

ESCOLA SUPERIOR ABERTA DO BRASIL – ESAB

LATO SENSU EM REDES DE COMPUTADORES

OUTRAS APOSTILAS EM:
www.projetoderedes.com.br

JOCIMAR FERNANDES

**CRIPTOGRAFIA E MODELO CRIPTOGRÁFICO DO
SISTEMA INFORMATIZADO DE ELEIÇÕES DO BRASIL**

VITÓRIA - ES

2007

JOCIMAR FERNANDES

**CRIPTOGRAFIA E MODELO CRIPTOGRÁFICO DO
SISTEMA INFORMATIZADO DE ELEIÇÕES DO BRASIL**

Trabalho de Conclusão de Curso de Pós-Graduação Lato Sensu em Redes de Computadores apresentado a ESAB – Escola Superior Aberta do Brasil, sob orientação da Profª Msc. Beatriz Gobbi.

VITÓRIA - ES

2007

JOCIMAR FERNANDES

**CRIPTOGRAFIA E O MODELO CRIPTOGRÁFICO DO
SISTEMA INFORMATIZADO DE ELEIÇÕES DO BRASIL**

BANCA EXAMINADORA

Profª Msc. Beatriz Gobbi

ORIENTADORA

Professor 2

Professor 3

Vitória-ES, ___ de _____ de ____

DEDICATÓRIA

Dedico esse trabalho a todos os que me apoiaram e acreditaram no meu esforço e dedicação, principalmente minha família, minha mãe que sempre se dedicou aos filhos.

Minha esposa, pela persistência e perseverança sempre fazendo o impossível para possibilitar a continuação dos meus estudos, meus filhos pelo aceite de minha ausência como pai na família.

Graças a vocês, mais um degrau de minha vida foi realizado e que essa obra possa servir de respaldo para futuras pesquisas e continuidade de meus estudos.

AGRADECIMENTOS

Agradeço a Deus em primeiro lugar, que me deu a oportunidade de iniciar este curso e forças para concluí-lo.

Meus amigos: Cláudio Guimarães e Elizabeth Martins, Darly Anacleto dos Vasconcelos e Odalva, meus tios José Louzada e Zélia, a todos os colaboradores da ESAB – Vila Velha - ES, em especial, minha orientadora Prof^a Msc. Beatriz Gobbi, com os quais tive a oportunidade e o prazer de aprender sobre este novo paradigma de aprendizagem.

EPÍGRAFE

“A felicidade não está na partida e nem na chegada, mas na travessia”.

Guimarães Rosa

RESUMO

A segurança de dados através da criptografia, mostrando que a cifragem de dados é um dos meios que possibilita a privacidade nos meios de comunicação, e que devem ser estabelecidos por uma política de segurança adotada pelas empresas, vem sendo largamente utilizadas. Este trabalho explana os métodos de criptografia de substituição e transposição além dos modelos algoritmos de criptografia simétrico e assimétrico. Será discutido ainda os conceitos de chaves públicas e privadas e a utilização destas pelos algoritmos de criptografia no processo de cifragem e decifragem de mensagens. Serão discutidos ainda, detalhes dos algoritmos DES, Triple-DES, AES, IDEA, RC5, e RSA. Depois de esclarecidos os detalhes da Criptografia, será apresentada a pesquisa sobre o modelo criptográfico utilizado no SIE (Sistema Informatizado de Eleições do Brasil). Finalmente, serão discutidas duas ferramentas de criptografia, o PGP Desktop, amplamente utilizadas nos mercados nacional e internacional e o Webcry que vem despontando como uma potencial ferramenta para criptografia.

Palavras-Chave: Algoritmo, Criptografia, Chave Pública, Chave Privada, Segurança, SIE.

ABSTRACT

The cryptography utilization on the enhancement of information security has been widely used today. Its use, established by the companies' security policies, shows an improvement concerning the privacy on systems that use telecommunication channels to transfer important data. This work examines the cryptography's substitution and transposition methods, and the symmetric and asymmetric cryptography algorithms. It also discusses the concepts of public and private keys, the importance of its sizes in cryptography algorithms, as well as how the previously mentioned algorithms utilize those keys in ciphering and deciphering messages. Moreover, the paper brings details about the algorithms DES, Tripe-DES, AES, IDEA, RC5, and RSA, and presents the cryptographic model used in the SIE (Sistema Informatizado de Eleições do Brasil) – Brazilian's Elections Computerized System. Finally, the study discusses two cryptography tools: the PGP Desktop, broadly used on both the national and international markets; and the WebCry 2.0, which has been presenting itself as a prospective cryptography tool.

Keywords: *Cryptography; Algorithm; Public Key; Private Key; Security; SIE.*

SUMÁRIO

1. INTRODUÇÃO	14
2. CONCEITOS DE SEGURANÇA E CRIPTOGRAFIA.....	15
2.1 Conceitos básicos.....	15
2.2 Política de segurança.....	15
2.3 Modelos de segurança	16
2.3.1 Modelos discricionários	16
2.3.2 Modelos obrigatórios	17
2.3.3 Modelos baseados em papéis.....	17
2.4 Fundamentos da criptografia.....	19
2.5 A importância da criptografia.....	20
2.6 Termologia da criptografia	21
2.7 A importância da chave dentro da criptografia	22
2.8 Chaves simétrica e assimétrica.....	23
2.9 Métodos de criptografia	25
2.9.1 Cifra de Substituição.....	25
2.9.2 Cifras de transposição.....	26
2.9.3 Cifras de uso único.....	28
2.10 Métodos de criptografia do SIE (SISTEMA INFORMATIZADO DE ELEIÇÕES)	28
3. ALGORITMOS DE CRIPTOGRAFIA.....	31
3.1 Algoritmos de chave simétrica.....	31
3.1.1 DES – <i>Data Encryption Standard</i>	32
3.1.2 Triple-DES	34
3.1.3 AES – <i>Advanced Encryption Standart</i>	35
3.1.4 IDEA - <i>International Data Encryption Algorithm</i>	39
3.1.5 RC5 - <i>Rivest Cipher</i>	42
3.2 Algoritmos assimétricos.....	44
3.2.1 RSA.....	45
4. FERRAMENTAS DE CRIPTOGRAFIA	49
4.1 WEBCRY 2.0.....	49
4.1.1 Funcionamento da WebCry 2.0.....	50
4.2 PGP – PRETTY GOOD PRIVACY.....	52
4.2.1 Funcionamento do PGP	52
CONCLUSÃO	57
REFERÊNCIAS	58

LISTA DE FIGURAS

Figura 2.4.1 – Processos criptográficos	19
Figura 2.8.1 – Modelo simétrico de criptografia.....	24
Figura 2.8.2 – Modelo assimétrico de criptografia	24
Figura 2.9.2.1 – Exemplo de cifra de transposição	26
Figura 3.1.1.1 – Espaço geral de algoritmo de DES.....	33
Figura 3.1.3.1 – Algoritmo de cifragem do AES.....	37
Figura 3.1.3.2 – Algoritmo de decifragem	38
Figura 3.1.4.1 – Esquema do algoritmo IDEA.....	41
Figura 3.2.1.2 – Exemplo de cifragem com RSA.....	47
Figura 3.2.1.3 – Exemplo de decifragem com RSA	47
Figura 4.1.1 – Tela principal do WebCry 2.0	50
Figura 4.1.2 – Tela de entrada de dados a ser criptografado	51
Figura 4.2.1.1 – Criando key no PGP	53
Figura 4.2.1.2 – Tela de ilustração PGP	54

LISTA DE TABELAS

Tabela 3.1.4.1 – Operação básica do IDEA	39
Tabela 3.2.1.1 – Valores dos caracteres para o exemplo RSA	46

LISTA DE ABREVIATURAS

BIOS - Basic Input Output System
BU - Boletim de Urna
CD - Compact Disk
CMOS - Complementary Metal Oxide Semiconductor
UE - Urna Eletrônica
EEPROM - Electrically Erasable Programmable ROM
FC - Cartão Fhash Carga
FI - Cartão Fhash Interno
FTP - File Transfer Protocol
FV - Cartão Flash Votação
HD - Hard Disk
IBM - International Business Machines
ICMP - Internet Control Message Protocol
IP - Internet Protocol
ISO - International Organization for Standardization
LAN - Local Area Network
MIT - Massachusetts Institute of Technology
NFS - Network File System
NBS – National Bureau of Standards
NIST - National Institute of Standards and Technology
NSA – National Security Agency
OSI - Open System Interconnection
PGP - Pretty Good Privacy
PPP - Point-to-Point Protocol
PS2 - Personal System 2
SIE - Sistema Informatizado De Eleições
SIS - Subsistema de Instalação e Segurança

SO - Sistema Operacional
SVI - Seção do Voto Informatizado
TSE - Tribunal Superior Eleitoral
TRE - Tribunal Regional Eleitoral
USB - Universal Serial Bus
URL - Universal Resource Locator
VLAN - Virtual Local Area Network
WAN - Wide Area Network

CAPÍTULO I – INTRODUÇÃO

A *internet* tem sido um grande meio de comunicação disponível para o mercado de hoje, nele se concentra inúmeros usuários que o utilizam para enviar e receber informações no dia a dia.

Um grande problema que existente, é devido ao fator de não ter como filtrar o tipo de usuário, ou seja, enquanto alguns trabalham, há sempre alguém interessado em capturar uma informação sigilosa para obter vantagem ou só pelo simples fato de danificar o conteúdo da mensagem.

Devido a este fator, nasce à necessidade de se criar um meio que possa garantir a privacidade de uma mensagem ou informação ate que ela chegue ao seu destinatário.

O objetivo desta pesquisa é mostrar os recursos necessários para prover a segurança da informação, utilizando a criptografia para garantir a privacidade dos dados, apresentando conceitos, algoritmos simétricos, assimétricos e ferramentas criptográficas e analisando o modelo de criptografia utilizado pelo SIE - Sistema Informatizado de Eleições do Brasil.

A criptografia de dados tem como objetivo principal “prover segurança”, mantendo a privacidade de uma mensagem desde o envio até o recebimento. A criptografia proporciona a confiança de que, mesmo que a mensagem possa ser capturada por um intruso não autorizado, o conteúdo fique intacto.

O capítulo 2 está centrado nos conceitos de segurança e termologias que envolvem a criptografia de dados, dando uma visão geral sobre a mesma. Será pesquisado o modelo de criptografia utilizado pelo SIE - Sistema Informatizado de Eleições do Brasil.

No capítulo 3 detalham-se os algoritmos simétricos e assimétrico de criptografia, destacando os algoritmos com as funções de cifrar e decifrar uma mensagem.

Já no capítulo 4 serão apresentadas ferramentas criptográficas, e uma apresentação das ferramentas Webcry 2.0 e PGP *Desktop*.

CAPÍTULO II - CONCEITOS DE SEGURANÇA E CRIPTOGRAFIA

2.1. Conceitos básicos

O termo segurança é usado com o significado de minimizar a vulnerabilidade de bens (qualquer coisa de valor) e recursos. Vulnerabilidade é qualquer fraqueza que pode ser explorada para se violar um sistema ou as informações que ele contém.

A segurança está relacionada à necessidade de proteção contra o acesso ou manipulação, intencional ou não, de informações confidenciais por elementos não autorizados, e a utilização não autorizada do computador ou de seus dispositivos periféricos. A necessidade de proteção deve ser definida em termos das possíveis ameaças e riscos e dos objetivos de uma organização, formalizados nos termos de uma política de segurança. (SOARES; LEMOS e COLCHER, 1995, p.448):

2.2. Política de segurança

Uma política de segurança de sistemas é definida como um conjunto de diretrizes, normas e procedimentos, os quais estabelecem os limites de operação dos usuários. A política de segurança é feita sob medida para um sistema específico e não para uma classe geral de sistemas.

As diretrizes ditadas em uma política de segurança indicam o que cada componente do sistema (usuários, máquinas, etc.) pode ou não pode fazer. As normas indicam o que cada componente está habilitado a fazer e como deverá ser feito. Portanto, são ditados os procedimentos que devem ser tomados para cada estado do sistema, onde o sistema poderia sair desde um estado normal de funcionamento até um estado de emergência após algum evento inesperado ou malicioso.

Segundo Melo (2003), as políticas de segurança de sistemas diferem em três ramos: segurança física, segurança gerencial e segurança lógica, observados a seguir.

- **Política de segurança física** – preocupa-se em proteger o meio físico em que opera o sistema. São medidas definidas contra desastres como: incêndio, alagamento, terremoto, etc; e também são definidas medidas para proteger o acesso físico ao provedor do sistema, fornecendo meios de proibir o acesso de pessoas não autorizadas.

- **Política de segurança gerencial** – preocupa-se com o ponto vista organizacional, definindo os processos que devem ser tomados para seleção de pessoal, e até definindo os processos para criação e manutenção das próprias políticas de segurança.

- **Política de segurança lógica** - é a mais habitual e bastante utilizada no dia-a-dia. Esta política define quais usuários terão direito de acesso ao sistema e quais são os direitos que cada usuário possuirá. São aplicados aqui dois conceitos: a autenticação, onde um usuário necessita se identificar ao sistema para que possa obter acesso ao recurso; e a autorização, onde o usuário precisa provar que possui direitos sobre o recurso o qual ele deseja acessar.

2.3. Modelos de segurança

Os modelos de segurança são comumente confundidos com mecanismos de segurança. Um modelo de segurança é uma expressão, muitas vezes, formal de uma classe de políticas de segurança, abstraindo detalhes e concentrando-se em conjunto de comportamentos que são utilizados como base para defini-las de políticas de segurança.

Em seu trabalho, Melo (2003) afirma que os modelos de segurança são divididos em três tipos básicos: discricionários (*discretionary*), obrigatórios (*mandatory*) e os modelos baseados em papéis (*roles*).

2.3.1. Modelos discricionários

Os modelos discricionários garantem o acesso de usuários às informações com base na identidade do usuário e nas autorizações que determinam, para cada usuário (ou grupo de usuários) e para cada objeto do sistema, a forma de acesso que o usuário está autorizado a realizar sobre o objeto.

O modelo de matriz de acesso é um modelo conceitual discricionário, o qual especifica os direitos que cada sujeito possui sobre cada objeto do sistema. Os principais são tipicamente usuários, processos ou máquinas representando o usuário, podendo assumir diferentes sujeitos em diferentes ocasiões.

2.3.2. Modelos obrigatórios

Enquanto os modelos discricionários são direcionados a definição, modelagem e controle do acesso à informação, os modelos obrigatórios (*mandatory models*), ou não-discricionários, também se preocupam com o fluxo de informações no sistema. Nos modelos obrigatórios, classes de segurança são atribuídas aos objetos e aos sujeitos, as quais são designadas como rótulos de segurança (*Security label*). Para os objetos o rótulo é chamado de classificação, enquanto que para os sujeitos o rótulo é chamado de habilitação (*clearance*). A classificação representa a sensibilidade dos dados rotulados, enquanto que a habilitação representa a confiança no sujeito em não revelar informações sensíveis a outros sujeitos.

2.3.3. Modelos Baseados em Papéis

Modelos baseados em papéis (*roles-based models*) restringem o acesso dos principais às informações de acordo com as atividades que estes desempenham no sistema. Os papéis podem ser definidos como um conjunto de ações e responsabilidades associadas com uma atividade de trabalho em particular. Logo, ao invés de especificar o que cada principal está autorizado a fazer, essa autorização é dada aos papéis. Um principal que desempenha um papel só estará apto a realizar ações no sistema de acordo com as permissões que o papel possui. Em diferentes situações um principal poderá assumir diferentes papéis e também um papel pode ser assumido por diferentes principais, às vezes simultaneamente.

As vantagens apresentadas pelo modelo baseado em papéis são:

- **Gerência de autorização** - As especificações de autorização são divididas em duas partes, a associação de direitos de acesso a papéis e associação destes papéis aos principais, deixando assim o gerenciamento de segurança mais simplificado, por exemplo, no caso de um principal ter que desempenhar um novo papel na organização.

- **Hierarquia de papéis** - Em muitos tipos de aplicações existe uma hierarquia natural de papéis baseada nos princípios da generalização e especialização, permitindo assim que permissões sejam herdadas e compartilhadas.

- **Privilegio mínimo** - Papéis permitem que um principal trabalhe com o mínimo privilégio exigido por uma determinada tarefa, garantindo assim, que um principal fará uso dos benefícios e privilégio somente para realizar as tarefas especificadas pelos papéis.

- **Separação de tarefas** (*duties*) - Tal propriedade parte do princípio que a nenhum principal deve ser dado privilégios o bastante além do que o mesmo possa utilizá-los, de forma maliciosa, em benefício próprio. Por exemplo, a pessoa que autoriza os pagamentos de salário não deveria ser a mesma pessoa que os efetua. E ainda, a efetuação dos pagamentos não poderá ocorrer sem que antes ocorra a autorização dos mesmos.

- **Classificação dos objetos** - No modelo baseado em papéis a classificação dos principais ocorre de acordo com as tarefas que cada um poderá executar no sistema. Analogamente, tal classificação poderia ser dada aos objetos do sistema. Assim, as autorizações de acesso dos papéis fariam sobre as classes de objetos e não em objetos específicos.

2.4. Fundamentos da criptografia

Através da necessidade de enviar informações sensíveis por meios de comunicação não confiáveis, ou seja, em meios onde não é possível garantir que um intruso possa interceptar o fluxo de dados para leitura¹, ou para modificá-la² surgiu a criptografia. A palavra criptografia vem do grego (kriptós, que significa escondido, oculto) e (grápho significa grafia, escrita).

Como apresentado por Schneier em Uchoa (2003), a criptografia é a arte e ciência de manter mensagens seguras. Ela envolve dois processos: 1 - criptografar (ou cifrar) uma mensagem **M**, transformando-a em um texto cifrado **C**, e 2 - decifrar (ou decriptografar) **C**, obtendo novamente a mensagem **M**, como ilustrado na figura 2.4.1.

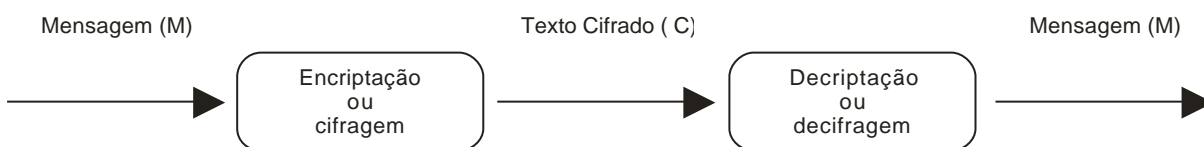


Figura 2.4.1 - Processos criptográficos, Uchoa (2003).

Conforme citado por Moreno (2005), pode-se criptografar informações basicamente por meios de códigos ou de cifras. Os códigos protegem as informações trocando partes destas por códigos predefinidos.

Todas as pessoas autorizadas a ter acesso a uma determinada informação devem conhecer os códigos utilizados.

Já as cifras são técnicas nas quais as informações são cifradas por meio da transposição e/ou substituição das letras da mensagem original. Assim, as pessoas autorizadas podem ter acesso às informações originais conhecendo o processo de cifragem.

¹ Intruso Passivo

² Intruso Ativo

Segundo Tanenbaum (1997), a arte de criar mensagens cifradas é chamada de criptografia, e de solucionar mensagens cifradas é chamada de criptoanálise (*cryptanalysis*). Já a coletividade é chamada de criptologia (*cryptology*).

2.5 A importância da criptografia

Com a evolução dos meios de comunicação, a criptografia tornou-se um fator extremamente necessário para se garantir confiabilidade com relação a envio e recebimento de informações. Sendo que em alguns países é de uso obrigatório, como caso da Suécia em que todas as comunicações de dados devem ser criptografada por ordem do governo, como foi citado em Arnett (1997).

A chegada da era da comunicação sem fio em 1901, trouxe a vantagem de uma comunicação de longa distancia através de um sistema aberto, isto aumentou a necessidade da utilização da criptografia de dados. Um outro passo marcante foi à chegada da *internet*.

A criptografia é necessária para manter a privacidade de dados de pessoas, empresas e sistemas. Muitos transtornos podem acontecer se uma pessoa não autorizada tiver acesso a dados pessoais, como: contracheque, saldo bancário, faturas do cartão de credito, diagnóstico de saúdes, senhas bancarias e principalmente um resultado de pleito eleitoral.

A segurança passou a ser mais que um ato de fechar portas de escritórios ou de cofres de empresas. Devido ao grande número de dados estratégicos que possuem dados como: previsão de vendas, detalhes técnicos de produtos, resultados de pesquisas e arquivos e arquivos pessoais são informações valiosas, às quais se alguma empresa concorrente tiver acesso de forma indevida, tal fato pode acarretar sérios problemas.

A criptografia de modo algum pode ser considerada como a única ferramenta para assegurar a segurança de dados, nem resolverá todos os problemas de segurança.

Baseado em Moreno (2005), a proteção por criptografia é uma das soluções praticas para proteger informações sigilosas. Independentemente do algoritmo criptográfico utilizado, sempre ocorrerá transformações de um texto legível em um ilegível. Mesmo que o invasor obtenha o conteúdo de um arquivo, esse será ilegível. Para ter acesso à informação original, o invasor terá que resolver um problema matemático de difícil solução. A criptografia pode adicionar também maior segurança ao processo de identificação de pessoas, criando identidades digitais fortes.

2.6. Termologia da criptografia

A criptografia utiliza um conjunto de métodos ou técnica que serve para transformar um texto compreensível, denominando texto original ou texto claro, em uma informação transformada, chamada de texto cifrado ou texto código ou simplesmente cifra que tem a aparência de um texto gerado aleatoriamente incompreensível.

O ato de transformar dados para uma forma ilegível é denominado cifra ou cifragem, e procura garantir a privacidade, mantendo a informação escondida de pessoas não autorizadas, mesmo que estas possam visualizar os dados criptografadas. O processo inverso é conhecido por decifrar.

Para que estes processos de cifragem e decifragem ocorram é necessário informações confidenciais, denominadas chaves. Tipos de chave existentes:

- ? Chave simétrica: Também conhecidas como chave única, utiliza a mesma chave para a cifragem como para a decifragem.
- ? Chave assimétrica: Também chamadas de algoritmos de chave pública, utilizam chaves diferentes para cifrar e decifrar os dados. Em um sistema de chave assimétrica cada pessoa tem duas chaves: uma chave pública que pode ser divulgada e outra privada que deve ser mantida em segredo.

Como definido por Moreno (2005), o algoritmo de criptografia é uma seqüência de procedimentos que envolvem uma matemática capaz de cifrar e decifrar dados sigilosos.

A execução de algoritmo criptográfico pode ser por um computador, por *hardware* dedicado e por um humano. Em todas as situações, o que diferencia um do outro é a velocidade de execução e a probabilidade de erros.

Um ato praticado é a publicação do algoritmo, que permite o especialista de criptografia se livrar de ter que consultar inúmeros criptólogos ansiosos por decodificar o sistema para que possam publicar artigos demonstrando suas esperteza e inteligência.

Um algoritmo recebe o título de muito bom após cinco anos de publicação do algoritmo, e não sendo comprovada nem uma decodificação por parte dos especialistas que tentaram ao longo deste período. Um outro motivo para publicar um algoritmo é o simples fato de poder examinar as fraquezas dele.

Um bom método de criptografia deve garantir que seja, senão impossível, pelo menos muito difícil que um intruso recupere, a partir de um texto criptografado e do conhecimento sobre o método de criptografia, o valor das chaves. Sendo isto verdade, a confidencialidade do texto transmitido é garantida enquanto as chaves mantiverem secretas (SOARES,1995, p.70).

A chave na criptografia computadorizada, é um numero ou conjunto de números. A chave fornece mais proteção à informação cifrada, pois para poder decifrar é necessário que o receptor possua a chave correta, que é única, para alimentar o algoritmo e somente depois ter acesso à informação.

2.7. A importância da chave dentro da criptografia

De acordo com Tanenbaum (1997), a importância da chave dentro da criptografia, deve-se ao sigilo e cuidado para que ninguém desautorizado possa obter de forma indevida e colocando em risco todo o algoritmo. Um outro fator importante é o tamanho, por exemplo, uma chave que possui um tamanho de dois dígitos dá uma margem de 100 possibilidades, e uma outra chave com seis dígitos tem uma margem de um milhão de possibilidades.

Através da chave é possível minimizar a preocupação com o algoritmo utilizado no sistema de criptografia, ou seja, é mais fácil proteger uma chave do que todo um algoritmo de criptografia. Uma outra opção que facilita é o fato de poder utilizar chaves diferentes para informações diferentes, fazendo com que, mesmo que um intruso possa descobrir uma chave não consiga obter todas as informações.

Um motivo para se cuidar da chave, dar-se ao fato de que é possível construir sistemas criptográficos nos quais o algoritmo é completamente conhecido e seguro, tendo em vista que possíveis invasores precisam conhecer a chave para descobrir as informações. Um sistema criptográfico que utiliza chave é superior na segurança e confiabilidade, afinal os outros tipos de sistemas criptográficos confia apenas no segredo do algoritmo.

No processo de gerar chaves são utilizadas várias técnicas. Entre elas, destacam-se o uso de sistema gerador de números aleatório. Estes sistemas funcionam agrupando números de diferentes tipos de entrada imprevisível.

Uma chave deve ser formada por números ou caracteres alfanuméricos sem coerência alguma, ou seja, ela deve ser aleatória para dificultar a quebra da mesma.

2.8. Chaves simétrica e assimétrica

No modelo simétrico de criptografia, ou seja, na criptografia de chaves simétrica, os processos de cifragem e decifragem utilizam uma única chave tanto para o remetente como para o destinatário. Nos algoritmos simétricos, por exemplo, o DES, ocorre o chamado “problema de distribuição de chaves”. A chave tem de ser enviada para todos os usuários autorizados antes que as mensagens possam ser trocadas.

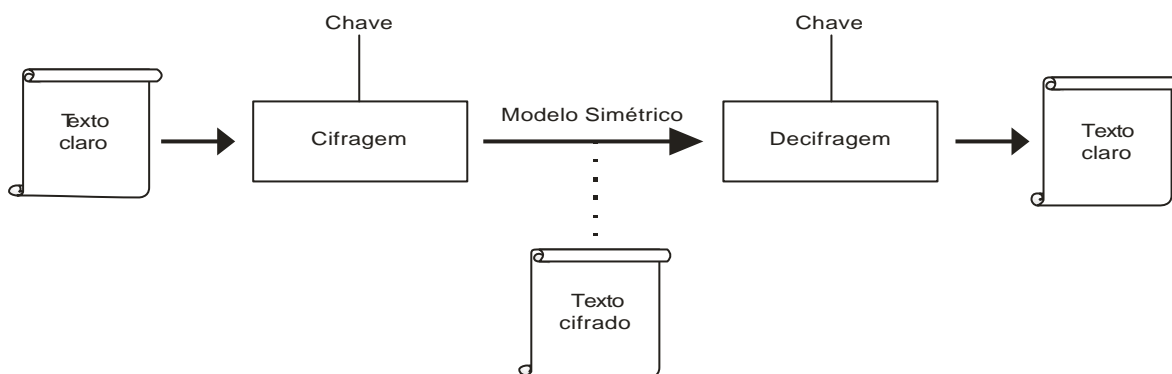


Figura 2.8.1 – Modelo simétrico de criptografia, Moreno (2005).

Como citado em Moreno (2005), a criptografia assimétrica contorna o problema da distribuição de chaves mediante o uso de chaves públicas. A criptografia de chaves pública foi inventada em 1976 por Whitfield Diffie e Martin Hellman, a fim de resolver o problema da distribuição de chaves.

Neste novo sistema, cada pessoa tem um par de chaves denominado chave pública e chave privada. A chave pública é divulgada, enquanto a chave privada é mantida em segredo.

Para mandar uma mensagem privada, o transmissor cifra a mensagem usando a chave pública do destinatário pretendido, que deverá usar a sua respectiva chave privada para conseguir recuperar a mensagem original.

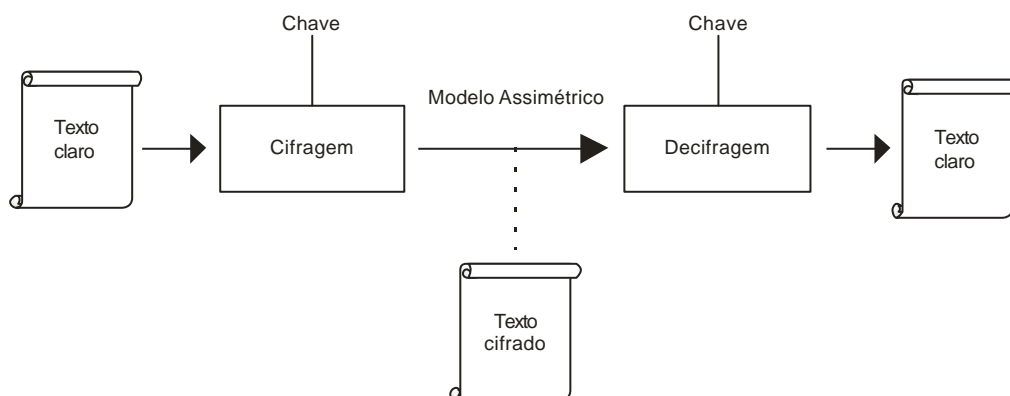


Figura 2.8.2 – Modelo assimétrico de criptografia, Moreno (2005).

2.9. Métodos de criptografia

Historicamente, os métodos de criptografia têm sido divididos em duas categorias: as cifras de substituição e as cifras de transposição.

2.9.1. Cifra de substituição

Na cifra de substituição, cada letra ou grupo de letra é substituído por outra letra ou grupo de letras, criando um tipo de disfarce. Este tipo de cifra é atribuída a Julio César, neste método, a se torna D, b se torna E, c se torna F e assim por diante.

Por exemplo, a palavra “**ataque**” passaria a ser **dwdtxh**. Generalizando, esta cifra permite que o alfabeto do texto cifrado seja deslocado **k** letras, neste exemplo acima o valor de **k** é três. Com isso **k** torna-se uma chave para o método genérico dos alfabetos deslocados em forma circular. Este sistema em geral é chamado de substituição mono alfabético.

A desvantagem de utilização deste método está quando é cifrado um volume pequeno de texto, que é quando a cifra apresenta sua facilidade de ser descoberta. Uma estratégia básica se beneficia das propriedades estatísticas dos idiomas. Tomando um exemplo citado por Tanenbaum (1996), em que ele disse que em inglês **e** é a mais comum, seguida de **t**, **o**, **a**, **i** etc. As combinações de duas letras, ou diagramas, mais comuns são **th**, **in**, **er**, **re** e **na**.

As combinações de três letras, ou trigramas, mais comuns são **the**, **ing**, **and** e **ion**. Um criptoanalista que esteja tentando decodificar uma cifra mono alfabética começaria contando as frequências relativas de todas as letras do texto cifrado. Depois disso, através de tentativas, ele atribuiria **e** à letra mais comum e **t** à próxima letra mais comum. Em seguida, verificaria os trigramas para encontrar um no formato **tXe**, o que poderia sugerir que **X** é **h**. Da mesma forma, se o padrão **thYt** ocorrer com frequência, provavelmente isso significa que **Y** representa **a**. Com essas informações, o criptoanalista poderá procurar por um trigrama com formato **aZW** que ocorra com frequência (muito provavelmente **and**). Fazendo estimativas em relação a diagramas, trigramas e letras mais comuns, e conhecendo os prováveis padrões de vogais e consoantes, o criptoanalista criaria um texto simples através de tentativas, letra por letra.

2.9.2. Cifras de transposição

Na cifra de transposição as letras são reordenadas, e não disfarçadas como no caso da cifra de substituição. Ela se baseia em uma chave que é uma palavra ou frase que não contém letras repetidas.

Na figura 2.9.2.1 mostra uma cifra de transposição muito comum, a transposição de colunas.

Neste exemplo, **MEGABUCK** é a chave. O objetivo é numerar as colunas de modo que a coluna 1 fique abaixo da letra chave mais próxima do início do alfabeto e assim pro diante.

O texto simples é escrito horizontalmente, em linhas, O texto cifrado é lido em colunas, a partir da coluna cuja letra da chave seja a mais baixa.

Para romper uma cifra de transposição o criptoanalista deve primeiro está ciente de que esta lidando com uma cifra de transposição, Examinando a frequência de E, T, A, O, I, N etc., fica fácil constatar se essas letras se encaixam no padrão normal para texto simples. Se houver correspondência, isso significa que a cifra é evidentemente uma transposição, pois nesse tipo de cifra cada letra é representada por ela mesma, mantendo intacta a distribuição de frequência.

M E G A B U C K								Texto simples:	
7	4	5	1	2	8	3	6	pleasetransferonemilliondollarsto	
p	l	e	a	s	e	t	r	myswissbankaccountsixtwo	
a	n	s	f	e	r	o	n		
e	m	i	l	l	i	o	n	Texto cifrado:	
d	o	i	l	a	r	s	t	AFLLSKSOSELAWAIATOOSSCTCLNMOMANT	
o	m	y	s	w	i	s	s		
b	a	n	k	a	c	c	o	ESILYNTWRNNTSOWDPAEDOBUEOERIRICXB	
u	n	t	s	i	x	t	w		
o	t	w	o	a	b	c	d		

Figura 2.9.2.1 - Exemplo de cifra de transposição, Tanenbaum (1997).

De acordo com Tanenbaum (1997), para se romper uma cifra de transposição, o criptoanalista deve primeiro estar ciente que está lidando com uma cifra de transposição. Examinando a frequência de **E, T, A, O, I, N** etc., fica fácil constatar se essas letras se encaixam no padrão normal para texto simples.

Se houver a correspondência, isso significa que a cifra é evidentemente de transposição, pois nesse tipo de cifra cada letra representa ela mesma.

A próxima etapa é fazer uma estimativa do número de colunas. Em muitos casos, uma palavra ou frase provável pode ser deduzida a partir do contexto da mensagem.

Por exemplo, suponha que o nosso criptoanalista tenha suspeitado de que a frase em texto *milliondollars* ocorre em algum lugar na mensagem.

Observe que os diagramas **MO, IL, LL, LA, IR** e **OS** ocorrem no texto cifrado, como um resultado do desdobramento dessa frase.

No texto cifrado, a letra **O** vem depois da letra **M** (ou seja, elas são verticalmente adjacentes na coluna 4), pois estão separadas na provável frase por uma distância igual ao tamanho da chave. Se tivesse sido usada uma chave de tamanho sete, teriam surgido os diagramas **MD, IO, LL, LL, IA, OR** e **NS**.

Na verdade, para cada tamanho de chave, é produzido um conjunto de diagramas diferentes no texto cifrado. Ao tentar encontrar diferentes possibilidades, muitas vezes o criptoanalista é capaz de determinar com facilidade o tamanho da chave. A última etapa é ordenar as colunas.

Quando o número de colunas **k** é pequeno, cada um dos **k(k-1)** pares de colunas pode ser examinado para que seja constatado se suas frequências de diagramas correspondem ao texto simples em inglês. O par que tiver a melhor correspondência será considerado como posicionado da forma correta. Em seguida cada uma das colunas restante é experimentada como sucessora desse par. A coluna cujas frequências de diagramas e trigramas proporcionem a melhor correspondência será experimentalmente considerada como correta. O processo inteiro continua até ser encontrada uma ordenação seqüencial.

2.9.3 Cifra de uso único

As chaves únicas são ótimas na teoria, mas já na aplicação deixa a desejar. As desvantagens estão nos seguintes aspectos:

- ? É que não podem ser memorizadas; com isso tanto o remetente quanto o destinatário tem que possuir a chave de forma escrita. E com isso existe a possibilidade de captura delas.
- ? Existe uma limitação da quantidade total de dados que podem ser transmitida, pois esta quantidade de dados esta relacionada ao tamanho da chave.
- ? Caso uma grande quantidade de dados seja descoberta por um espião, talvez o mesmo seja incapaz de transmiti-la de volta a matriz, por que a chave foi consumida.
- ? Um outro fator é a sensibilidade do método para caracteres perdidos ou inseridos. Se uma das partes ficarem fora de sincronismo, todos os dados a partir desse momento parecerão adulterados.

Esta técnica de cifragem depende de uma chave, uma *string* de *bits* aleatórios. Em seguida que o texto simples seja convertido em uma *string* de *bits*, utilizando, por exemplo, sua representação ASCII. Por fim, calcule OR exclusivo (XOR) dessas duas *strings*. Com isso o texto cifrado resultante não poderá ser violado porque, em uma amostra suficientemente grande de texto cifrado, cada letra ocorrerá com a mesma frequência, bem como diagrafa, cada trigrama e assim por diante.

2.10 Métodos de criptografia do SIE (Sistema Informatizado de Eleições)

Segundo a UNICAMP ⁽¹⁷⁾, a descrição do modelo de criptografia do SIE, utiliza dois algoritmos de cálculo de resumo criptográfico hash. Sendo utilizados para produzir, cada um, uma espécie de impressão digital de cada arquivo do sistema armazenado num arquivo como assinaturas, para posterior comparação com o recálculo das mesmas no momento da verificação, seguindo os passos abaixo:

O processo de compilação do código-fonte:

Para a compilação final dos programas da UE (urna eletrônica) é utilizado o mesmo ambiente comercial padrão empregado no desenvolvimento (Borland C versão 4.5), o que permitiria, em princípio, a reprodução do processo de compilação em instalações independentes. Neste processo, o TSE substitui rotinas fictícias de criptografia (usadas pela empresa contratada para testes de ciframento do BU- boletim de urna) pelas rotinas finais de criptografia, em código-objeto, desenvolvidas para uso específico na UE.

Os mecanismos de verificação de integridade de arquivos:

Após a compilação dos programas, é iniciada uma etapa de preparação de resumos criptográficos (funções hash) que ajudarão a verificar a integridade dos arquivos de programas e de dados durante o restante do processo da eleição.

Os algoritmos de resumo criptográfico (funções hash) utilizados são:

- ? **Message Digest 5 (MD5):** é universalmente adotado e de conhecimento público; fornece o resumo criptográfico de 128 bits a partir de um conjunto de dados de tamanho arbitrário; apesar de ser um algoritmo adequado para as aplicações em questão, sugere-se uma avaliação sobre a possibilidade de uso nas urnas atuais de um algoritmo mais robusto, como o SHA-1, por exemplo;
- ? **ASSINA:** é uma função não pública desenvolvida pela Microbase³, que implementa um resumo criptográfico de 256 bits; é usada principalmente para gerar o resumo e garantir a integridade e a autenticidade (já que não é pública) de um *conjunto de resumos criptográficos gerados pelo MD5*.

Estes dois algoritmos são empregados da forma descrita a seguir:

Usando o algoritmo convencional (MD5), é calculado um resumo criptográfico para cada arquivo da árvore de diretórios da aplicação de votação. Os nomes dos arquivos e seus resumos são gravados, um por linha, num arquivo com extensão do tipo CRC. Para prover um nível extra de segurança, é calculado também o resumo criptográfico (com o algoritmo ASSINA) de cada arquivo .CRC, o qual é guardado em um outro arquivo com extensão.SIG.

³ Microbase: Empresa desenvolvedora do Sistema Operacional VirtuOS, que funciona nas urnas eletrônicas brasileiras.

Estes resumos são verificados pelos programas executados durante a inseminação da UE e todas as vezes que ela sofrer uma inicialização (boot). Esta verificação também é realizada durante a execução de alguns programas que compõem o aplicativo de votação. Qualquer modificação feita em algum arquivo da UE que não seja acompanhada pela correspondente modificação dos arquivos .CRC e .SIG será detectada, já que os procedimentos de verificação recalculam os resumos e os comparam com aqueles que foram gravados nos diretórios na época da criação dos mesmos.

A combinação do uso das técnicas públicas e proprietárias de resumo criptográfico tornam muito difíceis o sucesso de qualquer tentativa de modificação posterior dos programas executáveis sem que tal tentativa seja detectada. Toda a segurança do mecanismo de verificação de integridade e autenticidade empregado se baseia no segredo do algoritmo de resumo criptográfico ASSINA responsável pelos 256 bits guardados nos arquivos .SIG.

O empacotamento e a transferência do software da UE:

O carregamento do software nas UEs (inseminação) é feito de forma descentralizada. Com a finalidade de preparar os programas da UE para serem enviados aos TREs, é realizado um processo de empacotamento no TSE com a ajuda de funcionários da empresa contratada. Este processo, consiste no agrupamento de todos os programas necessários para o funcionamento da UE, acrescidos de parte dos dados necessários (cadastro de eleitores, por exemplo). Outra parte dos dados precisa ser inserida em cada TRE de acordo com as candidaturas da região.

O pacote é cifrado e transmitido aos TREs e pólos de inseminação via rede (FTP) ou enviado em CD-ROM dos TREs aos pólos que não têm conexão à rede. Devido ao uso de criptografia sobre todo o conteúdo do pacote, é extremamente improvável que os programas possam ser substituídos ou alterados, desde que haja uma política apropriada de criação, distribuição e manutenção de senhas. O algoritmo de criptografia adotado nesta etapa é o IDEA-128, bastante documentado na literatura e considerado adequado para este propósito.

Além da proteção provida pela criptografia de todo o pacote, deve ser lembrado que cada arquivo nele contido está também protegido pelos resumos criptográficos descritos anteriormente (UNICAMP, 2002, p.25)

De acordo com UNICAMP ⁽¹⁷⁾, a ferramenta definida para utilização dos procedimentos de criptografia do SIE (Sistema Informatizado de Eleições) foi o PGP (*Pretty Good Privacy*) pelo motivo eficaz de seus códigos gerados e a confiabilidade de comunicação da ferramenta via intranet do TSE.

CAPÍTULO III – ALGORITMOS DE CRIPTOGRAFIA

Dentro da criptografia se destacam os algoritmos criptográfico que podem ser implementados tanto via *hardware* com *software*. Os algoritmos criptográficos, são denominados cifra, utilizam função matemática para criptografar ou decriptografar uma mensagem, em geral são duas função relacionadas, uma que será usada no processo de cifragem e outra na decifragem da mensagem.

Os algoritmos de criptográficos podem ser implementados em hardware (para se obter velocidade) ou em software (para se obter flexibilidade), embora a maior parte de nosso tratamento esteja relacionado aos algoritmos e protocolos, que são independentes da implementação real. (TANENBAUM, 1997, p.667)

Podem se classificar os algoritmos criptográficos por meio do tratamento dado às informações que serão processadas; Assim, têm-se os algoritmos de bloco e os algoritmos de fluxo.

Dentro da criptografia existem algoritmos criptográficos simétricos e assimétricos, a diferença entre um algoritmo e outro é a seguinte:

- ? **Algoritmo simétrico:** Quando a chave utilizada na encriptação da mensagem é a mesma utilizada na deciptação.
- ? **Algoritmo assimétrico:** Também chamados de algoritmos de chave pública, Utiliza duas chaves: Uma para criptografar e outra para decriptografar a mensagem graças a processos matemáticos, é possível escolher chaves de tal forma que o conhecimento de uma não signifique que a outra chave possa ser descoberta.

3.1. Algoritmos de chave simétrica

Em algoritmo criptográfico, o fato de ser simétrico aponta basicamente para a forma de utilização da chave de criptografia, ou seja, para duas pessoas que querem compartilhar uma informação sigilosa, através de criptografia simétrica, essas terão que compartilhar a chave de criptografia, pois esta será utilizada tanto para a cifragem dos dados, quanto para a decifragem dos dados.

Entre os algoritmos que utiliza chave simétrica, o DES é um dos mais famosos, é por este motivo começaremos por ele.

3.1.1. DES – *Data Encryption Standard*

O DES é um algoritmo de cifragem que foi mais utilizado no mundo. Por muitos anos, e para muitas pessoas, criptografia e DES foram sinônimas.

Em 15 de maio de 1973, durante o Governo de Richard Nixon, o NBS publicou carta solicitando formalmente propostas de algoritmos de criptografia para proteger dados durante transmissões e armazenamento. A primeira resposta da carta apareceu em 06 de agosto de 1974, quando a IBM apresentou um algoritmo candidato que ela havia desenvolvido inteiramente, e denominado *Lúcifer*.

Após avaliar o algoritmo com a ajuda da NSA, a NBS adotou o algoritmo *Lúcifer* com algumas modificações sob a denominação de DES em 15 de julho de 1977 como citado por TKOTZ ⁽¹³⁾.

O DES foi adotado rapidamente pela mídia não digital, como nas linhas telefônicas públicas. Com o passar de alguns anos, empresas como *International Flavors and Fragrances* começaram a utilizar o DES para proteger as transmissões por telefone e outros dados importantes para ela.

Um outro setor que adotou a cifra de DES foi o setor de informática para uso em produtos de segurança. Em sua forma original do DES, já não é mais tão segura; no entanto, em uma forma modificada ela ainda é útil.

O Algoritmo DES é composto de operações simples, como: permutações, substituições, XOR e deslocamentos.

No processo de criptografar dados, o DES divide a mensagem em blocos de 64 *bits* e retorna blocos de texto cifrado do mesmo tamanho. O algoritmo parametrizado por uma chave de 56 *bits*, tem 19 estágios distintos, sendo que o primeiro é uma transposição independente da chave no texto simples de 64 *bits*, e no ultimo estágio acontece o processo inverso dessa transposição. O penúltimo estágio troca os 32 *bits* mais à esquerda por 32 *bits* mais à direita.

Os 16 estágios restantes são iterações que cada bloco de 64 *bits* sofrerá. Porém cada uma delas com uma chave diferente. Antes de cada iteração, a chave é particionada em duas unidades de 28 *bits*, sendo cada uma delas girada à esquerda um número de *bits* que depende do número de iteração. Utiliza-se ainda um parâmetro *k* que é derivada da girada, pela aplicação de mais uma transposição de 56 *bits* sobre ela. E em cada rodada, um subconjunto de 48 *bits* dos 56 *bits* é extraído e permutado.

Baseado em Tanenbaum (1997), o algoritmo de DES foi projetado para permitir que a decodificação fosse feita com a mesma chave da codificação, uma propriedade necessária para algoritmo de chave simétrica. Porém as etapas são simplesmente executadas na ordem inversa. A figura 3.1.1.1 ilustra um esboço geral do funcionamento do algoritmo de DES.

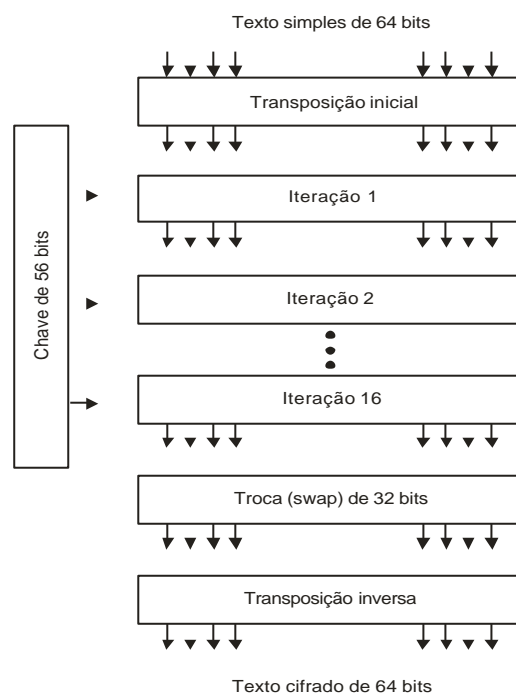


Figura 3.1.1.1 - Esboço geral do Algoritmo de DES, Tanenbaum (1997).

Existe uma técnica que é utilizada para tornar o DES mais forte, conhecida pelo nome de *witening*. Esta técnica consiste em operação XOR entre uma chave aleatória de 64 *bits* e cada bloco de texto simples, antes de sua entrega ao DES e depois uma operação XOR entre uma segunda chave de 64 *bits* e o texto cifrado resultante, antes de sua transmissão. O branqueamento pode ser removido com facilidade pela execução das operações inversas, para isto o receptor deve ter as duas chaves de branqueamento. Tendo em vista que essa técnica acrescenta efetivamente mais *bits* ao tamanho da chave.

Antes que o DES fosse adotado como padrão internacional, os criadores da criptografia de chaves pública, Martin Hellman e Whitfield Diffie, registraram algumas objeções quanto ao uso do DES como algoritmo de cifragem. Hellman escreveu a seguinte carta ao NBS dando sua opinião sobre a segurança do DES.

Baseado em TKOTZ ⁽¹³⁾, a carta apresentava uma preocupação com o fato de que o padrão de criptografia de dados proposto. Enquanto seria provavelmente seguro em relação a assaltos comerciais, poderia ser extremamente vulnerável a ataques efetuados por uma organização de inteligência.

Sobre a direção de John Gilmore do EFF, uma equipe construiu uma máquina que podia analisar todo o espaço de chaves de 56 *bits* de DES, e em 17 de julho de 1998, foi anunciado que havia conseguido quebrar uma chave de 56 *bits* em 48 horas. O computador que conseguiu tal proeza era chamado de *DES Key Search Machine*. E a única solução para isto, era criar um algoritmo com uma chave maior que pudesse ser quebrado com facilidade. Surgiu, então, o Triple-DES.

3.1.2. Triple-DES

Como citado por Moreno (2005), o algoritmo Triple-DES é apenas o DES com duas chaves de 56 *bits* aplicadas. Dada uma mensagem em texto claro, a primeira chave é usada para cifrar a mensagem em DES e a segunda chave, para decifrar o DES da mensagem cifrada. Como a segunda chave não é correta para decifrar, essa decifragem apenas embaralha ainda mais os dados. A mensagem duplamente embaralhada é, então, cifrada novamente com a primeira chave para se obter o texto cifrado final. Este procedimento em três etapas é denominado Triple-DES.

Com a utilização de duas chaves ate os criptográficos mais paranóicos concordaram que 112 *bits* seriam suficientes para aplicações comerciais durante algum tempo

3.1.3. AES – *Advanced Encryption Standard*

Com a evolução das maquinas, uma chave com apenas 56 *bits* já não fornecia a segurança necessária. Com isso um órgão do departamento de comercio dos Estados Unidos chamado NIST patrocinou um concurso de criptografia, com o intuito de conseguir um algoritmo para ser utilizado como o novo padrão.

O NIST estabeleceu como requisitos fundamentais:

- ? **Segurança forte:** o algoritmo projetado deve suportar ataques futuros.
- ? **Projeto simples:** facilitar a análise e certificação matemática da segurança oferecida pelo algoritmo.
- ? **Desempenho:** razoavelmente bom em uma variedade de plataformas, variando de *Smart Cards* a servidores.
- ? **Não serem patenteados:** os algoritmos devem ser de domínio público e esta disponível mundialmente.

O algoritmo vencedor foi o Rijndael de Joan Daemen e Vincent Rijmen, sendo que a decisão oficial anunciada no dia 2 de outubro de 2000. E a partir desta data o cifrador passou a se chamar de AES.

O AES utiliza a substituição, permutação e rodadas assim com o DES, porem o numero de rodadas depende do tamanho da chave e do tamanho do bloco, sendo 10 para cada chaves de 128 *bits* com blocos de 128 *bits*, passando para 14 no caso da maior chave ou do maior bloco. No entanto, diferente do DES, todas as operações envolvem *bytes* inteiros, para permitir implementações eficientes, tanto em *hardware* como em *software*.

Em Moreno (2005), o AES é classificado como um cifrador de bloco com tamanho de bloco e chave variáveis entre 128, 192 e 256 *bits*, o que significa que se pode ter tamanho de blocos com tamanhos de chaves diferentes. Em função do tamanho de bloco e chaves, determina-se a quantidade de rodadas necessárias para cifrar/decifrar.

O AES opera com um determinado número de 32 *bits*, que são ordenados em colunas de 4 *bytes* denominados **Nb**. Os valores possíveis são de 4, 6 e 8 equivalentes a blocos de 128, 192 e 256 *bits*.

Por isso sempre que **Nb** for referido, significa que se tem **Nb** x 32 *bits* de tamanho de bloco de dados. A chave é agrupada da mesma forma que o bloco de dados, isto é, em colunas, sendo representado pela sigla **Nk**.

Com base nos valores que **Nb** e **Nk** podem assumir é que se determina a quantidade de rodadas a serem executadas, identificada pela sigla **Nr**.

No processo de cifragem e decifragem do AES, as funções utilizadas não são as mesmas, como ocorre na maioria dos cifradores. Cada bloco ou estado é sujeito durante o processo para cifrar a seguintes iterações:

- ? **SubByte**: os bytes de cada bloco são substituídos por seus equivalentes em uma tabela de substituição (S-BOX);
- ? **ShiftRow** ou deslocamento de linha: nesta etapa, os *bytes* são rotacionados em grupos de quatro *bytes*;
- ? **MixColumn**: cada grupo de quatro *bytes* sujeita-se a uma multiplicação modular, o que proporciona a cada *byte* do grupo influenciar todos os outros *bytes*;
- ? **AddRoundKey** ou adição da chave de rodada: nesta fase, o bloco de dados é alterado por meio da subchave da rodada, a qual possui o mesmo tamanho do bloco, que realiza uma operação XOR com o bloco inteiro.

Através de uma análise de uma visão macro do AES que Nb, Nk e Nr possuem valores de acordo com o tamanho de bloco e chave a serem utilizados, observem que a última iteração é diferente das demais.

Na figura 3.1.3.1 mostra o algoritmo de cifragem AES.

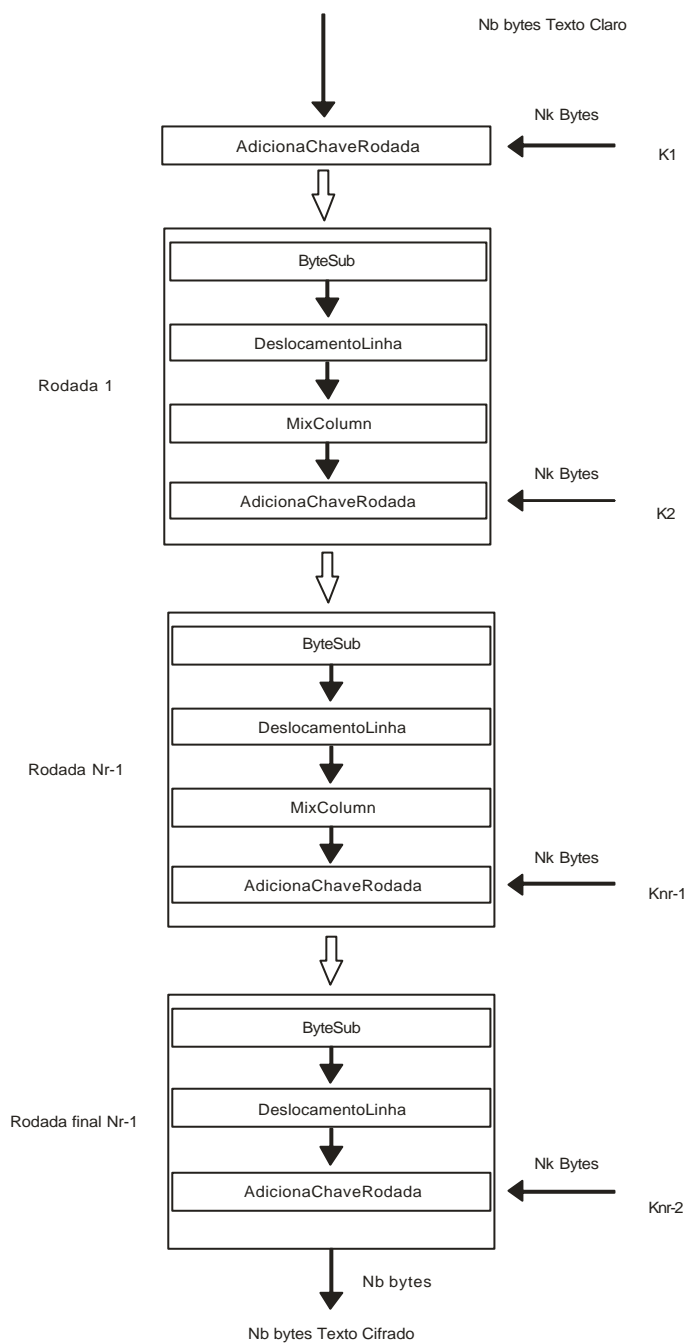


Figura 3.1.3.1 - Algoritmo de Cifragem do AES, Moreno (2005).

Ao observar a figura 3.1.3.2, podemos ver que o algoritmo poderia ser considerado como uma seqüência de transformações matemáticas e o seu processo reverso consistem, na aplicação da seqüência inversa à original.

O processo de expansão de chaves são os mesmo, porem as funções *SubByte*, *ShiftRow* *MixColumn* necessitam ser as suas inversas matemáticas para realizar o processo de decifragem.

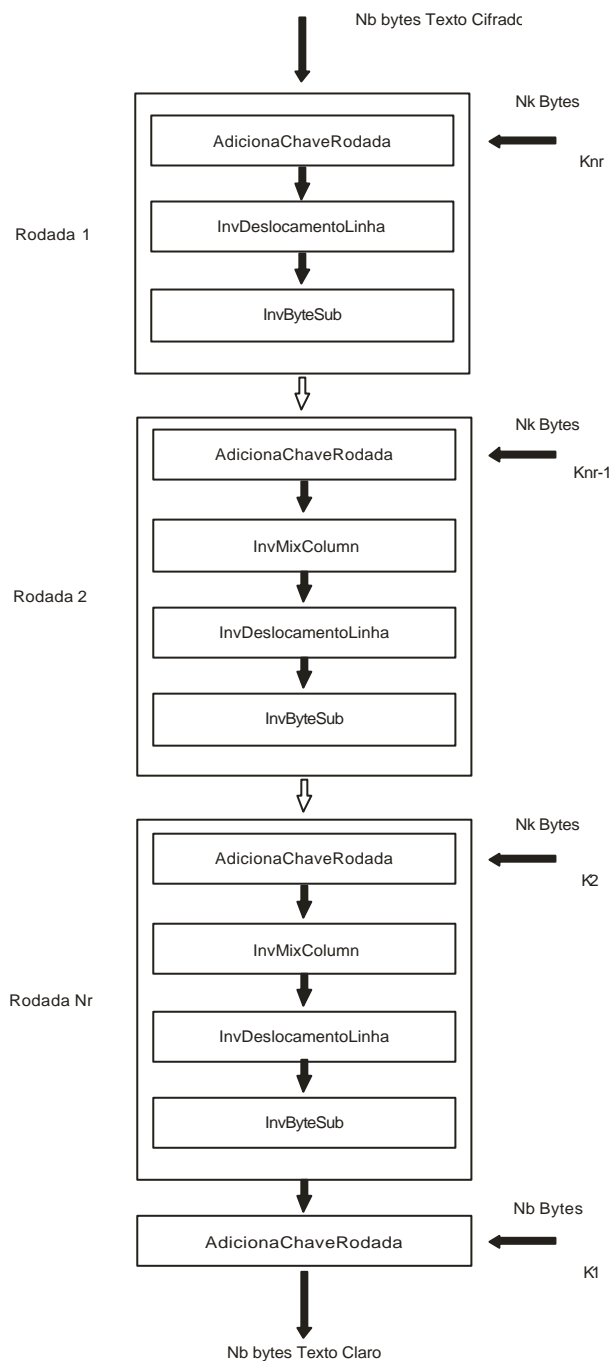


Figura 3.1.3.2 - Algoritmo de Decifragem do AES , Moreno (2005).



3.1.4. IDEA - *International Data Encryption Algorithm*

O algoritmo de IDEA, inicialmente chamado de IPES *Improved Proposed encryption standart*, foi idealizado por Lai e Massey em 1991, com intuito de ser eficiente em aplicações por *software* Lail (1990), citado em Moreno (2005).

O IDEA possui chave secreta de 128 *bits* e tanto na entrada de texto legível, como na saída de texto cifrado são 64 *bits*. É um algoritmo capaz de criptografar e de decriptografar, porem são utilizada duas fases diferente: 1 – é baseada na realização de oito iterações utilizando subchaves distintas, 2 – uma transformação final.

O IDEA está baseando em três operações de 16 *bits* conforme mostra a tabela 3.1.4.1.

Tabela 3.1.4.1 - Operações básicas do algoritmo de IDEA – Moreno (2005).

Operação	Descrição de Funcionamento
	Ou exclusivo (XOR) sobre 16 <i>bits</i>
+	Soma MOD 2^{16} , ou seja, somar dois valores de 16 <i>bits</i> desprezando o mais à esquerda, correspondente a 2^{16} .
	<p>Nesta operação são efetuados vários passos:</p> <ol style="list-style-type: none"> 1. Multiplica dois valores de 16 <i>bits</i> obtendo um valor que chamaremos de Z, e antes de multiplicar, se um desse valores for 0, deve ser alterado para 2^{16}. 2. Se o resultado da operação acima for 2^{16}, então o resultado final da operação será 0, caso contrario será o valor obtido em $2(Z \text{ MOD } (2^{16} + 1))$. 3. Calcular $Z \text{ MOD } (2^{16} + 1)$, ou seja , o resto da divisão de Z por $2^{16} + 1$.

O IDEA é um só algoritmo para criptografar e decriptografar, porém o que define a operação a ser realizada é a forma de geração das subchaves.

Conforme Moreno (2005), no processo de criptografia são geradas 52 subchaves de 16 *bits* a partir da chave secreta de 128 *bits*, também chamada em uma forma breve de **K**.

A primeira chave **K1** é gerada considerando 16 *bits* mais significativos de **K**, a segunda **K2** é gerada considerando os próximos 16 *bits*, e assim são geradas as subchaves até **K8**, que será os 16 bits menos significativos de **K**.

Como a cada 08 subchaves geradas a subchave seguinte inicia-se com um deslocamento de 25 *bits* a partir do início da chave **K**, a 9ª subchave é formada por 16 *bits* a partir do 25º *bit* a partir da direita de **K**, a **K10** é formada por pelos 16 próximos *bits*, e assim até gerar **K14**.

Seguindo o mesmo raciocínio, a 15ª subchave **K15** será formada pelos 7 *bits* menos significativos de **K** e para completá-la são usados os primeiros 9 bits (os 9 *bits* mais significativos de **K**).

Vale ressaltar que a cada 8 subchaves a chave seguinte inicia-se com um deslocamento de 25 *bits* à chave inicial **K**, a **K17** inicia-se a partir do 50º *bit* a partir da direita de **K**. Dessa forma são geradas as primeiras 48 subchaves.

A subchave **K49** inicia-se a partir do 22º *bit* à direita de **K**, a **K50** é formada pelos próximos 16 *bits*, e assim até formar a **K52**.

Como já descrito acima que o IDEA executa 08 iterações é uma transformação final. Para cada iteração possui duas partes e utiliza 06 subchaves que chamaremos de **Ka**, **Kb**, **Kc**, **Kd**, **Ke** e **Kf**.

Na primeira parte de cada iteração são utilizadas as quatro primeiras subchaves e na segunda parte as subchaves restantes; ambas utilizam as operações descritas no início.

As duas partes utilizam entrada e saída de 64 *bits* divididos em quatro blocos de 16 *bits* que daremos o nome de **Xa**, **Xb**, **Xc** e **Xd** para entrada e **Xa'**, **Xb'**, **Xc'** e **Xd'** para a saída, sendo que a saída da primeira parte é a entrada para a da segunda parte.

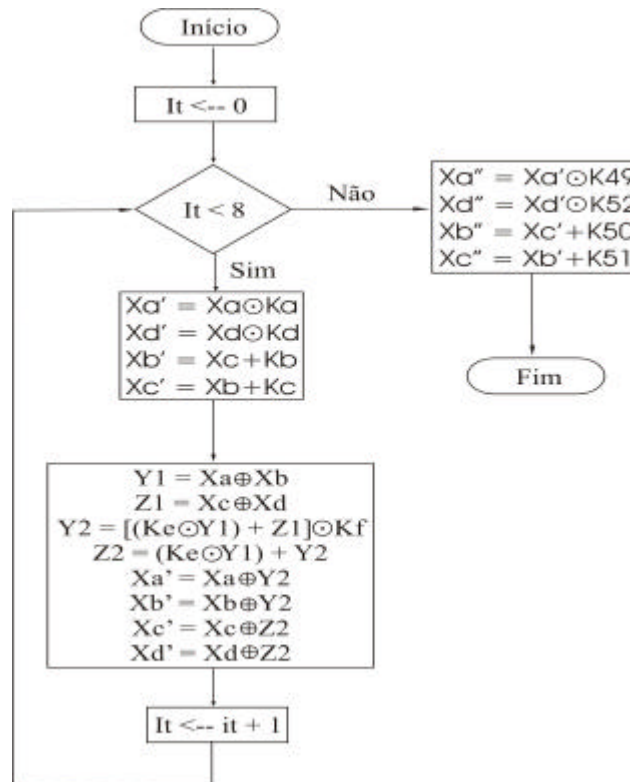


Figura 3.1.4.1 - Esquema do algoritmo IDEA – Moreno (2005).

No processo de decriptografar uma mensagem no IDEA basta alterar a forma de como as chaves são geradas, ou seja, é necessário calcular a inversa das chaves utilizadas na primeira parte de cada iteração e inverter a ordem em que são utilizadas. Primeiramente, utilizam-se as inversas multiplicativas MOD $2^{16} + 1$ das subchaves **K49** e **K52** para formar as subchaves **K1** e **K4** e as inversas aditivas das chaves **K50** e **K51** para gerar as subchaves **K2** e **K3**. Devem-se utilizar as subchaves **K47** e **K48** para formar **K5** e **K6** que serão utilizadas na segunda parte da primeira iteração. As inversas das subchaves **K43**, **K44**, **K45** e **K46** na primeira parte da segunda iteração e as subchaves **K41** e **K42** na segunda parte da segunda iteração, e assim por diante, até **K1**, **K2**, **K3** e **K4** na transformação final.

3.1.5. RC5 - *Rivest Cipher*

O algoritmo RC5 foi projetado pelo professor Ronald Rivest, do MIT, nos Estados Unidos, e publicado pela primeira vez em dezembro de 1994. Desde sua publicação o algoritmo de RC5 chamou atenção de muitos investigadores na comunidade científica em esforços para avaliar com precisão a segurança oferecida.

Segundo LANZARIN⁽⁸⁾, o algoritmo criptográfico RC5 foi projetado para qualquer computador de 16, 32 ou 64 *bits*. Possui uma descrição compacta e é adequado para implementações em software ou hardware. O número de iterações e o número de *bits* da chave são variáveis. Baseia-se na operação de deslocamento circular de um número variável de posições, e esse número depende de quase todos os bits resultantes da iteração anterior e do valor da subchave em cada iteração. Os objetivos que levaram ao desenvolvimento do algoritmo RC5 foram:

- ? Cifra simétrica de bloco;
- ? Adequado para *hardware* e *software*;
- ? Rápido;
- ? Flexível para processadores com diferentes comprimentos de palavras;
- ? Número variável de iterações;
- ? Tamanho de chaves variáveis;
- ? Simples;
- ? Baixa memória requerida;
- ? Alto nível de segurança.

Conforme Burnett (2002), define que o algoritmo Criptográfico Simétrico RC5 é o sucessor do RC4, pois apresenta significativas melhoras aos fatores segurança e velocidade. Ambos foram desenvolvidos pela empresa Americana *RSA Data Security Inc*, que foi criada pelos autores do sistema RSA, e é atualmente uma das mais importantes na área de sistemas de criptografia e proteção de dados.

De acordo com Moreno (2005), o algoritmo baseia-se na operação de rotação (deslocamento circular) de um número variável de posições, e esse número depende de quase todos os *bits* resultantes de iteração anterior e do valor da subchave em cada iteração.

O RC5 é um algoritmo parametrizado, isto é, possui flexibilidade quanto ao tamanho bloco, da chave e do número de iterações. Os parâmetros são os seguintes:

- ? **w** = Tamanho do bloco em *bits* que será cifrado. O valor padrão é 32 *bits*; podendo ser 16, 32 ou 64 *bits*. RC5 codifica blocos de duas palavras de forma que o texto claro e cifrado sejam de tamanho $2w$.
- ? **r** = Número de iterações (0 a 255).
- ? **b** = Número de *bytes* da chave secreta **k** (0 a 255).

O algoritmo de RC5 é dividido em três componentes básicos:

1. Um algoritmo de expansão fundamental.
2. Um algoritmo de cifragem.
3. Um algoritmo de decifragem.

Esses algoritmos utilizam cinco operações primitivas, todas sobre operandos de tamanho **w** *bits*. Essas operações são descritas por Rivest (1995) citadas em Moreno (2005):

1. **v + u** é a soma dos inteiros **v** e **u** de tamanho **w** *bits*, resultado um valor de **w** *bits* (soma **MOD** 2^w);
2. **v - u** é a subtração dos inteiros **v** e **u** de tamanho **w** *bits*, resultando um valor de **w** *bits* (subtração **MOD** 2^w);
3. **v XOR u** é o ou-exclusivo (**XOR**) **v** e **u** de tamanho **w** *bits*, resultando um valor de **w** *bits*;
4. **v << t** é o deslocamento circular (rotação) de **t** posições para a esquerda dos *bits* em **v**. Se **t** exceder, pode-se considerar os **w** *bits* menos significativos sem alterar o resultado;
5. **v >> t** é o deslocamento circular (rotação) de **t** posições para a direita dos *bits* menos significativos sem alterar o resultado.

Um texto claro no algoritmo RC5 consiste em duas palavras de tamanho w *bits*, denotadas por **A** e **B**. O primeiro *byte* ocupa as posições menos significativas do registro **A**, e assim por diante, de forma que o quarto *byte* ocupa o *byte* mais significativo de **A**, o quinto *byte* ocupa a posição menos significativa em **B** e o oitavo (ultimo) *byte*, a posição mais significativa em **B**.

Em Moreno (2005), fala que tanto na criptografia como na deciptografia, supõe que o computador armazena os *bytes* em modo little-endian, ou seja, tanto o texto legível como ilegíveis são armazenados da seguinte forma nas variáveis **A** e **B**, para $w = 32$: os primeiros 4 *bytes* do texto (i) legível (**x0, x1, x2, x3**) são armazenados em **A** na seqüência (**x3, x2, x1, x0**) e os 4 *bytes* seguido do texto (i)legível (**x4, x5, x6, x7**) são armazenados em **B** na seqüência (**x7, x6, x5, x4**).

3.2. Algoritmos assimétricos

O Algoritmo assimétrico ou algoritmo de chave pública nasceu com o intuito de resolver o problema que existia na hora da distribuição de chave dos algoritmos simétricos. Afinal não importa quanto um sistema criptográfico seja sólido, se um intruso conseguir roubar uma chave, o sistema acabava ficando inútil. Ou seja, devido à questão de ser apenas uma única chave necessária para cifrar e decifrar nos algoritmos simétricos, tornando as chaves objetos de extremo cuidados, sendo protegidas dentro de cofre contra roubos mais tinha que ser distribuídas ao mesmo tempo para que os destinatários tivessem como decifrar a mensagem recebida. Trazendo preocupações aos criptográficos da época, em relação a este aspecto de segurança.

Conforme Tanenbaum (1997), no ano de 1976, dois pesquisadores da *University of Stanford*, Diffie e Hellman, propuseram um algoritmo de criptografia radicalmente novo, no qual as chaves de criptografia e de descriptografia eram diferentes, e a chave a chave de descriptografia não podia ser derivada da chave de criptografia.

Tal algoritmo tinha com base três requisitos básicos:

- ? **D(E(P)) = P**: neste primeiro requisito, fala se aplicarmos **D** a uma mensagem criptografada, **E(P)**, obteremos outra vez a mensagem de texto simples

original **P**. Este requisito permite ao destinatário legítimo decodificar o texto cifrado.

- ? É extremamente difícil deduzir **D** a partir de **E**.
- ? É não pode ser decifrado por um ataque de texto simples escolhido.

No algoritmo assimétrico é utilizada uma chave pública para criptografar um texto legível e uma chave privada para a decodificação do texto cifrado. Iniciarem a falar do algoritmo RSA que é um dos mais importantes dentro desta linha de criptografia.

3.2.1 RSA

O RSA é um algoritmo de criptografia de chave assimétrica, ou de criptografia pública, desenvolvido por volta de 1977 pelos professores do MIT, Ronald Rivest e Adi Shamir, e o professor da USC *University of Southern California*, Leonard Adleman.

Este sistema de criptografia consiste em gerar uma chave pública, geralmente utilizada para cifrar dados e uma chave privada, utilizada para decifrar os dados, por meio de números primos grandes, o que dificulta a obtenção de uma chave a partir da outra.

A segurança proporcionada deste algoritmo de criptografia depende do tamanho dos números primos fornecidos, ou seja, quanto maior os números primos fornecidos maior segurança. Atualmente os números primos que são utilizados têm geralmente 512 *bits* de comprimento e combinados formam chaves de 1.024 *bits*. E já tem aplicações, como por exemplo, a bancárias que exigem o máximo de segurança, que a chave chega a ser de 2.048 *bits*.

Com o passar do tempo a tendência é aumentar o comprimento da chave. Este avanço ocorre devido à evolução nos sistemas computacionais que acompanham o surgimento de computadores que são capazes de fatorar chaves cada vez maiores em pouquíssimo tempo.

Os algoritmos para a geração de chaves públicas e privadas usadas para cifrar e decifrar são simples. De acordo com Rivest et al.(1978) citado em Moreno (2005).

1. Escolhem-se dois números primos grandes (**p** e **q**);
2. Gera-se um número **n** por meio da multiplicação dos números escolhidos anteriormente (**n** = **p** . **q**);
3. Escolhe-se um número **d**, tal que **d** é menor que **n** e **d** é relativamente primo à (**p** -1) . (**q** - 1).
4. Escolhe-se um número **e** tal que (**ed** -1) seja divisível por (**p** -1) . (**q** -1).
5. Os valores **e** e **d** são chamados de expoentes públicos e privado, respectivamente. O par (**n,e**) é a chave pública e o par (**n,d**), a chave privada. Os valores **p** e **q** devem ser mantidos em segredo ou destruídos.

Para cifrar uma mensagem através deste algoritmo, realizam-se os seguintes cálculos:

$$C = T^e \text{ MOD } n$$

Onde **C** é a mensagem cifrada, **T** é texto original **e** e **n** são dados a partir da chave pública (**n,e**). A única chave que pode decifrar a mensagem **C** é a chave privada (**n,d**) por meio do cálculo de:

$$T = C^d \text{ MOD } n$$

Na primeiramente etapa atribui-se valores a **p** e **q**, sendo que esses valores terão que ser números primos. Considerando o valor de **p** = 53 e **q** = 61. Nesta segunda etapa obtêm-se o valor de **n**, que é **n** = 53 . 61, ou seja, **n** = 3233.

Pela terceira etapa, escolhe-se um número **d** tal que **d** seja menor que **n** e relativamente primo a (**p** - 1) . (**q** - 1). Para tanto, basta escolher um número primo aleatório maior que **p** e **q**. para este exemplo escolhemos **d** = 193.

Pela quarta etapa, escolhe-se um número **e** tal que (**ed** -1) seja divisível por (**p** -1) . (**q** - 1). Para realizar tal cálculo, utiliza-se o algoritmo de Euclides estendido. Fazendo o uso desse algoritmo, calculamos que **e** = 97. Considerando os valores dados as variáveis **p**, **q**, **e**, **d** e **n**. Para realizar a criptografia, adotou-se a tabela 3.2.1.1, para representar numericamente letras maiúsculas do alfabeto. Em um algoritmo implementado em computador, a tabela utilizada é a ASCII.

Tabela 3.2.1.1 – valores dos caracteres para exemplo RSA – Moreno (2005).

	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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

Considerando a seguinte frase “ITS ALL GREEK TO ME” a ser criptografada e representada pela tabela pelos seguintes valores:

09 20 19 00 01 12 12 00 07 18 05 05 11 00 20 15 00 13 05 00

Para fazer a cifragem desta frase pegaremos blocos de duas letras, e para cada bloco, realizar o cálculo de $C = T^e \text{ MOD } n$, como demonstrado na figura 3.2.1.2.

Para $T = 0920$, temos: $C = 0920^{97} \text{ MOD } 3233 = 2546$.
 Para $T = 1900$, temos: $C = 1900^{97} \text{ MOD } 3233 = 1728$.
 Para $T = 0112$, temos: $C = 0112^{97} \text{ MOD } 3233 = 0514$.
 Para $T = 1200$, temos: $C = 1200^{97} \text{ MOD } 3233 = 2546$.
 Para $T = 0718$, temos: $C = 0718^{97} \text{ MOD } 3233 = 2304$.
 Para $T = 0505$, temos: $C = 0505^{97} \text{ MOD } 3233 = 0153$.
 Para $T = 1100$, temos: $C = 1100^{97} \text{ MOD } 3233 = 2922$.
 Para $T = 2015$, temos: $C = 2015^{97} \text{ MOD } 3233 = 2068$.
 Para $T = 0013$, temos: $C = 0013^{97} \text{ MOD } 3233 = 1477$.
 Para $T = 0500$, temos: $C = 0500^{97} \text{ MOD } 3233 = 2726$.

Figura 3.2.1.2 - Exemplo de cifragem com RSA

Assim temos a mensagem cifrada:

2546 1728 0514 0210 2304 0153 2922 2068 1477 2726

Para decifrar cada bloco o cálculo a ser utilizado é $T = C^d \text{ MOD } n$, como exemplificado na figura 3.2.1.3.

Para $C = 2546$, temos: $T = 2546^{193} \text{ MOD } 3233 = 0920$.
 Para $C = 1728$, temos: $T = 1728^{193} \text{ MOD } 3233 = 1900$.
 Para $C = 0514$, temos: $T = 0514^{193} \text{ MOD } 3233 = 0112$.
 Para $C = 0210$, temos: $T = 0210^{193} \text{ MOD } 3233 = 1200$.
 Para $C = 2304$, temos: $T = 2304^{193} \text{ MOD } 3233 = 0718$.
 Para $C = 0153$, temos: $T = 0153^{193} \text{ MOD } 3233 = 0505$.
 Para $C = 2922$, temos: $T = 2922^{193} \text{ MOD } 3233 = 1100$.
 Para $C = 2068$, temos: $T = 2068^{193} \text{ MOD } 3233 = 2015$.
 Para $C = 1477$, temos: $T = 1477^{193} \text{ MOD } 3233 = 0013$.
 Para $C = 2726$, temos: $T = 2726^{193} \text{ MOD } 3233 = 0500$.

Figura 3.2.1.3 - Exemplo de decifragem no RSA

Através destes cálculos obtemos o texto original que é:

0920 1900 0112 1200 0718 0505 1100 2015 0013 0500

Devido quantidade de cálculo que o algoritmo de RSA faz na hora de cifrar/decifrar, um software que o implementa fica cerca de 100 vezes mais lento, como relação a um software que implementa o algoritmo de DES. Mais o fato é que o RSA garante uma privacidade maior a seus usuários e com isso ele não deixará de ser utilizado.

CAPÍTULO IV—FERRAMENTAS DE CRIPTOGRAFIA

Hoje em dia existem muitas ferramentas criptográficas disponíveis no mercado, ferramentas capazes de criptografar arquivos independentes do tamanho do arquivo ou simplesmente uma mensagem a ser enviada a um destinatário através da internet. Esta grande quantidade oferece uma vantagem ao usuário, pois assim ele poderá escolher uma que seu equipamento possa trabalhar sem que lhe cause desconforto na hora de cifrar ou decifrar o texto ou mensagem requisitada.

As ferramentas que trabalham com algoritmos assimétricos exigem mais do computador na hora de fazer a sua operação independente de qual seja ela, mais também não é algo alarmante que faça com que elas não sejam utilizadas. Já as ferramentas que utilizam o algoritmo simétrico não exigem muito do equipamento, porém apresentam a desvantagem na hora da distribuição das chaves necessárias para cifragem ou decifragem da mensagem.

4.1. WebCry 2.0

Esta ferramenta criptográfica foi desenvolvida na linguagem de programação Java. Conforme Moreno (2005), ela é composta de quatro partes básicas. Na primeira parte o usuário seleciona o algoritmo que ele deseja cifrar/decifrar.

Na segunda parte define o tamanho da chave a ser utilizada; na terceira parte é possível realizar a criptografia de arquivos e na quarta e última parte permite a criptografia de texto pequeno.

Um ponto forte do WebCry 2.0 é que não é necessário instalar no computador devido ao fato dele ser um arquivo .jar, para utilizá-la basta ter a máquina virtual Java instalado. A figura a seguir, mostra a tela principal do WebCry 2.0.

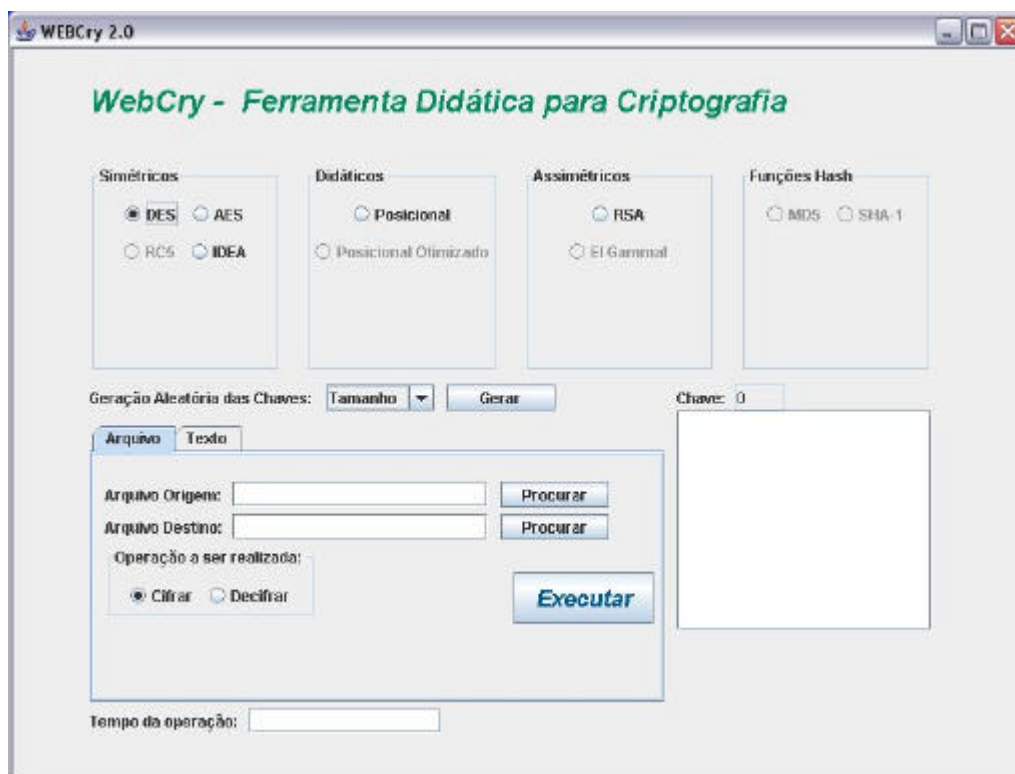


Figura 4.1.1 – Tela principal do WebCry 2.0

Apesar desta ferramenta ainda está em fase de desenvolvimento, ela é capaz de cifrar e decifrar arquivo de diversos tamanho possui a flexibilidade de trabalhar com diversas extensões de arquivo. De uma maneira simples e bastante eficiente, ela gera a chave a partir do tamanho determinado pelo usuário e informa o tempo gasto na operação de cifragem ou decifragem. Uma desvantagem apresentada nesta ferramenta é que o usuário não pode definir uma chave, este fator gera desconforto ao usuário, pois raramente ele conseguirá repetir chave que foi gerada para inverter o processo que a utilizou.

4.1.1. Funcionamento da WebCry 2.0

O processo de funcionamento desta ferramenta é bastante simples, são necessários apenas seis passos para cifrar ou decifrar arquivo desejado, os passos citados acima são:

1. Inicializar a ferramenta.
2. Escolher o algoritmo para a operação de cifragem ou decifragem.

3. Escolher o tamanho da chave e clicar no botão “gerar” da tela principal para que a ferramenta gere a chave, e necessário que siga esta ordem.
4. Escolher o arquivo a ser cifrado ou decifrado, cuidado para não sobrescrever o arquivo original na hora de informar o destino do arquivo cifrado/decifrado.
5. Selecionar a operação desejada cifrar/decifrar, por *default* a operação e a de criptografar/cifrar.
6. Clicar no botão “executar” para que a ferramenta realize a operação.

Esta ferramenta permite também que o usuário poderá digitar um texto, para a ferramenta criptográfica possa cifrar ou decifrar com pode ser observado na figura 4.1.2.

WebCry 2.0

WebCry - Ferramenta Didática para Criptografia

Simétricos
☒ DES ☐ AES
☐ RC5 ☐ IDEA

Didáticos
☐ Posicional
☐ Posicional Otimizado

Assimétricos
☒ RSA
☐ El Gammal

Funções Hash
☐ MD5 ☐ SHA-1

Geração Aleatória das Chaves: Tamanho ▼ Gerar

Chave: 0

Arquivo **Texto**

Origem: texto digitado pelo usuario

Hex: 746578746F20646967697461646F2070656C6F207573756172696F

Hex:

Cifrado:

Cifrar

Decifrar

Tempo da operação:

Figura 4.1.2. Tela de entrada de dado a ser criptografado.

4.2. PGP – Pretty Good Privacy

Baseando em TOMITA ⁽¹⁴⁾, o PGP é uma ferramenta criptográfica que originalmente foi escrita por Phil Zimmermann em 1991. Esta ferramenta possibilita troca de mensagens eletrônica, ou seja, de *e-mails* de modo seguro através de uma rede insegura, como a *internet*. Existe hoje no mercado uma versão desta ferramenta criptográfica chamada GPG que é livre e muito utilizada por usuários do sistema operacional *Linux*.

De acordo com UNICAMP ⁽¹⁷⁾, o SIE (Sistema Informatizado de Eleições), utiliza o PGP como meio seguro de transporte, o que é adequado, desde que a geração e manutenção das chaves sejam feitas corretamente dentro da política de segurança estabelecida. Além disso, há registros dos eventos e uso de proteção criptográfica (sigilo e autenticação) de manutenção dos serviços do SIE.

4.2.1. Funcionamento do PGP

Esta ferramenta criptográfica utiliza vários mecanismos criptográficos para seu funcionamento, entre eles criptografia assimétrica, criptografia simétrica, funções de *Hash* e compressão de dados.

O PGP está hoje na versão 9.0, disponível no site do desenvolvedor a versão *trial*, esta ferramenta é capaz de criptografar um arquivo e salva-lo em uma extensão **.pgp** que somente ela tem a capacidade de abri-la mediante a chave *key* do usuário que pode autorizar tal ação. Uma outra função seria criptografar todo o HD da máquina que ela está instalada, pode gerar um disco virtual.

O processo de instalação desta ferramenta é bastante simples basta seguir os passos abaixo:

1. Descompactar o arquivo baixado.
2. Executar o arquivo de nome PGPDesktop906_Inner.exe.
3. Logo após execute também PGPDesktop906_Inner.exe.sig.
4. Com isto o PGP estará instalado e pronto para ser utilizado.

Para iniciar a utilização do PGP, após a instalação abra o Programa e crie uma PGP key, como mostra a figura abaixo. A implementação foi feita na versão Desktop do PGP 9.0 Trial.

Os passos são os seguintes:

1. Vá em menu File\new\PGP key.
2. Preencha o assistente de criação de chave com suas informações.

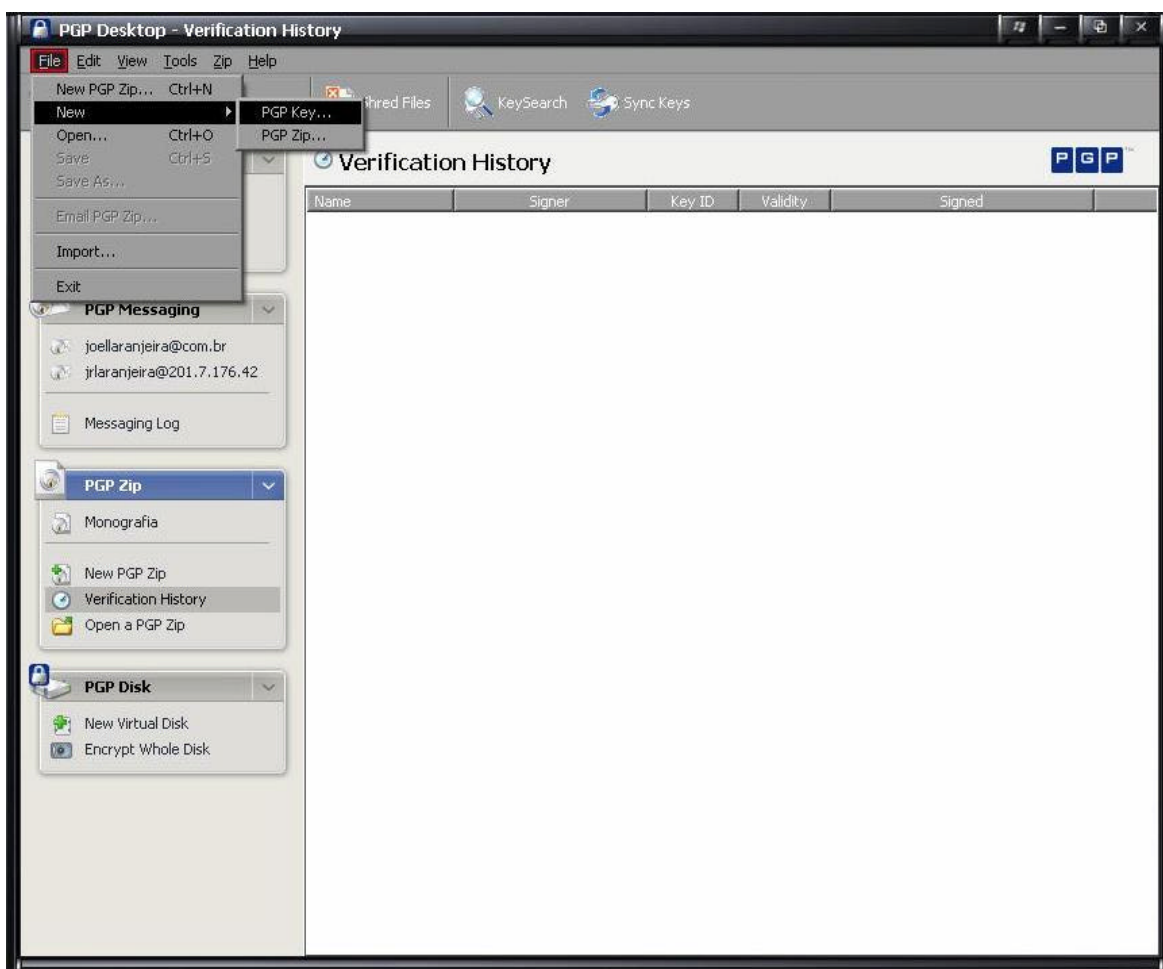


Figura 4.2.1.1 Criando key no PGP

Uma vez criada a key o PGP estará pronto para iniciar a utilização, os processos a seguir mostrará como fazer o PGP criptografar um arquivo de Word, salvar o arquivo com uma extensão **.pgp**.

1. Na tela principal do PGP e clique na opção *New PGP Zip*.
2. Na tela que Abrirá clique em *Add Recipients*.

3. Na mesma tela clique em *Add Files*, selecione o arquivo que você irá encriptar e compactar.
4. Último passo deste processo é clicar no botão *Save*, que o arquivo será compactado e salvo no lugar desejado.

O processo de descompactar o arquivo compactado pelo PGP é muito simples, basta clicar no botão *Verify PGP Zip*, que o PGP irá descompactar o arquivo no local onde está o arquivo compactado e criptografado.

Na figura abaixo mostra as opções requisitadas nos passos citados acima.

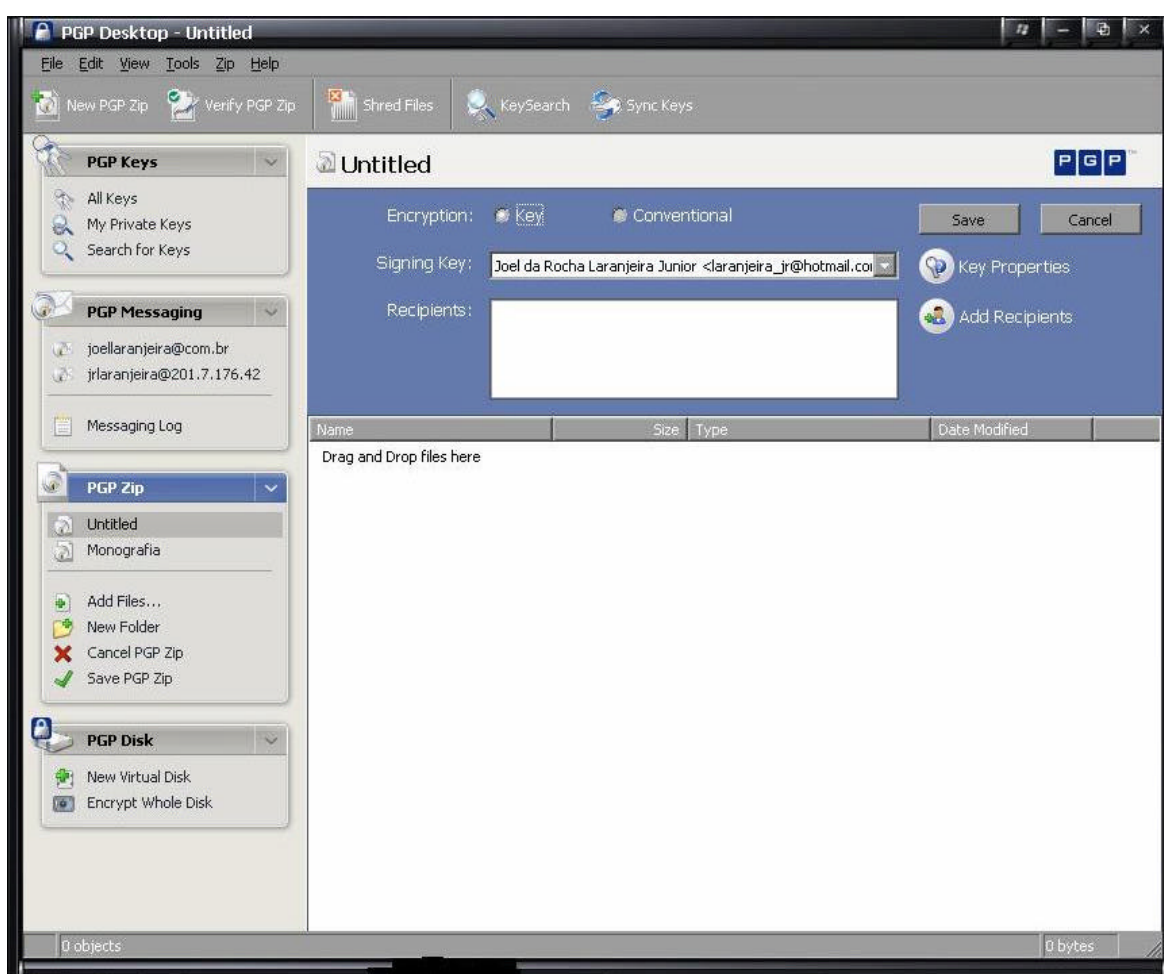


Figura 4.2.1.2 Tela de ilustração PGP

O PGP exige que todos os usuários possuam uma chave pública e uma privada. A chave pública pode ser enviada para outro usuário do PGP.

Ela é utilizada para que um indivíduo possa encriptar documentos. Os documentos encriptados com sua chave pública só podem ser decifrados com a sua chave privada (a qual você deve manter de forma secreta).

As chaves públicas podem ser obtidas nos sites das empresas ou organizações. Também existem os chamados *Key Servers* que disponibilizam essas chaves.

O PGP também pode ser usado para **assinar** mensagens de e-mail, ou seja, certificar que as mesmas não foram modificadas desde que foram enviadas, utilizando um *hash* baseado na chave privada.

Por debaixo dos panos o PGP também gera uma **chave de sessão**, de modo que toda mensagem criptografada com PGP está criptografada duas vezes. Isso torna o PGP o melhor sistema de criptografia existente na atualidade.

A *PGP Corporation* oferece uma linha de produtos diferentes para diferentes classes de clientes. Segue uma lista de seus produtos por categoria de cliente:

? **Enterprise**

- PGP Universal
- PGP Corporate Desktop
- PGP Corporate Disk
- PGP Command Line
- PGP Mobile

? **Workgroup**

- PGP Universal
- PGP Workgroup Desktop
- PGP Mobile

? **Personal**

- PGP Personal Desktop
- PGP Mobile

? **PGP Universal**

? **PGP Desktop**

- PGP Corporate Desktop
- PGP Workgroup Desktop
- PGP Personal Desktop

- PGP Corporate Disk

? **PGP Command Line**

? **PGP Mobile**

O PGP *Universal*, que consiste em um servidor aplicando o PGP nas mensagens que passam por ele, garantindo políticas de segurança da empresa de envio e recebimento de mensagens.

O PGP *Desktop*, que permite ao usuário controlar a cifragem de suas mensagens bem como do conteúdo armazenado em seu computador pessoal.

O PGP *Personal Desktop*, a solução para uso pessoal da linha, esta disponível para as plataformas *MacOS* e *Windows*. Ele inclui as ferramentas PGP *Disk*, de criptografia transparente do conteúdo dos discos rígidos e PGP *Mail*, de criptografia de mensagens eletrônicas.

O PGP *Desktop* utiliza os formatos de chave pública OpenPGP RFC 2440 e X.509, os algoritmos simétricos AES, CAST, TripleDES, IDEA e Twofish, os algoritmos assimétricos Diffie-Hellman, DSS e RSA, e as funções de hash SHA-1, MD5 e RIPE-MD-160.

CONCLUSÃO

A segurança de dados é uma situação preocupante para qualquer sistema, devido ao fato que com uma única ferramenta não se conseguem garantir ou prover um nível desejado de segurança.

Na política de segurança de uma empresa ou em sistema de informação, a criptografia de dados deve ser incluída para se ter um nível melhor de segurança, com isso quer dizer que a criptografia de dados não deve ser também a única ferramenta para garantir a segurança e sim fazer parte de um conjunto de ferramentas utilizadas pela política de segurança adotada.

A perspectiva é que a criptografia de dados seja embutida nos sistemas operacionais, como já acontece com as senhas. Mediante ao avanço dos computadores e dos sistemas operacionais que cada vez se mostram preocupados com a segurança de dados dos seus usuários.

Esta pesquisa serve para ampliar a visão a respeito da segurança de dados no Sistema Eleitoral Brasileiro, considerando que é preciso entender que o uso de criptografia por si só não torna um sistema seguro.

Seja qual for o sistema, tem que estar a par desta situação e não se deve preocupar somente com invasões, vírus ou conteúdo que os usuários da sua empresa estão acessando mais sim ter a confiança que as informações que estão sendo enviadas, estarão com a mínima segurança necessária. O Sistema de informação Eleitoral desenvolvido no Brasil detém todas as ferramentas necessárias para suprir as necessidades de segurança exigido por Lei (sigilo do voto) e a tranquilidade do eleitor desde a implantação do sistema na urna eletrônica até o momento da apuração e totalização dos votos.

REFERÊNCIAS

- [1] ARNETT, M. F.; DULANEY, E; HARPER, E.; HILL, D. L; KROCHMAL, J; KUO, P.; LEVALLEY, J; MCGARVEY, J.; MELLOR, A.; MILLER, M.; ORR, S; RAY, L.; RIMBEY, S.; WANG, C. – **Desvendando o TCP/IP- Métodos de instalação, manutenção e implementação de redes TCP/IP**; Campus; 2º reimpressão; 1997.
- [2] BRASIL. **Agência Brasileira de Inteligência**. Disponível em: <<http://www.abin.gov.br/abin/index.jsp>>. Acesso em: 11 de set. 2006.
- [3] BRASIL. Lei nº. 10.408, de 10 de janeiro de 2002. **Estabelece normas para as eleições, para ampliar a segurança e a fiscalização do voto eletrônico**. SENADO FEDERAL. **Legislação Republicana Brasileira**. Brasília, 2002. Disponível em: <<http://www6.senado.gov.br/legislacao/ListaPublicacoes.action?id=234238>>. Acesso em 11 de set. 2006.
- [4] BRUZANO FILHO, **Amílcar**. **Critérios para Avaliação da Segurança do Voto Eletrônico**. Disponível em: <<http://www.votoseguro.org/textos/SSI2000.htm>>. Acesso em: 09 de set. 2006.
- [5] BURGESS, Mark,; - **Princípios de Administração de Redes e Sistemas**. Rio de Janeiro: LTC; 2º edição; 2006.
- [6] BURNETT, S.; PAINE, S.; - **Criptografia e Segurança: o Guia Oficial RSA**; Campus; 1º edição; 2002.
- [7] DIAS, Cláudia. **Segurança e Auditoria da Tecnologia da Informação**. Rio de Janeiro: Axcel Books do Brasil, 2000. 218p.
- [8] LANZARIN, F.; **Algoritmo RC5**:. Disponível em: <<http://www.inf.ufsc.br/~lanzarin/seguranca/trabalho6.html>> Acesso em: 10 de set. 2006.
- [9] MELO, Emerson Ribeiro.; **Redes de Confiança**; 2003; Disponível por em: <<http://www.das.ufsc.br/seguranca/artigos/mello2003-dissertacao.pdf>> Acessado em: 12 de set. de 2006)

- [10] MORENO, E; PEREIRA, F. D.; CHIARAMONTE, R. B. – **Criptografia em Software e Hardware**; Novatec; 2005.
- [11] SOARES, L. F. G.; LEMOS, G.; COLCHER, S. – **Redes de Computadores Das Lans, Mans e Wans às Redes ATM**; Campus; 6º edição; 1995.
- [12] TANENBAUM, A. S. – **Redes de Computadores**; Campus; 3º edição; 1997.
- [13] TKOTZ, V. – **Criptologia NumaBoa.**: Disponível em:
<<http://www.numaboia.com/content/section/11/57/>> . Acesso em: 09 de set. 2006.
- [14] TOMITA, R. T.; **Monografia PGP**. Link:
<http://www.students.ic.unicamp.br/~ra992432/mo639/monografia_pgp.pdf>. Acesso em: 01 de out. 2006.
- [15] TRIBUNAL REGIONAL ELEITORAL DO ESPÍRITO SANTO. **Segurança do Voto**. Disponível em:
< http://www2.tre-es.gov.br/urna_eletronica/seguranca_do_voto.jsp>. Acesso em: 09 de nov. 2006.
- [16] TRIBUNAL SUPERIOR ELEITORAL. **RESOLUÇÃO N.º 21.000, 2002 - Instrução n.º 64 - Classe 12ª**. Disponível em: <<http://www.tse.gov.br>>. Acesso em: 10 de nov. 2006.
- [17] TRIBUNAL SUPERIOR ELEITORAL. **Relatório da UNICAMP / Urnas Eletrônicas**. Disponível em: <<http://www.tse.gov.br>>. Acesso em: 01 de set. 2006.
- [18] UCHÔA, J. Q. – **Segurança em Redes e Criptografia**; UFLA; Lavras – MG; 2003.