Neutro: seja P um ponto qualquer de E. Então, deve existir um ponto N em E, tal que $P + N = N + P = P$. Mais à frente, ficará evidente que esse ponto N é justamente o ponto no infinito O;

4. Elemento Inverso: seja P um ponto qualquer de E. Então, deve existir um ponto $-P$ em C, tal que $P + (-P) = (-P) + P = N$;

5. Comutatividade: sejam P e Q pontos quaisquer de E. Então, deve-se ter $P + Q = Q + P$.

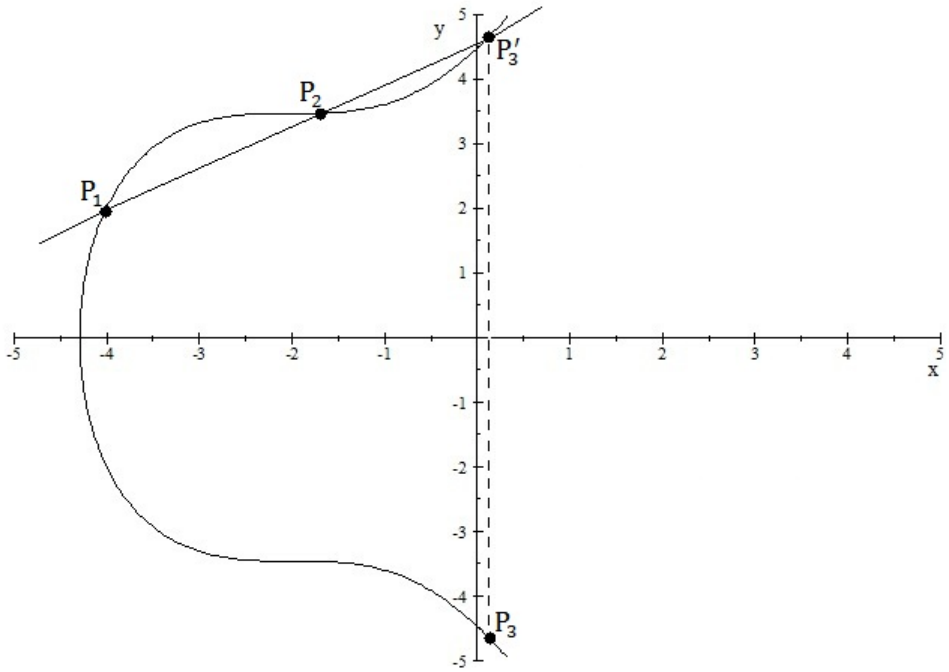
Das 5 propriedades acima, somente o fechamento e a associatividade não ficarão evidentes quando for apresentada a definição da "soma". Para uma demonstração de que tanto o fechamento quanto a associatividade, de fato, ocorrem com a definição dessa operação, podem ser consultados (WASHINGTON, 2008) ou (MARTINEZ et al., 2010).

Na definição da operação "soma" de pontos de uma curva elíptica, serão, inicialmente, consideradas curvas sobre o corpo $K = \mathbb{R}$, devido ao forte apelo geométrico da definição, utilizando-se retas secantes e tangentes, que são melhor visualizadas quando a curva é definida sobre $\mathbb{R}$. Porém, após o estabelecimento das expressões que definem

a operação, podem-se aplicar tais expressões em curvas elípticas definidas sobre qualquer corpo K cuja característica seja diferente de 2 e 3, basta que a curva elíptica seja não singular. Dessa forma, segue a definição da operação "soma":

Sejam $P_1 = (x_1, y_1)$ e $P_2 = (x_2, y_2)$ pontos pertencentes a uma curva elíptica E, dada pela equação $y^2 = x^3 + Ax + B$, com P_1 e $P_2 \neq O$. O ponto $P_3 = (x_3, y_3) = P_1 + P_2$, denominado "soma" de P_1 com P_2, é definido como sendo a reflexão, através do eixo x, do ponto de intersecção entre a reta que contém P_1 e P_2 e a curva elíptica E. Essa definição faz uso de conceitos geométricos, que podem ser melhor visualizados quando a curva elíptica E está definida sobre $\mathbb{R}$, e foi apresentada dessa forma dar algum significado geométrico às expressões seguintes, que valem para qualquer corpo K de característica diferente de 2 e 3. A figura 18 ilustra geometricamente o processo de adição de pontos em uma curva elíptica.

Figura 18 – Soma de Pontos em uma Curva Elíptica



Porém, o processo acima não contempla os casos em que algum dos pontos a serem somados, ou mesmo ambos, sejam o ponto no infinito O. Para estes casos, são necessárias algumas definições extras, que serão apresentadas a seguir. Na tentativa de deixar mais clara a definição da "soma" de pontos em uma curva elíptica de uma maneira mais geral, os casos possíveis serão abordados de maneira separada. Tem-se:

Caso 1 - $P_1 \neq P_2$ e ambos $\neq O$.

Se $x_1 = x_2$, para que os pontos P_1 e P_2 pertençam à curva elíptica e sejam pontos

distintos, é necessário que $y_1 = -y_2$, pois se tem:

$$x_1^3 + Ax_1 + B = x_2^3 + Ax_2 + B \Rightarrow y_1^2 = y_2^2 \Rightarrow y_1 = \pm y_2$$

Como não se pode ter $y_1 = y_2$, pois $P_1 \neq P_2$, então $y_1 = -y_2$. Dessa forma, a reta que passa por P_1 e P_2 é vertical e, aparentemente, não intercepta a curva elíptica. Porém, quando se utilizam coordenadas projetivas para se representar P_1 e P_2, a reta que passa por esses pontos intercepta a curva elíptica em $(0 : 1 : 0)$, que é o ponto no infinto O. Refletindo o ponto no infinito no eixo x, obtém-se o próprio ponto no infinito, pois $(0 : y : 0) = (0 : -y : 0) = (0 : 1 : 0)$. Portanto, a inversão do ponto O leva a ele próprio. Então, $P_1 + P_2 = O$. Nesse caso, denota-se $P_2 = -P_1$, pois O é o elemento neutro do grupo, conforme será mostrado nos casos 3 e 4 à frente.

Se $x_1 \neq x_2$, calcula-se a equação da reta que passa por P_1 e P_2 da forma: $(y - y_1) = m(x - x_1)$, com $m = \frac{(y_2 - y_1)}{(x_2 - x_1)}$. É necessário determinar o ponto em que essa reta interceptará a curva elíptica $y^2 = x^3 + Ax + B$. Fazendo-se y da reta coincidente com y da curva, obtém-se uma equação cúbica em x, dada por:

$$x^3 - m^2x^2 + (A - 2my_1 + 2m^2x_1)x + (B - m^2x_1^2 + 2mx_1y_1) = 0$$

As três raízes da equação acima fornecem as abscissas dos três pontos de intersecção entre a reta que passa por P_1 e P_2 e a curva elíptica. Porém, as abscissas de dois dos três pontos de intersecção já são conhecidas, pois P_1 e P_2 são pontos da curva. Portanto, é possível determinar a terceira abscissa utilizando-se a relação da soma das raízes da equação:

$$m^2 = x_1 + x_2 + x_3' \Rightarrow$$
$$x_3' = m^2 - x_1 - x_2 \text{ e } y_3' = m(x_3' - x_1) + y_1$$

Por fim, realiza-se a reflexão do ponto (x_3', y_3') no eixo x, e obtém-se o ponto $P_3 = (x_3, y_3) = P_1 + P_2$, com:

$$x_3 = x_3' = m^2 - x_1 - x_2$$
$$y_3 = -y_3' = m(x_1 - x_3) - y_1$$

Caso 2 - $P_1 = P_2 \neq O$.

Nesse caso, como $P_1 = P_2$, não é possível encontrar uma reta secante à curva que passe pelos dois pontos, pois eles são coincidentes. Porém, ao se construírem retas passando por dois pontos distintos de uma curva, quanto mais os pontos se aproximam um do outro, mais próxima essa reta se torna da reta tangente à curva naquele ponto. Portanto, no caso em que $P_1 = P_2$, basta que se tome a reta tangente à curva elíptica naquele ponto.

Se $y_1 = 0$, a reta é vertical, e o caso é análogo ao caso anterior, em que $P_1 \neq P_2$ e $x_1 = x_2$, resultando em $P_1 + P_2 = O$. Tem-se, portanto, $P_2 = -P_1$.

Se $y_1 \neq 0$, pode-se encontrar o coeficiente angular da reta tangente por meio de derivação implícita da equação da curva elíptica:

$$2y\frac{dy}{dx} = 3x^2 + A \Rightarrow$$

$$m = \frac{dy}{dx} = \frac{3x_1^2+A}{2y_1}$$

Então, obtém-se a equação da reta tangente $(y-y_1) = m(x-x_1)$ utilizando-se a inclinação m calculada acima. Fazendo-se y da reta coincidente com y da curva, obtém-se uma nova equação cúbica em x, dada por:

$$x^3 - m^2x^2 + (A - 2my_1 + 2m^2x_1)x + (B - m^2x_1^2 + 2mx_1y_1) = 0$$

Essa equação possui raiz dupla x_1, portanto:

$$x_3^{'} = m^2 - x_1 \quad \text{e} \quad y_3^{'} = m(x_3^{'} - x_1) + y_1 \Rightarrow$$

$$x_3 = m^2 - x_1 \quad \text{e} \quad y_3 = m(x_1 - x_3) - y_1$$

Caso 3 - $P_1 \neq O$ e $P_2 = O$.

Como $P_2 = O$, a reta que contém P_1 e P_2 é vertical para qualquer P_1 da curva elíptica considerada. Dessa forma, a reta irá interceptar a curva em um ponto $P_3^{'} = (-P_1)$, que é a reflexão do ponto P_1 pelo eixo x. Por fim, procedendo-se a reflexão do ponto $P_3^{'}$ pelo eixo x, obtém-se o próprio ponto P_1. Portanto, $P_1 + O = P_1$.

Caso 4 - $P_1 = P_2 = O$.

Este caso é uma extensão do caso anterior $P_1 + O = P_1$, permitindo-se que $P_1 = O$. Portanto, obtém-se: $O + O = O$.

Pela análise dos Casos 3 e 4, percebe-se que o ponto O funciona como elemento neutro da "soma" definida. Vale salientar também que o processo de "soma" de pontos de curvas elípticas descrito acima não equivale a uma simples adição das coordenadas desses pontos e, portanto, embora compartilhem o mesmo símbolo +, não se deve confundir a operação "soma" de pontos de uma curva elíptica com a operação usual de soma de vetores do $\mathbb{R}^2$.

A título de ilustração do método apresentado, seguem abaixo dois exemplos numéricos de "soma" de pontos de uma curva elíptica. Todos os cálculos foram realizados no software *SageMath*.

Exemplo 1: Seja a curva elíptica E, definida sobre o corpo dos reais $\mathbb{R}$, dada pela equação $y^2 = x^3 + 73$. Por inspeção, constata-se que $P_1 = (2, 9)$ e $P_2 = (3, 10)$ são pontos (pares ordenados) pertencentes à curva elíptica E. Para se obter o ponto $P_3 = P_1 + P_2$, inicialmente é necessário encontrar a equação da reta que passa pelos pontos P_1 e

P_2, que é $y = x + 7$. Essa equação é facilmente obtida conforme descrito no Caso 1. Fazendo-se y da equação coincidente com y da curva elíptica, obtém-se:

$$(x+7)^2 = x^3 + 73 \qquad \Rightarrow$$
$$x^3 - x^2 - 14x + 24 = 0$$

Como já se conhecem duas das raízes dessa equação ($x_1 = 2$ e $x_2 = 3$), obtém-se facilmente a terceira raiz $x_3^{'} = -4$. Substituindo-se $x_3^{'}$na equação da reta ou da própria curva elíptica, obtém-se $y_3^{'} = 3$. Procedendo-se a inversão do ponto $(x_3^{'}, y_3^{'})$, obtém-se o ponto $P_3 = (-4, -3)$. Alternativamente, caso se queira calcular P_3 de uma maneira direta, podem-se utilizar as fórmulas já determinadas no Caso 1:

$$x_3 = m^2 - x_1 - x_2$$
$$y_3 = m(x_1 - x_3) - y_1$$

Dessa forma, têm-se:

$$x_3 = 1^2 - 2 - 3 = -4$$
$$y_3 = 1 \cdot (2 - (-4)) - 9 = -3$$

Obtendo-se, assim, o ponto $P_3 = (-4, -3)$.

Exemplo 2: Seja a curva elíptica E, definida sobre $\mathbb{Z}_{11}$, dada pela equação $y^2 = x^3 + x + 6$. Como 11 é um número primo, $\mathbb{Z}_{11}$ é, de fato, um corpo, e essa demonstração é bastante simples e não será realizada aqui. Por inspeção, constata-se que $P_1 = (2, 7)$ pertencente à curva elíptica. Para se obter o ponto $2P_1 = P_1 + P_1$, utilizando-se as fórmulas já determinadas no Caso 2, têm-se:

$$m = \frac{dy}{dx} = \frac{(3x_1^2 + A)}{(2y_1)}$$

$$x_3 = m^2 - 2x_1 \qquad \text{e} \qquad y_3 = m(x_1 - x_3) - y_1$$

Portanto:

$$m = \frac{3 \cdot 2^2 + 1}{2 \cdot 7} \; mod(11) \quad \Rightarrow$$

$$m = \frac{13}{14} \; mod(11) \quad \Rightarrow$$

$$m = 2 \cdot 3^{-1} \; mod(11) \quad \Rightarrow$$

$$m = 2 \cdot 4 \; mod(11) \quad \Rightarrow$$

$$m = 8 \; mod(11)$$

Então:

$$x_3 = 8^2 - 2 \cdot 2 \ mod(11) \quad \Rightarrow$$

$$x_3 = 60 \ mod(11) \quad \Rightarrow$$

$$x_3 = 5 \ mod(11)$$

$$y_3 = 8 \cdot (2 - 5) - 7 \ mod(11) \quad \Rightarrow$$

$$y_3 = -31 \ mod(11) \quad \Rightarrow$$

$$y_3 = -9 \ mod(11) \quad \Rightarrow$$

$$y_3 = 2 \ mod(11)$$

Assim, obtém-se o ponto $2P_1 = P_1 + P_1 = (5, 2)$

Por fim, seja P um ponto pertencente a uma curva elíptica E. Se k for um inteiro não nulo, então o produto kP será $kP = P + P + ... + P$, com k parcelas, se $k > 0$. Caso $k < 0$, então $kP = (-k)(-P) = (-P) + (-P) + ... + (-P)$. Uma estratégia interessante para se realizar a multiplicação kP consiste em se realizar sucessivas duplicações e somas de P, até que se obtenha kP. Essa técnica é conhecida como *double-and-add method*, e permite o cálculo de kP sem a necessidade de grande quantidade de memória. A técnica funciona da seguinte forma: para se calcular $120P$, por exemplo, basta realizar a sequência $120P = 2(2(2(P + (2(P + (2(2P + P)))))))$ de duplicações e somas de pontos da curva elíptica.

3.2.3 Ordem de uma Curva Elíptica e o Teorema de Hasse

Define-se ordem de uma curva elíptica E sobre um corpo K, e denota-se por $\#E$, a quantidade de pontos (pares ordenados (x, y)) pertencentes à curva, mais o ponto no infinito O. Também, define-se ordem de um ponto P de uma curva elíptica como sendo o menor inteiro positivo k, tal que $kP = O$. Por exemplo, seja a curva elíptica E definida sobre o corpo $\mathbb{Z}_{11}$, dada pela equação $y^2 = x^3 + x + 6$, a mesma utilizada no exemplo 2. Uma maneira possível de se encontrar a ordem da curva é determinar todos os seus pontos. Isso pode ser feito por tentativas sucessivas, por exemplo, fazendo $x = 0, 1, 2, ..., 10$ e verificando quais valores de y satisfazem a equação da curva *modulo* 11. Dessa forma, encontram-se os seguintes pontos: $(2, 4)$, $(2, 7)$, $(3, 5)$, $(3, 6)$, $(5, 2)$, $(5, 9)$, $(7, 2)$, $(7, 9)$, $(8, 3)$, $(8, 8)$, $(10, 2)$ e $(10, 9)$. Além desses pontos, há ainda o ponto no infinito O, que pertence à curva. Portanto, como a curva possui 13 pontos, sua ordem $\#E = 13$. Como as curvas elípticas possuem uma estrutura de grupo abeliano com relação à sua

soma de pontos, caso a ordem de uma curva elíptica seja um número primo, pode-se mostrar que o grupo formado por seus pontos é cíclico e, nesse caso, cada ponto da curva, com exceção do ponto no infinito, é uma raiz primitiva desse grupo. Esse fato é uma consequência direta do teorema de Lagrange, que estabelece que a ordem de cada um dos elementos de um grupo finito deve ser um divisor da ordem do grupo. Portanto, no caso da curva $y^2 = x^3 + x + 6$ definida sobre $\mathbb{Z}_{11}$, todos os seus pontos possuem ordem 13. Por exemplo, para o ponto $P = (2, 4)$, isso pode ser verificado por meio dos seguintes cálculos:

$$2P = (5, 9)$$
$$3P = (8, 8)$$
$$4P = (10, 9)$$
$$5P = (3, 5)$$
$$6P = (7, 2)$$
$$7P = (7, 9)$$
$$8P = (3, 6)$$
$$9P = (10, 2)$$
$$10P = (8, 3)$$
$$11P = (5, 2)$$
$$12P = (2, 7)$$
$$13P = O$$

Para aplicações criptográficas de curvas elípticas, é fundamental que se conheça a ordem da curva sobre o corpo K, pois esse número é um dos principais parâmetros a serem escolhidos nos algoritmos de criptografia com base em curvas elípticas. Quando se utilizam corpos de ordem elevada, torna-se impraticável determinar a ordem da curva elíptica encontrando-se todos os seus pontos e os contando. Portanto, é fundamental que se conheça a ordem da curva de uma maneira indireta, sem que seja necessário determinar todos os seus pontos. O teorema de Hasse fornece uma boa estimativa da ordem de uma curva elíptica, sem que seja necessário encontrar todos os seus pontos. Esse teorema estabelece que, dada uma curva elíptica E definida sobre um corpo finito K com q elementos, então a ordem de E satisfaz a seguinte relação:

$$-2\sqrt{q} \leq q + 1 - \#E \leq 2\sqrt{q}$$

Portanto:

$$q + 1 - 2\sqrt{q} \leq \#E \leq q + 1 + 2\sqrt{q}$$

Utilizando-se o teorema de Hasse, pode-se estabelecer um intervalo de valores possíveis para a ordem da curva E. Em aplicações práticas, pode-se utilizar o teorema de Hasse juntamente com o teorema de Lagrange, pois, quando se conhece a ordem de algum elemento do grupo, sabe-se que se trata de um divisor da ordem do grupo.

Portanto, quando se conhece a ordem de algum dos pontos de uma curva elíptica, sabe-se que se trata de um divisor da ordem da própria curva. Essa informação, juntamente com o intervalo fornecido pelo teorema de Hasse, muitas vezes é suficiente para se estabelecer a ordem da curva elíptica. Para o caso particular de se conhecer a ordem de um ponto que seja uma raiz primitiva da curva, mesmo que não se saiba previamente que tal ponto é uma raiz primitiva, a união das duas informações (ordem do ponto e teorema de Hasse) sempre possibilita a determinação exata da ordem da curva. Porém, ainda há o problema de como se obter a ordem de um ponto da curva. Um dos métodos utilizados para isso é conhecido como *Baby Step, Giant Step*. Trata-se de um método que pode ser aplicado a alguns tipos de problemas, e que diminui pela metade o expoente da quantidade de tentativas necessárias para se resolver um problema utilizando-se o método da força bruta. Por exemplo, se um problema precisa de x tentativas para ser resolvido, utilizando-se o método *Baby Step, Giant Step*, são necessárias apenas $\sqrt{x}$ tentativas. Esse método pode ser utilizado, inclusive, para se buscar a ordem da própria curva elíptica, e será descrito juntamente com o procedimento para se buscar a ordem de um determinado ponto da curva elíptica. Seja P um ponto pertencente a uma curva elíptica E definida sobre um corpo finito K, com q elementos. Então, o algoritmo *Baby Step, Giant Step* prevê os seguintes passos:

1. Calcule $Q = (q + 1)P$
2. Escolha um inteiro $m > q^{\frac{1}{4}}$ e calcule todos os pontos $\pm jP$, para $j = 0, 1, 2, \ldots, m$
3. Calcule todos pontos $Q + k(2mP)$, para $k = -m, -(m-1), \ldots, m$, até que se obtenha um dos pontos $\pm jP$ calculados no passo anterior
4. O ponto $(q + 1 + 2mk \mp j)P$ será o ponto no infinito O. Seja $M = q + 1 + 2mk \mp j$
5. Fatore M. Sejam $p_1, ..., p_r$seus fatores primos distintos
6. Calcule $(\frac{M}{p_i})P$, para $i = 1, 2, \ldots, r$. Se $(\frac{M}{p_i})P = O$ para algum i, então substitua M por $(\frac{M}{p_i})$ e volte ao passo 5. Se $(\frac{M}{p_i})P \neq O$ para todos os valores de i, então M é a ordem do ponto P
7. Para se determinar $\#E$, basta repetir os passos 1 a 6 para vários pontos de E, até que o *MMC* das ordens desses pontos divida apenas um inteiro N no intervalo $q + 1 - 2\sqrt{q} \leq N \leq q + 1 + 2\sqrt{q}$
8. $\#E = N$

Muitas vezes, conhece-se a ordem de uma curva elíptica definida sobre um corpo finito K com uma quantidade q de elementos, sendo q um número pequeno, e deseja-se conhecer a ordem dessa mesma curva definida sobre um corpo finito K com

q^n elementos, para algum $n \in \mathbb{N}$. Nesses casos, pode-se recorrer ao seguinte teorema, derivado de uma das conjecturas de Weil (SILVERMAN, 2013), (JUNIOR, 2003), que estabelece o seguinte:

Seja a ordem de uma curva elíptica E definida sobre um corpo finito K com q elementos, dada por $\#E = q + 1 - a$, para algum inteiro a. Então, a ordem de E definida sobre um outro corpo finito K', com q^n elementos, é dada por $q^n + 1 - (\gamma^n + \beta^n)$, sendo γ e β determinados pela fatoração $x^2 - ax + q = (x - \gamma)(x - \beta)$. Esse teorema possibilita que se conheça a ordem de uma curva elíptica definida sobre um corpo finito com uma quantidade grande de elementos, conhecendo-se apenas a ordem da curva definida sobre um corpo finito com um número bem menor de elementos, desde que a quantidade de elementos do primeiro corpo seja uma potência da quantidade de elementos do segundo corpo. Como exemplo de aplicação prática dos teoremas definidos acima, juntamente com o teorema de Lagrange, a seguir serão apresentados alguns exemplos numéricos:

Exemplo 1 - Seja uma curva elíptica E, definida sobre $\mathbb{Z}_{79}$, que possui $q = 79$ elementos, descrita pela equação $y^2 = x^3 + 10x + 5$. Seja o ponto $P = (30, 45)$, pertencente a E. Utilizando-se o método *Baby Step, Giant Step*, pode-se encontrar a ordem de P:

1. $Q = (79 + 1)P = 80P = (33, 13)$
2. Seja $m = 5 > 79^{\frac{1}{4}}$. Então:

$$\begin{array}{ll} 0.P = O & (-0)P = O \\ 1.P = (30, 45) & (-1)P = (30, 34) \\ 2.P = (44, 25) & (-2)P = (44, 54) \\ 3.P = (28, 65) & (-3)P = (28, 14) \\ 4.P = (42, 75) & (-4)P = (42, 4) \\ 5.P = (33, 66) & (-5)P = (33, 13) \end{array}$$

3. Temos:

$$\begin{aligned} Q + (-5)(2 \cdot 5 \cdot P) &= Q - 50P = (3, 33) \\ Q + (-4)(2 \cdot 5 \cdot P) &= Q - 40P = (6, 26) \\ Q + (-3)(2 \cdot 5 \cdot P) &= Q - 30P = (45, 16) \\ Q + (-2)(2 \cdot 5 \cdot P) &= Q - 20P = (61, 47) \\ Q + (-1)(2 \cdot 5 \cdot P) &= Q - 10P = (14, 19) \\ Q + (0)(2 \cdot 5 \cdot P) &= Q = (33, 13) \end{aligned}$$

   O ponto $Q + k(2mP)$ está na lista do passo 2, quando $k = 0$ e $j = -5$

4. $M = 79 + 1 + 5 = 85$
5. $M = 5 \cdot 17$

6. $17P = (7, 24) \neq O$, $5P = (33, 66) \neq O$. Portanto, $M = 85$ é a ordem do ponto P.

Portanto, como a ordem de P é 85, a ordem da curva deve ser um múltiplo de 85. De acordo com o teorema de Hasse, tem-se: $63 \leq \#E \leq 97$. Como o único múltiplo de 85 pertencente ao intervalo é o próprio 85, conclui-se que a ordem da curva é exatamente 85. Em outras palavras, $\#E = 85$.

Exemplo 2 - Seja uma curva elíptica E, definida sobre $\mathbb{Z}_{103}$, que possui $q = 103$ elementos, descrita pela equação $y^2 = x^3 + 7x + 12$. De acordo com o teorema de Hasse, como $q = 103$, tem-se: $84 \leq \#E \leq 124$. Sejam os pontos $P = (19, 0)$ e $Q = (102, 2)$, pertencentes a E. Utilizando-se o método *Baby Step, Giant Step*, pode-se concluir que a ordem de P é 2, e a ordem de Q é 13. Portanto, a ordem de E deve ser um múltiplo de $2.13 = 26$. Como 104 é o único múltiplo de 26 pertencente ao intervalo do teorema de Hasse, conclui-se que a ordem da curva é 104. Isto é, $\#E = 104$.

Exemplo 3 - Seja uma curva elíptica E, definida sobre $K = \mathbb{Z}_{13}$, que possui $q = 13$ elementos, descrita pela equação $y^2 = x^3 + 10x + 5$. Sabe-se que a ordem de E definida sobre K é 10. Deseja-se determinar a ordem de E sobre um corpo $K' = GF(13^5)$, que possui $13^5 = 371293$ elementos. Essa ordem será dada por $q^n + 1 - (\gamma^n + \beta^n)$, sendo $n = 5$, e γ e β determinados por $x^2 - ax + 13 = (x - \gamma)(x - \beta)$, sendo $a = q + 1 - \#E = 13 + 1 - 10 = 4$. Portanto, realizando-se os cálculos, tem-se:

$$x^2 - 4x + 13 = (x - \gamma)(x - \beta) \quad \Rightarrow \quad \gamma = 2 + 3i \text{ e } \beta = 2 - 3i$$

Portanto:

$$13^5 + 1 - ((2 + 3i)^5 + (2 - 3i)^5) = 371293 + 1 - 244 = 371050$$

Dessa forma, conclui-se que a ordem de E definida sobre $K' = GF(13^5)$ é 371050.

A demonstração do teorema de Hasse e da validade do método *Baby Step, Giant Step* podem ser encontradas em (WASHINGTON, 2008).

3.2.4 Problema do Logaritmo Discreto para Curvas Elípticas

A utilização das curvas elípticas em criptografia só é possível porque se pode construir um problema com curvas elípticas que se mostra como uma função de única via, isto é, o cálculo é simples em uma direção, mas extremamente complicado na direção inversa. Isso se aplica às curvas elípticas da seguinte forma: dado um ponto P pertencente a uma curva elíptica E, definida sobre um corpo finito K, e um inteiro k, é computacionalmente simples calcular o produto $Q = kP$. Porém, dados Q, P, E e K, obter k é uma tarefa extremamente difícil do ponto de vista computacional. É nessa assimetria da função multiplicação de pontos por um inteiro, no universo das curvas elípticas, que se baseia a segurança das aplicações criptográficas que utilizam curvas elípticas. Essa assimetria produz o que se denomina problema do logaritmo discreto

para curvas elípticas. Apesar de não haver cálculo de exponenciações ou de logaritmos no processo de soma de pontos de curvas elípticas, esse nome se originou devido a sua semelhança com um outro problema, conhecido como problema do logaritmo discreto sobre corpos finitos, em que, de fato, ocorrem tais cálculos. O problema do logaritmo discreto sobre corpos finitos consiste no seguinte problema: Sejam a e b inteiros não nulos *modulo* p, com p primo e $MDC(a, p) = 1$. Caso exista um inteiro k, tal que $a^k \equiv b \ mod(p)$, é fácil calcular b a partir de a, k e p, mas é difícil determinar k a partir de a, b e p. Vale salientar que o valor de k, tal que $a^k \equiv b \ mod(p)$ não é único, haja vista que qualquer $k' = k + n(p-1)$, $n \in \mathbb{N}$, também é solução da equação modular, como consequência direta do teorema de Euler. Para não trabalhar com múltiplas soluções para a equação, costuma-se representar a sua solução *modulo* $(p-1)$, eliminando-se assim o tratamento de múltiplas raízes. Pode-se definir o problema do logaritmo discreto de maneira um pouco mais abrangente, para qualquer grupo multiplicativo G, da seguinte forma: Sejam a e $b \in G$. Dado um inteiro positivo k, tal que $a^k = b$, o problema de se determinar b a partir de a e k costuma ser fácil de se computar utilizando-se a técnica *double-and-add*, isto é, realizando-se uma sequências de duplicações a adições do elemento a. Porém, dados a e b, costuma ser muito difícil encontrar k, pois não se observa qualquer padrão nos resultados que surgem à medida que a é operado com o resultado da operação anterior, o que faz com que seja necessário computar uma grande quantidade de tentativas a^i, com $i = 2, 3, \ldots, k$ para que se obtenha b, e isso é computacionalmente muito custoso, daí o nome problema do logaritmo discreto. Como os pontos de curvas elípticas definidas sobre corpos finitos, juntamente com a operação "soma" definida anteriormente, possuem estrutura de grupo multiplicativo finito, o problema do logaritmo discreto também se aplica para as curvas elípticas. O problema consiste no fato de, dados P, k, E e K, é simples computar $Q = kP$, porém, dados P, Q, E e K, é computacionalmente difícil encontrar k. A segurança da aplicação criptográfica das curvas elípticas se baseia na dificuldade de se resolver esse problema, desde que sejam respeitadas algumas restrições acerca da escolha da curva E, do corpo K e do ponto P.

Antes de se apresentar como funcionam os algoritmos de ECC, convém ressaltar que a mensagem a ser cifrada precisa ser codificada em um valor numérico, para que as operações matemáticas pertinentes possam ser realizadas. Isso pode ser executado de maneira bastante simples, por exemplo, utilizando-se o padrão ASCII, já apresentado anteriormente. Porém, quando se utilizam algoritmos de ECC, é necessário que se estabeleça uma relação entre a mensagem numericamente codificada e um ponto pertencente à curva elíptica utilizada no algoritmo. Dessa forma, a mensagem numericamente codificada é inicialmente transformada em um ponto da curva elíptica e, após a realização das operações matemáticas pertinentes, obtém-se um novo ponto, também pertencente à curva elíptica. Esse novo ponto constitui a mensagem cifrada,

que deve ser enviada para o receptor. Portanto, é necessário utilizar-se um método de conversão entre uma mensagem numericamente codificada m a ser cifrada e um ponto pertencente à curva elíptica utilizada no algoritmo. Embora haja vários métodos conhecidos para se realizar essa correspondência, será apresentado apenas um deles, desenvolvido por Neal Koblitz, um dos pesquisadores pioneiros na aplicação de curvas elípticas em criptografia. Trata-se de um método probabilístico, que estabelece uma relação entre a mensagem numericamente codificada m e um ponto da curva elíptica utilizada, com uma probabilidade de sucesso de $1 - \frac{1}{2^T}$, sendo $T \in \mathbb{N}^*$ um parâmetro controlável. Dessa forma, controlando-se T, pode-se limitar em valores bem pequenos a probabilidade de fracasso do método. Segue uma descrição detalhada do método: Seja uma curva elíptica E definida sobre um corpo K com característica p, dada pela equação $y^2 = x^3 + Ax + B$. Seja m a mensagem que se quer cifrar, já previamente codificada em valor numérico. Estabelece-se o valor do parâmetro T, de forma que $\frac{1}{2^T}$ seja o máximo valor aceitável para a probabilidade de o método falhar. Deve-se ter $0 \leq m < \frac{p}{T}$. Caso $m \geq \frac{p}{T}$, deve-se quebrar a mensagem m em outras mensagens menores, e cifrá-las separadamente. Porém, na prática, como a grande maioria das mensagens cifradas com algoritmos assimétricos são chaves secretas pequenas, que serão utilizadas em criptografia simétrica, a necessidade de se quebrar a mensagens não ocorre com muita frequência. Seja $x_j = Tm + j$, para $0 \leq j < T$. Para cada um dos valores de x_j, deve-se calcular $s_j = x_j^3 + Ax_j + B$. Se $s_j^{\frac{p-1}{2}} \equiv 1 \ mod(p)$, então s_j é um quadrado *modulo p*, e a equação da curva elíptica está satisfeita para o ponto $(x_j, \sqrt{s_j})$. Para se recuperar a mensagem m a partir do ponto $(x_j, \sqrt{s_j})$, basta calcular $m = \lfloor \frac{x_j}{T} \rfloor$ em que $\lfloor \frac{x_j}{T} \rfloor$ representa o maior inteiro menor ou igual a $\frac{x_j}{T}$. Dessa forma, procedendo-se por tentativas, calcula-se s_j até que se encontre um quadrado *modulo p*, ou até que se tenha $j = T$ e nenhum quadrado *modulo p* tenha sido encontrado. Nesse caso, o método falhou. Porém, desde que se escolha um valor adequado para T, a probabilidade de falha do método pode ser limitada em valores bastante pequenos. Como cada um dos s_j é, em tese, um elemento aleatório de K, a probabilidade de s_j ser um quadrado *modulo p* é de aproximadamente $\frac{1}{2}$. Como faz-se j variar de 0 a $(T - 1)$, então a probabilidade de nenhum dos valores s_j corresponder a um quadrado *modulo p* é de $\frac{1}{2^T}$.

Segue um exemplo numérico a título de ilustração do método: Seja a curva elíptica $y^2 = x^3+2x+7$, definida sobre $\mathbb{Z}_{179}$. Admitindo-se uma probabilidade de falha do método de $\frac{1}{2^{20}} \cong 0,00000095$, toma-se $T = 20$. Seja $m = 5$ a mensagem que se quer cifrar. Então, faz-se $x_j = 100+j$, com $0 \leq j < 20$. Para $j = 4$, tem-se $104^3+2\cdot104+7 \equiv 64 \ mod(179)$. Como $64 \equiv 8^2 \ mod(179)$, então a mensagem $m = 5$ pode ser representada pelo ponto $P_m = (104, 8)$. Para se recuperar a mensagem m a partir do ponto P_m, basta fazer $m = \lfloor \frac{104}{20} \rfloor = 5$, que de fato é o valor de m.

3.2.5 Criptografia com Curvas Elípticas

Existem vários algoritmos criptográficos baseados na intratabilidade do problema do logaritmo discreto para curvas elípticas. Como exemplos, podem-se citar o algoritmo de troca de chaves de Diffie-Hellman ECDH (utilizado na troca de chaves do aplicativo WhatsApp), o algoritmo de Massey-Omura, o algoritmo Menezes-Vanstone, entre vários outros. Neste trabalho, serão abordados o algoritmo *ElGamal* baseado em curvas elípticas, e o algoritmo *ElGamal* para assinatura digital. Para uma descrição mais detalhada dos outros algoritmos, podem ser consultadas as referências (MENEZES; OORSCHOT; VANSTONE, 2001) e (WASHINGTON, 2008). Inicialmente, será apresentado o algoritmo criptográfico *ElGamal* na sua versão original e, em seguida, será apresentada a sua versão para curvas elípticas.

O algoritmo criptográfico *ElGamal* consiste em um algoritmo de criptografia de chave pública, isto é, emissor e receptor possuem, cada um, um par de chaves, uma pública e outra privada. Por exemplo, vamos supor que Alice e Bob desejam trocar mensagens cifradas utilizando o algoritmo *ElGamal*. Alice deseja enviar uma mensagem m cifrada para Bob. Primeiramente, é necessário que Bob escolha seu par de chaves e divulgue a sua chave pública. Para isso, Bob escolhe um número primo grande p, um número inteiro γ *modulo* p e um outro número inteiro a. Em seguida, ele calcula $\beta \equiv \gamma^a \ mod(p)$, divulga p, γ e β como sua chave pública e mantém a secreto, como sua chave privada. Alice, que deseja enviar a mensagem m para Bob, escolhe então aleatoriamente um número inteiro k e calcula os seguintes valores:

$$y_1 \equiv \gamma^k \ mod(p)$$

$$y_2 \equiv m\beta^k \ mod(p)$$

Alice envia então (y_1, y_2) para Bob, essa é a mensagem m cifrada. Para decifrá-la, Bob calcula:

$$y_2 y_1^{-a} \equiv m\beta^k(\gamma^k)^{-a} \equiv m\beta^k(\gamma^a)^{-k} \equiv m\beta^k\beta^{-k} \equiv m \ mod(p)$$

obtendo assim a mensagem original. A seguir, será apresentada a versão do algoritmo *ElGamal* com curvas elípticas.

O algoritmo criptográfico *ElGamal* com curvas elípticas consiste também em um algoritmo de criptografia de chave pública. Analogamente ao caso anterior, supõe-se que Alice deseja enviar para Bob uma mensagem m, cifrada utilizando-se o algoritmo *ElGamal* com curvas elípticas. Primeiramente, é necessário que Bob escolha seu par de chaves e divulgue a sua chave pública para Alice. Para isso, Bob escolhe uma curva elíptica E e um corpo finito K, tais que o problema do logaritmo discreto seja difícil de se resolver no grupo formado pelos pontos da curva. Bob também escolhe um ponto $P \in E$, tal que a ordem de P possua, pelo menos, um fator primo "grande", pois assim

previne-se um tipo de ataque ao problema do logaritmo discreto denominado método de Pohlig-Hellman (para detalhes, consultar (MENEZES; OORSCHOT; VANSTONE, 2001) e (STINSON, 2006)). Na prática, muitas vezes toma-se um ponto P, tal que sua ordem seja um número primo "grande", sem outros fatores. Esse ponto P é denominado ponto base do algoritmo. Por fim, Bob também escolhe um número inteiro s, e calcula $Q = s.P$. Então, Bob divulga E, K e P como parâmetros escolhidos para o algoritmo, divulga Q como sua chave pública e mantém s secreto, como sua chave privada. Alice, que deseja enviar a mensagem m para Bob, inicialmente expressa m como um ponto $M \in E$. Em seguida, ela escolhe um número inteiro k (mantendo-o em segredo) e calcula os seguintes pontos:

$$M_1 = kP$$

$$M_2 = M + kQ$$

Alice envia então (M_1, M_2) para Bob, essa é a mensagem m cifrada. Para decifrá-la, Bob calcula:

$$M_2 - sM_1 = M + kQ - skP = M + ksP - skP = M$$

obtendo assim a mensagem original. Supondo-se que o canal utilizado por Alice para transmitir a mensagem cifrada para Bob não seja seguro, e haja um espião, Eve, capaz de interceptar a mensagem (M_1, M_2) enviada por Alice. Eve também conhece E, K, P e Q, que são públicos. A partir dessas informações, para que Eve consiga obter a mensagem M, é necessário que ela conheça s e calcule $M = M_2 - sM_1$, ou que ela conheça k e calcule $M = M_2 - kQ$. Porém, como $Q = sP$, para se obter s a partir de P e Q, é necessário que Eve resolva o problema do logaritmo discreto para curvas elípticas, que não possui tratamento computacional eficiente caso se façam boas escolhas de E, K e P. O mesmo ocorre quando Eve tenta obter k a partir de M_1 e P, pois $M_1 = kP$. Nesses 2 problemas do logaritmo discreto para curvas elípticas reside a segurança do algoritmo.

Segue um exemplo numérico a título de ilustração: Supondo-se que Alice deseja enviar uma mensagem m, representada pelo ponto $M = (5, 1743)$ para Bob, que publicou os parâmetros do algoritmo como sendo $E : y^2 = x^3 + 3x + 45$, $K = \mathbb{Z}_{8831}$ e $P = (4, 11)$. Secretamente, Bob escolheu $s = 3$, e também publicou o ponto $Q = 3.P = (413, 1808)$, que é sua chave pública. Alice escolhe $k = 8$, por exemplo, e calcula:

$$M_1 = 8P = (5415, 6321)$$

$$M_2 = M + 8Q = (6626, 3576)$$

Então, Alice mantém k em segredo, e envia para Bob a mensagem cifrada (M_1, M_2). Para decifrá-la, Bob calcula:

$$M_2 - 3M_1 = (5, 1743)$$

obtendo-se assim a mensagem original.

Por fim, é preciso salientar que, para que o problema do logaritmo discreto para curvas elípticas seja difícil o suficiente para garantir a segurança do algoritmo, algumas restrições devem ser respeitadas na escolha da curva elíptica e do corpo finito utilizados. Algumas classes especiais de curvas elípticas, conhecidas como curvas supersingulares e curvas anômalas, devem ser evitadas para a aplicação criptográfica. Uma curva elíptica E definida sobre um corpo K com $q = p^m$ elementos (p primo e m inteiro), tal que $\#E = q + 1 - a$, é chamada de supersingular quando p divide a. Em outras palavras, E é supersingular se $a \equiv 0 \ mod(p)$. As curvas supersingulares devem ser evitadas porque o problema do logaritmo discreto para curvas elípticas nessa classe de curvas, quando submetido a um ataque conhecido como ataque MOV - Menezes, Okamoto, Vanstone, pode ser convertido em um problema do logaritmo discreto convencional, consideravelmente mais simples de ser resolvido, por exemplo, utilizando-se o ataque conhecido como *Index Calculus*. Para mais detalhes acerca desses métodos de ataque, podem ser consultadas as referências (MENEZES; OORSCHOT; VANSTONE, 2001), (STINSON, 2006) e (WASHINGTON, 2008). Uma curva elíptica E definida sobre um corpo K com $q = p^m$ elementos é chamada de anômala quando $\#E = q$. Vale ressaltar que uma curva E anômala quando definida sobre um corpo K não necessariamente será anômala quando definida sobre um corpo $K' \neq K$. As curvas anômalas devem ser evitadas porque o problema do logaritmo discreto para curvas elípticas nessa classe de curvas pode ser resolvido de maneira consideravelmente mais rápida que o normal, comprometendo-se assim a segurança do processo criptográfico. Para mais detalhes sobre a fragilidade do problema do logaritmo discreto para curvas elípticas anômalas, pode ser consultada a referência (WASHINGTON, 2008). Por fim, como já mencionado anteriormente, sempre se deve utilizar um ponto P cuja ordem possua ao menos um fator primo grande, pois, caso a ordem de P possa ser decomposta em fatores primos pequenos, o problema do logaritmo discreto é passível de sofrer ataques com o método de Pohlig-Hellman. Consequentemente, devem-se sempre utilizar curvas elípticas cuja ordem possua ao menos um fator primo grande, pois, caso contrário, de acordo com o teorema de Lagrange, não se pode obter pontos P cuja ordem possua ao menos um fator primo grande.

3.2.6 Assinatura Digital com Curvas Elípticas

De maneira análoga ao algoritmo RSA, as curvas elípticas também podem ser utilizadas em algoritmos de assinatura digital, algo fundamental no âmbito da criptografia assimétrica. Nas seções anteriores, foi apresentada a metodologia de funcionamento de um processo de assinatura digital utilizando-se como base o algoritmo RSA. Analogamente, nesta seção será apresentado o algoritmo *ElGamal* de assinatura digital com curvas elípticas, que utiliza como base a criptografia com curvas elípticas - ECC.

O algoritmo *ElGamal* de assinatura digital com curvas elípticas será apresentado por meio do seguinte exemplo: Supõe-se que Alice deseja enviar para Bob uma mensagem, assinada digitalmente utilizando-se o algoritmo *ElGamal* de assinatura digital com curvas elípticas. Caso a mensagem seja secreta, Alice não deve assinar a própria mensagem, mas sim o seu *hash*, conforme já foi explicitado anteriormente. Caso a mensagem não seja secreta, Alice pode assinar a própria mensagem, e o exemplo seguinte trata apenas deste caso. Primeiramente, Alice escolhe uma curva elíptica E e um corpo finito K, tais que o problema do logaritmo discreto seja difícil de ser resolvido com esses parâmetros. Alice também escolhe um ponto $P \in E$, tal que a ordem de P possua, pelo menos, um fator primo grande. A ordem de P será representada por n. Na prática, escolhe-se um ponto P, tal que sua ordem n seja um número primo grande, sem outros fatores. Esse ponto P é denominado ponto base do algoritmo, de maneira análoga ao algoritmo anterior. Por fim, Alice escolhe um número inteiro a, calcula $Q = aP$ e escolhe também uma função $f : E \longrightarrow \mathbb{Z}$ que relaciona cada ponto da curva E com um número inteiro. Alice divulga então E, K, f, P e Q, e mantém a em segredo. Para enviar a mensagem assinada digitalmente para Bob, Alice inicialmente codifica a mensagem a ser enviada assinada como um número inteiro m, tal que $m \leq n$. Caso $m > n$, Alice deve escolher uma outra curva elíptica, de ordem maior. Em seguida, Alice escolhe um número inteiro k, com $MDC(k, n) = 1$, e calcula $R = kP$. Por fim, Alice calcula $s \equiv k^{-1}(m - af(R))\ mod(n)$ e envia (m, R, s) para Bob, como a mensagem m assinada. Para verificar a autenticidade da assinatura de Alice, Bob calcula $V_1 = f(R)Q + sR$ e $V_2 = mP$. Se $V_1 = V_2$, então Bob considera a assinatura autêntica. Esse procedimento é válido, pois:

$$V_1 = f(R)Q + sR \quad \Rightarrow$$

$$V_1 = f(R)aP + skP$$

Mas como $s \equiv k^{-1}(m - af(R))\ mod(n)$, então pode-se escrever $sk = m - af(R) + zn$, com $z \in \mathbb{Z}$. Então:

$$V_1 = f(R)aP + (m - af(R) + zn)P \quad \Rightarrow$$

$$V_1 = f(R)aP + mP - af(R)P + znP \quad \Rightarrow$$

$$V_1 = mP + znP$$

Mas n é a ordem de P, de forma que $nP = O$. Como O é o elemento neutro da "soma"

de pontos, tem-se:

$$V_1 = mP + zO \quad \Rightarrow$$

$$V_1 = mP \quad \Rightarrow$$

$$V_1 = V_2$$

Dessa forma, para que um espião, por exemplo Eve, possa assinar uma mensagem tentando se passar por Alice, é necessário que Eve conheça a e k para calcular $s \equiv k^{-1}(m - af(R))\ mod(n)$ e $R = kP$, respectivamente. Porém, como $Q = aP$, para se obter a a partir de P e Q, é necessário que Eve resolva o problema do logaritmo discreto para curvas elípticas, que não possui tratamento computacional eficiente caso se façam boas escolhas de E, K e P. O mesmo ocorre quando Eve tenta obter k a partir de R e P, pois $R = kP$. Nesses 2 problemas de logaritmo discreto para curvas elípticas reside a autenticidade da assinatura de Alice, que Bob pode atestar quando $V_1 = V_2$, pois não há maneiras de Eve assinar a mensagem se passando por Alice sem conhecer a e k.

Segue um exemplo numérico a título de ilustração: Supondo-se que Alice deseja enviar uma mensagem $m = 100$, não secreta, assinada digitalmente para Bob. Alice escolhe e publica os parâmetros $E : y^2 = x^3 + 3x + 45$, $K = \mathbb{Z}_{8831}$, $P = (4, 11)$ e f dada por:

$$\begin{cases} f(P = (x, y)) = x \text{ , se } P \neq O \\ f(P) = 0 \text{ , se } P = O \end{cases}$$

que relaciona a cada ponto de E um número inteiro x, que corresponde a sua própria coordenada x. Alice calcula a ordem n do ponto P, que é igual a 4427, e também, secretamente, escolhe um número inteiro $a = 3$, publicando então o ponto $Q = 3P = (413, 1808)$, completando assim a lista de parâmetros a serem publicados. Para enviar a mensagem $m = 100$ assinada digitalmente para Bob, Alice escolhe um inteiro $k = 8$, por exemplo, tal que $MDC(8, 4427) = 1$, e calcula $R = 8P = (5415, 6321)$. Em seguida, Alice calcula:

$$s \equiv 8^{-1} \cdot (100 - 3 \cdot 5415) \equiv 4069\ mod(4427)$$

Alice então envia $(100, (5415, 6321), 4069)$ como a mensagem assinada para Bob. Para verificar a autenticidade da assinatura de Alice, Bob calcula:

$$V_1 = 5415 \cdot (413, 1808) + 4069 \cdot (5415, 6321) = (1296, 8024)$$

$$V_2 = 100 \cdot (4, 11) = (1296, 8024)$$

Como $V_1 = V_2 = (1296, 8024)$, Bob então concluiu que a assinatura de Alice é autêntica.

3.2.7 Comentários sobre Criptografia de Curvas Elípticas

Quando se analisam os algoritmos de ECC, surge o seguinte questionamento: por que utilizar algoritmos de ECC no lugar do já estabelecido algoritmo RSA? Os algoritmos de ECC apresentam várias vantagens em relação ao RSA. Por exemplo, uma grande vantagem dos algoritmos de ECC é a sua grande flexibilidade, pois se pode escolher qual o corpo finito sobre o qual a curva será definida, qual será a curva elíptica utilizada no processo, qual será o ponto base P, etc. Claro que há uma série de restrições a serem respeitadas para se garantir a segurança do algoritmo, porém ainda assim o usuário goza de mais autonomia para a definição dos seus parâmetros, em comparação com o algoritmo RSA. Dessa forma, é muito mais fácil adequar o "tamanho" dos parâmetros necessários para se obter um determinado nível de segurança com a demanda do usuário do algoritmo. Uma outra vantagem da utilização dos algoritmos de ECC é o tamanho pequeno das chaves utilizadas para se obter níveis de segurança semelhantes aos obtidos utilizando-se chaves muito maiores com o RSA. Observando-se a tabela 13, extraída de (BARKER, 2016), pode-se perceber a enorme diferença entre os comprimentos, todos em bits, das chaves requeridas em diversos tipos de algoritmos criptográficos para se obter, aproximadamente, o mesmo nível de segurança.

Tabela 13 – Comparação entre Tamanhos de Chaves com Nível de Segurança Semelhante (BARKER, 2016)

ALGORITMOS DE CRIPTOGRAFIA SIMÉTRICA	RSA	ECC
2TDEA	n = 1024	f = 160-223
3TDEA	n = 2048	f = 224-255
AES-128	n = 3072	f = 256-383
AES-192	n = 7680	f = 384-511
AES-256	n = 15360	f = 512+

As cifras 2TDEA e 3TDEA correspondem ao triploDES, utilizado com 2 e 3 chaves distintas, respectivamente. A coluna com o título RSA apresenta comprimentos do parâmetro $n = pq$ do algoritmo RSA. Já a coluna com o título ECC apresenta intervalos de comprimentos de f, que correspondem à ordem do ponto base P, utilizado nos algoritmos de ECC descritos anteriormente. A diferença entre os valores de n e f que proporcionam o mesmo nível de segurança com os algoritmos RSA e de ECC, respectivamente, é enorme, e isso se traduz em custo computacional para se realizar o processamento desses algoritmos na prática. Portanto, como os parâmetros necessários para implementações seguras de ECC são muito menores que os parâmetros dessas implementações com o RSA, os algoritmos de ECC são cada vez mais usados atualmente.

Como exemplo da utilização cada vez maior de criptografia de curvas elípticas, pode-se citar o aplicativo de troca de mensagens *WhatsApp*. Conforme já citado no capítulo anterior, o *WhatsApp* utiliza a cifra AES com chaves de 256 bits para cifrar as mensagens trocadas por meio da sua plataforma. Porém, a distribuição das chaves secretas do AES entre emissor e receptor é realizada por meio do algoritmo ECDH, de forma que nem mesmo o próprio *WhatsApp* tem acesso às chaves privadas dos usuários e, portanto, também não tem acesso à chave secreta do AES, o que impede que o próprio *WhatsApp* conheça o conteúdo das mensagens. Portanto, mesmo sob determinação de ordem judicial, o *WhatsApp* não possui mecanismos para revelar o conteúdo de mensagens trocadas entre seus usuários utilizando a combinação de criptografia simétrica (AES) e assimétrica (ECDH). Inclusive, o possível desconhecimento do poder judiciário sobre essa impossibilidade já gerou, no passado, ordem de suspensão do aplicativo em todo o território brasileiro, por 48 horas, sob a alegação de que o *WhatsApp* estaria descumprindo decisão judicial referente à entrega de conteúdos de mensagens à justiça. Por fim, sobre a segurança da troca de mensagens utilizando-se o *WhatsApp*, há uma ressalva que deve ser feita: embora o *WhatsApp* não possua mecanismos para conhecer o conteúdo das mensagens trocadas utilizando-se sua criptografia de ponta a ponta, caso o usuário realize *backup* das suas mensagens em algum serviço de armazenamento de dados, como por exemplo o *Google Drive*, essas mensagens transferidas para o *backup* não são protegidas pela criptografia do *WhatsApp*. Para mais detalhes sobre o protocolo utilizado na troca de mensagens pelo *WhatsApp*, pode ser consultado (WHATSAPP, 2017).

A utilização atual da matemática envolvendo curvas elípticas não se restringe ao campo da criptografia. Podem-se citar como exemplos de utilização de curvas elípticas fora do campo da criptografia o algoritmo de fatoração de números inteiros baseado em curvas elípticas, proposto pelo pesquisador H. W. Lenstra Jr. em 1987, e também a utilização de resultados sobre curvas elípticas por A. J. Wiles, na demonstração do último teorema de Fermat, que teve sua versão final publicada em 1995. Por fim, como último exemplo da importância da teoria de curvas elípticas, mesmo fora da atividade criptografia, pode-se citar um dos chamados Problemas do Milênio, conhecido como Conjectura de Birch e Swinnerton-Dyer. Trata-se de uma conjectura que, simplificadamente, estabelece condições necessárias e suficientes para que uma determinada curva elíptica possua infinitos pontos racionais. O *Clay Mathematics Institute*, uma fundação privada que se dedica ao acréscimo e à disseminação do conhecimento matemático pelo mundo, oferece um prêmio de 1 milhão de dólares para quem conseguir resolver qualquer um dos seus Problemas do Milênio. Para maiores detalhes sobre a utilização de curvas elípticas fora do campo da criptografia ou sobre os Problemas do Milênio, podem ser consultados, respectivamente, (WASHINGTON, 2008) e (DEVLIN, 2005).

4 Propostas de Utilização de Criptografia em Sala de Aula

Neste capítulo, serão apresentadas propostas de utilização de criptografia em sala de aula, em turmas do ensino fundamental II, com o objetivo de levar para o universo do aluno algumas atividades básicas sobre esse tema. A escolha de alunos do ensino fundamental se deu devido à sua maior receptividade para jogos e desafios matemáticos de caráter mais lúdico. A matemática e, em particular, a criptografia, quando apresentadas ao aluno em um contexto menos formal de sala de aula e mais ligado a jogos e desafios, tende a despertar maior interesse e curiosidade, sobretudo na faixa etária que abrange os alunos do ensino fundamental II. A divisão das atividades propostas foi relizada conforme o escalonamento de conteúdos constantes na BNCC (BRASIL, 2017), sempre de maneira progressiva em termos de complexidade. Dessa forma, para os alunos do 6º e 7º anos serão propostas uma atividade de cifração e decifração utilizando cifras de deslocamento e uma atividade de cifração e decifração utilizando cifras de transposição por várias linhas. Para os alunos do 8º ano será proposta uma atividade de cifração e decifração utilizando a cifra de Vigenère. Por fim, para os alunos do 9º ano serão propostas uma atividade de cifração e decifração utilizando o RSA e uma atividade de par ou ímpar por telefone, também utilizando o RSA.

4.1 Proposta 1 - Cifra de Deslocamento

A proposta 1 terá como foco a utilização prática de cifras de deslocamento pelos alunos do 6º e 7º anos do ensino fundamental, tanto para cifrar e decifrar mensagens, quanto na tentativa de criptoanálise da cifra.

4.1.1 Aula 1 - Cifração e Decifração com Cifra de Deslocamento

O primeiro passo para a utilização da cifra de deslocamento em sala de aula é a divisão dos alunos em grupos, para que haja interação entre eles nos grupos. Grupos de 5 ou 6 alunos parecem adequados para salas de aula com 40 alunos, sendo essa quantidade de alunos por grupo adaptável para a realidade de cada sala de aula e de cada ano letivo. Após a divisão dos grupos, faz-se necessário mostrar aos alunos exemplos de cifração e decifração com cifras de deslocamento com diferentes alfabetos deslocados. É importante estabelecer diretrizes para a utilização da cifra, como por exemplo a não utilização de caracteres acentuados ou com sinais gráficos distintos

das letras do alfabeto tradicional, como "ç", "ã" etc. Caso a mensagem a ser cifrada possua algum desses caracteres, eles devem ser substituídos por letras convencionais, o que, na maioria dos casos, não prejudica o entendimento da mensagem. Outra diretriz importante a ser estabelecida é a necessidade de ausência de espaços entre as palavras no texto a ser cifrado, de forma que os alunos devem escrever tanto o texto claro como o texto cifrado com as palavras de maneira concatenada, sem espaços, para dificultar a sua criptoanálise. Também, não devem ser diferenciados caracteres maiúsculos e minúsculos na utilização da cifra de deslocamento, também com o intuito de dificultar a sua criptoanálise.

Após a divisão dos grupos e explicações sobre o funcionamento da cifra e exemplos, podem-se formar pares de grupos e pedir para que cada par de grupos combine secretamente um alfabeto deslocado, que irá ser utilizado na cifração e decifração das mensagens trocadas entre esses grupos. Por fim, pode-se solicitar que cada grupo escolha uma mensagem, utilize o alfabeto deslocado combinado previamente para cifrar a mensagem e envie a mensagem cifrada para o outro grupo com o qual combinou o alfabeto deslocado, recebendo deste grupo uma mensagem cifrada da mesma forma. Após os grupos terem, cada um, enviado e recebido uma mensagem cifrada, pede-se que eles decifrem a mensagem recebida, utilizando o alfabeto deslocado combinado, e leiam a mensagem. Com isso, pode-se finalizar a aula 1.

Seguem alguns exemplos de mensagens cifradas com a cifra de deslocamento, utilizando-se alfabetos deslocados distintos:

- Mensagem Clara: *nossomosaturmadobarulho*
  Alfabeto Deslocado: *HIJKLMNOPQRSTUVWXYZABCDEFG*
  Mensagem Cifrada: *UVZZVTVZHABYTHKVIHYBSOV*

- Mensagem Clara: *naoacreditoemdoendespoiselesmentemmuito*
  Alfabeto Deslocado: *NOPQRSTUVWXYZABCDEFGHIJKLM*
  Mensagem Cifrada: *ANBNPERQVGBRZQBRAQRFCBVFRYRFZRAGRZZHVGB*

4.1.2 Aula 2 - Criptoanálise da Cifra de Deslocamento

Na aula 2, inicialmente pode-se relembrar rapidamente aos alunos como funcionam as cifras de deslocamento e solicitar que os grupos se juntem novamente. Nesse momento, pode-se solicitar aos grupos que divulguem as suas mensagens cifradas na aula 1, sem divulgar o alfabeto deslocado utilizado na cifração. Com isso, pode-se encorajar os outros grupos a tentarem decifrar as mensagens cifradas. Essa etapa dá aos alunos uma ideia sobre a segurança da cifra, dada sua maior facilidade ou dificuldade em ser quebrada. Não são necessárias técnicas elaboradas para se realizar a

criptoanálise da cifra de deslocamento, apenas tentativa e erro. Dessa forma, os alunos provavelmente conseguirão decifrar as mensagens enviadas pelos demais grupos sem grandes dificuldades. Pode-se encorajá-los a construir uma tabela com as 25 possibilidades de alfabetos deslocados distintos, e realizar a tentativa de cada um deles ordenadamente, até que um texto claro seja encontrado. Esse procedimento permite que eles percebam que, quanto maior for a quantidade de possibilidades de alfabetos deslocados que uma cifra possui, mais difícil será quebrá-la por meio do ataque da força bruta, isto é, por sucessivas tentativas. Com isso, pode-se finalizar a aula 2.

4.2 Proposta 2 - Cifra de Transposição por Linhas

A proposta 2 terá como foco a utilização prática de cifras de transposição por várias linhas pelos alunos do 6º e 7º anos do ensino fundamental, tanto para cifrar e decifrar mensagens, quanto na tentativa de criptoanálise da cifra.

4.2.1 Aula 1 - Cifração e Decifração com Cifra de Transposição por Linhas

Os primeiros passos para se utilizarem cifras de transposição em sala de aula são semelhantes aos adotados na utilização da cifra de deslocamento: divisão da sala em grupos, apresentação da metodologia de funcionamento da cifra, realização de exemplos e, por fim, solicitar que os grupos formem pares e combinem entre si qual será o número de linhas utilizadas na cifra de transposição. Novamente, é importante estabelecer as mesmas diretrizes para a utilização da cifra: não utilizar caracteres acentuados ou com sinais gráficos distintos das letras do alfabeto tradicional, não utilizar espaços entre as palavras no texto a ser cifrado e não diferenciar caracteres maiúsculos e minúsculos na utilização da cifra de transposição.

Em seguida, pode-se solicitar que cada grupo escolha uma mensagem, utilize o número de linhas combinado previamente para cifrar a mensagem e envie a mensagem cifrada para o outro grupo com o qual combinou o número de linhas, recebendo deste grupo uma mensagem cifrada da mesma forma. Após os grupos terem, cada um, enviado e recebido uma mensagem cifrada, pede-se que eles decifrem a mensagem recebida, utilizando o número de linhas combinado e leiam a mensagem. Com isso, pode-se finalizar a aula 1.

Seguem alguns exemplos de mensagens cifradas com a cifra de transposição por linhas, utilizando-se diferentes quantidades de linhas:

- Mensagem Clara: *emterradesacicalcajeansdaparadois*

Transposição com 3 Linhas:

e	*e*	*a*	*s*	*i*	*l*	*j*	*n*	*a*	*r*	*o*
m	*r*	*d*	*a*	*c*	*c*	*e*	*s*	*p*	*a*	*i*
t	*r*	*e*	*c*	*a*	*a*	*a*	*d*	*a*	*d*	*s*

Mensagem Cifrada: *EEASILJNAROMRDACCESPAITRECAAADADS*

- Mensagem Clara: *piratariaecrimeportantonaosedeveroubarnavios*

  Transposição com 5 Linhas:

p	*a*	*c*	*p*	*n*	*o*	*v*	*b*	*v*
i	*r*	*r*	*o*	*t*	*s*	*e*	*a*	*i*
r	*i*	*i*	*r*	*o*	*e*	*r*	*r*	*o*
a	*a*	*m*	*t*	*n*	*d*	*o*	*n*	*s*
t	*e*	*e*	*a*	*a*	*e*	*u*	*a*	

  Mensagem Cifrada: *PACPNOVBVIRROTSEAIRIIROERROAAMTND ONSTEEAAEUA*

4.2.2 Aula 2 - Criptoanálise da Cifra de Transposição por Linhas

Na aula 2, novamente, pode-se relembrar rapidamente aos alunos como funcionam as cifras de transposição por linhas e solicitar que os grupos se juntem novamente. Nesse momento, pode-se solicitar aos grupos que divulguem as suas mensagens cifradas na aula 1, sem divulgar o número de linhas utilizadas na transposição. Com isso, pode-se encorajar os outros grupos a tentarem decifrar as mensagens cifradas. Novamente, não são necessárias técnicas elaboradas para se realizar a criptoanálise da cifra de transposição por linhas, apenas tentativa e erro. Dessa forma, os alunos provavelmente conseguirão decifrar as mensagens enviadas pelos demais grupos sem grandes dificuldades. A técnica de criptoanálise para essa cifra consiste em se contar quantas letras tem a mensagem cifrada e dividir essas letras em linhas, de acordo com a quantidade de linhas que se deseja testar. Caso a quantidade de linhas esteja correta, ao se realizar a leitura das colunas de letras surgirá a mensagem clara. Logo os alunos perceberão que a criptoanálise da cifra de transposição por linhas é mais fácil que a criptoanálise da cifra de deslocamento, desde que o número de linhas utilizadas na transposição seja menor que 26, o que gera uma quantidade de tentativas necessárias menor que no caso da cifra de deslocamento. Com isso, pode-se finalizar a aula 2.

4.3 Proposta 3 - Cifra de Vigenère

A proposta 3 terá como foco a utilização prática da cifra de Vigenére pelos alunos do 8º ano, tanto para cifrar e decifrar mensagens, quanto na tentativa de criptoanálise da cifra. Toda a teoria necessária para a utilização da cifra foi apresentada nas subseções 1.2.4 e 1.2.5 do capítulo 1, de forma que, neste capítulo, a abordagem será essencialmente prática.

4.3.1 Aula 1 - Cifração e Decifração com a Cifra de Vigenère

Analogamente às propostas anteriores, inicialmente deve-se realizar a divisão da sala em grupos, apresentar a metodologia de funcionamento da cifra de Vigenère, seguida de exemplos com diferentes palavras-chave e, por fim, solicitar que os grupos formem pares e combinem entre si qual será a palavra-chave utilizada na cifra. Mais uma vez, é importante estabelecer as mesmas diretrizes para a utilização das cifras anteriores: não utilizar caracteres acentuados ou com sinais gráficos distintos das letras do alfabeto tradicional, não utilizar espaços entre as palavras no texto a ser cifrado e não diferenciar caracteres maiúsculos e minúsculos na utilização da cifra de Vigenère.

Em seguida, pode-se solicitar que cada grupo escolha uma mensagem, utilize a palavra-chave combinada previamente para cifrar a mensagem com a cifra de Vigenère, e envie a mensagem cifrada para o outro grupo com o qual combinou a palavra-chave, recebendo deste grupo uma mensagem cifrada com a mesma palavra-chave. Após os grupos terem, cada um, enviado e recebido uma mensagem cifrada, pode-se pedir que eles decifrem a mensagem recebida utilizando-se a palavra-chave combinada, e leiam a mensagem. Também, pode-se solicitar aos grupos que divulguem as suas mensagens cifradas, sem divulgar a palavra-chave, encorajando os outros grupos a tentar decifrar essas mensagens. Nessa etapa, provavelmente os alunos não serão capazes de decifrar as mensagens cifradas pelos outros grupos. Aproveitando-se desse momento de frustração, pode-se propor uma espécie de desafio entre o professor e a turma. Pode-se solicitar que a turma toda combine secretamente uma palavra-chave de, no máximo, 4 letras (para facilitar a criptoanálise), e que cada aluno escreva uma frase para compor a mensagem final, sem o conhecimento do professor. Os alunos realizam então a concatenação de todas as frases, sem espaços ou pontuação, obtendo-se assim uma grande mensagem a ser cifrada. Em seguida, os alunos devem utilizar a cifra de Vigenère para cifrar essa mensagem com a palavra-chave combinada entre si, e entregar a mensagem cifrada para o professor, que deverá tentar descobrir qual é a mensagem que os alunos escreveram. Nesse ponto, pode-se finalizar a aula 1, com a promessa de se trazer a mensagem devidamente decifrada na aula seguinte.

Seguem alguns exemplos de mensagens cifradas com a cifra de Vigenère, utilizando-

se palavras-chave distintas:

- Mensagem Clara: sevocesquiserembrincardepiqueescondenahoradorecreiopodemosnosencontrarnopatio

  Palavra-Chave: RECREIO

  Mensagem Cifrada: JIXFGMGHYKJIZSDFTZRKOIHGGMYIVIUTSVRVRCYSZOUSTVGZSZSRFHMAFWPFWMBTSPKVIFESRRXQC

- Mensagem Clara: arespostadaprimeiraquestaodaprovadematematicaeletradedaultimaquestaoeletrab

  Palavra-Chave: ZOROASTRO

  Mensagem Cifrada: ZFVGPGLKOCOGFIEXZFZELSSLTFRZDICVSWVAZHVAALBTODZVHRSWVRZICHIETHIDGKOOWEVHQOS

4.3.2 Aula 2 - Criptoanálise da Cifra de Vigenère

Como ao final da aula 1 foi estabelecido um desafio entre professor e alunos, em que o professor deveria decifrar uma mensagem cifrada pelos alunos com a cifra de Vigenère, na aula 2 devem ser apresentadas as técnicas de criptoanálise da cifra. Para a utilização de tais técnicas em um intervalo de tempo compatível com uma aula do 8º ano, é necessário que se utilize um computador e alguns *softwares* específicos, conforme demonstrado na subseção 1.2.5 do capítulo 1. Como é grande o número de escolas que não dispõem de um laboratório de informática para atender uma turma de 40 alunos, e também há a possibilidade de que grande parte dos alunos não esteja muito familiarizada com os *softwares* utilizados na criptoanálise da cifra, a sugestão para essa aula é que a criptoanálise fique centrada na figura do professor, e que este projete a tela do computador que utilizará para que os alunos possam acompanhar o processo de criptoanálise. Utilizando as técnicas descritas na subseção 1.2.5, o professor pode obter a palavra-chave e decifrar a mensagem, expondo seu conteúdo para os alunos, que poderão confirmar se a mensagem decifrada, de fato, está correta. Ao longo do processo de criptoanálise, o professor pode explicar aos alunos as técnicas utilizadas, desde a procura pelo comprimento da palavra-chave, até a resolução dos problemas de substituição monoalfabética com a utilização da tabela de frequências da língua portuguesa. Ao final da decifração, pode-se encerrar a aula 2. Apesar de essa abordagem não possibilitar que o aluno seja um participante ativo no processo de criptoanálise, optou-se por sugeri-la devido ao fato de ser uma abordagem mais simples de ser aplicada em um universo heterogêneo de salas de aula existente nas escolas do Brasil.

4.4 Proposta 4 - Algoritmo RSA

A proposta 4 terá como foco a utilização prática do algoritmo RSA por alunos do 9º ano, tanto na cifração e decifração de mensagens, quanto na criptoanálise do algoritmo. Toda a teoria necessária para a utilização da cifra foi apresentada na subseção 3.1.1 do capítulo 3, de forma que, neste capítulo, a abordagem será essencialmente prática.

4.4.1 Aula 1 - Introdução à Aritmética Modular

Para a utilização do algoritmo RSA, é necessário que os alunos realizem operações *modulo n*. Como a aritmética modular não faz parte do conteúdo ministrado no ensino básico brasileiro, sugere-se que, antes de se realizar a apresentação do algoritmo, seja ministrada uma aula básica sobre aritmética modular, introduzindo-se o conceito de congruência *modulo n* a partir dos conceitos de divisão euclidiana, que os alunos do 9º ano já conhecem. Também, como no RSA é necessário que os alunos encontrem o inverso multiplicativo de um número *modulo* $\phi(n)$ para gerar o par de chaves, sugere-se que sejam apresentados o algoritmo de Euclides e a teoria de resolução de equações diofantinas lineares de 1º grau. Caso necessário, o professor pode utilizar mais uma aula para a apresentação desses assuntos. Com isso, os alunos estarão aptos a realizarem as operações necessárias para o pleno funcionamento do algoritmo RSA. Caso o professor não queira se aprofundar tanto na aritmética modular, ele pode sugerir que os alunos encontrem o inverso multiplicativo por tentativas, pois os valores de n que serão utilizados nos exemplos de aplicação serão consideravelmente pequenos. Caso o aluno tenha dificuldade em encontrar o inverso multiplicativo por tentativas, o professor pode ajudá-lo nessa tarefa. Dessa forma, pode-se encerrar a aula 1.

4.4.2 Aula 2 - Cifração e Decifração com o RSA

O primeiro passo para a utilização do algoritmo RSA em sala de aula é, novamente, a divisão dos alunos em grupos. Após a divisão dos grupos, faz-se necessário mostrar aos alunos exemplos de codificação da mensagem textual em valor numérico, exemplos de geração do par de chaves pública e privada e exemplos de cifração e decifração da mensagem numérica previamente codificada. Para a codificação da mensagem textual em um valor numérico, sugere-se a adoção da tabela ASCII, apresentada na subseção 3.1.1. Nesse caso, podem-se utilizar os espaços em branco da mensagem, não sendo necessário escrevê-la com as letras todas concatenadas, e também podem-se utilizar as letras maiúsculas e os sinais de pontuação, pois a tabela ASCII codifica esses espaços em branco, as letras maiúsculas e a pontuação. Porém, continua sendo necessário não se utilizarem caracteres acentuados e o "ç", por exemplo.

Após a divisão dos grupos e explicações sobre o funcionamento do algoritmo, pode-se solicitar que cada grupo escolha uma palavra ou frase curta para ser a mensagem, e a codifique utilizando a tabela ASCII em um valor numérico. Em seguida, pode-se solicitar que cada grupo gere seu par de chaves pública e privada, escolhendo-se os respectivos parâmetros. Nesse ponto, sugere-se que os alunos possam fazer uso de calculadoras, para facilitar os cálculos. Caso os alunos não disponham desse recurso e necessitem realizar os cálculos manualmente, sugere-se limitar a escolha dos números primos p e q em valores menores que 20. Após a geração dos pares de chaves pública e privada, pode-se solicitar que os grupos formem pares e troquem mensagens cifradas entre si. Nesse momento, pode-se enfatizar aos grupos que não foi necessária a combinação prévia de qualquer chave secreta para a troca de mensagens cifradas. Assim, mostra-se aos alunos a verdadeira essência da criptografia assimétrica quando comparada com a criptografia simétrica. Por fim, solicita-se que os grupos decifrem as mensagens recebidas utilizando para isso suas chaves privadas. Em seguida, solicita-se que os grupos utilizem a tabela ASCII para decodificar o valor numérico obtido em letras e, por fim, possam ler o conteúdo da mensagem recebida. Neste ponto, pode-se encerrar a aula 2.

Seguem alguns exemplos de mensagens codificadas com a tabela ASCII e cifradas com o algoritmo RSA, utilizando-se como parâmetros de geração da chave pública os números $p = 13$, $q = 17$ e $e = 5$:

- Mensagem Clara: Dromedario

  Mensagem Codificada: 068114111109101100097114105111

  Mensagem Codificada Separada em Blocos: 068 114 111 109 101 100 097 114 105 111

  Chave Pública do Grupo Receptor: *(221,5)*

  Blocos de Mensagem Cifrados com o RSA: *204 173 076 096 186 172 054 173 209 076*

- Mensagem Clara: A casa caiu!

  Mensagem Codificada: 065032099097115097032099097105117033

  Mensagem Codificada Separada em Blocos: 065 032 099 097 115 097 032 099 097 105 117 033

  Chave Pública do Grupo Receptor: *(221,5)*

  Blocos de Mensagem Cifrados com o RSA: *182 002 216 054 098 054 002 216 054 209 104 050*

4.4.3 Aula 3 - Criptoanálise do Algoritmo RSA

A criptoanálise do algoritmo RSA consiste, basicamente, em se encontrar o inverso multiplicativo do número e *modulo* $\phi(n)$. Para isso, é preciso calcular $\phi(n) = (p-1)(q-1)$. Portanto, o primeiro passo necessário para se realizar a criptoanálise do RSA é encontrar a fatoração de $n = pq$. Em seguida, encontra-se $d = e^{-1}\ mod(\phi(n))$ e se pode decifrar a mensagem enviada com a chave pública (n, e). Pode-se sugerir que os grupos de alunos, que estavam organizados em pares para troca de mensagens, se misturem, gerando novos pares de grupos. Cada grupo divulga para seu novo par qual é a sua chave pública (n, e) e qual foi a mensagem recebida na etapa anterior da atividade, cifrada com essa chave pública, e desafia o grupo a descobrir qual o conteúdo da mensagem. Os grupos devem então fatorar n, calcular $\phi(n)$, encontrar $d = e^{-1}\ mod(\phi(n))$ e realizar a decifração da mensagem. Como os valores de n utilizados são pequenos, os alunos não devem encontrar grandes problemas na sua fatoração. Caso o grupo tenha dificuldades para encontrar o valor de d, pode ser solicitada ajuda do professor nessa etapa. Dessa forma, encoraja-se que os próprios alunos realizem a criptoanálise e percebam que o algoritmo se torna mais forte à medida que se aumenta o valor de n, dificultando assim a sua fatoração. Neste ponto, pode-se encerrar a aula 3.

4.5 Proposta 5 - Par ou Ímpar por Telefone com o RSA

A proposta 5 terá como foco a utilização do algoritmo RSA para se viabilizar um jogo de par ou ímpar por telefone, de maneira que nenhuma das partes possa trapacear. Essa proposta não será dividida em aulas, apenas será apresentada a sugestão de como propor a brincadeira aos alunos. Essa aplicação do algoritmo RSA foi baseada na sugestão contida em (MORAIS, 2019).

Propõe-se que os alunos Alice e Bob realizem um jogo de par ou ímpar pelo telefone, sem a possibilidade de que qualquer um deles trapaceie. Para isso, ambos devem possuir uma chave pública, sendo (n_A, e_A) a chave pública de Alice e (n_B, e_B) a chave pública de Bob. Então, tanto Alice quanto Bob escolhem uma mensagem textual que identifique qual será o número escolhido no par ou ímpar. Por exemplo, Alice pode escolher a mensagem "raiz quadrada de 36" para indicar que seu número escolhido é 6. Após a escolha das mensagens textuais, estas mensagens devem ser codificadas em valor numérico (utilizando o ASCII, por exemplo), de forma que as mensagens de Alice e Bob, já codificadas, serão denominada m_A e m_B, respectivamente. Então, Alice calcula $C_A \equiv (m_A)^{e_A}\ mod(n_A)$ e envia C_A para Bob. Analogamente, Bob calcula $C_B \equiv (m_B)^{e_B}\ mod(n_B)$ e envia C_B para Alice. Nesse momento, eles devem decidir quem será "par" e quem será "ímpar" no jogo. Pode-se observar que essa escolha será

aleatória, pois Alice não conhece a mensagem m_B com o número de Bob, nem Bob conhece a mensagem m_A com o número de Alice, ambos conhecem apenas C_B e C_A, respectivamente, que precisam das respectivas chaves privadas para serem decifrados. Portanto, a escolha do "par" ou "ímpar" é perfeitamente justa. Por fim, para se decidir quem ganhou o jogo, Alice envia m_A para Bob, e Bob envia m_B para Alice, de forma que eles ficam sabendo qual foi o número escolhido pelo adversário e podem verificar quem foi o vencedor. Supondo-se que Bob, após receber m_A de Alice, deseje trapacear e enviar para ela uma mensagem m_B^* diferente da sua escolha prévia m_B. Nesse caso, Alice pode facilmente identificar a trapaça realizando a operação $C_B^* \equiv (m_B^*)^{e_B} \ mod(n_B)$ e verificando que $C_B^* \neq C_B$, o que mostra que Bob alterou a sua escolha prévia m_B. Portanto, procedendo-se dessa maneira, pode-se promover um jogo de par ou ímpar perfeitamente seguro e realizado por telefone, por exemplo, sem a necessidade de Alice e Bob estarem fisicamente presentes no mesmo lugar.

5 Conclusões

Por fim, após a apresentação de alguns tópicos de criptografia e seu desenvolvimento ao longo da história, de algumas cifras de criptografia simétrica e de outras de criptografia assimétrica, bem como de algumas propostas de aplicação de criptografia em sala de aula, é preciso ressaltar que o assunto é extremamente rico e vasto, e este trabalho não teve a pretensão de cobri-lo de maneira abrangente. O leitor interessado encontrará vasta bibliografia sobre o assunto, que tem se tornado cada vez mais relevante na sociedade, sobretudo nas aplicações mais modernas de tecnologia da informação. Porém, apesar de vasta, grande parte dessa bibliografia se encontra escrita em língua inglesa, sobretudo os títulos que tratam de aplicações mais atuais. Com este trabalho, espera-se contribuir, mesmo que com uma ínfima parte, para a maior divulgação em língua portuguesa da criptografia na sociedade, sobretudo nos círculos escolares. Talvez a abordagem de alguns tópicos mais básicos de criptografia na escola, principalmente no ensino fundamental, possam aumentar a visibilidade e o interesse dos alunos por essa área. Foi com esse intuito que foram propostas as atividades do capítulo 4.

Como se tem observado uma divulgação cada vez maior das olimpíadas de matemática nas escolas brasileiras, acompanhada do crescimento do número de participantes e medalhistas, talvez num futuro próximo possam ser criadas no Brasil competições semelhantes na área de criptografia. Com certeza, essas competições seriam muito bem recebidas pelos alunos, sempre ávidos por desafios. Atualmente, há uma olimpíada internacional de criptografia, chamada *NSUCRYPTO*, organizada pela *Novosibirsk State University*, na Rússia, cuja participação é aberta ao público em geral, por meio da internet. Talvez, se houvesse mais divulgação e algum estímulo para que os estudantes brasileiros participassem dessa competição, haveria cada vez mais pessoas interessadas em estudar criptografia no país. Para mais informações sobre a *NSUCRYPTO*, pode-se consultar o *site* da olimpíada na internet: nsucrypto.nsu.ru.

Finalmente, encerrando este trabalho, fica a reflexão sobre a enorme importância da criptografia como instrumento de soberania nacional. Na visão do autor deste trabalho, um país que se propõe a manter o caráter secreto no trânsito das suas informações, sobretudo as mais sensíveis, não deveria importar tecnologia criptográfica desenvolvida por outros países, sob o risco de ser espionado pelos desenvolvedores. O desenvolvimento de ciência criptográfica nacional, desde o fomento da pesquisa básica, até o projeto e implementação de aplicações práticas em *hardware* e *software*, deveriam ser, novamente na visão do autor deste trabalho, uma pauta prioritária na agenda dos investimentos governamentais em segurança nacional.

Referências

AGRAWAL, M.; KAYAL, N.; SAXENA, N. Primes is in p. *Ann. of Math*, v. 2, p. 781–793, 2002. Citado na página 73.

BARKER, E. *NIST Special Publication 800-57 Recommendation for Key Management – Part 1: General. Rev. 4.* 2016. Citado 2 vezes nas páginas 10 e 96.

BRASIL. *Base Nacional Comum Curricular (BNCC)*. 2017. Brasília: MEC/Secretaria de Educação Básica. Citado 2 vezes nas páginas 14 e 98.

COUTINHO, S. *Números Inteiros e Criptografia RSA*. Rio de Janeiro, Brasil.: IMPA, 2000. ISBN 9788524401244. Citado 5 vezes nas páginas 16, 20, 41, 73 e 74.

DEVLIN, K. *The Millennium Problems: The Seven Greatest Unsolved Mathematical Puzzles of Our Time*. London - UK: Granta, 2005. Citado na página 97.

DOOLEY, J. F. *History of Cryptography and Cryptanalysis*. Galesburg, IL, USA.: Springer International Publishing, 2018. ISBN 9783319904436. Citado 2 vezes nas páginas 13 e 16.

GLEICK, J. *The Information: A History, a Theory, a Flood*. New York, USA.: Vintage Books, 2012. ISBN 9781400096237. Citado na página 17.

HEFEZ A.; VILLELA, M. L. T. *Códigos Corretores de Erros*. Rio de Janeiro, Brasil.: IMPA, 2008. ISBN 9788524401695. Citado na página 17.

ICP-BRASIL. *Padrões e Algoritmos Criptográficos da ICP-Brasil - DOC ICP-01.01 Versão 4.1*. 2018. Disponível em: <https://www.iti.gov.br/images/repositorio/legislacao/documentos-principais/DOC-ICP-01.01_-_versao_4.1_PADROES_E_ALGORITMOS_CRIPTOGRAFICOS_DA_ICP-BRASIL.pdf>. Citado 2 vezes nas páginas 40 e 45.

JUNIOR, E. A. D. C. *Curvas Elípticas e Aplicações a Criptografia*. Dissertação (Dissertação de Mestrado em Matemática) — Universidade de Brasília - UnB, Brasília, Brasil., 2003. Citado 3 vezes nas páginas 76, 79 e 87.

JUNIOR, E. C. V. B. *Introdução a Sistemas Criptográficos e o uso de Geradores de Sequências de Números Aleatórios e Pseudo-Aleatórios*. Dissertação (Dissertação de Mestrado Profissional em Matemática - PROFMAT) — Universidade de Brasília - UnB, Brasília, Brasil., 2014. Citado na página 33.

KAHN, D. *The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet*. New York, NY: Scribner, 1996. Citado 2 vezes nas páginas 16 e 18.

MARTINEZ, F. et al. *Teoria dos números: um passeio com primos e outros números familiares pelo mundo inteiro*. Rio de Janeiro - RJ: IMPA, 2010. (Projeto Euclides). Citado na página 79.

MENEZES, A. J.; OORSCHOT, P. C. V.; VANSTONE, S. A. *Handbook of Applied Cryptography*. [S.l.]: CRC Press, 2001. Citado 4 vezes nas páginas 56, 91, 92 e 93.

MORAIS, R. *Curiosidades da Matemática (para não matemáticos)*. Brasília, DF: Ideal, 2019. Citado na página 106.

NIST. *Data Encryption Standard (DES)*. 1999. FIPS Publication 46-3. Citado 3 vezes nas páginas 35, 36 e 46.

NIST. *Advanced Encryption Standard*. 2001. FIPS Publication 197. Citado 2 vezes nas páginas 36 e 65.

PAAR C.; PELZL, J. *Understanding Cryptography*. Berlin, Germany.: Springer-Verlag Berlin Heidelberg, 2010. Citado 5 vezes nas páginas 34, 43, 55, 56 e 73.

QUARESMA, P. *Frequency Analysis of the Portuguese Language*. Coimbra, Portugal, 2008. Citado 4 vezes nas páginas 10, 29, 30 e 31.

SALOMON, D. *Coding for Data and Computer Communications*. New York, USA.: Springer Science & Business Media, 2006. Citado na página 36.

SHANNON, C. E. A mathematical theory of communication. *The Bell System Technical Journal*, v. 27, n. 3, p. 379–423, 1948. Citado na página 17.

SHANNON, C. E. Communication theory of secrecy systems. *The Bell System Technical Journal*, v. 28, n. 4, p. 656–715, 1949. Citado na página 17.

SHOKRANIAN, S. *Uma Introdução à teoria dos números*. Rio de Janeiro: Ciencia Moderna, 2008. Citado na página 73.

SHOKRANIAN, S. *Criptografia Para Iniciantes*. Rio de Janeiro, Brasil.: CIENCIA MODERNA, 2012. ISBN 9788539902750. Citado na página 17.

SHOR, P. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, v. 26, n. 5, p. 1484–1509, 1997. Citado 2 vezes nas páginas 41 e 74.

SILVERMAN, J. *The Arithmetic of Elliptic Curves*. New York - USA: Springer New York, 2013. (Graduate Texts in Mathematics). Citado na página 87.

SINGH, S. *O Livro dos Códigos*. Rio de Janeiro, Brasil.: RECORD, 2011. ISBN 9788501055989. Citado 10 vezes nas páginas 16, 18, 20, 22, 23, 25, 32, 33, 35 e 37.

STINSON, D. *Cryptography: Theory and Practice, Third Edition*. Boca Raton, FL, USA.: Taylor & Francis, 2006. (Discrete Mathematics and Its Applications). ISBN 9781584885085. Citado 5 vezes nas páginas 34, 43, 73, 92 e 93.

TRAPPE W.; WASHINGTON, L. *Introduction to Cryptography: With Coding Theory, Second Edition*. Upper Saddle River, NJ, USA.: Pearson Prentice Hall, 2006. (Featured Titles for Cryptography Series). ISBN 9780131862395. Citado na página 43.

WASHINGTON, L. *Elliptic Curves: Number Theory and Cryptography, Second Edition*. Boca Raton, FL, USA.: CRC Press, 2008. (Discrete Mathematics and Its Applications). ISBN 9781420071474. Citado 6 vezes nas páginas 76, 79, 88, 91, 93 e 97.

WHATSAPP. *WhatsApp Encryption Overview - Technical White Paper*. California - USA, 2017. Citado 2 vezes nas páginas 66 e 97.