UNIVERSIDADE SAGRADO CORAÇÃO

MAURICIO FREDERICO

CRIPTOGRAFIA:
ESTUDO COMPARATIVO ENTRE SOFTWARES SOBRE A CRIPTOGRAFIA AES

BAURU
2011

MAURICIO FREDERICO

CRIPTOGRAFIA:
ESTUDO COMPARATIVO ENTRE SOFTWARES SOBRE A CRIPTOGRAFIA AES

Trabalho de conclusão de curso apresentado à Universidade Sagrado Coração, para a obtenção do título de Bacharel em Ciências da Computação, sob orientação do prof. Esp. Henrique Pachioni Martins

BAURU

2011

MAURICIO FREDERICO

CRIPTOGRAFIA:
ESTUDO COMPARATIVO ENTRE SOFTWARES SOBRE A CRIPTOGRAFIA AES

Trabalho de conclusão de curso apresentada ao curso de Ciência da Computação da Universidade Sagrado Coração, como parte dos requisitos para aprovação parcial e obtenção do grau de Bacharel em Ciência da Computação, sob a orientação do Prof. Especialista Henrique Pachioni Martins.

BANCA EXAMINADORA:

_____

Prof. Especialista Henrique Pachioni Martins
Orientador

_____

Prof. Dr. Kelton Augusto Pontara da Costa
Banca

_____

Prof. Especialista André Luiz Ferraz Castro
Banca

RESUMO

A criptografia é uma técnica milenar e a cada dia vem sendo mais utilizada por causa dos grandes avanços tecnológicos. No mundo atual há uma grande necessidade do uso de computadores para transferências de dados sigilosos, necessitando assim de ferramentas que garantem a sua confidencialidade, integridade e autenticação, sendo a criptografia importantíssima para que esses requisitos de segurança sejam preenchidos. Este trabalho faz uma explanação sobre as principais técnicas de criptografia simétrica e assimétrica, além de abordar a criptanálise,ou seja, técnicas usadas para obter o texto original a partir do texto cifrado, sem o conhecimento das chaves secretas. Por fim é realizada uma comparação entre softwares que utilizam a criptografia Advanced Encryption Standard mostrando as vantagens e desvantagens de seu uso.

Palavras Chave: Criptografia simétrica. Criptografia assimétrica. Segurança, Criptanalise.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE GRÁFICOS

Lista de Siglas

AES	Advanced Encryption Standard
DES	Data Encryption Standard
DH	Diffie Helman
FIPS	Federal Information Processing Standart
MIT	Instituto de Tecnologia de Massachusetts
NIST	National Institute of Standards and Tecnology
NSA	National Security Agency
RSA	Rivest Shamir Adleman

SUMARIO

1 INTRODUÇÃO

A criptografia vem sendo usada há vários séculos em contextos militares e diplomáticos para prover o sigilo das informações, pois ela torna ou tenta tornar os dados irreconhecíveis para qualquer pessoa que não seja o remetente ou receptor da mensagem.

Com os avanços tecnológicos e o uso do computador para transmissão e arquivamento de dados sigilosos foi preciso utilizar ferramentas automatizadas como: banco de dados, antivírus e firewall para prover a segurança destes arquivos e outras informações, garantindo assim a confidencialidade, integridade e autenticação. A criptografia vem sendo de grande importância para que esses requisitos sejam atingidos.

Segundo Stallings (2008) os dois componentes básicos para todas as técnicas de criptografia são substituição e transposição. A primeira ocorre quando letras do texto são substituídas por números ou símbolos, já a segunda ocorre quando há a permutação dos bits.

Apesar de sua forma de cifrar os textos continuam sendo iguais desde a História Antiga, quando Júlio Cesar fez pequenas substituições de letras em suas mensagens, a criptografia aumentou a sua capacidade de criptografar textos com surgimento dos computadores, tornando o algoritmo mais seguro. (Hinz, 2000)

Este trabalho tem o propósito de estudar, através de pesquisas bibliográficas, os vários tipos de criptografia existentes, mostrando as suas características. Posteriormente será realizada uma comparação de software utilizando a criptografia AES (Advanced Encryption Standard) que é a criptografia padrão dos EUA, aumentando assim o conhecimento de técnicas criptográficas utilizadas no mercado.

2 OBJETIVOS

2.1 OBJETIVO GERAL

Fazer uma analise comparativa sobre a técnica de criptografia simétrica AES comparando o funcionamento através de softwares.

2.2 OBJETIVOS ESPECÍFICOS

- Desenvolver uma pesquisa bibliográfica sobre os tipos de criptografia simétrica e assimétrica;
- Comparar por meio de software o funcionamento da criptografia AES;
- Mostrar as vantagens e desvantagens encontradas.

3 JUSTIFICATIVA

A criptografia é de suma importância quando se fala em segurança da informação. Através deste estudo serão mostradas as vantagens e desvantagens encontradas nos softwares que utilizam a criptografia AES. Este trabalho será feito para que a comunidade de alunos do curso de Ciências da Computação tenha um melhor conhecimento sobre o assunto e para a Universidade que ainda não tem muita bibliografia referente a este assunto.

4 CRIPTOGRAFIA

4.1 HISTÓRIA

O termo criptografia surgiu com a fusão de duas palavras de origem grega kryptós e gráphein que significam “escondido” e “escrita”, respectivamente. Essa técnica tem como objetivo de codificar a mensagem, sendo que apenas o emissor e o receptor, consigam acessá-las e obter corretamente a informação, evitando assim que pessoas não autorizadas obtenham conhecimento das informações sigilosas. (Mendes: 2007)

Segundo Tkotz (2005) a primeira criptografia que se tem registro é por volta dos anos 1900 a.C. numa vila egípcia próxima ao rio Nilo, chamada *Menet Khufu*.

A criptografia por várias vezes conseguiu mudar o rumo da história em virtude da preservação da informação por meio de métodos criptográficos ou a utilização da criptoanálise para a obtenção das informações. (Tkotz: 2005).

Tkotz (2005) afirma que a criptografia evoluiu e sofreu grandes transformações nos quatro mil anos de história, tendo como seus principais colaboradores, os militares, os quais utilizavam estes métodos para obter segurança dos dados transmitidos.

O imperador da Roma Antiga, Júlio Cesar, desenvolveu um algoritmo chamado Cifra de Cesar, que enganou muitas vezes os exércitos inimigos nas trocas de informações. Era um algoritmo baseado na técnica de substituição, na qual a letra do alfabeto era substituída por outra letra que estava a três posições à frente. (Hinz: 2000).

Na década de 70, como fala Benits (2003), os algoritmos criptógrafos eram secretos e eram utilizados pelas forças armadas e diplomacia de todos os países. Porém nesta mesma década foi provado que a segurança dos sistemas deveria estar na chave e não no algoritmo, criando assim o *Data Encryption Standard* (DES), o primeiro algoritmo de criptografia simétrica de domínio público.

Tkotz (2005) relata que os métodos da criptologia clássica são atemporais e que são base para toda a criptologia moderna e nos ajudam a entender os caminhos mais curtos e seguros para a utilização nos algoritmos atuais.

Atualmente, com os avanços tecnológicos, a utilização do computador passou a ser indispensável, necessitando assim um vasto sistema de proteção, nas quais as informações estejam seguras e sejam confiáveis, sendo este o papel da criptografia.

4.2 CONCEITOS

Abaixo serão relacionados os principais conceitos de criptografia que serão de suma importância para o entendimento no decorrer deste trabalho.

O transmissor é aquele que transforma a mensagem original em uma mensagem criptografada. O receptor recebe a mensagem criptografada e transforma novamente na mensagem original.

É chamado de mensagem ou texto claro, o texto original, aquele que ainda não foi modificado. O processo de torná-la ilegível é chamado encriptação ou cifragem, já o processo inverso, ou seja, tornar a mensagem legível é chamado decriptação ou decifragem.

A chave secreta é uma informação na qual é utilizada no método de cifragem e decifragem. Segundo Burnett e Paine (2002) “a chave secreta funciona da mesma maneira que uma chave convencional”, ou seja, é utilizada em uma fechadura (Algoritmo de Criptografia) e protegerá as informações, que somente com a chave certa o usuário poderá proteger e ter acesso as mesmas (Floriano: 2007).

A técnica para manter a mensagem segura é chamada criptografia. A técnica utilizada para desvendar, quebrar, descobrir o conteúdo tem o nome de criptoanálise. Essas duas técnicas usadas juntas é denominado criptologia.

4.2.1 Cifras de substituição

Segundo Fernandes (2007) a **técnica de substituição** consiste simplesmente na troca de bits, caracteres ou blocos de caracteres, ou seja, a troca acontece de acordo com uma tabela de substituição a qual mostrará qual regra seguirá a troca.

> A substituição é a família de cifras que possui o maior número de métodos criptográficos. As cifras de substituição se caracterizam pela troca de cada um

> dos caracteres originais por outros predefinidos, o que o diferencia dos códigos – estes permitem substituir caracteres, palavras ou até expressões. Tkotz (2005, p.180)

Esta técnica é a mais simples e também a mais fácil de ser quebrada, pois é necessário apenas saber a freqüência nas quais os caracteres aparecem e comparar com quais são as letras que mais aparecem em determinado idioma.

Um método de cifra de substituição conhecida é **substituição monoalfabética monográfica** na qual o caractere do texto original é substituído por outro, seguindo uma tabela de substituição, deixando assim a mesma freqüência de caracteres do texto original do cifrado, mantendo assim o tamanho dos textos iguais.

Outro método é **substituição monoalfabética poligâmica** que tem a mesma característica que o anterior, diferenciando apenas que cada caractere do texto original é substituído por mais de uma letra, deixando o comprimento do texto cifrado maior do que a mensagem original.

A substituição polialfabética ocorre utilizando mais de um alfabeto para efetuar a codificação da mensagem.

4.2.2 Cifra de transposição

As cifras de transposição ou de permutação como afirma Fernandes (2007) e Tkotz (2005) são técnicas com a função de reordenar a ordem dos bits, caracteres ou bloco de caracteres, ou seja, os caracteres são preservados, ocorrendo apenas trocas de posições das letras do texto.

Esta técnica segundo Fernandes (2007) e Schneier (2003) o texto simples e o cifrado tem o mesmo número de caracteres. Não ocorre a substituição das letras. Ela se baseia em uma chave que é uma palavra ou frase que não possuem letras repetidas.

4.3 CRIPTOGRAFIA SIMÉTRICA

A **criptografia simétrica** ou de **chave privada** é baseada na utilização de apenas uma chave tanto para emissor quanto para o receptor da mensagem. Benitz (2003) afirma que o grande problema deste tipo de criptografia é que

> [...] o número de chaves cresce proporcionalmente ao quadrado do número de participantes, visto que entre cada par de usuários existe uma, e somente uma chave e essa não pode ser utilizada para comunicação com um terceiro usuário

Precisa-se então de um grande numero de chaves caso a troca de informações ocorra com várias pessoas. Necessita assim de um gerenciamento nesta chave, pois se ela for interceptada por outra pessoa não autorizada, a mensagem criptografada poderá ser decifrada ou alterada facilmente. (Burnett; Paine: 2002)

Stalling (2008) ainda relata outros problemas na técnica de criptografia simétrica na qual a chave deve ser trocada entre as partes e armazenada de forma segura, o que nem sempre é fácil de ser garantido e também não garante a identidade de quem enviou ou recebeu a mensagem (autenticidade e não repúdio).

Stalling (2008) afirma como Benitz (2003) que a criptografia simétrica sofre problema das chaves, caso três pessoas quisessem se comunicar utilizando chaves secretas seria necessário o uso de três chaves.

A grande vantagem da criptografia simétrica é que seu algoritmo são bem mais rápidos que algoritmos de chave pública e sua implementação pode ser alcançada mais facilmente. (GARFINKEL; SPAFFORD: 1999)

Existem vários tipos de algoritmos de chave privada, podemos citar entre eles DES, Triple DES e AES, os quais serão abordados detalhadamente abaixo.

4.3.1 DES

O algoritmo DES (*Data Encryption Standard*) foi criado na década de 70 pela a IBM e foi adotado em 1977 pelo *National Institute of Standards and Tecnology* (NIST), o algoritmo era um rascunho do projeto Lúcifer, que é uma cifra de bloco de feistel que opera em blocos de 64 bits, utilizando uma chave de 128 bits. (Stalling, 2008).

Terada (2000) afirma que o desenvolvimento do DES tratou-se um grande avanço científico no histórico da criptografia por se tratar do primeiro algoritmo a se tornar público, pois antigamente os algoritmos do gênero eram secretos. O DES foi rapidamente aceito e foi padronizado para a utilização do governo dos EUA.

O DES é um algoritmo simétrico, ou seja, o processo de encriptação e decriptação é o mesmo. O algoritmo utiliza cifragem de blocos, a mensagem é dividida em bloco de 64 bits, usando chaves com o mesmo 64 bits, porém são apenas utilizados 56 bits reais. Os últimos 8 são utilizados para verificação da chave, empregando a informação a operações de XOR para obter a mensagem cifrada. (Floriano: 2007).

A transformação do texto simples em texto cifrado envolve 18 estágios no DES. O primeiro estágio é feita a transposição da chave no texto de 64 bits e no último estágio é feita a transposição inversa. Segundo Stallings (2008) excluindo a primeira e a última fase da transformação de texto, deixando-a com a estrutura exata da cifra de Feistel. A Figura 1, mostra o esquema geral da criptografia.

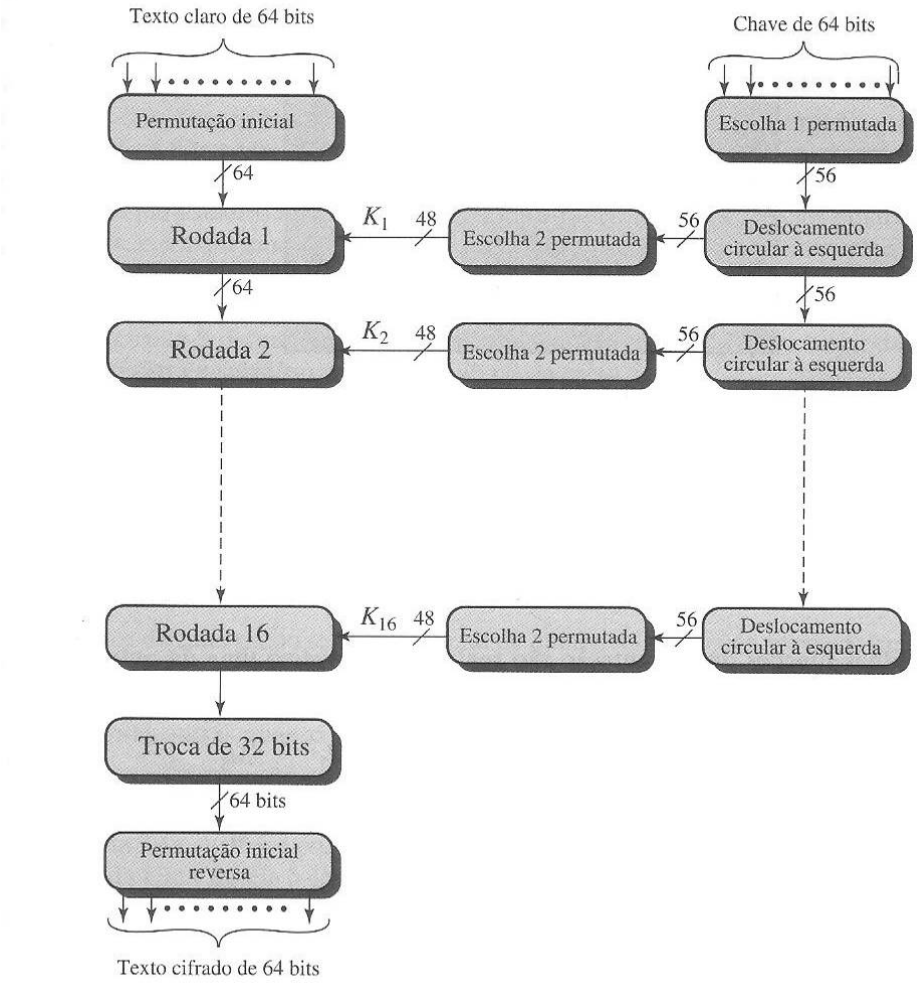

Figura 1 – Representação geral do algoritmo de criptografia DES

(Fonte: STALLING (2008))

A figura 1 como explica Stalling (2008) mostra a existência de duas entradas na função de criptografia: o texto claro a ser codificado e a chave. Examinando o lado esquerdo verificamos que o texto claro passa por uma permutação inicial, cujo objetivo é reorganizar os bits para produzir a entrada permutada.

A segurança do DES não é afetada diretamente com a permutação inicial, pois apenas executa uma simples permutação de bits facilitando assim a implementação do DES em chips. (Hinz: 2000).

Após a permutação inicial são executadas 16 rodadas da mesma função, que envolve funções de permutação e substituição. A cada volta é executada a função F. Na entrada da função o bloco de 64 bits é dividido na metade, formando um bloco de 32

bits à esquerda e outro de 32 bits à direita e uma sub chave. Estas chaves são unidas novamente na última volta com uma permutação final que é o inverso da permutação inicial. (Hinz: 2000).

A caixa-s é composta por uma matriz de 4 linhas por 16 colunas, recebendo 6 bits de entrada e devolvendo 4 bits na saída. Na caixa-s é onde acabam ocorrendo as substituições dos bits. (Hinz: 2000).

A *expansion permutation* além de deixar a metade direita dos dados do mesmo tamanho da chave, tem o objetivo de oferecer uma maior dependência dos dados de forma que os bits do código fonte sejam dependentes de cada bit do código que será gerado, ocasionando o efeito avalanche (Hinz: 2000).

A função F realiza em cada volta as seguintes operações segundo Hinz (2000):

- A chave é transformada em 48 bits;
- A metade direita dos dados sofre a expansão de 32 para 48 bits através da *expansion permutation;*
- Esta metade direita é aplicado um XOR com a chave;
- Os dados da metade da direita são mandados para oito caixa-s produzindo 32 bits no total;
- Estes dados são permutados novamente
- Na saída, a metade esquerda é submetida a um XOR
- A metade esquerda se tornará a metade direita dos dados e a metade direita se tornará a metade esquerda na próxima volta.

O processo de decriptografia utiliza o mesmo algoritmo de criptografia, exceto pela inversão da aplicação das subchaves como afirma Stalling (2008).

4.3.2 3DES

Em 1977, a IBM percebeu que o tamanho da mensagem DES era muito pequeno e criou uma forma de aumentá-lo usando a criptografia tripla (Tanenbaum: 2003)

Este algoritmo é utilizado aplicando o algoritmo DES três vezes e utilizando apenas duas chaves. No primeiro, o texto claro é usado DES no processo de criptografia e usando uma chave k1. Já no segundo estágio é utilizado o algoritmo DES

no processo de decriptografia e empregando outra chave k2 e no terceiro e último estágio é feita a criptografia com o algoritmo DES e utilizando a mesma chave do primeiro estagio (k1) conforme mostrado na Figura 2 abaixo. (Tanenbaum: 2003).

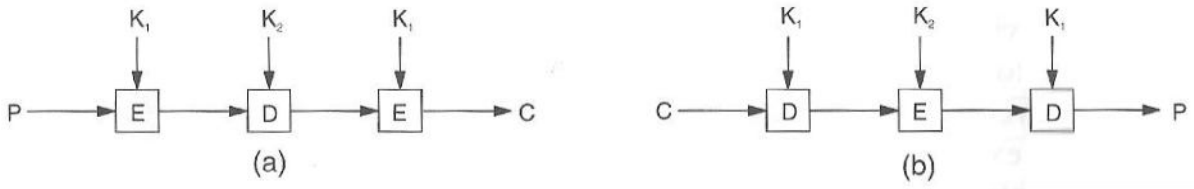

Figura 2 – (a) Criptografia tripla usando o DES. (b) Descriptografia

Fonte: TANENBAUM (2003)

Segundo Burnett e Paine (2002) o Triple DES apresentava dois problemas: o primeiro foi à descoberta de uma maneira para simplificar o ataque de força bruta, que ao invés de ter de quebrar 168 bits precisaria quebrar apenas 108 bits, considerado seguro por muitos estudiosos, porém esta falha incomodava. O segundo problema diz respeito a sua velocidade, como o DES o processo de criptografar e decriptografar é muito lento, o 3DES levaria 3 vezes mais tempo para efetuar estas tarefas.

4.3.3 AES (ADVANCED ENCRYPTION STANDARD)

O órgão NIST encarregado para aprovar padrões para o Governo Federal dos Estados Unidos percebeu que a necessidade de um novo sistema criptográfico, pois o DES e o 3DES estavam com a vida útil no fim. Porém eles tinham o receio de com o anúncio desse novo padrão, as pessoas que soubessem algo sobre criptografia acabassem deduzindo automaticamente que *National Security Agency* (NSA) havia criado uma porta dos fundos no DES, conseguindo assim ler tudo o que foi criptografado por ele e o novo padrão criptográfico não fosse utilizado e provavelmente desaparecesse (Tanenbaum: 2003)

Para solucionar este problema a NIST adotou uma estratégia diferente, patrocinando um concurso de criptografia chamado *Advanced Encryption Standard*

(AES). Em 1997, pesquisadores do mundo inteiro foram convidados para participar deste concurso. (Tanenbaum: 2003)

Os algoritmos para serem aprovados deveriam ser desenvolvidos em cifra de blocos simétrica que aceitavam chaves iguais a 128, 192 e 256 bits. Além disso, todo o projeto teria de ser público e com a possibilidade de implantação tanto em hardware como em software. (Tanenbaum: 2003)

Os principais critérios para a avaliação dos algoritmos estavam ligados a segurança (não possuir fraqueza algorítmica), desempenho (rapidez em várias plataformas) e tamanho (ocupar pouco espaço e memória). Para a escolha do vencedor foram realizadas conferências públicas nas quais os participantes eram encorajados a descobrir falhas. Na última conferência foi realizada uma votação que teve como vencedor Rijndael dos autores Joan Daemen e Vicent Rijmen, com 84 votos. (Burnett; Paine: 2002)

No algoritmo vencedor, o tamanho do estado, ou seja, a matriz de bytes que inicialmente contem a mensagem, depende do bloco utilizado, sendo composta por 4 linhas e Nb colunas, onde Nb é o número de bits do bloco dividido por 32. (Souza; Oliveira : 2011)

O Algoritmo apresenta rodadas, iterações**,** que por sua vez possuem 4 etapas: *AddRoundKey, SubBytes, ShiftRows e MixColumns*, esta última não é realizada na iteração final. O Algoritmo também apresenta uma chave principal e a partir dela, serão geradas Nr +1 chaves, onde Nr é o número de rodadas utilizadas durante a execução do algoritmo. (Souza; Oliveira : 2011)

A transformação *SubBytes* é a etapa onde cada byte do estado é substituído por outro em uma *S-Box*(caixa de substituição). “Os quatro primeiros e os quatro últimos bits do byte a ser substituído representam, respectivamente, a linha e a coluna onde se encontra o novo byte”. (Souza; Oliveira: 2011)

Na etapa de transformação *ShiftRows* consiste, segundo Souza e Oliveira, “em rotacionar à esquerda as linhas do estado, trocando assim a posição dos bytes. O número de posições a serem rotacionadas depende da linha e do tamanho do bloco utilizado”, como podemos observar no Quadro 1 onde Ci representa o número de posições a serem rotacionadas na linha de posição i de um bloco com Nb colunas.

N_b	C_0	C_1	C_2	C_3
4	0	1	2	3
6	0	1	2	3
8	0	1	3	4

Quadro 1 - Deslocamento em função de Nb e Ci

A transformação *MixCollumns* pode ser representada por uma multiplicação de matrizes. “O resultado da operação em uma determinada coluna não influencia o resultado nas demais. Porém, a mudança de um byte em uma coluna influencia o resultado na coluna inteira”. (Souza; Oliveira: 20??)

Outra transformação utilizada é a *AddRoundKey* como afirma Souza e Oliveira “é uma operação de XOR byte a byte entre o estado e a chave da rodada”.

Tanenbaum (2003) relatou que em novembro de 2001, o Rijndael se tornou um padrão do Governo dos Estados Unidos publicado como *Federal Information Processing Standart* (FIPS 97), conjunto de normas públicas desenvolvidas pelo Governo dos Estados Unidos.

4.4 CRIPTOGRAFIA ASSIMÉTRICA

A criptografia simétrica teve sempre como o seu elo mais fraco a distribuição de chaves, pois como afirma Tanenbaum(2003) não adianta ter uma sistema criptográfico sólido, impenetrável, se os usuários forem forçados a compartilhar suas chaves ou terem elas roubadas.

A grande vantagem da criptografia assimétrica é que permite qualquer pessoa enviar uma mensagem, apenas utilizando a chave pública de quem vai recebê-la, não havendo a necessidade de um compartilhamento da mesma chave, aumentando assim a segurança (Schneier: 1996).

Burnett e Paine (2002) afirmam que são necessárias ambas as chaves para criptografar e decriptografar, podendo uma se tornar pública, sem colocar a segurança em perigo. Para fazer a criptografia é utilizada a chave pública e para decriptografar é utilizado à chave privada.

A criptografia assimétrica segundo Stallings (2008) possui as seguintes etapas essências:

- Cada usuário gera um par de chaves a ser utilizado na criptografia e na decriptografia;
- Cada usuário coloca uma das chaves em registro público, tornando-a uma chave pública e a outra chave permanece privada;
- Se Bob desejar enviar uma mensagem para Alice, Bob faz a criptografia usando a chave pública de Alice;
- Quando Alice receber a mensagem ela decriptografa utilizando sua chave privada.

A Figura 3 mostra o funcionamento da criptografia da mensagem enviada de Bob para Alice e também como ocorre à autenticação do usuário.

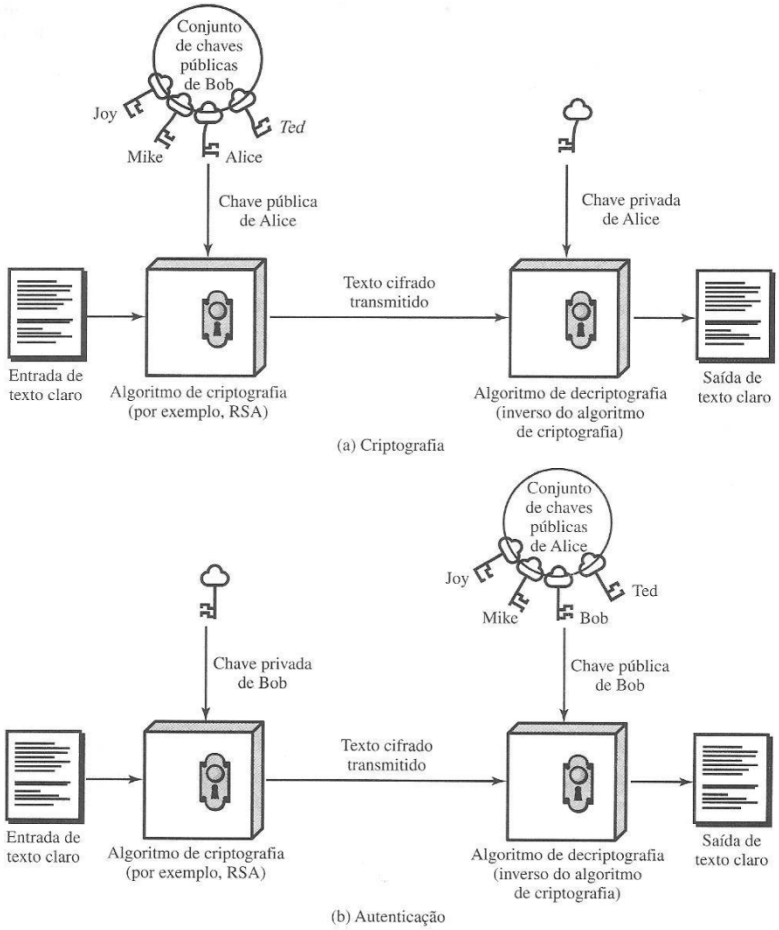

Figura 3 – Criptografia de chave publica

Fonte: STALLING (2003)

Esta técnica permite que todos os participantes tenham acesso as chaves públicas, já as chaves privadas são geradas localmente para cada participante, não sendo preciso à distribuição das mesmas. A informação recebida estará protegida desde que a chave privada de um usuário permaneça protegida e secreta (Stallings: 2008)

4.4.1 DH (Diffie, Hellman)

"Em 1976, dois pesquisadores da University of Stanford, Diffie e Hellman, propuseram um sistema de criptografia radicalmente novo, no qual as chaves de criptografia e decriptografia eram diferentes". Este algoritmo desenvolvido foi chamado de DH (Diffie, Hellman). (Tanenbaum: 2003).

Com isso os pesquisadores tentaram resolver dois dos problemas mais difíceis relacionados à chave simétrica. O primeiro como já foi dito seria sobre a distribuição das chaves e o outro seria relacionado à assinatura digital, ou seja, todas as partes teriam a certeza de quem teria enviado a mensagem. (Stalling: 2008).

"O DH resolve o problema de distribuição de chaves, mas não é o algoritmo final. Ele não é efetivo para a criptografia, mas ainda é muito utilizado até hoje", cita Burnett e Paine (2002).

4.4.2 RSA

O algoritmo de Diffie e Hellman, o DH, apresentou uma nova técnica para a criptografia e devido as suas vantagens potenciais, estão desafiando os criptologistas a desenvolverem um algoritmo criptográfico que atendessem os requisitos para o sistema de chave pública. (Stalling: 2008)

Logo nas primeiras respostas ao desafio, um grupo de pesquisadores do Instituto de Tecnologia de Massachusetts (MIT), criou em 1977 um algoritmo cujo nome é dado pelas iniciais de seus nomes Rivest, Shamir, Adleman (RSA). Ele suportou por mais de quatro séculos todas as tentativas de rompimento, sendo considerado muito forte e

tornando a técnica mais aceita e implementada para a criptografia de chave pública. (Tanenbaum: 2003)

O método RSA é uma cifra de blocos em que o texto claro e o cifrado são inteiros entre 0 e n-1, onde n normalmente tem o tamanho de 1024 bits. A principal desvantagem desta técnica é exigir chaves de pelo menos 1024 bits para manter um bom nível de segurança, tornando bastante lento. (Tanenbaum: 2003)

As chaves utilizadas para cifrar e decifrar as mensagens são geradas passando pelas seguintes etapas, descritas por Kurose; Ross (2006):

- Primeiramente é necessário escolher dois números primos grandes **p** e **q** (normalmente 1024 bits);
- Estes números serão multiplicados gerando outro número **n** (n= p *q) ;
- Gera-se um numero **z** através do produto de **z**= (p-1)*(q-1);
- Escolhe um numero ***e*** tal que ***z*** e ***e*** sejam primos entre si ;
- Encontre **d** de forma que (**d** * **e**) mod[1] **z**=1 **;**

Após gerar estas chaves pode se realizar o processo de cifragem e decifragem. Para criptografar a mensagem (M) é necessário calcular $C= M^e$ mod n, gerando a mensagem criptografada (C). Para decriptografar, calcule $M= C^d$ mod n. Portanto no processo de cifragem utiliza-se o par de chaves {e,n} e para a decifragem são necessários {d,n}. Desta forma a chave pública consiste no par (e,n) e a chave privada (d,n). Floriano (2007)

Um exemplo didático do algoritmo RSA é mostrado por Kurose; Ross (2006) nos quadros 2 e 3. Neste exemplo foi escolhido p=5 e q=7, gerando n=35 e z=24. Pode ser escolhido e= 5, já que 5 e 24 não tem fatores comuns. Por fim é escolhido d=29 já que 29*5 mod 24=1.

[1] resto da divisão

Letra do texto aberto	m: representação numérica	m^e	Texto cifrado $c = m^e \bmod n$
L	12	248832	17
O	15	759375	15
V	22	5153632	22
E	5	3125	10

Quadro 2 – Criptografia RSA para Alice: e=5, n=35

Fonte: KUROSE; ROSS (2006)

Observa-se no quadro 3 que embora utilizado este pequeno exemplo os números produzidos foram extremamente grandes, pois como afirma Kurose; Ross (2006) os números “p e q devem ter, cada um, algumas centenas de bits de comprimento”

Texto cifrado c	c^d	$m=c^d \bmod n$	Letra do texto
17	48196857210675091509141182522307169 7	12	L
15	12783403948858939111232757568359375	15	O
22	85164331908653770195619449972111060305 92	22	V
10	10000000000000000000000000000000	5	E

Quadro 3 – Decriptação RSA para Bob: d=29, n=35

Fonte: KUROSE; ROSS (2006)

Segundo Tanenbaum (2003) “a segurança do método se baseia na dificuldade de fatorar números extensos”. Se o valor de n publicamente conhecido pudesse ser fatorado, o criptoanalista poderia então encontrar p e q e, a partir desses calcular z. Conhecendo z e e, será possível encontrar d utilizando o algoritmo de Euclides. “Felizmente, os matemáticos tem tentado fatorar números extensos por pelo menos 300 anos, e o conhecimento acumulado sugere que o problema é extremamente difícil” (Tanenbaum:2003)

O processo de exponenciação exigida pela RSA consome uma grande quantidade de tempo. Já o DES é no mínimo cem vezes mais veloz em software e

cerca de 10 mil vezes mais veloz no hardware. Como resultado, o RSA é utilizado frequentemente com o DES ou com o AES. Primeiramente o DES gera a chave e ela é criptografada pelo RSA. Assim os dados são criptografados pelo DES e as chaves pelo RSA, tornando o processo de cifragem e decifragem muito mais rápido e oferecendo a segurança na chave simétrica. (Kurose; Ross: 2006)

4.5 CRIPTANÁLISE

"A criptanálise é a parte da criptografia que estuda as técnicas para obter o texto original a partir do texto cifrado" (SCHNEIER:1996), ou seja, a criptanálise é um ramo da criptografia que estuda técnicas para decodificar as mensagens sem possuir as chaves secretas. (Floriano:2007)

Schneier (1996) explica quatro tipos gerais de ataques criptoanalítico, em todos é preciso ter o conhecimento completo do algoritmo de criptografia utilizado, sendo eles:

- **Ataque sobre mensagens cifradas**: o criptoanalista necessita ter várias mensagens cifradas pelo mesmo algoritmo e assim recuperar o texto original das mensagens possíveis, para assim deduzir a chave ou chaves usadas para decifrar a mensagem e no futuro decifrar novas mensagens que utilizam esta mesma chave.
- **Ataque sobre mensagens originais:** nesta técnica o criptoanalista além de possuir a mensagem cifrada, ele tem a mensagem original e o seu objetivo é de descobrir a chave secreta utilizada, a fim de decifrar novas mensagens que venham a ser cifradas com esta chave.

- **Ataque escolhendo blocos das mensagens originais**: o criptoanalista tem o acesso as mensagens originais e as cifradas. Este ataque leva vantagem ao anterior sendo visto que o criptoanalista pode escolher os blocos específicos das mensagens, os quais poderão oferecer um melhor resultado e assim deduzir a chave utilizada e

também poder decifrar outras mensagens cifradas que utilizam esta chave.

- **Ataque adaptável escolhendo os blocos das mensagens originais:** este ataque é muito similar ao anterior, exceto pela vantagem da escolha de um bloco de mensagem menor e poder escolher outros blocos durante o processo.

Outro tipo de ataque que pode ser utilizado pelos criptoanalistas é o método chamado de Força Bruta que segundo Stalling (2008) “envolve a tentativa de cada chave possível até que seja obtida uma tradução inteligível de texto cifrado para texto claro”. Na média é necessário utilizar metade das chaves possíveis para que consiga obter sucesso. (Stalling: 2008)

“O criptoanalista deve levar em consideração fatores de complexidade de uma criptanálise, ou seja, devem levar em consideração recursos de tempo, memória e dados necessários para realizar a criptanálise.” (Floriano:2007)

5 METODOLOGIA

Como proposta para o presente estudo, inicialmente foi realizada uma pesquisa bibliográfica que segundo DOMINGUES; HEUBEL; ABEL (2003) as pesquisas devem conter assuntos gerais e particulares podendo ser localizadas em diversas fontes de pesquisas como periódicos, livros e materiais digitais nos quais tende a ter a facilidade em encontrar assuntos sobre criptografia.

Com o término desta pesquisa foi realizada uma procura por software de maneira aleatória, priorizando os que apresentavam uma interface amigável e utilizava a criptografia AES. Foram escolhidos: *Chiave File Encryption, Sfx Creator e S.L Encrypt File*, todos freeware. Também foi utilizado o software auxiliar Gerador de Chaves para Criptografia assimétrica, responsável por gerar as chaves usadas para realizar a criptografia.

Após o entendimento do funcionamento dos softwares foram realizados testes para coletar as vantagens e desvantagens, podendo assim ter um comparativo de cada software utilizando a criptografia AES.

Para realizar a criptografia foram escolhidos quatro tipos de arquivos de maneira aleatória, visando os arquivos mais comuns utilizados no computador. Foram criados arquivos de: música, imagem, documentos e filme com aproximadamente 300 MB cada. Por fim estes arquivos foram unidos criando um único arquivo de 1,20 GB que recebeu o nome de “todos”. Estes arquivos com exceção do de vídeo foram zipados utilizando o programa *Winrar*.

Para a medição do tempo foi utilizado um cronometro de relógio, podendo assim ocorrer uma variação do tempo obtido. Na verificação da utilização da CPU e da memoria foi utilizado o gerenciador de tarefas do Windows, deixando o visível enquanto os testes eram realizados.

Os testes foram realizados utilizando um notebook HP Pavilion dv5-1260br com processador AMD TurionTM X2 ultra, com 4 GB de memória DDR2 e um disco rígido de 250 GB. O sistema utilizado foi Windows 7 Professional de 64 bits.

6 Resultados e Discussões

Neste capítulo serão apresentadas as características de cada software e os resultados obtidos nos testes realizados. Posteriormente será mostrado o desempenho obtido entre os softwares avaliados.

- ***Chiave File Encryption***

O *Chiave File Encryption* apresenta uma interface intuitiva como mostra a figura 4, além de proporcionar ao usuário o poder de adicionar vários arquivos para criptografar sem a necessidade de compactá-los.

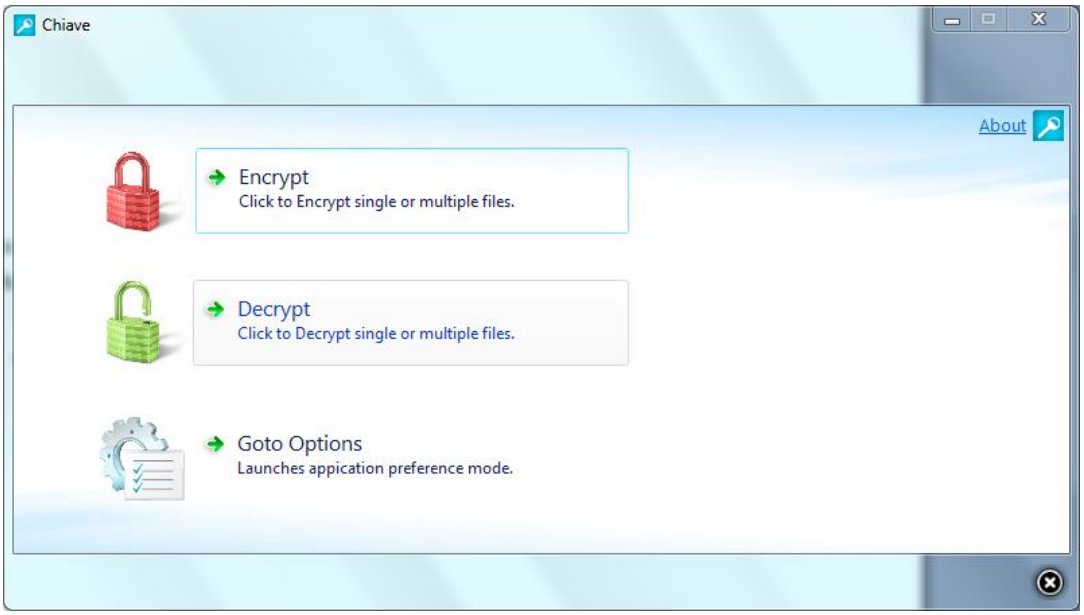

Figura 4 – *Chiave File Encryption* Interface

O software permite ao usuário escolher se os arquivos que foram criptografados irão ser excluídos. Os arquivos criptografados possuem uma extensão particular .enf que só podem ser decriptografadas utilizando o próprio software.

Para realizar a criptografia é necessário clicar no botão “Encrypt” e escolher quais arquivos deseja criptografar em “add Files” ou “add Folder”, neste exemplo foi escolhido o arquivo imagem.rar, mostrado na figura 5 . Será necessário digitar uma senha de seis dígitos no mínimo e clicar no botão Start Encryption.

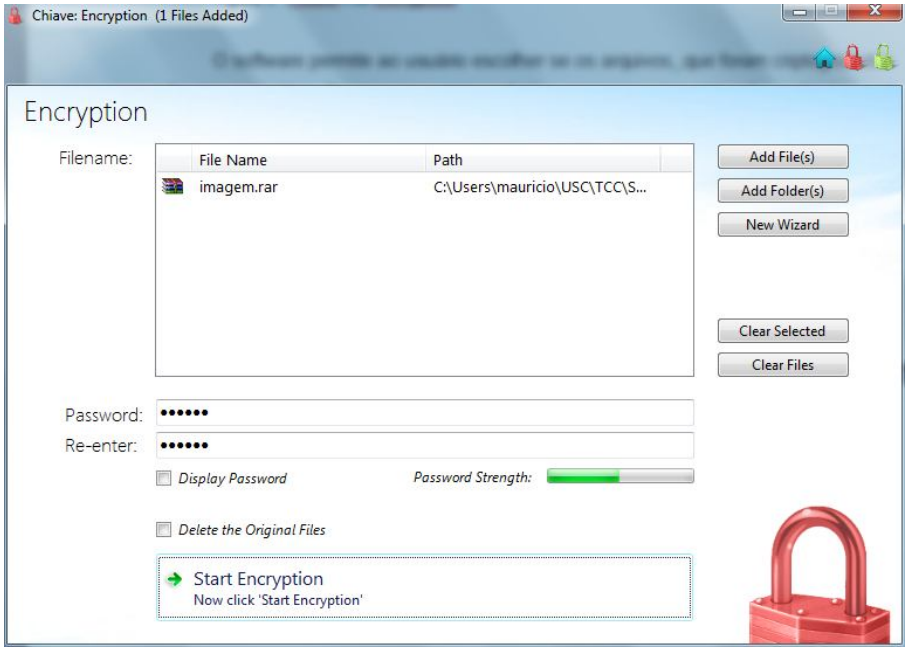

Figura 5 - Tela de criptografia do Chiave File Encryption

O processo de decriptografia é semelhante ao anterior, basta escolher a opção “Decrypt” e digitar a mesma senha utilizada para a criptografia. Clique em “Start Decryption” para retornar os arquivos originais. A figura 6 abaixo mostra o processo de decriptografia.

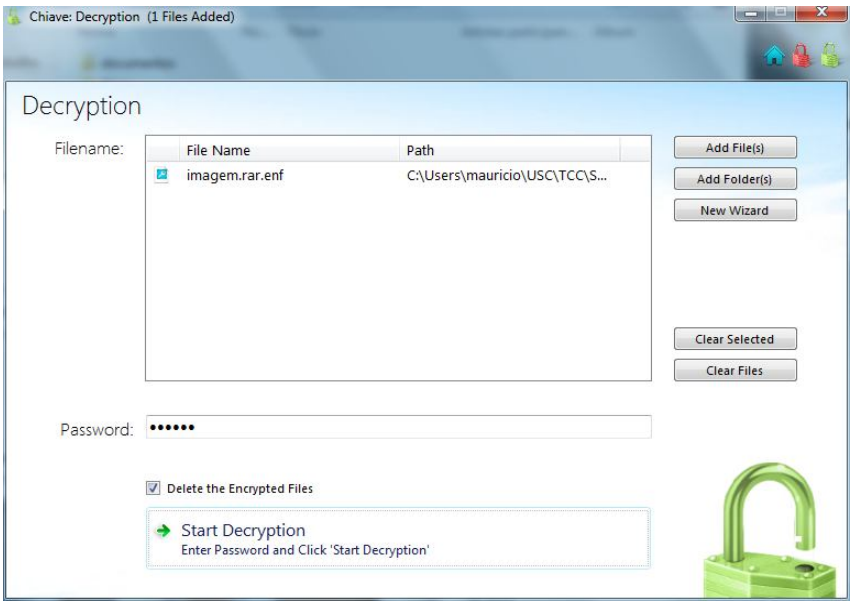

Figura 6 - Tela de decriptografia do Chiave File Encryption

No gráfico 1 serão mostrados os resultados obtidos a partir da criptografia realizada nos arquivos, utilizando a seguinte chave de 128 bits: 0B3555A25CC76C154E7B4273C88E0F8E

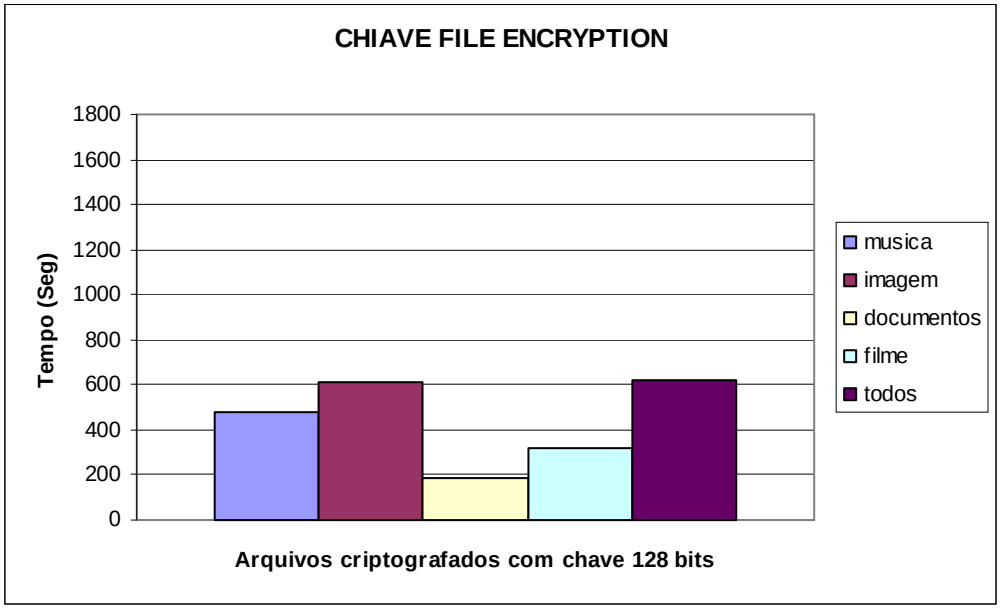

Gráfico 1 – Desempenho do Chiave File Encryption na criptografia com chave de 128 bits

Verifica-se que o arquivo formado por documentos possui o menor tempo de processamento em torno de 200 segundos, seguido por arquivo de filme e música. Um fator relevante é em relação aos arquivos de imagem e todos que apresentam o mesmo tempo de execução 600 segundos embora o arquivo todos tenha o tamanho quatro vezes maior.

Foi realizado outro teste com os mesmos arquivos, mas utilizando uma chave de 256 bits: 819590E083BA2429596E1423B3A838635FBE6B53E782F43F1B27EC18EBCAA28F e os resultados são mostrados no Gráfico 2:

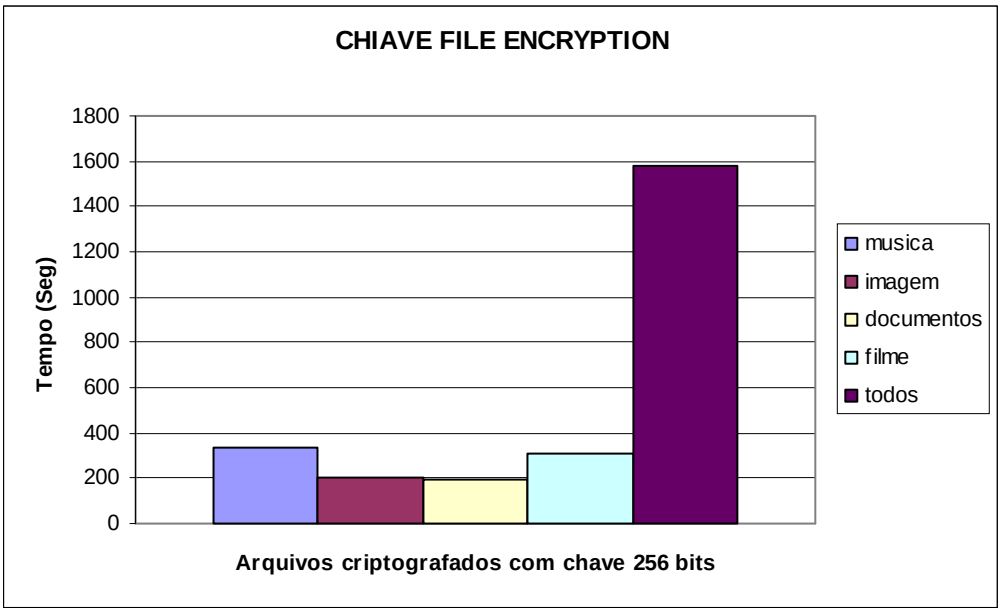

Gráfico 2 - Desempenho do Chiave File Encryption na criptografia com chave de 256 bits

Observa-se que o arquivo documentos permanece sendo o mais rápido com 200 segundos, acompanhado pelo arquivo de imagem que quando testado com a chave de 128 bits obteve o pior resultado. O arquivo de musica melhorou o seu desempenho enquanto o arquivo de filme continuou com o mesmo tempo e o arquivo todos foi o qual sofreu a maior alteração em seu tempo, aumentando mais de duas vezes do seu tempo em relação ao teste realizado anteriormente.

O software *Chiave File Encryption* também foi testado no processo de decriptografia dos arquivos encriptografados e o seu desempenho é mostrado no gráfico 3.

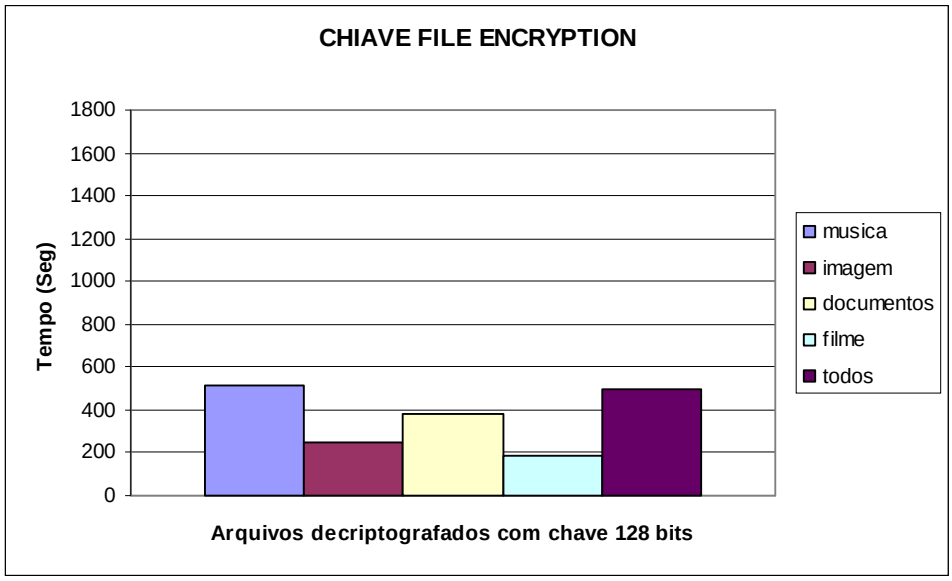

Gráfico 3 - Desempenho do Chiave File Encryption na decriptografia com chave de 128 bits

Os tempos obtidos no processo de decriptografia foram melhores do que o processo de criptografia, exceto pelo arquivo de documentos que teve queda no seu desempenho. Faz-se necessário destacar o desempenho do arquivo de imagem que junto com arquivo de filme alçaram os melhores tempos em torno de 200 segundos.

O gráfico 4 mostra que embora o número de bits da chave tenha dobrado, a decriptografia atingiu melhores resultados do que a decriptografia com chave de 128 bits, menos o arquivo todos que perdeu seu rendimento quando o teste utilizou chaves de 256 bits.

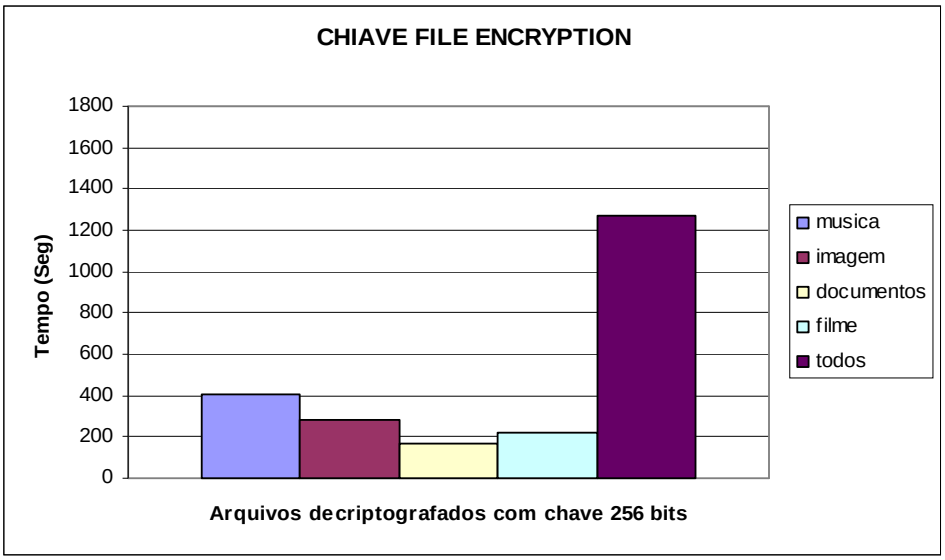

Gráfico 4 - Desempenho do Chiave File Encryption na decriptografia com chave de 256 bits

Nos testes também foram analisados a memória utilizada pelo programa e a utilização da CPU durante o processo de criptografia e decriptografia. O *Chiave file Encryption* utilizou em média de 45 a 50 por cento da CPU e necessitou entre 23 à 27 MB.

- **SFX CREATOR**

A grande diferença do *SFX Creator* é oferecer ao usuário a opção de escolher os algoritmos de criptografia finalista do concurso AES e o *blowfish*. O software apresenta também um gerador de chaves que não foi utilizado nos testes.

Este programa proporciona escolher apenas um arquivo a ser criptografado, havendo a necessidade de "zipar" os documentos caso deseje criptografar mais arquivos.

Após criptografado, o programa proporciona a escolha de três extensões: exe, zip e cab. Nos testes foi desprezada a extensão zip por necessitar um período de tempo muitas vezes maior do que os outros, cerca de 3 horas (10800 segundos).

O processo de criptografia é bem simples, basta escolher o arquivo a ser criptografado, digitar a senha e escolher a criptografia, neste exemplo foi utilizado o arquivo imagem e a criptografia *Rijndael*, mostrado na figura 7. Para iniciar a criptografia é preciso clicar no botão "Encrypt".

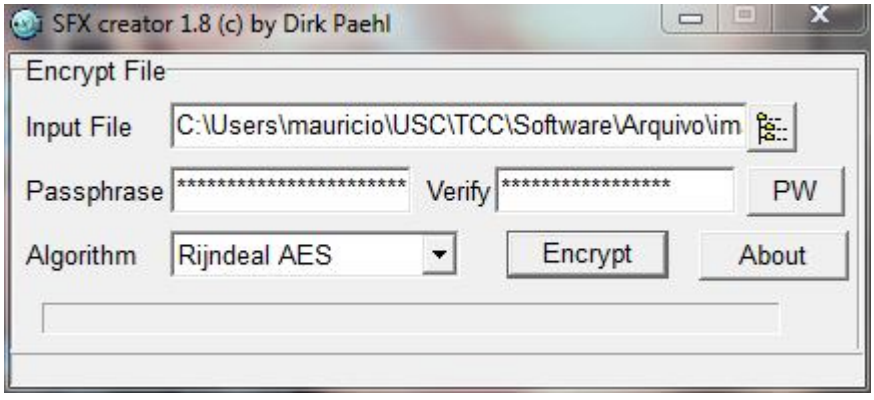

Figura 7 - Tela de criptografia do SFX Creator

Após o procedimento será perguntado qual a extensão deseja escolher. Este é um lado negativo do software por exigir uma interação do usuário no meio do processo se desejar escolher as extensões cab ou zip, pois a extensão exe já é gerada automaticamente.

Para realizar a decriptografia não é necessário executar o *SFX Creator* e sim o arquivo gerado pela criptografia, sendo necessário digitar a senha utilizada e clicar em "decrypt" como mostra a figura 8.

Figura 8 - Tela de decriptografia do SFX Creator

Os testes realizados por este software foram divididos em duas partes: arquivos com a extensão exe e o outro com a extensão cab.

No primeiro teste foi utilizada a mesma chave de 128 bits do teste realizado pelo software *Chiave File Encryption* e o resultado pode ser visualizado no gráfico 5 abaixo:

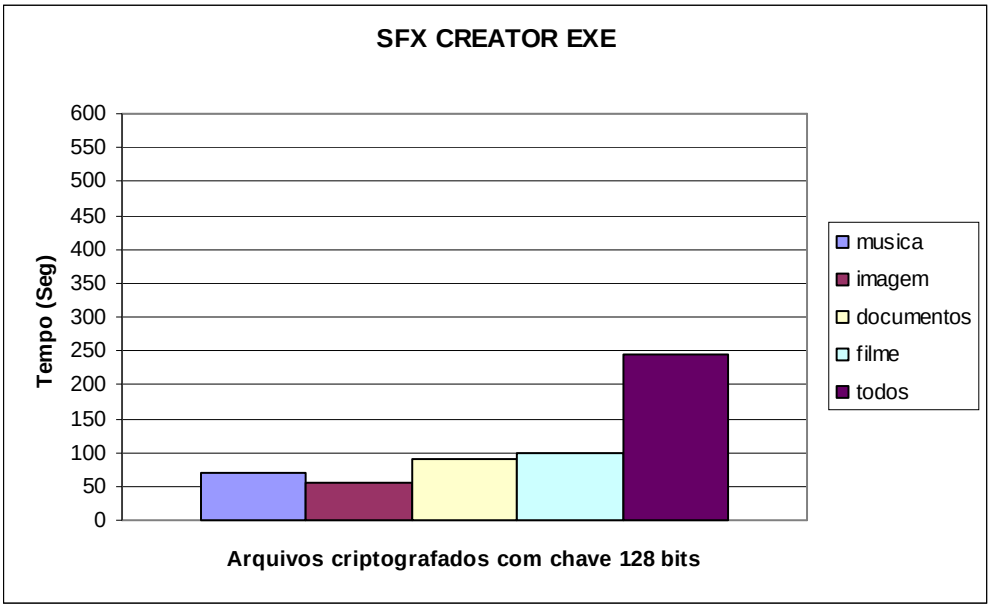

Gráfico 5 - Desempenho do *SFX Creator* EXE na criptografia com chave de 128 bits

Na extensão exe os resultados obtidos entre os arquivos foram semelhantes, variando entre 50 a 100 segundos com exceção do arquivo todos que teve um desempenho equivalente se for levado em conta o seu tamanho quatro vezes maior.

Já no gráfico 6, verifica-se que o desempenho do *SFX Creator* CAB possui uma variação mais acentuada entre 100 a 200 segundos. Os arquivos imagem, documentos e filme obtiveram um acréscimo por volta de 50 segundos ao contrário dos arquivos de musica e todos que tiveram um aumento de 142 e 296 segundos respectivamente.

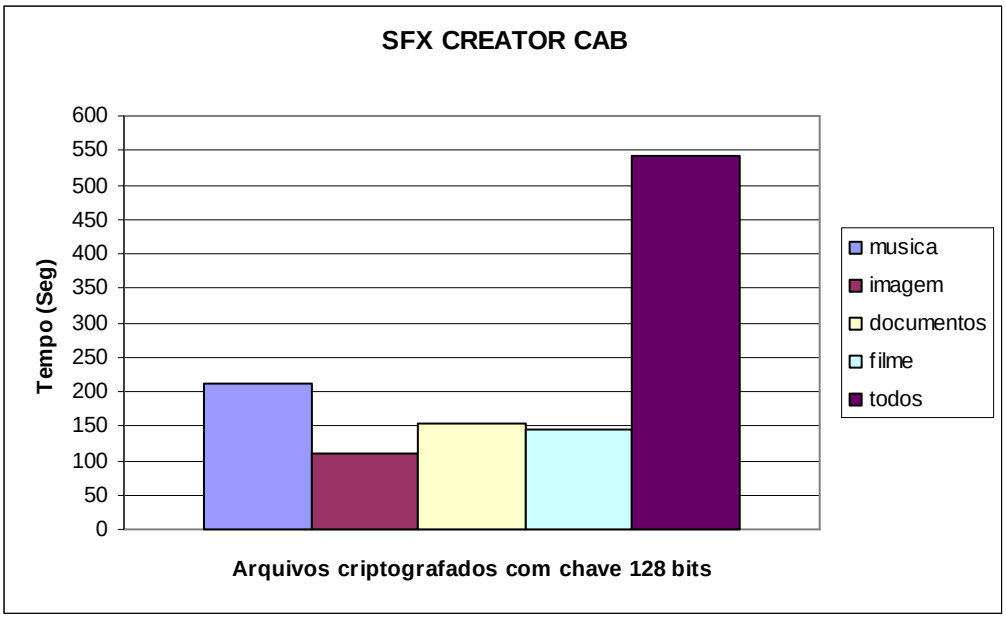

Gráfico 6 - Desempenho do SFX Creator CAB na criptografia com chave de 128 bits

No segundo teste foi alterada a chave de utilização de 128 para 256 bits e os resultados obtidos pela extensão exe teve um tempo menor de execução do que o teste realizado anterior, visualizados no Gráfico 7. O arquivo “todos” como no experimento do outro software teve um decréscimo no teu desempenho quando a chave de criptografia foi aumentada.

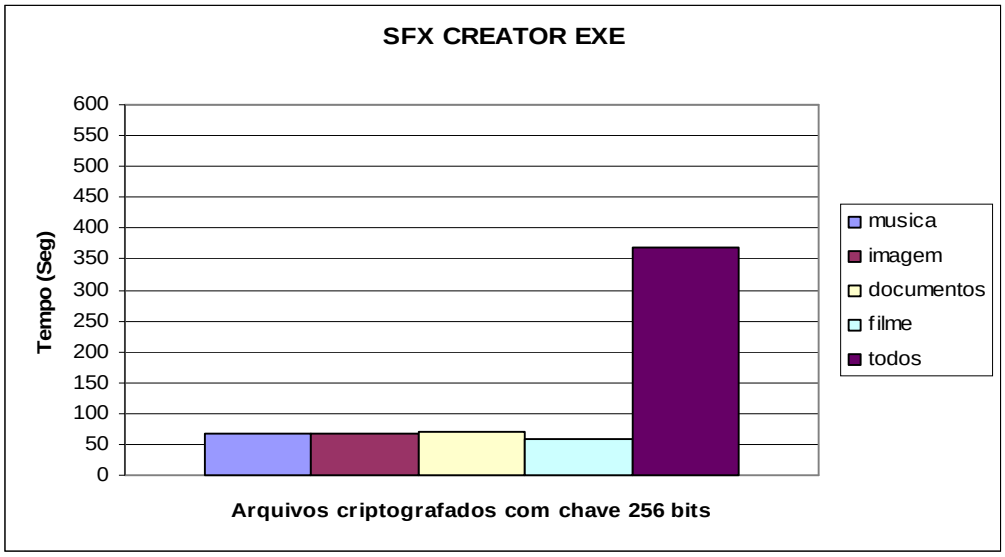

Gráfico 7 - Desempenho do SFX Creator EXE na criptografia com chave de 256 bits

No Gráfico 8 são observados os arquivos criptografados com 256 bits com a extensão cab que tiveram os tempos parecidos dos que foram criptografados com 128 bits, obtendo uma variação de 5 a 35 segundos com exceção do arquivo de imagem que teve seu tempo acrescido em 51 segundos.

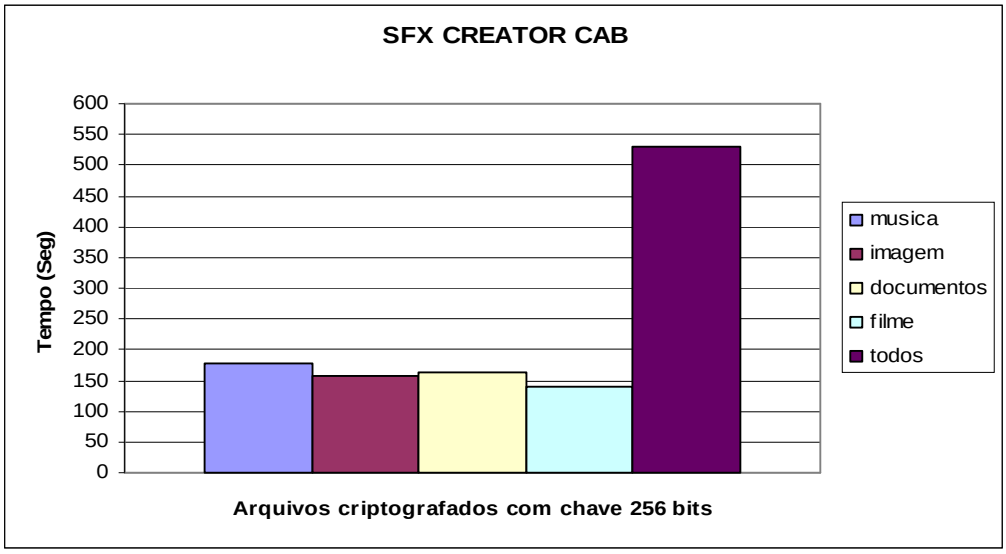

Gráfico 8 - Desempenho do SFX Creator CAB na criptografia com chave de 256 bits

Após analisados os resultados obtidos pelo software *SFX Creator* no processo de criptografia, será discutido o desempenho do mesmo com a utilização da decriptografia.

Excluindo o arquivo todos que teve seu tempo próximo aos 140 segundos, os demais arquivos obtiveram tempos próximos, variando entre 25 a 34 segundos empregando a chave de 128 bits, como pode ser analisado no Gráfico 9 abaixo:

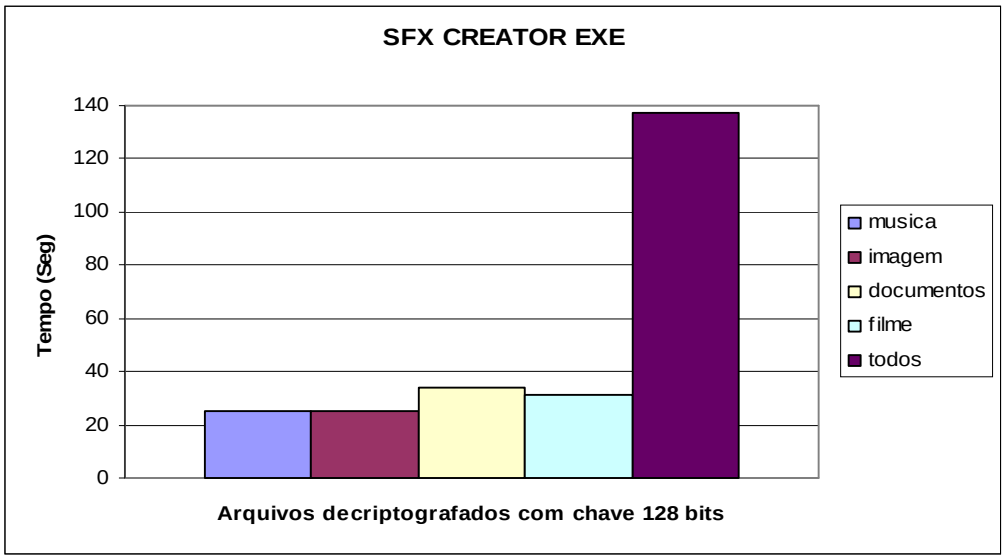

Gráfico 9 - Desempenho do SFX Creator EXE na decriptografia com chave de 128 bits

Comparando o Gráfico 9 com o Gráfico 10 pode ser observado que o tempo de decriptografia aumentou em torno de 20 segundos para os arquivos de musica e imagem e em 7 segundos para o arquivo de filme. Ao contrário, os arquivos documentos e todos melhoraram o seu desempenho, destacando-se o último citado devido à diminuição de seu tempo em 29 segundos.

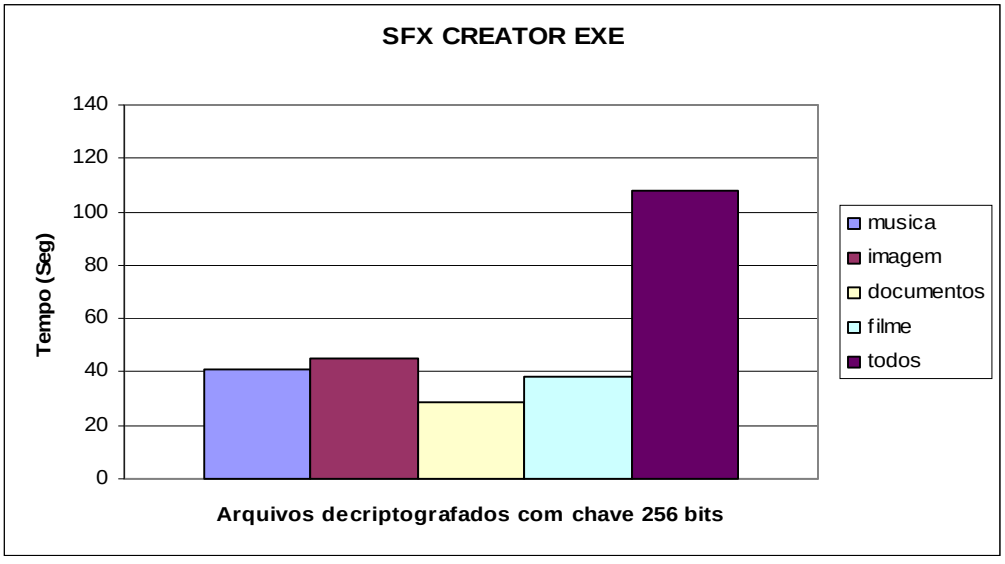

Gráfico 10 - Desempenho do SFX Creator EXE na decriptografia com chave de 256 bits

A decriptografia realizada pelo software *SFX Creator* com a extensão cab obteve um resultado parecido para os arquivos de musica e filmes mesmo quando sua chave de 128 bits foi alterada para 256 bits. Observa-se nos gráficos 11 e 12 que a maior diferença alcançada entre os arquivos foi de dois segundos.

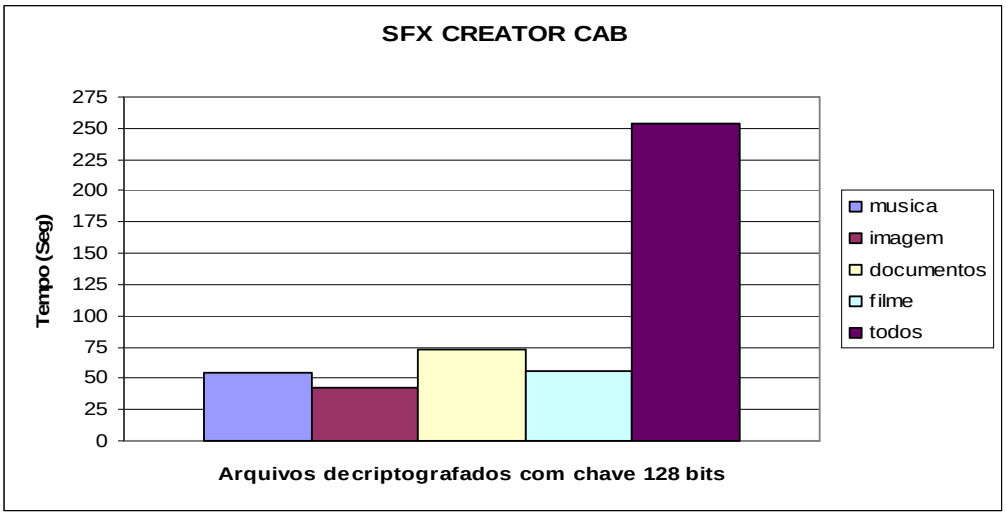

Gráfico 11 - Desempenho do SFX Creator CAB na decriptografia com chave de 128 bits

Já os arquivos de imagem e documentos apresentaram a maior diferença em seus tempos, 21 e 23 segundos respectivamente.

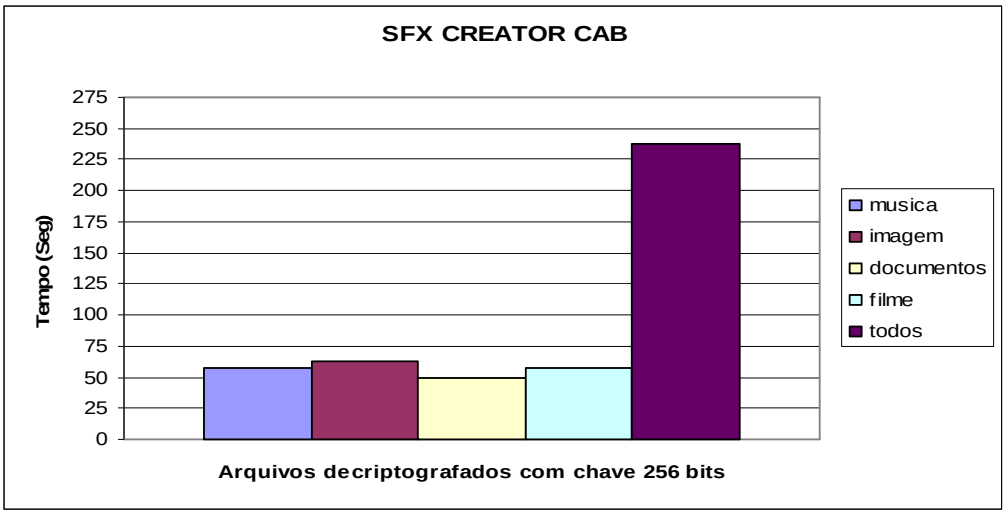

Gráfico 12 - Desempenho do SFX Creator CAB na decriptografia com chave de 256 bits

O software *SFX Creator* no teste de utilização da CPU alcançou a marca de 40 a 50 por cento e utilizou a memória entre 1,8 a 7,8 MB com exceção do arquivo todos que por um período de tempo utilizou uma grande quantidade de memória, com valores superiores a 1000 MB.

- ***S.L ENCRYPT FILE***

O software *S.L Encrypt File* possui a vantagem de não precisar ser instalado para realizar a criptografia e a decriptografia, mas tem a desvantagem de não suportar o tamanho da chave de 256 bits, por isso os testes realizados que tinham como base a chave de 256 bits foram realizados utilizando a maior chave suportada pelo software (168 bits).

A interface do programa é bastante intuitiva como verificado na Figura 9, porem é necessário ficar atento em alguns detalhes que serão descritos abaixo. O processo de criptografia e decriptografia adotam a mesma tela.

Figura 9 – S.L Encrypt files Interface

Para realizar a criptografia é necessário digitar a senha em “Your password” e escolher o arquivo a ser criptografado, sendo possível somente um arquivo por vez, caso necessite criptografar mais de um arquivo este deverá ser “zipado”. Em

“Destination file” o usuário precisará escolher onde deseja salvar o arquivo, o nome do arquivo e a sua extensão como pode ser observado na figura 10

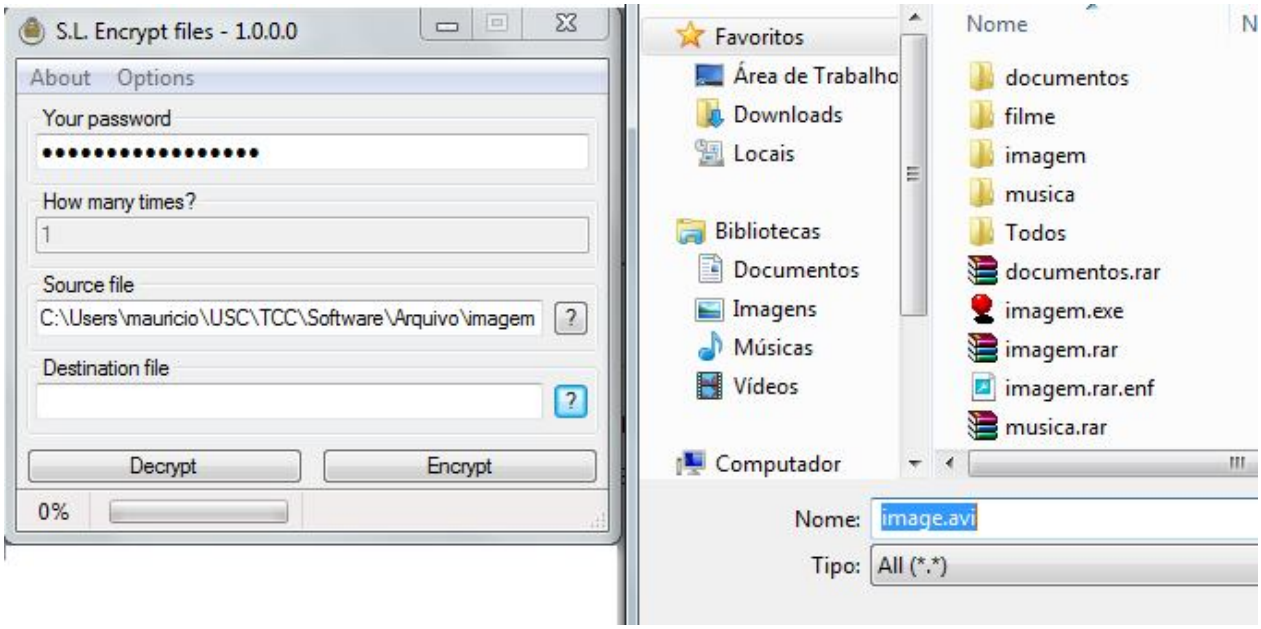

Figura 10 - Tela de criptografia do S.L. Encrypt Files com a extensão AVI

Com os experimentos realizados neste software foi verificado que com o aumento do tamanho da chave, o processo de criptografia aumentou em todos os arquivos, exceto no arquivo de musica que melhorou o seu tempo em 57 segundos, observado através dos Gráficos 13 e 14.

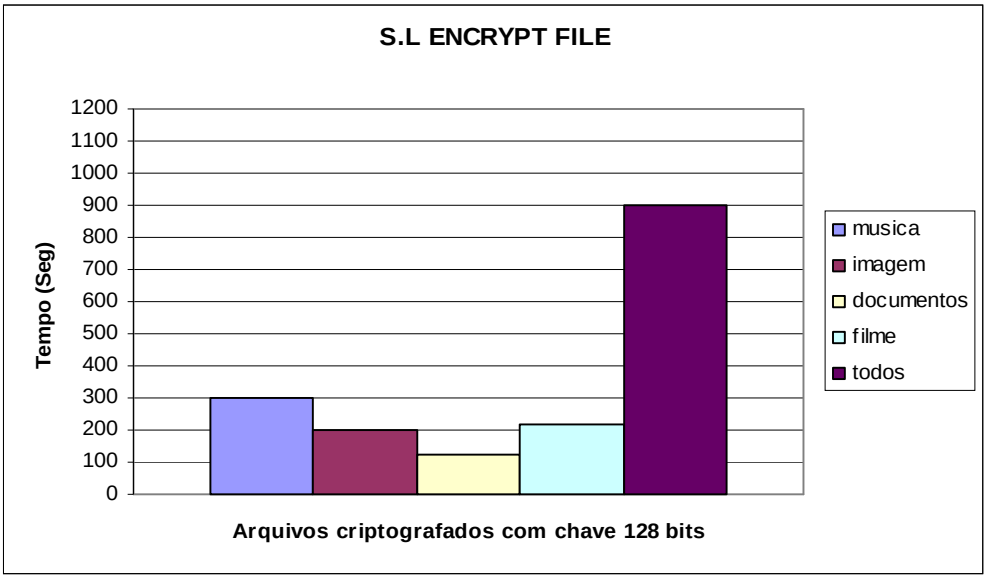

Gráfico 13 - Desempenho do S.L Encrypt File na criptografia com chave de 128 bits

A diferença entre os tempos dos arquivos foram acentuadas, aumentando 233 segundos no arquivo de todos, 106 segundos no arquivo imagem, 84 segundos nos documentos e 12 segundos nos filmes.

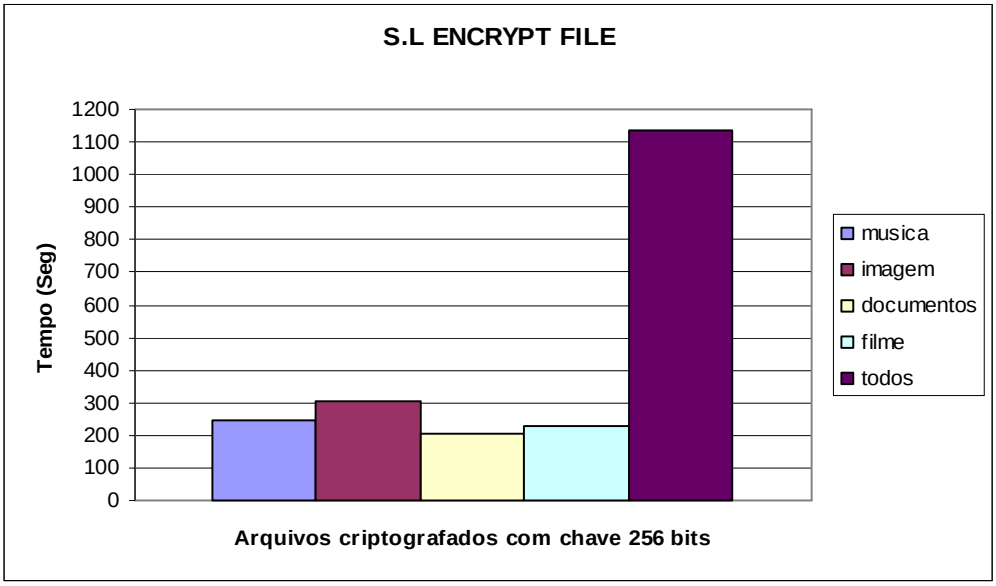

Gráfico 14 - Desempenho do S.L Encrypt File na criptografia com chave de 256 bits

Nos processos de decriptografia o arquivo todos se destaca negativamente pelo seu baixo desempenho alcançando a marca de 1200 segundos enquanto os outros arquivos obtiveram em media 223 segundos como pode ser observado pelo Gráfico 15.

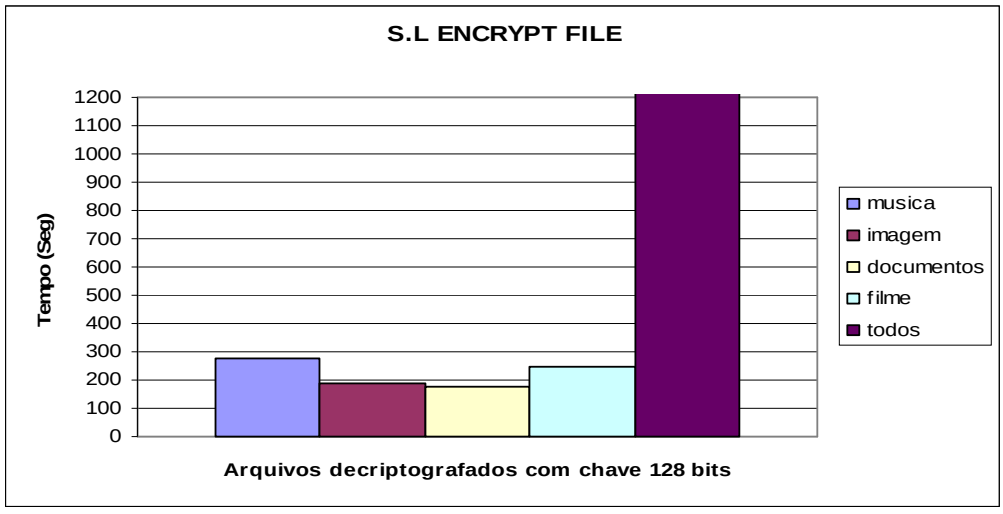

Gráfico 15 - Desempenho do S.L Encrypt File na decriptografia com chave de 128 bits

O gráfico 16 mostra a decriptografia utilizando a chave de 256 bits que obtiveram os resultados similares a criptografia de 256 bits, chegando ao diferencial máximo de 18 segundos, excluindo o arquivo todos, cuja diferença foi de 50 segundos.

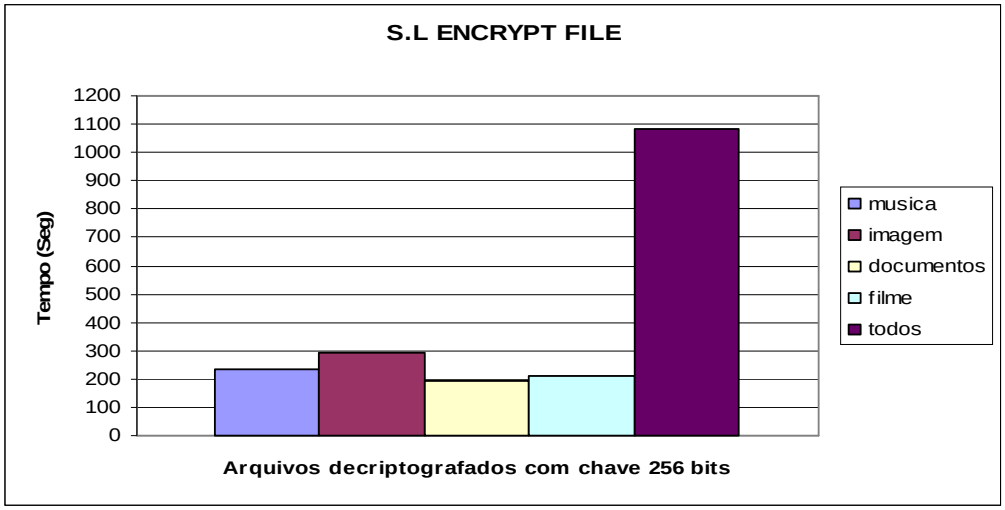

Gráfico 16 - Desempenho do S.L Encrypt File na decriptografia com chave de 256 bits

Quando foi realizado o teste da utilização da CPU, o software *S.L Encrypt File* usou de 48 a 50 por cento da CPU e necessitou uma memória de 2,1 a 3,9 MB para realizar os processos de criptografia e decriptografia.

Depois de feita a descrição detalhada de cada software e medido o seu desempenho, será realizada a comparação entre eles referente a cada tipo de arquivo. Esta comparação será feita no processo de criptografia e decriptografia onde os gráficos mostrarão o desempenho alcançado na utilização da chave de 128 e 256 bits.

O primeiro arquivo a ser analisado será o de musica. Observando o gráfico 17 é verificado que o software *SFX Creator* obteve o melhor desempenho em ambas as extensões exe e cab no processo de criptografia. Vale ressaltar o desempenho alcançado pela criptografia de 256 bits que embora utilize uma chave duas vezes maior conseguiu um tempo menor de execução, chegando no software *Chiave File Encryption* uma diferença de 150 segundos.

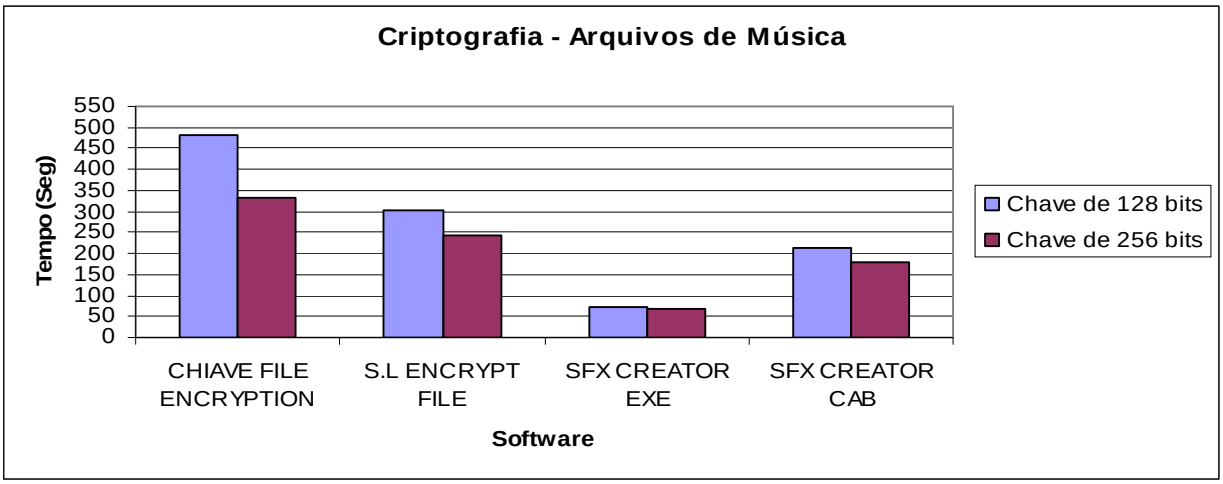

Gráfico 17- Desempenho dos softwares na criptografia do arquivo de música

No processo de decriptografia o tempo de execução foi melhor do que a criptografia como pode ser visto na comparação do gráfico acima com o Gráfico 18. A utilização da chave de 256 bits ainda possui o melhor desempenho, entretanto a diferença entre as chaves diminuiu enquanto no processo de criptografia a diferença média ficou em 218,75 segundos, o processo de decriptografia obteve a média de 129,5 segundos.

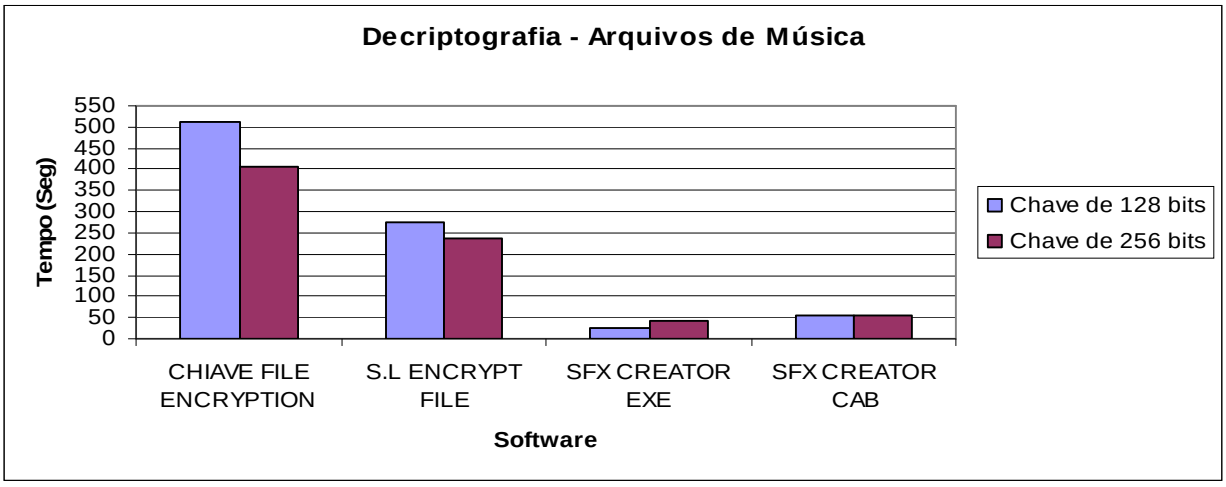

Gráfico 18 - Desempenho dos softwares na decriptografia do arquivo de música

O próximo tipo de arquivo a ser analisado é o de imagem como no teste anterior o *SFX Creator* atingiu o melhor desempenho na utilização das duas chaves, seguido pelo *S.L Encrypt File* na chave de 128 bits ou pelo *Chiave File Encryption* na chave de 256 bits como mostra o Gráfico 19. A utilização da chave de 256 bits acarretou o

aumento no tempo em todos os softwares, exceto no *Chiave File Encryption* que teve seu tempo três vezes menor.

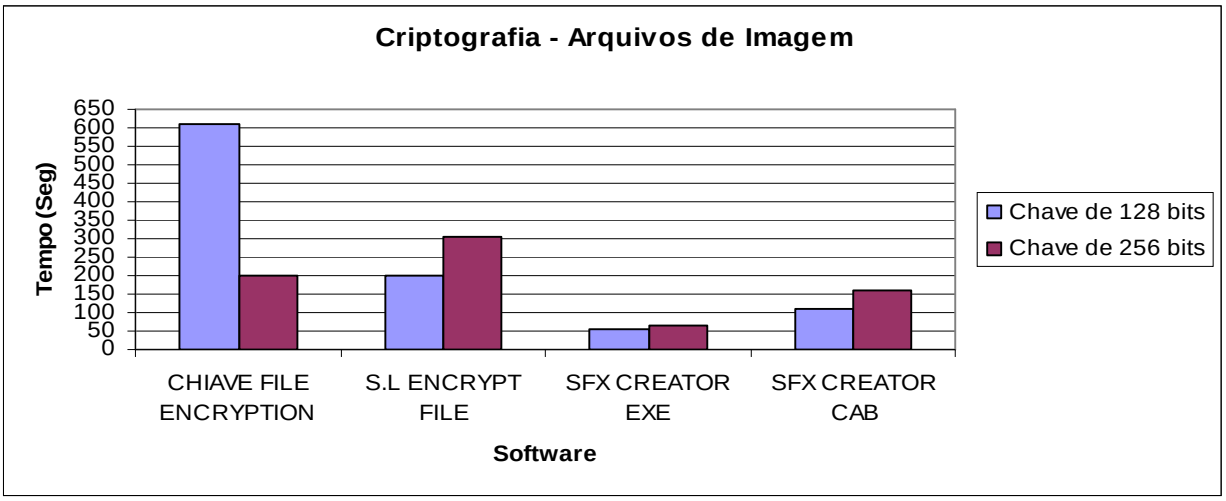

Gráfico 19 - Desempenho dos softwares na criptografia do arquivo de imagem

O processo de decriptografia necessitou a metade do tempo do processo de criptografia utilizando a chave de 128 bits. No gráfico 20, verifica-se que o *SFX Creator* além de conseguir o melhor desempenho, seu tempo foi 10 vezes melhor do que *Chiave File Encryption* e sete vezes melhor do que o *S.L Encrypt File* quando utilizada a chave de 128 bits e 6 vezes menor em relação aos outros softwares utilizando a chave de 256 bits.

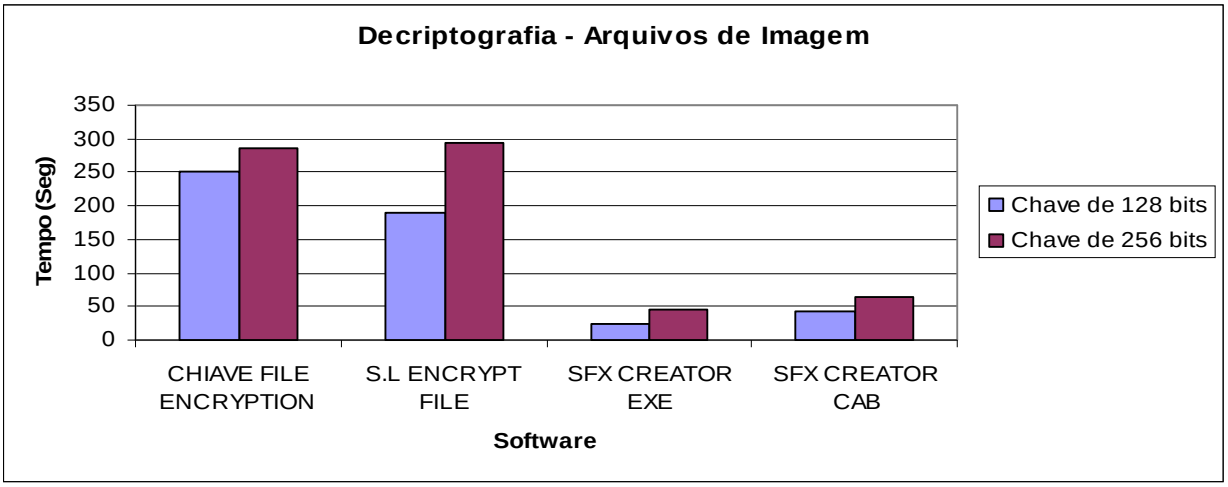

Gráfico 20 - Desempenho dos softwares na decriptografia do arquivo de imagem

Analisando o gráfico 21 observa-se que os resultados obtidos pelos softwares *Chiave File Encryption* e *S.L Encrypt File*, utilizando o arquivo documentos, ficaram mais próximos do SFX Creator, embora este último continue com o melhor desempenho na extensão exe.

Na criptografia usando a chave de 128 bits, o software *S.L Encrypt File* se destaca por conseguir o segundo tempo superando o SFX CAB na extensão cab em 32 segundos.

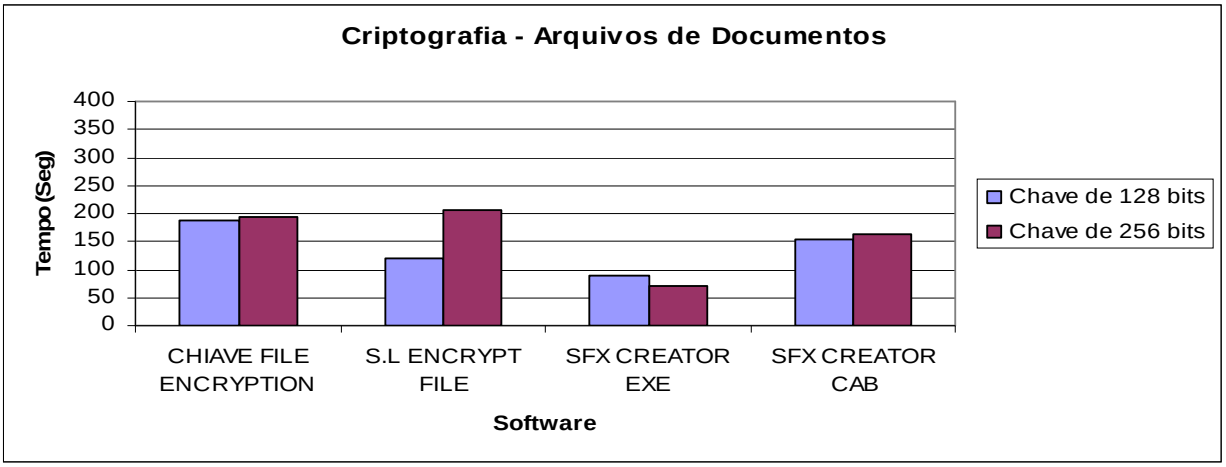

Gráfico 21 - Desempenho dos softwares na criptografia do arquivo de documentos

Enquanto os softwares *Chiave File Encryption e S.L Encrypt File* aumentaram o tempo de decriptografia em relação à criptografia, quando utilizaram a chave de 128 bits, o *SFX Creator* teve seu tempo melhorado como pode ser visto na comparação do Gráfico 21 com o Gráfico 22.

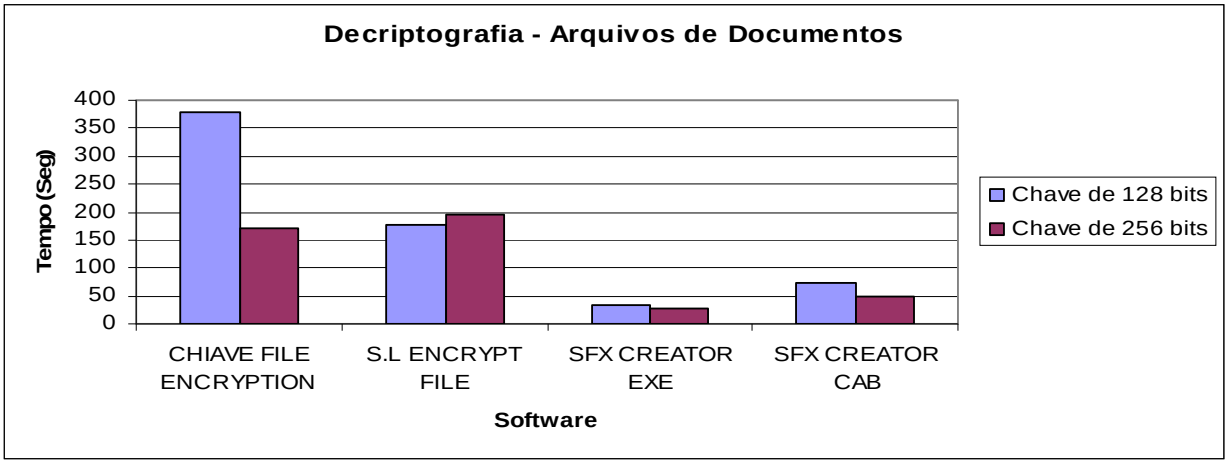

Gráfico 22 - Desempenho dos softwares na decriptografia do arquivo de documentos

No Gráfico 23 é avaliado o desempenho obtido pelos softwares quando testados com arquivos de filmes. Novamente o *SFX Creator* atingiu o melhor resultado, seguido pelo S.L Encrypt e o *Chiave File Encryption*.

Nota-se que a variação de tempo entre a utilização da chave de 128 e a chave de 256 bits foi baixa, sendo a maior variação encontrada pertencente ao software SFX Creator, cerca de 50 segundos.

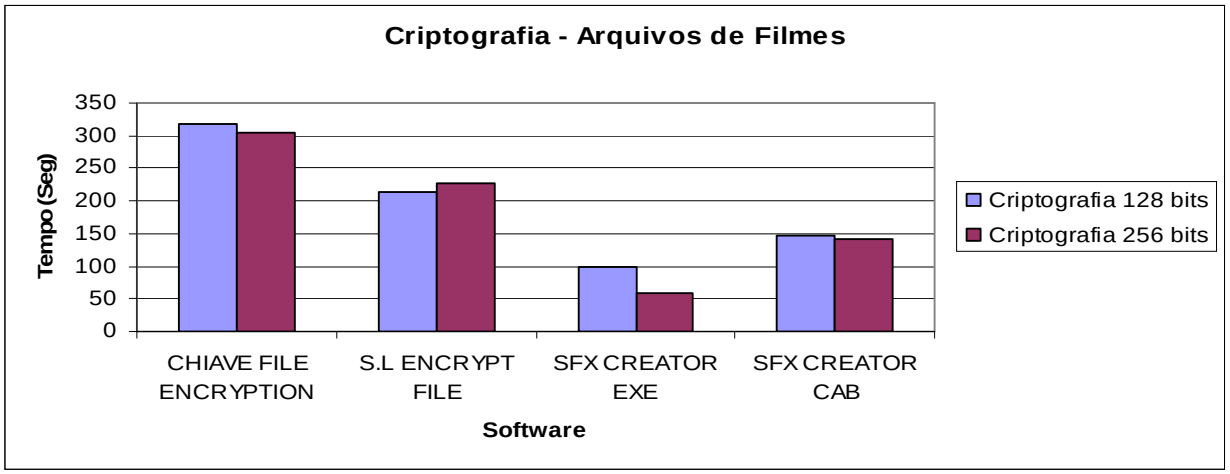

Gráfico 23 - Desempenho dos softwares na criptografia do arquivo de filmes

No processo de decriptografia o software *S.L Encrypt File* obteve o pior resultado levando 250 segundos para realizar a operação como mostra o gráfico 24. Na comparação realizada entre o tempo de criptografia e decriptografia, o software citado acima foi o único que teve o seu tempo acrescido.

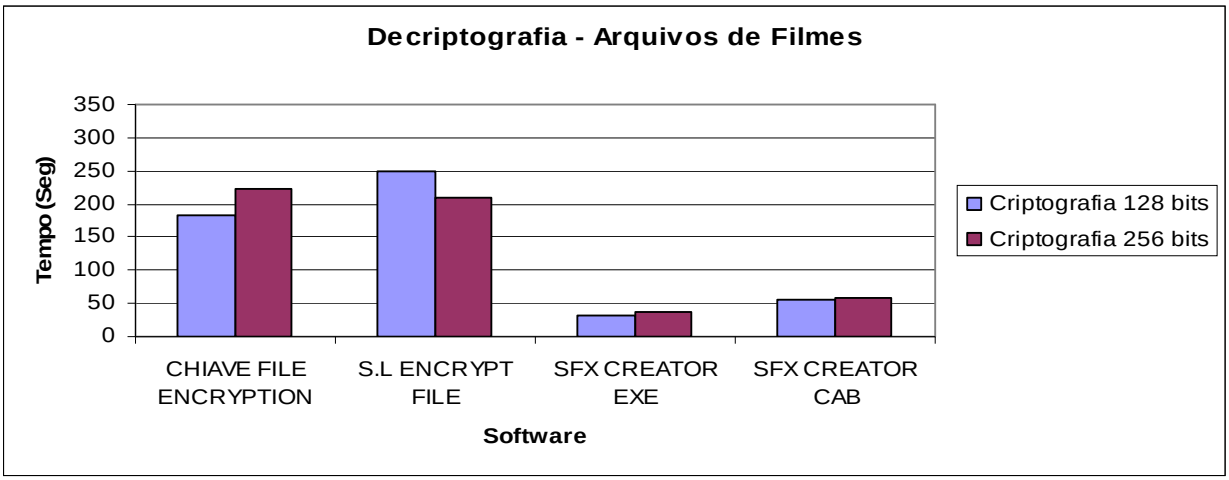

Gráfico 24 - Desempenho dos softwares na decriptografia do arquivo de filmes

O último arquivo a ser analisado é o todos que abrange os quatro tipos observados anteriormente: musica, imagem, documentos e filmes. Observa-se no Grafico 25 que *SFX Creator* como nos testes realizados em outros arquivos conseguiu o melhor desempenho.

O software *Chiave File Encryption* conseguiu superar o *S.L Encrypt file* quando criptografado com a chave de 128 bits, cerca de 300 segundos. Porém quando a chave foi aumentada seu tempo quase triplicou ficando em último lugar. Vale destacar que o arquivo todos foi o único arquivo onde o *Chiave File Encryption* perdeu rendimento significativo, alcançando a diferença de 1000 segundos.

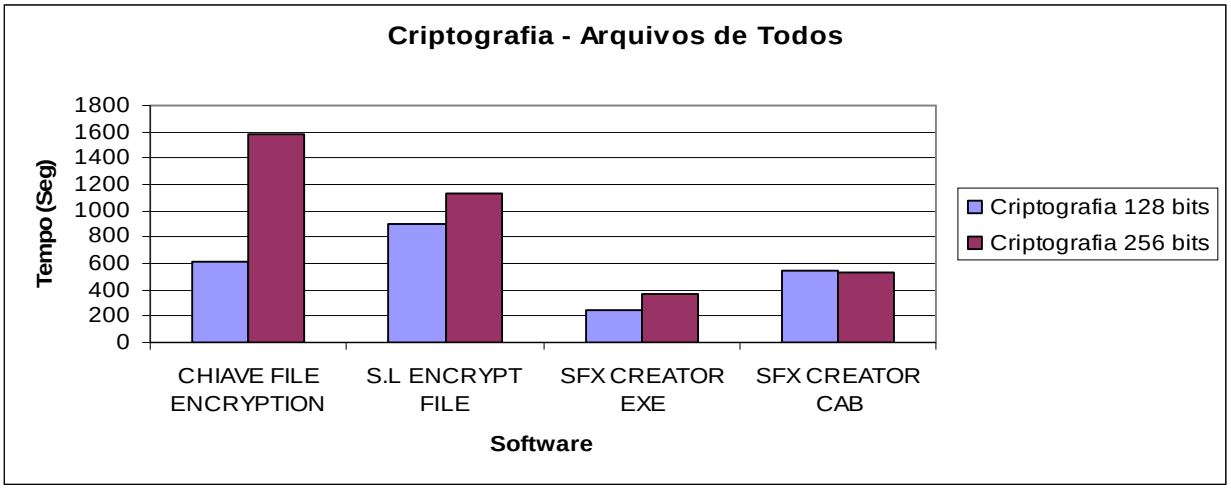

Gráfico 25 - Desempenho dos softwares na criptografia do arquivo de todos

No processo de decriptografia os tempos melhoraram em relação à criptografia como pode ser observado na comparação do gráfico acima com o gráfico 26, apenas o *S.L Encrypt File* quando utilizou a chave de 128 bits teve o seu tempo acrescido em 548 segundos. O Chiave File Encrypt continuou aumentando seu tempo em três vezes quando sua chave foi duplicada.

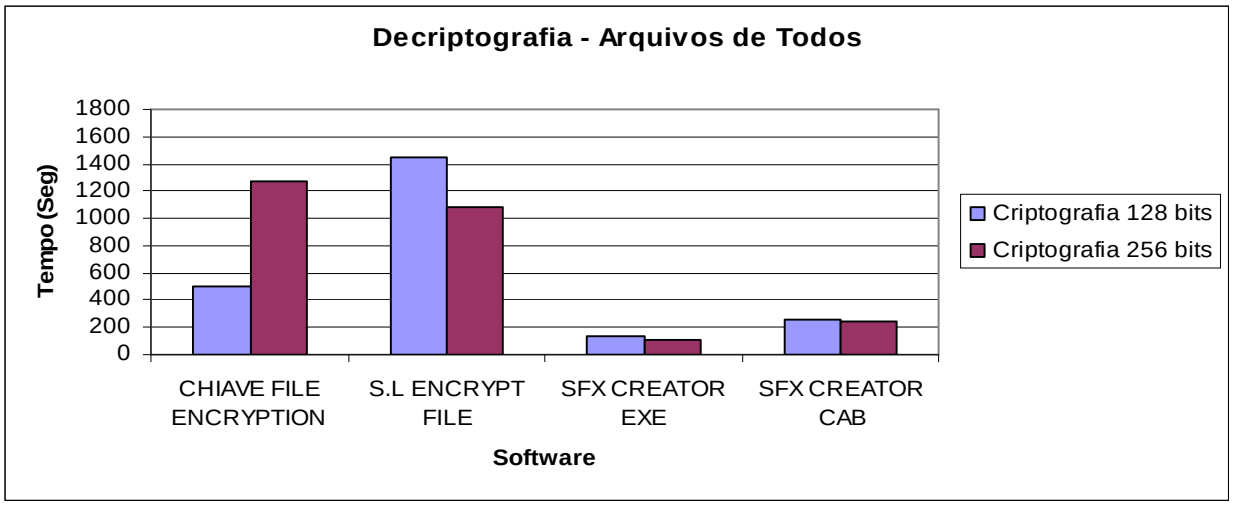

Gráfico 26 - Desempenho dos softwares na decriptografia do arquivo de todos

Nos testes referentes ao desempenho da CPU obtiveram os resultados parecidos, havendo uma pequena variação de 2% entre o maior e menor valor como pode ser observado nos resultados obtidos no gráfico 27.

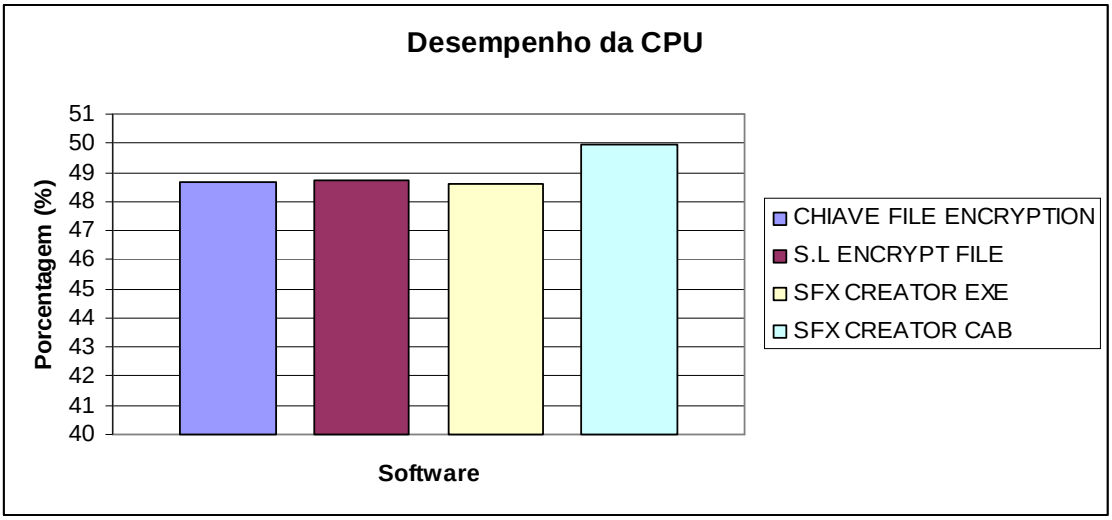

Gráfico 27 - Desempenho da CPU utilizadas pelos softwares

Já a memória utilizada sofreu uma variação maior e o Gráfico 28 mostra a média de memória utilizada por cada software. O *Chiave File Encryption* foi o que mais utilizou a memória cerca de 20 MB quase 7 vezes mais que o segundo colocado. Entretanto a memória utilizada por eles representam um baixo custo em relação das memórias disponíveis atualmente no mercado.

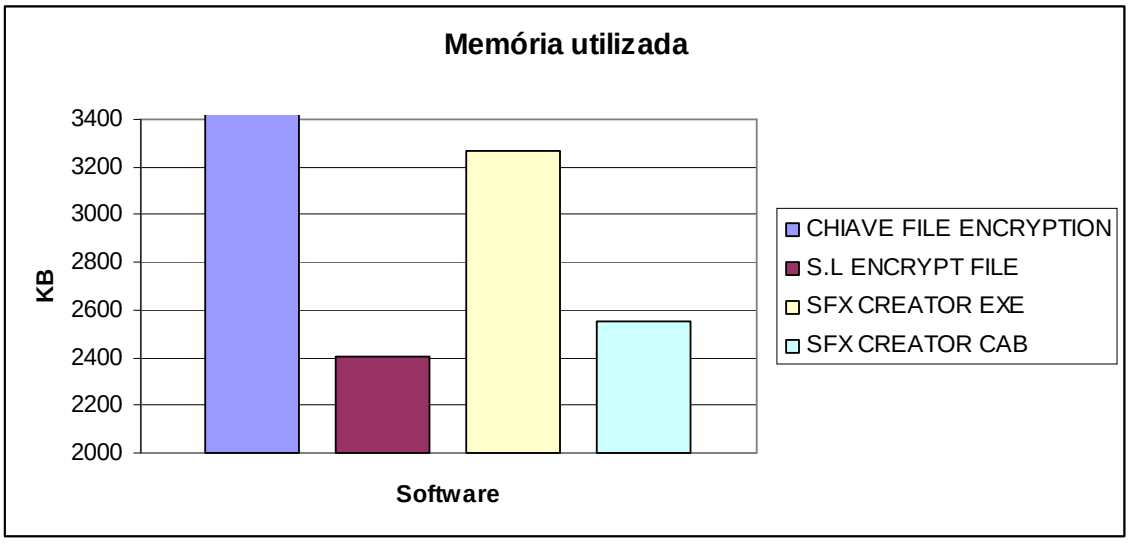

Gráfico 28 - Memória utilizada pelos softwares

7 Considerações finais

Para o levantamento teórico do presente trabalho foram estudadas as principais criptografias simétricas e assimétricas, bem como as cifras de substituição e transposição que são as bases para as novas criptografias, também foi apresentado de maneira sucinta os principais tipos de criptanálise que consiste em obter o texto original a partir do texto cifrado sem possuir a chave de criptografia utilizada.

Depois de realizado o levantamento bibliográfico, foram escolhidos três softwares freeware que possuíam como base a criptografia AES, criptografia padrão nos EUA, para a realização dos testes.

Os testes realizados por esses softwares utilizaram um único computador e aplicaram dois tamanhos de chaves: 128 e 256 bits em quatro tipos de arquivos: música, imagem, documentos e filme e por fim estes tipos de arquivos foram unidos transformando em um arquivo “todos”.

Com os testes concluídos, podemos verificar que o Software *SFX Creator* alcançou o melhor desempenho em todos os testes, conseguindo atingir o melhor tempo. O *S.L Encrypt File* conseguiu o segundo lugar superando o *Chiave File Encryption* em sete testes dos dez realizados quando utilizando a chave de 128 bits e em seis quando a chave foi dobrada.

O *SFX Creator* junto com o S.L Encrypt possui a desvantagem no momento da escolha dos arquivos, pois só é permitido a escolha de um arquivo para a realização tanto da criptografia como a decriptografia, necessitando assim fazer uma compactação ou outra forma para transformar estes arquivos em um.

Ao contrário do *Chiave File Encryption* que permite ao usuário escolher vários arquivos de uma única vez, além de oferecer a opção de apagar os arquivos originais após a criptografia, acrescentando assim segurança e praticidade para o utilizador.

Finalmente, como contribuição acadêmica, espera-se que este trabalho desperte o interesse de outros acadêmicos pelo tema criptografia e leve-os a dar continuidade aos testes realizados, escolhendo outras criptografias para que assim haja uma comparação com a criptografia AES.

REFERÊNCIAS

BENITS, Waldyr Dias. **Sistemas Criptográficos baseados em identidades pessoais**. São Paulo, 2003.Disponível em: http://www.ime.usp.br/dcc/posgrad/teses/benits.pdf . Acesso em: 15 mar 2011

BURNETT, Steve; PAINE, Stephen. **Criptografia e segurança: o guia oficial RSA**. 1ª ed. Rio de janeiro: Campus Ltda., 2002.

DOMINGUES, M.; HEUBEL, M.T.C.D.; ABEL, I.J.; **Base metodológica para o trabalho científico para alunos iniciantes.** Bauru, São Paulo: Edusc, 2003. 188p.

Fernandes, Jocimar. **Criptografia e o modelo criptográfico do sistema informatizado de eleições do Brasil.** Vitoria, 2007. Disponível em: < http://www.bou.com.br/arquivos/monografias/TccRedesJF.pdf>. Acesso em: 03 maio 2011

FLORIANO, Guilerme Martinez. **Camouflaged security system: protótipo de software que emprega técnicas de criptografia, assinatura digital e esteganografia paracomunicacao segura**. Porto Alegre, 2007. Disponível em: < http://www.uniritter.edu.br/graduacao/informatica/sistemas/downloads/tcc2k7-2/GuilhermeMartinezFloriano.pdf>. Acesso em: 03 mar 2011

GARFINKEL, Simson; SPAFFORD, Gene. **Comércio & Segurança na Web – Riscos, Tecnologias e Estratégias**. São Paulo: Market Press, 1999

HINZ, M. A. M. Um **estudo descritivo de novos algoritmos de criptografia**. Pelotas, 2000. Disponível em: < http://www.ufpel.edu.br/prg/sisbi/bibct/acervo/info/2000/Mono-MarcoAntonio.pdf>. Acesso em: 01 mar 2011

KUROSE, James F.; ROSS, Keith W. **Redes de computadores e a Internet: uma abordagem top-down**. 3ª ed. São Paulo: Pearson Addison Wesley, 2006

MENDES, Aliane V. **Estudo de Criptografia com chave pública baseada em curvas elípticas.** Montes Carlos, 2007. Disponível em: < http://www.ccet.unimontes.br/arquivos/monografias/261.pdf >. Acessado em: 15 mar 2011

SCHNEIER, Bruce. **Applied Cryptography: protocols, algorithms, and source code in C**. 2ª ed. New Jersey: Wiley, 1996

SCHNEIER, Bruce. **Pratical Cryptography**. 1ª ed. New Jersey: Wiley, 2003

STALLINGS, William. **Criptografia e segurança de redes**. 4ª ed. São Paulo: Pearson Prentice Hall, 2008.

SOUZA, R. A.; OLIVEIRA, F.B O **padrão de Criptografia simétrica AES**. Disponível em:<http://www.lncc.br/~borges/doc/O%20padr%E3o%20de%20criptografia%20sim%E9trica%20AES.pdf>. Acessado em: 04 out 2011

TANENBAUM, Andrew S. **Redes de computadores**. 4ª ed. Rio de Janeiro: Elsevier, 2003

TERADA, Routo. **Segurança de dados**. ed: São Paulo : Edgard Blucher, 2000.

TKOTZ, Vicktoria. **Criptografia – Segredos embalados para viagem**. 1ª ed: São Paulo: Novatec, 2005.