UNIVERSIDADE FEDERAL DE MINAS GERAIS
DEPARTAMENTO DE MATEMÁTICA
ESPECIALIZAÇÃO EM MATEMÁTICA PARA PROFESSORES

MONOGRAFIA
CRIPTOGRAFIA RSA

ANDRÊZA GRAZIELE SANTOS PEDRA
ORIENTADOR: PAULO ANTÔNIO FONSECA MACHADO

BELO HORIZONTE- 2011

ANDRÊZA GRAZIELE SANTOS PEDRA

CRIPTOGRAFIA RSA

Trabalho final do Curso de Especialização em Matemática para Professores.

Belo Horizonte – 2011

CRIPTOGRAFIA RSA

COMISSÃO EXAMINADORA

_____

Professor Paulo Antônio Fonseca Machado
Orientador

_____

Professor Israel Vainsencher
Examinador

_____

Professor Jeroen Antonius Maria vam de Graaf
Examinador

Belo Horizonte, 07 de novembro de 2011

Agradecimentos

Agradeço a Deus, pois sem Ele nada seria possível. À minha mãe Maria das Graças Santos Pedra, pela confiança em mim depositada; aos meus irmãos, principalmente Polyana, pelo apoio nas horas mais difíceis, dando-me força e coragem para não desistir.

Ao meu pai José Pedra que, mesmo não estando presente, dedico a ele este trabalho.

Agradeço ao meu orientador Paulo Antônio Fonseca Machado, pela paciência, por ter me apoiado e acreditado que este trabalho era possível realizar.

A engenhosidade não pode arquitetar uma escrita secreta que a própria engenhosidade humana não possa resolver.

(Edgar Allan Poe, 1988)

SUMÁRIO

INTRODUÇÃO

A *criptologia* é uma ciência muito antiga, que vem sendo utilizada desde o sistema de escrita *hieroglífica* dos egípcios, há quase quatro mil anos. Durante o decorrer dos séculos, generais, reis, rainhas, militares e outros, buscavam formas eficientes de comunicação, de comandar seus exércitos e de governar seus países de forma sigilosa, sem revelar segredos e estratégias às forças inimigas.

Grandes matemáticos e *criptoanalistas* desenvolveram *códigos*, *cifras* e técnicas que permitiram a difícil compreensão de uma mensagem originalmente escrita com clareza, de forma que apenas o destinatário a decifrasse e a compreendesse.

> *Já se falou que a Primeira Guerra Mundial foi a guerra dos químicos, devido ao emprego, pela primeira vez, do gás mostarda e do cloro, que a Segunda Guerra Mundial foi a guerra dos físicos devido à bomba atômica. De modo semelhante se fala que uma Terceira Guerra Mundial seria a guerra dos matemáticos, pois os matemáticos terão o controle sobre a próxima grande arma de guerra, a informação. Os matemáticos têm sido responsáveis pelo desenvolvimento dos códigos usados atualmente para a proteção das informações militares. E não nos surpreende que os matemáticos também estejam na linha de frente da batalha para tentar decifrar esses códigos. (SINGH, 2007, p.13).*

O uso da *criptologia* foi fundamental durante a Segunda Guerra Mundial para a quebra dos *códigos* alemão e japonês e o sucesso dos Aliados. Contudo, durante as grandes guerras, ocorreram modificações em amplos aspectos, o que propiciou uma grande evolução. Portanto, este trabalho descreve como o avanço tecnológico ajudou no uso da *criptologia* no decorrer das grandes guerras. Contextualizando a forma evolutiva que se deu através dos anos e o surgimento de vários sistemas de *criptografia*, em que alguns se mostraram mais seguros que outros. O presente estudo aborda os dois sistemas *criptográficos*: o sistema de *criptografia simétrico* e o sistema *criptografia assimétrico*; enfatizando a criptografia assimétrica RSA[1],que contém a descrição do método RSA.

[1]RSA é um algoritmo de criptografia de dados, que deve o seu nome a três pesquisadores do Instituto MIT, Ronald Rivest, Adi Shamir e Leonard Adleman

CAPÍTULO 1

1.1 DEFINIÇÕES

A *criptografia* é a arte ou a ciência de escrever em *cifra* ou em *código*; em outras palavras, é um conjunto de técnicas que permitem tornar incompreensível uma mensagem originalmente escrita com clareza, de forma a permitir normalmente que apenas o destinatário a decifre e a compreenda. Quase sempre a decodificação requer o conhecimento de uma *chave*, uma informação secreta disponível ao destinatário. Assim, o processo de aplicar algum tipo de *criptografia* a uma mensagem é chamado de *cifrar* a mensagem ou *codificação*. Já o processo inverso, ou seja, o de voltar à mensagem original é chamado de *decifrar* a mensagem ou *decodificação*. Contudo, terceiros podem, através de "escuta", ter acesso à mensagem *cifrada* e determinar o texto original ou mesmo a *chave*, "quebrando" o sistema.

Portanto, a c*riptoanálise é* a arte ou a ciência de determinar a *chave* ou *decifrar* mensagens sem conhecer a *chave*. Por sua vez, a c*riptologia é* a ciência que reúne a *criptografia* e a *criptoanálise*, sendo uma ciência muito antiga presente no sistema de escrita *hieroglífica* dos egípcios há quase quatro mil anos e desde então vem sendo muito utilizada, principalmente para fins militares e diplomáticos. Sua utilização durante a Segunda Guerra Mundial e a consequente quebra dos *códigos* alemães e japonês foi fundamental para o sucesso dos Aliados. Logo, depois da Segunda Guerra Mundial, com a invenção do computador, a área cresceu, incorporando complexos algoritmos matemáticos.

Nos dias atuais, em toda operação realizada pela internet que envolva troca de dados sigilosos, como envio de uma senha, número do cartão de crédito ou uma mensagem para alguém, a *criptografia* deve ser utilizada, para garantir a segurança do processo e para que somente o remetente tenha acesso a estas informações. Assim, para *codificar* ou *decodificar* uma mensagem, necessitamos de informações confidenciais geralmente denominadas *chaves* ou *senhas*. Dependendo do método de *criptografia* empregado, a mesma *chave* pode ser utilizada tanto para *codificar*, como para *decodificar* mensagens, enquanto outros mecanismos utilizavam *chaves* diferentes.

1.2 APLICAÇÕES DA CRIPTOGRAFIA

Um exemplo de aplicação da *criptografia* ocorre no caso de caixas automáticos, onde um usuário pode retirar dinheiro de sua conta bancária. Para maior segurança, a identidade do cliente, dada pelo seu cartão magnético e sua senha pessoal, é transmitida pela linha a um computador central, que faz as verificações adequadas de validade da senha e saldo disponível. Uma escuta poderia ler as mensagens entre terminal e computador, adulterá-las ou mesmo inserir mensagens para validar um cartão roubado ou perdido; poderia também fazer o chamado "*ataque da meia noite"*, em que as mensagens, gravadas anteriormente, poderiam ser repetidas, iludindo terminal e computador. Contudo, a *criptografia* pode ser usada para impedir esse tipo de fraude.

Outro exemplo de aplicação de *criptografia* ocorre no caso de compartilhamento de arquivos grandes, em que certas informações no arquivo não devem ser acessíveis a parte dos usuários, exemplificando o caso de arquivos de pacientes em um hospital, em que se deseja proteger a identidade do paciente, mas não outras informações de natureza médica, com potencial uso estatístico. Nesse caso, poder-se-ia *cifrar* a identificação do paciente, permitindo a outros usuários o acesso ao mesmo arquivo.

Atualmente, a *criptografia* é aplicada em várias áreas, como por exemplo:

- Recursos humanos;
- Compras e vendas;
- Jurídico;
- Automação de escritórios.

Basicamente, a *criptografia* computadorizada é usada para garantir:

- Sigilo de informações;
- Integridade de informações;
- Autenticação de usuário;
- Autenticação de remetentes;
- Autenticação de destinatários;

1.3 A INFLUÊNCIA DA CRIPTOLOGIA AO LONGO DA HISTÓRIA

Desde os tempos do imperador romano Júlio César, os governantes já percebiam as vantagens fornecidas pela *criptologia*. Com uma *criptografia* forte conseguiram manter protegidas suas informações sensíveis e com a *criptoanálise* buscaram as informações de seus adversários. Mas é na Primeira Guerra Mundial que se manifesta de forma clara e evidente, a influência da *criptologia* nos destinos dos povos.

Era início de 1917 e os ingleses, saturados por uma guerra que se arrastava, ansiavam pela adesão dos Estados Unidos. Os alemães, também desgastados, pretendiam iniciar uma guerra submarina irrestrita, a fim de cortar os suprimentos para os ingleses. Sabendo que esta atitude fatalmente iria encerrar a neutralidade americana, o Ministro do Exterior da Alemanha, Arthur Zimmermann (1864 – 1940), envia um telegrama endereçado ao presidente do México, propondo uma aliança militar contra os Estados Unidos. O telegrama foi interceptado pelos ingleses, que o *decifraram*, e passaram aos americanos. Atônitos com a ousadia dos alemães, os americanos entram na guerra.

Durante a Segunda Guerra Mundial, novamente a *criptografia* e a *criptoanálise* desempenharam papel decisivo no desenrolar dos confrontos. Os americanos não foram capazes de prever a audaciosa operação conduzida pelos japoneses, apesar de terem interceptado e *decifrado* mensagens diplomáticas daquele governo indicando um possível ataque. Contudo, a situação foi revertida pelos americanos ao longo do conflito. Além de quebrarem as *cifras* japonesas "Red" e "Purple", conseguiram manter sua principal *cifra* a "*Sigaba*", intacta.

Mas é do velho continente que vem o maior exemplo de sucesso da criptografia. Além de utilizarem sua *cifra* "*Typex*" com êxito, os ingleses protagonizaram uma das maiores operações de quebra de *cifra* de que se tem notícia, herdando informações iniciais dos poloneses e contando com a genialidade de pessoas como Alan Turing que foi capaz de *decifrar* a máquina alemã "Enigma". A guerra se transformava em um jogo de cartas marcadas, com os ingleses podendo prever grande parte das jogadas dos alemães. Por conseguinte, uma aliança entre americanos e ingleses, para a troca de informações oriundas de suas operações de *criptoanálise* permitiu, segundo historiadores, que a guerra fosse abreviada em dois a três anos, depois de desvendar os mistérios da máquina Enigma.

1.4 - MÁQUINAS POR TRÁS DOS CÓDIGOS

Com o início da Segunda Guerra Mundial, os governos procuravam desenvolver computadores para explorar o seu potencial de importância estratégica. Isto se deve ao aumento do financiamento para projetos de desenvolvimento do computador, o que acelerou o progresso técnico.

Durante a Segunda Guerra Mundial foi notável a construção das máquinas para quebrar *códigos* e *cifras* alemães. O trabalho foi iniciado pelos poloneses, em virtude da conjuntura bélica que cercava o país, e aperfeiçoada pelos ingleses em Bletchley Park - Bletchley Park é o nome de uma antiga instalação militar secreta inglesa, localizada na cidade de Bletchley na Inglaterra, dedicada a quebrar o *código* alemão durante a Segunda Guerra Mundial, onde foi notável o desenvolvimento de mecanismos computadorizados, fruto do trabalho dos melhores matemáticos ingleses.

Apesar do surpreendente avanço tecnológico, pouco foi divulgado, pelo fato do sigilo que envolvia toda a atividade e dos métodos de classificação das informações. Assim, para elucidar a empregabilidade do objeto deste estudo, torna-se necessário apresentar e conhecer algumas das principais máquinas criadas.

1.4.1- Enigma

Enigma é uma máquina eletromecânica com rotores utilizada para a codificação e decodificação de mensagens secretas. A história da máquina Enigma é uma das mais espetaculares de toda a Segunda Guerra Mundial. O invento do engenheiro alemão Arthur Scherbius (1878-1929), em 1918, era capaz de gerar mensagens *codificadas,* virtualmente indecifráveis. Inicialmente foi usada na iniciativa privada para evitar espionagem industrial. A partir de meados da década de 1920, vários ramos das forças armadas alemães começaram a usar a Enigma, fazendo uma série de mudanças, a fim de aumentar a sua segurança.

A partir de 1926, o Exército Germânico começou a usar a Enigma com fins militares e enlouqueceu os *criptoanalistas* de todo o mundo. Ninguém parecia capaz de *decifrar* as mensagens *codificadas*, até que surgiram um polonês Marian Rejewski (1905 – 1980) e um traidor alemão Hans-Thilo Schmidt (1888-1943). Com seus esforços, foi possível desvendar a arquitetura da máquina e alguns dos seus segredos, fazendo com que eles pudessem ler algumas mensagens do exército alemão. Porém, com o começo da Segunda Guerra Mundial, os alemães resolveram complicar ainda mais a *codificação* da Enigma, levando todos os analistas de volta à estaca zero.

O contra-ataque dos Aliados surgiu na Inglaterra, no setor do governo, em Bletchley Park. Lá foram reunidos matemáticos, linguistas, *criptoanalistas*, todos em um esforço conjunto para vencer a máquina Enigma. O destaque nesta batalha foi o matemático britânico Alan Turing (1912 –1954), que atacou com precisão todas as fraquezas da máquina. Contudo, o sucesso dos esforços não foi total e completamente seguro, mas os britânicos, e posteriormente os americanos, foram capazes de ler o suficiente das mensagens alemãs, para que pudessem ajudar suas forças armadas a derrotar a Alemanha Nazista.

É interessante ressaltar que a quebra do *código* das máquinas Enigma, pelos ingleses, foi tornada pública somente na década de 70. Até então, várias nações, principalmente ex-colônias britânicas, utilizaram a Enigma para a proteção de suas informações. Desde então, o interesse pela história da máquina Enigma tem crescido consideravelmente e uma série de Enigmas estão em exibição pública em museus na Europa e os EUA.

Máquina Enigma

Hans-Thilo Schmidt

Marian Rejewski

Arthur Scherbius

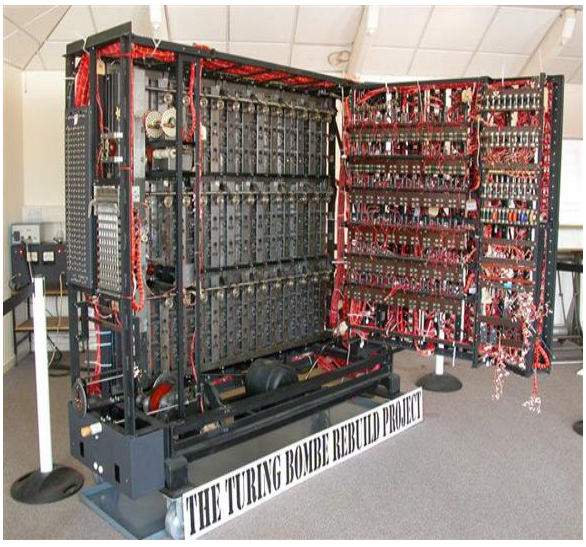
1.4.2- BOMBA

A Máquina de Turing era um dispositivo eletromecânico usado por *criptoanalistas* britânicos para ajudar a *decifrar* mensagens alemãs *codificadas* pela máquina Enigma, durante a Segunda Guerra Mundial. Deste modo, Alan Turing e seus colegas usaram suas habilidades matemáticas para melhorar uma máquina que tinha sido projetada em 1938 pelo polonês Marian Rejewski, conhecida mais tarde por Máquina de Turing.

Explorado pelo fato de que muitas vezes as mensagens *codificadas* pelos alemães continham palavras ou frases comuns, como nomes de generais ou previsões do tempo, então os britânicos foram capazes de adivinhar peças curtas da mensagem original, comparando, de forma relativamente rápida, as *chaves* de várias partes da máquina Enigma, decifrando as mensagens alemãs.

Os dispositivos eletromecânicos foram implantados em vários estabelecimentos militares na Grã- Bretanha em segredo, onde jovens mulheres *decifravam* mensagens *codificadas* enviadas pelos alemães. Todas as 210 Máquinas de Turing originais foram destruídas após a guerra, mas anotações sobre a máquina foram encontradas mais tarde em Bletchley Park. Então começou a missão de recriar uma réplica da Máquina de Turing, que foi concluída em julho de 2007.

Depois de vários anos, o público pode imaginar o que era trabalhar em Bletchley Park na quebra dos *códigos* alemães, e assim evitar muitas mortes. Isso só foi possível graças ao trabalho de uma equipe, que se dedicou durante 12 anos na construção de uma réplica da *Máquina Turing* britânica.



Fonte: theregister.com.uk

Alan Turing

1.4.3- Lorenz

O exército alemão precisava de uma máquina de alta segurança, para que pudessem se comunicar por rádio ou telegrama, em completo sigilo. A empresa Lorenz projetou uma máquina conhecida como Lorenz com base no método aditivo para *cifrar* mensagens de telegrama inventado em 1918 por Gilbert Vernam (1890 – 1960). A máquina Lorenz era considerada mais complexa que a Enigma.

Quando os circuitos de terra-linha telegráfica não estavam disponíveis, as mensagens eram transmitidas por rádio e captadas pela Grã-Bretanha. As mensagens eram enviadas ao *código* do governo e para a escola Cypher em Bletchley Park. Um pequeno número foi *decifrado* por métodos manuais antes que o processo fosse parcialmente automatizado com a criação do computador Colossus.

A máquina Lorenz SZ42 cifra em exposição em Bletchley Park.

Gilbert Vernam

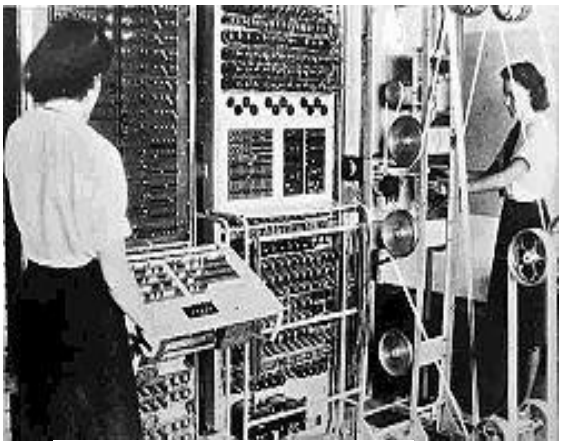
1.4.4- Colossus.

A primeira máquina projetada para quebrar a *cifra* Lorenz foi criada em 1943, pelo britânico Tommy Flowers (1905-1998), denominada Colossus. Com a entrada dos engenheiros eletrônicos Harry Fensom, Coombs Allen (1911 – 1995), Broadhurst Sid e Bill Chander no Post Office(correios) Research Station, Dollis Hill, no noroeste de Londres, para resolver um problema suscitado pelo matemático Max Newman (1897 -1984) em Bletchley Park. A seção em Bletchley Park foi nomeada Newmanry em homenagem ao um dos *decifradores*, o matemático Max Newman responsável pela máquina Robinson e os dez computadores Colossus.

O projeto e a própria existência dos computadores Colossus era altamente secreta e assim permaneceu por muitos anos após a guerra, mesmo que seus detalhes técnicos tivessem perdido a importância. Isto se deve ao fato do Reino Unido ter vendido para outros governos equipamentos de *decodificação*, baseados na máquina Enigma e, em seguida, quebrar os

códigos, usando uma variedade de métodos, baseados nos conhecimentos adquiridos com o Colossus. Vale ressaltar que as informações sobre a máquina Colossus só começaram a aparecer publicamente no final da década de 70, após o silêncio forçado pela lei de Segredos Oficiais, que terminou em 1976.

Uma réplica totalmente funcional de um Colossus Mark 2 foi recriada por uma equipe liderada por Tony Sale, concluída em 2007. Mesmo depois que a maior parte de plantas e *hardware* havia sido destruída, parte do material sobreviveu, principalmente nas anotações dos engenheiros. A reconstrução está em exposição no Museu Nacional de Computação, no bloco H em Bletchley Park.



Colossus sendo utilizado

Tommy Flowers

Em 1994, uma equipe liderada por Tony Sale começou a reconstrução de um Colossus em Bletchley Park. Aqui, em 2006, Tony supervisiona a quebra de uma mensagem *cifrada* com a máquina completa.

CAPÍTULO 2

CRIPTOGRAFIA SIMÉTRICA E ASSIMÉTRICA

Conforme dito anteriormente, a criptografia é a arte ou a ciência de escrever em *cifras* ou em *códigos*, de forma que apenas o destinatário *decifre* e compreenda a mensagem.

O ato de transformar os dados para uma forma ilegível é denominado *cifrar* ou *codificar*, e procura garantir a privacidade, mantendo a informação escondida de pessoas não autorizadas, mesmo que estas possam visualizar os dados *criptografados*.

Ao *codificar* ou *decodificar* uma mensagem, precisamos de informações confidenciais, denominadas *chaves*. Os algoritmos de *criptografia* podem ser classificados em dois tipos, de acordo com o tipo de *chave* que usam: de *chave simétrica* e de *chave assimétrica*.

2.1 - SISTEMAS CRIPTOGRÁFICOS SIMÉTRICOS

O *sistema criptográfico simétrico,* também chamado de *criptografia de chave única* ou *criptografia de chave secreta, é* um *algoritmo* de *criptografia* que utiliza duas *chaves* iguais, onde a mesma *chave* é usada tanto para *codificar* quanto para *decodificar* a mensagem. A *chave* deve ser mantida em sigilo e conhecida apenas pelo remetente e destinatário, sendo que os procedimentos para a *codificação* e a *decodificação* são funções computacionalmente viáveis da *chave*.

Os algoritmos usados na *criptografia simétrica* são mais simples e rápidos do que os algoritmos usados na *criptografia assimétrica*. Portanto, é mais adequada para *codificação* e *decodificação* de uma grande quantidade de blocos.

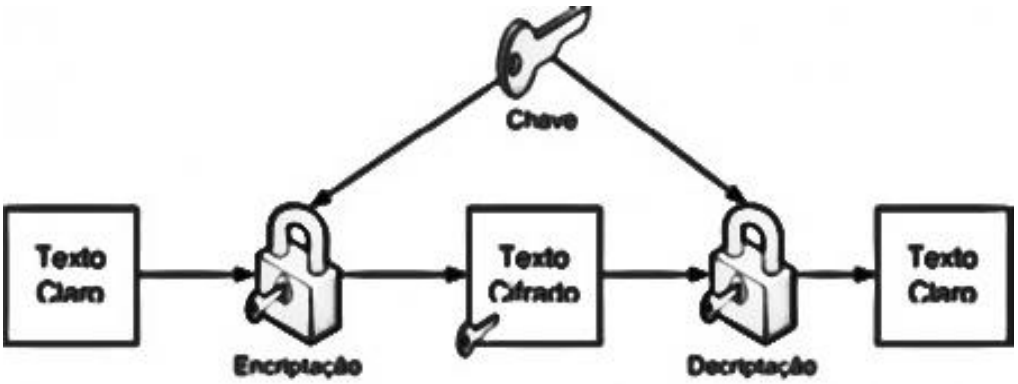


Diagrama retirado do site penseesponda.wordpress.com

Dois exemplos de algoritmos de *criptografia simétrica* são: O algoritmo DES[2] e o Algoritmo AES[3].

2.1.1 - O algoritmo DES

Uma função simétrica é o DES que teve origem em 1973, quando o NBS (National Bureau of Standards), atualmente conhecida como NIST (National Institute of Standards and Technology) solicitou propostas para algoritmos *criptográficos* que respondesse aos seguintes critérios:

- Possuir um elevado nível de segurança ligado a uma *chave* pequena que sirva para a *codificação* e *decodificação*;
- Ser compreensível;
- Não depender da confidencialidade do algoritmo;
- Ser adaptável e econômico;
- Ser eficaz e exportável.

Entretanto, nenhuma das propostas recebidas se mostrou viável. Uma segunda solicitação foi aberta. Até que a IBM (International Business Machines - empresa dos Estados Unidos voltada para a área da informática) apresentou uma proposta candidata que ela havia desenvolvida internamente, denominado LUCIFER, projetado por Horst Feistel. Horst Feistel foi um *criptógrafo* que chegou como imigrante alemão aos Estados Unidos em 1934. Quando os Estados Unidos entrou na guerra, Feistel, como era alemão, ficou em prisão domiciliar, a qual se estendeu até 1944. Consequentemente, por muitos anos manteve-se distante da *criptologia* para evitar levantar suspeitas das autoridades. Assim, após esses anos Feistel acabou indo para o Thomas J. Watson Laboratory da IBM, perto de Nova Iorque, onde, por alguns anos, conseguiu prosseguir com suas pesquisas sem ser cassado. Foi neste laboratório que, no início de 1970, ele desenvolveu o sistema LUCIFER (que era o nome dado as primeiras cifras desenvolvido por ele) e foi este o sistema apresentado à NBS.

Em 1977, o NBS adotou o DES como padrão americano de *codificação* de dados para aplicações não ligadas à segurança nacional. O DES é disponível em “chips”, e sua exportação está condicionada à autorização do governo americano. O DES *cifra* blocos de 64 bits em blocos de 64 bits, utilizados; porém, somente 56 deles pelo algoritmo, os oitos bits restantes são utilizados para verificar a paridade e depois são descartados. Portanto, o tamanho efetivo da

[2]Data Encryption Standard
[3]Advanced Encryption Standard

chave é de 56 bits. O DES é atualmente considerado inseguro para muitas aplicações. Isto se deve a pequena *chave* de 56 bits.

Apesar das diversas críticas, o DES foi aprovado pelo governo dos EUA como padrão em 1976 e publicado em 15 de janeiro de 1977 como **FIPS PUB 46**, autorizado a ser utilizado. Subsequentemente, foi reafirmado como padrão em 1983, revisado em 1988, como **FIPS-46-1**, em 1993 (**FIPS-46-2**) e novamente em 1999 (**FIPS-46-3**); o último é conhecido como "**Triplo DES**".

Em 26 de Maio de 2002, DES foi finalmente substituído pelo AES após uma competição pública. Mesmo assim, até 2004, os restos do DES continuaram a ser utilizados em larga escala. Em 19 de Maio de 2005, FIPS 46-3 foi oficialmente substituído, mas a NIST(National Institute of Standards and Technology) aprovou o "Triplo DES" até o ano 2030 para informações sensíveis do governo.

2.1.2- O algoritmo AES

Outro exemplo de função simétrica é o AES, o atual padrão de *criptografia* dos EUA, que se originou de um concurso lançado em 1997 pelo NIST. Pois havia na época a necessidade de escolher um algoritmo mais seguro e eficiente para substituir o DES que apresentou fragilidades. Assim, o novo algoritmo deveria atender a certos pré-requisitos como:

- Ser divulgado publicamente e não possuir patentes;
- *Cifrar* em blocos de 128 bits usando *chaves* de 128, 192 e 256 bits;
- Ser implementado tanto em software quanto em hardware;
- Ter mais rapidez em relação ao 3DES - uma variação recursiva do antigo padrão DES.

Em 2000 foi conhecido o vencedor: Rijndael. O nome é uma fusão de Vincent Rijmen e Joan Daemen, os dois criadores belgas. Onde foi escolhido com base em qualidades como segurança, flexibilidade, bom desempenho em software e hardware.

O Rijndael apresenta alta resistência a ataques como "Power attack" e "timing attack" e exige pouca memória, o que o torna adequado para operar em ambientes restritos como "smart cards", PDAs e telefones celulares. Assim, depois de cinco anos de um processo de padronização, o AES tornou-se um padrão efetivo em 26 de maio de 2002, sendo considerado um dos algoritmos mais populares usados para *criptografia de chave simétrica*.

2.2-SISTEMA CRIPTOGRAFICO ASSIMÉTRICO

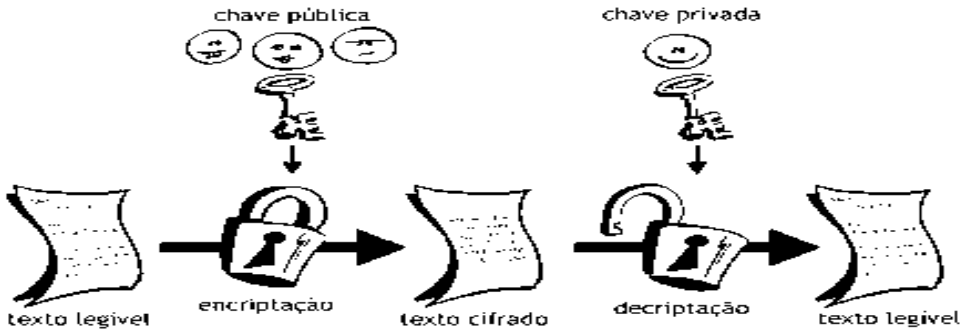


Imagem retirada do site atributoseguros.com.

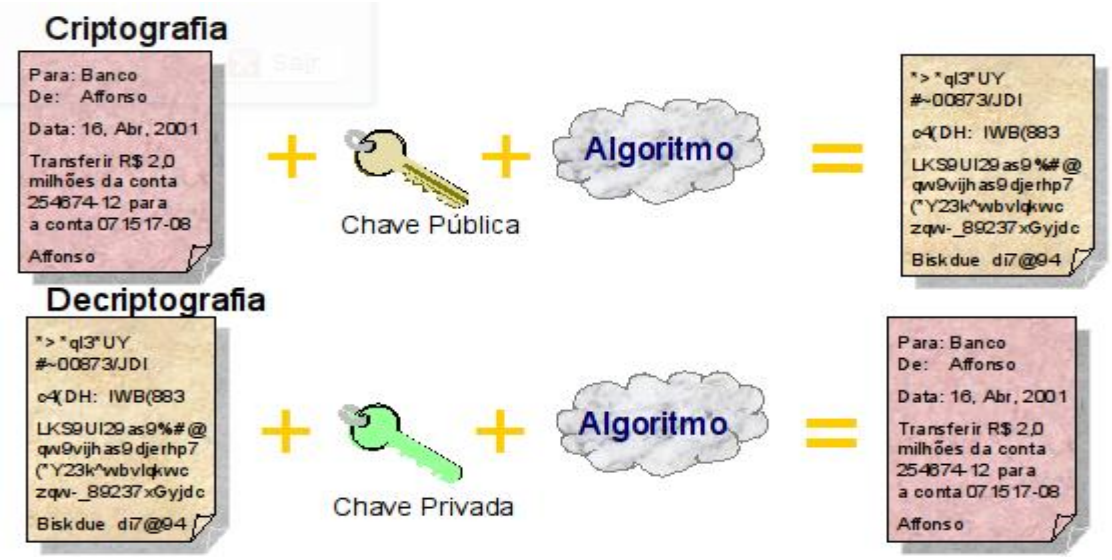
A *criptografia assimétrica* usa duas *chaves* diferentes, que estão matematicamente relacionadas:

- Uma de *chave privada*, que é uma informação pessoal que permanece em posse da pessoa, não publicável.
- Uma de *chave pública*, que é distribuída livremente para todos os correspondentes via e-mail ou de outras formas.

A *criptografia assimétrica* é considerada mais segura do que a *criptografia simétrica*, porque a *chave* usada para *codificar* os dados é diferente da que é usada para *decodificar*. Porém, o processo de *codificação* na *criptografia assimétrica* é muito mais lento do que na *criptografia simétrica*, pois usa algoritmos mais complexos. Um exemplo de *criptografia assimétrica* é a *criptografia de chave pública* que se refere a um conjunto de métodos para transformar uma mensagem escrita em uma forma que só pode ser lida pelo destinatário. Essa abordagem de *criptografia* envolve o uso de algoritmos de *chave assimétrica*.

Portanto, em um algoritmo de *criptografia assimétrica*, a mensagem *codificada* com a *chave pública* só pode ser *decodificada* pela sua *chave* privada correspondente. Imagine a situação hipotética:

1. Alice cria seu par de *chaves* (uma pública e outra privada) e envia sua *chave* pública para todos, inclusive para Bob;
2. Bob escreve sua mensagem para Alice. Depois de escrita, Bob faz a *codificação* do texto final com a *chave* pública de Alice, gerando um texto *criptografado*;
3. Alice recebe o texto *criptografado* de Bob e faz a *decodificação* utilizando a sua *chave* privada



O procedimento é realizado com sucesso porque somente a *chave* privada de Alice é capaz de *decifrar* um texto *criptografado* com a sua *chave pública*.

É importante destacar que se aplicarmos a *chave pública* de Alice sobre o texto *criptografado* não teremos a mensagem original de Bob. Dessa forma, mesmo que a mensagem seja interceptada é impossível *decifrá-la* sem a *chave* privada de Alice.

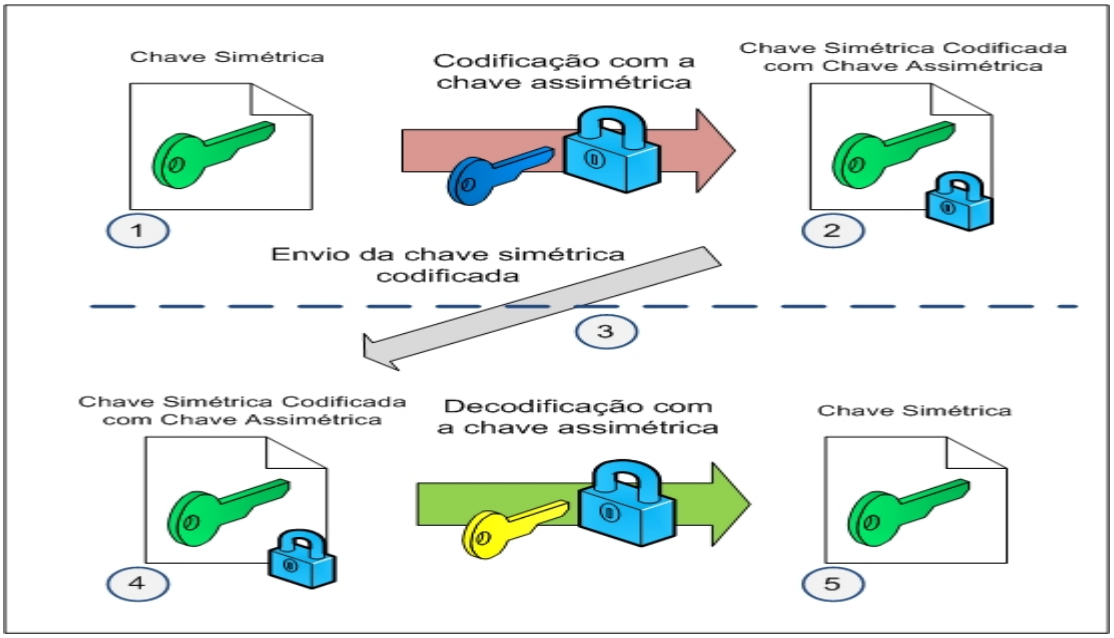
2.3 – CRIPTOGRAFIA SIMÉTRICA X ASSIMÉTRICA

Como elucidado anteriormente, para *codificar* e *decodificar* uma mensagem usando o sistema *criptográfico simétrico* é necessário que ambas as partes compartilhem a mesma *chave* de *codificar* e *decodificar* a mensagem. Contudo, para proporcionar privacidade esta *chave* precisa ser mantida em segredo.

Usando o sistema *criptográfico assimétrico* é necessário usar pares de *chaves*, uma para *codificar* outra para *decodificar*. Assim, a *chave* de *decodificar* deve ser mantida em sigilo e a *chave* de *codificar* pode ser enviada para todos os que possam querer enviar mensagens *codificadas*.

Entretanto, podemos aproveitar as vantagens do modelo *simétrico* e *assimétrico* em um só modelo, este modelo é denominado *híbrido*. Em que o algoritmo *simétrico* é utilizado para *criptografar* as informações em si, pois é mais rápido. Já o *assimétrico*, apesar de ser lento, possibilita distribuir as *chaves* de forma segura e utilizar a assinatura digital. O funcionamento do modelo pode ser descrito da seguinte forma:

A partir de um dos pontos envolvidos na comunicação é gerada a *chave* secreta, através de algoritmos para geração de *chave simétrica*. Essa *chave* vai ser utilizada na *criptografia* da mensagem, mas para que alguém possa *decodificar* a mensagem é necessário conhecer a *chave* secreta. Então, além de enviar a mensagem *codificada* para o destinatário, também é necessário enviar a *chave* de forma segura, *codificando-a* com a *chave pública* do destinatário como mostra o seguinte esquema:



Assim, quando o usuário receber a *chave* secreta que foi *codificada* com sua *chave* pública, vai utilizar sua *chave* privada para *decodificar* e obter a *chave* secreta, a qual será utilizada para *decodificar* a mensagem.

Um exemplo em que é empregada a combinação de métodos *criptográficos* é uma transação comercial, em que são utilizados sistemas de *chave pública* para se transmitir uma *chave secreta simétrica*. Após a transmissão segura da *chave*, pode-se utilizar a *chave simétrica* para trocar dados seguramente entre as duas entidades relacionadas no processo.

2.4- CRIPTOGRAFIA RSA

Clifford Cocks, um matemático britânico que trabalhava para a agência de inteligência Government Communications Headquarters, inventou um algoritmo de *criptografia de chave pública* análogo ao que conhecemos hoje como RSA. Porém, não é reconhecido pela sua realização, porque o seu trabalho foi mantido em sigilo, isso se deve ao fato de que na época os computadores necessários para a implementação dos algoritmos eram relativamente caros. Deste modo, em 1978, três anos após a descoberta de Clifford Cocks, o trio de matemáticos R.L. Rivest, A. Shamire L. Adlemano desenvolveu de forma independente do trabalho de Cocks, descrevendo o algoritmo de *criptografia de chave pública*, esse é motivo pelo qual as siglas RSA levam as letras iniciais de seus sobrenomes.

Atualmente o RSA é considerado um dos algoritmos de *criptografia de chave pública*, mais usado em aplicações comerciais como a internet que o utiliza nas mensagens de emails, de compras on-line e outros. Portanto, tudo é *codificado* e *decodificado* pela *criptografia* RSA, em que sua segurança está baseada na dificuldade de fatorar números inteiros grandes.

Para que haja a comunicação entre duas fontes A e B, usando o sistema RSA é preciso passar por duas etapas: geração de *chaves* de *codificar* e *decodificar*. Logo, para entender melhor como este método funciona, é necessário o estudo de alguns conceitos presentes em uma área da Matemática chamada Teoria dos Números, explanada no capítulo seguinte.

Ronald Rivest (centro), com Alan Sherman (esquerda) e David Chaum (direita), em 2007.

CAPÍTULO 3

EMBASAMENTO TEÓRICO

Vários sistemas de *criptografia* em uso são justificados por meio da área da Matemática denominada Teoria dos Números, que estuda as propriedades dos números, em particular dos números inteiros. É uma das áreas mais antigas da matemática. Logo, para um melhor entendimento referente ao objeto de estudo desta dissertação, se faz necessário enunciar algumas definições e propriedades da Teoria dos Números, que servirá de embasamento para o entendimento da *Criptografia RSA* no que diz respeito à formulação teórica.

3.1- DIVISIBILIDADE

Definição: *Sejam a e b números inteiros. Diz-se que b divide a (ou que b é um divisor de a, ou ainda, que a é um múltiplo de b) se existe um inteiro c tal que b.c = a.*

Em geral usamos a notação a $\mid$ b pra indicar que "a divide b", e a $\not\mid$ b indica que "a não divide b". A relação "a divide b", indicada pelo símbolo $\mid$, é chamada de relação de divisibilidade nos inteiros (Z).

3.1.1- Propriedades básicas da divisão

Se a, b e c são números inteiro, valem as seguintes propriedades:

(1) Se $a \neq 0$, então a divide a e a divide 0.
(2) Para qualquer a, $1 \mid a$.
(3) Se $a \mid b$ e $a \mid c$, então $a \mid bm + cn$, $\forall\, m,n \in \mathrm{Z}$
(4) Se $a \mid b$ e $b \mid c$, então $a \mid c$.
(5) Se $a > 0$ e $b > 0$, $a \mid b$ e $b \mid a$, então $a = b$.
(6) Se $a > 0$ e $b > 0$, $a \mid b$, então $a \leq b$.

3.2 – O MÁXIMO DIVISOR COMUM

Definição: *Dados dois números inteiros a e b. O máximo divisor comum entre a e b é o maior inteiro positivo d que é divisor de a e também é divisor de b. Se d é o máximo divisor comum entre a e b, escrevemos d = m.d.c. (a, b).*

Por exemplo, o número inteiro 4 divide 12 e também 4 divide 20 e, além disso, podemos verificar que 4 é o maior número inteiro positivo com essa propriedade. Dizemos, então, que 4 é o máximo divisor comum de 12 e 20, ou seja m.d.c.(12,20) = 4.

3.2.1- Propriedades do MDC

1. m.d.c. (a,b) = m.d.c. (b,a)
2. m.d.c. (a,b) = m.d.c. (-a,b) = m.d.c.(a,-b) = m.d.c.(-a,-b)
3. m.d.c.(a,b) = m.d.c.($|a|$, $|b|$)
4. se a $|$b então m.d.c. (a,b) = $|a|$
5. se a $\neq 0$ então m.d.c.(a,0) = $|a|$
6. se a $\neq 0$ então m.d.c.(a, a) = $|a|$
7. para todo inteiro a, m.d.c. (a, 1) = 1

3.3 - NÚMEROS PRIMOS E NÚMEROS PRIMOS ENTRE SI

3.3.1- Número primo

Definição: *Um número inteiro p é chamado de número primo se as seguintes condições se verificam:*

- $p \neq 0$
- $p \neq \pm 1$
- Os únicos divisores de p são ± 1 e $\pm p$.

3.3.2- Números primos entre si

Definição: *Dados dois números a e b, dizemos que são primos entre si se o m.d.c.(a, b) = 1.*

Para que dois números sejam primos entre si não é necessário que sejam primos. Observe que 27 e 25 são primos entre si, pois m.d.c. (25,27) = 1, mas nenhum deles é primo.

No entanto, se dois números distintos são primos, também são primos entre si. Os números 3 e 7 são primos e portanto primos entre si.

3.4 - O ALGORITMO DA DIVISÃO

Teorema da divisão: *Sejam a e b inteiros positivos. Existem números inteiros q e r tais que* a = b . q + r e $0 \leq r < b$.

Além disso, os valores de q e r satisfazendo as relações acima são únicos.
(q é chamado de quociente e r de resto da divisão de a por b)

O algoritmo calcula o quociente e o resto na divisão de um número inteiro por outro. Esse processo é conhecido como **Algoritmo da Divisão** (apresentado por Euclides).

EXEMPLO:

1. Sabemos que 38 não é divisível por 5, mas, no entanto, podemos escrever: 38 = 5 . 7 + 3. Nesse caso, 7 é **o quociente** e 3 é **o resto** da divisão de 38 por 5.

2. - 26 = (- 7). 4 + 2, nesse caso, -7 é o quociente e 2 é resto da divisão de –26 por 4.

3.5 – O ALGORITMO DE EUCLIDES

O algoritmo euclidiano é usado para calcular o máximo divisor comum entre dois números inteiros.

Podemos provar que *o máximo divisor comum entre a e b é o número d tal que*:

- d $|a$ (ou d é divisor de a)
- d $|b$
- Se d`é divisor de a e b, então d` $|d$, ou seja, d é o maior divisor de a e b.

Teorema: *Dados a e b inteiros positivos, o último resto diferente de zero da sequência de divisões dada pelo algoritmo euclidiano para a e b é máximo divisor comum entre a e b.*

- $a = q_0.b + r_0$
- $b = q_1. r_0 + r_1$
- $r_0 = q_2. r_1 + r_2$
- $r_1 = q_3. r_2 + r_3$

⋮

- $r_{n-1} = q_{n+1} .r_n + r_{n+1}$
- $r_n = q_{n+2} .r_{n+1} + 0$

Temos que $r_0 > r_1 > r_{2.....} > r_n \geq 0$.

Dados dois números inteiros positivos a e b tais que a $\geq$ b, divide-se a por b, encontrando resto r_1. Se $r_1 \neq 0$, dividimos b por r_1, obtendo r_2. Se $r_2 \neq 0$, dividimos r_1 por r_2 e assim por diante.

O último resto diferente de zero dessa sequência de divisões é o m.d.c.(a, b).

EXEMPLO 1:

Usando o Algoritmo de Euclides, vamos calcular o m.d.c.(360,126).

360 = 126 x 2 +108, sendo 2 o **quociente** e 108 é o **resto**.

126 = 108 x 1 + 18, sendo 1 o **quociente** e 18 é **o resto**.

108 = 18 x 6 + 0, sendo 6 o **quociente** e 0 é o resto.

Como o resto da última divisão é nulo, então o divisor 18 é o Máximo Divisor de 360 e 126, isto é m.d.c.(360,126) = 18.

É comum esse processo ser representado pelo esquema a seguir:

	2	1	6
360	126	108	18
108	18	0	

EXEMPLO 2:

Encontre o m.d.c. (752, 193)

752 = 193 x 3 + 173

193 = 173 x 1 +20

173 = 20 x 8 + 13

20 = 13 x 1 +7

13 = 7 x 1 + 6

7 = 6 x 1 +1

6 = 6 x 1 +0

Daí o m.d.c. entre 752 e 193 é igual a 1.

3.6- ALGORITMO DE EUCLIDES ESTENDIDO

Quando calculamos o m.d.c. podemos extrair outra informação, ou seja:

Sejam, a e b inteiros positivos, com b $\neq 0$*, e d é o máximo divisor comum entre eles. Então é possível achar* α *e* β *inteiros tais que:*

$$\alpha .a + \beta . b = d$$

O primeiro passo é acharmos o valor de d, através do algoritmo euclidiano. Como por exemplo, sejam a = 172 e b = 20, temos que:

172 = 20 x 8 +12

20 = 12 x 1 + 8

12 =8 x 1 + 4

8 = 4 x 2 + 0

Então o m.d.c. entre 752 e 20 é igual a 4. Para calcular **xa + yb = d = m.d.c. (a,b),** temos que:

4 = 12-1 x 8

4 = 12 – 1 x (20 – 1x12)

4 = 2 x 12 – 20

4 = 2 x (172 – 20 x 8) – 20

4 = 2 x 172 – 2x 8 x 20 -20

4 = 2 x 172 –17x 20

Logo, **x** = 2 e **y** = -17

Podemos observar que o teorema não diz que os valores de α e β são únicos. Na verdade, existe uma infinidade de números que satisfazem a equação α. a + β.b = d. O cálculo do algoritmo euclidiano estendido é uma das bases do algoritmo RSA.

3.7 - CONGRUÊNCIAS

Definição: *Sejam a e b dois números inteiros e m um número natural. Dizemos que a é congruente a b módulo m se e somente se*

$$m|(a-b)$$

Observe que se $m|(a-b)$, naturalmente $m|(b-a)$, pois a – b = (-1). (b – a).

Escrevemos:

a ≡ b (mod n)

Se m $\nmid$ (a – b), então escrevemos a $\not\equiv$ b ou seja, a não é congruente a b.

EXEMPLOS:

1. 33 ≡ 18 (mod3) ⇔ 33 – 18 é divisível por 3.
2. 33 é congruente a 18 módulo 3 se, e somente se, a diferença 33 – 18 é divisível por 3.
3. 25 $\not\equiv$ 12 (mod 17), 25 não é congruente 12 módulo 17, pois 25 – 12 não é divisível por 17.
4. 18 ≡ -1 (mod 19) ⇔ 18 – (-1) = 18 + 1 = 19

3.7.1– Propriedades básicas das congruências

Se a, b, c e d são inteiros quaisquer e n é um natural maior do que ou igual a 2, são verdadeiras as seguintes propriedades:

I. $a \equiv a$ (mod n) (Reflexividade)

II. Se $a \equiv b$ *(mod n),* então $b \equiv a$ *(mod n)* (Simetria)

III. Se $a \equiv b$ *(mod n)* e $b \equiv c$ *(mod n),* então, $a \equiv c$ *(mod n) (Transitividade)*

IV. Se $a \equiv b$ *(mod n) e* $c \equiv d$ *(mod n), então,* $a + c \equiv b + d$ *(mod n) (Soma)*

V. Se $a \equiv b$ *(mod n) e* $c \equiv d$ *(mod n), então,* $a - c \equiv b - d$ *(mod n)* (Diferença)

VI. Se $a \equiv b$ *(mod n)* e c é um inteiro não negativo, então, $ac \equiv bc$ *(mod n)*

VII. Se $a \equiv b \pmod{n}$ *e* $c \equiv d \pmod{n}$*, então,* $a.c \equiv b.d \pmod{n}$ *(*Produto*)*

VIII. Se $a \equiv b \pmod{n}$ *e k* é um inteiro positivo, então, $ak \equiv bk \pmod{n}$ *(*Potência*)*

IX. Se $a + c \equiv b + c \pmod{n}$, então, $a \equiv b \pmod{n}$ (Cancelamento para a soma)

3.8 – CÁLCULO DE INVERSO MÓDULO N

Definição: *Sejam* $a, m \in Z, m > 0$*. Diz-se que* $b \in Z$ *é um inverso de a módulo m se* $ab \equiv 1 \pmod{m}$*.*

Teorema de inversão: *Sejam* $a, m \in Z, m > 0$*, então existe* $b \epsilon Z$ *com* $ab \equiv 1 \pmod{n}$ *se e somente se a e m são primos entre si.*

O inverso é sempre único módulo m.

Para que um número **a** seja o inverso multiplicativo de **b** módulo m, é necessário que **a** e **b** sejam primos entre si, portanto se o m.d.c. de **a** e **b** encontrado for maior que 1, pode-se concluir que o inverso multiplicativo não existe.

Para calcular o inverso multiplicativo de a mod m, basta usar Algoritmo de Euclides Estendido. O processo é obter o valor do resto na última equação não nulo e ir substituindo os valores dos restos, usando as outras equações, até chegarmos à primeira. Veja:

Vamos calcular o inverso de 193 módulo 752. Primeiro calculamos o m.d.c.(193,752):

752 = 193 x 3 + 173
193 = 173 x 1 +20
173 = 20 x 8 + 13
20 = 13 x 1 +7
13 = 7 x 1 + 6
7 = 6 x 1 +1
6 = 6 x 1 +0

Daí o m.d.c. entre 752 e 193 é igual a 1. Agora vamos substituindo a última equação até chegarmos à primeira.

1 = 7 - 6 x 1

1 = 7 – (13 – 7)

1 = 2 x 7 – 13

1 = 2 x (20 – 13) – 13

1 = 2 x 20 – 3 x 13

1 = 2 x 20 – 3 x (173 – 20 x 8)

1 = 26 x 20 – 3 x 173

1 = 26 x (193 –173) – 3 x 173

1 = 26 x 193 – 29 x 173

1= 26 x 193 – 29 x (752 – 3x 193)

1 = 113 x 193 – 29 x 752

Concluímos que 1 = 113 x 193 – 29 x 752, então se reduzimos o módulo 752 à expressão, obtemos o inverso de 193 que é igual a 113, ou seja, 193. 113 $\equiv$ 1(mod 752)

3.9 – POTÊNCIAS

Como para codificar e decodificar a mensagem escolhida, vamos trabalhar com potências de números grandes, então, outra aplicação muito útil da congruência é determinar o resto da divisão de uma potência a^x por um inteiro n.. Ou seja:

$$a^x \equiv b(\mathrm{mod}\, n)$$

EXEMPLO:

1) Calcule o resto da divisão de 10^{33} por 99.

Temos que: $10^2(\mathrm{mod}99) \equiv 1(\mathrm{mod}99)$

$$10^{33} \equiv 10^{2.16+1} \equiv (10^2)^{16}.10^1 \equiv (1)^{16}.10 \equiv 1.10(\mathrm{mod}99) \equiv 10(\mathrm{mod}99)$$

Portanto 10 é o resto da divisão de 10^{33} por 99

2) Calcule o resto da divisão de 3^{125} por 7.

Temos que: $3^2(\mathrm{mod}7) \equiv 2(\mathrm{mod}7)$

$$3^{125} \equiv 3^{2.62+1} \equiv (3^2)^{62}.3^1 \equiv (2)^{62}.3^1 \equiv 2^{3.20+2}.3^1 \equiv$$
$$(2^3)^{20}.2^2.3^1 \equiv (1)^{20}.2^2.3^1 \equiv 1.4.3 \equiv 5(\mathrm{mod}7)$$

Portanto 5 é o resto da divisão de 3^{125} por 7 .

Estes exemplos demonstram o uso de congruências para encontrar restos de potências. Vale destacar que a potência exata a ser usada em cada caso depende do problema. E também nem sempre a potência é a única ou a melhor escolha.

3.10– PEQUENO TEOREMA DE FERMAT

O pequeno teorema de Fermat nos diz como trabalhar com certas congruências envolvendo expoentes quando o módulo é primo.

Teorema de Fermat: *Se p um número primo e a um número inteiro. Então*

$$a^p \equiv a(\mathrm{mod}\, p)$$

Teorema de Fermat II: *Seja p um número primo e a um inteiro que não é divisível por p. Então.*

$$a^{p-1} \equiv 1(\mathrm{mod}\, p)$$

3.11- FUNÇÃO DE EULER

Definição: *Seja n um número inteiro positivo. A função de Euler $\phi(n)$ é definida como o número de inteiros positivos não excedendo n que são relativamente primos com n.*

Portanto, para $n \geq 1$,

ϕ (n) = número de inteiros entre 1 e n – 1 coprimos com n.

Por exemplo, ϕ (8) = 4, uma vez que 1, 3, 5 e 7 são coprimos de 8.

Se o valor de *n* é pequeno, podemos calcular o valor de ϕ (n) simplesmente contando os inteiros entre 1 e n – 1 coprimos com *n.* Porém, para os valores maiores que *n*, precisamos usar algumas propriedades dessa função. Essas propriedades são enunciadas como proposições.

Proposição 1:*Se p é primo, então $\phi(p) = p - 1$*

EXEMPLOS:

$\phi(5) = 4$

$\phi(11) = 10$

31

Proposição 2: *Se p é primo, então* $\phi(p^{\alpha}) = p^{\alpha} - p^{\alpha-1}$, $\forall\ \alpha \geq 1$.

EXEMPLOS:

ϕ (27) = $\phi(3^3) = 3^3 - 3^{3-1} = 27 - 9 = 18$

ϕ (25) = $\phi(5^2) = 5^2 - 5^{2-1}$ = 25 – 5= 20

Proposição 3: *Se a e b são inteiros positivos e m.d.c.(a, b) = 1, então* $\phi(a.b) = \phi(a).\ \phi(b)$

EXEMPLOS:

ϕ (15) = $\phi(3).\phi(5)$ = 2.4 = 8(3 e 5 são primos entre si)

ϕ (10) = $\phi(2).\ \phi(5)$ = 1.4 = 4(2 e 5 são primos entre si)

ϕ (40) $\neq \phi(4).\phi(10)$, pois 4 e 10 não são primos entre si, o m.d.c.(4,10) = 2.

Finalmente temos que, se $n = p_1^{e}1 ... p_k^{e}k$, então:

$$\phi(n) = p_1^{e}1^{-1} ..... p_k\, ek - 1.(p_1 - 1)..(p_k - 1)$$

Onde e é um número inteiro positivo e $p_1 < p_2 < ... < p_k$ são primos distintos.

3.11.1 – Teorema de Euler

Para que possamos decodificar a mensagem no algoritmo RSA que iremos ver no próximo capítulo, será necessário saber calcular a função de Euler e aplicar o teorema de Euler.

O teorema de Euler é uma generalização do teorema de Fermat para o caso em que o módulo não é primo.

Teorema de Euler: *Se n é um inteiro positivo e a é um inteiro tal que m.d.c.(a, n) = 1*, então

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

EXEMPLOS:

1.Sejam a = 3 e n = 10. Temos que m.d.c.(3,10) = 1. Pelo Teorema temos:

$3^{\phi(10)} \equiv 1(\mathrm{mod}\, 10)$, ou seja, $3^4 \equiv 1(\mathrm{mod}\, 10)$. De fato: $3^4 = 81 \equiv 1(\mathrm{mod}\, 10)$

2. Sejam a = 5 e n =12. Temos que m.d.c.(5,12) = 1. Pelo Teorema temos:

$5^{\phi(12)} \equiv 1(\mathrm{mod}\, 12)$, ou seja, $5^4 \equiv 1(\mathrm{mod}\, 12)$. De fato: $5^4 = 25 \equiv 1(\mathrm{mod}\, 12)$

Assim, foram enunciados alguns dos tópicos da Teoria dos Números. Não é o objetivo desse trabalho esgotar o entendimento sobre esse assunto, porém o que foi apresentado servirá de base para o entendimento da Criptografia RSA e suas aplicações.

33

CAPÍTULO 4

Este capítulo detalha o funcionamento do RSA.

4.1- PROCESSO DE GERAÇÃO DAS CHAVES

As *chaves* são elementos fundamentais que interagem com um conjunto de algoritmos para a *codificação* e a *decodificação* da informação. A *chave* decorre de equações matemáticas aplicadas a partir do conteúdo do arquivo, podendo também derivar de sua associação a outros dados digitalizados.

Do ponto de vista do usuário, as *chaves de criptografias* são similares às senhas de acesso. Usando a senha correta, o usuário tem acesso ao conteúdo da informação, caso contrário, o acesso é negado. Consequentemente, assim como as senhas, as *chaves* na *criptografia* também possuem diferentes tamanhos, sendo seu grau de segurança relacionado ao tamanho da *chave*. Na *criptografia* moderna, as *chaves* são longas sequências de bits.

Então, como mencionado, o RSA é um sistema de *criptografia* que envolve uma *chave pública* e uma *chave privada*. A *chave pública* pode ser conhecida por todos e é usada para *codificar* mensagens, e uma *chave privada* é usada para *decodificar* a mensagem e a mesma deve ser mantida em sigilo ou a segurança do método estará comprometida. Toda mensagem *codificada* usando uma *chave pública* só pode ser *decodificada* usando a respectiva *chave privada*.

As *chaves* para o algoritmo RSA são geradas da seguinte forma:

1. Escolha dois números primos muito grandes distintos **p** e **q**.
2. Calcular **n = p.q**, onde **n** será usado como módulo para ambas às chaves públicas e privadas.
3. Calcular $\phi(n) = (p-1).(q-1)$, onde ϕ é a função de Euler.
4. Escolha um inteiro e tal que $1 < e < \phi(n)$ e m.d.c.$(e, \phi(n)) = 1$ e, isto é, e e ϕ **(n) são primos entre si.**
5. Determinar **d** que é o inverso multiplicativo de e modulo n, ou seja:

$d.e \equiv 1 (\mathrm{mod}\, \phi(n))$.

Desta forma geramos o par (n, e) que é a **chave pública do sistema RSA**, e o par (n, d) é **a chave privada do sistema RSA**.

A título de exemplificação, serão escolhidos números primos pequenos, para permitir um acompanhamento de todo o processo de codificação e decodificação.

1. Tomemos os números p = 11 e q = 13,

2. n = 11. 13= 143**.**

3. ϕ (n) = (p-1). (q-1)

34

ϕ	A	B	C	D	E	F	G	H	I	J	K	L	M	(n) = (11-1). (13-1)
	10	11	12	13	14	15	16	17	18	19	20	21	22	
ϕ	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	(n) = 10.12
ϕ	23	24	25	26	27	28	29	30	31	32	33	34	35	(n) =120

4. m.d.c.(e, 120) = 1, observe que 1,2,3,4,5 e 6 são os menores divisores de 120. Para facilitar os cálculos, tomamos o menor valor que não divide 120, ou seja, o menor primo com 120, então e = 7.

5. Para calcularmos o inverso de e em ϕ(n), basta aplicar o Algoritmo de Euclides Estendido. Sabemos que e = 7 e ϕ(n) = 120, e que o m.d.c.(7,120) = 1.

Pelo Algoritmo de Euclides temos que:

120 = 7.17 +1

120 + 7. (-17) = 1

Logo o inverso de 7(e) módulo 120 é – 17. Como o d na fórmula é um expoente, precisamos que seu valor seja positivo. Logo tomamos d = 120 – 17 = 103

Após os cálculos acima, temos o par (120,7) que é a chave pública do sistema RSA e o par (120,103) que é a chave privada do sistema RSA.

4.2 – PRÉ-CODIFICAÇÃO

Para enviar uma mensagem usando o algoritmo RSA, primeiramente devemos converter a mensagem em uma sequência de números. Existem vários padrões usados em sistema de mensagem. Por exemplo, podemos considerar que a mensagem é constituída pelas letras que formam as palavras e pelos espaços entre elas.

Também vamos adicionar espaço entre duas palavras, que será substituído pelo número 99. Sendo que cada letra corresponde a um número com exatamente dois algarismos. Isso evita ambiguidades.

O último passo da *pré-codificação* é quebrar a mensagem em blocos. Esses blocos devem ser números menores que **n**.

A maneira de escolher os blocos não é única, mas é importante evitar duas situações:

- Nenhum bloco deve começar com o número zero (problemas na decodificação);
- Os blocos não devem corresponder a nenhuma linguística (palavra, letra, etc.).

Assim a decodificação por contagem de frequência fica impossível.

4.3 – PROCESSO DE CODIFICAÇÃO

Considerando a palavra **Matemática**, devemos convertê-la em uma sequência de números.

22102914221029181210

Tendo o valor de n = 120, podemos agora quebrar a mensagem escolhida em blocos, devemos tomar cuidado para que os blocos não ultrapassem o valor de n.

22-10-29-14-22-10-29-18-18-10

De posse da chave pública que é o par(120,7), começamos o processo de codificação da mensagem no RSA, para isso basta utilizar a seguinte fórmula:

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

Onde:

C(b) é o bloco codificado e C(b) é o resto da divisãode b^{e} por n). Fazendo os cálculos dos blocos temos:

$C(22) \equiv 22^{7}$ (Mod 143)

$\equiv 22^{2^{2}}.22^{2}.22$ (Mod 143) 36

$\equiv 55^{2}.55.22$ (Mod 143)

$\equiv 22.55.22$ (Mod 143)

$\equiv 22^{2}.55$ (Mod 143)

$\equiv 55.55$ (Mod 143)

$\equiv 22$ (Mod 143)

$C(10) \equiv 10^{7}$ (Mod 143)

$\equiv 10^{3}.10^{3}.10$ (Mod 143)

$\equiv 142.142.10$ (Mod 143)

$\equiv 1.10$ (Mod 143)

$\equiv 10$

$C(29) \equiv 29^7 \pmod{143}$

$\equiv 29^3.29^3.29 \pmod{143}$

$\equiv 79.79.29 \pmod{143}$

$\equiv 92.29 \pmod{143}$

$\equiv 94$

$C(14) \equiv 14^7 \pmod{143}$

$\equiv 14^3.14^3.14 \pmod{143}$

$\equiv 27.27.14 \pmod{143}$

$\equiv 14.14 \pmod{143}$

$\equiv 53 \pmod{143}$

$C(18) \equiv 18^7 \pmod{143}$

$\equiv 18^3.18^3.18 \pmod{143}$

$\equiv 112.112.18 \pmod{143}$

$\equiv 112.14 \pmod{143}$

$\equiv 138 \pmod{143}$

$C(12) \equiv 12^7 \text{ (Mod 143)}$

$\equiv 12^3.12^3.12 \text{ (Mod 143)}$

$\equiv 12.12.12 \text{ (Mod 143)}$

$\equiv 12 \quad \text{(Mod 143)}$

Codificando toda a palavra, obtemos a seguinte sequencia de blocos:

22-10-94-53-22-10-94-138-12-22

Esta sequencia de números é a palavra codificada.

4.4 – PROCESSO DE DECODIFICAÇÃO

Com a chave privada que é o par (120,103), começamos o processo de decodificação da mensagem recebida. Para o cálculo de D(C) é utilizado a seguinte fórmula:

D(C) $\equiv C^d$ **(mod n),** onde $0 \le D(C) \le n$ e D(C) é o resto da divisão de C^d por n.

Onde denotaremos D(C) de bloco decodificado

D(C) $\equiv C^d$ **(mod n)**

$D(C) \equiv 22^{103} \text{ (Mod 143)}$

$\equiv 22^{2^6}.22^{2^5}.22^{2^2}.22^2.22 \text{ (Mod 143)}$

$\equiv 22.55.22.55.22 \text{ (Mod 143)}$

$\equiv 22^3.55^2 \text{ (Mod 143)}$

$\equiv 22^3.22 \text{ (Mod 143)}$

$\equiv 22^4 \text{ (Mod 143)}$

$\equiv 22$

$22^{2^1} \equiv 55 (\text{mod} 143)$

$22^{2^2} \equiv (22^2)^2 \equiv (55)^2 \equiv 22 (\text{mod} 143)$

$22^{2^3} \equiv (22^{2^2})^2 \equiv (22)^2 \equiv 55 (\text{mod} 143)$

$22^{2^4} \equiv (22^{2^3})^2 \equiv (55)^2 \equiv 22 (\text{mod} 143)$

$22^{2^5} \equiv (22^{2^4})^2 \equiv (22)^2 \equiv 55 (\text{mod} 143)$

$22^{2^6} \equiv (22^{2^5})^2 \equiv (55)^2 \equiv 22 (\text{mod} 143)$

$D(10) \equiv 10^{103} \text{ (Mod 143)}$

$\equiv 10^{2^6}.10^{2^5}.10^{2^2}.10^2.10 \text{ (Mod 143)}$

$\equiv 133.100.133.100.10 \text{ (Mod 143)}$

$\equiv 133^2.100^2.10 \text{ (Mod 143)}$

$\equiv 100.133.10 \text{ (Mod 143)}$

38

$\equiv 10^3.133 \pmod{143}$

$\equiv 142.133 \pmod{143}$

$\equiv 10$

$D(94) \equiv 94^{103} \pmod{143}$

$\equiv 94^{2^3}.94^{2^5}.94^{2^2}.94^2.94 \pmod{143}$

$\equiv 42.113.42.113.94 \pmod{143}$

$\equiv 42^2.113^2.94 \pmod{143}$

$\equiv 48.42.94 \pmod{143}$

$\equiv 14.94 \pmod{143}$

$\equiv 29$

$D(53) \equiv 53^{103} \pmod{143}$

$\equiv 53^{2^6}.53^{2^5}.53^{2^2}.53^2.53 \pmod{143}$

$\equiv 27.92.27.92.53 \pmod{143}$

$\equiv 27^2.92^2.53 \pmod{143}$

$\equiv 14.27.53 \pmod{143}$

$\equiv 92.53 \pmod{143}$

$\equiv 14$

$D(138) = 138^{103} \pmod{143}$

$\equiv 138^{2^6}.138^{2^5}.138^{2^2}.138^2.138 \pmod{143}$

$\equiv 53.14.53.25.138 \pmod{143}$

$\equiv 53^2.14.25.138 \pmod{143}$

$\equiv 92.14.25.138 \pmod{143}$

$\equiv 1.25.138 \pmod{143}$

$\equiv 18$

$D(12) \equiv 12^{103} \pmod{143}$

$12^{2^1} \equiv 1 \pmod{143}$

$12^{2^2} \equiv (12^2)^2 \equiv (1)^2 \pmod{143}$

$12^{2^3} \equiv (12^{2^2})^2 \equiv (1)^2 \pmod{143}$

$$\equiv 12^{2^6}.12^{2^5}.12^{2^2}.12^2.12 \pmod{143}$$
$$\equiv 1.1.1.1.12 \pmod{143}$$
$$\equiv 12$$

39

Evidentemente, o módulo escolhido no exemplo é muito pequeno para oferecer qualquer segurança real, mas foi escolhido apenas a título de exemplificação. Quando se trata de um exemplo real, devem ser escolhidos números primos maiores para que o algoritmo seja seguro.

As contas de exponenciação são grandes para serem feitas à mão, por isso, devem ser feitas utilizando um pacote de computação algébrica. Existem muitos programas matemáticos de uso geral que lidam bem com aritmética modular. Vários deles comerciais, como o Maple e o Mathematica, alguns gratuitos e outros ainda de código livres.

4.5 - SEGURANÇA DO RSA

A segurança do RSA esta baseada na grande dificuldade computacional de fatorar números grandes como elucidado anteriormente, portanto às *chaves* são geradas matematicamente através do produto de dois números primos grandes. Mesmo que se tenha esse produto (que faz parte da chave publicada), a segurança ainda esta garantida devido à grande dificuldade de se fatorá-lo e obter os primos, essências para o algoritmo.

Essencialmente, a única forma de quebrar o RSA é fatorar o inteiro n. Logo, se um matemático conseguisse fatorar n = p. q, então seria possível calcular o valor de φ(n) e assim, o valor do inverso multiplicativo módulo m, descobrindo a *chave privada*.

Até o momento ninguém conseguiu descobrir um algoritmo eficiente de fatoração, em um computador clássico, que funcione em tempo polinomial. Contudo, acredita-se que quebrar o RSA e fatorar *n* seja um problema equivalente, apesar de não ter sido demonstrado.

CONCLUSÃO

A partir deste estudo é possível concluir que a *criptografia* sempre foi e será uma necessidade de todos. Desde os povos antigos até os tempos atuais a *criptografia* é uma maneira segura de se enviar mensagens, informações secretas, sem que outras pessoas compreendessem o seu conteúdo.

Criadores e decifradores estendem uma guerra infinita, pois cada vez mais temos a necessidade de guardar dados importantes. Este estudo descreve os princípios gerais dos sistemas de *chaves* e comprova que eles funcionam com base em funções matemáticas que envolvem conhecimentos na área de Teoria dos Números. Para dar embasamento e fundamentação, aborda alguns tópicos da área de Teoria dos Números como: Divisibilidade, Máximo Divisor Comum, Algoritmo da Divisão, Algoritmo de Euclides, Congruência e Função de Euler, para desenvolvimento da base Matemática na Criptografia.

REFERÊNCIAS BIBLIOGRÁFICAS

- LUCCHESI, Cláudio. **Introdução à criptografia computacional**. Campinas: UNICAMP, 1986
- COUTINHO, S. C. **Números inteiros e criptografia RSA.** Rio de Janeiro, IMPA/SBM 2000.
- SINGH, S. **O livro dos códigos.** Editora Record, 2007.
- **Wikipédia**, a enciclopédia livre.
- http://en.wikipedia.org/wiki/World_War_II_cryptography
- http://en.wikipedia.org/wiki/Enigma_machine
- www.numaboa.com.br/criptografia
- Lemos, Manoel. **Criptografia, Números Primos e Algoritmos.** Rio de Janeiro, IMPA
- Shokranian, Salahoddin, **Criptografia para iniciantes.** Brasília: UNIVERSIDADE DE BRASILIA, 2005
- JURKIEWIEZ, Samuel. **Divisibilidade e Números Inteiros** – OBMEP 2006