UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ELETRÔNICA

CONSTRUÇÃO DE UM AMBIENTE WEB COM FERRAMENTAS PARA ESTUDO DE ALGORITMOS DE CRIPTOGRAFIA ATRAVÉS DO MATLAB

Autora: _____
Rosane França de Moraes

Orientador: _____
Luis Felipe M. de Moraes, Ph.D.

Examinador: _____
Aloysio de Castro P. Pedroza, Dr.

Examinador: _____
Antônio Cláudio Gómez de Sousa, M.Sc.

DEL
Junho de 2004

Agradecimentos

Em primeiro lugar agradeço à FAPERJ pela bolsa de iniciação científica a mim concedida a qual me auxiliou muito durante a execução deste trabalho.

Agradeço aos professores Antônio Cláudio Gómez de Sousa e Aloysio de Castro P. Pedrosa pela participação na banca examinadora. E ao meu orientador Luis Felipe M. de Moraes.

Gostaria de agradecer também à Alessandro Martins por toda a ajuda e apoio que me deu durante todas as fases de desenvolvimento do meu projeto, até o último minuto! Obrigada!

Resumo

Com a crescente utilização dos computadores a nível residencial, das redes de computadores nas empresas e especialmente da Internet, a segurança tornou-se um requisito essencial.

Apesar de uma grande parte das informações transferidas via Internet ser de acesso público, existe um conjunto de operações que requerem algum nível de segurança a ser acertado entre as partes, como por exemplo transações com número de cartões de crédito ou dados de contas bancárias, além de acesso a informações privadas usando a Internet.

Esses aspectos da segurança podem ser obtidos através de procedimentos matemáticos, genericamente conhecidos como algoritmos de criptografia.

Este trabalho visa Implementar uma caixa de ferramentas para o Matlab de fácil manuseio, em linguagem interpretada e não compilada, onde será possível além de aprender sobre cada algoritmo, interagir com as funções que serão implementadas possibilitando um alto grau de ajuste para cada caso de uso.

O projeto está concentrado na implementação de algoritmos de criptografia simétrica e assimétrica.Os algoritmos simétricos implementados foram: DES, Triple-DES, Blowfish, RC5 e IDEA, já os algoritmos assimétricos foram: RSA e El-Gamal.

Criptografia simétrica: requer que o transmissor e o receptor compartilhem uma mesma chave. Essa informação secreta deve ser usada para cifrar e decifrar as mensagens.

Criptografia assimétrica: também conhecida como criptografia de chave pública. São empregados dois algoritmos que usam cada um uma chave diferente, uma chamada pública e a outra privada.

Palavras-chave

Criptografia simétrica

Criptografia assimétrica

Sistema para Web

Matlab

Algoritmos de criptografia

Indice

Indice de figuras

Glossário

A

ALFABETOS - conjunto de símbolos utilizados nos textos planos ou nos criptogramas. Os símbolos utilizados nos textos planos e nos criptogramas não têm que ser os mesmos. Denotaremos como SM ao alfabeto utilizado nos textos planos e SC ao alfabeto utilizado nos criptogramas.

ALGORITMO - Conjunto de operações elementares que devem ser efetuadas para se obter um resultado desejado. Por exemplo, uma receita de bolo é um algoritmo.

ALGORITMO ASSIMÉTRICO - Veja "Assimétrico".

ALGORITMO DE CHAVE PÚBLICA - Veja "Assimétrico".

ALGORITMO DE CHAVE SECRETA - Veja "Simétrico".

ALGORITMO DE HASH SEGURO - Algoritmo que cria a partir da mensagem original, uma assinatura digital que garante a autenticidade da mensagem.

ALGORITMO SIMÉTRICO - Veja "Simétrico".

ASCII (American Standard Code for Information Interchange) - Código Padrão Americano para o Intercâmbio de Informação que traduz os nomes dos caracteres de um alfabeto para outros. Por exemplo, a letra "A" é traduzida para 65.

ASSIMÉTRICO - Um algoritmo de criptografia que utiliza uma chave pública para encriptar e uma chave privada (diferente) para decifrar as mensagens. Os algoritmos assimétricos são

capazes de muitas operações, incluindo criptografia, assinaturas digitais e acordo de chave. Também conhecido como algoritmo de chave pública..

ASSINATURA - Associada a uma mensagem, prova a identidade do remetente.

ATAQUE - 1. Tentativa de criptoanálise. 2. Ato de tentar desviar dos controles de segurança de um sistema. Um ataque pode ser ativo, tendo por resultado a alteração dos dados; ou passivo, tendo por resultado a liberação dos dados. Nota: O fato de um ataque estar acontecendo não significa necessariamente que ele terá sucesso. O nível de sucesso depende da vulnerabilidade do sistema ou da atividade e da eficácia de contramedidas existentes

AUTENTICAÇÃO - Verificação reivindicada de uma identidade. O processo de determinar a identidade de um usuário que esteja tentando alcançar um sistema

AUTENTICAR - 1. Verificação da identidade de um usuário, de dispositivo, ou de outra entidade em um sistema computatorizado, frequentemente como um pré-requisito a permitir o acesso aos recursos em um sistema. 2. Se assegurar da identidade do remetente de uma mensagem e da integridade da mensagem recebida.

AUTORIDADE DE PROTEÇÃO DE DADOS - É uma autoridade de supervisão interna responsável pela monitoração e implementação da Política de Segurança.

B

BIGRAMA - Sequência de duas letras consecutivas. Exemplo: pa, le,...

C

CHAVE - Num sistema de encriptação, corresponde a um nome, uma palavra, uma frase, etc, que permite, mediante o algoritmo de encriptação, cifrar ou decifrar uma mensagem.

CHAVE DUPLA - Cifra de chave dupla. Outro nome para cifra polialfabética.

CHAVE FRACA - Chave que, por uma razão qualquer (seu comprimento, uma propriedade matemática, etc), permite quebrar rapidamente o código.

CHAVE PRIVADA - Chave que deve ser mantida secreta, (ver Assimétrico).

CHAVE PÚBLICA - Uma chave criptográfica disponível para distribuição sem necessidade de segredo. É o oposto de uma chave privada ou chave secreta. Veja "Assimétrico".

CHECKSUM - Um valor calculado a partir de parte de dados que pode ser usado para verificar que o dado não foi alterado.

CIFRA - Conjunto de procedimentos e conjunto de símbolos (letras, nomes, sinais, etc) usados para substituir as letras de uma mensagem para encriptá-la. É geralmente classificada como cifra de transposição e cifra de substituição.

CIFRAGEM - Cifrar ou cifragem. Procedimento pelo qual se torna impossível a compreensão de um documento a qualquer pessoa que não possua a chave da cifra.

CIFRANTE - O mesmo que chave.

CIFRAR - O mesmo que fazer uma cifragem.

CODIFICAR - Modificar a estrutura de um conjunto de documentos aplicando um algoritmo (cifra, método de compressão, etc).

CÓDIGO - Sistema de símbolos (palavras, nomes, símbolos, etc) que substituem palavras inteiras. Por exemplo, a substituição de "007" por "James Bond".

COMUNICAÇÕES SEGURAS - Assegura a autenticidade das telecomunicações através de medidas tomadas para negar à pessoas desautorizadas o acesso a estas informações.

CONFIDENCIALIDADE - Propriedade de certas informações que não podem ser disponibilizadas ou divulgadas sem autorização para pessoas, entidades ou processos. O conceito de garantir a informação sensível confidencial, limitada para um grupo apropriado de pessoas ou organizações. Assegurar a confidencialidade de documentos é assegurar que apenas pessoas autorizadas tenham acesso à informação.

CRIPTOANÁLISE - Criptoanálise ou criptanálise. 1. Métodos de analisar mensagens cifradas com o objetivo de decifrá-las. 2. Arte ou ciência de determinar a chave ou decifrar mensagens sem conhecer a chave. Uma tentativa de criptoanálise é chamada ataque.

CRIPTOGRAFIA - 1. Disciplina de criptologia que trata dos princípios, dos meios e dos métodos de transformação de documentos com o objetivo de mascarar seu conteúdo, impedir modificações e o uso ilegal dos mesmos. 2. Ciência que estuda os princípios, meios e métodos para tornar ininteligíveis as informações, através de um processo de cifragem, e para restaurar informações cifradas para sua forma original, inteligível, através de um processo de decifragem. A criptografia também se preocupa com as técnicas de criptoanálise, que dizem respeito a formas de recuperar aquela informação sem se ter os parâmetros completos para a decifragem.

CRIPTOGRAMA - Mensagem cifrada ou codificada.

CRIPTOLOGIA - Ciência das mensagens secretas. É composta pelas disciplinas de criptografia e de criptanálise.

CRIPTOSISTEMA - Cifra.

D

DECIFRAR - Operação inversa de cifrar, ou seja, obter a versão original de uma mensagem cifrada. Ao contrário da desencriptação, aqui se conhece o método de cifragem.

DES - Algoritmo de criptografia simétrico com chave de 56 bits. Existe também uma variação chamada 3DES ou triplo DES, em que se usa três vezes a chave de 56 bits. Mesmo resultando

em uma chave de 168 bits, um tipo de ataque chamado "meet in the middle" pode quebrar um triplo DES com o mesmo esforço que seria necessário para quebrar um algoritmo de 112 bits.

DESAFIO/RESPOSTA - Uma técnica de autenticação na qual um servidor emite um desafio desconhecido ao usuário, que computa uma resposta usando algum processo do token de autenticação.

DECRIPTAR - Restaurar documentos cifrados, restaurando-os ao estado original, sem dispor das chaves teoricamente necessárias.

DICIONÁRIO - Lista de palavras e expressões mais utilizadas que servem de base para se procurar uma senha.

DIFFIE-HELLMAN - Um algoritmo assimétrico que permite um acordo de chaves: as duas partes trocam suas chaves públicas e as usam em conjunto com suas chaves privadas para gerar uma terceira chave secreta compartilhada. Um curioso que veja as chaves públicas mas não tenha o acesso à chave confidencial de um ou de outro não pode descobrir a terceira chave compartilhada.

DIGRAMA - O mesmo que bigrama.

DSA - Algoritmo de assinatura digital é um algoritmo assimétrico que permite criar assinaturas digitais.

DSS (Digital Signature Standard) - Padrão do governo dos E.U.A. que combina DSA e SHA-1 para especificar um formato para assinatura digital.

E

ENCRIPTAÇÃO - Um processo de disfarçar a informação de modo que não possa ser compreendida por uma pessoa desautorizada. A transformação de uma seqüência de caracteres em outra por meio de uma cifra, de uma tabela de transposição, ou de um algoritmo a fim fazer

com que a informação não seja entendida a qualquer um que não possua o mecanismo da descodificação.

ENIGMA - Máquina Enigma, utilizada durante a Segunda Guerra Mundial.

ESTEGANOGRAFIA - Do grego "escrita escondida". Ramo particular da criptologia que consiste, não em fazer com que uma mensagem seja ininteligível, mas em camuflá-la, mascarando a sua presença. Ao contrário da criptografia, que procura esconder a informação da mensagem, a esteganografia procura esconder a existência da mensagem.

F

FORÇA BRUTA - É um ataque que consiste em testar todas as chaves possíveis até encontrar a correta. Não é um bom método de acesso porque pode demorar dias, meses, ou até anos.

FREQUÊNCIA - Porcentagem de ocorrência de uma letra ou palavra em uma determinada língua. Calcular a frequência de ocorrência é uma das primeiras etapas de um processo de desencriptação.

H

HACKER - Pessoa que tenta acessar sistemas sem autorização, usando técnicas próprias ou não, no intuito de ter acesso a determinado ambiente para proveito próprio ou de terceiros. Dependendo dos objetivos da ação, podem ser chamados de Cracker, Lammer ou BlackHat.

HACKING - Uma tentativa desautorizada em alcançar uma base de dados de um sistema. Usado frequentemente na referência a uma pessoa que tente entrar numa base de dados de um sistema, de uma posição remota passando pela rede controlada.

HOMOFÔNICA - Do grego "o mesmo som". O conceito de ter sequências diferentes de letras que sejam pronunciadas do mesmo modo. Em criptografia, uma cifra que traduz um único

símbolo do texto plano em qualquer um de múltiplos símbolos do texto cifrado, todos com o mesmo significado. Veja também polifônica, poligrâmica e monogrâmica.

I

IDEA - (International Data Encryption Algorithm)

INTEGRIDADE - 1. A condição na qual a informação ou os recursos da informação são protegidos contra modificações não autorizadas. 2. Garantia oferecida ao usuário de que a informação correta, original, não foi alterada, nem intencionalmente, nem acidentalmente.

M

MARCA D'ÁGUA - Aplicação especial da esteganografia que consiste em camuflar informações a respeito da origem do documento (nome do autor, data, copyright, ect) numa imagem adicionada ao próprio documento.

MD5 - Algoritmo seguro de hash criado por Ron Rivest. Message Digest Algorithm. Veja "Algoritmo seguro de hash".

MENSAGEM CLARA - Mensagem clara ou mensagem original. Também denominado texto plano.

MENSAGEM ORIGINAL - Mensagem com o texto original, sem ter sofrido qualquer alteração de métodos criptográficos.

MONOALFABÉTICA - Substituição usando um único alfabeto. Também chamada de substituição simples.

MONOGRÁFICA - O mesmo que monogrâmica.

MONOGRÂMICA - Monogrâmica ou monográfica, do grego "uma letra". Uma cifra que traduz um a um os símbolos do texto original em texto cifrado. O oposto de poligrâmico; veja também homofônica e polifônica.

N

NÃO REPÚDIO - Não poder negar a autenticidde de um documento, a sua assinatura ou o seu envio, Por exemplo, emissor não pode negar, após o destinatário ter recebido uma mensagem, que tal não foi enviada.

NULL - Veja "Padding".

P

PADDING - Caracteres sem significado adicionados a uma mensagem por certos algoritmos. São utilizados para obter um comprimento constante para uma mensagem. São caracteres nulos.

PASSWORD - Veja "Senha".

PGP (Pretty Good Privacy) - Algoritmo de cifragem informatizada desenvolvido por Phil Zimmermann e baseado no RSA.

PKCS The Public Key Cryptography Standards - Conjunto de especificações criadas pela RSA para padronizar os formatos e operações de criptografia.

PKCS#1 RSA Encryption Standard - Especificação de padrão de dados para o protocolo RSA, incluindo o padrão para criptografia e assinatura digital RSA e padrão para estocagem de chaves públicas e privadas.

PKCS#5 Password-Based Encryption Standard - Especificação de um padrão para proteção de dados para ser usar a criptografia baseada em senha com o DES.

PKCS#8 Private-Key Information Syntax Standard - Especificação de um padrão para estocagem de chaves privadas, incluindo a vantagem de criptografá-las com PKCS#5.

PKCS#10 Certification Request Syntax Standard - Especificação de um padrão para codificar requisições de certificados, incluindo o nome da pessoa que requisita o certificado e sua chave pública.

POLIALFABÉTICA - Um tipo de substituição na qual múltiplos alfabetos de substituição distintos são usados.

POLIFÔNICA - Do grego "múltiplos sons". O conceito de ter uma sequência de letras que é pronunciada de formas diferentes e distintas, dependendo do contexto. Em criptografia, uma cifra que usa um símbolo único do texto cifrado para representar múltiplos símbolos diferentes do texto plano. Veja também homofônica, poligrâmica e monogrâmica.

POLIGRÂMICA - Poligrâmica ou poligráfica, do grego "múltiplas letras". Uma cifra que traduz vários símbolos do texto original, em grupo e ao mesmo tempo, em texto cifrado. Exemplos: a cifra de Playfair e a cifra de Hill. O oposto de monogrâmico; veja também homofônica e polifônica.

R

RC4 - Algoritmo simétrico desenvolvido por Ron Rivest que pode usar chaves de tamanho variável. Usualmente usado com 40 bits ou 128 bits.

RECIFRAGEM - Fazer nova cifragem à partir de uma mensagem que já tenha sido cifrada por outro método. Geralmente, a encriptação de nível mais alto (ou mais externo) de uma encriptação múltipla. Classicamente, recifragens são muito fracas, dependendo do efeito randômico do nível de encriptação anterior. Também conhecida como superencriptação ou supercifragem.

REPERTÓRIO - Tabela contendo códigos com seus respectivos significados. Por exemplo, você encontra na Internet o código 404, que significa página inexistente.

REPÚDIO - Veja "Não Repúdio".

REVERSA - falta

RSA - Algoritmo de cifragem por chave pública utilizado principalmente no PGP, usado principalmente na cifragem da assinatura, permitindo a identificação do documento. Permite criptografar dados, criar e verificar assinaturas digitais.

S

SCYTALE - Um bastão de madeira ao redor do qual se enrolava uma tira de couro ou papiro. (Vide História da Criptografia)

SENHA - Uma única palavra ou seqüência de caracteres usada para autenticar uma identidade. A senha é confidencial, opostamente a identificação do usuário.

SIGILO - Somente os usuários autorizados têm acesso à informação.

SIMÉTRICO - Algoritmo de criptografia que usa somente uma chave, tanto para criptografar como para descriptografar. Esta chave deve ser mantida secreta para garantir a confidencialidade da mensagem. Também conhecido como algoritmo de chave secreta.

SSL Secure Sockets Layer - Protocolo que possibilita realizar comunicações seguras através de criptografia e autenticação.

SUBSTITUIÇÃO - Uma cifra de substituição troca os caracteres de uma mensagem original por símbolos (caracteres, nomes, sinais, etc) predefinidos.

SUPERCIFRAGEM - Veja "Recifragem".

SUPERENCRIPTAÇÃO - Veja "Recifragem".

T

TEXTO CIFRADO - "Veja Criptograma".

TEXTO PLANO - Texto que não foi criptografado e pode ser lido com facilidade.

TOMOGRÂMICA - Os sistemas tomogrâmicos são aqueles nos quais cada letra é representada por um grupo de duas ou mais letras ou cifras. Estas letras ou cifras são obtidas através de uma cifragem por substituição ou por transposição separada.

TRANSPOSIÇÃO - Uma cifra de transposição não modifica o conteúdo da mensagem. Os caracteres permanecem os mesmos, porém são embaralhados através de um método predefinido.

TRIGRAMA - Sequência de três letras consecutivas. Exemplo: ong, tio, ...

W

WATERMARKING - Veja "Marca d'água".

1 Motivação

1.1 Introdução

Reza a Declaração Universal dos Direitos Humanos, art. 19:

"Todo o homem tem direito à liberdade de opinião e expressão; este direito inclui a liberdade de, sem interferências, ter opiniões e de procurar, receber e transmitir informações e idéias por quaisquer meios, independentemente de fronteiras"

No entanto, o direito à livre expressão de opiniões e repasse ou guarda de informações, com a evolução dos meios de comunicação e com o advento da tecnologia da informação, vem sendo ameaçado por mecanismos cada vez mais sofisticados de "invasão de privacidade".

Com o crescente uso das redes de computadores e a massificação do uso da Internet, surgiu a necessidade de se utilizar melhores mecanismos para prover a segurança das transações de informações confidenciais.

Vários serviços financeiros como pagamentos de conta, corretagem, seguros e Home Banking estão ou estarão disponíveis em larga escala na Internet. Assim, a questão da segurança é bastante enfatizada, principalmente, quando imagina-se a possibilidade de se ter suas informações expostas a atacantes ou intrusos da Internet, que surgem com meios cada vez mais sofisticados para violar a privacidade e a segurança das comunicações.

Devido a estas preocupações, a proteção da informação tem se tornado um dos maiores interesses dos administradores de sistemas.

A criptografia ajuda a imputar responsabilidade, promover a justiça, prover certeza e privacidade. Pode prevenir fraudes em comércio eletrônico e garantir a validade de transações financeiras. Se usada apropriadamente, protege a anonimato e fornece provas de identidade de pessoas. Pode, ainda, impedir vândalos de alterarem sua página na Internet e competidores

industriais de lerem seus documentos confidenciais. Com o comércio seguindo sua marcha pelas redes de computadores, a Criptografia se tornará cada vez mais vital.

A principal motivação de sistemas de criptografia é proporcionar segurança a todos usuários e evitar que a transação possa ser decifrada por pessoas não autorizadas, especialmente transações bancárias e de compras.

1.2 Necessidades e utilização

A Internet está alterando a maneira pela qual nos comunicamos e pagamos por serviços, acessamos as informações, pagamos e adquirimos mercadorias. Vários serviços financeiros como pagamentos de conta, corretagem, seguros, e home banking estão ou estarão disponíveis em larga escala na Internet.

Para que estes serviços possam se dar com confiabilidade e segurança, certas necessidades têm que ser supridas:

a) Identidade dos participantes, após o que cada um terá acesso ou não a certos recursos computacionais;

b) confidencialidade, garantindo que terceiros não tenham acesso aos dados em tráfego;

c) integridade, permitindo detectar quando houve alguma alteração nos dados trocados;

d) unicidade de transação, que impeça a replicação indevida;

e) autoria de transação impedindo a qualquer momento o repúdio.

Os engenheiros de software têm desenvolvido meios para enviar informaçoes confidenciais de forma segura. As informaçoes precisam ser encriptadas - isto é, alteradas de modo que para uma pessoa que não o receptor pretendido pareçam totalmente deturpadas. E devem poder ser decifradas - isto é, retornadas à mensagem original pelo receptor, e somente pelo receptor. Muitos sistemas complexos foram criados para permitir esse tipo de encriptação e decriptação e são chamados de sistemas de criptografia.

A Criptografia apresenta-se como uma ferramenta de grande utilidade para uma série de aplicações. Dentre estas aplicações incluem segurança de comunicações, identificação e autenticação. Outras aplicações envolvem sistemas para comércio eletrônico, certificação, correio eletrônico seguro, etc.

- **Segurança de Comunicações**

As aplicações que envolvem segurança de comunicações são as que mais demandam o uso da criptografia. Duas pessoas podem se comunicar de forma segura encriptando as mensagens trocadas entre elas. Isto pode ser feito de forma que uma terceira pessoa que esteja interceptando estas mensagens nunca possa ser capaz de decifrá-las.

- **Identificação e Autenticação**

Identificação e Autenticação são as mais vastas aplicações da criptografia. Identificação é o processo de verificação da identidade de alguém ou de alguma coisa. Por exemplo, quando se retira dinheiro em um banco o caixa pede para que a pessoa se identifique para verificar a identidade do proprietário da conta. O mesmo processo pode ser feito de forma eletrônica com o uso da criptografia. Todos os cartões de terminais automáticos são associados a uma senha a qual vincula o proprietário do cartão ao proprietário da conta.

Quando o cartão é inserido em um terminal, a máquina pede a quem tem este cartão a senha. Caso esta senha esteja correta, a máquina infere que aquela pessoa seja o proprietário da conta e libera o acesso. Uma outra aplicação importante da criptografia é a Autenticação. A autenticação é similar à identificação, uma vez que ambos os processos permitem a uma entidade o acesso a determinados recursos. Porém a autenticação é mais abrangente dado que ela não envolve necessariamente a identificação da pessoa ou entidade. A autenticação determina simplesmente se dada pessoa ou entidade é autorizada para aquilo em questão.

- **Comércio Eletrônico**

Ao longo dos últimos anos têm havido um crescimento do número de negócios conduzidos via Internet. Esta forma de negócio é conhecido como Comércio Eletrônico ou E-Commerce. O comércio eletrônico envolve uma série de atividades realizadas de forma

eletrônica dentre as quais se destacam as transferências de fundos que também são realizadas desta forma.

Entretanto a simples apresentação de um número de cartão de crédito pode levar o seu proprietário a ser fraudado tendo o seu número de cartão de crédito usado sem sua permissão. O uso de mecanismos de transação segura na Internet, onde o número do cartão de crédito é enviado junto com outras informações de forma encriptada tem permitido que estes pagamentos possam se dar de forma segura.

- **Certificação**

Uma outra aplicação da criptografia é a Certificação. É grande o crescimento do PKI (Public Key Infrastructure) ou (Infraestrutura de chaves públicas), onde todos podem criptografar informações e assinar digitalmente as mesmas, garantindo assim a integridade, confidencialidade e autenticidade. O PKI ou ICP funciona através da certificação digital, onde são criados os cartórios (Autoridades Certificadora) também conhecidos como CA - Certificate Authority, como a Verisign, Certisign, Thawte, Entrust e etc. Estes cartórios por sua vez emitem certificados digitais que servem para comprovar a identidade, confidencialidade e fazer autenticação de uma pessoa ou empresa no mundo digital. Suponha que um hacker envie um e-mail para você se passando pela Microsoft, dizendo que o arquivo em anexo é um patch de segurança, no entanto quem garante que realmente ele é quem está dizendo ser ? Se a mensagem não está assinada digitalmente quer dizer que qualquer pessoa poderia tê-la enviado.

O Brasil através do Comitê Gestor de Segurança, órgão responsável pela segurança da informação do Governo Federal tornou-se uma Autoridade Certificadora - CA e passará a emitir e assinar digitalmente certificados para diversos tipos de documentos como CNDs, Procurações, Decretos e outros. Estes certificados terão valor jurídico perante a constituição, isto leva a pensar que em breve os cartórios existentes onde reconhecemos firma e fazemos nossas autenticações autenticarão documentos assinados digitalmente. Em um futuro breve os estados se tornarão CA - subordindados e depois os municípios e então todos nós teremos um certificado digital em vez de CPF e Identidade.

A Criptografia, como é sabido, não resolve todos os problemas de segurança, mas boa parte sim e ela tende a estar cada vez mais presente no nosso dia-a-dia, seja em um caixa

eletrônico ou em uma maquineta de débito automático quando pagamos um abastecimento com um cheque eletrônico ou até mesmo aquele livro que compramos via internet.

1.3 O que é Matlab?

MATLAB é um ambiente de desenvolvimento com uma linguagem intuitiva e um ambiente computacional técnico de fácil utilização.

Fornece um conjunto de ferramentas (toolbox) que contém diversas funções matemáticas já implementadas, como multiplicação e inversão de matrizes e desenho de gráficos, o que torna o trabalho de programação bem mais simples do que em C, C++ ou Java.

1.4 Por que utilizar o Matlab?

Com base nas características acima citadas podemos constatar que o ambiente MATLAB acelera o desenvolvimento e comprime o tempo gasto na programação reduzindo assim os custos do projeto.

Ele consegue integrar um ambiente de computação matemática, uma linguagem intuitiva e uma alta taxa de reutilização de código, uma vez que diversas funções já estão implementadas na forma de toolbox (conjunto de ferramentas). Com isso pode-se dar mais enfoque no projeto e validação do algoritmo, minimizando a perda de tempo com desenvolvimento de funções para manipulação de dados, matrizes, criação de gráficos, etc.

No entanto, essas facilidades têm um preço, o desempenho! O processo de criptografia utilizando o MATLAB torna-se extremamente lento, muito mais lento do que os resultados obtidos utilizando-se bibliotecas compiladas como a cryptlib por exemplo.

Como o foco deste trabalho é didático, a desvantagem obtida com o baixo desempenho é superada pela vantagem obtida com a facilidade de entendimento e alteração dos scripts por usuáios com poucos conhecimentos em programação. O que justifica o seu uso neste trabalho.

No apêndice B podemos encontrar outros exemplos de trabalhos e simulações implementados em MATLAB.

1.5 Como conseguir uma cópia do Matlab?

A MathWorks disponibiliza uma versão para estudantes que pode adquirida através da loja virtual da MathWorks no site www.mathworks.com.

Existe um livro em português que traz a versão 5 do Matlab. Esta publicação é da Makron Books e se chama “Matlab 5 versão estudante”.

2 Introdução

Desde que as sociedades humanas estruturam-se tem havido a necessidade de se ocultar informações entendidas, cada uma a seu tempo, como segredos. Sejam segredos familiares, segredos sentimentais, segredos pessoais, segredos religiosos, ou segredos militares ou governamentais.

Tão forte quanto a necessidade de guardar estes segredos é o desejo de outros de desvendar esses mesmos segredos. Seja por dinheiro, poder, vingança, curiosidade, arrogância, ou qualquer outro sentimento. Essa tem sido uma batalha que, ao longo dos anos vem sendo travada entre aqueles que querem guardar segredos e os que querem desvendar esses segredos.

O objetivo deste trabalho é descrever uma das maneiras de se evitar o acesso indevido a informações confidenciais através da codificação ou cifragem da informação, conhecida como criptografia, que faz com que apenas as pessoas às quais estas informações são destinadas, consigam compreendê-las.

2.1 O que é criptografia?

A palavra Criptologia deriva da palavra grega κρυπτος (oculto) e logos (estudo). Este campo de estudo mais abrangente abarca as disciplinas da Criptografia e da Criptoanálise combinadas.

Um conceito que possa definir Criptologia em poucas palavras é que ela seria o estudo das escritas secretas.

Na verdade Criptologia é o estudo de Códigos e Cifras (não necessariamente secretos). Mensagens ocultas que não são nem codificadas nem cifradas são, simplesmente, ocultas. A técnica da tinta invisível é um exemplo de mensagem oculta.

Um código é um sistema pré-estabelecido de substituição de palavras ou de parágrafos. Um idioma estrangeiro, por exemplo, é como um código secreto onde cada palavra em português possui uma equivalente nele. Assim, "ôi" em português equivale a "hola" em espanhol ou "hi" em inglês. A maioria dos códigos funcionam com um "livro de códigos" onde estão relacionadas as equivalências, como se fosse um dicionário.

Já a palavra cifra vem do hebraico saphar, que significa "dar número". A maioria das cifrações são intrinsecamente sistemáticas, freqüentemente baseadas em técnicas de sistemas numéricos.

O termo Criptoanálise é o estudo de como "quebrar" os mecanismos criptográficos, podendo assim revelar o conteúdo das mensagens cifradas.

A palavra criptografia vem do grego κρυπτος que significa oculto, e γραφειυ, que significa escritura, sua definição é: "Arte de escrever com chave secreta ou de um modo enigmático". Obviamente há anos que a criptografia deixou de ser uma arte para virar uma técnica, ou melhor, um conjunto de técnicas que tratam da proteção - ocultamento frente a observadores não autorizados - da informação.

Dentro da criptologia a ciência da criptografia tem como seu objeto de estudos os processos de Encriptação ou seja, a transformação dos dados em uma forma que torna impossível a sua leitura sem o apropriado conhecimento. O seu propósito é assegurar privacidade da informação mantendo o entendimento da mensagem oculto de qualquer um a qual ela não seja destinada. A decriptação, por outro lado, é o reverso da encriptação; é a transformação de dados encriptados novamente em uma forma inteligível. Ou seja, é uma ciência capaz de prover meios através dos quais seja possível transormar um texto “plano” (inteligível) em um texto “cifrado” (ininteligível) e vice-versa.

Encriptação e decriptação geralmente requerem o uso de uma informação secreta que atua como uma chave. Para alguns mecanismos de encriptação a mesma chave é usada para tanto para a cifragem dos dados quanto para a sua decifragem; para outros mecanismos as chaves usadas para a encriptação e decriptação são diferentes.

A criptografia fornece técnicas para codificar e decodificar dados, tais que os mesmos possam ser armazenados, transmitidos e recuperados sem sua alteração ou exposição. Em outras

palavras, técnicas de criptografia podem ser usadas como um meio efetivo de proteção de informações suscetíveis a ataques, estejam elas armazenadas em um computador ou sendo transmitidas pela rede.

Todos os algoritmos de criptografia residem no conhecimento de uma chave secreta que é utilizada pelos algoritmos para criptografar dados. Em resumo:

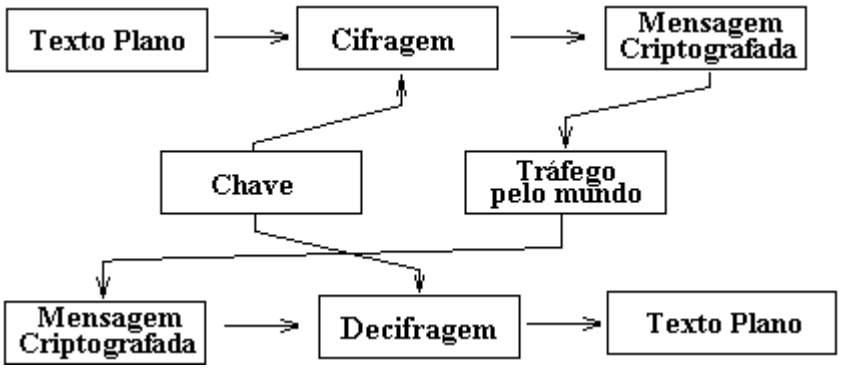


Figura 2.1- Criptografia e transmissão de dados

Como os algoritmos são conhecidos (e devem sê-lo, pois precisam ser exaustivamente testados e validados), o que garante os serviços é a chave secreta. Ela deve ter um tamanho suficientemente grande que impeça sua descoberta por busca exaustiva, mas suficientemente pequena para viabilizar o processamento com o mínimo de overhead.
Podemos definir criptosistema como uma quíntupla (M, C, K, E, D), onde:

•M representa o conjunto de todas as mensagens sem cifrar (denominado texto plano, plaintext) que podem ser enviadas.

•C representa o conjunto de todas as possíveis mensagens cifradas, ou criptogramas.

•K representa o conjunto de chaves que se pode empregar no criptosistema

•E é o conjunto de transformações de criptografia, ou a família de funções que se aplica a cada elemento de M para obter um elemento de C. Existe uma transformação diferente E_k para cada valor de k.

•D é o conjunto de transformações de decriptografia, análogo a E.

Existem dois tipos fundamentais de criptosistemas:

•**Criptosistemas simétricos ou de chave privada:** são aqueles que empregam a mesma chave k tanto para cifrar/criptografar quanto para decifrar/decriptografar. Apresentam o inconveniente de que para serem empregados a chave k deve estar tanto no emissor como no receptor, e isto nos leva ao problema de como transmitir a chave de forma segura.

•**Criptosistemas assimétricos ou de chave pública:** estes criptosistemas empregam uma dupla chave (k_p e k_P). kp é conhecida como chave privada e k_P, como chave pública. Uma delas deve ser usada para criptografia (E) e a outra para a decriptografia (D). Em muitos casos não tem uma ordem específica de utilização, ou seja, uma mensagem criptografada com uma das chaves pode ser decriptografada pela outra, não importando qual será usada para criptografar, desde que a outra seja usada para decriptografar. (ou seja, se encripto com a chave privada, posso decriptar com a chave pública e vice-versa.) Estes criptosistemas devem impedir que a partir da chave pública seja possível calcular a chave privada. Estes criptosistemas oferecem um leque maior de possibilidades, podendo ser empregados para o estabelecimento de comunicações seguras por canais inseguros - uma vez que só trafega pelo canal a chave pública, que só vai ser usada para encriptar.

Na prática empregamos uma combinação dos dois tipos de criptosistemas, pois o segundo possui o inconveniente de ter um custo computacional muito maior que o primeiro. Esta combinação se dá da seguinte forma: as mensagens (geralmente longas) são criptografadas utilizando-se algoritmos simétricos, que podem ser muito eficientes, e a criptografia assimétrica é utilizada para codificar as chaves simétricas (curtas) resolvendo assim o problema de transmissão das chaves no criptosistema simétrico.

O objetivo fundamental da criptografia é tornar possível que duas pessoas, organizações ou entidades, se comuniquem através de um canal inseguro, de forma que um adversário não possa entender o que foi dito. Para isto utilizamos o processo de cifragem/decifragem de mensagens.

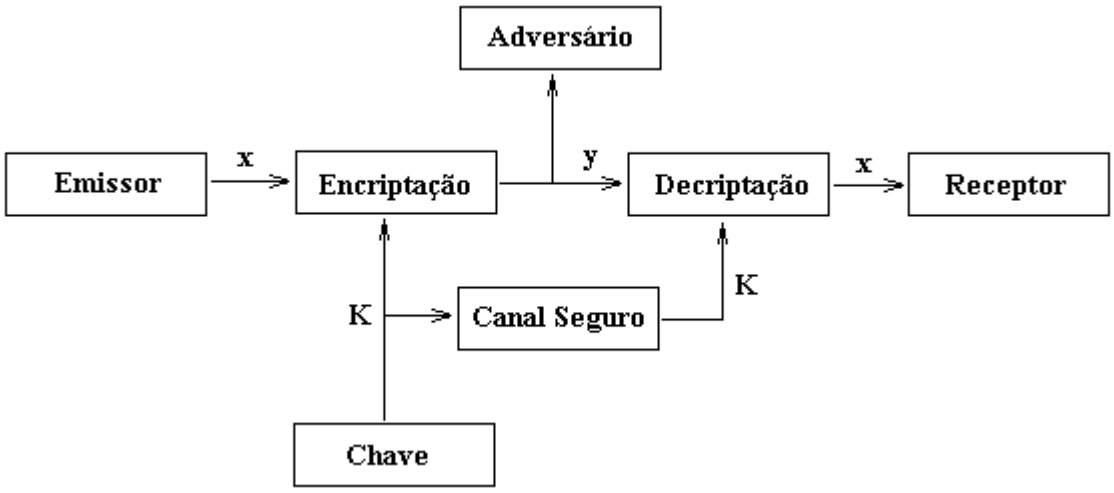


Figura 2.2- Criptografia e ataques

Cifrar (ou criptografar) é o ato de transformar dados em alguma forma ilegível. Seu propósito é o de garantir a privacidade, mantendo a informação escondida de qualquer pessoa não autorizada, mesmo que esta consiga visualizar os dados criptografados. Enquanto que, decifrar (ou decriptografar) é o processo inverso, ou seja, transformar os dados criptografados na sua forma original, inteligível.

Para cifrarmos ou decifrarmos uma mensagem, necessitamos de informações confidenciais geralmente denominadas chaves ou senhas. Dependendo do método de criptografia empregado, a mesma chave pode ser utilizada tanto para criptografar como para decriptografar mensagens, enquanto outros mecanismos utilizam chaves diferentes.

Em geral, considera-se a necessidade de transmitir uma mensagem (M), entre um emissor e um receptor. A mensagem designa-se por texto plano. Na realidade, esta designação não significará texto propriamente dito, corresponderá antes a qualquer seqüência de bits que se pretenda transmitir em segurança. O processo de disfarçar a mensagem, cifragem, transforma o texto simples num criptograma (C); e o processo inverso, decifragem, permite recuperar o texto simples original a partir do criptograma.

Os algoritmos de criptografia, também denominados cifras, são as funções matemáticas que fazem a cifragem e a decifragem, tendo em geral dois componentes o algoritmo ou função de cifragem (E)

$$E(M) = C$$

e o algoritmo ou função de decifragem (D)

$$D(C) = M$$

devendo, naturalmente, verificar-se

$$D(E(M)) = M$$

Todos os atuais algoritmos seguros são conhecidos e usam, no seu funcionamento, uma chave. Basicamente a chave é um número k

$$k \in K$$

em que K é o espaço finito das chaves, que interessa ser de grande dimensão, para uma maior segurança.

Em algoritmos com chave, as três primeiras equações, podem rescrever-se na forma

$$E_k(M) = C$$

$$D_k(C) = M$$

$$D_k(E_k(M)) = M$$

As chaves devem definir univocamente o criptograma, ou seja

$$E_{k1}(M) \neq E_{k2}(M)$$

se

$$k_1 \neq k_2$$

Com maior generalidade, considera-se a utilização de duas chaves diferentes, uma de cifragem e outra de decifragem sendo:

$$E_{k1}(M) = C$$

$$D_{k2}(C) = M$$

$$D_{k2}(E_{k1}(M)) = M$$

Nos algoritmos simétricos, a chave de cifragem pode ser obtida a partir da chave de decifragem, e vice-versa, sendo as duas chaves, normalmente, idênticas. Em qualquer caso, é necessário, que o emissor e o receptor acordem numa chave, antes de poderem usar o sistema.

Nestes algoritmos, a segurança reside no secretismo da chave. Quando a segurança de um algoritmo é baseada no seu secretismo, ele é classificado como restrito. Visto que o funcionamento do algoritmo é conhecido, sabendo-se a chave, é possível cifrar e decifrar qualquer mensagem. Atualmente este tipo de algoritmo é pouco utilizado porque quanto maior é o número de utilizadores, maior é a probabilidade de algum deles revelar o segredo, quebrando a segurança de todo o sistema. A sua utilização restringe-se a aplicações de baixa segurança (codificadores de vídeo, por exemplo).

De modo a minimizar este problema surgiram os algoritmos assimétricos ou de chave pública, em que as duas chaves são obrigatoriamente diferentes, com a condicionante de a chave de decifragem ser impossível de obter a partir da chave de cifragem (pelo menos num tempo aceitável).

Denomina-se chave pública, porque a chave de cifragem pode ser divulgada, permitindo a qualquer emissor enviar mensagens cifradas para um destinatário. Como somente o destinatário conhece a chave de decifragem, somente a ele será possível a decifragem da mensagem. O sistema inverso também é possível, isto é, publicar a chave de decifragem e manter secreta a chave de cifragem, e designa-se por sistema de assinatura. Permite garantir a autenticidade das mensagens do emissor.

Os algoritmos de cifragem podem também se dividir em duas categorias conforme a maneira de subdividir as mensagens a cifrar. Os algoritmos de cifra corrida (stream cipher) fazem um tratamento bit a bit do texto original. Os algoritmos de blocos tratam um determinado número de bits simultaneamente. Em implementações usando computador a dimensão do bloco é normalmente 64 bits - um valor suficientemente grande para afastar a criptoanálise e suficientemente pequeno para ser tratável.

Na grande maioria dos casos tanto os textos planos (M) como os textos criptografados (C) são representados utilizando-se o mesmo alfabeto. Por isso, pode ocorrer que exista alguma chave k pertencente ao conjunto de chaves possíveis ($k \in K$) de tal forma que

$$E_k(M) = M$$

o que seria desastroso para nossos propósitos, pois o uso desta chave deixaria a mensagem sem alteração.

Também pode acontecer o caso de chaves que geram textos codificados de baixa qualidade. Uma possibilidade muito comum em certos algoritmos é que algumas chaves têm a seguinte propriedade:

$$E_k(E_k(M)) = M$$

ou seja, bastaria criptografar novamente o criptograma para recuperar o texto original.

A existência de chaves com estas características, depende de cada algoritmo e em muitos casos também depende dos parâmetros escolhidos na hora de aplica-lo. Denominaremos estes tipos de chave como chaves fracas (weak keys). Normalmente em um bom criptosistema, a quantidade de chaves fracas é zero ou muito pequena em comparação com o número total de chaves possíveis.

2.2 Princípios básicos da criptografia

Os princípios básicos da segurança são: a Autenticidade, Confidencialidade, Integridade e Disponibilidade das Informações. Os benefícios evidentes são reduzir os riscos com vazamentos, fraudes, erros, uso indevido, sabotagens, roubo de informações e diversos outros problemas e, assim, consequentemente aumentar a produtividade dos usuários através de um ambiente mais organizado, maior controle sobre os recursos de informática e, finalmente, viabilizar aplicações críticas das empresas.

•Autenticidade

O controle de autenticidade está associado com a identificação correta de um usuário ou computador. O serviço de autenticação em um sistema deve assegurar ao receptor que a mensagem é realmente procedente da origem informada em seu conteúdo. Normalmente, isso é implementado a partir de um mecanismo senhas ou de assinatura digital. A verificação de autenticidade é necessária após todo processo de identificação, seja de um usuário para um sistema, de um sistema para o usuário ou de um sistema para outro sistema. Ela é a medida de proteção de um serviço/informação contra a personificação por intrusos.

Um ataque contra a autenticidade envolve alguma forma de personificação (spoofing). Um tipo comum de personificação consiste em um usuário externo assumir a identidade de um usuário interno, atuando no sistema no lugar deste usuário legítimo. A maneira mais simples de personificação está associada com infiltrações de senha, onde o intruso informa uma combinação de nome do usuário/senha, depois outra e assim por diante, até que uma determinada combinação permita sua entrada no sistema. Tal técnica (usualmente denominada como Força Bruta ou Brute Force) consome, com freqüência, um volume considerável de tempo e esforço de máquina. Assim classes de softwares como os sniffers, que possibilitam o rastreamento de senhas, estão sendo utilizados cada vez em maiores escalas.

Muitos tipos de sistemas não bloqueiam tentativas de login após um determinado número de insucessos. Essa fraqueza inerente em termos de segurança, permite que um intruso dê início a um grande número de tentativas de login que não são impedidas. Consequentemente, possibilita aos violadores várias formas de invasão: acessando mensagens de correio eletrônico, os quais contêm senhas; ou decifrando-as com uma ferramenta que permite localizar e obter informações sobre senhas vulneráveis em sistemas. Na verdade, alguns invasores utilizam TFTP ou FTP para tentar obter a senha, em seguida o invasor deverá identificar as senhas verdadeiras.

No Unix, as senhas contidas em /etc/passwd são cifradas através de um esquema de criptografia não-convencional, mas o algoritmo de criptografia em si está largamente disponível e pode ser até mesmo incorporado em algumas ferramentas utilizadas pelos invasores. Os invasores as utilizam para obter senhas em textos simples que serão informadas durante sessões de telnet ou de rlogin.

•Confidencialidade

Confidencialidade significa proteger informações contra sua revelação para alguém não autorizado - interna ou externamente. Consiste em proteger a informação contra leitura e/ou cópia por alguém que não tenha sido explicitamente autorizado pelo proprietário daquela informação. A informação deve ser protegida qualquer que seja a mídia que a contenha, como por exemplo, mídia impressa ou mídia digital. Deve-se cuidar não apenas da proteção da informação como um todo, mas também de partes da informação que podem ser utilizadas para interferir sobre o todo. No caso da rede, isto significa que os dados, enquanto em trânsito, não serão vistos, alterados, ou extraídos da rede por pessoas não autorizadas ou capturados por dispositivos ilícitos.O objetivo da confidencialidade é proteger informação privada (cidadãos, indústrias, governo, militar). Na comunicação, ela é obtida evitando-se a escuta (meio físico, topologia), ou se isto não for possível, evitando-se a inteligibilidade dos dados durante o processo de transmissão (cifra).

Uma rede de meios físicos compartilhados é uma rede na qual os pacotes são transmitidos para várias partes da rede à medida que trafegam dos pontos de origem para os de destino. As redes de meios físicos compartilhados impõem um tipo especial de risco de segurança, pois os pacotes podem ser interceptados em qualquer ponto dessas redes. A captura de pacotes dessa forma é conhecida como Rastreamento da Rede. Para o rastreamento de uma rede é preciso usar um dispositivo físico ou um programa. Normalmente, os dispositivos físicos de rastreamento são instalados onde há conexão de cabos, através de um conector dentado que penetra no isolamento do cabo, ou em interfaces de porta de máquina host individuais. Os programas de captura de pacotes proporcionam uma interface com um dispositivo de hardware que é executado no modo promíscuo (sniffer), ou seja, copiando todos os pacotes que chegam até ele, independentemente do endereço de destino contido no pacote.

Se um sniffer for instalado em alguma parte da rota entre dois hosts de uma rede, senhas e informações confidenciais podem ser capturadas, causando transtornos e prejuízos. Tal ação pode proporcionar, também, a ocorrência de futuros ataques contra autenticidade, usando senhas, usernames e endereços de host capturados por sniffers.

•Integridade

A integridade consiste em proteger a informação contra modificação sem a permissão explicita do proprietário daquela informação. A modificação inclui ações como escrita, alteração de conteúdo, alteração de status, remoção e criação de informações. Deve-se considerar a proteção da informação nas suas mais variadas formas, como por exemplo, armazenada em discos ou fitas de backup. Integridade significa garantir que se o dado está lá, então não foi corrompido, encontra-se íntegro. Isto significa que aos dados originais nada foi acrescentado, retirado ou modificado. A integridade é assegurada evitando-se alteração não detectada de mensagens (ex. tráfego bancário) e o forjamento não detectado de mensagem (aliado a violação de autenticidade).

•Disponibilidade

Ter as informações acessíveis e prontas para uso representa um objetivo crítico para muitas organizações. No entanto, existem ataques de negação de serviços, onde o acesso a um sistema/aplicação é interrompido ou impedido, deixando de estar disponível; ou uma aplicação, cujo tempo de execução é crítico, é atrasada ou abortada.

Disponibilidade consiste na proteção dos serviços prestados pelo sistema de forma que eles não sejam degradados ou se tornem indisponíveis sem autorização, assegurando ao usuário o acesso aos dados sempre que deles precisar. Isto pode ser chamado também de continuidade dos serviços.

Um sistema indisponível, quando um usuário autorizado necessita dele, pode resultar em perdas tão graves quanto as causadas pela remoção das informações daquele sistema. Atacar a disponibilidade significa realizar ações que visem a negação do acesso a um serviço ou informação, como por exemplo: bloqueamento do canal de comunicação ou do acesso a servidores de dados.

2.3 Algumas definições importantes

Algoritmos de criptografia em blocos

Estes algoritmos dividem o texto plano em blocos de tamanho fixo e estes blocos são encriptados um por vez.

Algoritmos de criptografia de fluxo

São algoritmos que vão encriptando os dados conforme estes são recebidos. Não contém uma fase preparatória de divisão do texto plano.

Robustez criptográfica

Alguns sistemas sao mais fáceis de atacar que outros. A habilidade do sistema de criptografia de proteger informaçoes para ataques é chamada robustez. Robustez depende de muitos fatores, incluindo:

•A segredo da chave.

•A dificuldade de adivinhar a chave ou as árduas possibilidades de buscar a chave.

•A dificuldade de inverter o algoritmo de encriptaçao sem conhecer a chave de encriptaçao.

•A existência de portas traseiras, ou caminhos adicionais pelas quais um arquivo encriptado pode ser desencriptado mais facilmente sem conhecer a chave.

• A habilidade para decriptar uma mensagem encriptada totalmente se você conhece o caminho que um porçao dele decriptado (chamado um atacar texto conhecido).

• A propriedade do plaintext e conhecimento destas propriedades por um ataque.

•O objetivo no projeto da criptografia é desenvolver um algoritmo que é difícil reverter sem a chave, e necessitar de grande esforço para poder adivinhar a chave. Algumas matemáticas sofisticadas sao envolvidas em tais desenvolvimentos.

Tamanho criptográfico

Quando dizemos que um método de criptografia utiliza 512, 1024 ou 2048 bits estamos dizendo que o número de combinaçoes que alguém precisa fazer para tentar decifrar nossa mensagem sao respectivamente 2 elevado à potência 512, 2 elevado à potência 1024 e 2 elevado à potência 2048 potência.

Já pensou quantas vezes alguém teria que tentar para decifrar uma mensagem criptografada em 2048 bits? Claro que quem quer decifrar uma mensagem de correio eletrônico também vai se valer de meios e programas cada vez mais sofisticados para isto, e por isto mesmo a criptografia deve ser cada vez mais forte para impedir tentativas de quebra de sigilo. Qualquer método acima de 512 bits é considerado hoje bastante seguro para mensagens de correio eletrônico.

Os programas de criptografia utilizam como código algoritmos (equaçoes matemáticas, polinômios) complexos que, em sua fórmula, utilizam como parte integrante o próprio texto a ser criptografado e uma "chave de criptografia".

Expiraçao de chaves

Para se precaver de ataques fatoriais a longo prazo, cada chave deve ter uma data de expiraçao após a qual ela se torna inválida. O tempo de vida da chave deve ser, portanto, bem menor do que o tempo previsto para poder quebrá-la, ou, em outras palavras, o tamanho da chave deve ser grande o suficiente para tornar mínimas as chances de quebrá-la antes de sua expiraçao.

A data de expiraçao de uma chave acompanha a chave pública em uma Identificaçao Digital. O programa de verificaçao de assinatura deve verificar a data de expiraçao, e nao aceitar uma mensagem assinada com uma chave expirada. Isto significa que quando a chave de alguém expira, tudo o que for assinado com esta chave nao deve ser considerado válido. Quando for necessário que um documento assinado seja considerado válido por períodos mais longos, o documento deve receber um selo cronológico.

Após a expiraçao, o usuário escolhe uma nova chave que deve ser mais longa do que a chave antiga, possivelmente alguns dígitos a mais. Um usuário pode reverificar uma chave expirada, se ela for suficientemente longa e se nao foi comprometida. A Autoridade

Certificadora Digital gerará entao uma nova Identificaçao Digital para a mesma chave, e todas as novas assinaturas farao mençao à nova Identificaçao Digital, em vez da antiga. Entretanto, pelo fato de os computadores se tornarem mais potentes e velozes, é recomendável que novas chaves sejam mais longas a cada ano.

Trocas de chaves têm a vantagem de aumentar a segurança no sistema de criptografia.

Criptoanálise

A força de um sistema criptográfico depende de vários fatores: dificuldade de adivinhar a chave (o uso de chaves maiores são mais difíceis de adivinhar, mas podem tornar o processo mais lento), dificuldade de subverter o algoritmo de cifragem, dificuldade de se quebrar o código mesmo já conhecendo a mensagem cifrada, entre outros.

Tipos de Ataques

1 - Ataque do texto cifrado (cyphertext-only): o criptoanalista tem a sua disposição uma grande quantidade de mensagens cifradas, mas desconhece as originais e as chaves utilizadas. Sua tarefa é recuperar as mensagens normais (deduzir as chaves utilizadas).

2 - Ataque do texto conhecido (known-plaintext): o criptoanalista tem a sua disposição uma grande quantidade de mensagens cifradas e também as mensagens originais equivalentes. Sua tarefa é deduzir as chaves usadas (ou um método para recuperar mensagens cifradas com a mesma chave).

3 - Ataque adaptativo do texto escolhido (adaptative-choosen-plaintext): no método anterior, o criptoanalista poderia ser capaz de fornecer somente uma grande quantidade de mensagens de uma só vez; agora ele pode fornecer um pequeno conjunto, analisar os resultados, fornecer outro conjunto e assim por diante. Sua tarefa é deduzir as chaves utilizadas. Alguns métodos de cifragem como o RSA são muito vulneráveis a este ataque.

4 - Ataque do texto cifrado escolhido (choosen-cyphertext): o criptoanalista não só tem uma grande quantidade de mensagens e seus equivalentes cifrados, mas pode produzir uma mensagem cifrada específica para ser decifrada e obter o resultado produzido. É

utilizado quando se tem uma "caixa-preta" que faz decifragem automática. Sua tarefa é deduzir chaves utilizadas.

5 - Ataque da chave escolhida (choosen-key): o criptoanalista pode testar o sistema com diversas chaves diferentes, ou pode convencer diversos usuários legítimos do sistema a utilizarem determinadas chaves. Neste último caso, a finalidade imediata seria de decifrar as mensagens cifradas com essas chaves.

6 - Ataque de força bruta: o criptoanalista procura descobrir a senha usada na cifragem de uma mensagem tentando todas as chaves possíveis. Portanto, quanto maior a chave, mais difícil que esse ataque seja bem-sucedido.

Um sistema é dito seguro se ele é teoricamente inquebrável, ou seja, não interessa qual a quantidade de texto normal ou decifrado a disposição, nunca se tem informação suficiente para deduzir as chaves utilizadas ou decifrar um texto qualquer cifrado.

Não existem mecanismos de cifragem/decifragem 100% eficazes, numa abordagem puramente teórica é imediato que qualquer chave pode ser quebrada pela força bruta (supondo que dispõe de um exemplar de uma mesma mensagem original e cifrada, e o algoritmo é conhecido, basta tentar com todas as chaves possíveis até acertar).

A solução é entrar no domínio prático e atender às capacidades do equipamento de processamento atual de modo a usar algoritmos e chaves que não possam ser descobertas em tempo útil. O tempo necessário para quebrar uma chave pela "força bruta" depende do número de chaves possíveis (número de bits da chave) e do tempo de execução do algoritmo.

O grande problema desta abordagem é que a capacidade de processamento dos equipamentos tem duplicado de 18 em 18 meses, logo de 18 em 18 meses é necessário aumentar um bit às chaves.

Só se conhece um método nesta categoria: a Cifra de Vernam ou One-time pad (cifra de uso único). Em essência dois elementos que desejam se comunicar possuem cópias idênticas de uma seqüência randômica de valores, que são usados como chave. O método entretanto, exige que cada chave seja usada uma única vez e que o comprimento da seqüência (chave) seja maior, ou no mínimo igual ao comprimento da mensagem a ser cifrada.

3 Uma breve introdução sobre algoritmos simétricos

3.1 Introdução

Sistema simétrico ou de chave secreta é o método de encriptação que utiliza uma mesma chave (o segredo, como nos tempos antigos) para encriptar e decriptar a mensagem. Esta forma também é conhecida como Criptografia por Chave Secreta ou Chave Única, Criptografia Simétrica ou Criptografia Tradicional.

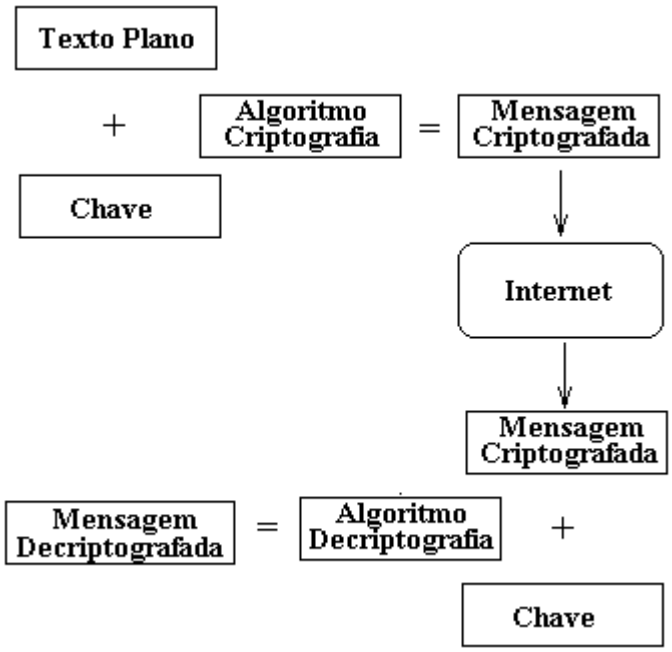


Figura 3.1- Esquema para cifragem e decifragem com chaves simétricas

A ilustração acima, demostra de forma simples a chave simétrica em funcionamento. A soma da mensagem mais chave gera uma mensagem criptografada, após a geração, ela é

enviada através da rede, e ao chegar ao lado oposto, ela é descriptografada através da chave que está no destino.

A chave pode ser uma palavra, uma frase ou uma sequência aleatória de números e deve ser conhecida tanto pelo remetente quanto pelo receptor da mensagem (o remetente a usa para cifrar a mensagem e o receptor, para decifrá-la). Seu tamanho é medido em bits e, via de regra, quanto maior a chave, mais seguro será o documento encriptado. Sua geração, transmissão e armazenamento é denominada Gerência de Chaves.

O principal desafio (e fragilidade) deste método é garantir que ninguém mais conheça a chave além do transmissor e receptor originais. Para tanto eles a devem trocar pessoalmente ou possuir um sistema de entrega, telefone ou outro meio de transmissão confiável capaz de garantir a confiabilidade do segredo. Pois qualquer um que venha, de alguma forma, a ter conhecimento desta chave pode mais tarde ler, modificar ou forjar mensagens encriptadas ou autenticadas que utilizem aquela chave.

Dada esta necessidade os sistemas de criptografia por chave única apresentam dificuldades em garantir plena segurança, especialmente em ambientes abertos com um grande número de usuários. Eles funcionam bem em aplicações limitadas, onde o remetente e o destinatário se preparam antecipadamente para o uso da chave.

Um outro problema é que como ambos conhecem a chave, este método permite o repúdio (foi ele, não fui eu...). Uma forma de evitar o repúdio seria cada par de parceiros ter a sua chave, mas neste caso o problema de distribuição cresceria exponencialmente.

A grande vantagem da criptografia de chave secreta é que ela é muito rápida (existem implementados em hardware).

A encriptaçao por chave privada funciona muito bem quando o usuário que encripta é o mesmo que desencripta o arquivo (por exemplo, para proteger arquivos que ficam armazenados no próprio disco rígido).

3.2 Sumário de sistemas de chaves privadas

Abaixo temos alguns algoritmos usados hoje.

•**Crypt** – o programa de encriptaçao original do UNIX que é modelado no Enigma da Máquina de encriptaçao Alemã. Crypt usa uma chave de tamanho variável. Alguns programas podem automaticamente decriptar arquivos crypt encriptados sem prévio conhecimento da chave ou o plaintext. Crypt nao é seguro. O programa crypt é usado pelo UNIX para encriptar senhas.

•**DES** – o Data Encryption Standard (DES) (padrão encriptação de dados), um algoritmo de encriptaçao desenvolvido na década de 70 pelo National Bureau of Standards and Technology (desde então entitulado o National Institute of Standards and Technology, ou NIST) e a IBM. O DES utiliza uma chave de 56 bits e opera em blocos de 64 bits.

Foi projetado inicialmente para ser utilizado em componentes de hardware, nos dias atuais, ele é usado na Internet em conexões Web segura (o SSL utiliza o DES). Ele é um algoritmo seguro para a maioria das aplicações, entretanto, em aplicações altamente secretas, ele não deve ser usado, pois existe o perigo de violação.

•**IDEA** - International Data Encryption Algorithm, desenvolvido na década de 80 por Xuejia Lai e James Massey da ASCOM Tech AG da Suiça, em Zurique. Ele foi projetado para ser facilmente programado, é forte e resistente a muitas formas de criptoanálise. O seu método baseia-se na utilização de uma chave de 128 bits, onde blocos de texto de 64 bits da mensagem de entrada são alterados em uma seqüência de interações, produzindo blocos de saída. Sua chave de 128 bits, é o suficiente para resistir à maior parte dos ataques. IDEA é usado pelo popular programa PGP, para encriptar arquivos e correio eletrônico. Infelizmente, o largo uso do IDEA é impedido por uma série de patentes de software da ASCOM Tech. A ASCOM Tech apenas concedeu o uso gratuito em implementações do PGP fora dos Estados Unidos, mas devendo verificar os termos e suas licenças.

•**RC2 e RC4**. Mais rápidos do que o DES, esses códigos podem se tornar mais seguros com o simples aumento do tamanho das chaves. O RC2 pode substituir perfeitamente o DES com a vantagem de ser 2 vezes mais rápido, já o RC4 fica 10 vezes mais rápido.

•**RC5** – um bloco de encriptaçao desenvolvido por Ronald Rivest e publicado em 1994.

O RC5 permite um uso definido do tamanho da chave, tamanho dos blocos de dados, e número de encriptaçoes salvas.

•**Skipjack** – um algoritmo desenvolvido pela National Security Agency (NSA) usa uma chave de 80 bits. É classificado como secreto.

4 Cifras clássicas

4.1 Introdução

Neste capítulo faremos um breve comentário dos algoritmos de criptografia considerados clássicos. Podemos chamar assim a todos os sistemas de criptografia anteriores à 2ª Guerra Mundial, o que coincide com o nascimento dos computadores.

Estas técnicas têm em comum o fato de poderem ser empregadas usando-se apenas lápis e papel, e poderem ser decifradas praticamente da mesma forma. Atualmente com a ajuda dos computadores, as mensagens criptografadas empregando-se estes algoritmos são facilmente decifradas, por isso caíram rapidamente em desuso.

Todos os algoritmos clássicos são simétricos, já que até meados dos anos sessenta ainda não havia nascido a criptografia assimétrica.

A transição da criptografia clássica para a moderna se fez presente durante a 2ª Guerra Mundial, quando o serviço de inteligência aliado decifra/quebra a máquina de criptografia do exército alemão, chamada Enigma.

Os sistemas de criptografia clássicos perderam sua eficácia devido à facilidade com que atualmente são decodificados/criptanalizados empregando-se qualquer computador doméstico, mas que foram empregados com êxito até princípios do século XX. Alguns se mantiveram restritos, como o caso do algoritmo de Caesar, usado apenas na Roma Imperial. Sem emprego na atualidade estão aqui apresentados, além dos motivos históricos, pelo fato de nos permitir observar algumas características presentes também nos algoritmos modernos.

4.2 Cifras por substituição

Cifras por substituição são aquelas em que os caracteres da mensagem em texto plano são sistematicamente substituídos por outros caracteres.

Na criptografia clássica existem quatro tipos de cifras por substituição:

•Cifras de substituição simples, de deslocamento ou monoalfabéticas:

•Cifras por substituição homofônica:

•Cifras de substituição polialfabética

•Cifras por substituição de múltiplos símbolos:

4.2.1 Cifras por deslocamento ou monoalfabéticas

Cifras monoalfabéticas são todos os algoritmos de criptografia que, sem desordenar os símbolos dentro da mensagem, estabelecem um mapeamento único para todos eles em todo o texto. Ou seja, se o símbolo A corresponde ao símbolo D, esta correspondência se mantém por toda a mensagem.

Como exemplo temos a cifra de caesar, rot13 e shift-n

As cifras por deslocamento descritas nesta seção são baseadas na aritmética modular.

Para uma pequena introdução:

O resultado da operação módulo é o resto da divisão inteira entre os dois números considerados.

Suponha que queiramos calcular 11 x 13 em Z16. Nos inteiros teríamos 11 x 13 = 143. Para reduzir 143 em módulo 16, fazemos uma divisão: 143 = 8 x 16 + 15 e conseqüentemente, 11 x 13 = 15 em Z16.

(20 + 15) mod 8 = 35 mod 8 = 3

No decorrer deste trabalho adotaremos o módulo sempre como um resultado não negativo, assim:

-18 mod 7 = 3 , e não -4

(7 - 10) mod 15 = 7 - 10 + 15 = 12, e não -3

Primeiramente faremos uma correspondência entre as letras do alfabeto e seus resíduos módulo 35, conforme pode ser visto na tabela abaixo:

A	00	I	08	Q	16	Z	24	7	32
B	01	J	09	R	17	0	25	8	33
C	02	K	10	S	18	1	26	9	34
D	03	L	11	T	19	2	27		
E	04	M	12	U	20	3	28		
F	05	N	13	W	21	4	29		
G	06	O	14	X	22	5	30		
H	07	P	15	Y	23	6	31		

Figura 4.1- Correspondênica numérica entre as letras do alfabeto e seus resíduos

4.2.1.1 Caesar

Talvez o primeiro dos sistemas de criptografia teve sua origem com Julio Caesar, o algoritmo de Caesar. Ele é chamado assim por ser o que Júlio Caesar empregava para enviar suas mensagem secretas, é um dos algoritmos mais simples. Consiste em somar 3 ao número de ordem de cada letra. Desta forma o A corresponde ao D, o B ao E, ..., o W ao Z, o X ao A, o Y ao B e o Z ao C. Como podemos notar, o alfabeto do texto criptografado é simplesmente uma rotação do alfabeto do texto plano.

Se usarmos a tabela 4.1 e considerarmos o alfabeto de 35 letras, a transformação seria:

$$C = (M + 3) \bmod 35$$

onde C é o texto criptografado e P é o texto plano (original).

Podemos observar, devido à simplicidade do algoritmo, que para decifrá-lo basta subtrair 3 do número correspondente a cada letra na mensagem criptografada. Ou seja, este algoritmo nem sequer possui chave, uma vez que a transformação é sempre a mesma.

Suponha queiramos enviar a mensagem abaixo:

mensagem a ser enviada

Para encriptar a mensagem acima primeiramente retiramos todos os espaços entre as letras:

mensagemaserenviada

Transformamos cada letra em seu número correspondente, seguindo a ordem anteriormente citada: M = 12, E = 04,...

12 04 13 18 00 06 04 12 00 18 04 17 04 13 21 08 00 03 00

Somamos 3 a cada letra, reduzindo mod 26

15 07 16 21 03 09 07 15 03 21 07 21 07 14 24 11 03 06 03

Reescrevemos a mensagem utilizando novamente a tabela

PHQVDJHPDVHVHOYLDGD

a qual corresponde à mensagem criptografada.

Para decriptar procedemos de maneira análoga.

PHQVDJHPDVHVHOYLDGD

Transformamos a mensagem em números:

15 07 16 21 03 09 07 15 03 21 07 21 07 14 24 11 03 06 03

Subtraímos 3 de cada número, pois queremos decriptografá-la:

12 04 13 18 00 06 04 12 00 18 04 17 04 13 21 08 00 03 00

Fazemos novamente a transformação utilizando a tabela acima:

mensagemaserenviada

Obtendo assim, a mensagem original em texto plano.

4.2.1.2 Rot 13

Este método, muito semelhante ao de Caesar, consiste em substituir uma letra por outra situada treze posições mais à frente no alfabeto, por isso Rot 13.

Para encriptar:

Utilizaremos o mesmo texto plano do exemplo anterior

mensagemaserenviada

Transformamos cada letra em seu número correspondente, seguindo a ordem anteriormente citada: M = 12, E = 04,...

12 04 13 18 00 06 04 12 00 18 04 17 04 13 21 08 00 03 00

Somamos 13 a cada letra, reduzindo mod 35

25 17 00 05 13 19 17 25 13 05 17 05 17 24 08 21 13 16 13

Reescrevemos a mensagem utilizando novamente a tabela

ZRAFNTRZNFRFRYIVNQN

a qual corresponde à mensagem criptografada.

Para decriptar:

Procedemos de maneira análoga ao exemplo anterior, subtraindo 13.

4.2.1.3 Shift-n

Neste método, como nos anteriores, substituímos as letras por outras situadas mais à frente no alfabeto. A diferença neste caso é que além de podermos alterar o alfabeto, fazendo-o acomodar quaisquer caracteres que queiramos, definimos o número de deslocamentos a serem realizados em cada letra. Este deslocamento será dado pela chave K.

Assim, na hora de decriptar a mensagem, além da chave, deve-se conhecer o alfabeto utilizado.

A	00
B	01
C	02
D	03
E	04
F	05
G	06
H	07

I	08
J	09
K	10
L	11
M	12
N	13
O	14
P	15

Q	16
R	17
S	18
T	19
U	20
W	21
X	22
Y	23

Z	24
0	25
1	26
2	27
3	28
4	29
5	30
6	31

7	32
8	33
9	34

Para encriptar:

Para o alfabeto acima vamos encriptar a mensagem abaixo utilizando K = 10

senha 253158

transformando em números

18 04 13 07 00 28 31 29 27 31 34

somando a chave (K = 10) e reduzindo mod 35

28 14 23 17 10 03 06 04 02 06 09

que de volta à tabela 4.2 torna-se

2OXRKDGECGJ

Para decriptar, procedemos como nos anteriores, subtraímos o valor da chave e reduzimos mod 35.

4.2.2 Cifras polialfabéticas

Foram inventadas por Leon Battista em 1568.

No caso das cifras por substituição, uma vez escolhida a chave, cada caractere do texto plano é mapeado sempre no mesmo caractere do texto criptografado. Por essa razão, estes sistemas de criptografia são chamados monoalfabéticos.

No caso das cifras polialfabéticas, existem múltiplas chaves e cada uma é usada para encriptar uma letra do texto plano. Ou seja, a substituição aplicada a cada caractere varia em função da posição que ele ocupa dentro do texto plano.

A primeira chave encripta a primeira letra do texto plano, a segunda chave encripta a segunda letra, e assim sucessivamente. Depois que todas as chaves são usadas, volta-se para a primeira chave prosseguindo a encriptação até o término do texto. Se tivermos 20 chaves, a cada 20 letras teremos o texto plano sendo encriptado com a mesma chave. Isto é chamado período da cifra.

Na realidade a cifra polialfabética corresponde à aplicação cíclica das cifras monoalfabéticas.

Na criptografia clássica cifras com períodos grande são mais difíceis de quebrar do que aquelas com períodos curtos.

4.2.2.1 Vigenère

Viginère é um exemplo típico de cifra polialfabética.

Neste algoritmo a chave é uma seqüência de símbolos K = { k0, k1,..., kn-1 } e que emprega a seguinte função de cifragem:

$$E_k (m_i) = m_i + k(i \bmod d) \pmod n$$

Onde mi é o i-ésimo símbolo do texto plano e n é a cardinalidade do alfabeto de entrada.

Se decidir colocar as informações em itálico, devo acrescentar ao exemplo, o valor da cardinalidade e falar sobre o alfabeto de entrada.

Usando a correspondência dada pela tabela 4.1 podemos associar cada chave K com uma seqüência de caracteres de tamanho n, chamada palavra chave (Keyword).

Como podemos observar, a cifra de Vigenère encripta n caracteres por vez: cada elemento de texto plano é equivalente aos n caracteres da chave.

Para encriptar:

Dada a palavra chave CIFRA (m=5) fazemos sua correspondência para inteiros (tabela 4.1) K = (02, 08, 05 , 17 00). Suponha que o texto plano seja:

sistema de criptografia

Devemos converte-lo em elementos de resíduo mod 26 (tabela 4.1), escreve-los em grupos de cinco (tamanho da palavra chave) e então adicionar os valores da palavra chave reduzindo mod 26 como segue

Texto Plano: 18 08 18 19 04 12 00 03 04 02 47 08 15 19 14 06 17 00 05 08 00

Chave: 02 08 05 17 00 02 08 05 17 00 02 08 05 17 00 02 08 05 17 00 02

Texto cifrado: 20 16 23 10 04 14 08 08 21 02 19 16 20 10 14 08 25 05 22 08 02

que é o equivalente à:

UQXKEOIVCTQUKOIZFWIC

Para decriptar usamos a mesma palavra chave, mas devemos subtraí-la mod 26 em vez de adicioná-la.

Numa cifra de Vigenère com uma palavra chave de tamanho m, um caracter pode ser mapeado utilizando-se um dos m caracteres da palavra chave (assumindo que a palavra chave contenha m caracteres distintos). Por isso é chamada de polialfabético.
Blaise Vigenère viveu no século 16.

4.2.2.2 Hill

A seguir descreveremos outro sistema de criptografia polialfabética chamado cifra de Hill. Essa cifra foi inventada em 1929 por Lester S. Hill.

A idéia deste sistema de criptografia é fazer m combinações lineares dos m caracteres do texto plano, produzindo os m caracteres do texto criptografado.

Para m=2, um dado texto plano P = (p1, p2) será levado no texto criptografado C = (c1, c2), onde c1 é uma combinação linear de p1 e p2 descrita pela chave K (uma matriz m x m).

Antes de passarmos ao exemplo vamos lembrar que:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{\angle 1} \quad \text{x} \quad \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad = \quad \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

sendo $\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{\angle 1}$ chamada matriz inversa de $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ (Para maiores informações ver REFERÊNCIA).

Para encriptar:

Suponha que queiramos encriptar o texto plano

hoje

com a chave

$$\text{K} = \begin{bmatrix} 11 & 8 \\ 3 & 7 \end{bmatrix}_{2x2}$$

Primeiramente dividimos o texto plano em elementos de 2 caracteres (m=2)

$$\text{ho} = (07, 14) \quad \text{e} \quad \text{je} = (09,04)$$

cada grupo de caractere será multiplicado pela chave K mod 26, gerando um grupo de caracteres criptografados

$$(07, 14) \text{K} = (77 + 42 , 56 + 98) = (119, 154) = (15, 24)$$

e

$$(09, 04) \text{K} = (99 + 12 , 72 + 28) = (111, 100) = (07, 22)$$

assim, o texto criptografado é

PYHW

Para decriptografá-lo deve-se calcular a inversa de K:

$$K^{-1} = \begin{bmatrix} 7 & 18 \\ 23 & 11 \end{bmatrix}_{2x2}$$

e multiplicar cada grupo do texto criptografado por K -1

$$(15, 24)\ K^{-1} = (105 + 552 , 270 + 264) = (657, 534) = (07, 14)$$

e

$$(07, 22)\ K^{-1} = (49 + 506 ,\ 126 + 242) = (555, 368) = (09, 04)$$

recuperando assim, o texto plano:

07 14 09 04

hoje

Neste ponto notamos que a decifragem só é possível se K tiver a inversa. Logo estamos interessados nas matrizes K que são inversíveis.

A inversibilidade de uma matriz quadrada depende do valor de seu determinante. Uma matriz será inversível se e somente se seu determinante for diferente de zero.

4.2.3 Cifras homofônicas

Para diminuir a sensibilidade frente aos ataques baseados no estudo e freqüências de surgimento dos símbolos, existe uma família de algoritmos que trata de ocultar as propriedades estatísticas do texto plano, empregando um alfabeto de saída com mais símbolos que o alfabeto de entrada.

Suponhamos que nosso alfabeto de entrada fossem as quatro letras {a, b, c, d}. Suponhamos também, que em nossos textos a letra a aparece com freqüência 0.4 enquanto as

demais aparecem com freqüência 0.2. Poderíamos empregar o seguinte alfabeto de saída {α, β, γ, δ, ε} efetuando a seguinte associação:

E(a) = α com probabilidade 0.5

β com probabilidade 0.5

E(b) = γ

E(c) = δ

E(d) = ε

Agora no texto cifrado todos os símbolos aparecem com igual probabilidade, o que impossibilita um ataque baseado em freqüências.

Diferente do que se poderia pensar a princípio, este método apresenta vários inconvenientes para ser utilizado na prática; além do problema de precisar de um alfabeto de saída maior que o de entrada, para aplica-la faltaria conhecer antecipadamente a distribuição estatística dos símbolos no texto plano, informação que nem sempre está disponível.

4.2.4 Cifras por substituição de múltiplos símbolos

São cifras nas quais um grupo de caracteres são encriptados juntos, por exemplo a cifra Playfair criada em 1854.

4.3 Cifras por permutação

Todos os sistemas de criptografia discutidos anteriormente envolvem substituição: o texto plano é substituído por diferentes caracteres no texto criptografado, caracteres estes que podem não aparecer na mensagem original (texto plano). A idéia da cifra por permutação é manter os mesmos caracteres do texto plano, apenas rearranjando suas posições.

A cifra por permutação também é conhecida como cifra por transposição e tem sido usada por centenas de anos. De fato a distinção entre cifras por permutação e por substituição foi feita por Giovanni Porta em 1563.

Como no caso das cifras por substituição, usaremos a tabela 4.1 a fim de fazer as trasformações.

Para encriptar:

Suponha m=6 e a chave da permutação p:

1	2	3	4	5	6
3	5	1	6	4	2

Então, a permutação inversa p-1 é a seguinte:

1	2	3	4	5	6
3	6	1	5	2	4

Agora, suponhamos um dado texto plano:

tinha uma pedra no meio do caminho

primeiro agrupamos o texto plano em grupos de seis letras (m=6)

tinhau | mapedr | anomei | odocam | inhoww

Observação: completamos o texto plano com a letra w de modo que o número de caracteres seja múltiplo de m. Esta técnica é chamada padding e geralmente se escolhe uma letra com menor probabilidade de aparecer na linguagem em questão.

Agora, cada grupo de seis letras é rearranjado de acordo com a permutação p:

NATUHI | PDMREA | OEAIMN | OAOMCD | HWIWON

levando ao texto criptografado

NATUHIPDMREAOEAIMNOAOMCDHWIWON

Para decriptá-lo o processo é semelhante, utilizando neste caso a permutação inversa p-1.

Um outro algoritmo de transposição poderia consistir em colocar o texto em uma tabela com n colunas, e dar como texto criptografado os símbolos de uma coluna - ordenados de cima para baixo - concatenados com os de outra, etc. A chave K seria o número n junto com a ordem em que se deve ler as colunas.

Suponhamos que queremos cifrar o texto "cifra por transposição" com n=5 e a permutação {3, 2, 5, 1, 4} como chave. Colocaríamos o texto em uma tabela com 5 colunas, rearrumaríamos suas colunas e teríamos o seguinte resultado para o texto criptografado: "FIACRRORPTSNOAPÇIOSA"

1	2	3	4	5
c	i	f	r	a
p	o	r	t	r
a	n	s	p	o
s	i	ç	â	o

3	2	5	1	4
f	i	a	c	r
r	o	r	p	t
s	n	o	a	p
ç	i	o	s	â

Figura 4.2-Exemplo mostrando o funcionamento de um algoritmo de transposição

5 Data Encription Standard

5.1 História

Em 15 de maio de 1973 a NBS(National Bureau of Standards) realizou um anúncio público solicitando propostas de algoritmos para proteção de dados durante sua transmissão e armazenamento.

Não houve resposta até 6 de agosto de 1974 quando a IBM apresentou o criptossistema desenvolvido chamado LUCIFER. A NSA (National Security Agency) e a NBS avaliaram o algoritomo e em 15 de julho de 1977 adotou-se uma modificação no LUCIFER e passou a chamar-se DES (Data Encription Standard).
A modificação efetuada pela NSA reduziu o tamanho original da chave de 128 bits para 64 bits.

5.2 Características

•DES é um algoritmo simétrico (a chave para cifrar e decifrar é a mesma), com chave de 64 bits, dos quais 8 são bits de paridade (fazendo com que a chave real tenha 56 bits. DES opera cifrando blocos de 64 bits.)

•DES foi adotado muito rapidamente em todo tipo de aplicação, como por exemplo para comunicações telefônicas, como padrão de segurança para os bancos.

•DES atua sobre bits, ou números binários (0 e 1). Cada grupo de 4 bits é um número hexadecimal (base 16). Por exemplo, 1010 em binário corresponde a A em hexadecimal.

•DES atua cifrando grupos de 64 bits binários (que é o mesmo que 16 números hexadecimais). Para isso utiliza uma chave de 64 bits. Se o bloco a ser cifrado é menor do que 64 bits devemos completar com zeros

•DES cifra em blocos, isto quer dizer que com um bloco de 64 bits (texto plano) e uma chave de 64 bits obtemos um novo bloco de 64 bits (texto cifrado).

•DES fornece uma permutação do texto plano original, ou seja uma das 2^{64} permutações possíveis.

5.3 Funcionamento do algoritmo

5.3.1 Obtenção das chaves:

O usuário fornece uma chave de 64 bits, o oitavo bit de cada byte é o de paridade e por tanto é descartado deixando a chave com 56 bits.

1. Efetua-se uma permutação chamada PC-1 sobre a chave, de acordo com a seguinte tabela.

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

Tabela 5.1-Tabela de permutação PC-1

Isto que dizer que o primeiro bit da nova chave permutada é o 57ºbit da chave original, o segundo bit da chave permutada é o 49º da original e assim sucessivamente até chegar ao 56º bit da chave permutada que é o 4º bit da original. Note que o 8º,16º,24º… bits não

estão na tabela, ou seja esta permutação já retira os bits de paridade, transformando a chave de 64 bits em uma de 56 bits.

2. Divide-se a chave em 2 partes. Os primeiros 28 bits da chave passam a se chamar C[0] e os últimos 28 D[0].

3. Calcula-se 16 sub-chaves as quais chamaremos de K[i] (i=1..16):

   a. Iniciamos a iteração i=1, com C[i-1] e D[i-1] (que corresponde ao C[0] e D[0] calculados a partir da tabela PC-1). Realiza-se um deslocamento circular para a esquerda de n bits sobre C[i-1] e D[i-1]. Quando n=1 retiramos 1 bit da esquerda e colocamo-lo à direita, repetindo duas vezes este processo quando n=2. Note que n pode valer 1 ou 2 dependendo da iteração a ser feita, veja abaixo.

   b. Iteração # 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
   Valor de n 1 1 2 2 2 2 2 2 1 2 2 2 2 2 2 1

   c. Uma vez realizado o deslocamento correspondente à iteração (ou sub-chave) que estamos calculando, obtemos C[i] e D[i].

   d. Unimos novamente C[I] e D[i] e realizamos uma nova permutação seguindo a tabela abaixo:

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

Tabela 5.2-Tabela de permutação PC-2

   e. Obtendo a sub-chave K[i]. Voltamos ao passo (a) com os recém calculados C[i] e D[i] para calcular os novos C[i+1] e D[i+1].

   f. Quando terminarmos as 16 iterações teremos as 16 sub-chaves K[i] necessárias ao algoritmo

5.3.2 Cifrando um bloco:

Primeiramente separamos do texto plano um bloco de 64 bits. Se o bloco for menor do que 64 bits (o que geralmente acontece ao último bloco), preenchemo-lo com zeros.

1. Realiza-se uma permutação inicial (IP) do bloco completo segundo a seguinte tabela:

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

Tabela 5.3-Tabela de permutação IP

Isto quer dizer que o primeiro bit do bloco permutado é o bit 58 do bloco original, o 2º bit do bloco permutado é o 50º do original, até chegar ao ultimo bit do bloco permutado, que é o 7º do original.

2. Dividimos o bloco de 64 bits resultante em duas metades L[0] e R[0], cada uma com 32 bits.

3. Feito isso deve-se processar o bloco de dados L[0] R[0] com as 16 sub-chaves como sera descrito abaixo, são 16 passos. Começamos em i=1.

   a. Expandimos o bloco R[i-1], com 32 bits em um bloco de 48 bits, através da tabela abaixo e do deslocamento de bits afim de obter E(R[i-1]):

Expansão E:

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

Tabela 5.4-Tabela de expansão E

Assim como com as outras tabelas, o primeiro bit do novo bloco corresponde ao 32º do antigo, assim sucessivamente (podemos ver que alguns bits estão repetidos é assim que ocorre a expansão).

b. Realiza-se um OU exclusive (XOR) do bloco expandido com a chave i: E(R[i-1])+ XOR K[i].

c. O resultado da operação anterior é repartido em 8 blocos de 6 bits. Chamamos de B[1] ao bloco de bits 1-6, B[2] ao bloco de bits 7-12 e assim até chegar ao B[8] com os bits 43-48.

d. Realiza-se agora uma substituição dos 8 blocos utilizando 8 caixas de substituição chamadas S-Boxes as quais nos referimos por S[1], S[2], …, S[8]. Começando com o bloco 1 (j=1).

i. Devemos lembrar que cada B[j] tem 6 bits. Separamos os bits 1 e 6 para formarmos com ele um novo número de 2 bits chamado m.

ii. Separamos agora os bits do meio de B[j] (bits 2,3,4 e 5) e formamos um número de 4 bits ao qual chamaremos n.

iii. O novo valor para B[j] sera obtido buscando-se na S-Box correspondente a fila m e a coluna n. O novo número não é de 6 bits, mas sim de 4 bits.

iv. Por exemplo: A S-Box[1] é a seguinte:

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

Tabela 5.5- Caixa - S[1]

Teremos 4 filas e 16 colunas. O valor de m de 2 bits nos dá a fila (00, 01, 10, 11) e o valor de n nos dá a coluna. Por exemplo, para um B[1] de 01110 teremos m=00 (1º e 6º bits) e um n=1111 (bits 2,3,4 e 5) o que nos dá a fila 0 (00) e a

última coluna (1111). Por tanto o novo valor de B[1] é 7, que escrito em binário com 4 bits torna-se 0111.

v. Realizamos a mesma operação com os 7 restantes B[j], uilizando as seguintes S-Boxes:

S[2]

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

Tabela 5.6-Caixa-S [2]

S[3]

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

Tabela 5.7-Caixa-S [3]

S[4]

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

Tabela 5.8-Caixa-S [4]

S[5]

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

Tabela 5.9-Caixa-S [5]

S[6]

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

Tabela 5.10-Caixa-S [6]

S[7]

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

Tabela 5.11-Caixa-S [7]

S[8]

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

Tabela 5.12-Caixa-S [8]

E finalmente obtemos novos valores para todas as B[1]..B[8].

e. Devemos agora permutar a concatenação dos recém obtidos B[1]..B[8] segundo a seguinte tabela de permutação P.

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

Tabela 5.13-Tabela de permutação P

f. Obteremos R[i] fazendo um XOR de L[i-1] com o resultado obtido da permutação anterior. De forma resumida podemos dizer que R[i]= L[i-1] xor P(S[1](B[1])...S[8](B[8])). Sendo B[j] um bloco de 6 bits resultante de E(R[i-1]) xor K[i].

g. Obteremos L[i] fazendo L[i]=R[i-1].

h. Com isto acabamos de processar um bloco com a mesma sub-chave. Voltamos em (a) para uma nova iteração com um novo i (+1) e seus L[i] e R[i] correspondents até processarmos o bloco com as 16 subchaves.

4. Depois de termos processado o bloco com as 16 subchaves e obtido R[16] e L[16]. Invertemos o bloco pondo primeiro o bloco da direita e depois o da esquerda: R[16]L[16], e a este novo bloco invertido aplicamos a permutação final ($IP^{\angle 1}$)

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

Tabela 5.14-Tabela de permutação IP^{-1}

5. Conseguimos assim, um bloco de 64 bits cifrado como algoritmo DES.

5.3.3 Esquema gráfico de funcionamento

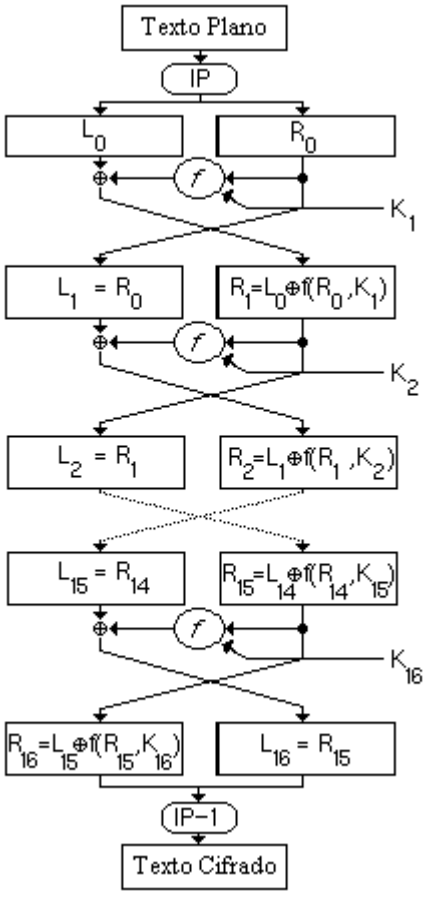


Tabela 5.15-Esquema gráfico do funcionamento do algoritmo DES

5.3.4 Decifrando um bloco

Devemos apenas repetir a operação de cifragem, mas tomando o cuidado de utilizar as chaves de forma inversa, quer dizer aplicando primeiramente K[16], depois k[15], até chegar a K[1] nos passos (a) até (h).

5.4 Implementação

5.4.1 Requisitos de Software

• Microsoft Windows 95, Windows 98 (original and Second Edition), Windows Millennium Edition, Windows NT 4.0 (com Service Pack 5 para correção Y2K ou Service Pack 6a) ou Windows 2000

• MATLAB 6.0 (R12) ou 6.1 (R12.1)

5.4.2 Requisitos de Hardware

• Pentium, Pentium Pro, Pentium II, Pentium III, Pentium IV** ou AMD Athlon

• 64 MB RAM (mínimo), 128 MB RAM (recomendadop) adaptador gráfico de 8 bits e monitor (para 256 cores simultâneas)

5.4.3 Descrição dos módulos principais

O DES é composto de vários módulos, os quais devem estar no diretório de trabalho do Matlab.

Os módulos principais são: GeraChavesDES e CifraBlocoDES

GeraChavesDES:

Este é o responsável pela geração das 16 sub-chaves necessárias ao algoritmo, K1 a K16 com 48 bits cada.

Sintaxe:

```
>>Chaves_Ki = GeraChavesDES(Chave_string);
```

Onde:

Chave_string é uma string no formato hexadecimal com o valor da chave de entrada, esta deve ter 64 bits. Exemplo:

>> Chave_string = '0123456789abcdef';

Chaves_Ki é uma matriz que armazena todas as 16 sub-chaves, onde a primeira chave é representada por Chaves_Ki(1,:), a segunda por Chaves_Ki(2,:), e assim sucessivamente.

CifraBlocoDES

Este módulo é o responsável pelo processo de cifragem propriamente dito. Ele trabalha cifrando um bloco de 64 bits.

Sintaxe:

>> [BlocoCifrado] = CifraBlocoDES(Bloco_64bits, Chaves_Ki)

Onde:

Chaves_Ki é o resultado do módulo anterior, ou seja, as 16 sub-chaves.

Bloco_64bits é o bloco de dados a ser cifrado, ele deve ser uma string no formato hexadecimal. Exemplo:

>> Bloco_64bits = 'abcdef0123456789';

BlocoCifrado é o resultado da cifragem do Bloco_64bits através do algoritmo DES.

DecifraBlocoDES

Este é o módulo responsável pelo processo inverso ao de cifragem.

Sintaxe:

>>[BlocoDecifrado] = DecifraBlocoDES(BlocoCifrado, Chaves_Ki, FormatoIn, FormatoOut)

Onde:

Chaves_Ki é o resultado do módulo GeraChavesDES, conforme explicado anteriormente.

BlocoCifrado é o bloco de 64 bits criptografado a partir do algoritmo DES.

BlocoDecifrado é o bloco cifrado novamente em texto plano, ou seja, decriptografado.

FormatoIn representa o modo em que a entrada sera apresentada (0 para hexadecimal ; 1 para binario)

FormatoOut representa o modo em que a saida sera apresentada (0 para ascii ; 1 para binario).

Outros Módulos:

Os módulos acima (GeraChavesDES, CifraBlocoDES e DecifraBlocoDES) são formados de diversos módulos necessários a sua implementação, na próxima seção veremos em detalhes sua composição.

5.4.4 Descrição dos módulos auxiliares

GeraChavesDES

o RotLeft - Implementa a rotação para a esquerda de uma ou duas posições conforme o número da iteração. (Ver Passo 1 item 3a)

Sintaxe: [C_out, D_out] = RotLeft(C_in, D_in, Iteracao)

o PC_1 - Implementa a permutação PC-1 (Ver Passo 1 item 1)

Sintaxe: [C_28bits, D_28bits] = PC_1(Chave_64bits)

CifraBlocoDES

- o Bin2Hex - Transforma uma string binaria em representacao hexadecimal.
  Exemplo: 00011100 = 0001 1100 = 1 12 = 1C
  Sintaxe: Resultado = Bin2Hex(String_Binaria), onde String_Binaria é da forma: String_Binaria = '00011100';
- o IP - Realiza a permutação inicial IP (Ver Passo 2 item 1)
  Sintaxe: [L0_32bits, R0_32bits] = IP(Bloco_64bits)
- o IP_1 - Calcula a permutação IP-1 (Ver Passo 2 item 4)
  Sintaxe: [BlocoFinal_64bits] = IP_1(R_32bits, L_32bits)

o Feistel - Este módulo calcula as iterações da rede de Feistel (Ver Passo 2 item 3) e é composto de dois outros módulos: FuncaoF e BinXor.

Sintaxe: [Lfim_32bits, Rfim_32bits] = Feistel(L_32bits, R_32bits, Chave_K, Num_Rounds)

o BinXor - Realiza a operaçao xor entre dois strings binarias de 16 bits

Exemplo: BinXor('1000', '0100') = 1000 xor 0100 = 1100

Sintaxe: BinXor(Num1, Num2)

o FuncaoF - Implementa a função F contida na rede de Feistel para o algoritmo DES (Ver figura).

Sintaxe: [F_32bits] = FuncaoF(R_32bits,K_48bits)

Por sua vez, este módulo é formado de outros três ainda não descritos: Expan, SBox e PermutP.

o Expan - Computa a Expansão E (Ver Passo 2 item 3a)

Sintaxe: [EA_48bits] = Expan(R_32bits)

o SBox - Realiza as substituições utilizando as S-Box (Ver Passo 2 item 3d)

Sintaxe: [C_32bits] = SBox(EAxorJ_48bits)

o PermutP - Realiza a permutação P (Ver Passo 2 item 3e)

Sintaxe: [F_32bits] = PermutP(C_32bits)

DecifraBlocoDES

Este módulo é idêntico ao CifraBlocoDES, sendo que antes de iniciar a cifragem (decifragem) ele inverte a posição da chave K, necessária para decifrar o texto (Ver Passo 3)

5.5 Laboratório Virtual

5.5.1 Criptografando com DES no Laboratório Virtual

Primeiramente geramos uma chave válida para o algoritmo, para tanto devemos selecionar "Obter chaves para", escolher o algoritmo "DES" e pressionar o botão "Enviar", como ilustrado na figura abaixo:

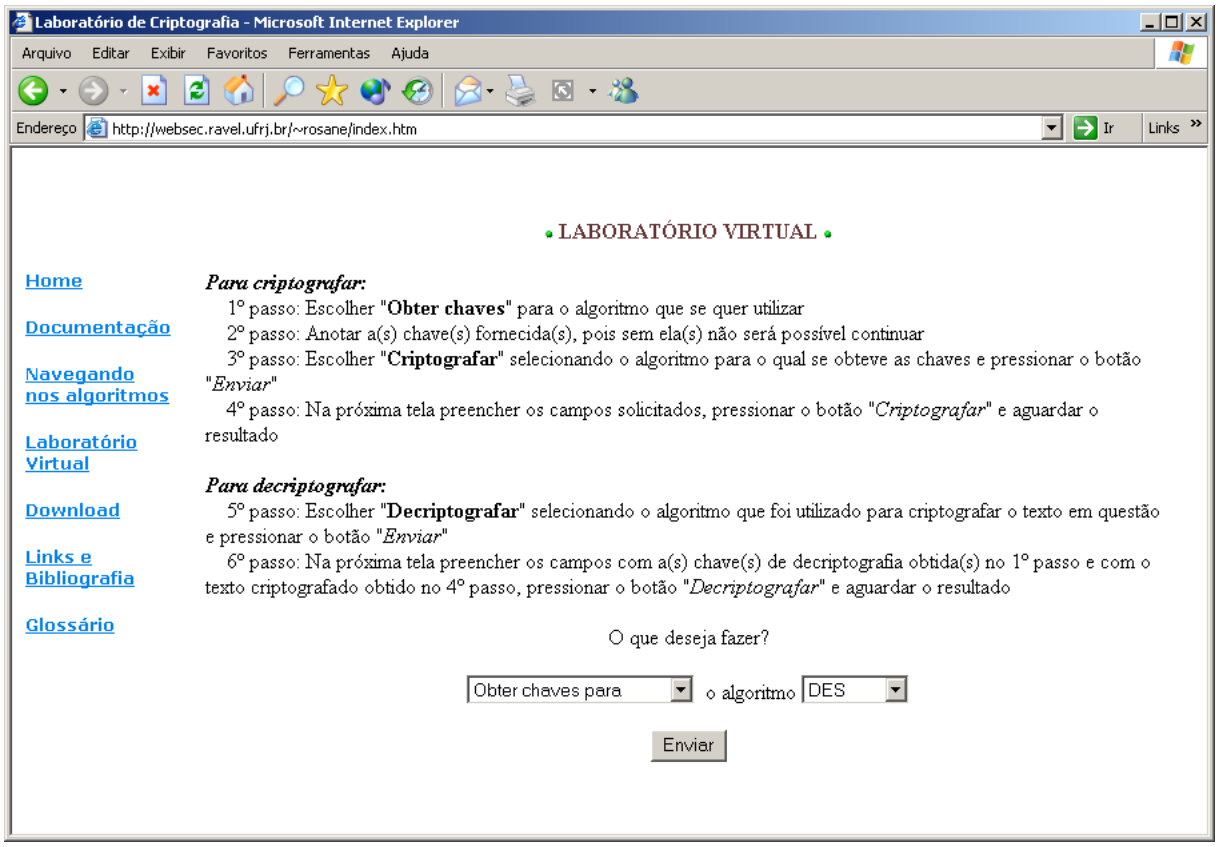


Figura 5.1- Obtendo uma chave válida

Após alguns segundo um Pop-Up como o mostrado na figura abaixo sugirá. Ele contém a chave que poderá ser utilizada no processo de cifragem.

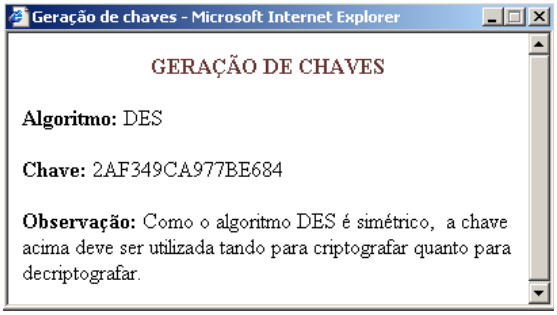


Figura 5.2- Resultado da geração da chave

Após obtida a chave, deve-se escolher "Criptografar utilizando" e selecionar o algoritmo "DES" como mostra a figura abaixo:

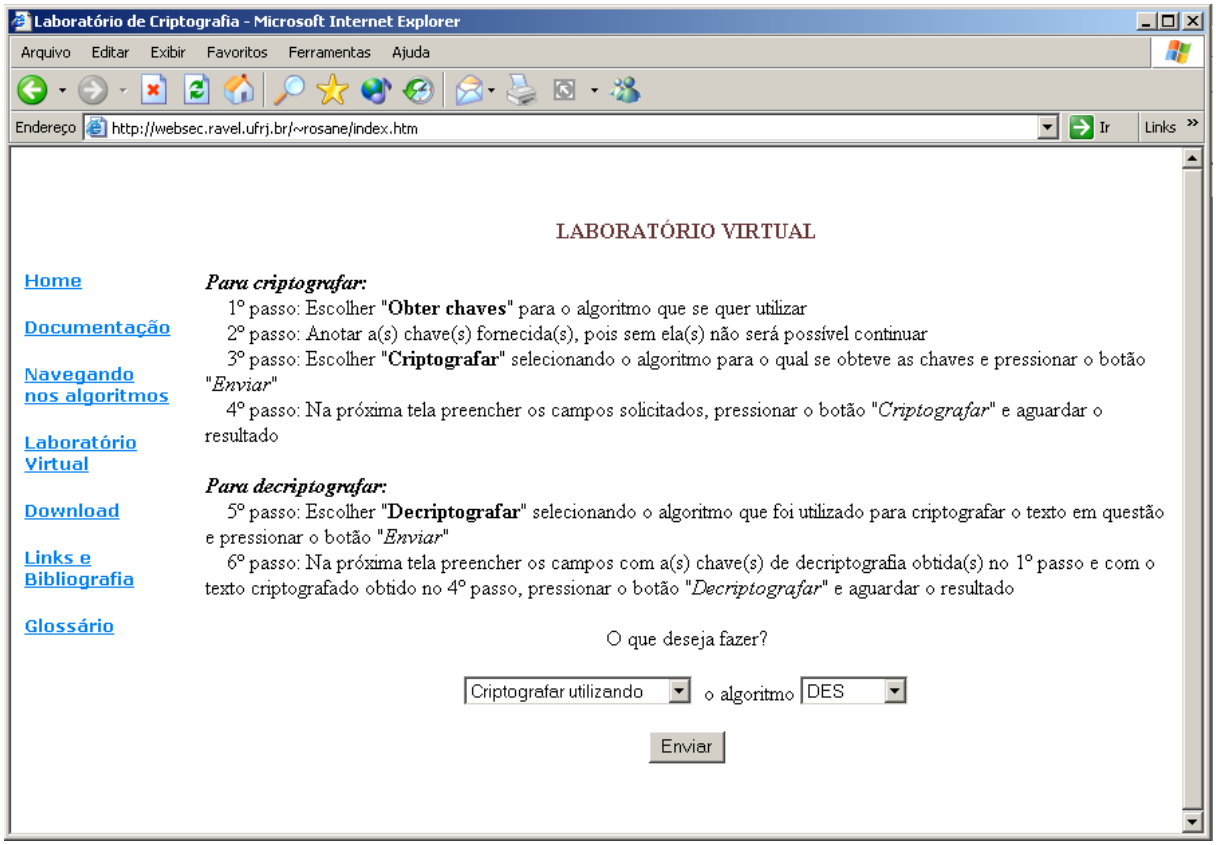


Figura 5.3- Optando pelo processo de cifragem

Após pressionar o botão "Enviar" uma nova tela será apresentada, nela será possível digitar a Chave Secreta (obtida no passo 1) e a informação a ser criptografada. Como pode ser visto na próxima figura.

A informação a ser criptografada pode conter quaisquer caracteres alfanuméricos inclusive caracteres de pontuação.

Após preencher estes campos deve-se pressionar o botão "Criptografar".

Caso algum erro seja cometido, pode-se utilizar o botão Limpar para remover todas as informações digitadas.

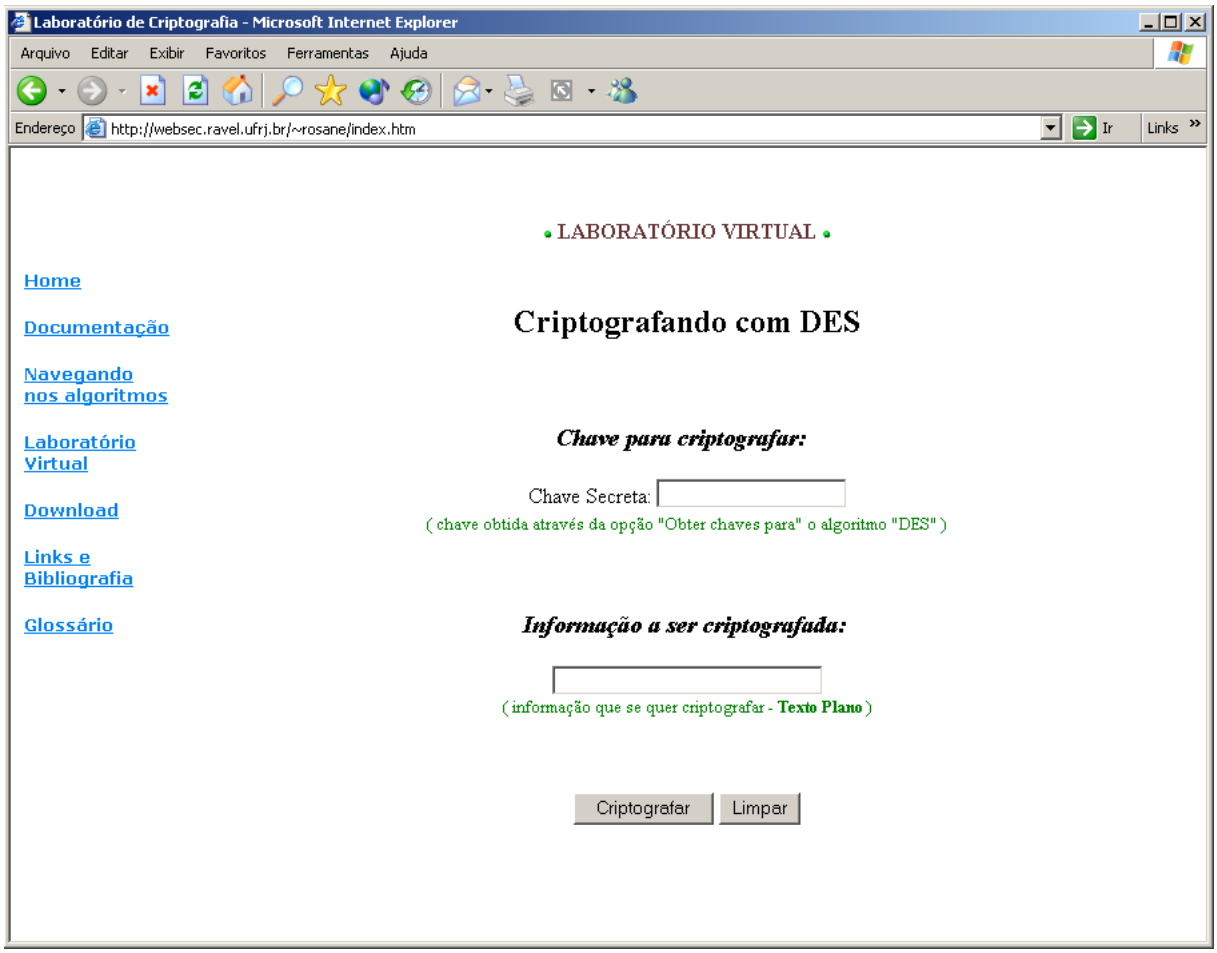


Figura 5.4- Inserindo os dados para cifrar

Após o pressionamento do botão "Criptografar" as informações serão processadas pelo Matlab. Este processo poderá levar alguns minutos, dependendo do tamanho da informação a ser criptografada. Após decorrido o tempo necessário, um Pop-Up será exibido para o usuário com o resultado da cifragem.

Neste Pop-Up, além da informação criptografada, serão exibidas também outras informações, como o texto plano, a chave utilizada bem como uma figura ilustrativa do processo de cifragem. Esta figura possui áreas clicáveis que revelará alguns detalhes do processo, como por exemplo os blocos em que foram divididos o texto plano antes de ser criptografado, as subchaves geradas no processo, etc.

A figura a seguir mostra um Pop-Up resultante no qual foram utilizadas as seguites informações:

Após o pressionamento do botão "Criptografar" as informações serão processadas pelo Matlab. Este processo poderá levar alguns minutos, dependendo do tamanho da informação a ser criptografada.

Após decorrido o tempo necessário, um Pop-Up será exibido para o usuário com o resultado da cifragem.

Neste Pop-Up, além da informação criptografada, serão exibidas também outras informações, como o texto plano, a chave utilizada bem como uma figura ilustrativa do processo de cifragem. Esta figura possui áreas clicáveis que revelará alguns detalhes do processo, como por exemplo os blocos em que foram divididos o texto plano antes de ser criptografado, as subchaves geradas no processo, etc.

A figura a seguir mostra um Pop-Up resultante no qual foram utilizadas as seguites informações:

Chave Secreta: 2AF349CA977BE684

Informação a ser Criptografada: Criptografando com DES.

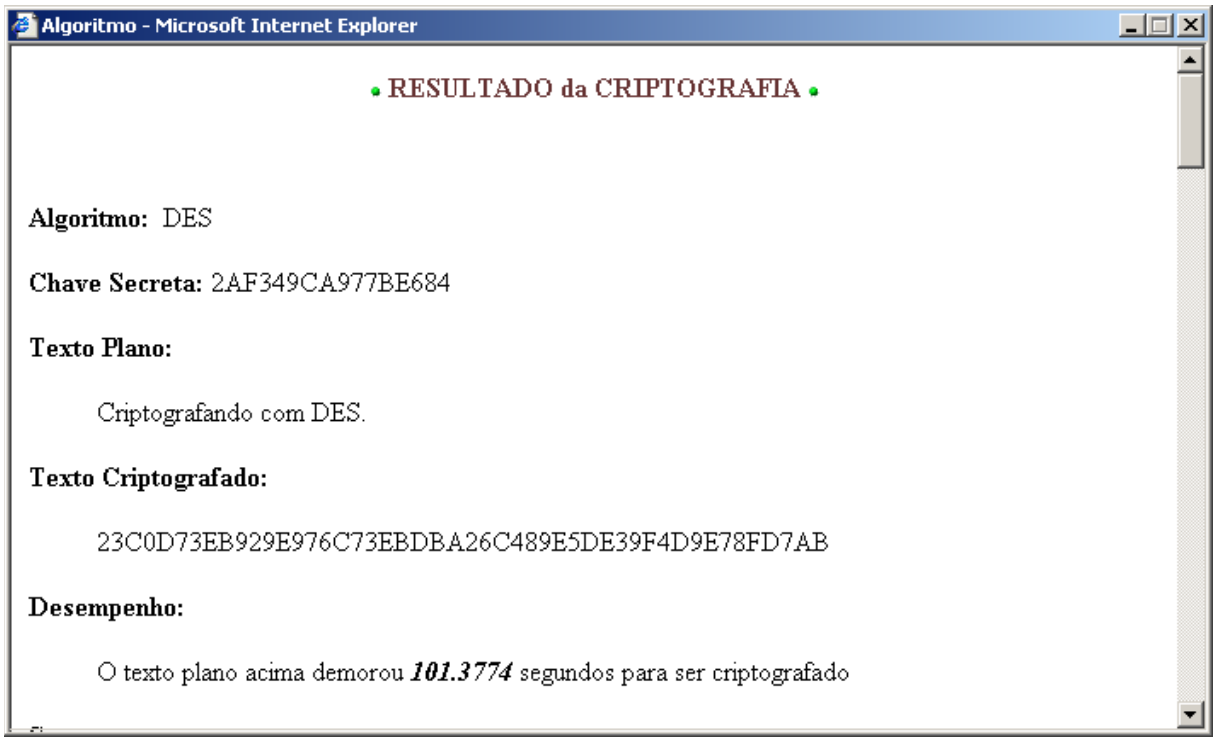


Figura 5.5- Resultado da cifragem

As demais informações do Pop-Up são apresentadas abaixo:

Segurança:

O método mais simples é tentar decriptar o bloco com todas as possíveis chaves. A informação do texto plano nos permitirá reconhecer a chave correta e parar a busca. Em media deveríamos tentar 36'028'797'018'963'968 (36 milhões de bilhões) de chaves. Levando-se em conta que um PC moderno pode checar entre um e dois mihões de chaves a cada segundo, isto representa um trabalho de cerca de 600 a 1200 anos para uma máquina simples.

Obviamente uma quantia significativa pode ser empregada para criar uma máquina mais potente, por exemplo: em 1993 foi construida uma máquina que levou em média 3,5 horas para quebrar o DES, mas ela custou 1 milhão de dólares!

Referência: Applied Cryptography, Second Edition: Protocols, Algorthms, and Source Code in C, by Bruce Schneier

Detalhes da Cifragem:

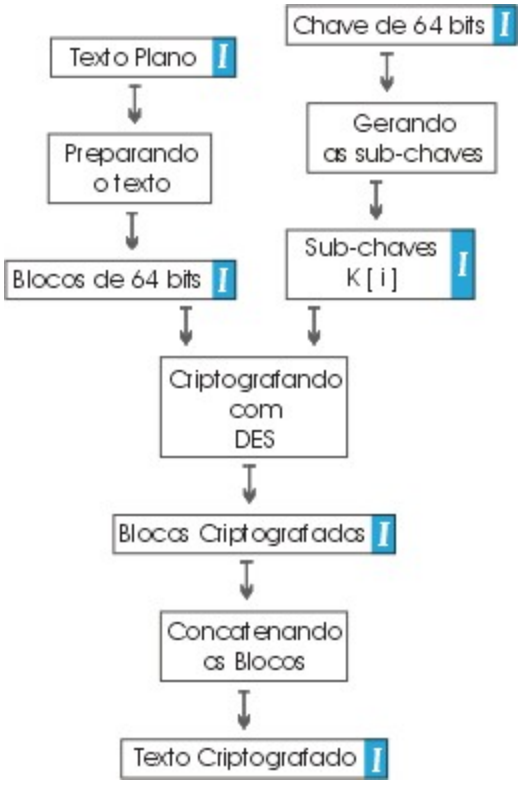


Figura 5.6- Diagrama da cifragem

Texto Plano:

Criptografando com DES

voltar ao diagrama

Blocos de 64 bits:

43726970746F6772
6166616E646F2063
6F6D204445530000

voltar ao diagrama

Chave Secreta:

2AF349CA977BE684

voltar ao diagrama

Subchaves K:

CB5B8A3296A7
6722EB2A71CE
BBDC9424F1A7
5C2BDAE60CE3
36F43DCE8B5B
CF0D4617D758
6AEABD599560
9DB52AC8EC2C
D38BCE11DCE9
38F2B38ABC31
B51D6EAB6F34
E262D5394B92
1DDF34D54017
C631FBC722CC
BFC665B0B3CD
BD9D1B7F240D

voltar ao diagrama

Blocos Criptografados:

23C0D73EB929E976
C73EBDBA26C489E5
8EBA8E4D88AEBF50

voltar ao diagrama

Texto Criptografado:

23C0D73EB929E976C73EBDBA26C489E58EBA8E4D88AEBF50

voltar ao diagrama

5.5.2 Decriptografando com DES no Laboratório Virtual

Como já possuímos a Chave Secreta devemos passar direto à decifragem, para tanto devemos escolher "Decriptografar utilizando" o algoritmo "DES", como mostra a figura abaixo:

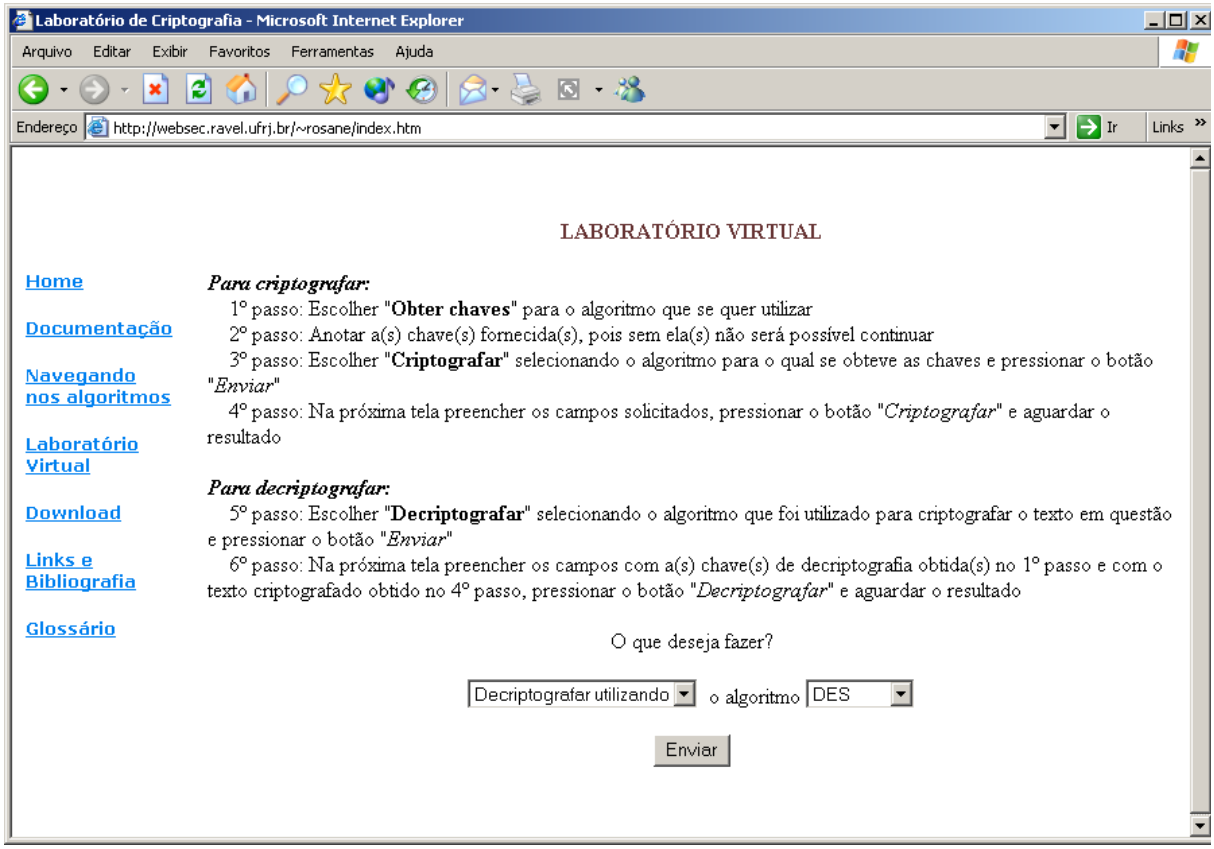


Figura 5.7- Optando pelo processo de decifragem

Semelhante ao processo de cifragem ao pressionarmos o botão Enviar uma nova tela se abrirá, como mostra a figura a seguir.

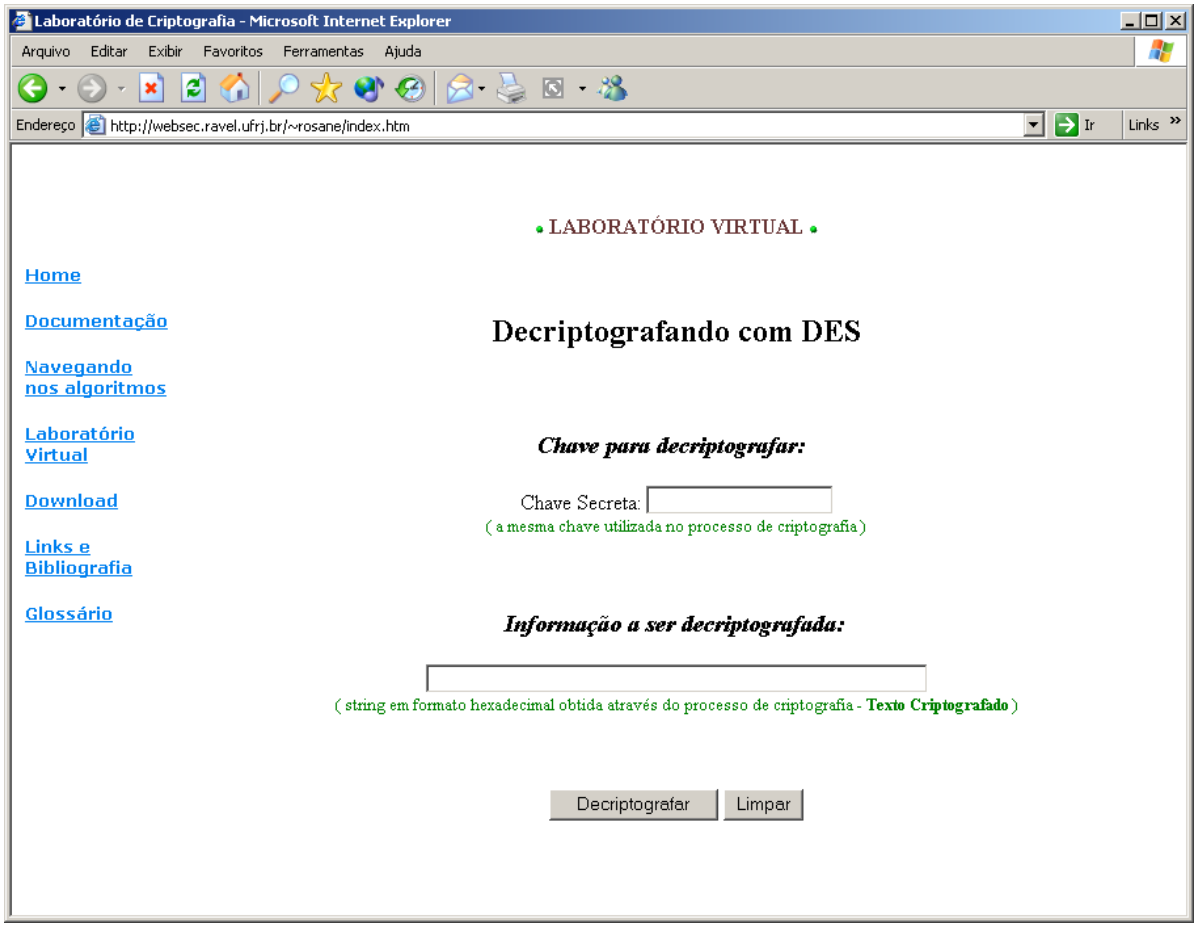


Figura 5.8- Inserindo dados para decifrar

Nesta tela será possível digitar a Chave a ser utilizada para decifrar a informação (note que como DES é um algoritmo simétrico a mesma chave utilizada para cifrar deve ser utilizada para decifrar - é a chamada "Chave Secreta").

Também será possível digitar a informação a ser decriptografada, ou seja, o texto ininteligível obtido através do processo de cifragem. Após o pressionamento do botão "Decriptografar" as informações serão processadas pelo Matlab. Este processo poderá levar alguns minutos, dependendo do tamanho da informação a ser decriptografada.

Após decorrido o tempo necessário, um Pop-Up será exibido para o usuário com o resultado da decifragem.

Neste Pop-Up, além da informação decriptografada, serão exibidas também outras informações, como o texto criptografado, a chave utilizada bem como uma figura ilustrativa do

processo de decifragem. Esta figura possui áreas clicáveis que revelará alguns detalhes do processo, como por exemplo os blocos em que foram divididos o texto criptografado antes de ser decriptografado, as subchaves geradas no processo, etc.

A figura a seguir mostra um Pop-Up resultante no qual foram utilizadas as seguites informações:

Chave Secreta: 2AF349CA977BE684

Informação a ser Decriptografada:

23C0D73EB929E976C73EBDBA26C489E5DE39F4D9E78FD7AB

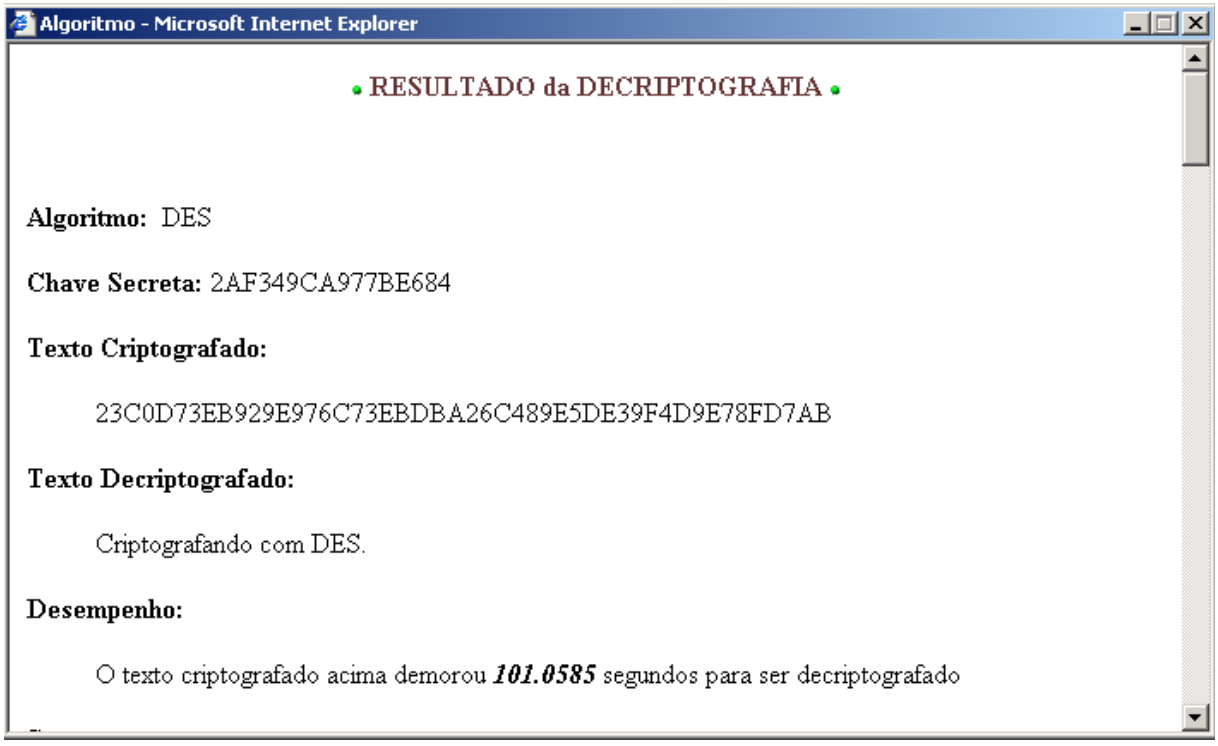


Figura 5.9- Resultado da decifragem

As demais informações do Pop-Up são apresentadas abaixo:

Segurança:

O método mais simples é tentar decriptar o bloco com todas as possíveis chaves. A informação do texto plano nos permitirá reconhecer a chave correta e parar a busca. Em media

deveríamos tentar 36'028'797'018'963'968 (36 milhões de bilhões) de chaves. Levando-se em conta que um PC moderno pode checar entre um e dois mihões de chaves a cada segundo, isto representa um trabalho de cerca de 600 a 1200 anos para uma máquina simples.

Obviamente uma quantia significativa pode ser empregada para criar uma máquina mais potente, por exemplo: em 1993 foi construida uma máquina que levou em média 3,5 horas para quebrar o DES, mas ela custou 1 milhão de dólares! .

Detalhes da Decifragem:

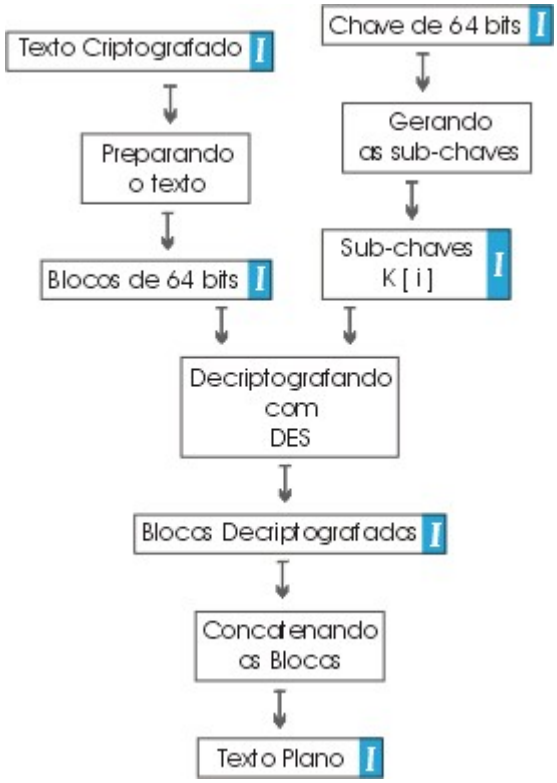


Figura 5.10- Diagrama da decifragem

Texto Criptografado:

23C0D73EB929E976C73EBDBA26C489E5DE39F4D9E78FD7AB

voltar ao diagrama

Blocos de 64 bits:

23C0D73EB929E976
C73EBDBA26C489E5
DE39F4D9E78FD7AB

voltar ao diagrama

Chave Secreta (Chave de 64 bits):

2AF349CA977BE684

voltar ao diagrama

Subchaves K:

CB5B8A3296A7
6722EB2A71CE
BBDC9424F1A7
5C2BDAE60CE3
36F43DCE8B5B
CF0D4617D758
6AEABD599560
9DB52AC8EC2C
D38BCE11DCE9
38F2B38ABC31
B51D6EAB6F34
E262D5394B92
1DDF34D54017
C631FBC722CC
BFC665B0B3CD
BD9D1B7F240D

voltar ao diagrama

Blocos Decriptografados:

43726970746F6772
6166616E646F2063
6F6D204445532E00

voltar ao diagrama

Texto Decriptografado:

Criptografando com DES.

voltar ao diagrama

6 Triple - Data Encription Standard

6.1 Características

•O DES triplo é uma extensão do algoritmo DES utilizando 3 chaves.

•O DES triplo utiliza o sistema EDE para cifragem, ou seja, o texto plano é encriptado primeiro com a primeira chave, decriptado com a segunda e encriptado novamente com a terceira.

•O efeito geral é o de haver-se cifrado com uma chave de tamanho duplo e por tanto muitíssimo mais seguro.

6.2 Funcionamento do Algoritmo

6.2.1 Obtenção das chaves

As chaves são obtidas utilizando-se o mesmo método descrito no capítulo anterior (DES).

O processo é então repetido três vezes, uma para cada chave, gerando assim, três conjuntos de chaves distintos.

6.2.2 Cifrando um bloco

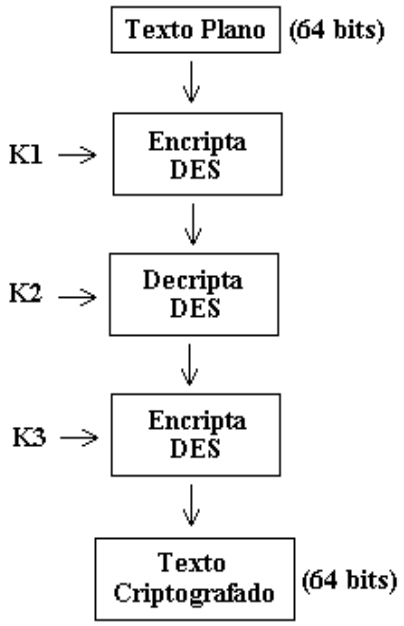


Figura 6.1-Esquema gráfico de cifragem utilizando o algoritmo Triplo-DES

6.2.3 decifrando um bloco

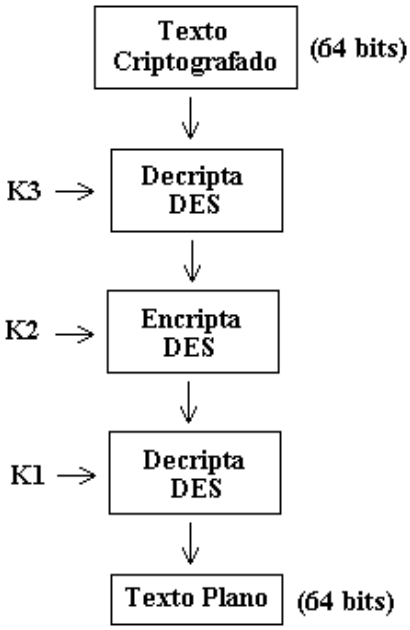


Figura 6.2-Esquema gráfico de decifragem utilizando o algoritmo Triplo-DES

6.3 Implementação

6.3.1 Requisitos de Software

•Microsoft Windows 95, Windows 98 (original and Second Edition), Windows Millennium Edition, Windows NT 4.0 (com Service Pack 5 para correção Y2K ou Service Pack 6a) ou Windows 2000

•MATLAB 6.0 (R12) ou 6.1 (R12.1)

6.3.2 Requisitos de Hardware

•Pentium, Pentium Pro, Pentium II, Pentium III, Pentium IV** ou AMD Athlon

•64 MB RAM (mínimo), 128 MB RAM (recomendadop) adaptador gráfico de 8 bits e monitor (para 256 cores simultâneas)

6.3.3 Descrição dos módulos principais

O triplo-DES é composto de vários módulos, os quais devem estar no diretório de trabalho do Matlab.

Os módulos principais são: GeraChavesDES, CifraBloco3DES e DecifraBloco3DES.

GeraChavesDES:

É o mesmo módulo utilizado no Algoritmo DES, já descrito anteriormente.

CifraBloco3DES

Este módulo é o responsável pelo processo de cifragem propriamente dito. Ele trabalha cifrando um bloco de 64 bits 3 vezes, cada uma com uma chave diferente.

Sintaxe:

>> [BlocoCifrado] = CifraBloco3DES(Bloco_64bits, Chave1_Ki, Chave2_Ki, Chave3_Ki)

Onde:

Chave1_Ki, *Chave2_Ki* e *Chave3_Ki* são os resultado do módulo anterior para cada uma das chaves a serem utilizadas.

Bloco_64bits é o bloco de dados a ser cifrado, ele deve ser uma string no formato hexadecimal. Exemplo:

>> Bloco_64bits = 'abcdef0123456789';

BlocoCifrado é o resultado da cifragem do Bloco_64bits através do algoritmo DES.

DecifraBlocoDES

Este é o módulo responsável pelo processo inverso ao de cifragem.

Sintaxe:

>>[BlocoDecifrado] = DecifraBloco3DES(BlocoCifrado, Chave1_Ki, Chave2_Ki, Chave3_Ki)

Onde:

Chave1_Ki, *Chave2_Ki* e *Chave3_Ki* são os resultado do módulo GeraChavesDES para as três chaves, conforme explicado anteriormente.

BlocoCifrado é o bloco de 64 bits criptografado a partir do algoritmo 3DES.

BlocoDecifrado é o bloco cifrado novamente em texto plano, ou seja, decriptografado.

Outros Módulos:

Os módulos acima (GeraChavesDES, CifraBlocoDES e DecifraBlocoDES) são formados de diversos módulos necessários a sua implementação, os quais já foram descritos anteriormente.

6.3.4 Descrição dos módulos auxiliares

GeraChavesDES

Já descrito no capítulo anterior.

CifraBloco3DES

É formado pelo CifraBlocoDES aplicado 3 vezes, cada uma delas com uma chave diferente.

>>BlocoCifrado1 = CifraBloco(Bloco_64bits, Chave1_Ki);

>>BlocoCifrado2 = DecifraBloco(BlocoCifrado1, Chave2_Ki);

>>BlocoCifrado = CifraBloco(BlocoCifrado2, Chave3_Ki);

DecifraBloco3DES

É formado pelo DecifraBlocoDES aplicado 3 vezes, cada uma delas com uma chave diferente, e utilizando a ordem inversa

>>BlocoDecifrado1 = DecifraBloco(BlocoCifrado, Chave3_Ki);

>>BlocoDecifrado2 = CifraBloco(BlocoDecifrado1, Chave2_Ki);

>>BlocoDecifrado = DecifraBloco(BlocoDecifrado2, Chave1_Ki);.

6.5 Laboratório Virtual

6.5.1 Criptografando com Triplo-DES no Laboratório Virtual

Primeiramente geramos uma chave válida para o algoritmo, para tanto devemos selecionar "Obter chaves para", escolher o algoritmo "Triplo-DES" e pressionar o botão "Enviar", como ilustrado na figura a seguir:

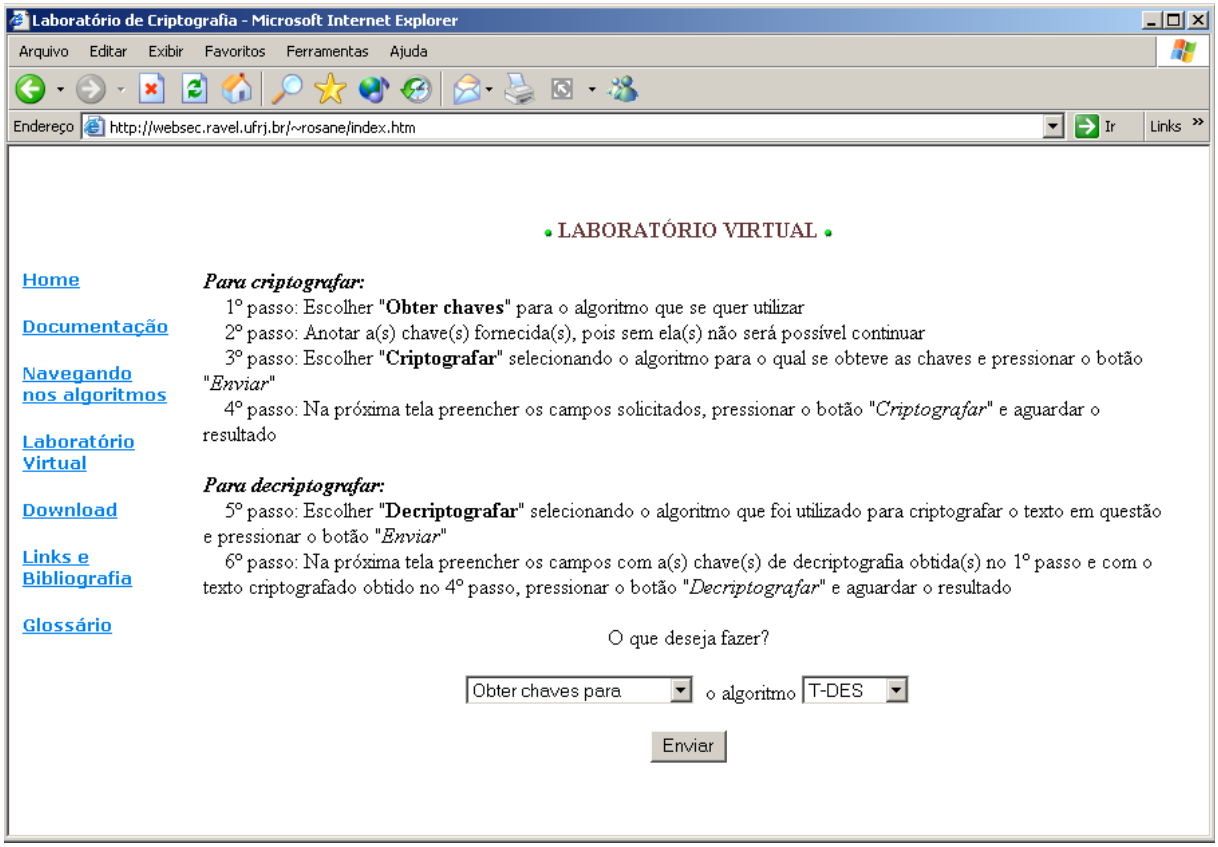


Figura 6.3- Obtendo uma chave válida

Após alguns segundo um Pop-Up como o mostrado na figura abaixo sugirá. Ele contém a chave que poderá ser utilizada no processo de cifragem.

Figura 6.4- Resultado da geração das chaves

Após obtidas as chaves, deve-se escolher "Criptografar utilizando" e selecionar o algoritmo "DES" como mostra a figura abaixo:

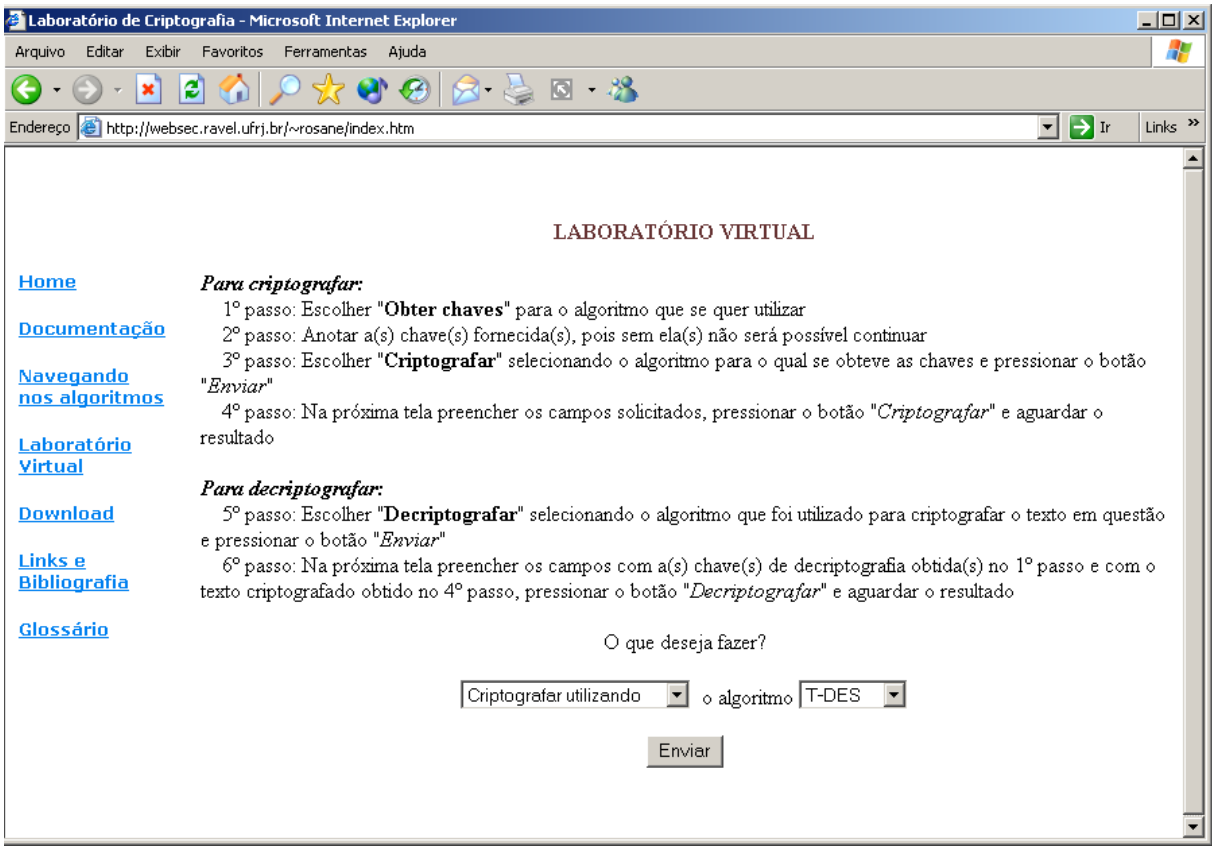


Figura 6.5- Optando pelo processo de cifragem

Após pressionar o botão "Enviar" uma nova tela será apresentada, nela será possível digitar as Chaves Secretas (obtidas no passo 1) e a informação a ser criptografada. Como pode ser visto na próxima figura.

A informação a ser criptografada pode conter quaisquer caracteres alfanuméricos inclusive caracteres de pontuação.

Após preencher estes campos deve-se pressionar o botão "Criptografar".

Caso algum erro seja cometido, pode-se utilizar o botão Limpar para remover todas as informações digitadas.

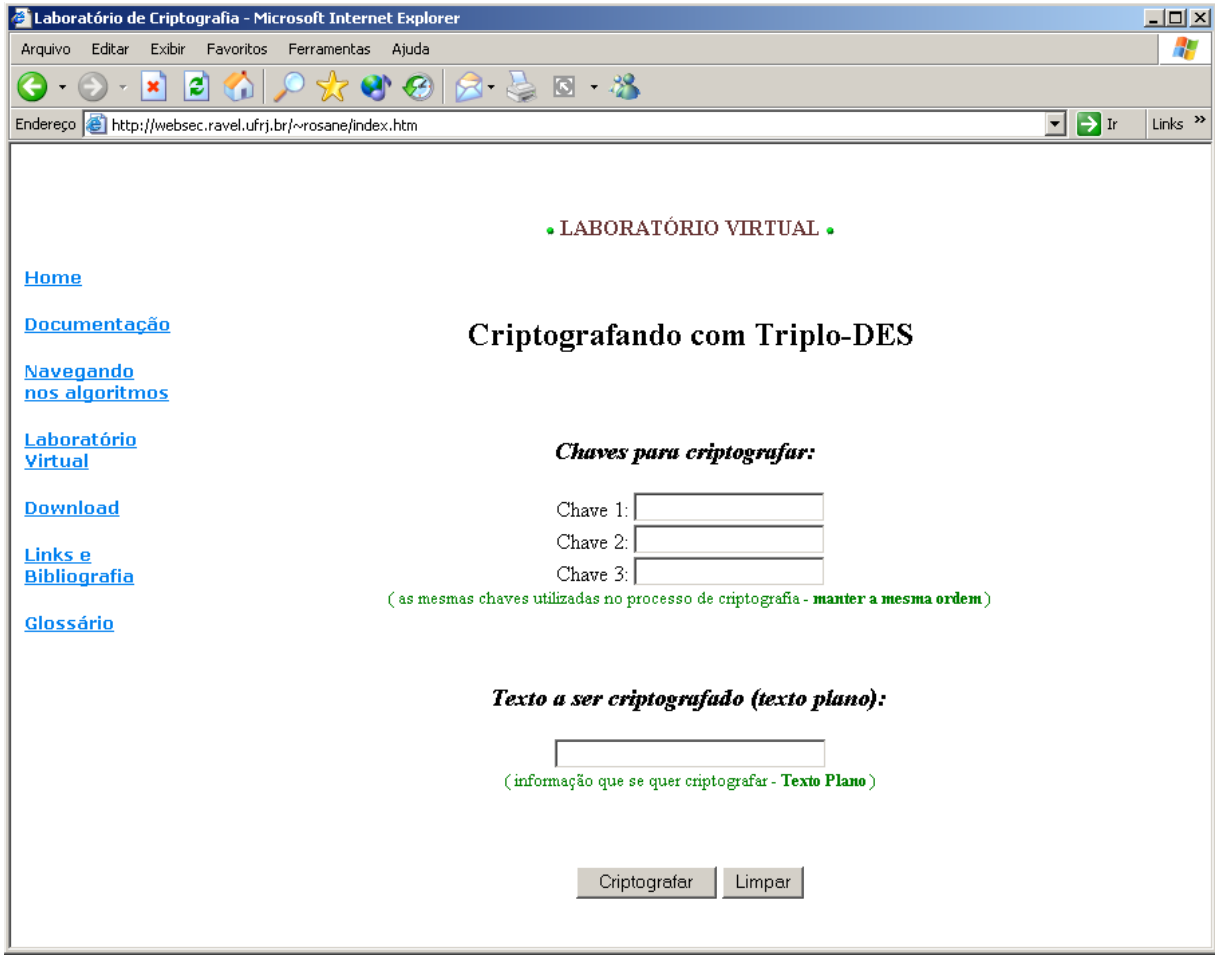


Figura 6.6- Inserindo os dados para cifrar

Após o pressionamento do botão "Criptografar" as informações serão processadas pelo Matlab. Este processo poderá levar alguns minutos, dependendo do tamanho da informação a ser criptografada. Após decorrido o tempo necessário, um Pop-Up será exibido para o usuário com o resultado da cifragem.

Neste Pop-Up, além da informação criptografada, serão exibidas também outras informações, como o texto plano, a chave utilizada bem como uma figura ilustrativa do processo de cifragem. Esta figura possui áreas clicáveis que revelará alguns detalhes do processo, como por exemplo os blocos em que foram divididos o texto plano antes de ser criptografado, as subchaves geradas no processo, etc.

A figura a seguir mostra um Pop-Up resultante no qual foram utilizadas as seguites informações:

Após o pressionamento do botão "Criptografar" as informações serão processadas pelo Matlab. Este processo poderá levar alguns minutos, dependendo do tamanho da informação a ser criptografada.

Após decorrido o tempo necessário, um Pop-Up será exibido para o usuário com o resultado da cifragem.

Neste Pop-Up, além da informação criptografada, serão exibidas também outras informações, como o texto plano, a chave utilizada bem como uma figura ilustrativa do processo de cifragem. Esta figura possui áreas clicáveis que revelará alguns detalhes do processo, como por exemplo os blocos em que foram divididos o texto plano antes de ser criptografado, as subchaves geradas no processo, etc.

A figura a seguir mostra um Pop-Up resultante no qual foram utilizadas as seguites informações:

Chave 1: 2AF349CA977BE684

Chave 2: 433A29E478469800

Chave 3: 218022C29AF6EF8B

Informação a ser Criptografada: Criptografia

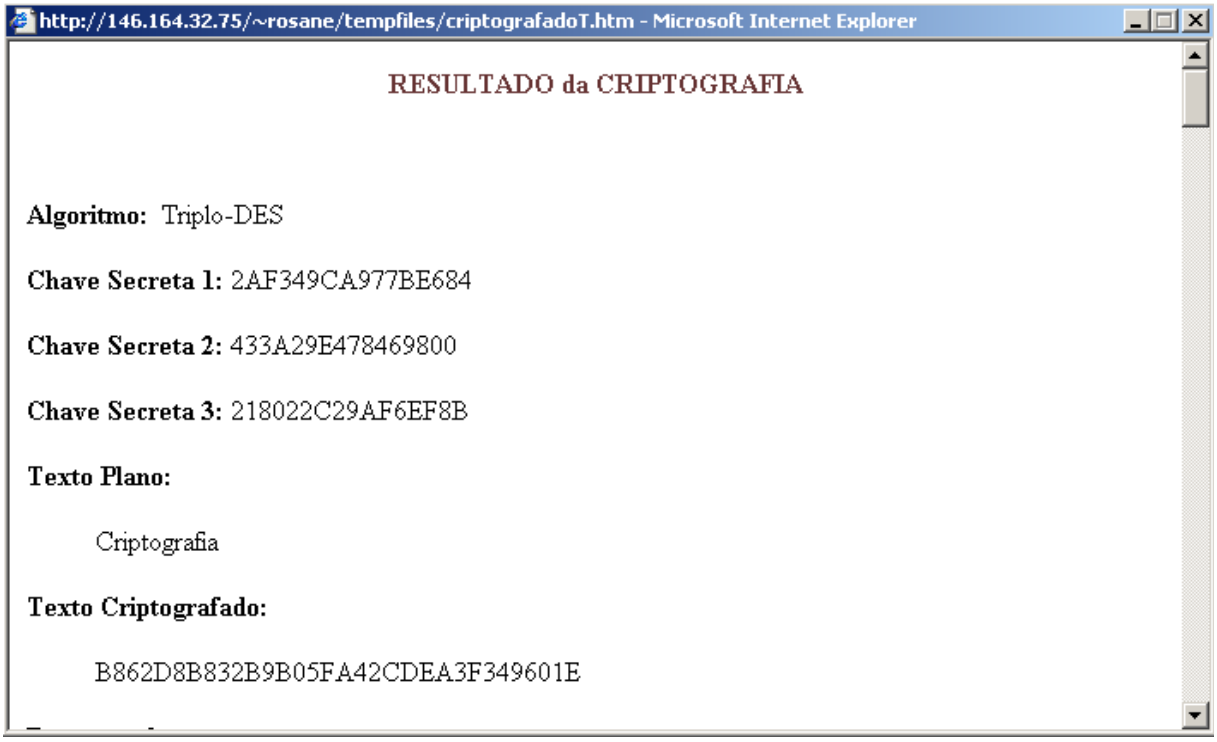


Figura 6.7- Resultado da cifragem

As demais informações do Pop-Up são apresentadas abaixo:

Desempenho:

O texto criptografado acima demorou 201.9786 segundos para ser decriptografado.

Segurança:

Para três chaves distintas com n-bits cada o melhor tipo de ataque utilizaria 22n passos e requereria 2n blocos de memória! Ou seja, seriam: 3.41 x 1038 passos e necessitaria de 18 x 1016 blocos de memória...

Referência: Applied Cryptography, Second Edition: Protocols, Algorthms, and Source Code in C, by Bruce Schneier

Detalhes da Cifragem:

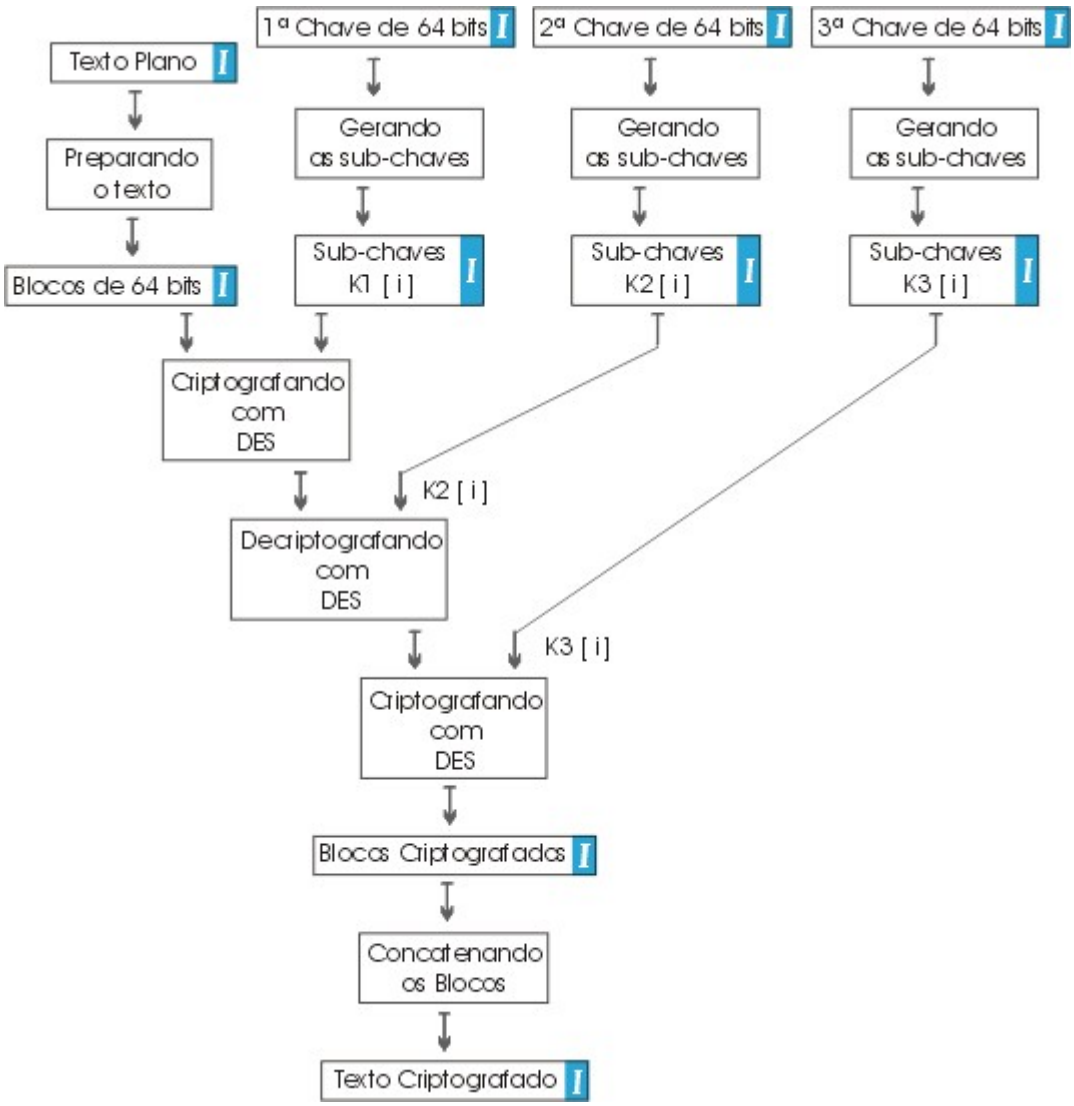


Figura 6.8- Diagrama da cifragem

Texto Plano:

Criptografia

voltar ao diagrama

Blocos de 64 bits:

43726970746F6772
6166696100000000

voltar ao diagrama

Chave Secreta 1:

2AF349CA977BE684

voltar ao diagrama

Chave Secreta 2:

433A29E478469800

voltar ao diagrama

Chave Secreta 3:

218022C29AF6EF8B

voltar ao diagrama

Subchaves K1:

CB5B8A3296A7
6722EB2A71CE
BBDC9424F1A7
5C2BDAE60CE3
36F43DCE8B5B
CF0D4617D758
6AEABD599560
9DB52AC8EC2C
D38BCE11DCE9
38F2B38ABC31
B51D6EAB6F34
E262D5394B92
1DDF34D54017
C631FBC722CC
BFC665B0B3CD
BD9D1B7F240D

voltar ao diagrama

Subchaves K2:

28B70C0B4488
B33484103EE2
5806F03C8831
D4D03C036C52
8683462DA110
2A5A27A14446
A931684C8286
8046F99444CD

3CC0D2613C04
36492A6820BA
AA211525580F
0D0E1D0610F2
4730B8858965
9E8CE0028ED0
DA6A0A598515
252AA7A4922C

voltar ao diagrama

Subchaves K3:

4B7B893A6903
6F00ED360927
9BC89C8609D2
1C2BBA45A351
B63C2D738448
CB2E4448950E
48FEBC0C74AC
D4B56A6858E1
D28BB66AE830
BCB233A14D1A
A7166E8D1212
EA52D4D54264
1CDB78108ACC
86717B90B495
AF4F452B26A1
3F151B2272A3

voltar ao diagrama

Blocos Criptografados:

B862D8B832B9B05F
A42CDEA3F349601E

voltar ao diagrama

Texto Criptografado:

B862D8B832B9B05FA42CDEA3F349601E

voltar ao diagrama

6.5.2 Decriptografando com Triplo-DES no Laboratório Virtual

Como já possuímos a Chave Secreta devemos passar direto à decifragem, para tanto devemos escolher "Decriptografar utilizando" o algoritmo "Triplo-DES", como mostra a figura abaixo:

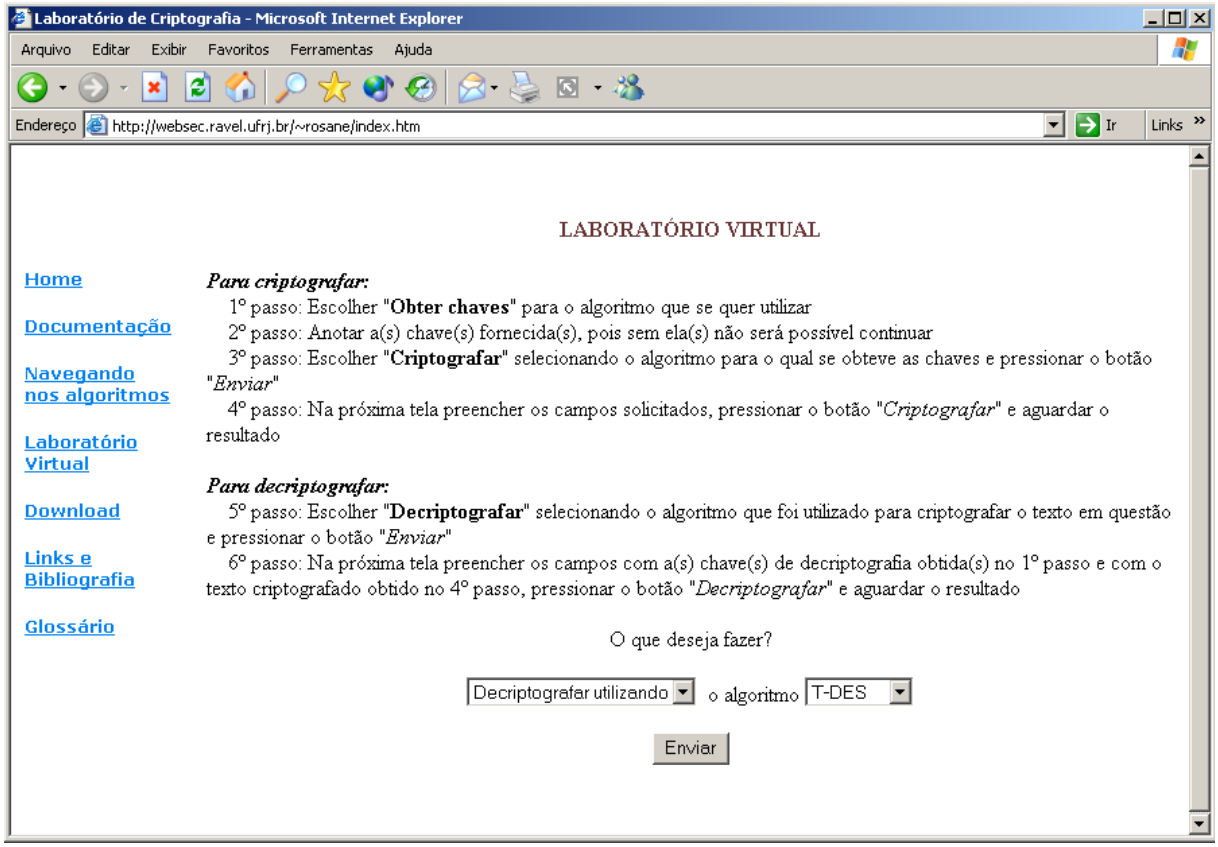


Figura 6.9- Optando pelo processo de decifragem

Semelhante ao processo de cifragem ao pressionarmos o botão Enviar uma nova tela se abrirá, como mostra a figura a seguir.

Nesta tela será possível digitar a Chave a ser utilizada para decifrar a informação (note que como Triplo-DES é um algoritmo simétrico as mesmas chaves utilizadas para cifrar devem ser utilizada para decifrar - com o cuidado de manter-se a mesma ordem utilizada no processo de cifragem).

Também será possível digitar a informação a ser decriptografada, ou seja, o texto ininteligível obtido através do processo de cifragem.

Semelhante ao processo de cifragem ao pressionarmos o botão Enviar uma nova tela se abrirá, como mostra a figura a seguir.

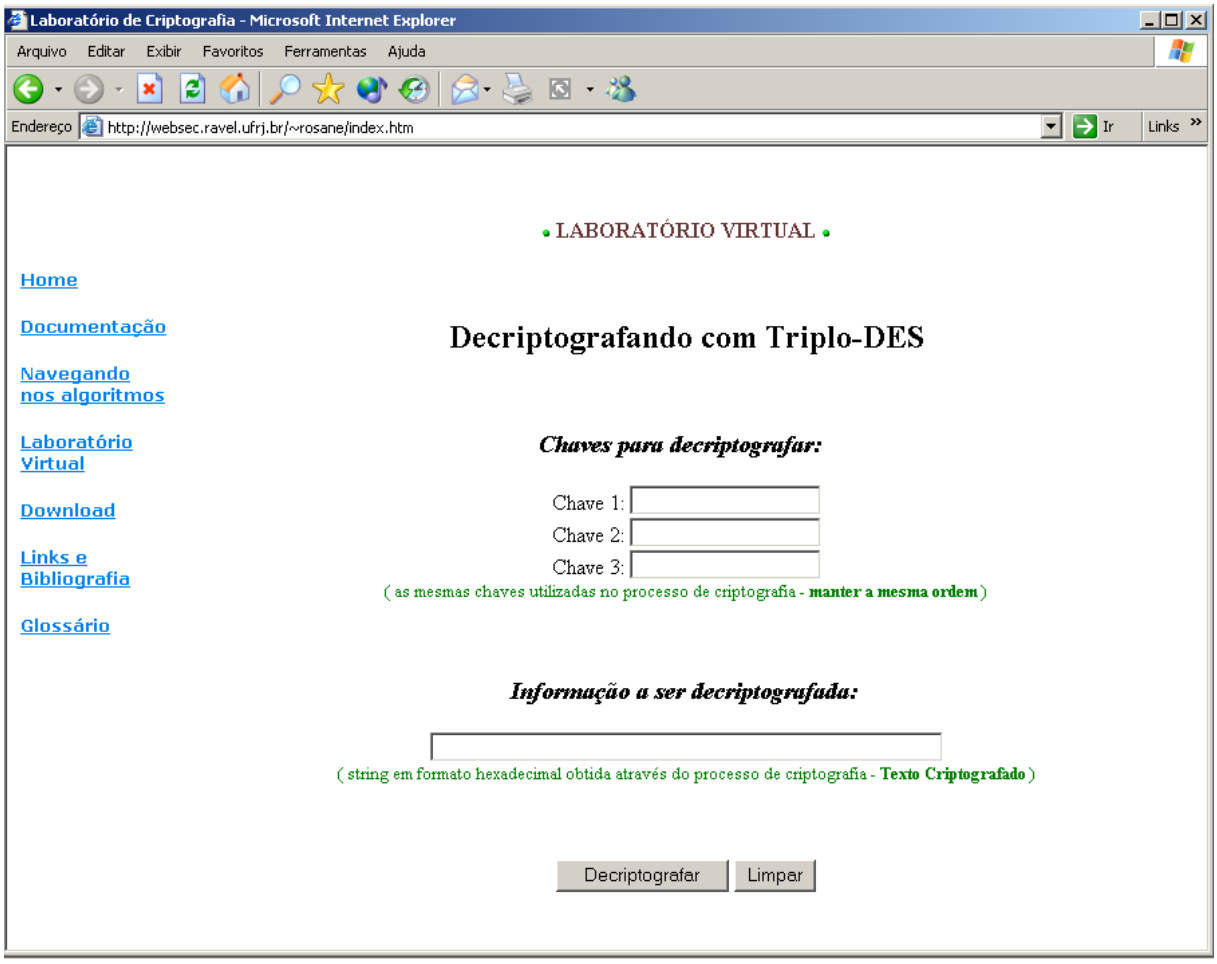


Figura 6.10- Inserindo dados para decifrar

Nesta tela será possível digitar a Chave a ser utilizada para decifrar a informação (note que como DES é um algoritmo simétrico a mesma chave utilizada para cifrar deve ser utilizada para decifrar - é a chamada "Chave Secreta").

Também será possível digitar a informação a ser decriptografada, ou seja, o texto ininteligível obtido através do processo de cifragem. Após o pressionamento do botão "Decriptografar" as informações serão processadas pelo Matlab. Este processo poderá levar alguns minutos, dependendo do tamanho da informação a ser decriptografada.

Após decorrido o tempo necessário, um Pop-Up será exibido para o usuário com o resultado da decifragem.

Neste Pop-Up, além da informação decriptografada, serão exibidas também outras informações, como o texto criptografado, a chave utilizada bem como uma figura ilustrativa do processo de decifragem. Esta figura possui áreas clicáveis que revelará alguns detalhes do processo, como por exemplo os blocos em que foram divididos o texto criptografado antes de ser decriptografado, as subchaves geradas no processo, etc.

A figura a seguir mostra um Pop-Up resultante no qual foram utilizadas as seguites informações:

Chave 1: 2AF349CA977BE684

Chave 2: 433A29E478469800

Chave 3: 218022C29AF6EF8B

Informação a ser Decriptografada:

B862D8B832B9B05FA42CDEA3F349601E

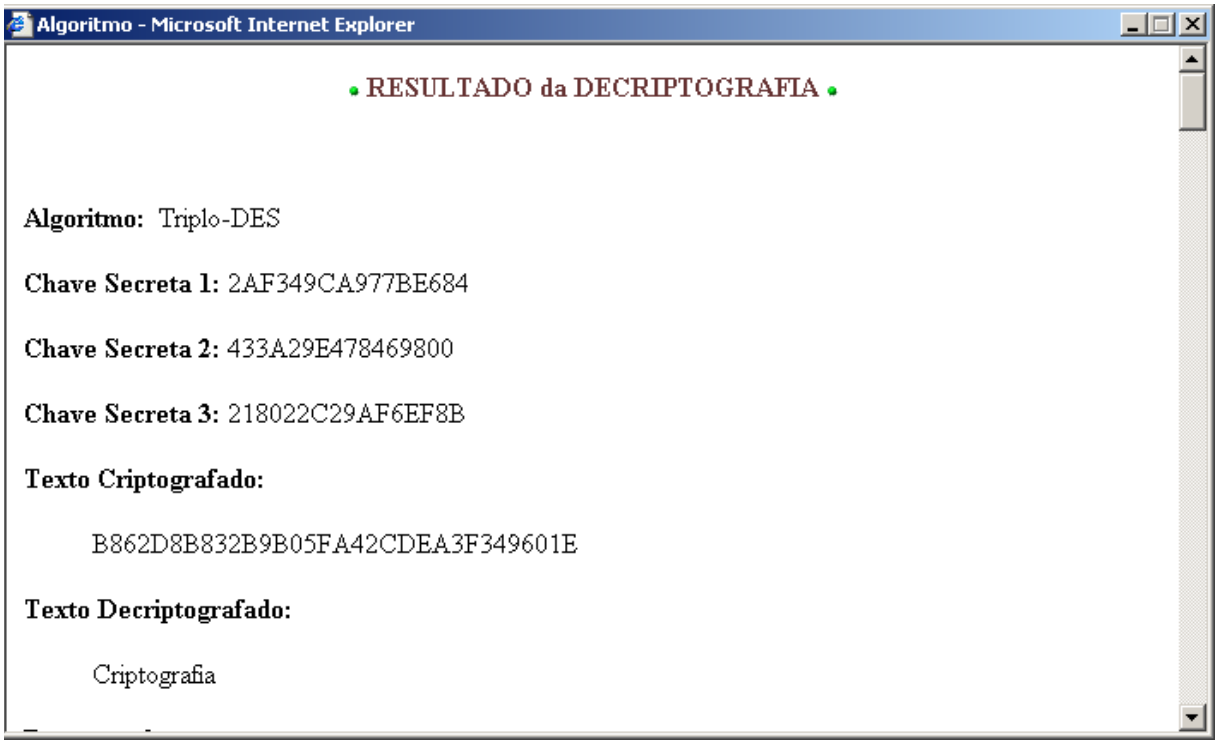


Figura 6.11- Resultado da decifragem

As demais informações do Pop-Up são apresentadas abaixo:

Desempenho:

O texto criptografado acima demorou 202.8868 segundos para ser decriptografado.

Segurança:

Para três chaves distintas com n-bits cada o melhor tipo de ataque utilizaria 22n passos e requereria 2n blocos de memória! Ou seja, seriam: 3.41 x 1038 passos e necessitaria de 18 x 1016 blocos de memória...

Referência: Applied Cryptography, Second Edition: Protocols, Algorthms, and Source Code in C, by Bruce Schneier

Detalhes da cifragem:

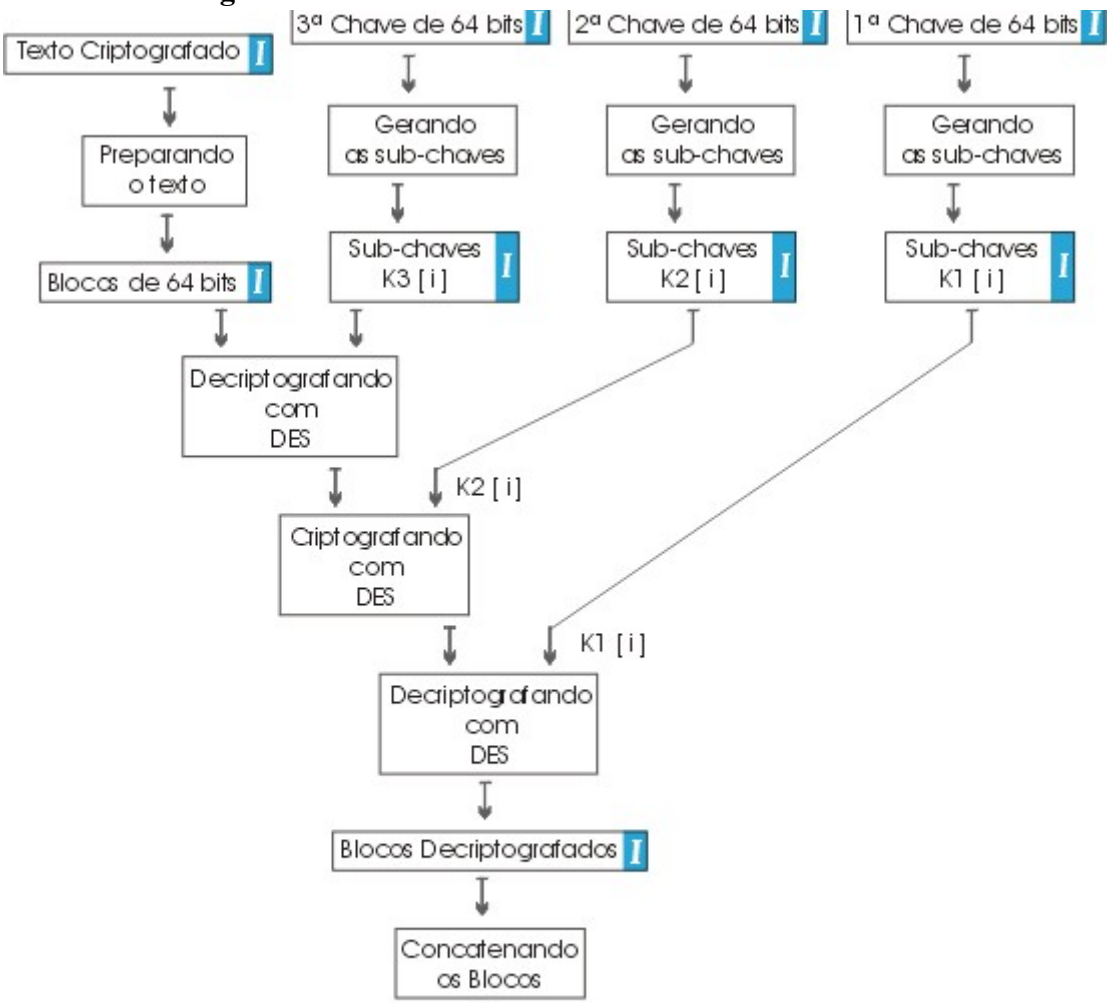


Figura 6.12- Diagrama da decifragem

Texto Plano:

B862D8B832B9B05FA42CDEA3F349601E

voltar ao diagrama

Blocos criptografados de 64 bits:

B862D8B832B9B05F
A42CDEA3F349601E

voltar ao diagrama

Chave Secreta 1:

2AF349CA977BE684

voltar ao diagrama

Chave Secreta 2:

433A29E478469800

voltar ao diagrama

Chave Secreta 3:

218022C29AF6EF8B

voltar ao diagrama

Subchaves K1:

CB5B8A3296A7
6722EB2A71CE
BBDC9424F1A7
5C2BDAE60CE3
36F43DCE8B5B
CF0D4617D758
6AEABD599560
9DB52AC8EC2C
D38BCE11DCE9
38F2B38ABC31
B51D6EAB6F34
E262D5394B92
1DDF34D54017

C631FBC722CC
BFC665B0B3CD
BD9D1B7F240D

voltar ao diagrama

Subchaves K2:

28B70C0B4488
B33484103EE2
5806F03C8831
D4D03C036C52
8683462DA110
2A5A27A14446
A931684C8286
8046F99444CD
3CC0D2613C04
36492A6820BA
AA211525580F
0D0E1D0610F2
4730B8858965
9E8CE0028ED0
DA6A0A598515
252AA7A4922C

voltar ao diagrama

Subchaves K3:

4B7B893A6903
6F00ED360927
9BC89C8609D2
1C2BBA45A351
B63C2D738448
CB2E4448950E
48FEBC0C74AC
D4B56A6858E1
D28BB66AE830
BCB233A14D1A
A7166E8D1212
EA52D4D54264
1CDB78108ACC
86717B90B495
AF4F452B26A1
3F151B2272A3

voltar ao diagrama

Blocos Decriptografados:

43726970746F6772
6166696100000000

voltar ao diagrama

Texto Decriptografado:

Criptografia

voltar ao diagrama

7 Blowfish

7.1 História

Blowfish é um algoritmo criptográfico de chave simétrica desenvolvido por Bruce Schneier. Consiste de um cifrador em blocos de 64 bits com chaves de tamanho variável (até 448 bits). O Blowfish ganhou uma grande aceitação no mercado sendo utilizado em diversas aplicações, dentre elas, o Nautilus e o PGPfone.O Blowfish utiliza uma função não inversível, caixas-S(S-boxes) e uma rede de Feistel. Este algoritmo nunca foi quebrado.

O Blowfish é um algorítmo criptográfico simétrico de blocos que pode ser usado em substituição ao DES ou IDEA. Ele toma uma chave de tamanho variável, de 32 a 448 bits, tormando-o ideal para aplicações tanto domésticas, quanto comerciais. O Blowfish foi desenvolvido em 1993 por Bruce Schneier como uma alternativa grátis mais rápida para os algorítmos criptográficos existentes. Desde então ele vem sendo analisado de forma considerável e está conquistando a aceitação do mercado como um algorítmo forte. O Blowfish não é patenteado, tem sua licença grátis e está a disposição para todos.

O paper original do Blowfish foi apresentado no "First Fast Software Encryption Workshop" ocorrido em Cambridge, UK, USA. Os resultados do "workshop" foram publicados por Springer-Verlanf, "Lecture Notes in Computer Science" #809, 1994). A edição de Abril de

1994 de "Dr. Dobb's Journal" também tratou expôs a proposta de Bruce Schneier. "Blowfish -- One Year Later" foi publicada em Setembro de 1995 em outra edição de "Dr. Dobb's Journal".

Muitos estudiosos em criptografia examinaram o Blowfish, entretanto, aida são poucos os resultados publicados. Serge Vaudenay examinou chaves fracas no Blowfish: existe uma casse de chaves que podem ser detectadas - mas não "quebradas" - no Blowfish com variantes de 14 iterações ou menos. A tese de pós doutorado de Vince Rijmen inclui um ataque pelo uso de diferenciais de segunda ordem em um Blowfish de 4 iterações que não pode ser extendido para mais iterações.

7.2 Características

•Blowfish é um algoritmo de cifragem por blocos criado por Bruce Schneider.

•O Blowfish trabalha com uma chave variável de até 448 bits e blocos de 64 bits.

•Blowfish foi desenvolvido para ser: Rápido

•Blowfish foi desenvolvido para ser: Compacto

•Blowfish foi desenvolvido para ser: Simples - as únicas operações utilizadas são OR, XOR e buscas em uma tabela

•Blowfish foi desenvolvido para ser: Seguro - o comprimento da chave do Blowfish é variável e pode ter um comprimento de até 448 bits.

7.3 Funcionamento do algoritmo

7.3.1 Obtenção das chaves

Nesta parte devemos gerar as subchaves. As chaves de expansão convertem-se de uma chave de 448 bits (ou menos) em várias subchaves que totalizam 4168 bytes.

a) Inicializar o array P e as quatro S-Boxes com uma cadeia fixa formada por dígitos hexadecimais do número pi.

b) Fazer um XOR de P1 com os primeiros 32 bits da chave, denominando-o P1, o XOR de P2 com os 32 bits subsequentes é denominado P2 e assim sucessivamente até obtermos P18. Se os bits da chave terminam antes de haver-se completado o array P, deve-se recomeçar com os primeiros bits da chave e continuar a operação.

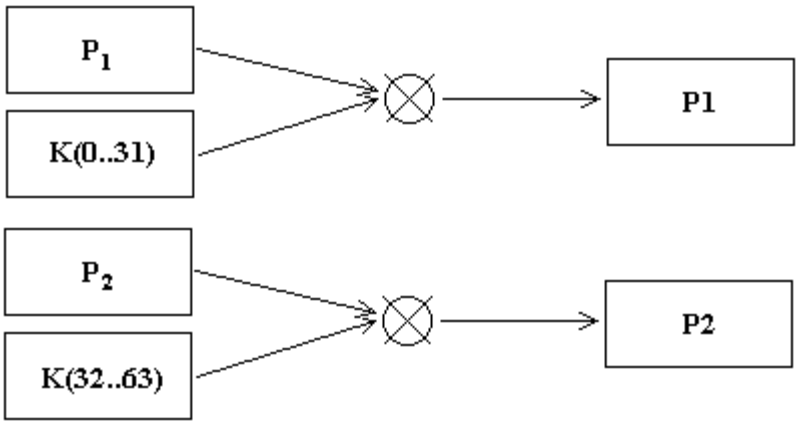


Figura 7.1- Esquema gráfico para obtenção das subchaves intermediárias P1 e P2

c) Encriptar uma cadeia formada por zeros segundo o algoritmo Blowfish utilizando para tanto as subchaves obtidas nos itens a e b.

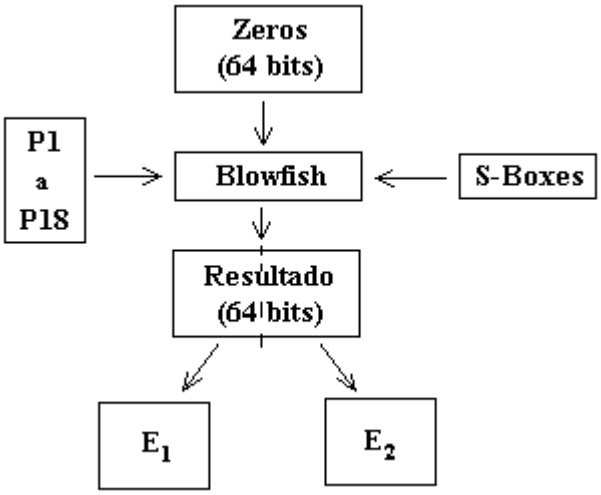


Figura 7.2-Esquema gráfico para obtenção das subchaves intermediárias E1 e E2 que se tornarão as subchaves finais P1 e P2

d) Substituir P1 e P2 pela saída do passo c.

e) Encriptar a saída do passo c usando o algoritmo Blowfish com as novas subchaves.

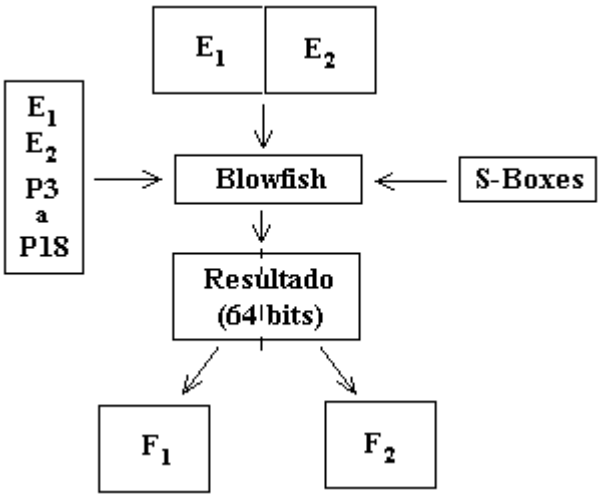


Figura 7.3-Esquema gráfico para obtenção das subchaves intermediárias F1 e F2 que se tornarão as subchaves finais P3 e P4

f) Substituir P3 e P4 pela saída do passo e

g) Continuar com o processo substituindo todos os elementos do array P, e depois todos os elementos das S-Boxes.

Este processo se repete 521 vezes, 9 para obter-se P e 512 para obter-se S.

7.3.2 Cifrando um bloco

a) Divide-se o bloco de 64 bits em 2 blocos de 32 bits (X1 e X2).

b) XR = P1 XOR X1

c) XL = X2 XOR F(X1).

d) Repete-se os passos b e c 16 vezes substituindo-se X1 por XL e X2 por XR e utilizando P2 em vez de P1 na segunda iteração, P3 em vez de P2 na terceira e assim sucessivamente.

e) Finalmente a partir de XL e XR da 16º iteração fazemos XL = XL XOR P17 e XR = XR XOR P18

f) Une-se XL e XR obtendo-se o bloco de dados encriptado

Função F(X1)

Para calcular a função F(X1) divide-se X1 em quatro partes de 8 bits cada uma, a, b, c e d, e se calcula utizando a seguinte fórmula:

$$F(x1) = ((S_{1,a} + S_{2,b} \bmod 2^{32}) \text{ XOR } S_{3,c}) + S_{4,d} \bmod 2^{32}$$

7.3.3 Decifrando um bloco

Para decifrar um bloco, devemos apenas repetir a operação de cifragem.

Devemos tomar o cuidado de utilizar as chaves de forma inversa, quer dizer aplicando primeiramente P[18], depois P[17], até chegar a P[1] nos itens (a) até (f) do Passo 2.

7.3.4 Esquema gráfico de funcionamento

Função F(X)

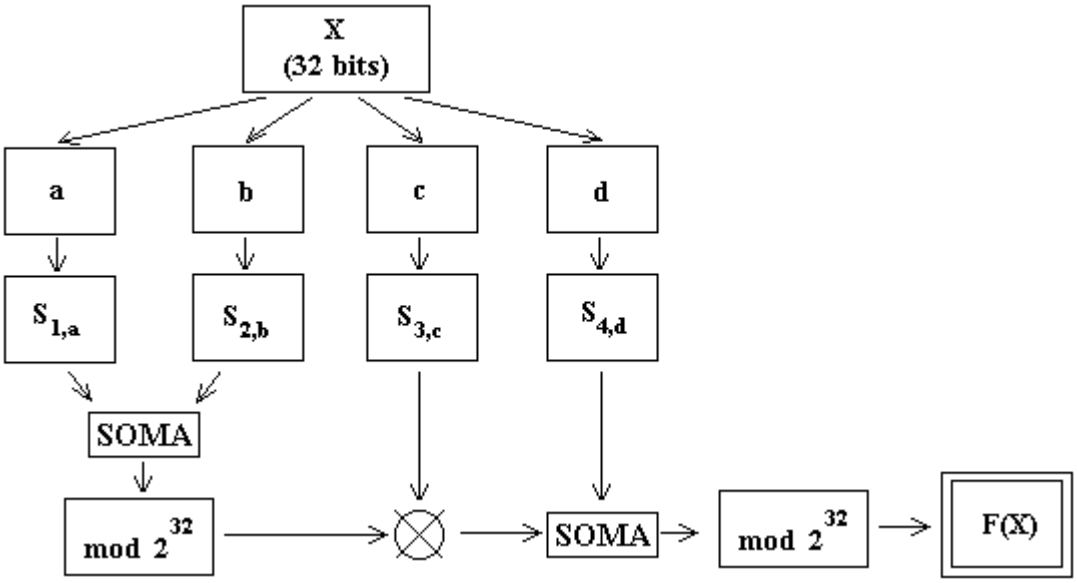


Figura 7.4-Esquema gráfico de funcionamento da função F(x)

Funcionamento do algoritmo:

1º round

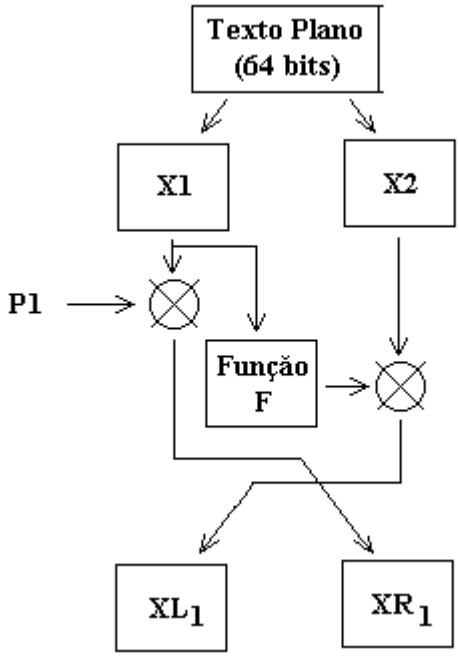


Figura 7.5-Esquema gráfico do funcionamento dos rounds 1 a 15

O esquema acima se repete do 2º ao 15º round

16º round

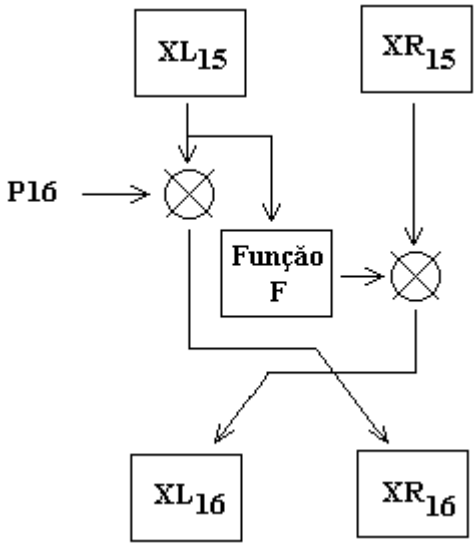


Figura 7.6-Esquema gráfico do funcionamento do round 16

Último round

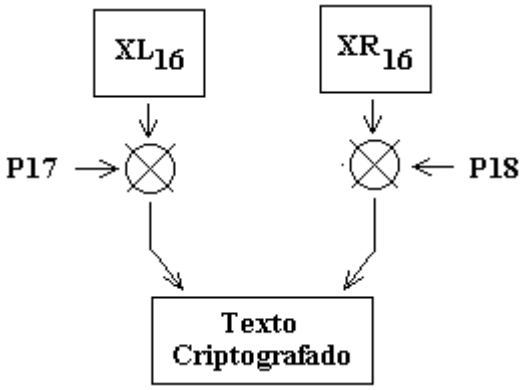


Figura 7.7-Esquema gráfico do funcionamento do último round

7.4 Implementação

7.4.1 Requisitos de Software

•Microsoft Windows 95, Windows 98 (original and Second Edition), Windows Millennium Edition, Windows NT 4.0 (com Service Pack 5 para correção Y2K ou Service Pack 6a) ou Windows 2000

•MATLAB 6.0 (R12) ou 6.1 (R12.1)

7.4.2 Requisitos de Hardware

•Pentium, Pentium Pro, Pentium II, Pentium III, Pentium IV** ou AMD Athlon

•64 MB RAM (mínimo), 128 MB RAM (recomendadop)

•adaptador gráfico de 8 bits e monitor (para 256 cores simultâneas)

7.4.3 Descrição dos módulos principais

O Blowfish é composto de vários módulos, os quais devem estar no diretório de trabalho do Matlab.

Os módulos principais são: GeraChavesBlowfish, CifraBlocoBlowfish e DecifraBlocoBlowfish

GeraChavesBlowfish

Este é o responsável pela geração das 18 sub-chaves e pela substituição das SBox (Ver Passo1) necessárias ao algoritmo.

Sintaxe:

>>[Chaves_P Sbox1 SBox2 SBox3 SBox4] = GeraChavesBlowfish(Chave_string);

Onde:

Chave_string é uma string no formato hexadecimal com o valor da chave de entrada, esta deve ter até 448 bits. Exemplo:

>> Chave_string = '0123456789abcdef';

Chaves_P é uma matriz que armazena todas as 18 sub-chaves, onde a primeira chave é representada por Chaves_P(1,:), a segunda por Chaves_P(2,:), e assim sucessivamente.

SBox1, *SBox2*, *SBox3* e *SBox4* são as 4 SBoxes utilizadas no processo de cifragem (Ver Passo 2 item 2 - Função F)

CifraBlocoBlowfish

Este módulo é o responsável pelo processo de cifragem propriamente dito. Ele trabalha cifrando um bloco de 64 bits.

Sintaxe:

>> [XLfim,XRfim,F,XL,XR] = CifraBlocoBlowfish(BlocoLeft32, BlocoRight32, Chaves_P, SBox1, SBox2, SBox3, SBox4)

Onde:

Chaves_P, *SBox1*, *SBox2*, *SBox3* e *SBox4* são os resultados do módulo anterior, ou seja, as 18 sub-chaves e as SBoxes.

BlocoLeft32 é o conjunto dos 32 primeiros bits do bloco de dados a ser cifrado, ele deve ser uma matriz linha binária. Exemplo:

>> BlocoLeft32 = [0 1 0 0 ... 1 1 0];

BlocoRight32 é o conjunto dos 32 últimos bits do bloco de dados a ser cifrado, ele deve ser uma matriz linha binária. Exemplo:

>> BlocoRight32 = [0 1 0 0 ... 1 1 0];

XLfim, XRfim são o resultado da cifragem do Bloco de 64bits formado pelo BlocoLeft32 e o BlocoRight32 através do algoritmo Blowfish. São blocos de 32 bits cada.

F - matriz que armazena os resultados gerados pela função F a cada iteração

XL, XR - são duas matrizes que armazenam os valores intermediários que os blocos assumem a cada iteração.

DecifraBlocoBlowfish

Este é o módulo responsável pelo processo inverso ao de cifragem.

Sintaxe:

>>[XLfim,XRfim,F,XL,XR] = DecifraBlocoBlowfish(BlocoCifrLeft32, BlocoCifrRight32, Chaves_P, sbox1, sbox2, sbox3, sbox4)

Onde:

Chaves_P, *SBox1*, *SBox2*, *SBox3* e *SBox4* são os resultados do módulo GeraChavesBlowfish, conforme explicado anteriormente.

BlocoCifrLeft32 é o conjunto dos 32 primeiros bits do bloco cifrado a ser decriptografado, ele deve ser uma matriz linha binária. Exemplo:

>> BlocoCifrLeft32 = [0 1 0 0 ... 1 1 0];

BlocoCifrRight32 é o conjunto dos 32 últimos bits do bloco cifrado a ser decriptografado, ele deve ser uma matriz linha binária. Exemplo:

>> BlocoCifrRight32 = [0 1 0 0 ... 1 1 0];

BlocoDecifrado é o resultado da decifragem do Bloco de 64bits formado pelo *BlocoCifrLeft32* e o BlocoCifrRight32 através do algoritmo Blowfish.

XLfim, XRfim são o resultado da decifragem do Bloco de 64bits formado pelo BlocoCifrLeft32 e o BlocoCifrRight32 através do algoritmo Blowfish. São blocos de 32 bits cada.

F - matriz que armazena os resultados gerados pela função F a cada iteração

XL, XR - são duas matrizes que armazenam os valores intermediários que os blocos assumem a cada iteração.

Outros Módulos

Os módulos acima (GeraChavesBlowfish, CifraBlocoBlowfish e DecifraBlocoBlowfish) são formados de diversos módulos necessários a sua implementação, na próxima seção veremos em detalhes sua composição.

7.4.4 Descrição dos módulos

GeraChavesBlowfish

o Constantes - inicializa o array P e as Sboxes com números binários (Ver Passo 1 item1). Os valores utilizados (em formato hexadecimal) são apresentados no final deste anexo.

o Hex2Bin - Converte uma string em hexa para um array binario.
Sintaxe: [ABin] = Hex2Bin(h);
Exemplo: h = '8f'; ABin = [0 1 0 0 1 1 1 1];

o CifraBlocoBlowfish - Função de cifragem propriamente dita, ver sua composição abaixo e detalhes de funcionamento no Passo 2.

CifraBlocoBlowfish

o FuncaoFBlowf - Implementação da função F(XI) para o Blowfish. (Ver Passo 2 item Função F(X1))
Sintaxe: [F] = FuncaoFBlowf(XI, sbox1, sbox2, sbox3, sbox4)
XI é um array linha no formato binario de 32 posições/bits
(XI = [1 0 0 1 0 ... 1 1 0];).

DecifraBlocoBlowfish

Este módulo é idêntico ao CifraBlocoBlowfish, sendo que antes de iniciar a cifragem (decifragem) ele inverte a posição das sub-chaves P, necessárias para decifrar o texto (Ver Passo 3)

7.5 Laboratório Virtual

7.5.1 Criptografando com Blowfish no Laboratório Virtual

Primeiramente geramos uma chave válida para o algoritmo, para tanto devemos selecionar "Obter chaves para", escolher o algoritmo "Blowfish" e pressionar o botão "Enviar", como ilustrado na figura abaixo:

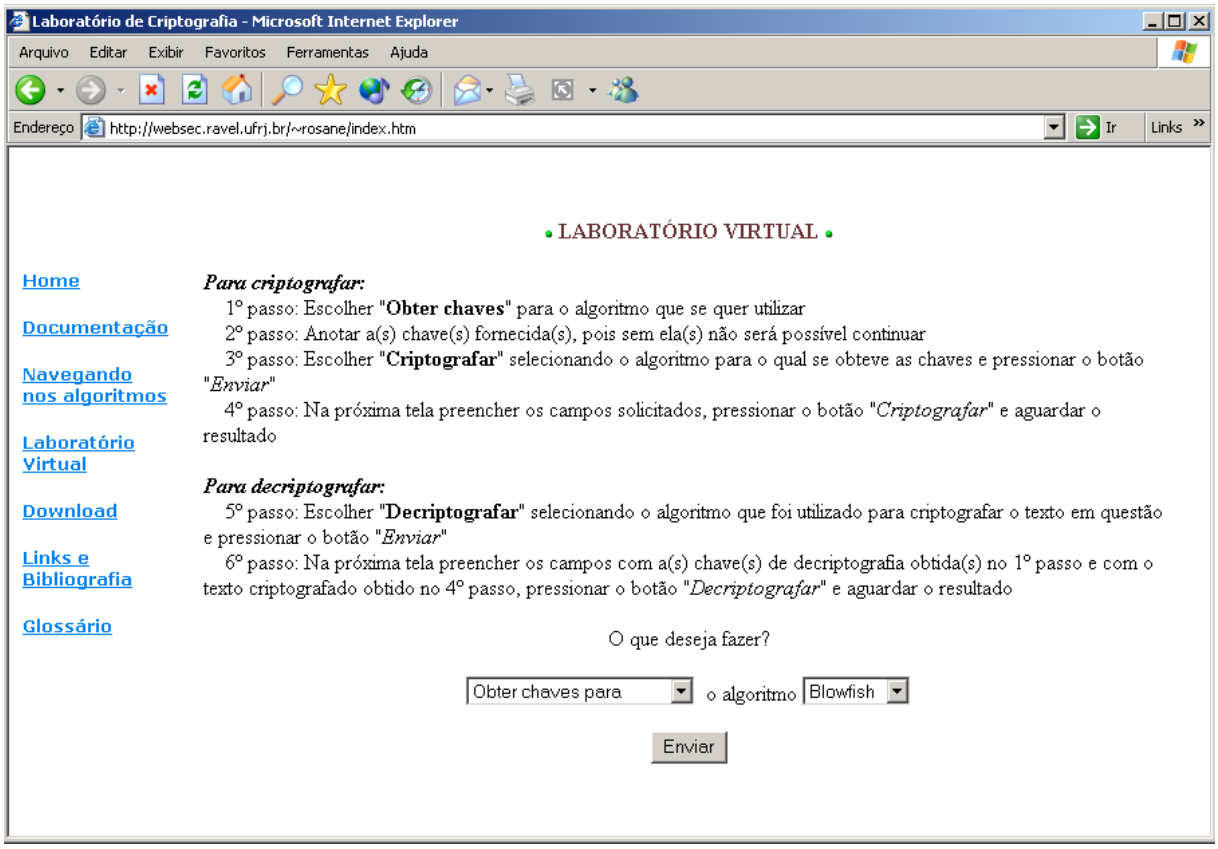


Figura 7.8- Obtendo uma chave válida

Após alguns segundo um Pop-Up como o mostrado na figura abaixo sugirá. Ele contém a chave que poderá ser utilizada no processo de cifragem.

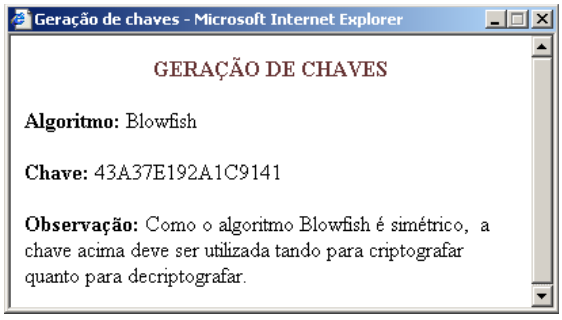


Figura 7.9- Resultado da geração da chave

Após obtida a chave, deve-se escolher "Criptografar utilizando" e selecionar o algoritmo "Blowfish" como mostra a figura abaixo:

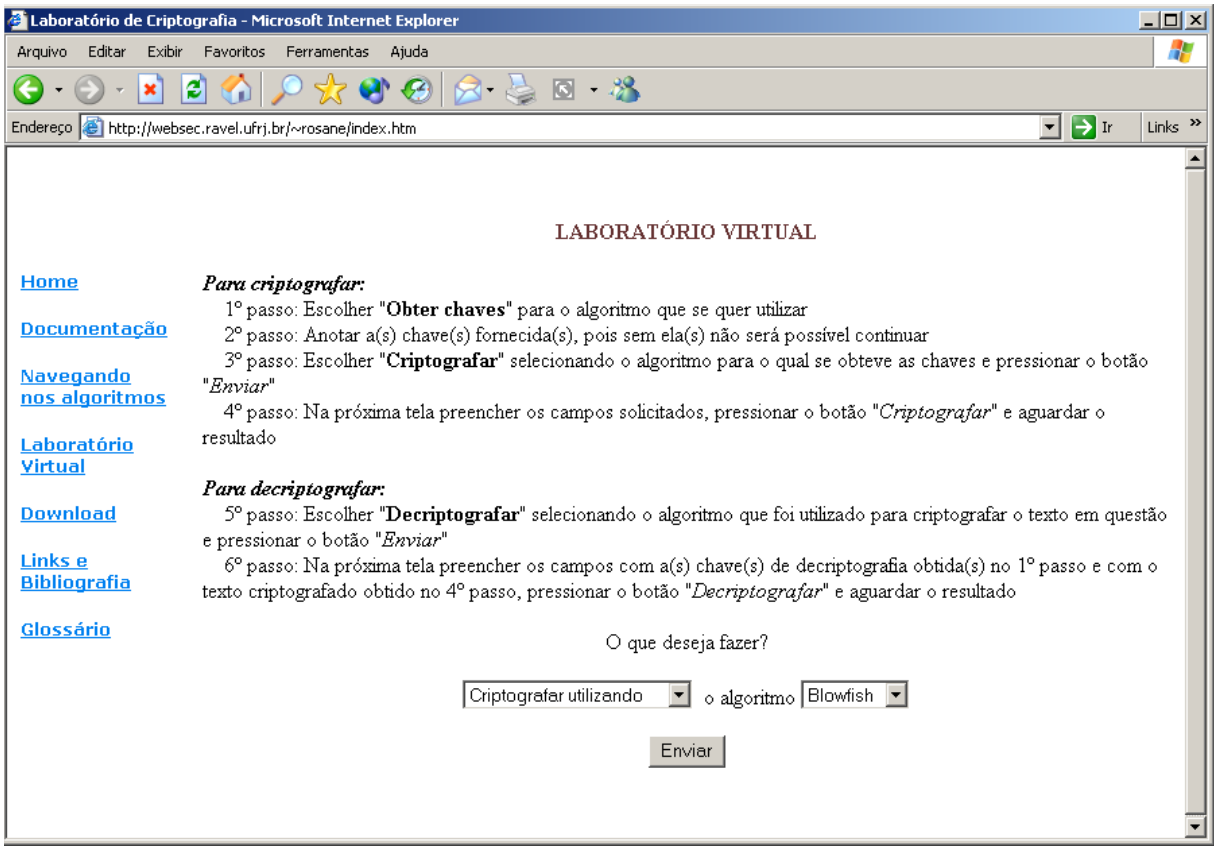


Figura 7.10- Optando pelo processo de cifragem

Após pressionar o botão "Enviar" uma nova tela será apresentada, nela será possível digitar a Chave Secreta (obtida no passo 1) e a informação a ser criptografada. Como pode ser visto na próxima figura.

A informação a ser criptografada pode conter quaisquer caracteres alfanuméricos inclusive caracteres de pontuação.

Após preencher estes campos deve-se pressionar o botão "Criptografar".

Caso algum erro seja cometido, pode-se utilizar o botão Limpar para remover todas as informações digitadas.

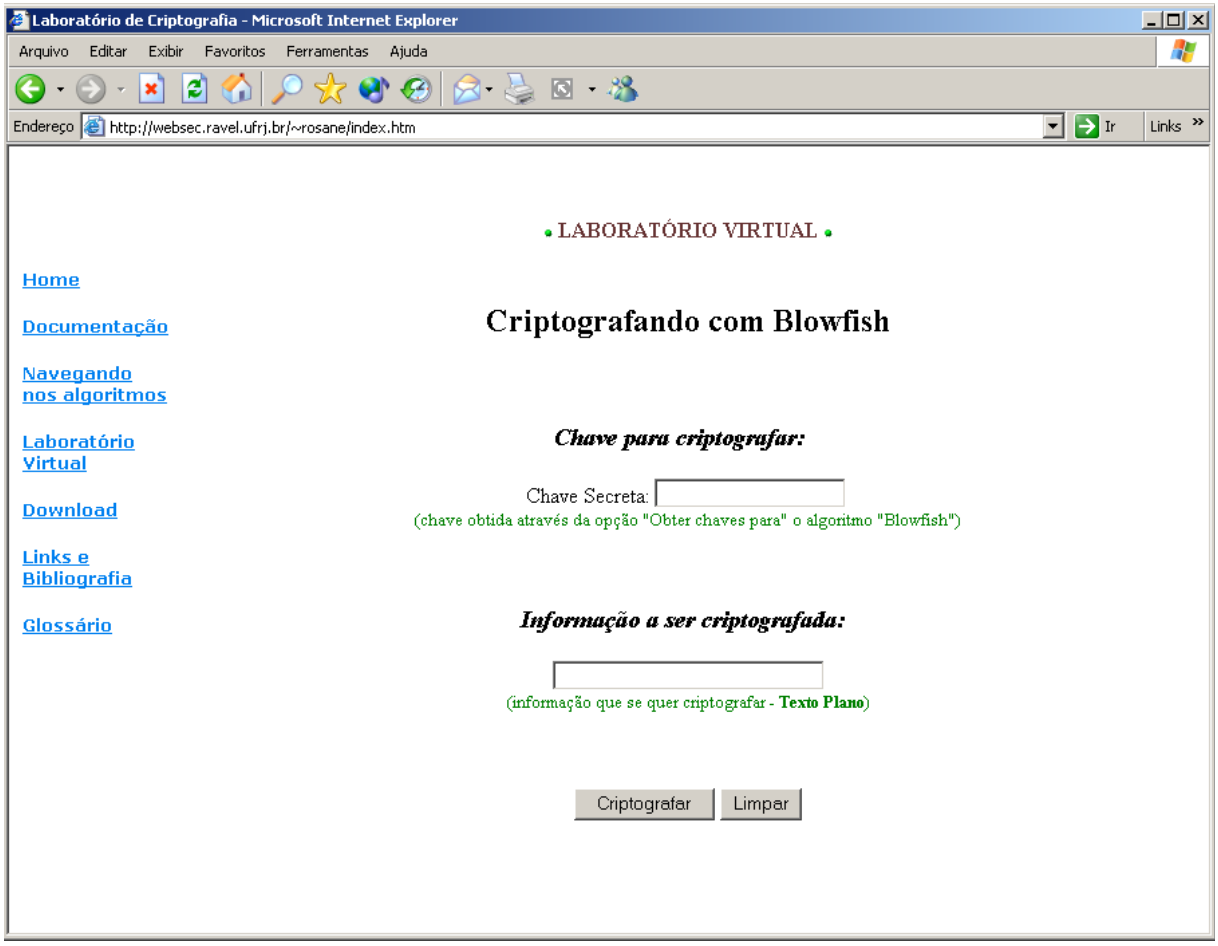


Figura 7.11- Inserindo os dados para cifrar

Após o pressionamento do botão "Criptografar" as informações serão processadas pelo Matlab. Este processo poderá levar alguns minutos, dependendo do tamanho da informação a ser criptografada. Após decorrido o tempo necessário, um Pop-Up será exibido para o usuário com o resultado da cifragem.

Neste Pop-Up, além da informação criptografada, serão exibidas também outras informações, como o texto plano, a chave utilizada bem como uma figura ilustrativa do processo de cifragem. Esta figura possui áreas clicáveis que revelará alguns detalhes do processo, como por exemplo os blocos em que foram divididos o texto plano antes de ser criptografado, as subchaves geradas no processo, etc.

A figura a seguir mostra um Pop-Up resultante no qual foram utilizadas as seguites informações:

Após o pressionamento do botão "Criptografar" as informações serão processadas pelo Matlab. Este processo poderá levar alguns minutos, dependendo do tamanho da informação a ser criptografada.

Após decorrido o tempo necessário, um Pop-Up será exibido para o usuário com o resultado da cifragem.

Neste Pop-Up, além da informação criptografada, serão exibidas também outras informações, como o texto plano, a chave utilizada bem como uma figura ilustrativa do processo de cifragem. Esta figura possui áreas clicáveis que revelará alguns detalhes do processo, como por exemplo os blocos em que foram divididos o texto plano antes de ser criptografado, as subchaves geradas no processo, etc.

A figura a seguir mostra um Pop-Up resultante no qual foram utilizadas as seguites informações:

Chave Secreta: 43A37E192A1C9141

Informação a ser Criptografada: Criptografando com Blowfish.

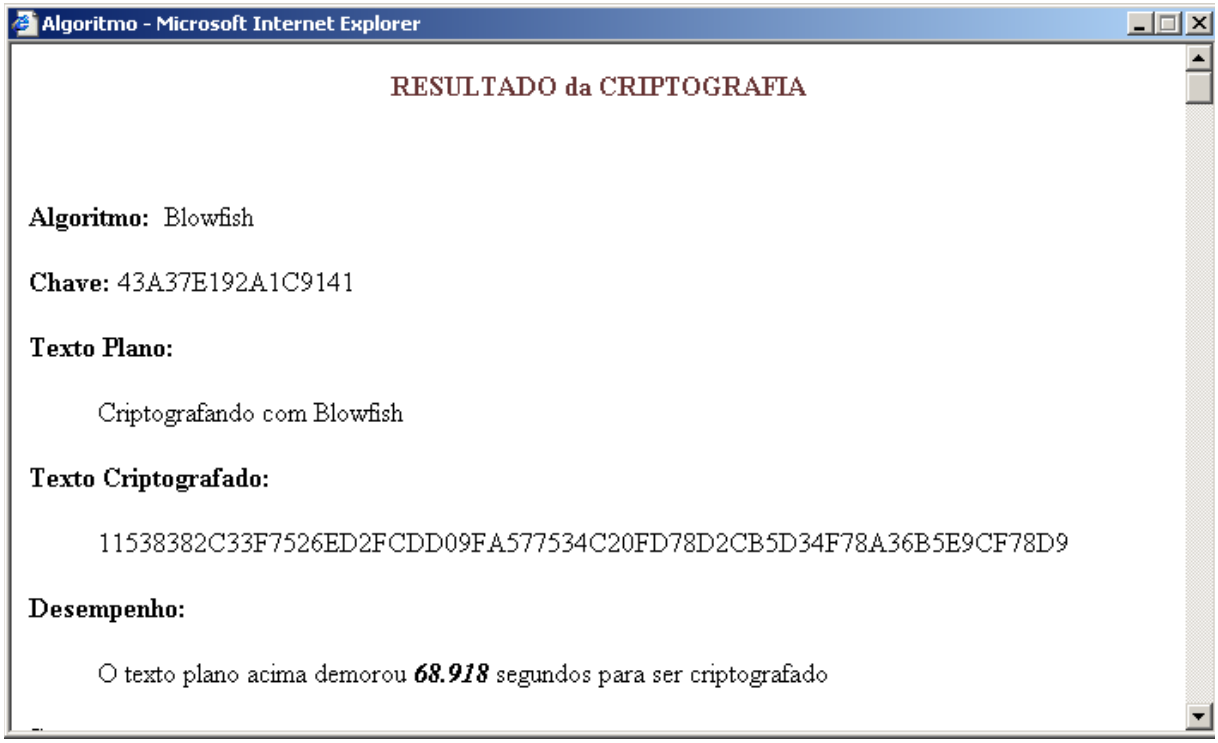


Figura 7.12- Resultado da cifragem

As demais informações do Pop-Up são apresentadas abaixo:

Segurança:

Até hoje não foi comprovada nenhuma forma de ataque bem sucedida ao Blowfish.

Referência: Applied Cryptography, Second Edition: Protocols, Algorthms, and Source Code in C, by Bruce Schneier

Detalhes da cifragem:

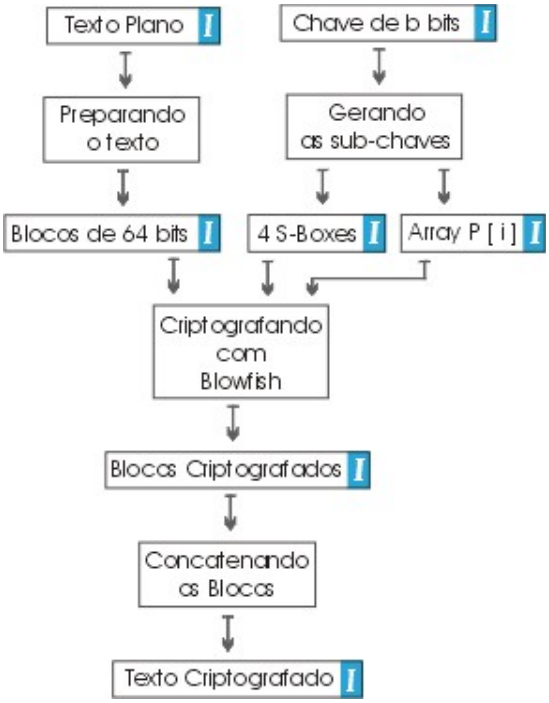


Figura 7.13- Diagrama da cifragem

Texto Plano:

Criptografando com Blowfish.

voltar ao diagrama

Blocos de 64 bits:

43726970746F6772
6166616E646F2063
6F6D20426C6F7766
6973682E00000000

voltar ao diagrama

Chave de b bits:

43A37E192A1C9141

voltar ao diagrama

Array P:

P1:	62789E1A
P2:	54B4E422
P3:	D49E1D61
P4:	2F60517
P5:	173646E1
P6:	7058E1FD
P7:	972352BF
P8:	7B21E225
P9:	1E65A19E
P10:	4A38A608
P11:	BEEFB5C2
P12:	DDD2BA40
P13:	170E8D5D
P14:	45481D00
P15:	4CEA28D5
P16:	16D5C7A3
P17:	1078FF8D
P18:	69A6B917

voltar ao diagrama

S-Boxes:

S-Box 1:

34BEF543 5E4F848C EC9A3819 B546EB4D 1F1113F2 DFC70BD5 1D964E69 C25E2416

1102E4AE C491A9FB 59A4BA9E A8AD7A3D B18EF7 2FACA1D3 6536B040 BBDADB63

7DEECE60 E1174015 7E16A42F 4BAC406D C0A11F33 C975B243 E9EC6825 96E30CFB

E9A6DC24 6F822EA5 6C30B29D 3932D348 B84475E8 D8DBE49C E48BBBD8 F25849C2

3E0C5A9D 5868D836 D431025A B0910F4F E98940C4 CC5B28F5 8E880E6B FF67102

3971A1BB 3448024D E587C909 7547A49E F073A6BB C69226DB 17F8C39D E6954FB7

286E9C46 987FBB3E C7A2D386 A3F3B20B 55CE5B93 8E87B917 78D43AC3 EF2761BE

292BAF6B D661E2BC E3FAB49C B41D26A8 11C281DE 4ECCBAB3 E483D329 76A5E4DB

F44F8178 56CC5F12 64F5BBCA D3C568A5 E9683A81 AB8A6689 60B80A90 1F2EA8D

2471EF38 2CD4349A 4ADC5E9B B56ACD89 E4277744 3B1E946B E960A780 867A22F1

92438886 2920CD3D 29EC5EE1 9CFD75D3 37C29D8F DFEEE7F9 1E10B487 714AB6A6

BFEF1EE9 E1AE98E6 C657AE91 B75147A7 EF011C59 A540B1B5 39783AA3 F0C1D26F

F527EC4 469D3215 97F19C4D AA367CE7 41B4F2FC 4C0D1D45 49F400D0 884C86AC

A82939A6 A30616EA 90C32A5A B7667A03 9A585FB6 1A3823E3 BEEFD38F 3F135296

FAC34BF5 B990B99B B2C4348A 4A2B171E A4535F44 EF958BB2 699BB7C8 DB1CF47

34B93E5E 896AF6BA 2869D600 25BC736 8D11D928 AB354EAA F5C3F3F0 D5D8EF7E

6DF00464 8A8FFB49 E1C1F00E 7856FC6 D1458F9B 459BCA6E 57CF773F C97721F7

55802D8F A54BA7E1 468CD8B4 F04C9400 6103DFCE DDE075B 36AF8461 9172712D

EE66B08E D9CA05CD DD4878EF D2E9EA69 3D7079DA 28D90F0F F7D807B1 C7FCC4E9

8A908CB4 5FC77147 318F11E8 DDB540A3 4D9D125B DA24A814 241CB8D7 9EDCA26D

5905CB1A 4B4BA800 8FA2E6C8 12662270 15B93B48 86B7A5A5 137422EE 6FA0A7C2

86796D7D 7E03DB7E D7C83A4B 2C5E0F8A 8457F5DB 4C570C1E F1D05570 A27052C7

CD18B703 979B7104 D30331C5 A41AE956 71D72DC2 BAFD0D5E C7844DB2 C3095C3B

1A798D60 5CF55FF 2B19A34 344B30BF D6CEC0F0 F2D789DF 6FA39F46 7F882D53

BC964F0A 192D7F4A 81A0E544 FC4DF33B 3D0EC0EC ED647B66 40C27B77 B5ACD531

C0D7B0D7 E4138731 C9B4F71 8EBCDF20 59BD5615 4D094398 E4AFF438 67F35500

A1FF7FAE 9EF558EB C7F97E46 C1EE5DF4 ABCE662D 14A989 4DF48E48 C7A74822

6095F926 DFFA224F 98C8205A 8F621AB4 D911AE7E 522BD78B FEEBBE28 3B84A9A4

A8F55E82 9051070B FC42F98C 919381DC 7B8FF91C DEA9CE85 6725B796 F3D67F14

13F32C74 B1F94377 D5185552 99AB4619 25162003 B37F5E09 39FD4D87 B0A5DFD0

EEE627AC 7736DA6C F3EE5C8E CB230430 34CCC444 CF31C62C CEBE532 15DC6090

DE0BF8F5 379BA0BC 12F08C82 E167A7B2 EBA48889 A8E94CB8 C35B86B5 7DB233EA

voltar ao diagrama

S-Box 2:

7AD82439 EAAE0E90 16D889A5 52771C0 965180C EBA46202 9D162A2C 18A7E6B1

AC8DCB5A EB4A4F76 AACEE06D 251E92D9 666D5B27 253FE2D9 B00D26E4 26490626

175EB6A8 9BB2E4F7 56ED9EAE 88CEA47B 9B22B216 86570BE2 2EECB865 B87137C1

AAD5C16B DCE218E5 3F6D9962 ED86C660 3CF98AAD F77DFFC8 EABEF524 3D522BE1

5DADC6D0 426CC451 8BA160F1 E775E60E A1B98166 D55ED25 813ED286 EC52E01B

3F7D33AD B97D1A31 900280FF 99143344 94F13828 5B2C1C4A 33C1F999 93B30A56

747BCD0C DF03C77E E37B39B5 D75931C9 8CD276EE 6D39A1E1 2533BF3D 7A7F8B53

63F8172F 7B49E5C 97585E07 A3A9F4D4 162C9718 326C0757 33B3D8F7 61B3F912

4D373566 5DED3CB6 D4EB4078 E91C3E0D B23081D3 4BF2EB8D C4439699 3EE3FA9D

1361A30C BE071B72 DDDDEBD6 142C239F 4D825F00 A24CBA20 4766A044 1DFF47F1

5FEA07CF D0F6BA45 334C98D6 2F042A19 C4DCA55E FF64E710 74B55245 CEA98113

ACB9311C B40A8CFB 10BEDCA7 50534C21 E22AA80B 711EF208 B8AFA143 7B1154E3

BBC0800C 8D0AC0E3 AA7E7EF 31C7BF71 7730A46 67DD7393 953B436A 1E323105

33B04395 B6DD1CD 555DC148 D6094B45 94E5204C DEE4CFD5 2FFBD71E F685145D

34E0D3A7 AE6AADBE 14A3C730 1F01BD55 55ACA948 91AE647C EEF0604 100181D8

386DFCB 40393936 1D96ABFA 22D756A5 65FF3D10 E5652409 280DC531 18B58EA9

C04CD9B2 E58D9AAD 705E3D67 AE84D23B 1373BF38 ADB87146 1C45BB5 608E0B1C

27DA974A 15D676AF 7DA761C7 5C9447AD B341966D 70CE25E 62166F1C 2BC5ED39

67C5402C 1AAB392B 9262260E B84E1DFE FD0187BD 2E935950 9B890EBC DF402070

8A8265CD 4C00FAF BF3D7CEE F7FE82E6 9983F7B8 BF0D1DF8 1040E993 5BC00665

792BBDFB 95516DB7 CDEF5582 72688A95 BADA85C2 EBCB8490 218D0E77 4A693626

E76CE6E3 B01993E4 A05E8939 782F5BA3 429ED3A1 22D576DB 380C921A BF3467F4

293090C2 883EE920 2AF2B8A5 66159D11 139C4996 8837B278 3BA1AF22 21F25F5F

AB9791FF 7733B92F CA989A0E 76336B17 7CF38FE7 219D60AC B506FB5 70B1E8B2

1001D965 B392F4A2 ABF76299 B1437611 CDA17415 804267B7 47346159 2E2454DE

99157FCD 52EDC410 C39439E5 361C46F9 5455D9F7 54594C5B 10417587 6BB36D46

FD5C8AE8 D4F43B2A 49707375 E4612428 B0F3C05E 91A72279 2E06E6C9 67FD5C00

CEFE9697 4A393F6F 57C7ECE C57ED843 4CF30B19 A13FF2F1 8F5470FA FF7C2FEA

E60D59FC CB900E3D 6ABD54DB E684DC5E ED26E17B 6D05A00E BE0021C2 DE6F03

FFEE5CEE 667DFA85 691F2830 EE44265 D36F88C0 FAABAF 1C586E7C 8755D6ED

3D9E79F2 D47C928A 9AA5293C E77FBAB A5C0C5C7 7737A19 D689A024 85F76585

D3D446A2 EB4E124E D8DC5151 E7B8C046 6FEABBD3 E9394644 2FA41E89 35BCF345

voltar ao diagrama

S-Box 3:

58F3D85F 21CB6C22 968702B3 5AAFEC2A E86EE56D 42A7F59B 78DDBA74 BA8A66F

BA72780B 5FB36D84 7F25314A FAA68242 D7A6D555 83EA6CA8 E9E63235 1813BA17

58124362 A754156B AC6509A3 A78C11B5 FAC285FC 28BB0B90 87206F27 BA788957

5BF0810F BF12FC91 34397AAB F584588D 3DCB3A80 19C9DD5A 512F1BBC D10EB0BF

FF10416C 7BDF315D AF360BA4 128CD76 687E0349 286B8576 27D439A AE59C4D3

B4560F2B A03B4235 3F5E39AF 4E895080 2012D835 62258DAA 90B49F92 AEFBEE3B

F642400C B1F1E629 321F68A7 62D2B3D3 1755CA13 51FB1632 5CF48F9 D0A378F2

E7AFE5AE 30C3ADC9 BAA9F631 C92D178 F3B817C3 E5630296 DA72C366 F43511D9

365CA368 8192F284 B617013C A82E0026 46096502 F8CA5187 AF45E401 D6C7AEAF

8AA7506 98F7CE0A 55E048CA 6E902183 B9558292 4445F415 B409F1B9 3D326446

FD959C28 446ECEB7 FAC59B60 9E39FCC3 56557BF8 80A66B68 5133A88 B75AC167

507A7D0B D695B01B 9A889435 CA2D27F4 3A3ABE1A BF1E9E85 8DE70549 68A936B2

9A8E9C22 C8E61C11 A58B3A57 4BF08B8F EF30E230 F4B89937 F9A16D7F 399C9432

485AC828 32D60C83 5CC02015 C159EF26 3D047B9D D8959853 AD28269D 29CDB635

73D512A2 205C6453 1B7790D7 4CC1BA40 34CBF250 1087935 76864E4C 1647CC79

17BBC2D8 DD489045 BAFE89CD 17B8EC09 7A5C6B2 3D0428A7 D1A70C2A 5C70A569

2EC8D0C F9716915 37410CA5 F5C4913A DC21A99E 262D9C16 E0881800 3EEC2FF6

95DE086B 5F3D5CE6 1EE63F1B E7A3C2FE 18CD11A1 B5A7F117 CE509136 6206B768

1FEC2E92 B2636C2 9AF9F335 9CF63F21 724F9345 2DFB5D4A DC6F4B22 C02FC8C6

A4AC334C 2DFFBB34 3DC764F2 882D71B2 61DD8C1D 73C72B8C FB798E4B 9B9BE74E

E8438958 DBA1B45A B77AC604 6BB6E978 715AD846 B4EF4915 8CD851C1 33896D8B

36185A38 A5A264FF 584C01E1 90A81A63 D9EBF5C8 83CB20C0 6E561FF3 7D7A2931

3D5FDB9A F8B7FF37 8B52F0F7 D39B6943 5B63274F 7F8C201B 6480289C 8A1B4D05

9C34408A BB8E1520 7A9801FE 444EEA34 DD7D0A45 69ABF2C1 7201F326 B2FF6616

93B3086A 5485F6DB BF856FFF 8FD61D04 BA645CB0 8A3E40F4 45503322 3C1C81D1

B5C3B046 A602C81B DE40457F 46BEA746 308BEE62 FE69BE70 AD657E5A B622AED1

D9C4A1F7 FE32D988 10A437A4 D9792517 BCCF3CA8 F7F96621 E8509B3F BDF1F03

B4A6A09B AF466013 37341A59 BF362DBB 66607A07 6E961B73 70BA646 2E9452E8

BA919B4D 91C91136 6F8C4F27 800583C4 57C0D4F8 44ACEF1D EC1D6797 95C2E083

AEEEE807 5C13D91 ED772058 4F560E67 A9B5D3F8 49941770 29B1736B FB699F7F

BBE5B082 844BB77E 48FBCC06 F622E66D E283FB02 F5082613 E677C95D BD9518E4

C5A81E4D 2C351920 571C06EE EE8E3B77 85D58452 E3357123 733BF6F4 D7439B00

voltar ao diagrama

S-Box 4:

C2F65B96 D55D7B20 93D9CB92 E728A974 88D725A3 43BDD2DF 67A07C58 6CFA4C52

728A04AF 56360E3A 1B1BA253 7CBEA775 CC9ADB3 C9FD7E6E C577A15B 424268B7

61A9F6F3 C7B0EFAA A8E7C472 CAD83B5D D892939B 2892164A 547024A1 1DD36A29

4AA1519D 94599C3B FFA9F3BA 6EDF08DA 395780DE 325D15AA B97AF9E2 A67AE05E

F4519181 1ABC373 DD23A4F0 3E22256D 2E11ECDE EFED6EEC D5169A3B 3B90CA28

631F0687 5D7AFC8D B6C43620 77226B09 DDCB6E97 F72BFD08 4AB3B05E 3DAAF7D3

A367E4CD C8FC332B 834E7630 AE8FDC55 613463B8 6D7FD530 56E5F536 DA17AF94

8D13B1B 4B25757E A144B8CB 7E5F874B 183404A9 3FD6FEE8 EEA788C3 9F75BA9C

A71CF5BD 71526676 EC8FC870 FCBD73BF 44C96A37 6042637B A60A2F8D 3BBEDFE9

DA6CA2EF 3D2C3B94 AA696074 46BF4065 A5AAE6B8 7BA51B20 9B8B83FA D8DA1E4A

9B650AB0 ECC2B4D5 9BCB15F1 EC7D5EC3 EAAED0B0 16E7C0B4 40928CE2 EF789CF1

CA760F31 7F974AB2 116E51EF ED2B0697 46A8C482 9659A984 B707F55A 7B0FAF4

446AA381 8DB625A0 ED5B0DB8 2EC20FF7 AB30FEA8 56CBF21 BBE62FC1 DDADD0B2

9DB436D8 69ED54FA 7639D0EA 1108BA2E 5CBAA45D BD43F03D 2AD3564B E427D9F3

86B7223F 1CA1AAC0 1170F625 C2BF6D98 352ABAE4 EF8ECEE3 8B2FAA9C 346C3516

2F7B1BC3 2952E354 EC53127 EEFD6323 434589AF 1224ACE7 78165B08 E914500C

F61077E9 C126D57E 10DBE13C 83D86C05 5CC9FB0A CAF5C035 9F502BF8 7E656509

F707CE0 AA764CB4 2A30B3D3 ED8C2317 5A489347 72AA821B 1EB6FFF1 B39602EA

8D0F547B F56A72A5 760AEACA 8A69D2B7 9F9CDC38 9C6DA0B C7D2E179 878AEEA1

729E76A5 B0E1EC0E 4D2A5111 8818133B A53AB3ED 4C8AA1E1 B6111D3E C3A1C98

7861EF80 AB653AE8 C4A1D584 F5582EB 19687C1F 9FC95592 F60DA9EE C1AAD410

7E43D7A3 CEF3D6F6 78A5E927 F4B5A907 78921155 A36B4D75 59BBAB7A 8EF37716

87F95BD0 E3F56FE5 27149C4C BAE3D258 72539E4B FAD8BD9 9A9EB542 43C2FDB8

F30D7707 A762E208 10207466 DAD8341A B16F60DC 54115338 46629565 229D0680

5A7663CD 174B49E9 96AE03E1 1E220824 8FAB0174 764F6071 7C5935C7 6FA0373

7F7F3605 1854A0CA BD94D1AE 4D36C6F3 692408C7 E1C7163C D6111E65 D06CD286

BBC161B5 3358D7B9 3838D62D 90D07E8B 8FD76458 46E80ED3 9E979AFA 9A8A32B3

A332B300 86950957 A9B5FEEB 5335C19C C5939D89 102830DE BEAF06BA 9FFD39E0

64C16C6B E2658DB7 BEC1956B F7CAFFE2 75E1F700 7E4001E5 F07F6D41 A448F52A

6FCFD7A2 830B9330 A6014418 86FDB7F5 89E0E241 C2B10FFF 1591B3A1 333A1914

59208402 B44839B4 BC470546 6641C8FF 798D7724 6190E534 23204A0F D98FB66

A8133C1D A01595D4 851F43DB DA704A0A 2E5A8AF4 894F07A 3219C10C 970B620B

voltar ao diagrama

Blocos Criptografados:

11538382C33F7526
ED2FCDD09FA57753
4C20FD78D2CB5D34
46A1892E540E31F6

voltar ao diagrama

Texto Criptografado:

11538382C33F7526ED2FCDD09FA577534C20FD78D2CB5D3446A1892E540E31F6

voltar ao diagrama

7.5.2 Decriptografando com Blowfish no Laboratório Virtual

Como já possuímos a Chave Secreta devemos passar direto à decifragem, para tanto devemos escolher "Decriptografar utilizando" o algoritmo "Blowfish", como mostra a figura abaixo:

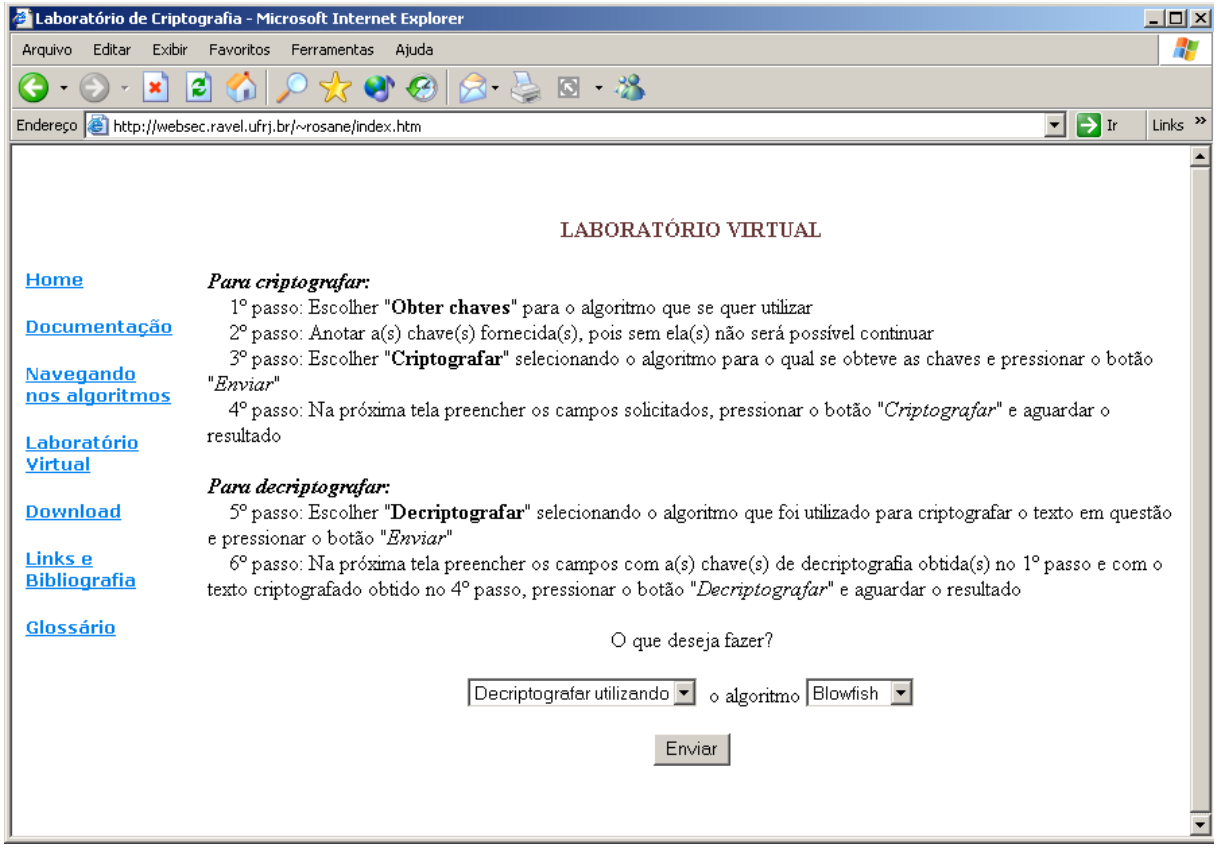


Figura 7.14- Optando pelo processo de decifragem

Semelhante ao processo de cifragem ao pressionarmos o botão Enviar uma nova tela se abrirá, como mostra a figura a seguir.

Nesta tela será possível digitar a Chave a ser utilizada para decifrar a informação (note que como Blowfish é um algoritmo simétrico a mesma chave utilizada para cifrar deve ser utilizada para decifrar - é a chamada "Chave Secreta").

Também será possível digitar a informação a ser decriptografada, ou seja, o texto ininteligível obtido através do processo de cifragem.

Semelhante ao processo de cifragem ao pressionarmos o botão Enviar uma nova tela se abrirá, como mostra a figura a seguir.

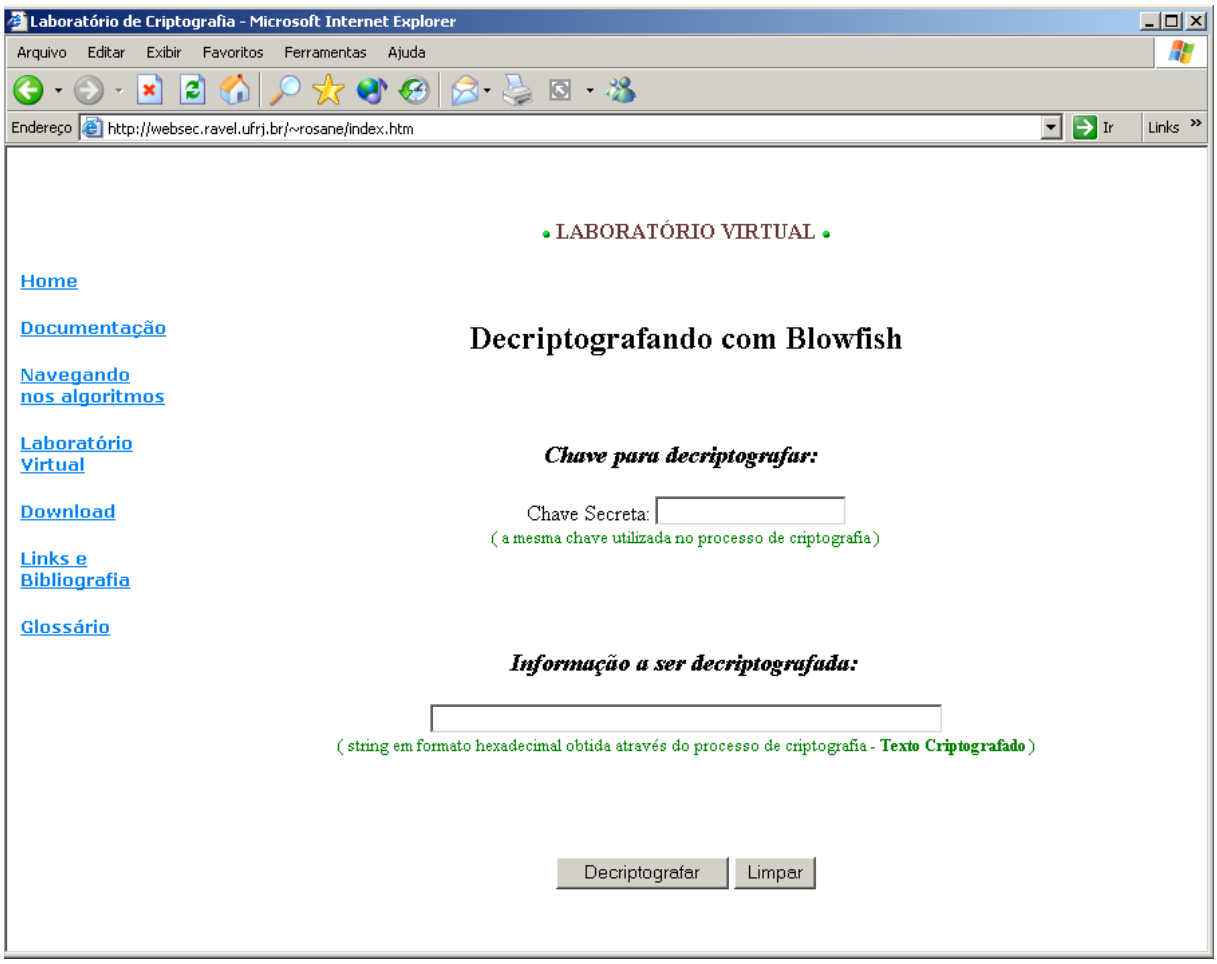


Figura 7.15- Inserindo dados para decifrar

Nesta tela será possível digitar a Chave a ser utilizada para decifrar a informação (note que como Blowfish é um algoritmo simétrico a mesma chave utilizada para cifrar deve ser utilizada para decifrar - é a chamada "Chave Secreta").

Também será possível digitar a informação a ser decriptografada, ou seja, o texto ininteligível obtido através do processo de cifragem. Após o pressionamento do botão "Decriptografar" as informações serão processadas pelo Matlab. Este processo poderá levar alguns minutos, dependendo do tamanho da informação a ser decriptografada.

Após decorrido o tempo necessário, um Pop-Up será exibido para o usuário com o resultado da decifragem.

Neste Pop-Up, além da informação decriptografada, serão exibidas também outras informações, como o texto criptografado, a chave utilizada bem como uma figura ilustrativa do processo de decifragem. Esta figura possui áreas clicáveis que revelará alguns detalhes do processo, como por exemplo os blocos em que foram divididos o texto criptografado antes de ser decriptografado, as subchaves geradas no processo, etc.

A figura a seguir mostra um Pop-Up resultante no qual foram utilizadas as seguites informações:

Chave Secreta: 2AF349CA977BE684

Informação a ser Decriptografada:

23C0D73EB929E976C73EBDBA26C489E5DE39F4D9E78FD7AB

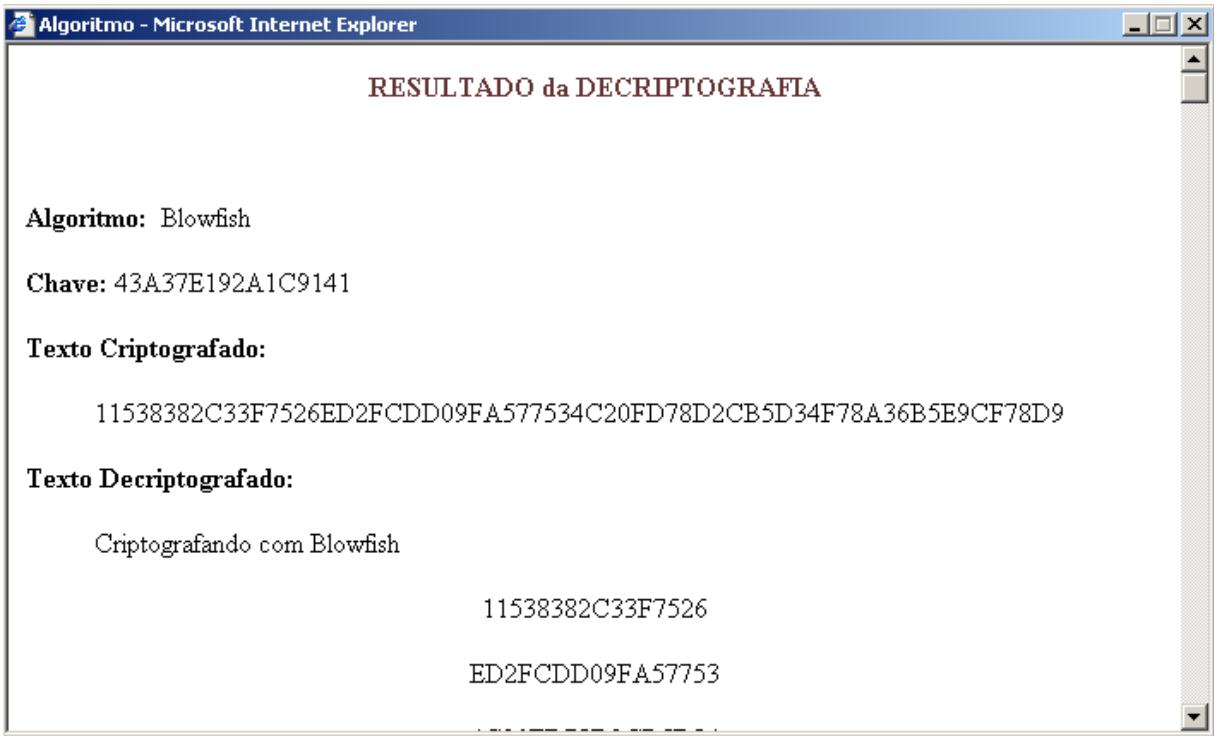


Figura 7.16- Resultado da decifragem

Segurança:

Até hoje não foi comprovada nenhuma forma de ataque bem sucedida ao Blowfish.

Referência: Applied Cryptography, Second Edition: Protocols, Algorthms, and Source Code in C, by Bruce Schneier

Detalhes da Decifragem:

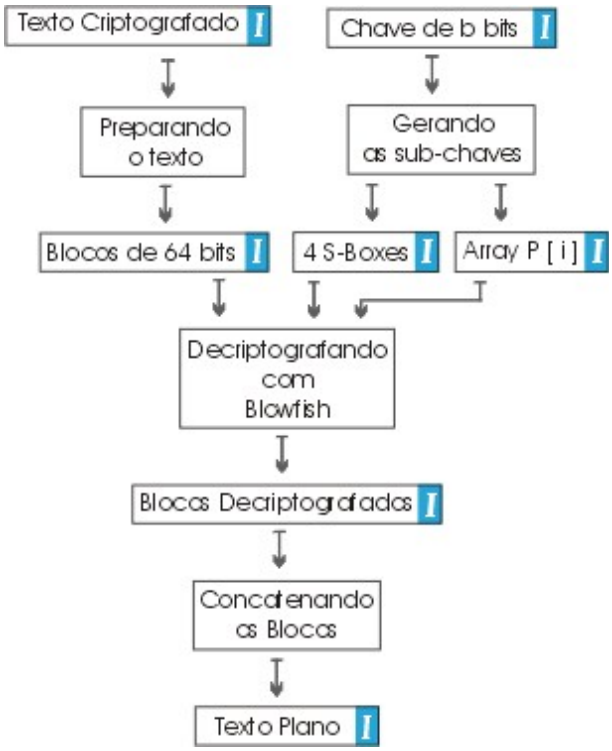


Figura 7.17- Diagrama da decifragem

Texto Criptografado:

11538382C33F7526ED2FCDD09FA577534C20FD78D2CB5D3446A1892E540E31F6

voltar ao diagrama

Blocos de 64 bits:

11538382C33F7526
ED2FCDD09FA57753
4C20FD78D2CB5D34

46A1892E540E31F6

voltar ao diagrama

Chave Secreta:

43A37E192A1C9141

voltar ao diagrama

Array P:

P1:	62789E1A
P2:	54B4E422
P3:	D49E1D61
P4:	2F60517
P5:	173646E1
P6:	7058E1FD
P7:	972352BF
P8:	7B21E225
P9:	1E65A19E
P10:	4A38A608
P11:	BEEFB5C2
P12:	DDD2BA40
P13:	170E8D5D
P14:	45481D00
P15:	4CEA28D5
P16:	16D5C7A3
P17:	1078FF8D
P18:	69A6B917

voltar ao diagrama

S-Boxes:

S-Box 1:

34BEF543 5E4F848C EC9A3819 B546EB4D 1F1113F2 DFC70BD5 1D964E69 C25E2416

1102E4AE C491A9FB 59A4BA9E A8AD7A3D B18EF7 2FACA1D3 6536B040 BBDADB63

7DEECE60 E1174015 7E16A42F 4BAC406D C0A11F33 C975B243 E9EC6825 96E30CFB

E9A6DC24 6F822EA5 6C30B29D 3932D348 B84475E8 D8DBE49C E48BBBD8 F25849C2

3E0C5A9D 5868D836 D431025A B0910F4F E98940C4 CC5B28F5 8E880E6B FF67102

3971A1BB 3448024D E587C909 7547A49E F073A6BB C69226DB 17F8C39D E6954FB7

286E9C46 987FBB3E C7A2D386 A3F3B20B 55CE5B93 8E87B917 78D43AC3 EF2761BE

292BAF6B D661E2BC E3FAB49C B41D26A8 11C281DE 4ECCBAB3 E483D329 76A5E4DB

F44F8178 56CC5F12 64F5BBCA D3C568A5 E9683A81 AB8A6689 60B80A90 1F2EA8D

2471EF38 2CD4349A 4ADC5E9B B56ACD89 E4277744 3B1E946B E960A780 867A22F1

92438886 2920CD3D 29EC5EE1 9CFD75D3 37C29D8F DFEEE7F9 1E10B487 714AB6A6

BFEF1EE9 E1AE98E6 C657AE91 B75147A7 EF011C59 A540B1B5 39783AA3 F0C1D26F

F527EC4 469D3215 97F19C4D AA367CE7 41B4F2FC 4C0D1D45 49F400D0 884C86AC

A82939A6 A30616EA 90C32A5A B7667A03 9A585FB6 1A3823E3 BEEFD38F 3F135296

FAC34BF5 B990B99B B2C4348A 4A2B171E A4535F44 EF958BB2 699BB7C8 DB1CF47

34B93E5E 896AF6BA 2869D600 25BC736 8D11D928 AB354EAA F5C3F3F0 D5D8EF7E

6DF00464 8A8FFB49 E1C1F00E 7856FC6 D1458F9B 459BCA6E 57CF773F C97721F7

55802D8F A54BA7E1 468CD8B4 F04C9400 6103DFCE DDE075B 36AF8461 9172712D

EE66B08E D9CA05CD DD4878EF D2E9EA69 3D7079DA 28D90F0F F7D807B1 C7FCC4E9

8A908CB4 5FC77147 318F11E8 DDB540A3 4D9D125B DA24A814 241CB8D7 9EDCA26D

5905CB1A 4B4BA800 8FA2E6C8 12662270 15B93B48 86B7A5A5 137422EE 6FA0A7C2

86796D7D 7E03DB7E D7C83A4B 2C5E0F8A 8457F5DB 4C570C1E F1D05570 A27052C7

CD18B703 979B7104 D30331C5 A41AE956 71D72DC2 BAFD0D5E C7844DB2 C3095C3B

1A798D60 5CF55FF 2B19A34 344B30BF D6CEC0F0 F2D789DF 6FA39F46 7F882D53

BC964F0A 192D7F4A 81A0E544 FC4DF33B 3D0EC0EC ED647B66 40C27B77 B5ACD531

C0D7B0D7 E4138731 C9B4F71 8EBCDF20 59BD5615 4D094398 E4AFF438 67F35500

A1FF7FAE 9EF558EB C7F97E46 C1EE5DF4 ABCE662D 14A989 4DF48E48 C7A74822

6095F926 DFFA224F 98C8205A 8F621AB4 D911AE7E 522BD78B FEEBBE28 3B84A9A4

A8F55E82 9051070B FC42F98C 919381DC 7B8FF91C DEA9CE85 6725B796 F3D67F14

13F32C74 B1F94377 D5185552 99AB4619 25162003 B37F5E09 39FD4D87 B0A5DFD0

EEE627AC 7736DA6C F3EE5C8E CB230430 34CCC444 CF31C62C CEBE532 15DC6090

DE0BF8F5 379BA0BC 12F08C82 E167A7B2 EBA48889 A8E94CB8 C35B86B5 7DB233EA

voltar ao diagrama

S-Box 2:

7AD82439 EAAE0E90 16D889A5 52771C0 965180C EBA46202 9D162A2C 18A7E6B1

AC8DCB5A EB4A4F76 AACEE06D 251E92D9 666D5B27 253FE2D9 B00D26E4 26490626

175EB6A8 9BB2E4F7 56ED9EAE 88CEA47B 9B22B216 86570BE2 2EECB865 B87137C1

AAD5C16B DCE218E5 3F6D9962 ED86C660 3CF98AAD F77DFFC8 EABEF524 3D522BE1

5DADC6D0 426CC451 8BA160F1 E775E60E A1B98166 D55ED25 813ED286 EC52E01B

3F7D33AD B97D1A31 900280FF 99143344 94F13828 5B2C1C4A 33C1F999 93B30A56

747BCD0C DF03C77E E37B39B5 D75931C9 8CD276EE 6D39A1E1 2533BF3D 7A7F8B53

63F8172F 7B49E5C 97585E07 A3A9F4D4 162C9718 326C0757 33B3D8F7 61B3F912

4D373566 5DED3CB6 D4EB4078 E91C3E0D B23081D3 4BF2EB8D C4439699 3EE3FA9D

1361A30C BE071B72 DDDDEBD6 142C239F 4D825F00 A24CBA20 4766A044 1DFF47F1

5FEA07CF D0F6BA45 334C98D6 2F042A19 C4DCA55E FF64E710 74B55245 CEA98113

ACB9311C B40A8CFB 10BEDCA7 50534C21 E22AA80B 711EF208 B8AFA143 7B1154E3

BBC0800C 8D0AC0E3 AA7E7EF 31C7BF71 7730A46 67DD7393 953B436A 1E323105

33B04395 B6DD1CD 555DC148 D6094B45 94E5204C DEE4CFD5 2FFBD71E F685145D

34E0D3A7 AE6AADBE 14A3C730 1F01BD55 55ACA948 91AE647C EEF0604 100181D8

386DFCB 40393936 1D96ABFA 22D756A5 65FF3D10 E5652409 280DC531 18B58EA9

C04CD9B2 E58D9AAD 705E3D67 AE84D23B 1373BF38 ADB87146 1C45BB5 608E0B1C

27DA974A 15D676AF 7DA761C7 5C9447AD B341966D 70CE25E 62166F1C 2BC5ED39

67C5402C 1AAB392B 9262260E B84E1DFE FD0187BD 2E935950 9B890EBC DF402070

8A8265CD 4C00FAF BF3D7CEE F7FE82E6 9983F7B8 BF0D1DF8 1040E993 5BC00665

792BBDFB 95516DB7 CDEF5582 72688A95 BADA85C2 EBCB8490 218D0E77 4A693626

E76CE6E3 B01993E4 A05E8939 782F5BA3 429ED3A1 22D576DB 380C921A BF3467F4

293090C2 883EE920 2AF2B8A5 66159D11 139C4996 8837B278 3BA1AF22 21F25F5F

AB9791FF 7733B92F CA989A0E 76336B17 7CF38FE7 219D60AC B506FB5 70B1E8B2

1001D965 B392F4A2 ABF76299 B1437611 CDA17415 804267B7 47346159 2E2454DE

99157FCD 52EDC410 C39439E5 361C46F9 5455D9F7 54594C5B 10417587 6BB36D46

FD5C8AE8 D4F43B2A 49707375 E4612428 B0F3C05E 91A72279 2E06E6C9 67FD5C00

CEFE9697 4A393F6F 57C7ECE C57ED843 4CF30B19 A13FF2F1 8F5470FA FF7C2FEA

E60D59FC CB900E3D 6ABD54DB E684DC5E ED26E17B 6D05A00E BE0021C2 DE6F03

FFEE5CEE 667DFA85 691F2830 EE44265 D36F88C0 FAABAF 1C586E7C 8755D6ED

3D9E79F2 D47C928A 9AA5293C E77FBAB A5C0C5C7 7737A19 D689A024 85F76585

D3D446A2 EB4E124E D8DC5151 E7B8C046 6FEABBD3 E9394644 2FA41E89 35BCF345

voltar ao diagrama

S-Box 3:

58F3D85F 21CB6C22 968702B3 5AAFEC2A E86EE56D 42A7F59B 78DDBA74 BA8A66F

BA72780B 5FB36D84 7F25314A FAA68242 D7A6D555 83EA6CA8 E9E63235 1813BA17

58124362 A754156B AC6509A3 A78C11B5 FAC285FC 28BB0B90 87206F27 BA788957

5BF0810F BF12FC91 34397AAB F584588D 3DCB3A80 19C9DD5A 512F1BBC D10EB0BF

FF10416C 7BDF315D AF360BA4 128CD76 687E0349 286B8576 27D439A AE59C4D3

B4560F2B A03B4235 3F5E39AF 4E895080 2012D835 62258DAA 90B49F92 AEFBEE3B

F642400C B1F1E629 321F68A7 62D2B3D3 1755CA13 51FB1632 5CF48F9 D0A378F2

E7AFE5AE 30C3ADC9 BAA9F631 C92D178 F3B817C3 E5630296 DA72C366 F43511D9

365CA368 8192F284 B617013C A82E0026 46096502 F8CA5187 AF45E401 D6C7AEAF

8AA7506 98F7CE0A 55E048CA 6E902183 B9558292 4445F415 B409F1B9 3D326446

FD959C28 446ECEB7 FAC59B60 9E39FCC3 56557BF8 80A66B68 5133A88 B75AC167

507A7D0B D695B01B 9A889435 CA2D27F4 3A3ABE1A BF1E9E85 8DE70549 68A936B2

9A8E9C22 C8E61C11 A58B3A57 4BF08B8F EF30E230 F4B89937 F9A16D7F 399C9432

485AC828 32D60C83 5CC02015 C159EF26 3D047B9D D8959853 AD28269D 29CDB635

73D512A2 205C6453 1B7790D7 4CC1BA40 34CBF250 1087935 76864E4C 1647CC79

17BBC2D8 DD489045 BAFE89CD 17B8EC09 7A5C6B2 3D0428A7 D1A70C2A 5C70A569

2EC8D0C F9716915 37410CA5 F5C4913A DC21A99E 262D9C16 E0881800 3EEC2FF6

95DE086B 5F3D5CE6 1EE63F1B E7A3C2FE 18CD11A1 B5A7F117 CE509136 6206B768

1FEC2E92 B2636C2 9AF9F335 9CF63F21 724F9345 2DFB5D4A DC6F4B22 C02FC8C6

A4AC334C 2DFFBB34 3DC764F2 882D71B2 61DD8C1D 73C72B8C FB798E4B 9B9BE74E

E8438958 DBA1B45A B77AC604 6BB6E978 715AD846 B4EF4915 8CD851C1 33896D8B

36185A38 A5A264FF 584C01E1 90A81A63 D9EBF5C8 83CB20C0 6E561FF3 7D7A2931

3D5FDB9A F8B7FF37 8B52F0F7 D39B6943 5B63274F 7F8C201B 6480289C 8A1B4D05

9C34408A BB8E1520 7A9801FE 444EEA34 DD7D0A45 69ABF2C1 7201F326 B2FF6616

93B3086A 5485F6DB BF856FFF 8FD61D04 BA645CB0 8A3E40F4 45503322 3C1C81D1

B5C3B046 A602C81B DE40457F 46BEA746 308BEE62 FE69BE70 AD657E5A B622AED1

D9C4A1F7 FE32D988 10A437A4 D9792517 BCCF3CA8 F7F96621 E8509B3F BDF1F03

B4A6A09B AF466013 37341A59 BF362DBB 66607A07 6E961B73 70BA646 2E9452E8

BA919B4D 91C91136 6F8C4F27 800583C4 57C0D4F8 44ACEF1D EC1D6797 95C2E083

AEEEE807 5C13D91 ED772058 4F560E67 A9B5D3F8 49941770 29B1736B FB699F7F

BBE5B082 844BB77E 48FBCC06 F622E66D E283FB02 F5082613 E677C95D BD9518E4

C5A81E4D 2C351920 571C06EE EE8E3B77 85D58452 E3357123 733BF6F4 D7439B00

voltar ao diagrama

S-Box 4:

C2F65B96 D55D7B20 93D9CB92 E728A974 88D725A3 43BDD2DF 67A07C58 6CFA4C52

728A04AF 56360E3A 1B1BA253 7CBEA775 CC9ADB3 C9FD7E6E C577A15B 424268B7

61A9F6F3 C7B0EFAA A8E7C472 CAD83B5D D892939B 2892164A 547024A1 1DD36A29

4AA1519D 94599C3B FFA9F3BA 6EDF08DA 395780DE 325D15AA B97AF9E2 A67AE05E

F4519181 1ABC373 DD23A4F0 3E22256D 2E11ECDE EFED6EEC D5169A3B 3B90CA28

631F0687 5D7AFC8D B6C43620 77226B09 DDCB6E97 F72BFD08 4AB3B05E 3DAAF7D3

A367E4CD C8FC332B 834E7630 AE8FDC55 613463B8 6D7FD530 56E5F536 DA17AF94

8D13B1B 4B25757E A144B8CB 7E5F874B 183404A9 3FD6FEE8 EEA788C3 9F75BA9C

A71CF5BD 71526676 EC8FC870 FCBD73BF 44C96A37 6042637B A60A2F8D 3BBEDFE9

DA6CA2EF 3D2C3B94 AA696074 46BF4065 A5AAE6B8 7BA51B20 9B8B83FA D8DA1E4A

9B650AB0 ECC2B4D5 9BCB15F1 EC7D5EC3 EAAED0B0 16E7C0B4 40928CE2 EF789CF1

CA760F31 7F974AB2 116E51EF ED2B0697 46A8C482 9659A984 B707F55A 7B0FAF4

446AA381 8DB625A0 ED5B0DB8 2EC20FF7 AB30FEA8 56CBF21 BBE62FC1 DDADD0B2

9DB436D8 69ED54FA 7639D0EA 1108BA2E 5CBAA45D BD43F03D 2AD3564B E427D9F3

86B7223F 1CA1AAC0 1170F625 C2BF6D98 352ABAE4 EF8ECEE3 8B2FAA9C 346C3516

2F7B1BC3 2952E354 EC53127 EEFD6323 434589AF 1224ACE7 78165B08 E914500C

F61077E9 C126D57E 10DBE13C 83D86C05 5CC9FB0A CAF5C035 9F502BF8 7E656509

F707CE0 AA764CB4 2A30B3D3 ED8C2317 5A489347 72AA821B 1EB6FFF1 B39602EA

8D0F547B F56A72A5 760AEACA 8A69D2B7 9F9CDC38 9C6DA0B C7D2E179 878AEEA1

729E76A5 B0E1EC0E 4D2A5111 8818133B A53AB3ED 4C8AA1E1 B6111D3E C3A1C98

7861EF80 AB653AE8 C4A1D584 F5582EB 19687C1F 9FC95592 F60DA9EE C1AAD410

7E43D7A3 CEF3D6F6 78A5E927 F4B5A907 78921155 A36B4D75 59BBAB7A 8EF37716

87F95BD0 E3F56FE5 27149C4C BAE3D258 72539E4B FAD8BD9 9A9EB542 43C2FDB8

F30D7707 A762E208 10207466 DAD8341A B16F60DC 54115338 46629565 229D0680

5A7663CD 174B49E9 96AE03E1 1E220824 8FAB0174 764F6071 7C5935C7 6FA0373

7F7F3605 1854A0CA BD94D1AE 4D36C6F3 692408C7 E1C7163C D6111E65 D06CD286

BBC161B5 3358D7B9 3838D62D 90D07E8B 8FD76458 46E80ED3 9E979AFA 9A8A32B3

A332B300 86950957 A9B5FEEB 5335C19C C5939D89 102830DE BEAF06BA 9FFD39E0

64C16C6B E2658DB7 BEC1956B F7CAFFE2 75E1F700 7E4001E5 F07F6D41 A448F52A

6FCFD7A2 830B9330 A6014418 86FDB7F5 89E0E241 C2B10FFF 1591B3A1 333A1914

59208402 B44839B4 BC470546 6641C8FF 798D7724 6190E534 23204A0F D98FB66

A8133C1D A01595D4 851F43DB DA704A0A 2E5A8AF4 894F07A 3219C10C 970B620B

voltar ao diagrama

Blocos Decriptografados:

43726970746F6772
6166616E646F2063
6F6D20426C6F7766
6973682E00000000

voltar ao diagrama

Texto Decriptografado:

Criptografando com Blowfish.

voltar ao diagrama

8 RC5

8.1 História

RC2, RC4 e RC5 - Ronald Rivest, da RSADSI (RSA Data Security INC.), projetou essas cifras com tamanho de chaves variável para proporcionar uma criptografia em alto volume que fosse muito rápido.

Um pouco mais rápido do que o DES, essas cifras podem se tornar mais seguras escolhendo-se um tamanho de chave mais longo, eles permitem chaves com o tamanho entre 1 e 2048 bits.

O RC2 pode servir muito bem como um substituto para o DES, pois ambos são cifras de bloco.

O RC4 é um outro tipo de cifra conhecido como cifra de fluxo. Em software, o RC2 é a proximadamente 2 vezes mais rápido que o DES, ao passo que o RC4 é 10 vezes mais rápido que o DES.

O RC5 foi publicado em 1994 e permite que o usuário defina o tamanho da chave, o tamanho do bloco, e o número de passos pelo algoritmo de encriptação.

8.2 Características

•RC5 é um algoritmo de Ron Rivest (The RC5 encryption algorithm, Fast Software Encryption, 2nd. International Workshop, Lec. Notes in Comp. Sci. 1008, pp 86-96, Springer-Verlag, 1995).

• RC5 foi projetado para qualquer computador de 16 ou 32 ou 64 bits. Possui uma descrição compacta e é adequado para implementações em software ou hardware.

•RC5, assim com o DES, possui várias iterações e as várias subchaves são utilizadas em uma função de iteração. Ao contrário do DES, o número de iterações e o número de bytes na chave são variáveis.

•RC5 baseia-se na operação de rotação (i.e., 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.

•Após alguns anos de análise dos especialistas sabe-se que após 8 terações todo bit da entrada legível afeta pelo menos uma rotacão. Há uma criptanálise diferencial para o RC5 com 64 bits de entrada que necessita 224 textos legíveis escolhidos para 5 iterações, e 268 para 15 iterações.

•RC5 é considerado seguro após 6 iterações contra a criptanálise linear.

8.3 Parâmetros do algoritmo

•**w** = 16,32 ou 64 bits, é o número de bits em cada variável (i.e, cada posição de memória); no nosso caso usaremos w = 32 bits;

•**r** é o número de iterações (rounds);

•**b** é o número de bytes na chave.

•O algoritmo genérico é representado por **RC5-w/r/b**. De fato é uma família de algoritmos.

•No RC5-32 o autor recomenda 12 iterações, e para o RC5-64, 16 iterações.

•A **entrada** é de 2w bits, e a **saída** também.

•As duas metades da entrada A e B são alteradas a cada iteração.

•São usadas 2 subchaves antes da primeira iteração, e 2 subchaves em cada iteração, tendo-se no total 2r + 2 subchaves.

•A chave **K** é de **b** bytes, da qual são geradas as subchaves $\mathbf{K_j}$ de **w** bits cada, como descrito a seguir.

8.4 Funcionamento do algoritmo

Seguiremos a recomendação do autor, sendo assim, como estaremos utilizando w = 32 bits, utilizaremos r = 12 bits

8.4.1 Operações básicas

As operações básicas usadas são todas sobre operandos de **w** bits, e são:

1. $v \boxplus u$ é a soma dos inteiros **v**, **u** de **w** bits, resultando um valor de **w** bits (i.e., soma mod 2^w);

2. $v \boxminus u$ é a subtração dos inteiros **v**, **u** de **w** bits, resultando um valor de **w** bits (i.e., subtração mod 2^w);

3. $v \oplus u$ é o ou-exclusivo (XOR) de **v**, **u** de **w** bits, resultando um valor de **w** bits;

4. $v \ll t$ é o deslocamento circular (i.e. rotação) de **t** posições para a esquerda dos bits em **v**. Se **t** exceder **w** bits, pode-se considerar os **w** bits menos significativos sem alterar o resultado.

5. $v \gg t$ é o deslocamento circular (i.e. rotação) de **t** posições para a direita dos bits em **v**. Se **t** exceder **w** bits, pode-se considerar os **w** bits menos significativos sem alterar o resultado.

8.4.2 Obtenção das sub-chaves

1. Seja **u** = **w**/8 (i.e. número de bytes em cada variável de **w** bits, e.g. **u**=32/8 = 4 para **w** = 32 bits);

2. Seja **c** = [**b**/**u**] (i.e. número de posições de **w** bits ocupados por **K**, (por exemplo, **c** = [10/4] = 3, para **b** = 10 bytes);

3. Se necessário, preencher **K** à direita de **K**[**b** - 1] até completar um comprimento total em bytes divisível por **u**.

para j = b, b + 1, ... c x u - 1 faça: K[j]=0;

4. para j = 0, 1, ... c - 1 faça:

$$L_j \leftarrow \sum_{t=0}^{3} 2^{8t} K[j \times u + t]$$

i.e. preencher os u bytes, da direita para a esquerda, de cada **Lj** com os bytes **K**[0], **K**[1], **K**[2], ... **K**[c x u - 1];

5. $K_0 \leftarrow P_w$; (veja o valor de Pw na tabela a seguir)

6. para **j** = 1, 2, ... 2**r**+1 faça: $K_j \leftarrow K_{j-1} \boxplus Q_w$; (veja o valor de Qw na tabela a seguir)

7. $i \leftarrow 0$; $j \leftarrow 0$; $A \leftarrow 0$; $B \leftarrow 0$; $t \leftarrow \max\{c, 2r+2\}$

8. para **s** = 1, 2, 3, ... 3**t** faça

{ 1. $K_i \leftarrow (K_i \boxplus A \boxplus B) \lll 3$; $A \leftarrow K_i$; $i \leftarrow i+1 \bmod (2r+2)$

2. $L_j \leftarrow (L_j \boxplus A \boxplus B) \lll (A \boxplus B)$; $B \leftarrow L_j$; $j \leftarrow j+1 \bmod c$

}

9. A saída é K_0, K_1, K_2, ... K_{2r+1}

Onde são constantes baseadas na representação binária do número **e** e **phi**.

w	16	32	64
Pw	b7e1	b7e15163	b7e15163 8aed2a6b
Qw	9e37	9e3779b9	9e3779b9 7f4a7c15
Valores em hexadecimal			

Tabela 8.1-Valores possíveis para P_w e Q_w

8.4.3 Cifrando um bloco

Primeiramente separamos do texto plano um bloco de 64 bits. Se o bloco for menor do que 64 bits (o que geralmente acontece ao último bloco), preenchemo-lo com zeros.

1. Dividimos o bloco de 64 bits em duas metades A e B, cada uma com 32 bits.

2. Feito isso deve-se processar o bloco de dados A B com as 2r+1 sub-chaves como será descrito abaixo, são r passos. Começamos em i=1.

a. Primeiro fazemos:

$A \leftarrow A \boxplus K_0$;

$$B \leftarrow B \boxplus K_1$$

b. para j = 1, 2, 3, ... r faça:

$$A \leftarrow ((A \oplus B) \ll B) \boxplus K_{2j}$$

$$B \leftarrow ((B \oplus A) \ll A) \boxplus K_{2j+1}$$

c. Unimos novamente os blocos de 32 bits L e R formando o bloco criptografado de 64 bits.

8.4.4 Decifrando um bloco

Primeiramente separamos do texto criptografado um bloco de 64 bits.

1. Dividimos o bloco de 64 bits em duas metades A e B, cada uma com 32 bits.

2. Feito isso deve-se processar o bloco de dados A B com as 2r+1 sub-chaves como será descrito abaixo, são r passos. Começamos em i=1.

a. repetimos o cálculo abaixo r vezes, iniciando em r e terminando em 1, ou seja, na ordem inversa da cifragem:

$$B \leftarrow ((B \boxminus K_{2j+1}) \gg A) \oplus A$$

$$A \leftarrow ((A \boxminus K_{2j}) \gg B) \oplus B$$

b. finalmente fazemos:

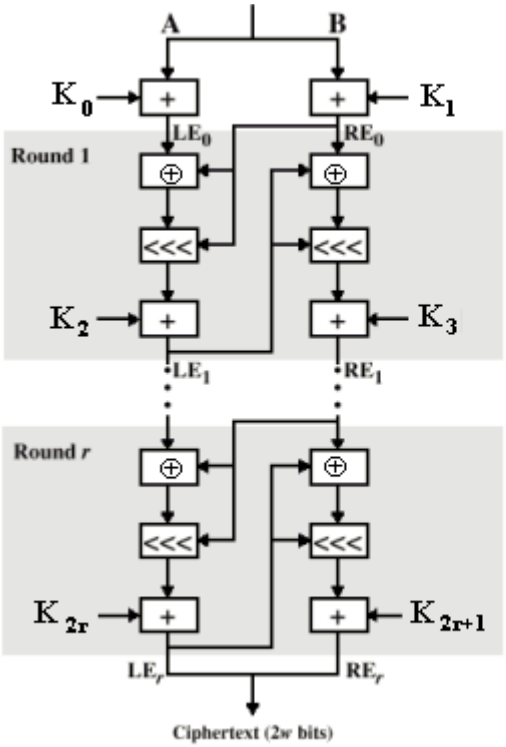
$$A \boxminus K_0$$

$$B \boxminus K_1$$

c. Unimos novamente os blocos de 32 bits A e B formando o bloco decriptografado de 64 bits.

8.4.5 Esquema gráfico de funcionamento

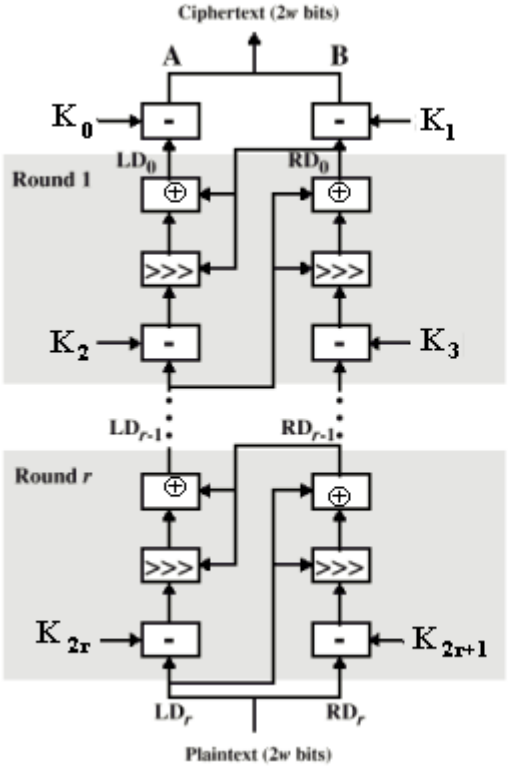
8.4.4.1 Cifrando



$LE_0 = A + K_0$
$RE_0 = B + K_1$
Para i = 1 até r Faça
$LE_i = ((LE_{i-1} \text{ xor } RE_{i-1}) <<< RE_{i-1}) + K_{2i}$
$RE_i = ((RE_i\text{-}1 \text{ xor } LE_i) <<< LE_i) + K_{2i+1}$
Fim Para

Figura 8.1-Esquema gráfico da cifragem utilizando o algoritmo RC5

8.4.4.2 Decifrando



Para i = r até 1 Faça
$RD_{i-1} = ((RD_i - K_{2i+1}) >>> LD_i) \text{ xor } LD_i$
$LD_{i-1} = ((LD_i - K_{2i}) >>> RD_{i-1}) \text{ xor } RD_{i-1}$
Fim Para
$B = RD_0 - K_1$
$A = LD_0 - K_0$

Figura 8.2-Esquema gráfico da decifragem utilizando o algoritmo RC5

8.5 Implementação

8.5.1 Requisitos de Software

•Microsoft Windows 95, Windows 98 (original and Second Edition), Windows Millennium Edition, Windows NT 4.0 (com Service Pack 5 para correção Y2K ou Service Pack 6a) ou Windows 2000

•MATLAB 6.0 (R12) ou 6.1 (R12.1)

8.5.2 Requisitos de Hardware

•Pentium, Pentium Pro, Pentium II, Pentium III, Pentium IV** ou AMD Athlon

•64 MB RAM (mínimo), 128 MB RAM (recomendadop) adaptador gráfico de 8 bits e monitor (para 256 cores simultâneas)

8.5.3 Descrição dos módulos principais

O RC5 é composto de vários módulos, os quais devem estar no diretório de trabalho do Matlab.

Os módulos principais são: RC5Cifra e RC5Decifra

RC5Cifra

Este módulo é o responsável pelo processo de cifragem propriamente dito..

Sintaxe:

```
>>[MatrixBlocosHex,SH,CriptoMatrix,TextoCripto] = RC5Cifra(TextoPlano,Chave)
```

Onde:

TextoPlano é o texto que se quer criptografar.

Chave é chave secreta fornecida pelo usuário.

MatrixBlocosHex matrix onde cada linha corresponde a um bloco a ser cifrado.

SH Matrizes contendo as S-Boxes criptografadas

CriptoMatrix matriz onde cada linha corresponde a um bloco criptografado

TextoCripto texto criptografado em formato hexadecimal

RC5Decifra

Este é o módulo responsável pelo processo inverso ao de cifragem.

Sintaxe é:

>>[MatrixBlocosHex,SH,CriptoMatrix,TextoDecripto] = RC5Decifra(TxtCrpt,Chave)

Onde:

TextoDecripto é o texto recuperado a partir do texto criptografado.

Chave é chave secreta fornecida pelo usuário.

MatrixBlocosHex matrix onde cada linha corresponde a um bloco a ser cifrado.

SH Matrizes contendo as S-Boxes criptografadas

CriptoMatrix matriz onde cada linha corresponde a um bloco criptografado

TxtCrpt texto criptografado em formato hexadecimal

Outros Módulos:

Os módulos acima são formados de outros módulos necessários a sua implementação, na próxima seção veremos em detalhes sua composição.

8.5.4 Descrição dos módulos auxiliares

RC5Cifra

o MaisMod - calcula (a+b) mod n

Sintaxe: [Resultado] = MaisMod(Ahex,Bhex,n)

o RotLeftHex - faz um deslocamento circular de n posiçoes para a esquerda

Sintaxe: [Resultado] = RotLeftHex(NumHex,nHex,w)

o HexXor - Realiza a operaçao xor entre dois numeros hexadecimais

Sintaxe: [Resposta] = HexXor(Num1,Num2,w)

o BinXor - Realiza a operaçao xor entre dois strings binarias

Sintaxe: [Resposta] = BinXor(Bin1, Bin2)

RC5Decifra

- o MaisMod - já descrito acima
- o RotLeftHex - já descrito acima
- o MenosMod - Calcula (a-b) mod n
  Sintaxe: [Resultado] = MenosMod(Ahex,Bhex,n)
- o RotRightHex - Faz um deslocamento circular de n (nHex) posições para a direita
  Sintaxe: [Resultado] = RotRight(NumHex,nHex,w)
- o HexXor - já descrito acima
- o BinXor - já descrito acima

8.6 Laboratório Virtual

8.6.1 Criptografando com RC5 no Laboratório Virtual

Primeiramente geramos uma chave válida para o algoritmo, para tanto devemos selecionar "Obter chaves para", escolher o algoritmo "RC5" e pressionar o botão "Enviar", como ilustrado na figura a seguir:

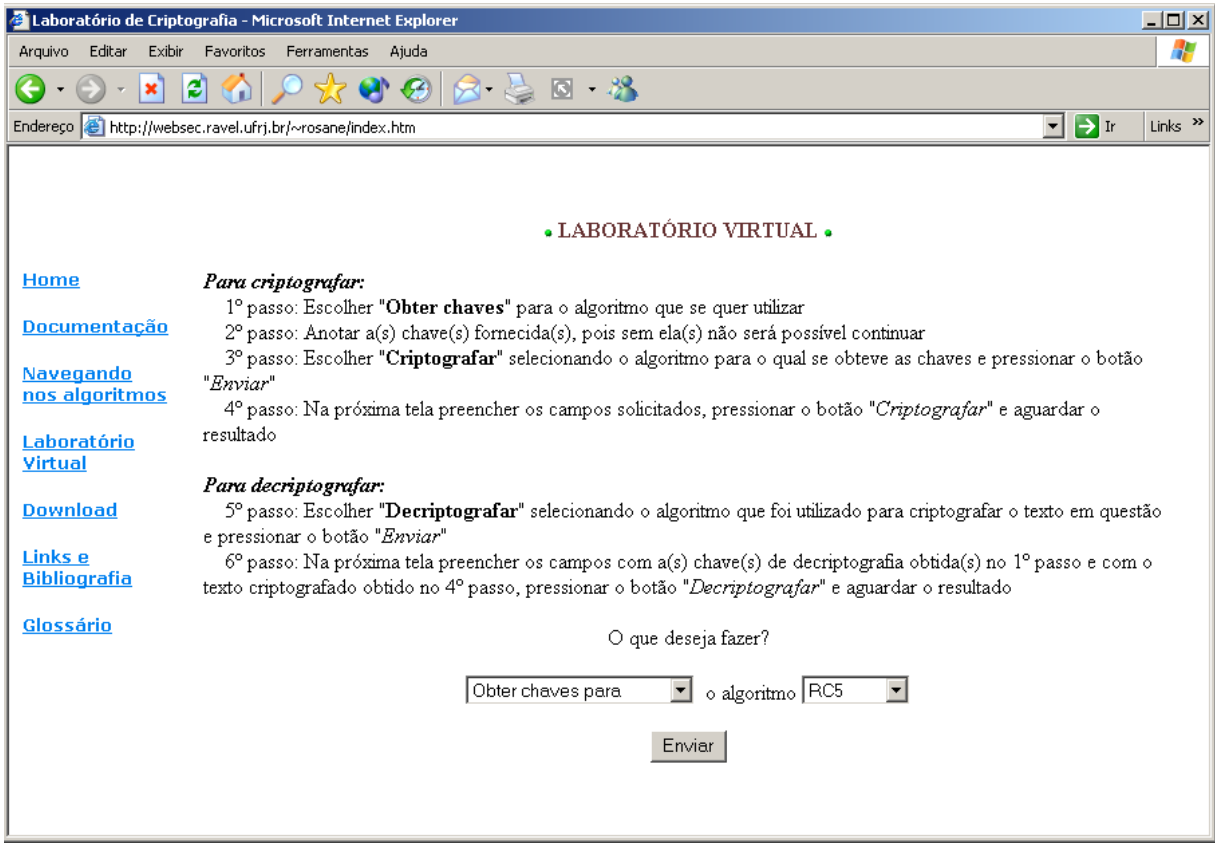


Figura 8.3- Obtendo uma chave válida

Após alguns segundo um Pop-Up como o mostrado na figura abaixo sugirá. Ele contém a chave que poderá ser utilizada no processo de cifragem.

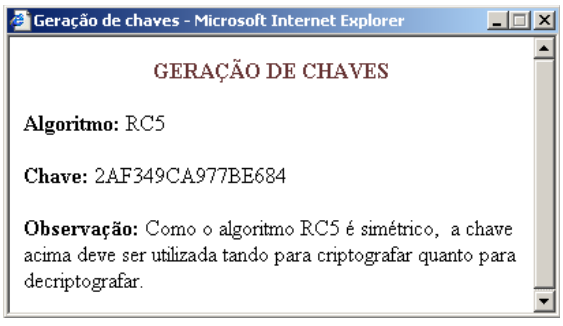


Figura 8.4- Resultado da geração da chave

Após obtida a chave, deve-se escolher "Criptografar utilizando" e selecionar o algoritmo "RC5" como mostra a figura a seguir:

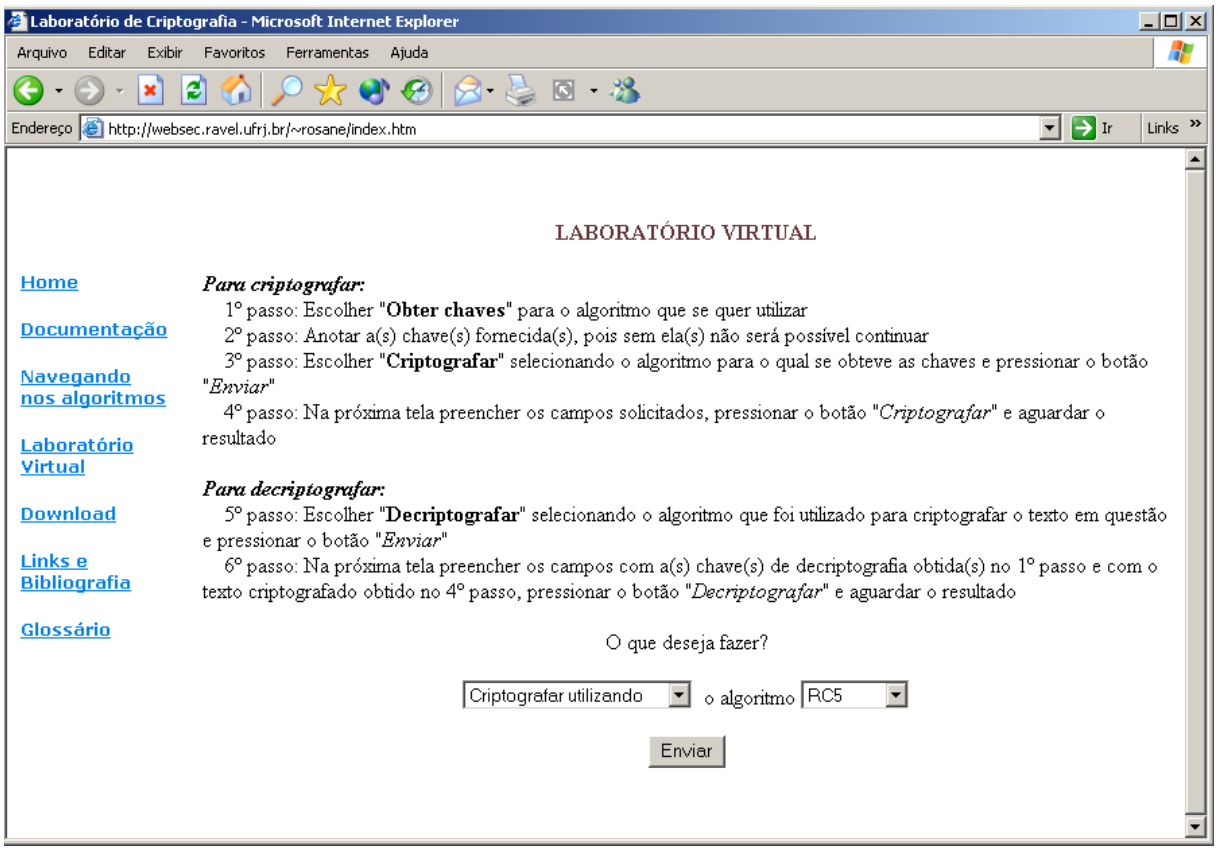


Figura 8.5- Optando pelo processo de cifragem

Após pressionar o botão "Enviar" uma nova tela será apresentada, nela será possível digitar a Chave Secreta (obtida no passo 1) e a informação a ser criptografada. Como pode ser visto na próxima figura.

A informação a ser criptografada pode conter quaisquer caracteres alfanuméricos inclusive caracteres de pontuação.

Após preencher estes campos deve-se pressionar o botão "Criptografar".

Caso algum erro seja cometido, pode-se utilizar o botão Limpar para remover todas as informações digitadas.

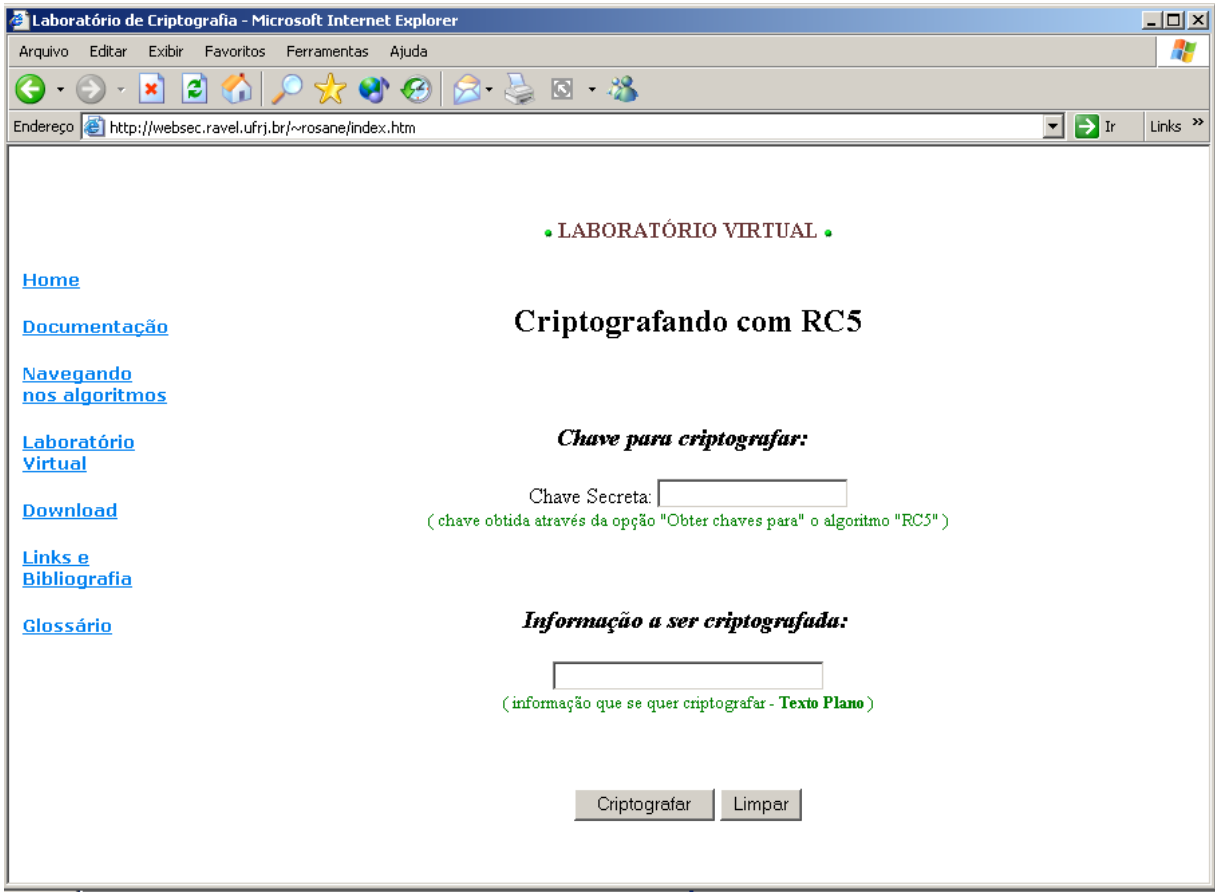


Figura 8.6- Inserindo os dados para cifrar

Após o pressionamento do botão "Criptografar" as informações serão processadas pelo Matlab. Este processo poderá levar alguns minutos, dependendo do tamanho da informação a ser criptografada. Após decorrido o tempo necessário, um Pop-Up será exibido para o usuário com o resultado da cifragem.

Neste Pop-Up, além da informação criptografada, serão exibidas também outras informações, como o texto plano, a chave utilizada bem como uma figura ilustrativa do processo de cifragem. Esta figura possui áreas clicáveis que revelará alguns detalhes do processo, como por exemplo os blocos em que foram divididos o texto plano antes de ser criptografado, as subchaves geradas no processo, etc.

A figura a seguir mostra um Pop-Up resultante no qual foram utilizadas as seguites informações:

Após o pressionamento do botão "Criptografar" as informações serão processadas pelo Matlab. Este processo poderá levar alguns minutos, dependendo do tamanho da informação a ser criptografada.

Após decorrido o tempo necessário, um Pop-Up será exibido para o usuário com o resultado da cifragem.

Neste Pop-Up, além da informação criptografada, serão exibidas também outras informações, como o texto plano, a chave utilizada bem como uma figura ilustrativa do processo de cifragem. Esta figura possui áreas clicáveis que revelará alguns detalhes do processo, como por exemplo os blocos em que foram divididos o texto plano antes de ser criptografado, as subchaves geradas no processo, etc.

A figura a seguir mostra um Pop-Up resultante no qual foram utilizadas as seguites informações:

Chave Secreta: 2AF349CA977BE684

Informação a ser Criptografada: Criptografando com RC5.

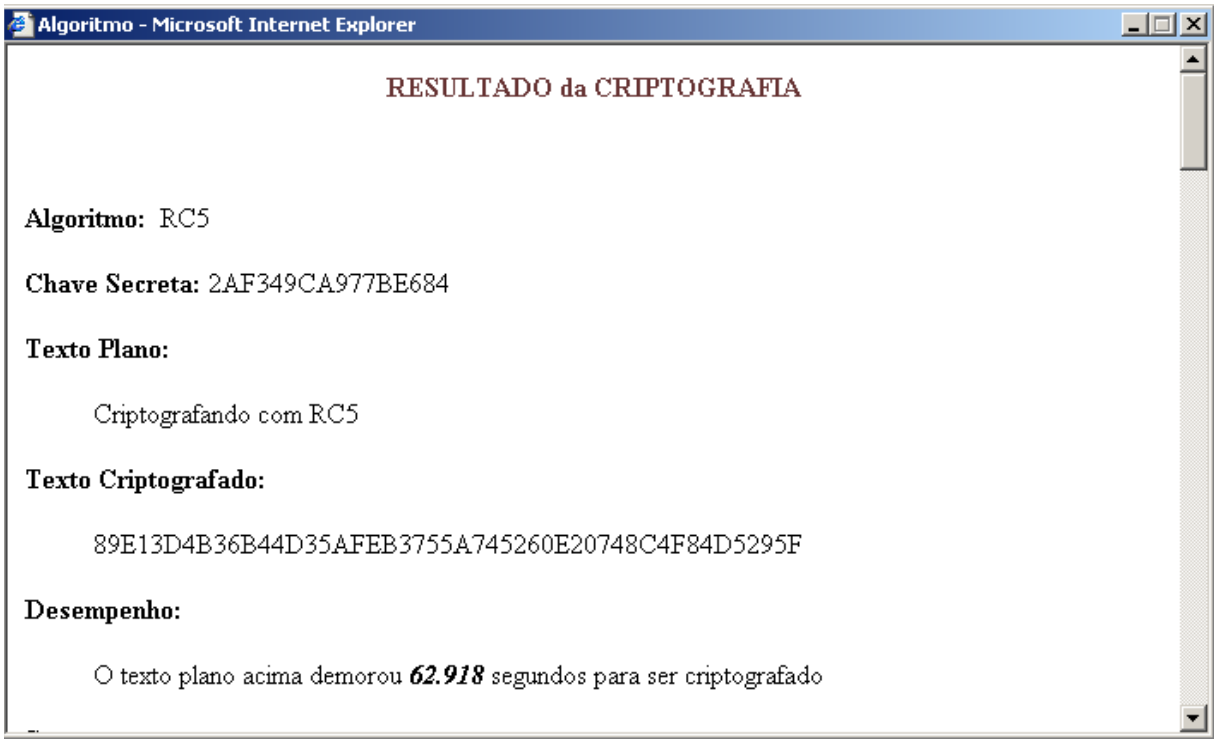


Figura 8.7- Resultado da cifragem

As demais informações do Pop-Up são apresentadas abaixo:

Segurança:

RC5 é relativamente novo, mas os laboratórios RSA já gastaram um tempo considerável analizando sua segurança. Para criptoanalizar blocos de 64 bits criptografados com 12 rounds (a forma de implementação utilizada neste laboratório) seriam necessários 253 tentativas, ou seja 9.007.199.254.740.992 tentativas!

Referência: Applied Cryptography, Second Edition: Protocols, Algorthms, and Source Code in C, by Bruce Schneier

Detalhes da cifragem:

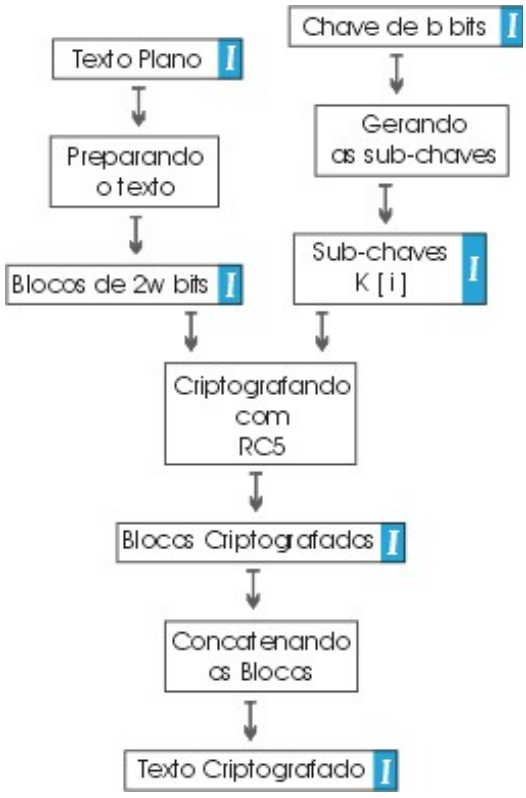


Figura 8.8- Diagrama da cifragem

Texto Plano:

Criptografando com RC5

voltar ao diagrama

Blocos de 64 bits:

43726970746F6772
6166616E646F2063
6F6D205243350000

voltar ao diagrama

Chave Secreta:

2AF349CA977BE684

voltar ao diagrama

Subchaves K:

AE949903
9CE10049
4477EFA0
FE2EC9B8
2A158741
DC242AC1
9DE1BB3F
0273146E
6CB62D2E
BAB6CC74
40F941D3
BF854DC1
72CC90E9
6E356626
EE8D2091
AC20E6FE
27FBDA00
860AF197
856767F8
0ECA05BC
1B39EE7E
2A39EBA4
E8D204B9
D0CE6CE4
2F3CFCB7
102BE160

voltar ao diagrama

Blocos Criptografados:

89E13D4B36B44D35
AFEB3755A745260E

20748C4F84D5295F

voltar ao diagrama

Texto Criptografado:

89E13D4B36B44D35AFEB3755A745260E20748C4F84D5295F

voltar ao diagrama

8.6.2 Decriptografando com RC5 no Laboratório Virtual

Como já possuímos a Chave Secreta devemos passar direto à decifragem, para tanto devemos escolher "Decriptografar utilizando" o algoritmo "RC5", como mostra a figura abaixo:

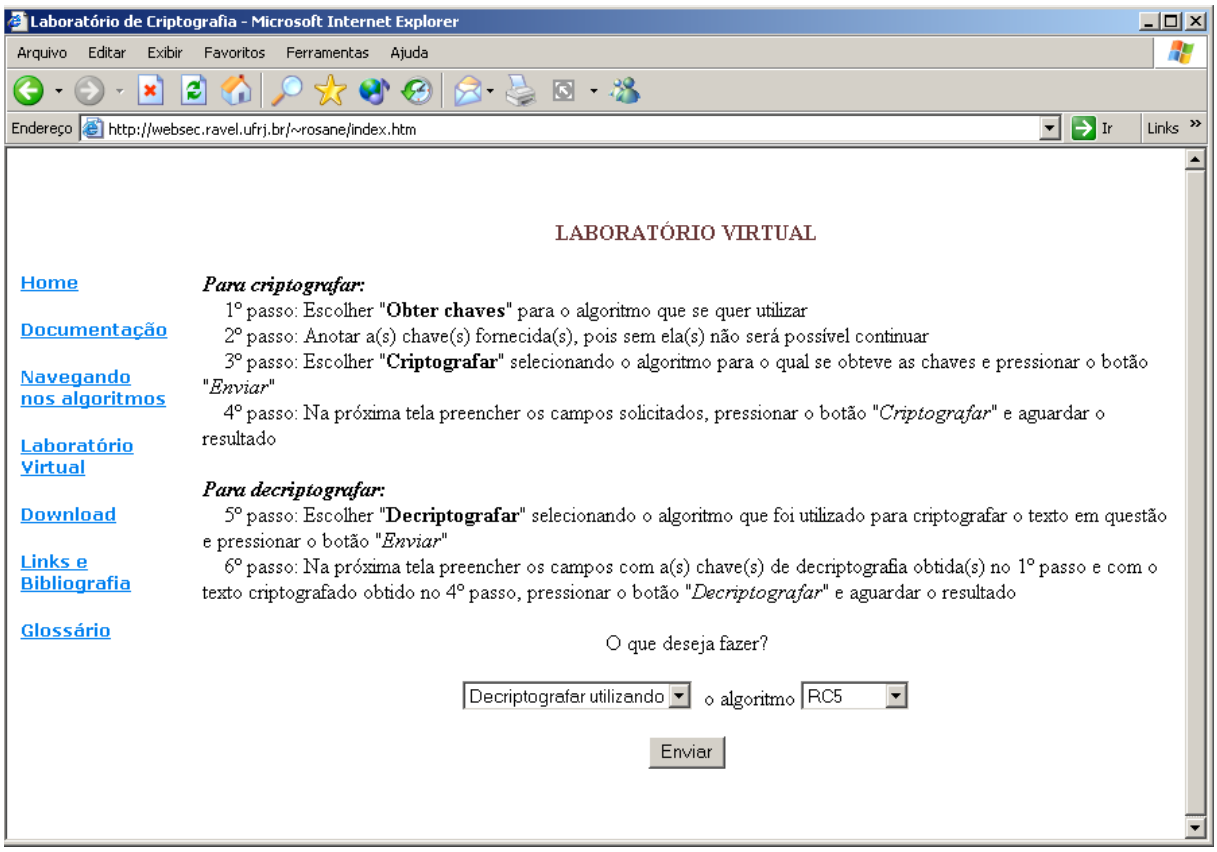


Figura 8.9- Optando pelo processo de decifragem

Semelhante ao processo de cifragem ao pressionarmos o botão Enviar uma nova tela se abrirá, como mostra a figura a seguir.

Nesta tela será possível digitar a Chave a ser utilizada para decifrar a informação (note que como RC5 é um algoritmo simétrico a mesma chave utilizada para cifrar deve ser utilizada para decifrar - é a chamada "Chave Secreta").

Também será possível digitar a informação a ser decriptografada, ou seja, o texto ininteligível obtido através do processo de cifragem.

Semelhante ao processo de cifragem ao pressionarmos o botão Enviar uma nova tela se abrirá, como mostra a figura a seguir.

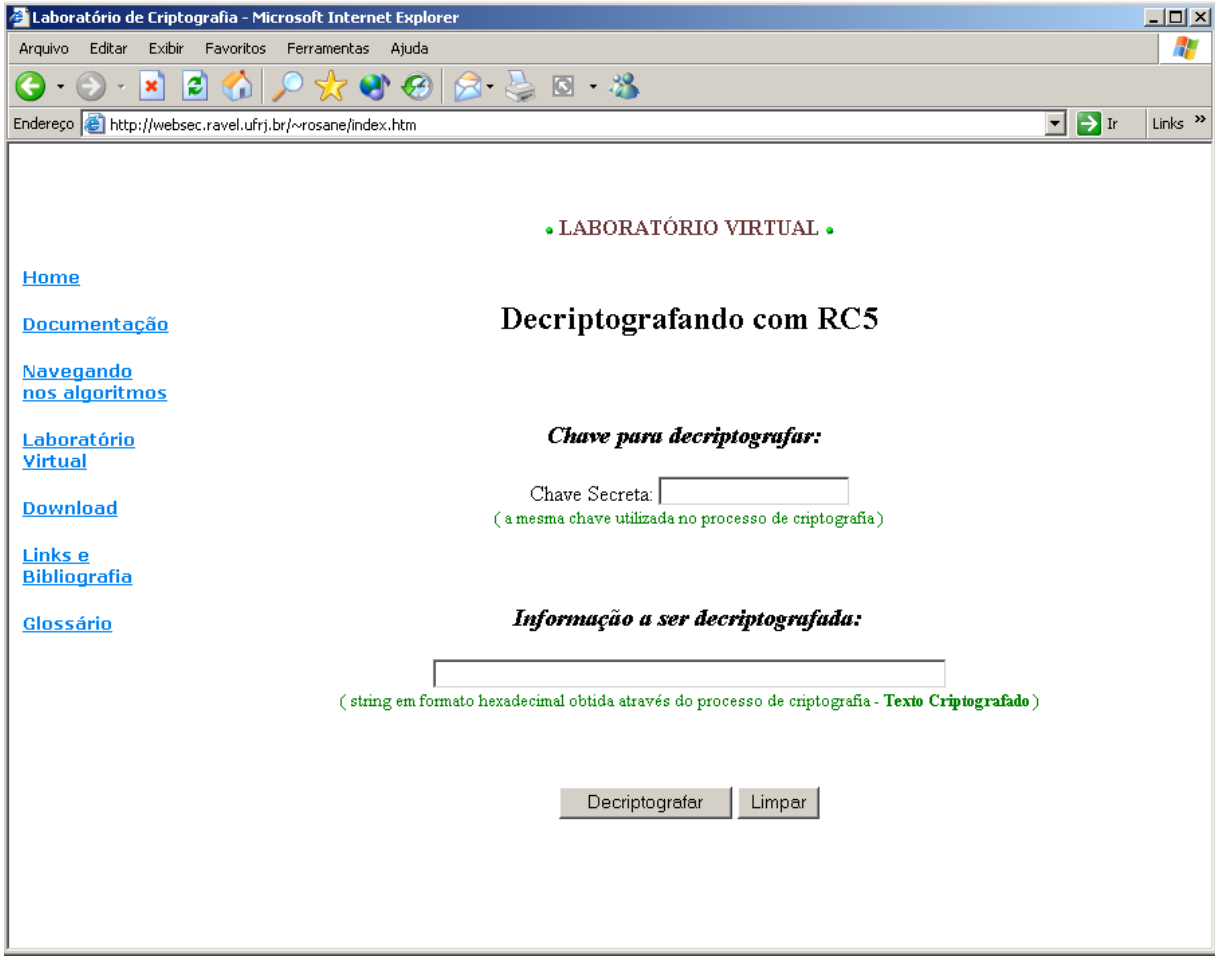


Figura 8.10- Inserindo dados para decifrar

Nesta tela será possível digitar a Chave a ser utilizada para decifrar a informação (note que como RC5 é um algoritmo simétrico a mesma chave utilizada para cifrar deve ser utilizada para decifrar - é a chamada "Chave Secreta").

Também será possível digitar a informação a ser decriptografada, ou seja, o texto ininteligível obtido através do processo de cifragem. Após o pressionamento do botão

"Decriptografar" as informações serão processadas pelo Matlab. Este processo poderá levar alguns minutos, dependendo do tamanho da informação a ser decriptografada.

Após decorrido o tempo necessário, um Pop-Up será exibido para o usuário com o resultado da decifragem.

Neste Pop-Up, além da informação decriptografada, serão exibidas também outras informações, como o texto criptografado, a chave utilizada bem como uma figura ilustrativa do processo de decifragem. Esta figura possui áreas clicáveis que revelará alguns detalhes do processo, como por exemplo os blocos em que foram divididos o texto criptografado antes de ser decriptografado, as subchaves geradas no processo, etc.

A figura a seguir mostra um Pop-Up resultante no qual foram utilizadas as seguites informações:

Chave Secreta: 2AF349CA977BE684

Informação a ser Decriptografada:

89E13D4B36B44D35AFEB3755A745260E20748C4F84D5295F

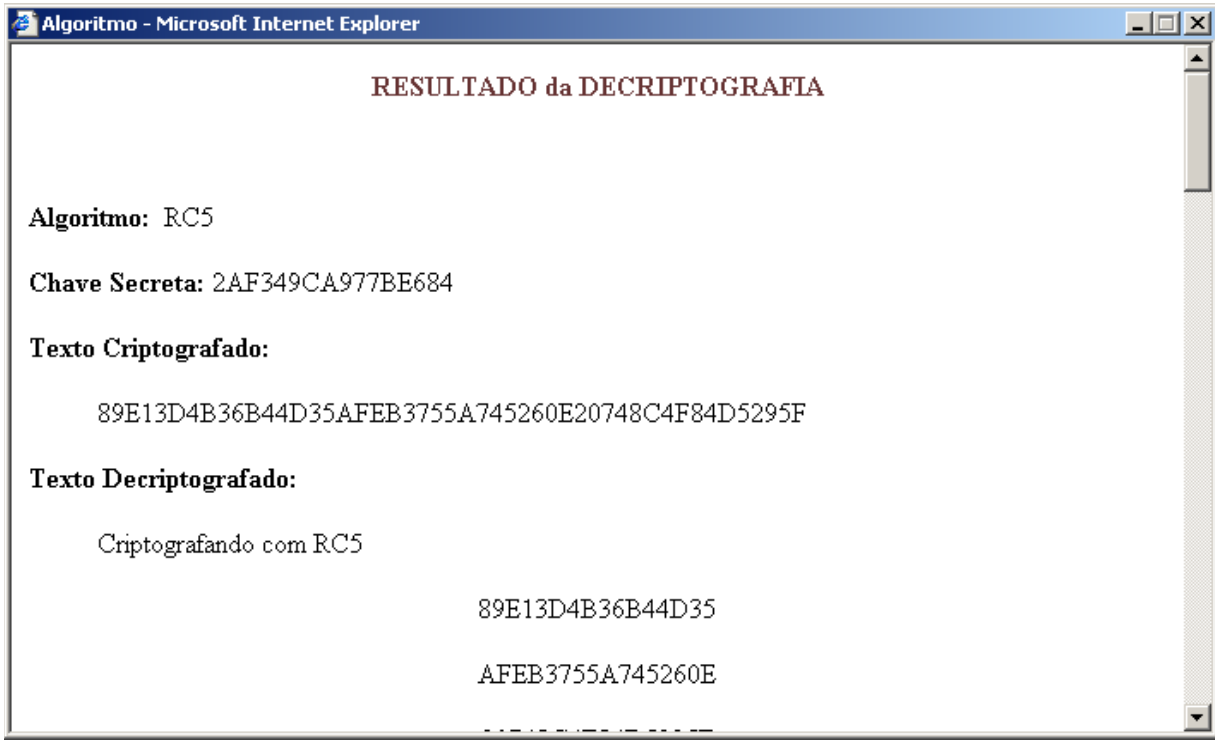


Figura 8.11- Resultado da decifragem

As demais informações do Pop-Up são apresentadas abaixo:

Segurança:

RC5 é relativamente novo, mas os laboratórios RSA já gastaram um tempo considerável analizando sua segurança. Para criptoanalizar blocos de 64 bits criptografados com 12 rounds (a forma de implementação utilizada neste laboratório) seriam necessários 253 tentativas, ou seja 9.007.199.254.740.992 tentativas!

Referência: Applied Cryptography, Second Edition: Protocols, Algorthms, and Source Code in C, by Bruce Schneier

Detalhes da cifragem:

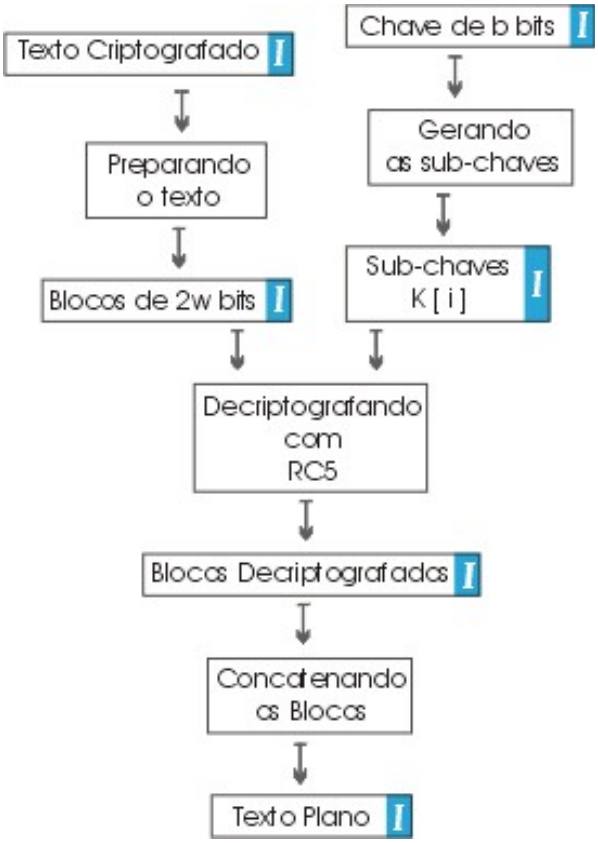


Figura 8.12- Diagrama da decifragem

Texto Criptografado:

89E13D4B36B44D35AFEB3755A745260E20748C4F84D5295F

voltar ao diagrama

Blocos de 64 bits:

43726970746F6772
6166616E646F2063
6F6D205243350000

voltar ao diagrama

Chave Secreta:

2AF349CA977BE684

voltar ao diagrama

Subchaves K:

AE949903
9CE10049
4477EFA0
FE2EC9B8
2A158741
DC242AC1
9DE1BB3F
0273146E
6CB62D2E
BAB6CC74
40F941D3
BF854DC1
72CC90E9
6E356626
EE8D2091
AC20E6FE
27FBDA00
860AF197
856767F8
0ECA05BC
1B39EE7E
2A39EBA4
E8D204B9
D0CE6CE4
2F3CFCB7
102BE160

voltar ao diagrama

Blocos Decriptografados:

43726970746F6772
6166616E646F2063
6F6D205243350000

voltar ao diagrama

Texto Decriptografado:

Criptografando com RC5

voltar ao diagrama

9 IDEA

9.1 História

IDEA nasceu na Europa como resposta à falta de segurança do DES a longo prazo. O algoritmo foi desenvolvido através de uma colaboração do Swiss Federal Institute of Technology de Zürich (Dr. X. Lai / Prof. J. Massey) e Ascom.

Em 1990 Xuejia Lai e James Massey publicam na Suiça "A Proposal for a New Block Encryption Standard" ("Uma Proposta para um Novo Padrão de Encriptação de Bloco"), o assim chamado IDEA (International Data Encryption Algorithm), para substituir o DES.

O IDEA utiliza uma chave de 128 bits e emprega operações adequadas para computadores de uso geral, tornando as implementações do software mais eficientes.

9.2 Características

•IDEA é um algoritmo simétrico (a mesma chave é utilizada para cifrar e decifrar), com chave de 128 bits (o dobro do tamanho da chave do DES).

•IDEA está disponível a todas as pessoas, totalmente especificado, não tem restrições de exportação e é de fácil compreensão.

•IDEA cifra em blocos, isto quer dizer que com um bloco de 64 bits (texto claro) e uma chave de 128 bits obtemos um novo bloco de 64 bits (criptografado).

•IDEA atua cifrando grupos binários de 64 bit. Para isto utiliza uma chave de 128 bits. Se o bloco a cifrar for menor do que 64 bits devemos completar com zeros.

9.3 Funcionamento do algoritmo

9.3.1 Obtenção das chaves

IDEA utiliza 52 sub-chaves para o processo de cifragem de cada bloco de 64 bits.

a) Para obter as chaves Ki divide-se a chave principal em 8 sub-blocos de 16 bits. Estas são as primeiras 8 sub-chaves.

b) Depois, deve-se deslocar a chave de 128 bits de 25 posições à esquerda.

c) Divide-se novamente em 8 sub-blocos e assim obtém-se as 8 sub-chaves seguintes. Este procedimento deve ser repetido até obter as 52 chaves.

9.3.2 Cifrando um bloco

O algoritmo de cifragem consta de 8 etapas nas quais utilizam-se 6 sub-chaves por etapa e mais uma transformação final que requer 4 sub-chaves, totalizando 52 sub-chaves.

a) Primeiro divide-se o texto plano em 4 sub-blocos de 16 bits cada um.

O primeiro e o quarto sub-blocos se combinam com a primeira e quarta sub-chaves utilizando-se multiplicação modulo $(2^{16})+1$ (isto quer dizer que multiplicamos 2 números de 16 bits, correspondendo 0000000000000000 a 2^{16}).

A estes resultados chamaremos r_1 e r_4. O segundo e terceiro blocos se combinam com a segunda e terceira sub-chave utilizando-se soma módulo 2^{16} que chamaremos de r_2 e r_3.

b) Em seguida realiza-se um XOR entre r_1 e r_3 obtendo-se s_1 e um XOR entre r_2 e r_4 obtendo-se s_2. Realiza-se uma multiplicação modulo $(2^{16})+1$ com a quinta chave a fim de obter-se t_1.

Em seguida é feita uma soma modulo (2^{16}) entre t_1 e s_2 obtendo-se t_2, que se multiplica modulo $(2^{16})+1$ com a sexta para se obter t_3, a qual se soma modulo (2^{16}) com t_1 para obter-se t_4.

c) Fazendo um XOR de t_3 com r_1 obtém-se o primeiro bloco para o passo seguinte. O segundo bloco é t_3 XOR r_3. O terceiro bloco é t_4 XOR r_2 e o quarto bloco é t_4 XOR r_4

d) Com estes 4 blocos e 6 chaves seguintes repetimos todo o passo anteriormente descrito.

Ao realizar os 8 passos, utilizando 48 subchaves, obteremos 4 blocos.

Por ultimo realiza-se uma transformação final na qual intervém as 4 últimas sub-chaves.

O primeiro e o quarto bloco são multipicados modulo $(2^{16})+1$ com a 49ª e 52ª chaves, e o segundo e o terceiro bloco são somados modulo 2^{16} com a 50ª e a 51ª chaves.

O boco de 64 bits resultante é texto criptografado.

9.3.3 Decifrando um bloco

O algoritmo de decifragem é exatamente o mesmo que o de cifragem, o mesmo esquema gráfico. A única diferença é nas sub-chaves a serem utilizadas, que serão as seguintes:

1ª passo s49* s50# s51# s52* s47 s48

2ª passo s43* s45# s44# s46* s41 s42

3ª passo s37* s39# s38# s40* s35 s36

4ª passo s31* s33# s32# s34* s29 s30

5ª passo s25* s27# s26# s28* s23 s24

6ª passo s19* s21# s20# s22* s17 s18

7ª passo s13* s15# s14# s16* s11 s12

8ª passo s7* s9# s8# s10* s5 s6

Transformação final: s1* s2# s3# s4*

Onde:

sXX* = inverso multiplicativo de sXX modulo $((2^{16})+1)$

sXX# = inverso aditivo de sXX modulo (2^{16})

9.3.4 Esquema gráfico de funcionamento

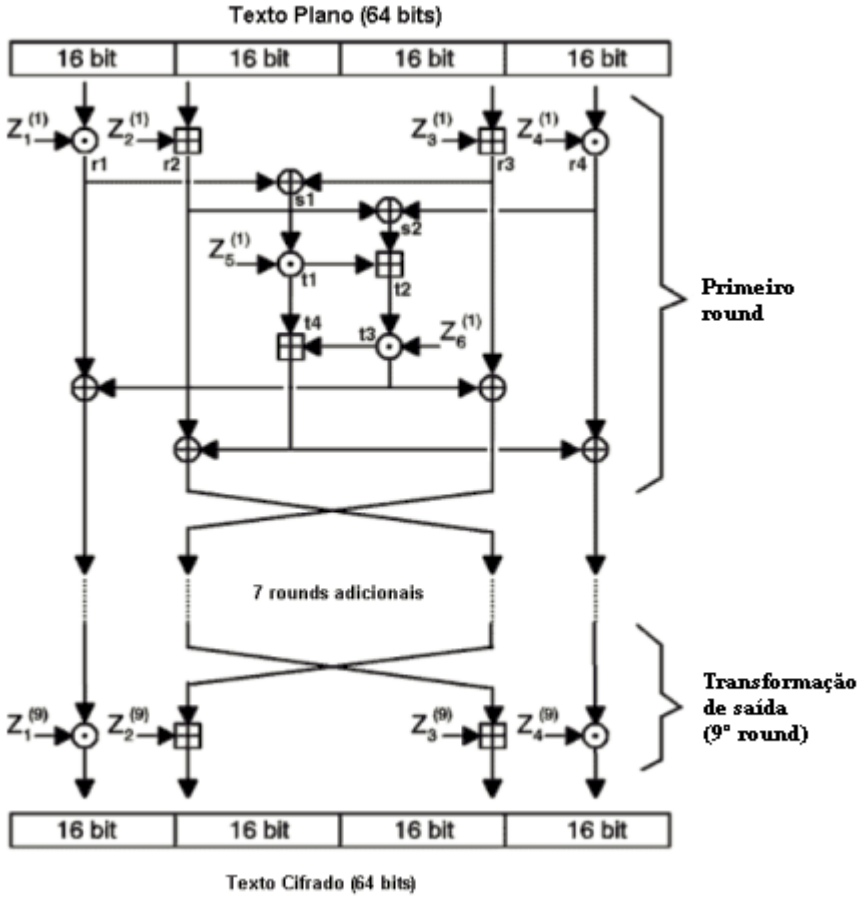
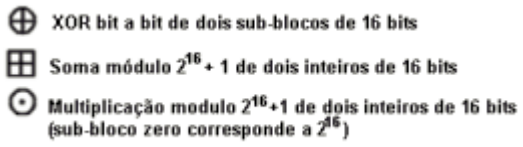


Figura 9.1-Esquema gráfico do funcionamento do algoritmo IDEA

Onde:



9.4 Implementação

9.4.1 Requisitos de Software

•Microsoft Windows 95, Windows 98 (original and Second Edition), Windows Millennium Edition, Windows NT 4.0 (com Service Pack 5 para correção Y2K ou Service Pack 6a) ou Windows 2000

•MATLAB 6.0 (R12) ou 6.1 (R12.1)

9.4.2 Requisitos de Hardware

• Pentium, Pentium Pro, Pentium II, Pentium III, Pentium IV** ou AMD Athlon

• 64 MB RAM (mínimo), 128 MB RAM (recomendadop) adaptador gráfico de 8 bits e monitor (para 256 cores simultâneas)

9.4.3 Descrição dos módulos principais

O IDEA é composto de vários módulos, os quais devem estar no diretório de trabalho do Matlab.

Os módulos principais são: GeraChavesIDEA, CifraBlocoIDEA e DecifraBlocoIDEA

GeraChavesIDEA

Este é o responsável pela geração das 52 sub-chaves necessárias ao algoritmo.

Sintaxe:

```
>>[K, Resultado, Kinv] = GeraChavesIdea(Key)
```

Onde:

Key deve ser um numero binario de 128 bits, na forma:

```
>>Key = '11010...011';
```

K é uma matriz que armazena todas as 52 sub-chaves, onde a primeira chave é representada por K(1,:), a segunda por K(2,:), e assim sucessivamente.

Kinv é uma matriz que armazena o inverso (multiplicativo ou aditivo - conforme o caso) de todas as 52 sub-chaves, onde a primeira chave é representada por Kinv(1,:), a segunda por Kinv(2,:), e assim sucessivamente

Resultado - retorna 0 se todas as sub-chaves K possuírem inverso, e retorna diferente de 0 se pelo meos uma sub-chave não possuir inverso

CifraBlocoIDEA

Este módulo é o responsável pelo processo de cifragem propriamente dito. Ele trabalha cifrando um bloco de 64 bits. Ele realiza as 8 etapas do processo de cifragem, cada etapa utiliza 6 chaves.

Sintaxe:

>> [BlocoCifrado] = CifraBlocoIdea(Chaves,TextoPlano_64bits)

Onde:

Chaves está na forma Chaves(1,:), Chaves(2,:)... e o texto plano um array linha em binario, TextoPlano_64bits = '111010...110'

BlocoCifrado é o resultado da cifragem do Bloco de 64bits através do algoritmo IDEA.

DecifraBlocoIDEA

Este é o módulo responsável pelo processo inverso ao de cifragem.

Sintaxe:

>> [BlocoDecifrado] = DecifraBlocoIdea(Kinv,BlocoCifrado)

Onde:

Kinv é o resultado do módulo GeraChavesIDEA, conforme explicado anteriormente.

BlocoCifrado é o bloco de 64 bits em formato hexadecimal a ser decriptografado.

BlocoDecifrado é o resultado da decifragem do Bloco de 64bits através do algoritmo IDEA.

Outros Módulos

Os módulos acima (GeraChavesIDEA, CifraBlocoIDEA e DecifraBlocoIDEA) são formados de diversos módulos necessários a sua implementação, na próxima seção veremos em detalhes sua composição.

9.4.4 Descrição dos módulos auxiliares

GeraChavesIDEA

o ChecaInverso - Testa se a chave gera sub-chaves que possuem o inverso multiplicativo, ou seja, garante que o texto cifrado poderá ser decifrado.
Sintaxe: [Resultado,Kinv] = ChecaInverso(K)

CifraBlocoIDEA

o DecXor - Realiza a operaçao xor entre dois numeros decimais
Sintaxe: DecXor(Num1, Num2)
Exemplo: DecXor(8, 4) = 8 xor 4 = 1000 xor 0100 = 1100 = 12

DecifraBlocoIDEA

Este módulo é idêntico ao CifraBlocoIDEA, sendo que antes de iniciar a cifragem (decifragem) ele gera o inverso multiplicativo e aditivo das chaves, necessárias para decifrar o texto (Ver Passo 3)

9.5 Laboratório Virtual

9.5.1 Criptografando com IDEA no Laboratório Virtual

Primeiramente geramos uma chave válida para o algoritmo, para tanto devemos selecionar "Obter chaves para", escolher o algoritmo "IDEA" e pressionar o botão "Enviar", como ilustrado na figura abaixo:

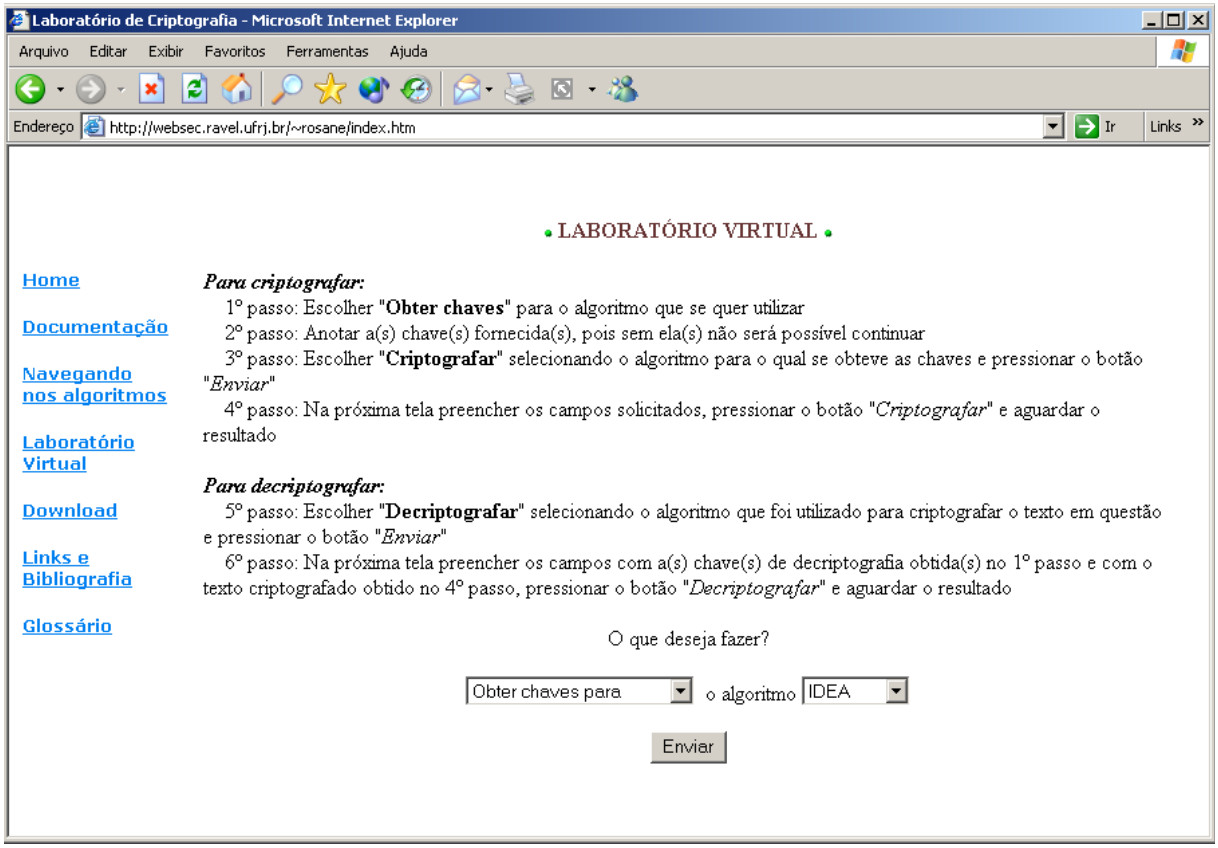


Figura 9.2- Obtendo uma chave válida

Após alguns segundo um Pop-Up como o mostrado na figura abaixo sugirá. Ele contém a chave que poderá ser utilizada no processo de cifragem.

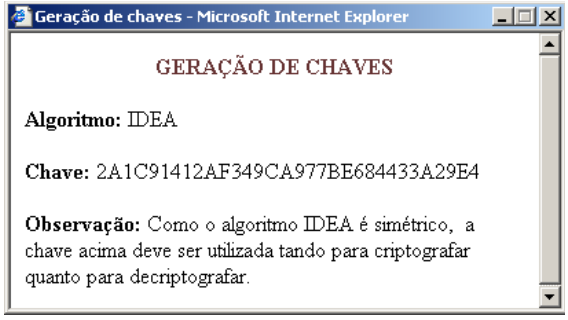


Figura 9.3- Resultado da geração da chave

Após obtida a chave, deve-se escolher "Criptografar utilizando" e selecionar o algoritmo "IDEA" como mostra a figura abaixo:

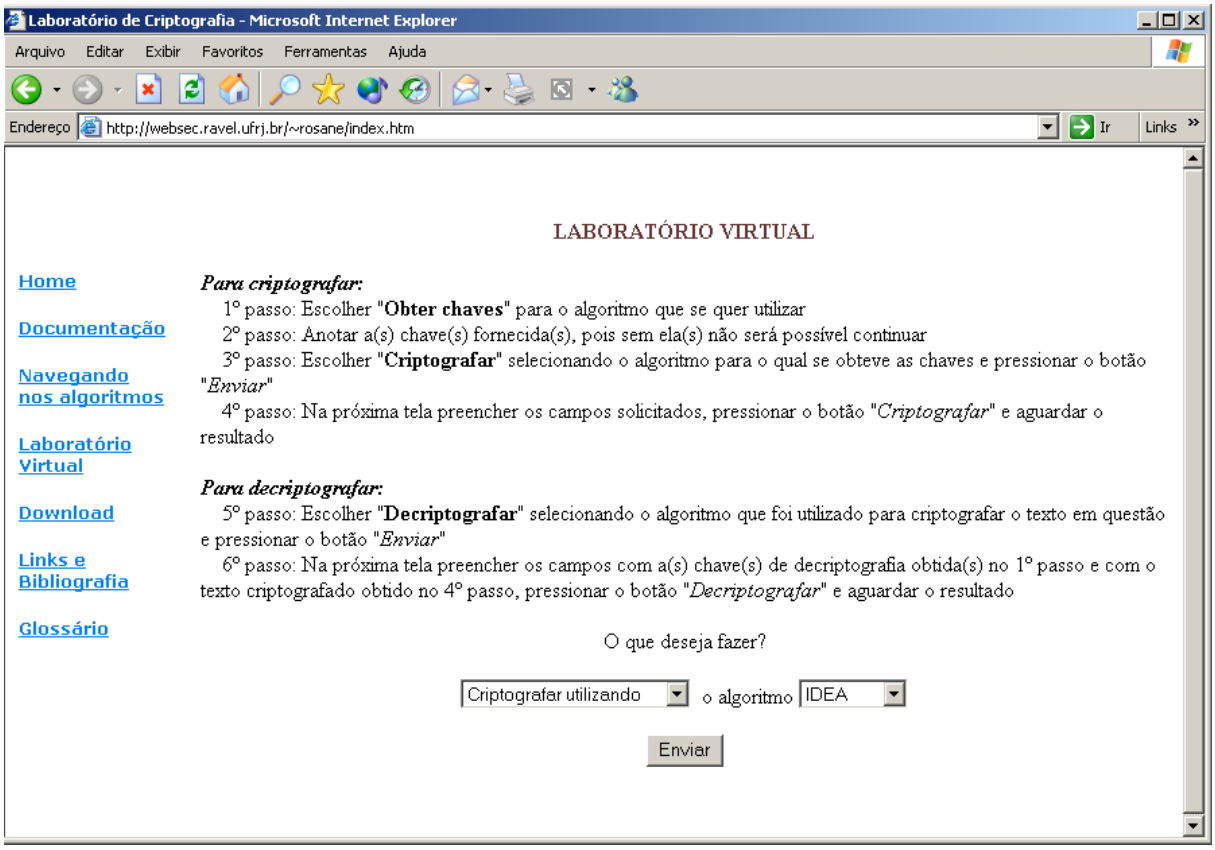


Figura 9.4- Optando pelo processo de cifragem

Após pressionar o botão "Enviar" uma nova tela será apresentada, nela será possível digitar a Chave Secreta (obtida no passo 1) e a informação a ser criptografada. Como pode ser visto na próxima figura.

A informação a ser criptografada pode conter quaisquer caracteres alfanuméricos inclusive caracteres de pontuação.

Após preencher estes campos deve-se pressionar o botão "Criptografar".

Caso algum erro seja cometido, pode-se utilizar o botão Limpar para remover todas as informações digitadas.

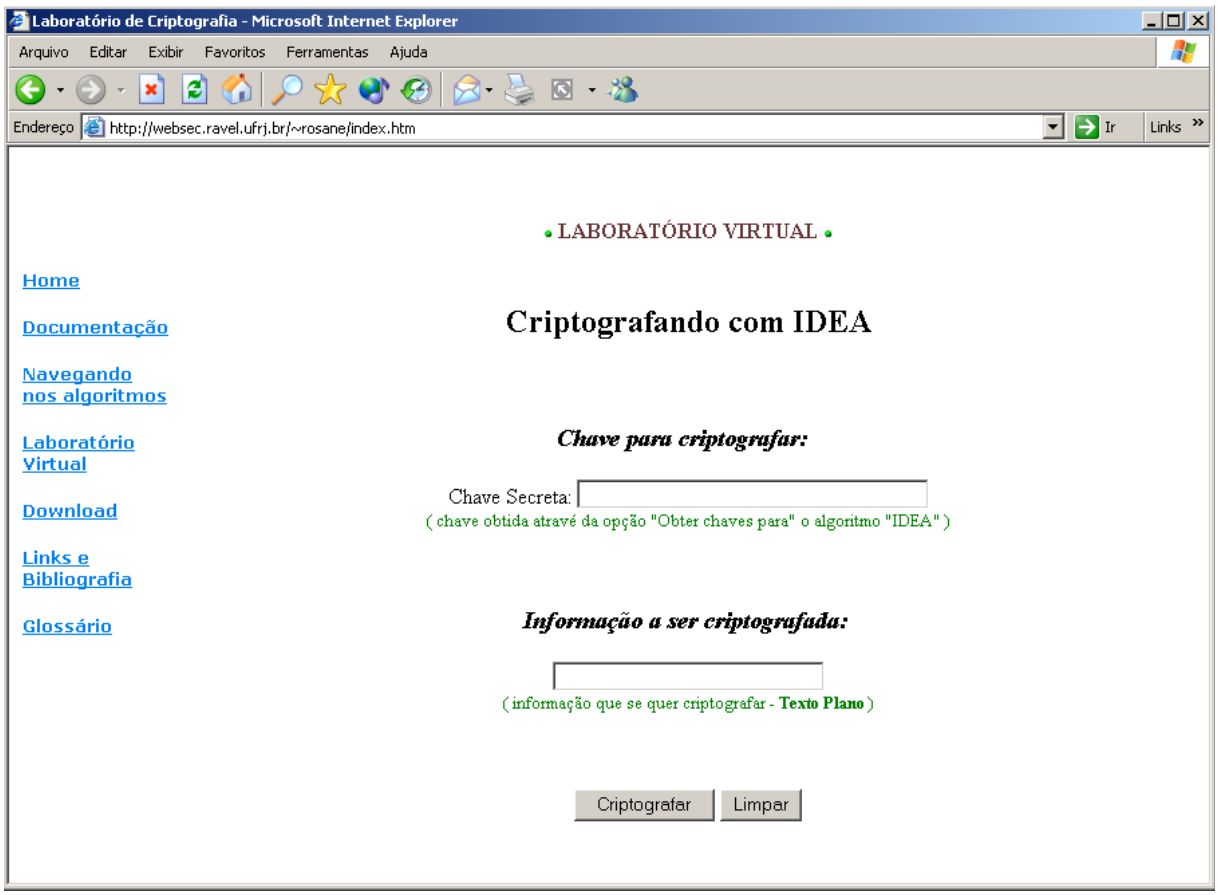


Figura 9.5- Inserindo os dados para cifrar

Após o pressionamento do botão "Criptografar" as informações serão processadas pelo Matlab. Este processo poderá levar alguns minutos, dependendo do tamanho da informação a ser criptografada. Após decorrido o tempo necessário, um Pop-Up será exibido para o usuário com o resultado da cifragem.

Neste Pop-Up, além da informação criptografada, serão exibidas também outras informações, como o texto plano, a chave utilizada bem como uma figura ilustrativa do processo de cifragem. Esta figura possui áreas clicáveis que revelará alguns detalhes do processo, como por exemplo os blocos em que foram divididos o texto plano antes de ser criptografado, as subchaves geradas no processo, etc.

A figura a seguir mostra um Pop-Up resultante no qual foram utilizadas as seguites informações:

Após o pressionamento do botão "Criptografar" as informações serão processadas pelo Matlab. Este processo poderá levar alguns minutos, dependendo do tamanho da informação a ser criptografada.

Após decorrido o tempo necessário, um Pop-Up será exibido para o usuário com o resultado da cifragem.

Neste Pop-Up, além da informação criptografada, serão exibidas também outras informações, como o texto plano, a chave utilizada bem como uma figura ilustrativa do processo de cifragem. Esta figura possui áreas clicáveis que revelará alguns detalhes do processo, como por exemplo os blocos em que foram divididos o texto plano antes de ser criptografado, as subchaves geradas no processo, etc.

A figura a seguir mostra um Pop-Up resultante no qual foram utilizadas as seguites informações:

Chave Secreta: 2A1C91412AF349CA977BE684433A29E4

Informação a ser Criptografada: Criptografando com IDEA.

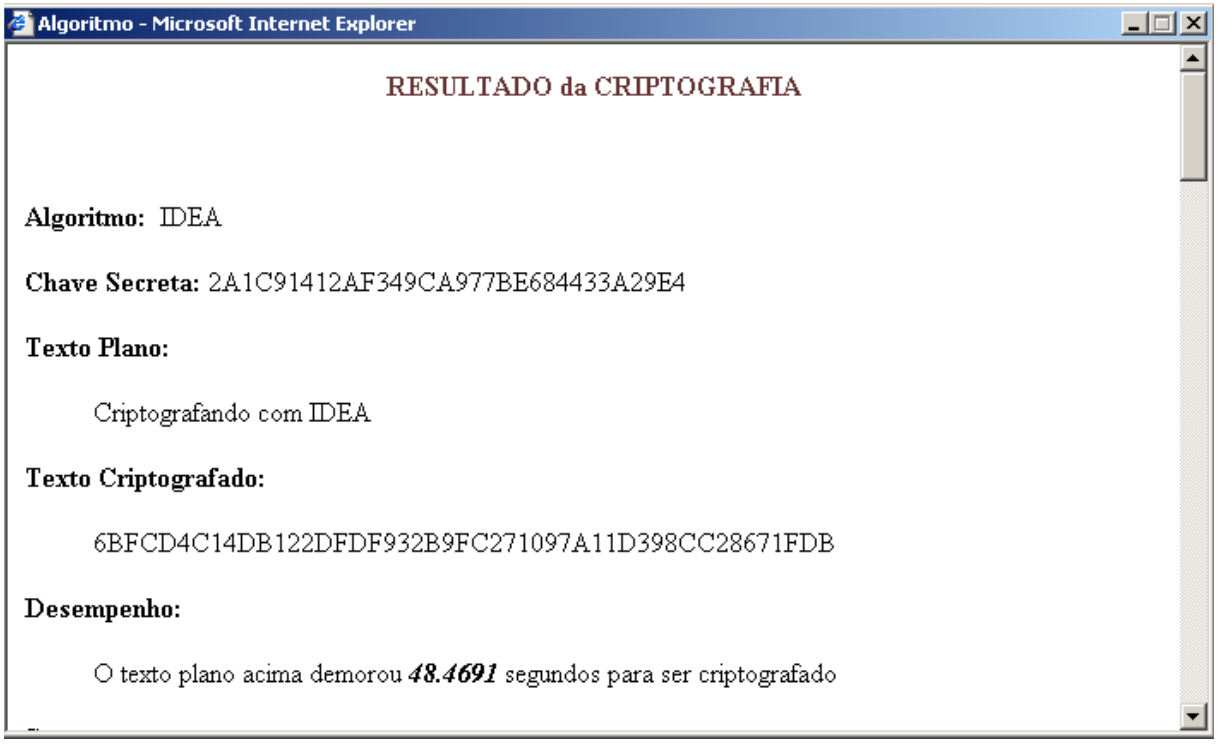


Figura 9.6- Resultado da cifragem

As demais informações do Pop-Up são apresentadas abaixo:

Segurança:

O comprimento da chave do IDEA é de 128 bits - duas vezes mais longa que a do DES. Assumindo que um ataque de força bruta muito eficiente seja realizado, ele necessitaria de 2128(1038) tentativas para recuperar a chave. Com um chip que testa um bilhão de chaves por segundo ainda levaria 1013 anos. Uma combinação de 1024 destes chips poderiam encontrar a chave em 1 dia, mas não haveria silício suficiente para construir esta máquina.

Referência: Applied Cryptography, Second Edition: Protocols, Algorthms, and Source Code in C, by Bruce Schneier

Detalhes da cifragem:

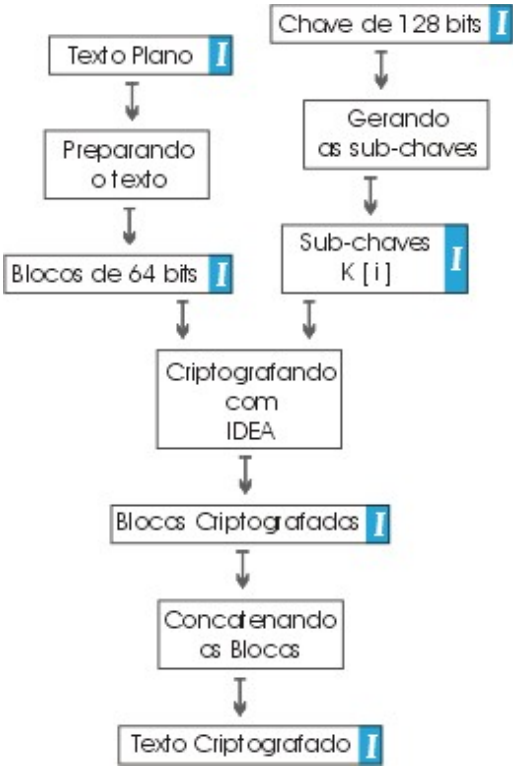


Figura 9.7- Diagrama da cifragem

Texto Plano:

Criptografando com IDEA

voltar ao diagrama

Blocos de 64 bits:

43726970746F6772
6166616E646F2063
6F6D204445530000

voltar ao diagrama

Chave Secreta:

2A1C91412AF349CA977BE684433A29E4

voltar ao diagrama

Blocos Criptografados:

6BFCD4C14DB122DF
DF932B9FC271097A
11D398CC28671FDB

voltar ao diagrama

Texto Criptografado:

6BFCD4C14DB122DFDF932B9FC271097A11D398CC28671FDB

voltar ao diagrama

9.5.2 Decriptografando com IDEA no Laboratório Virtual

Como já possuímos a Chave Secreta devemos passar direto à decifragem, para tanto devemos escolher "Decriptografar utilizando" o algoritmo "IDEA", como mostra a figura abaixo:

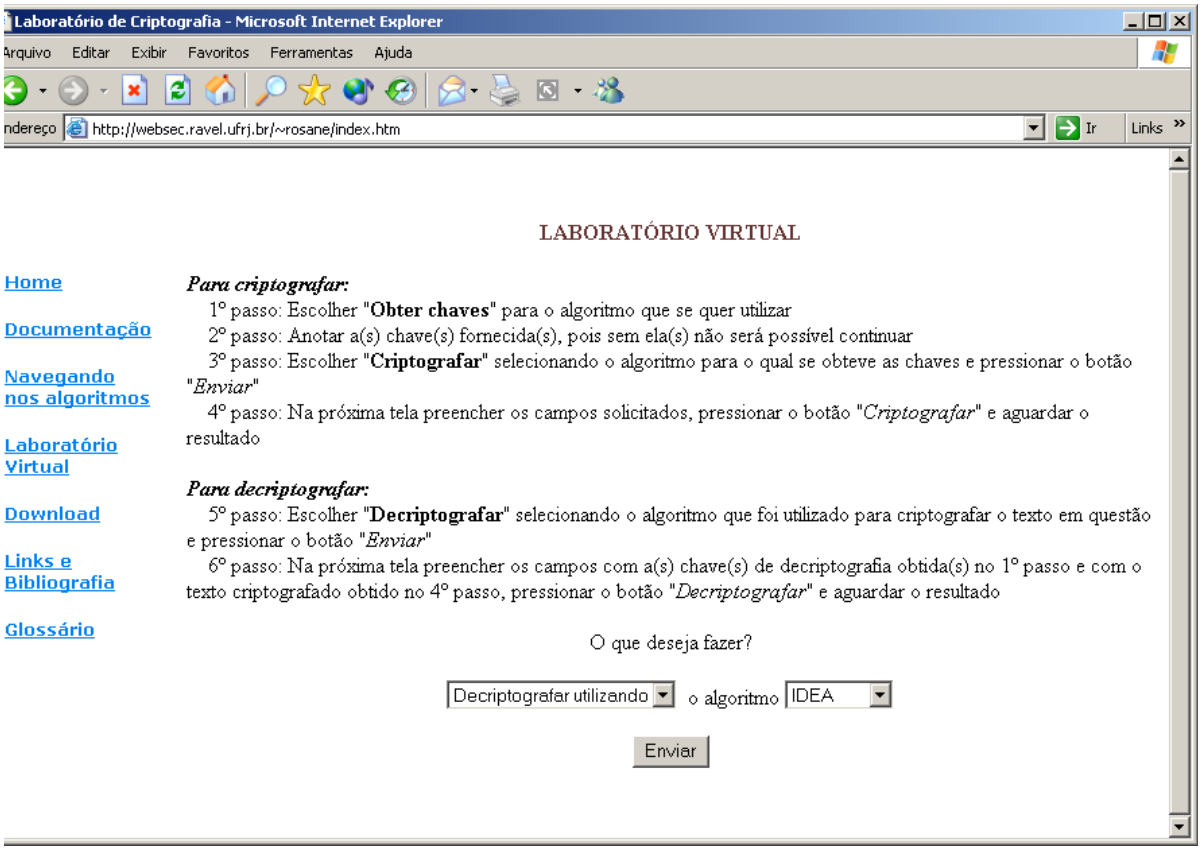


Figura 9.8- Optando pelo processo de decifragem

Semelhante ao processo de cifragem ao pressionarmos o botão Enviar uma nova tela se abrirá, como mostra a figura a seguir.

Nesta tela será possível digitar a Chave a ser utilizada para decifrar a informação (note que como IDEA é um algoritmo simétrico a mesma chave utilizada para cifrar deve ser utilizada para decifrar - é a chamada "Chave Secreta").

Também será possível digitar a informação a ser decriptografada, ou seja, o texto ininteligível obtido através do processo de cifragem.

Semelhante ao processo de cifragem ao pressionarmos o botão Enviar uma nova tela se abrirá, como mostra a figura a seguir.

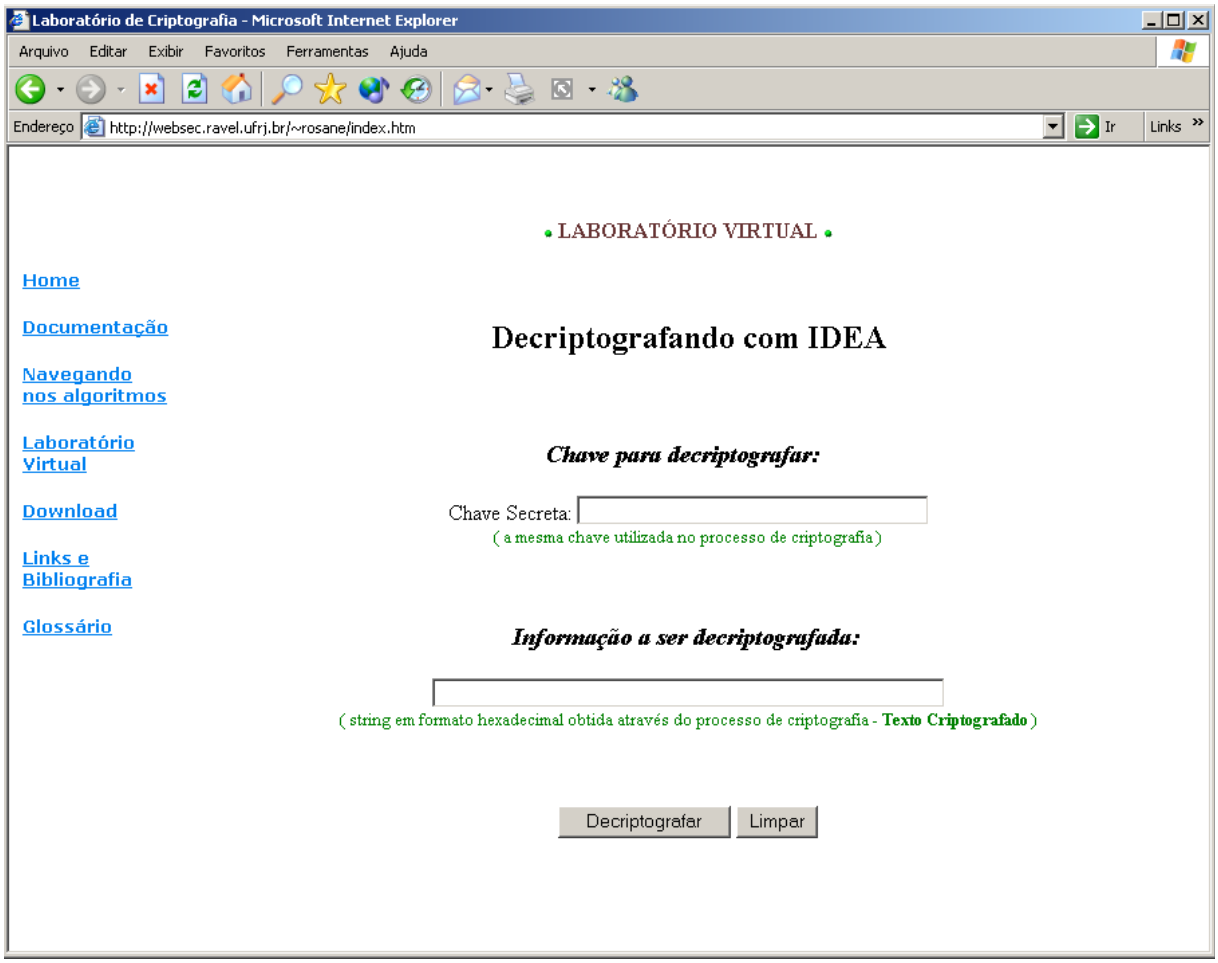


Figura 9.9- Inserindo dados para decifrar

Nesta tela será possível digitar a Chave a ser utilizada para decifrar a informação (note que como IDEA é um algoritmo simétrico a mesma chave utilizada para cifrar deve ser utilizada para decifrar - é a chamada "Chave Secreta").

Também será possível digitar a informação a ser decriptografada, ou seja, o texto ininteligível obtido através do processo de cifragem. Após o pressionamento do botão "Decriptografar" as informações serão processadas pelo Matlab. Este processo poderá levar alguns minutos, dependendo do tamanho da informação a ser decriptografada.

Após decorrido o tempo necessário, um Pop-Up será exibido para o usuário com o resultado da decifragem.

Neste Pop-Up, além da informação decriptografada, serão exibidas também outras informações, como o texto criptografado, a chave utilizada bem como uma figura ilustrativa do processo de decifragem. Esta figura possui áreas clicáveis que revelará alguns detalhes do processo, como por exemplo os blocos em que foram divididos o texto criptografado antes de ser decriptografado, as subchaves geradas no processo, etc.

A figura a seguir mostra um Pop-Up resultante no qual foram utilizadas as seguites informações:

Chave Secreta: 2A1C91412AF349CA977BE684433A29E4

Informação a ser Decriptografada:

6BFCD4C14DB122DFDF932B9FC271097A11D398CC28671FDB

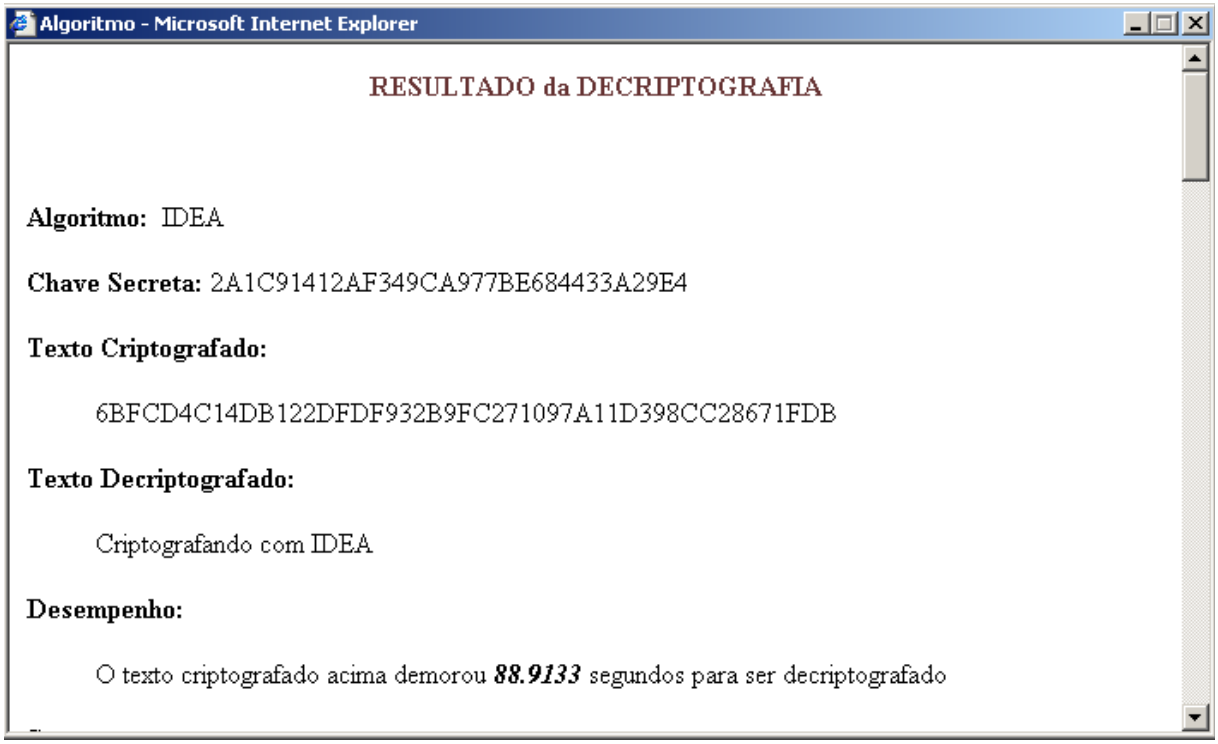


Figura 9.10- Resultado da decifragem

As demais informações do Pop-Up são apresentadas a seguir:

Segurança:

O comprimento da chave do IDEA é de 128 bits - duas vezes mais longa que a do DES. Assumindo que um ataque de força bruta muito eficiente seja realizado, ele necessitaria de 2128(1038) tentativas para recuperar a chave. Com um chip que testa um bilhão de chaves por segundo ainda levaria 1013 anos. Uma combinação de 1024 destes chips poderiam encontrar a chave em 1 dia, mas não haveria silício suficiente para construir esta máquina.

Referência: Applied Cryptography, Second Edition: Protocols, Algorthms, and Source Code in C, by Bruce Schneier

Detalhes da cifragem:

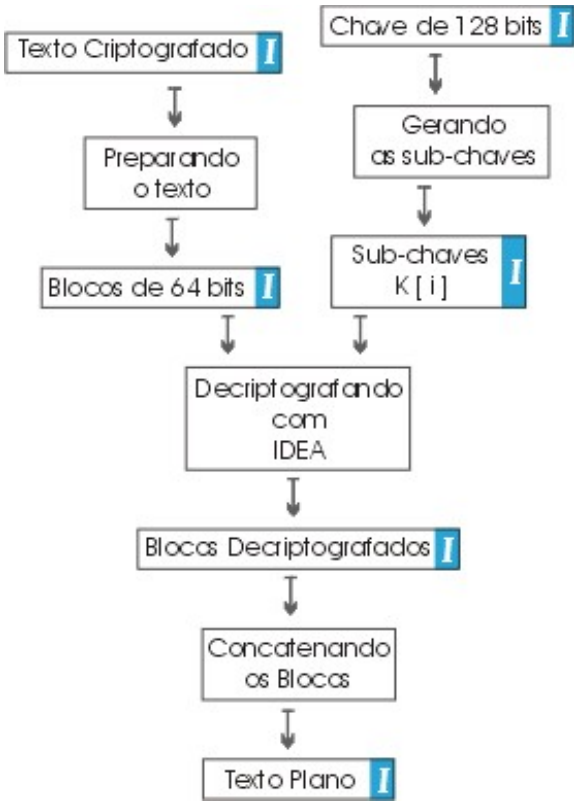


Figura 9.11- Diagrama da decifragem

Texto Criptografado:

6BFCD4C14DB122DFDF932B9FC271097A11D398CC28671FDB

voltar ao diagrama

Blocos de 64 bits:

6BFCD4C14DB122DF
DF932B9FC271097A
11D398CC28671FDB

voltar ao diagrama

Chave Secreta (Chave de 64 bits):

2A1C91412AF349CA977BE684433A29E4

voltar ao diagrama

Blocos Decriptografados:

3433373236393730
3631363636313645
3646364432303439

voltar ao diagrama

Texto Decriptografado:

Criptografando com IDEA

voltar ao diagrama

10 Uma breve introdução sobre algoritmos assimétricos

10.1 Introdução

Para fins de transações comerciais virtuais os algoritmos simétricos (esta propriedade) se torna pouco prática e insegura, porque a própria chave deve ser transmitida por meios eletrônicos.

Para resolver este problema se criou a criptografia de chave pública.

Em 1976 Whitfield Diffie e Martin Hellman [RSA00] apresentaram o conceito de Criptografia por Chave Pública. Este sistema possui duas aplicações principais: Encriptação e Assinaturas Digitais.

Neste sistema cada pessoa possui um par de chaves, uma denominada Chave Pública e outra denominada Chave Privada. Enquanto a chave pública tem seu conhecimento difundido, a chave privada deve ser mantida em segredo.

Desta forma a necessidade das partes comunicantes de trocar informações sigilosas é eliminada sendo que todas as comunicações irão envolver somente a chave pública não sendo necessária a troca de chaves secretas por nenhuma das partes. Ao mesmo tempo este sistema não exige credibilidade dos meios de transmissão envolvidos. O único requisito deste sistema é que a chave pública esteja associada aos seus usuários de uma forma autenticável.

Qualquer um dos possuidores da chave publica pode usa-la para enviar uma mensagem. Porém a mesma mensagem só pode ser lida mediante o uso da chave privada a qual é de uso restrito de seu proprietário.

Neste sistema criptográfico a chave privada é matematicamente derivada da chave pública. Se, em tese, é probalisticamente impossível a um atacante derivar a chave privada da chave pública, esta propriedade ainda não pode ser matematicamente comprovada.

As implementações mais conhecidas da criptografia de chave pública é o RSA e o PGP. Em 1977, Rivest, Shamir e Adelman desenvolveram o RSA e publicaram o algoritmo de encriptação apesar da oposição do governo norte americano, que considera a criptografia um assunto de estado.

Mais tarde a patente do RSA e dada ao Instituto Tecnológico de Massachusetts (MIT) que logo a cede a um grupo denominado PKP (Public Key Partners). Em 1991, o programador Phil Zimmermann autoriza a publicação em boletins eletrônicos e grupos de noticias de um programa por ele desenvolvido e batizado como Pretty Good Privacy ou PGP. O PGP tem como base os algoritmos do RSA publicados em 1978.

Quando Zimmermann publicou o PGP se viu em problemas com o Departamento de Estado Norte Americano que abriu uma investigação para determinar se ele havia violado as restrições de exportação de criptografia ao autorizar a divulgação do código fonte do PGP na Internet. Apesar do mesmo ter se comprometido a deter seu desenvolvimento, diversos programadores em várias partes do mundo continuaram adiante, portando-o para distintas plataformas e assegurando sua expansão. Stale Schumacher, um programador norueguês, tem se encarregado das versões internacionais do PGP, que são totalmente compatíveis com sua contraparte norte americana.

O método de encriptaçao por chave publica resolve o problema de transmitir uma mensagem totalmente segura através de um canal inseguro (sujeito a observaçao, "grampo" etc.), como criar assinaturas digitais, correio eletrônico, verificar a origem dos dados e integridade. O receptor da mensagem cria duas chaves que sao relacionadas entre si, uma publica e uma privada. A chave publica pode e deve ser distribuída livremente. Quem envia a mensagem tem que utilizar a chave publica do receptor para encripta-la. Uma vez encriptada, esta mensagem só pode ser desencriptada pela chave do receptor.

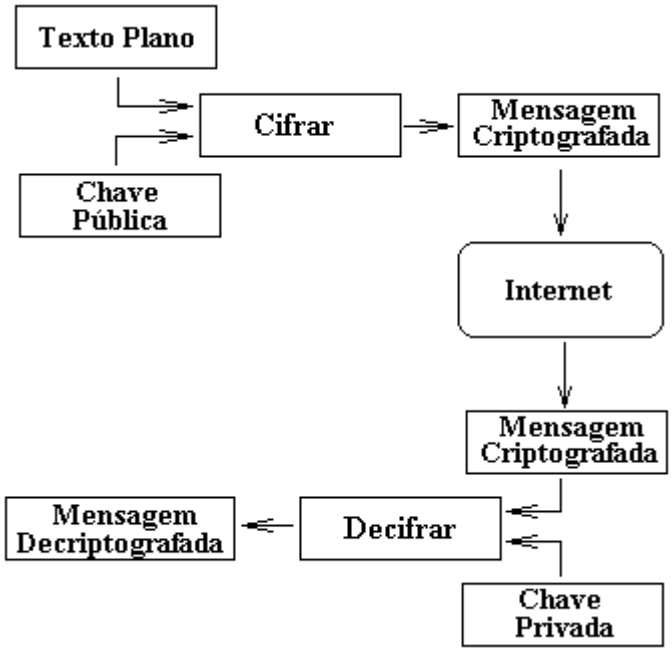


Figura 10.1-Encriptação por chave pública

O conjunto de chaves públicas, uma para cada usuário, pode ser colocado numa lista acessível a todos os membros da rede.

10.2 Funcionamento da Criptografia Assimétrica

Em um sistema de chave pública, cada pessoa tem duas chaves: uma pública que é usada por todos que queiram enviar a mensagem ao usuário e uma chave privada que o usuário utiliza para descifrar as mensagens recebidas. As mensagens criptografadas com uma chave só podem ser decifradas com a outra correspondente. Portanto, qualquer mensagem cifrada com a chave privada somente pode ser decifrada com a chave pública e vice-versa. Nesse caso, não existe uma questão de segurança, mas de identificação, certificando a origem do dado.

O funcionamento da criptografia de chave pública é o seguinte, se três pessoas querem se comunicar usando criptografia assimétrica, eles terão de gerar inicialmente os seus respectivos pares de chaves.

Depois de gerado o par de chaves (privada e pública), a pessoa (A) torna disponível, de alguma maneira, sua chave pública para as pessoas (B) e (C), que por sua vez também fará o mesmo com a própria chave pública.

As pessoas (A) e (C), escrevem mensagens, utilizando a chave pública da pessoa (B), note que, a partir desse momento somente ela, poderá ler as mensagens.

As mensagens são enviadas a pessoa (B) através da Internet.

A pessoa (B), recebe as mensagens de (A) e (C), na qual ela usa a chave privada para decifrar.

A pessoa (B), lê as mensagens, e se, tiver que responde-las, deverá usar as chaves públicas de criptografia de (A) e ou (C).

Nesse momento, é importante enfatizar que o sigilo da chave privada é muito importante, pois, a criptografia assimétrica, se baseia no fato de que a chave privada, é realmente privada, por isso, somente seu detentor deve ter acesso.

Em alguns programas, como o PGP (usa RSA), que usam criptografia assimétrica, essa chave fica armazenada no computador, protegida por uma senha conhecida somente pelo usuário.

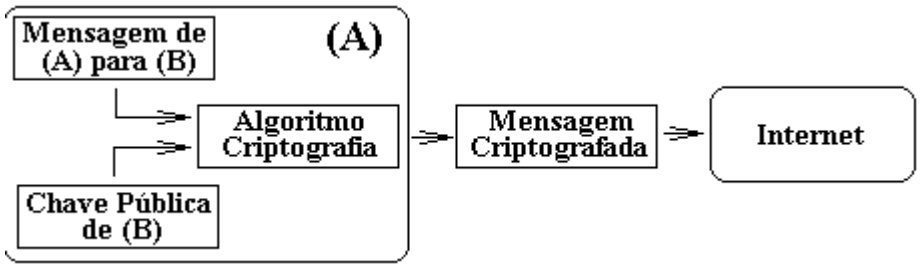


Figura 10.2- Pessoa (A) criptografando uma mensagem para pessoa (B)

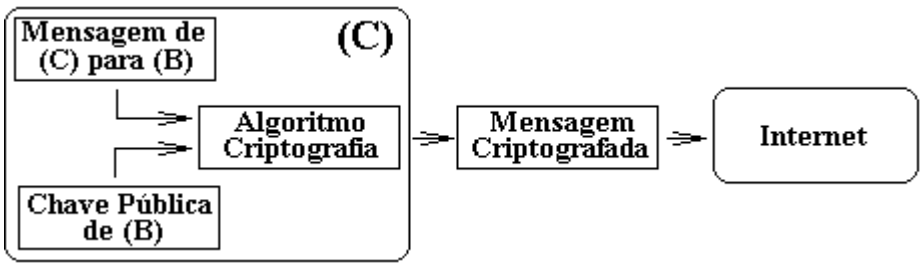


Figura 10.3- Pessoa (C) criptografando uma mensagem para pessoa (B)

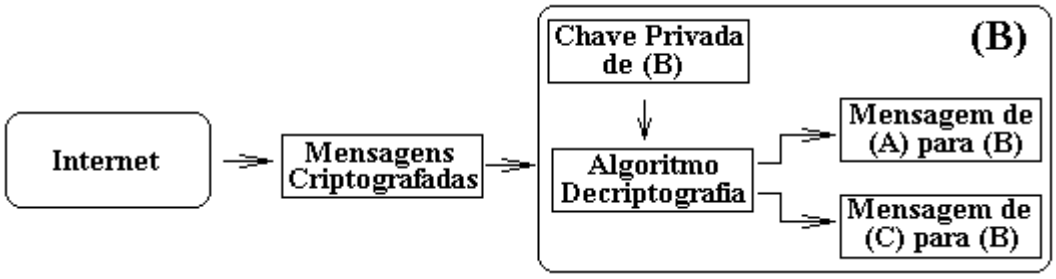


Figura 10.4- Pessoa (B) decriptografando as mensagens recebidas de (A) e (C)

10.3 Algumas Observações

Esse método por mais que pareça seguro possui desvantagens, ele é bastante lento, volumoso, as chaves não podem ser facilmente divididas e a sua capacidade de canal limitada, ou seja, o número de bits de mensagem que ele pode transmitir por segundo. Enquanto um chip que implementa o algoritmo de uma chave DES pode processar informação em alguns milhões de bits por segundo, um chip RSA consegue apenas na ordem de mil bits por segundo. É mais freqüentemente usado para certificar a origem do dado e integridade.

Pode-se concluir que sistemas de uma chave são bem mais rápidos, e sistemas de duas chaves são bem mais seguros. Uma possível solução é combinar as duas, fornecendo assim um misto de velocidade e segurança. Simplesmente usa-se a encriptação de uma chave para encriptar a mensagem, e chave secreta é transmitida usando a chave pública do destinatário, não pode-se confundir chave privada com chave secreta sendo que chave privada é mantida em segredo e a chave secreta é enviada para as pessoas que efetivarão a comunicação.

10.4 Sumário de sistemas de Chaves Públicas

Abaixo temos uma lista com alguns dos sistemas de chaves publicas:

• Diffie-Hellman – um sistema para trocas de chaves criptográficas entre partes ativas. Diffie-Hellman nao é atualmente um método de encriptaçao e desencriptaçao, mas um método de desenvolvimento e trocas um compartilhador de chaves privadas em cima de um canal de comunicaçao pública. No efetuar, as duas partes concordam em alguns valores numéricos, e a cada parte cria-se uma chave. Transformaçoes matemáticas das chaves sao

trocadas. Cada parte pode entao calcular uma terceira sessao da chave que nao pode facilmente ser derivada por um ataque sem conhecer ambas as trocas de valores. Existem várias versoes deste protocolo, envolvendo um diferente numero de partes e diferentes transformaçoes. As chaves podem Ter vários tamanhos dependendo do tipo de implementaçao. Chaves maiores geralmente sao mais seguras.

• RSA – Nome de um procedimento criptográfico desenvolvido por estas três pessoas. (Ronald Rivest, Shamir e Leonard Adleman do MIT). RSA pode se usado para encriptaçao de informaçoes e assinaturas digitais. As chaves podem Ter vários tamanhos dependendo do tipo de implementaçao. Chaves maiores geralmente sao mais seguras.

• ElGamal – Outro algoritmo baseado na exponenciaçao e aritmética modular. ElGamal é usado principalmente para encriptaçao e assinaturas digitais de uma maneira similar ao algoritmo RSA. Chaves maiores geralmente sao mais seguras.

• DSA – desenvolvido por NSA e adotado como um Federal Information Processing Standard (FIPS) pelo NIST. Neste algoritmo somente chaves entre 512 e 1024 bits sao permitidas sob o FIPS. DSA pode ser somente usado para assinaturas digitais.

11 RSA

11.1 História

A encriptação por chaves públicas foi desenvolvido por Diffie and Hellman.

Diffie-Hellman criou uma forma aceitável de algoritmos para dois sistemas trocarem chaves criptográficas de forma segura (DES ou 3DES).

A primeira reunião para discutir o algorismo e suas características foi feita em 1977 por Rivest, Shamir e Adelman. A patente deste algorismo já expirou.

O RSA é um dos algoritmos assimétricos mais utilizados.

A sua segurança está baseada na dificuldade de se determinar fatores primos de um número inteiro muito grande, que são os que gerarão as chaves.

11.2 Características

• Rivest-Shamir-Adleman

• Baseado na dificuldade de fatorar números primos grandes (Teoria dos Números)

• Até pouco tempo era patenteado, requeria licença (Licença expirou há poucos meses)

• Maduro e bastante estudado

• Segurança estrutural é uma conjectura que nunca foi provada matematicamente, mas que resiste até hoje, especialistas da área duvidam que seja contrariada

• Tamanho típico da chave de 512 a 4096 bits

• Permite sigilo e autenticação

• Possui 2 chaves públicas e 1 chave privada

• Provê Confidencialidade, Autenticação, Não-repúdio e Integridade

11.3 Funcionamento do algoritmo

11.3.1 Obtenção das chaves:

A fim de obter as duas chaves, uma para cifrar e outra para decifrar, devemos proceder como segue:

a) Escolhe-se de forma aleatória dois primos de valores altos, p e q. Para maior segurança deve-se escolher dois primos com o mesmo número de bits.

b) Efetua-se o produto entre eles:

$$n = p \times q$$

c) Escolhe-se de forma aleatória um outro número, e (chave para cifrar), de forma que e e (p-1)(q-1) sejam primos relativos, ou seja seu gcd ("greatest common divisor" ou máximo divisor comum) seja 1.

d)Calcula-se d (chave para decifrar), segundo a fórmula:

$$d = e^{-1} \bmod((p-1)(q-1))$$

Podemos notar que d e n também são primos relativos

d) Os números e e n são as chaves públicas e o número d é a chave privada, e vice-versa.

Abaixo está representada uma chave de 1024 bits:

Modulo (1024 bit):

00:f6:9c:64:49:18:7f:c7:47:db:07:b6:a3:43:2e:

ef:6c:7a:56:dd:8a:87:18:37:cb:af:70:ea:5b:33:

96:d8:fa:4c:46:c3:be:f4:0a:6f:e4:d0:31:82:17:

f9:c2:3d:d9:6d:c7:57:79:fe:98:d7:64:12:80:84:

44:89:cd:f9:66:43:d4:ea:d2:54:5b:89:85:23:ff:

18:70:87:7d:f5:37:33:0c:3d:30:53:45:51:e9:4d:

cf:b7:31:5a:c8:a1:a9:3b:80:92:58:8b:a6:0e:a9:

83:16:83:91:3a:3f:99:72:23:5f:8a:dc:a1:1e:34:

73:5f:10:a9:fa:f0:d9:d4:ad

Expoente: 65537 (0x10001)

11.3.2 Cifrando um bloco

a) De posse do bloco a ser criptografado, que deve ser menor do que n, devemos aplicar a seguinte fórmula:

$$c = m^e \bmod n$$

onde:

m representa o bloco numérico a ser cifrado

c representa o bloco criptografado

Por exemplo se desejarmos encriptar a mensagem:

Criptografando

b) Transformamos em código ascii:

067114105112116111031140971020971101001 11

c) Geramos as chaves (para este exemplo usaremos primos de valores pequenos). Por exemplo, para p = 6703 e q = 4909 temos:

e = 365 ; d = 24241997 ; n = 32905027

d) Separamos o texto plano em blocos (consideramos o número de algarismos de m menor do que o de n, como n tem 8 algarismos, utilizaremos blocos de 7 algarismos):

0671141 0511211 6111103 1140971 0209711 0100111

OBSERVAÇÃO: Por coincidência obtivemos todos os blocos com 7 algarismos, mas caso haja algum bloco com um número menor de algarismos devemos preenchê-lo com zeros à esquerda.

e) Aplicando a fórmula de cifragem, obtemos os blocos encriptados abaixo:

32829373 04473412 26575878 24480056 31737173 26591854

f) Assim, a mensagem encriptada é:

3282937304473412265758782448005631737173 26591854

OBSERVAÇÃO: Neste caso deveremos preencher com zeros a esquerda de forma que cada bloco criptografado tenha o mesmo número de algarismos de n, como aconteceu com o segundo bloco em vez de 4473412 escrevemos 04473412. Isto é importante na hora de separar os blocos para decifrar.

11.3.3 Decifrando um bloco

a) Para decriptar, devemos aplicar a fórmula abaixo ao bloco criptografado **c**

$$m_i = c_i^{\,d} \bmod n$$

Para o exemplo anterior temos a mensagem criptografada:

3282937304473412265758782448005631737173 26591854

b) Primeiramente devemos separá-la em blocos de mesmo número de algarismos de n:

32829373 04473412 26575878 24480056 31737173 26591854

c) E aplicar a fórmula de decifragem a cada bloco, gerando o texto decriptografado:

0671141 0511211 6111103 1140971 0209711 0100111

d) Convertendo os códigos ascii:

Criptografando

Temos novamente a mensagem original.

11.4 Implementação

11.4.1 Requisitos de Software

•Microsoft Windows 95, Windows 98 (original and Second Edition), Windows Millennium Edition, Windows NT 4.0 (com Service Pack 5 para correção Y2K ou Service Pack 6a) ou Windows 2000

•MATLAB 6.0 (R12) ou 6.1 (R12.1)

11.4.2 Requisitos de Hardware

•Pentium, Pentium Pro, Pentium II, Pentium III, Pentium IV** ou AMD Athlon

•64 MB RAM (mínimo), 128 MB RAM (recomendadop) adaptador gráfico de 8 bits e monitor (para 256 cores simultâneas)

11.4.3 Descrição dos módulos principais

O RSA composto de vários módulos, os quais devem estar no diretório de trabalho do Matlab.

Os módulos principais são: GeraChaveRSA, CifraBlocoRSA e DecifraBlocoRSA.

GeraChaveRSA

Este módulo é o responsável pela geração das chaves públicas (e e n) e da chave privada (d) utilizadas pelo algoritmo.

Sintaxe:

```
>> [e, n, d] = GeraChaveRSA
```

Onde:

e e *n* são as chaves públicas

d é a chave privada

RSACifra

É o responsável pela geração do criptograma (texto criptografado).

Sintaxe:

>>Cifrado = RSACifra(TextoPlano, e,n)

Onde:

Cifrado é o resultado do processo de cifragem sobre o texto plano

TextoPlano é texto que se quer cifrar

e e **n** são as chaves públicas

RSADecifra

É o responsável pela decifragem, restaurando assim o texto plano original.

Sintaxe:

>>[TextoDecript,BlocMatrix,CriptMatrix] = RSADecifra(TxtoCript,ChaveD,ChaveN)

Onde:

TextoDecript - Texto decriptado

BlocMatrix - matrix contendo os blocos criptografados

CriptMatrix - matrix contendo os blocos decriptografados

TxtoCript - texto criptografado

ChaveD é a chave privada

ChaveN é uma das chaves públicas

Outros Módulos

Os módulos acima são formados de diversos módulos necessários a sua implementação, na próxima seção veremos em detalhes sua composição.

11.4.4 Descrição dos módulos auxiliares

GeraChaveRSA

- GeraPrimo - Gera um numero Primo de N bits, e verifica se o número gerado é primo (PrimoVerif). Para detalhes ver 11.4.5.
  Sintaxe: [Primo] = GeraPrimo(N)

- PrimoVerif - Verificaçao de primos segundo Rabin-Miller. Para detalhes ver 11.4.5.
  Sintaxe: Resultado = PrimoVerif(Primo)

- ExpModular - Calcula a exponenciação modular, ou seja, M^e mod n.
  Sintaxe: Resultado = ExpModular(M,e,n)

- gcd - Calcula o máximo divisor comum entre dois números, se gcd = 1 os números são primos relativos. Para detalhes ver 11.4.5.
  Sintaxe: g = gcd(x,y);

- InvModulo - Calcula o inverso modulo n segundo o "Extended Euclidean Algorithm", ou seja, b-1 mod n. Para detalhes ver 11.4.5.
  Sintaxe: Resultado = InvModulo(b,n)

RSACifra

- ExpModular - descrito acima
  Sintaxe: Resultado = ExpModular(M,e,n)

RSADecifra

- ExpModular - descrito acima
  Sintaxe: Resultado = ExpModular(M,e,n)

11.4.5 Considerações gerais para implementação

11.4.5.1 Verificação de primos - critério de Rabin-Miller

1. Dado um número p para teste

2. Calcula-se b, onde b é o número de vezes que 2 divide p-1 (isto é, 2b é a maior potência de 2 que divide p-1).

3. Calcula-se m, tal que p = 1 + 2b * m

4. Escolhe-se um número aleatório a, tal que a<p

5. j = 0

6. z = am mod p

7. Se z =1 ou z = p-1 então p pode ser primo

8. Se j>0 e z = 1 então p não é primo

9. j = j+1

10. Se j < b e z ¹ p-1 então z = z2 mod p e volta-se para o passo (8). Se z= p-1 então p pode ser primo

11. Se j = b e z ¹ p-1 então p não é primo

Referência: [APPLIED]

11.4.5.2 Geração de Primos

1. Gera-se um número aleatório de n bits, p
   Faz-se o bit de maior ordem igual a 1, para garantir que o número terá n bits.
   Faz-se o bit de menor ordem igual a 1, para garantir que é ímpar.

2. Verifica-se se o número gerado é divisível pelos primos de menor ordem: 3,5,7,11,13,...
   Testando até 7, elimina-se 54% dos ímpares. Testando até 100, elimina-se 76%, até 256, 80%. Em geral testando até n, elimina-se 1.12/ln n dos ímpares.

3. Efetua-se o teste de Rabin-Miller pelo menos 5 vezes, o número gerado deve passar em todos os cinco testes. Se não passar em um teste, gera-se outro número e tenta-se novamente

Referência: [APPLIED]

11.4.5.3 Exponenciação Modular (Método binário)

Sendo **e**(1) o bit mais significativo da chave **e**

1. k = número de bits da chave **e**
2. Se e(1) = 1 então C = M senão C =1
3. Para i = 2 até k
   C = C*C mod n
   Se e(i) = 1 então C=C*M mod n
4. Resultado = C

Referência: [KOC] http://islab.oregonstate.edu/koc/ece679/notes/rsa1.pdf

11.4.5.4 Máximo divisor comum (gcd - greatest common divisor)

Dados dois números x e y desejamos encontrar seu gcd

1. g = 1
2. Enquanto x e y forem pares: x = x/2, y = y/2, g = 2g
3. Enquanto x ¹ 0
   Enquanto x é par: x = x/2
   Enquanto y é par: y = y/2
   t = | x - y | / 2
   Se x >= y então x = t, senão y = t
4. Retorna (g)

Referência: [HANDBOOK]

11.4.5.5 Inverso módular

Suponha que queiramos calcular 28^{-1} mod 75, o Algoritmo Extendido de Euclides nos fornece meios para tanto

Para o caso acima, b=28 e n =75

1. $n_0 = n$
2. $b_0 = b$
3. $t_0 = 0$
4. $t = 1$
5. q = parte inteira de (n_0/t_0)
6. $r = n_0 - q \times b_0$
7. while $r > 0$ do
8. temp = $t_0 - q \times t$
9. if temp>= 0

   then temp = temp mod n
10. if temp < 0 then temp = n - ((-temp)mod n)
11. $t_0 = t$
12. t = temp
13. $n_0 = b_0$
14. $b_0 = r$
15. q = parte inteira de (n_0/b_0)
16. $r = n_0 - q \times b_0$
17. if $b_0 \neq 1$ then

    b nãio tem inverso módulo n

else

$b^{-1} = t \bmod n$

Referência: [CRYPTOGRAPHY]

11.5 Laboratório Virtual

11.5.1 Criptografando com RSA no Laboratório Virtual

Primeiramente geramos uma chave válida para o algoritmo, para tanto devemos selecionar "Obter chaves para", escolher o algoritmo "RSA" e pressionar o botão "Enviar", como ilustrado na figura abaixo:

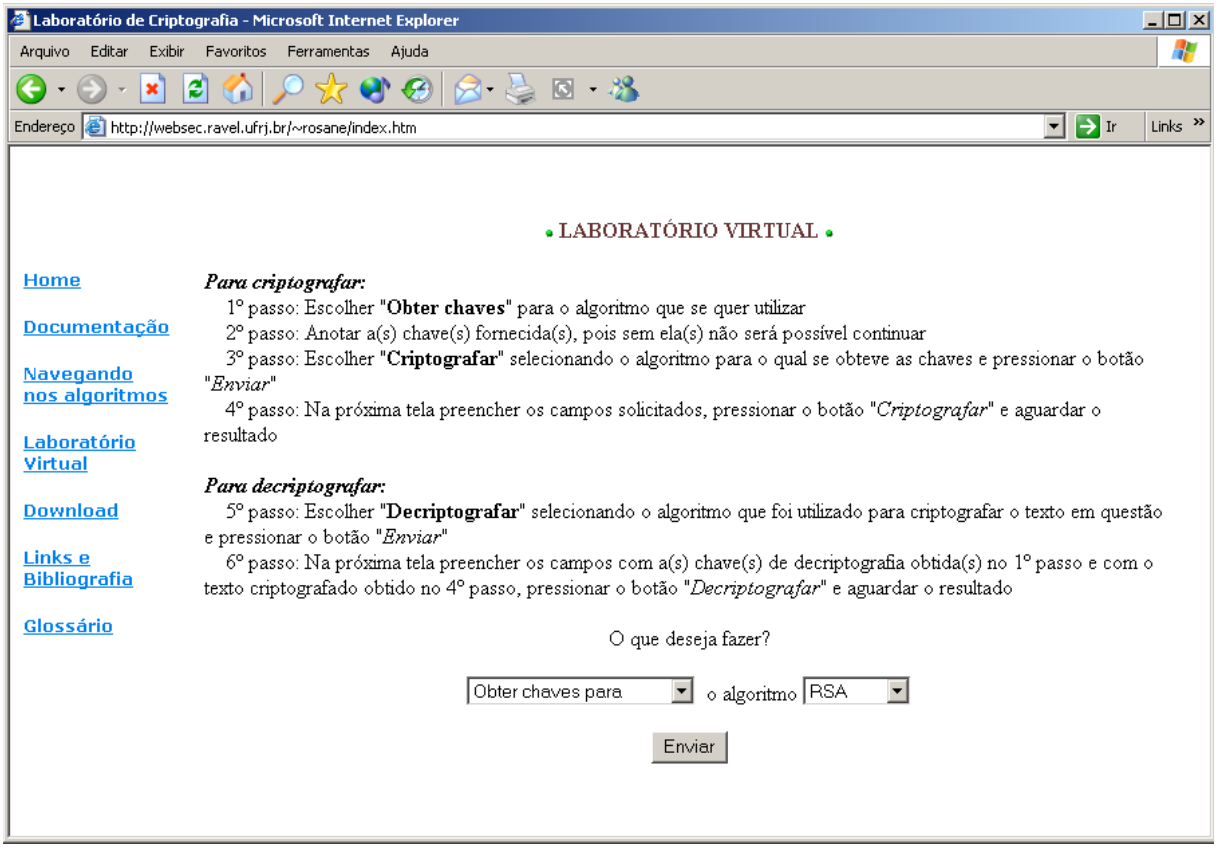


Figura 11.1- Obtendo uma chave válida

Após alguns segundo um Pop-Up como o mostrado na figura abaixo sugirá. Ele contém a chave que poderá ser utilizada no processo de cifragem.

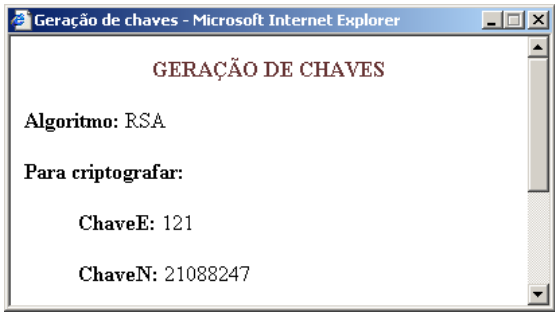


Figura 11.2- Resultado da geração da chave

Chave D: 16375465

Chave N: 21088247

Que são as chaves do processo de decifragem.

Após obtidas as chaves, deve-se escolher "Criptografar utilizando" e selecionar o algoritmo "RSA" como mostra a figura abaixo::

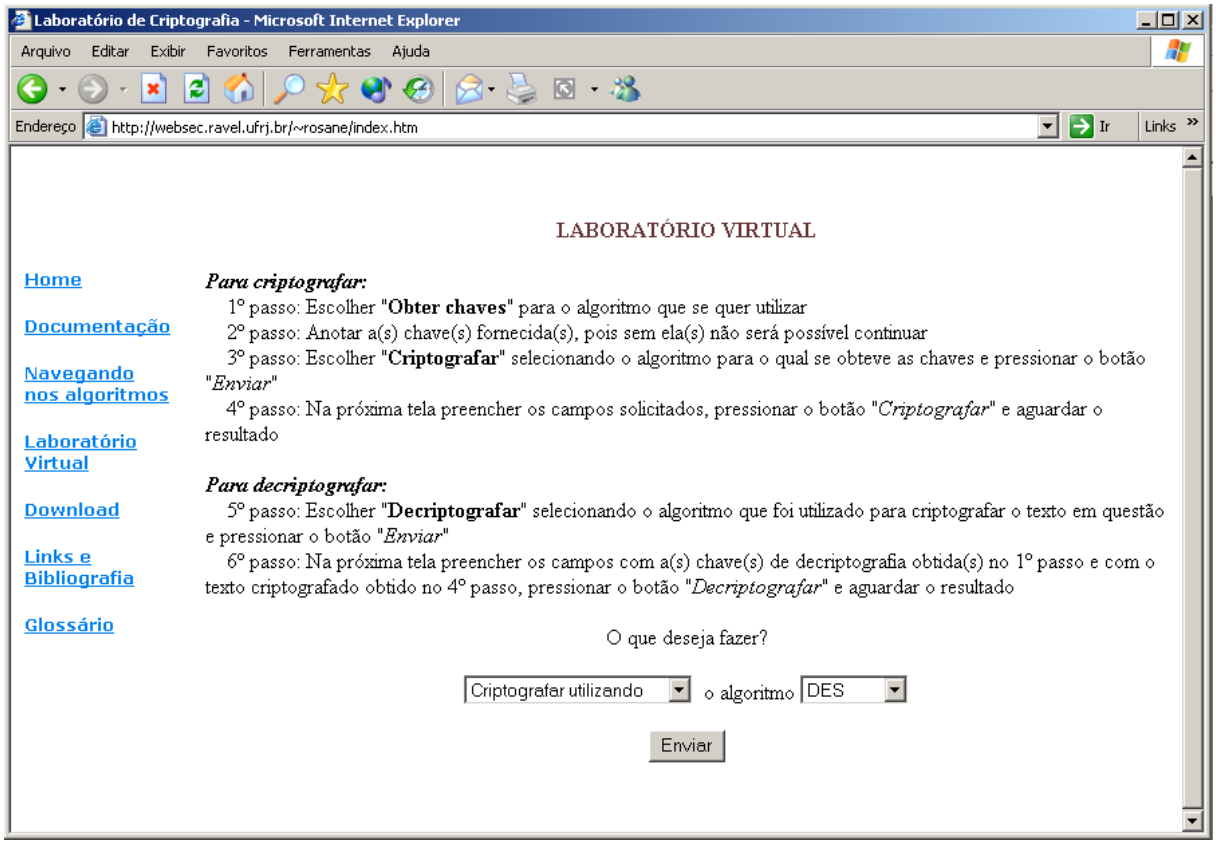


Figura 11.3- Optando pelo processo de cifragem

Após pressionar o botão "Enviar" uma nova tela será apresentada, nela será possível digitar as chaves públicas (as chaves E e N obtidas no passo 1) e a informação a ser criptografada. Como pode ser visto na próxima figura.

A informação a ser criptografada pode conter quaisquer caracteres alfanuméricos inclusive caracteres de pontuação.

Após preencher estes campos deve-se pressionar o botão "Criptografar".

Caso algum erro seja cometido, pode-se utilizar o botão Limpar para remover todas as informações digitadas.

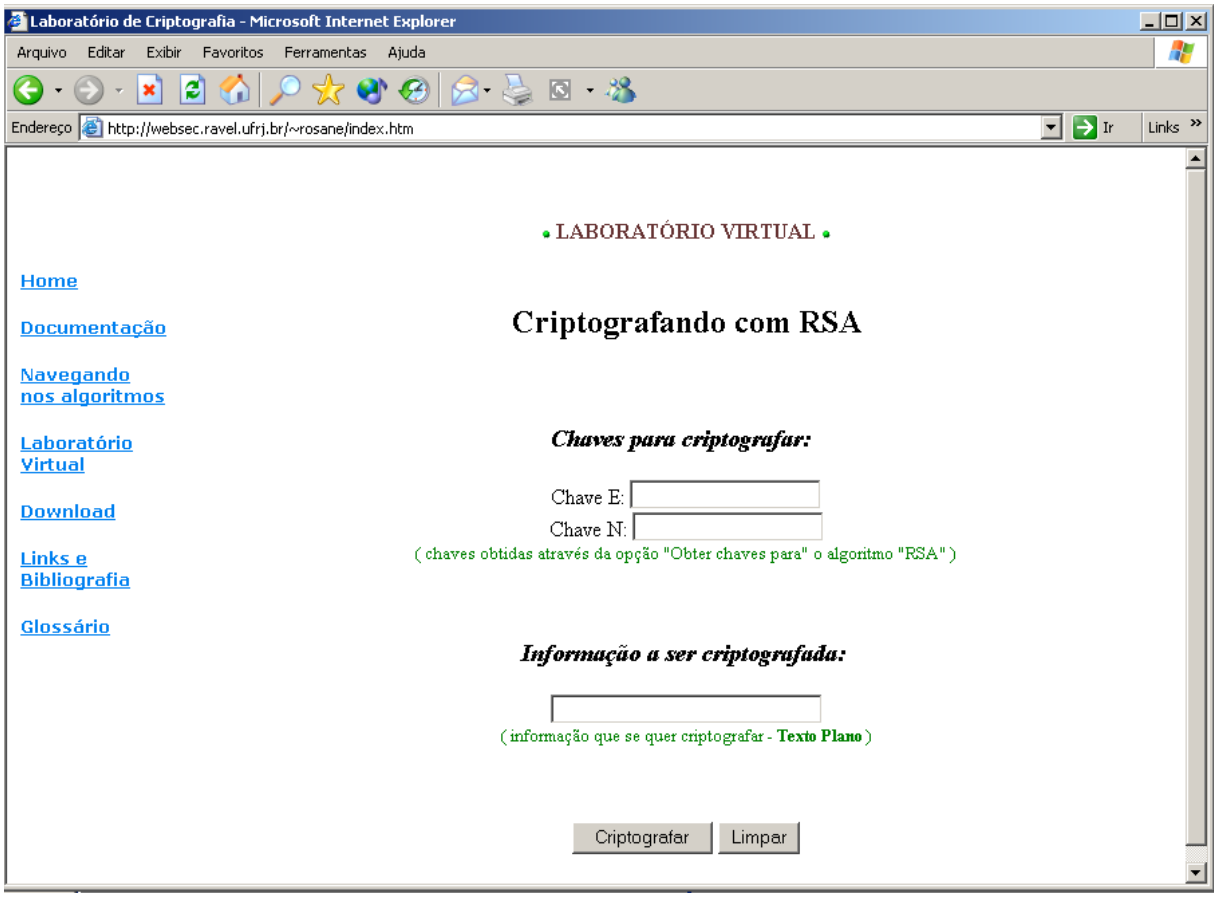


Figura 11.4- Inserindo os dados para cifrar

Após o pressionamento do botão "Criptografar" as informações serão processadas pelo Matlab. Este levará alguns segundos, dependendo do tamanho da informação a ser criptografada.

Após decorrido o tempo necessário, um Pop-Up será exibido para o usuário com o resultado da cifragem.

Neste Pop-Up, além da informação criptografada, serão exibidas também outras informações, como o texto plano, as chaves utilizadas bem como uma figura ilustrativa do processo de cifragem. Esta figura possui áreas clicáveis que revelará alguns detalhes do processo, como por exemplo os blocos em que foram divididos o texto plano antes de ser criptografado, as subchaves geradas no processo, etc.

A figura a seguir mostra um Pop-Up resultante no qual foram utilizadas as seguites informações:

Chave E: 121

Chave N: 21088247

Informação a ser Criptografada: Criptografando com RSA

.

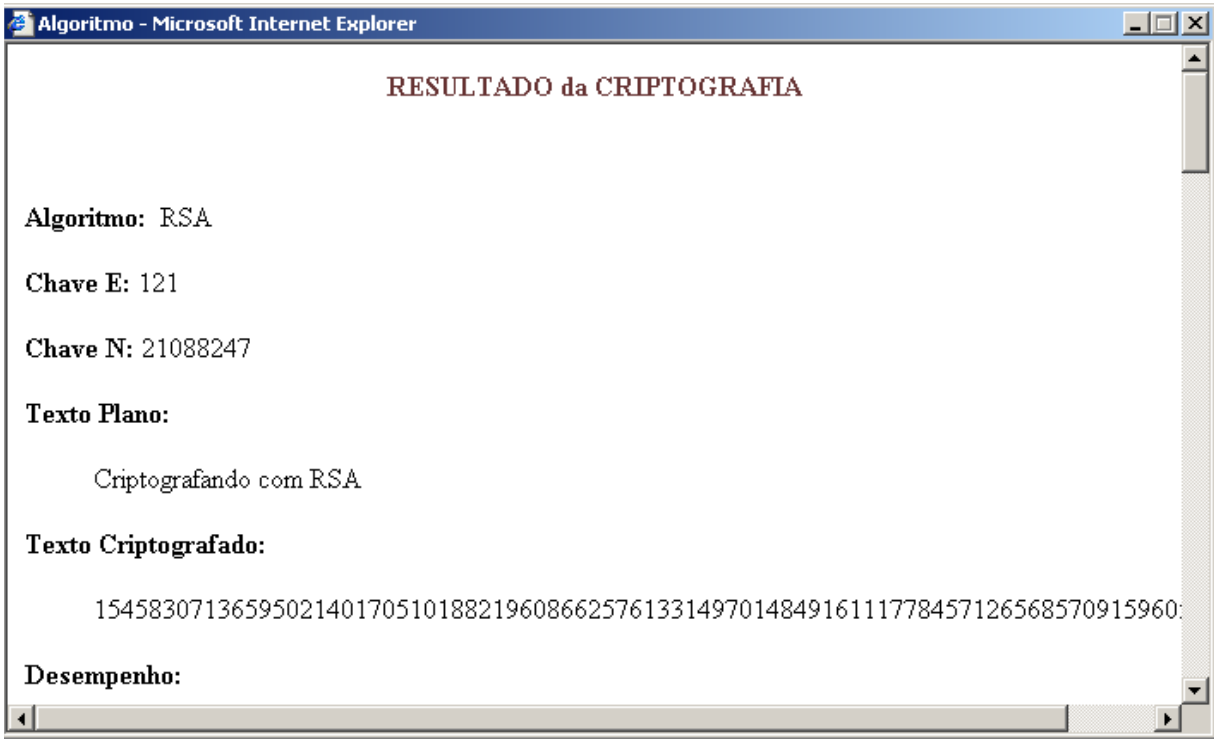


Figura 11.5- Resultado da cifragem

As demais informações do Pop-Up são apresentadas abaixo:

Desempenho:

O texto plano acima demorou 1.8742 segundos para ser criptografado

Segurança:

A segurança do RSA está baseada na dificuldade de fatorar grandes números: as chaves são calculadas matematicamente combinando dois números primos de grande tamanho. Mesmo se conhecendo o produto desses número primos (que faz parte da chave pública divulgada), a segurança do algoritmo é garantida pela complexidade de fatorar esse produto e se obter os valores secretos.

No presente momento, o melhor algoritmo de fatoração tem um tempo de solução na ordem de O(esrqt(ln n x ln(ln n))). Se n tem 200 dígitos (aproximadamente 664 bits) e assumindo-se um computador que possa realizar quatro milhões de passos por segundo (uma suposição generosa, dada a magnitude dos números envolvidos) a fatoração consumiria 4 milhões de anos. E se mais segurança for necessária, basta aumentar n por alguns bits.

. Referência: Applied Cryptography, Second Edition: Protocols, Algorthms, and Source Code in C, by Bruce Schneier

Detalhes da cifragem:

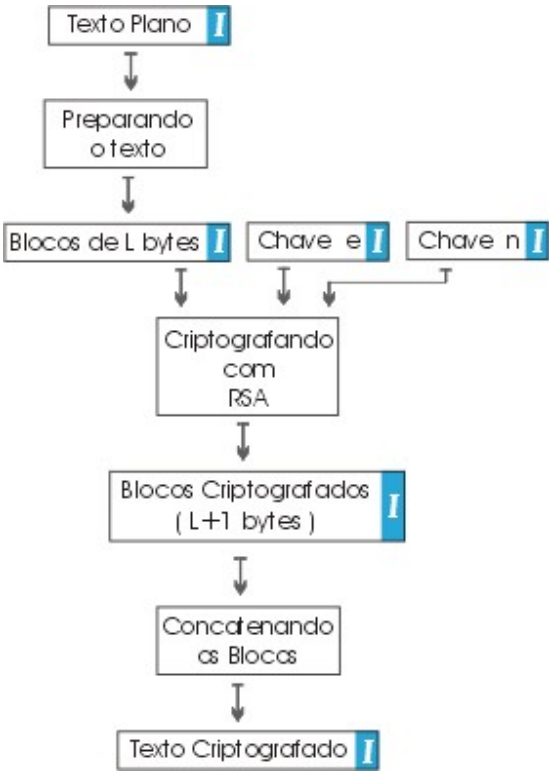


Figura 11.6- Diagrama da cifragem

Texto Plano:

Criptografando com RSA

voltar ao diagrama

Blocos de L bytes:

0671141
0511211
6111103
1140971
0209711
0100111
0320991
1110903
2082083
0000065

voltar ao diagrama

Chave E:

121

voltar ao diagrama

Chave N:

21088247

voltar ao diagrama

Blocos Criptografados:

15458307
13659502
14017051
1882196
8662576
13314970
14849161
11778457
12656857
9159605

voltar ao diagrama

Texto Criptografado:

1545830713659502140170510188219608662676133149701484916111778457126568570 91

voltar ao diagrama

11.5.2 Decriptografando com RSA no Laboratório Virtual

Como já possuímos a Chave Secreta devemos passar direto à decifragem, para tanto devemos escolher "Decriptografar utilizando" o algoritmo "RSA", como mostra a figura abaixo:

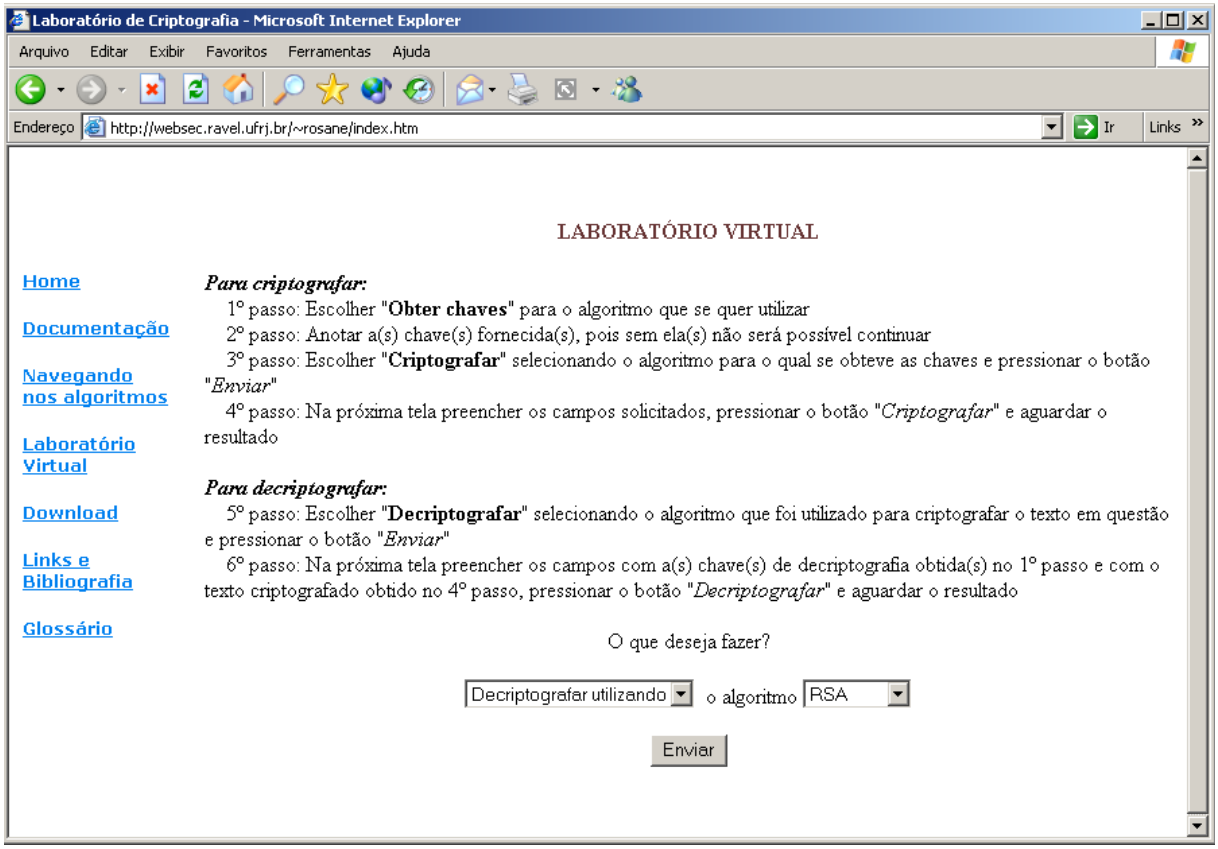


Figura 11.7- Optando pelo processo de decifragem

Semelhante ao processo de cifragem ao pressionarmos o botão Enviar uma nova tela se abrirá, como mostra a figura a seguir.

Nesta tela será possível digitar as Chaves a serem utilizadas para decifrar a informação (ATENÇÃO: como o RSA é um algoritmo assimétrico, as chaves utilizadas não são todas iguais, para cifrar utilizamos a Chave E e a Chave N, para decifrar deveremos utilizar a Chave D e a Chave N).

Também será possível digitar a informação a ser decriptografada, ou seja, o texto ininteligível obtido através do processo de cifragem..

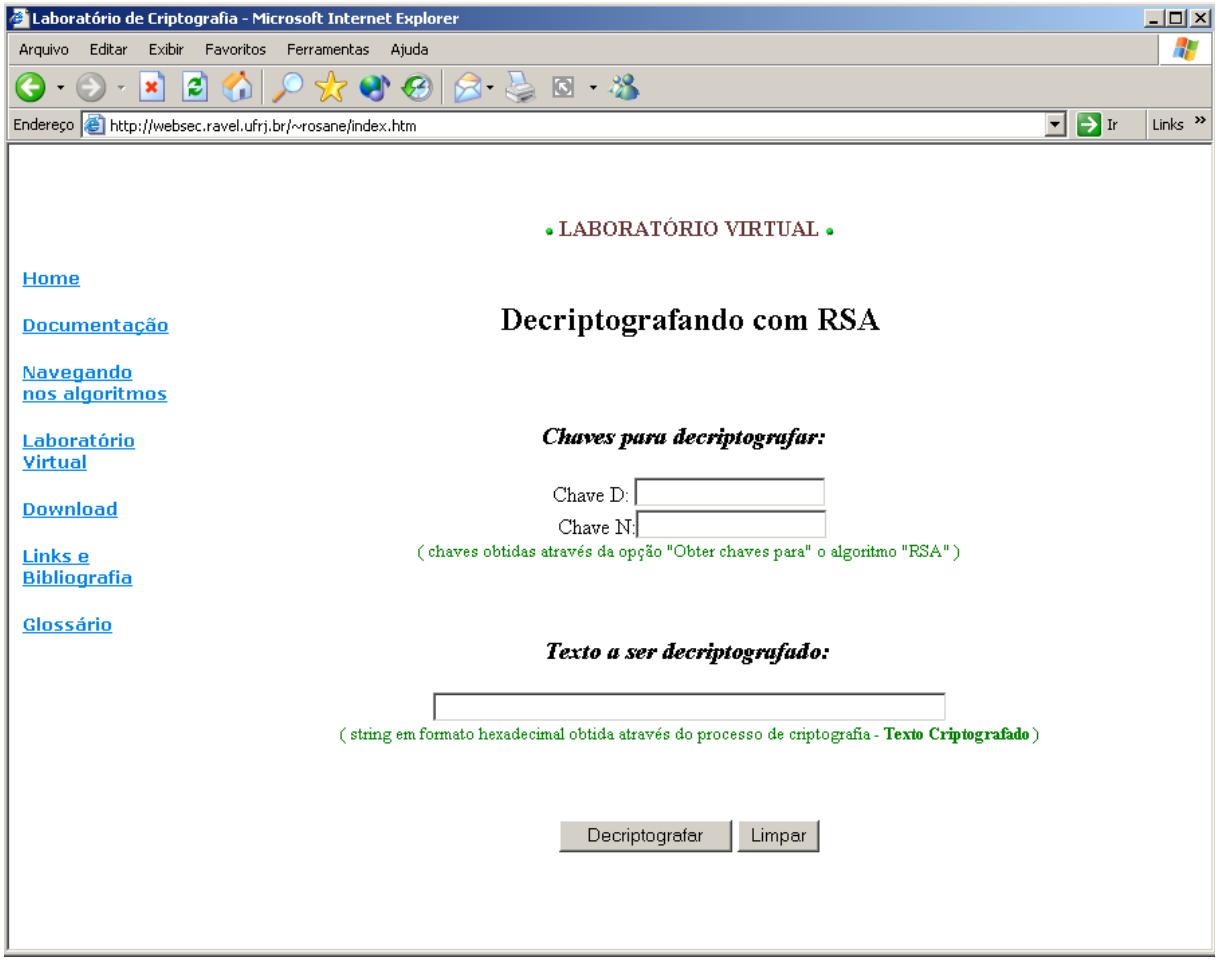


Figura 11.8- Inserindo dados para decifrar

Após o pressionamento do botão "Decriptografar" as informações serão processadas pelo Matlab. Este processo poderá levar alguns minutos, dependendo do tamanho da informação a ser decriptografada.Após decorrido o tempo necessário, um Pop-Up será exibido para o usuário com o resultado da decifragem.

Neste Pop-Up, além da informação decriptografada, serão exibidas também outras informações, como o texto criptografado, a chave utilizada bem como uma figura ilustrativa do processo de decifragem. Esta figura possui áreas clicáveis que revelará alguns detalhes do processo, como por exemplo os blocos em que foram divididos o texto criptografado antes de ser decriptografado, as subchaves geradas no processo, etc.

A figura a seguir mostra um Pop-Up resultante no qual foram utilizadas as seguites informações:

Chave D: 16375465

Chave N: 21088247

Informação a ser Decriptografada:

154583071365950214017051018821960866257613314970148491611177845712656857091
59605

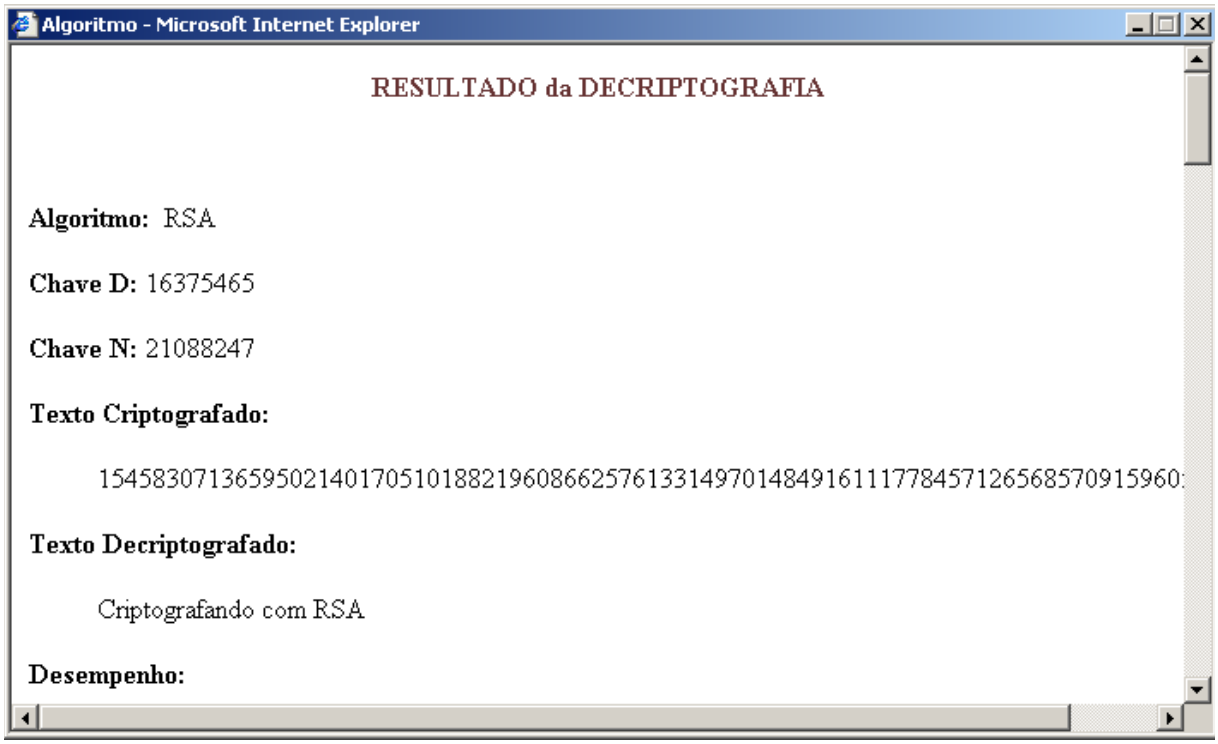


Figura 11.9- Resultado da decifragem

As demais informações do Pop-Up são apresentadas abaixo:

Desempenho:

O texto plano acima demorou 1.8183 segundos para ser criptografado

Segurança:

A segurança do RSA está baseada na dificuldade de fatorar grandes números: as chaves são calculadas matematicamente combinando dois números primos de grande tamanho. Mesmo se conhecendo o produto desses número primos (que faz parte da chave pública divulgada), a segurança do algoritmo é garantida pela complexidade de fatorar esse produto e se obter os valores secretos.

No presente momento, o melhor algoritmo de fatoração tem um tempo de solução na ordem de O(esrqt(ln n x ln(ln n))). Se n tem 200 dígitos (aproximadamente 664 bits) e assumindo-se um computador que possa realizar quatro milhões de passos por segundo (uma suposição generosa, dada a magnitude dos números envolvidos) a fatoração consumiria 4 milhões de anos. E se mais segurança for necessária, basta aumentar n por alguns bits.

. Referência: Applied Cryptography, Second Edition: Protocols, Algorthms, and Source Code in C, by Bruce Schneier

Detalhes da cifragem:

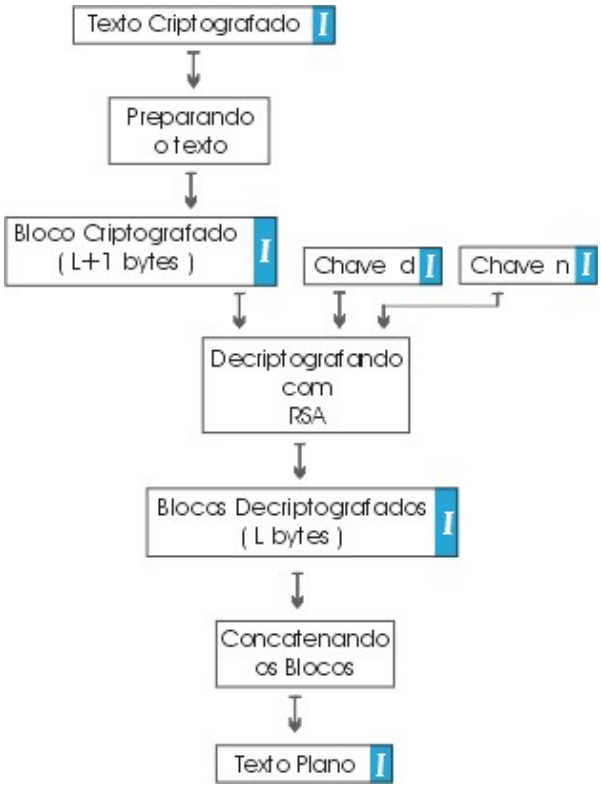


Figura 11.10- Diagrama da decifragem

Texto Criptografado:

154583071365950214017051018821960866257613314970148491611177845712656857091596 05

voltar ao diagrama

Blocos criptografados:

15458307
13659502

14017051
01882196
08662576
13314970
14849161
11778457
12656857
09159605

voltar ao diagrama

Chave D:

16375465

voltar ao diagrama

Chave N:

21088247

voltar ao diagrama

Blocos Decriptografados:

671141
511211
6111103
1140971
209711
100111
320991
1110903
2082083
65

voltar ao diagrama

Texto Decriptografado:

Criptografando com RSA

voltar ao diagrama

12 El Gamal

12.1 História

O Algoritmo de ElGamal, assim como o RSA é baseado no problema do logaritmo discreto.

O primeiro foi publicado por ElGamal em "A public-key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans. Inf. Theory TI-31 (1985) 469-472", enquanto o segundo foi publicado pelo NIST em "The Digital Signature Standard. Commun. of ACM 35(7), 1992, 36-40.".

Na sua forma mais simples, o sistema do logaritmo discreto tenta descobrir o expoente x na fórmula $y = g^x \bmod p$.

A segurança dos sistemas depende da dificuldade de determinar o valor de x.

Modificações desse problema utilizam a exponencial discreta, que parece ser mais segura que o problema original.

12.2 Características

•Baseado no problema matemático de computar logaritmos discretos de números inteiros grandes (Teoria dos Números)

•Maduro e bastante estudado

•Permite sigilo e autenticação

•Chaves de 512 e 4096 bits

•Possui 3 chaves públicas e 1 chave privada

•Possui 1 chave para cada sessão

•Mensagem criptografada em forma do pares: (Chave de sessão, Bloco criptografado)

•Provê Confidencialidade, Autenticação, Não-repúdio e Integridade

12.3 Funcionamento do algoritmo

12.3.1 Obtenção das chaves:

a) Primeiramente é gerado um número primo **p**, e dois outros valores quaisquer **a** e **alfa**.

b) De posse desses valores geramos beta segundo a fórmula abaixo:

$$\beta \equiv \alpha^{a} \bmod p$$

Os valores alfa, beta e p são públicos e a é secreto

12.3.2 Cifrando um bloco

a) Antes de começar a cifragem propriamente dita, devemos gerar uma chave de sessão, para tanto devemos escolher um valor randômico **k** < **p**.

b) De posse do valor de **k**, geramos a chave de sessão y segundo a fórmula abaixo:

$$y_1 = \alpha^k \bmod p$$

c) A partir daí podemos gerar o bloco criptografado da seguinte forma:

$$y_2 = x \times \beta^k \bmod p$$

onde **x** representa o bloco em texto plano.

d) O bloco de texto plano deve ser escolhido de forma que **x** < **p**

e) A chave de sessão deve ser gerada para cada bloco separadamente. Ou seja, cada bloco terá sua própria chave **y**, assim o texto criptografado consiste no par (y_1, y_2).

f) Por exemplo para a mensagem:

Criptografando

g) Transformamos em código ascii:

067114105112116111103114097102097110100111

h) Utilizando as chaves:

p = 7457 ; alfa = 4 ; a = 93 e beta = 725

i) Separamos o texto plano em blocos (consideramos o número de algarismos de x menor do que o de p, como p tem 4 algarismos, utilizaremos blocos de 3 algarismos):

067 114 105 112 116 111 103 114 097 102 097 110 100 111

j) Criptografando temos e_i (x,k) = (y_1, y_2):

e1 (067, 271) = (0526, 2208)

e2 (114, 252) = (4341, 1600)

e3 (105, 876) = (1852, 4800)

e4 (112, 737) = (3711, 2769)

e5 (116, 137) = (4738, 2400)

e6 (111, 012) = (4096, 1181)

e7 (103, 894) = (3963, 0318)

e8 (114, 199) = (2987, 5578)

e9 (097, 299) = (4034, 5564)

e10 (102, 661) = (2686, 1576)

e11 (097, 284) = (2656, 4753)

e12 (110, 469) = (1547, 3305)

e13 (100, 065) = (1277, 0897)

e14 (111, 988) = (0824, 2753)

Ou seja, a mensagem criptografada é:

(0526, 2208) (4341, 1600) (1852, 4800) (3711, 2769) (4738, 2400)

(4096, 1181) (3963, 0318) (2987, 5578) (4034, 5564) (2686, 1576)

(2656, 4753) (1547, 3305) (1277, 0897) (0824, 2753)

12.3.3 Decifrando um bloco

a) De posse do par (y_1, y_2), que consiste no bloco criptografado, aplicamos a fórmula abaixo:

$$d(y_1, y_2) = y_2 \times (y_1^a)^{-1} \bmod p$$

obtendo assim, o bloco decriptografado.

b) Para o exemplo acima, utilizando a chave secreta a = 93 chegamos novamente no texto:

067 114 105 112 116 111 103 114 097 102 097 110 100 111

12.4 Implementação

12.4.1 Requisitos de Software

•Microsoft Windows 95, Windows 98 (original and Second Edition), Windows Millennium Edition, Windows NT 4.0 (com Service Pack 5 para correção Y2K ou Service Pack 6a) ou Windows 2000

•MATLAB 6.0 (R12) ou 6.1 (R12.1)

12.4.2 Requisitos de Hardware

•Pentium, Pentium Pro, Pentium II, Pentium III, Pentium IV** ou AMD Athlon

•64 MB RAM (mínimo), 128 MB RAM (recomendadop) adaptador gráfico de 8 bits e monitor (para 256 cores simultâneas)

12.4.3 Descrição dos módulos principais

O El Gamal é composto de vários módulos, os quais devem estar no diretório de trabalho do Matlab.

Os módulos principais são: GeraChaveElGamal, CifraBlocoElGamal e DecifraBlocoElGamal.

GeraChaveElGamal

Este módulo é o responsável pela geração das chaves públicas (alfa, beta e p) e da chave privada (a) utilizadas pelo algoritmo.

Sintaxe:

>>[p, alfa, a, beta] = GeraChaveElGamal

Onde:

p, *alfa* e *beta* são as chaves públicas

a é a chave privada

ElGamalCifra

É o responsável pela cifragem da mensagem.

Sintaxe:

>>Cifrado = ElGamalCifra(TextoPlano, beta, alfa, p)

Onde:

Cifrado é o resultado do processo de cifragem sobre o texto plano

TextoPlano é texto que se quer cifrar

Beta, *alfa* e *p* são as chaves públicas

ElGamalDecifra

É o responsável pela decifragem do criptograma (texto criptografado).

Sintaxe:

>>Decifrado = ElGamalDecifra(Cifrado,a,p)

Onde:

Decifrado é o texto decifrado

Cifrado é o texto que se quer decifrar

a é a chave privada

p é uma das chaves públicas

Outros Módulos

Os módulos acima são formados de diversos módulos necessários a sua implementação, na próxima seção veremos em detalhes sua composição.

12.4.4 Descrição dos módulos auxiliares

GeraChaveElGamal

o GeraPrimo - Gera um numero Primo de N bits

Sintaxe: [Primo] = GeraPrimo(N)

o ExpModular - Calcula a exponenciação modular, ou seja, Me mod n

Sintaxe: Resultado = ExpModular(M,e,n)

ElGamalCifra

o ExpModular - descrito acima

Sintaxe: Resultado = ExpModular(M,e,n)

o CifraBlocoElGamal - responsável pela cifragem de uma quantidade limitada de caracteres (dependentes do tamanho da chave p), um bloco.

Sintaxe: BlocoCifrado = CifraBlocoRSA(Bloco,beta,k,p)

ElGamalDecifra

o DecifraBlocoElGamal - responsável pela decifragem de uma quantidade limitada de caracteres, um bloco.

Sintaxe: BlocoDecifrado = DecifraBlocoRSA(Bloco,y1,a,p)

o InvModulo - calcula o inverso modulo n segundo o "Extended Euclidean Algorithm", ou seja, b-1 mod n

Sintaxe: Resultado = InvModulo(b,n)

o ExpModular - descrito acima

Sintaxe: Resultado = ExpModular(M,e,n)

12.5 Laboratório Virtual

12.5.1 Criptografando com El Gamal no Laboratório Virtual

Primeiramente geramos uma chave válida para o algoritmo, para tanto devemos selecionar "Obter chaves para", escolher o algoritmo “El Gamal" e pressionar o botão "Enviar", como ilustrado na figura abaixo:

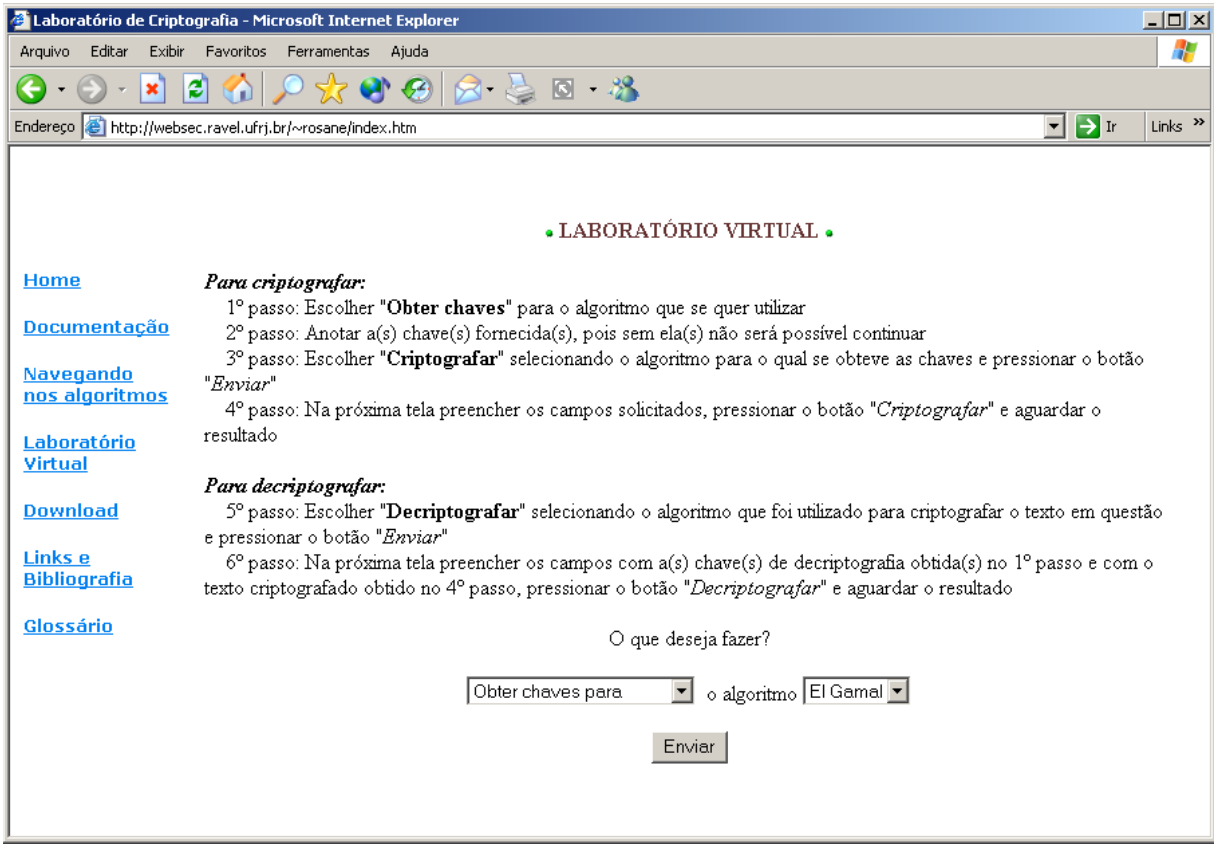


Figura 12.1- Obtendo uma chave válida

Após alguns segundo um Pop-Up como o mostrado na figura abaixo sugirá. Ele contém a chave que poderá ser utilizada no processo de cifragem.

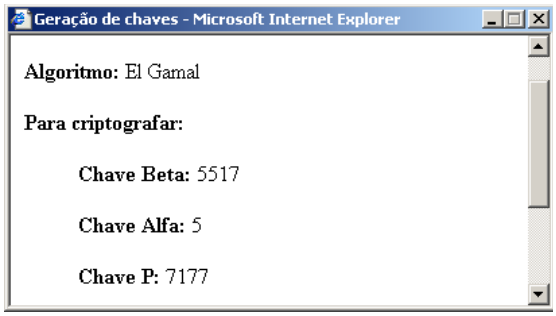


Figura 12.2- Resultado da geração da chave

Este processo gera as chaves de cifragem e de decifragem, movendo a barra de rolagem tem-se acesso as demais chaves que estão descritas abaixo:

Chave A: 46

Chave P: 7177

Que são as chaves do processo de decifragem.

Após obtidas as chaves, deve-se escolher "Criptografar utilizando" e selecionar o algoritmo "El Gamal" como mostra a figura abaixo:

:

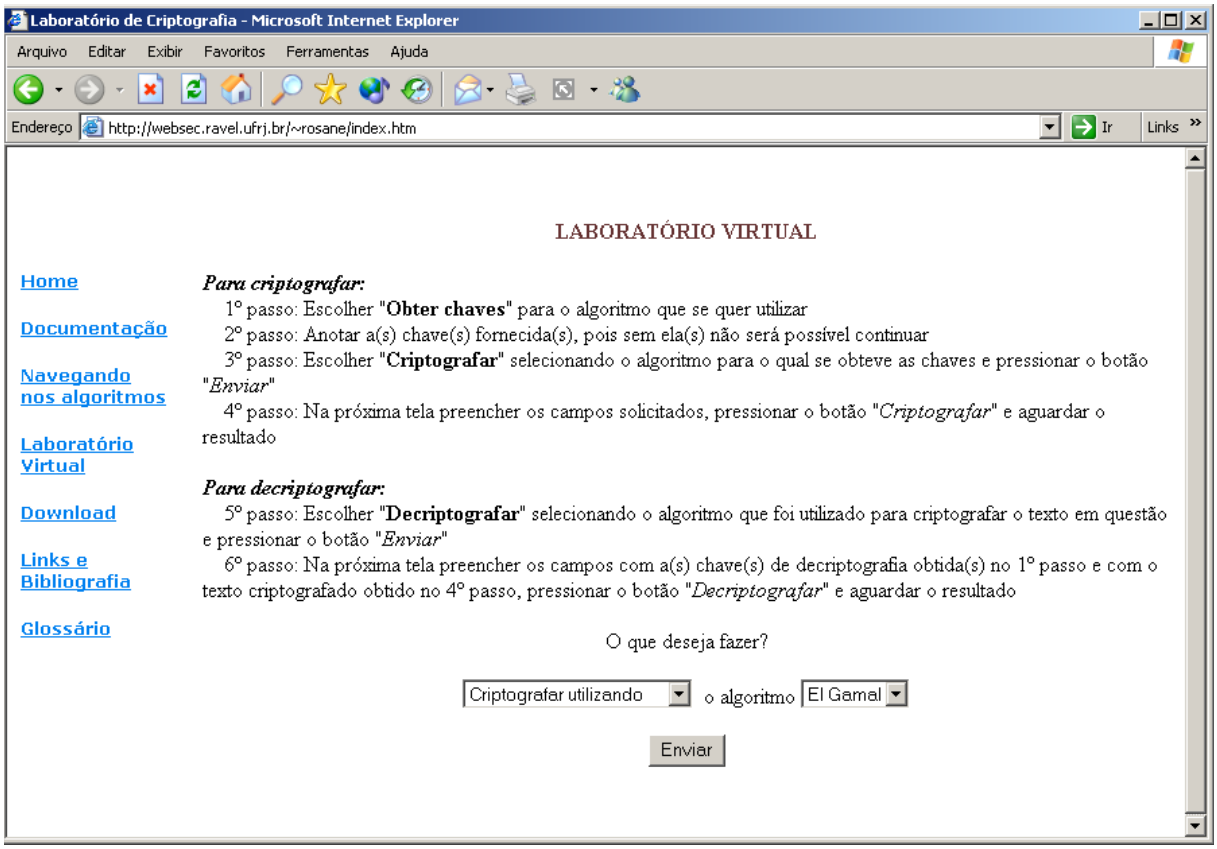


Figura 12.3- Optando pelo processo de cifragem

Após pressionar o botão "Enviar" uma nova tela será apresentada, nela será possível digitar as chaves públicas (as chaves Alfa, Beta e P obtidas no passo 1) e a informação a ser criptografada. Como pode ser visto na próxima figura.

A informação a ser criptografada pode conter quaisquer caracteres alfanuméricos inclusive caracteres de pontuação.

Após preencher estes campos deve-se pressionar o botão "Criptografar".

Caso algum erro seja cometido, pode-se utilizar o botão Limpar para remover todas as informações digitadas.

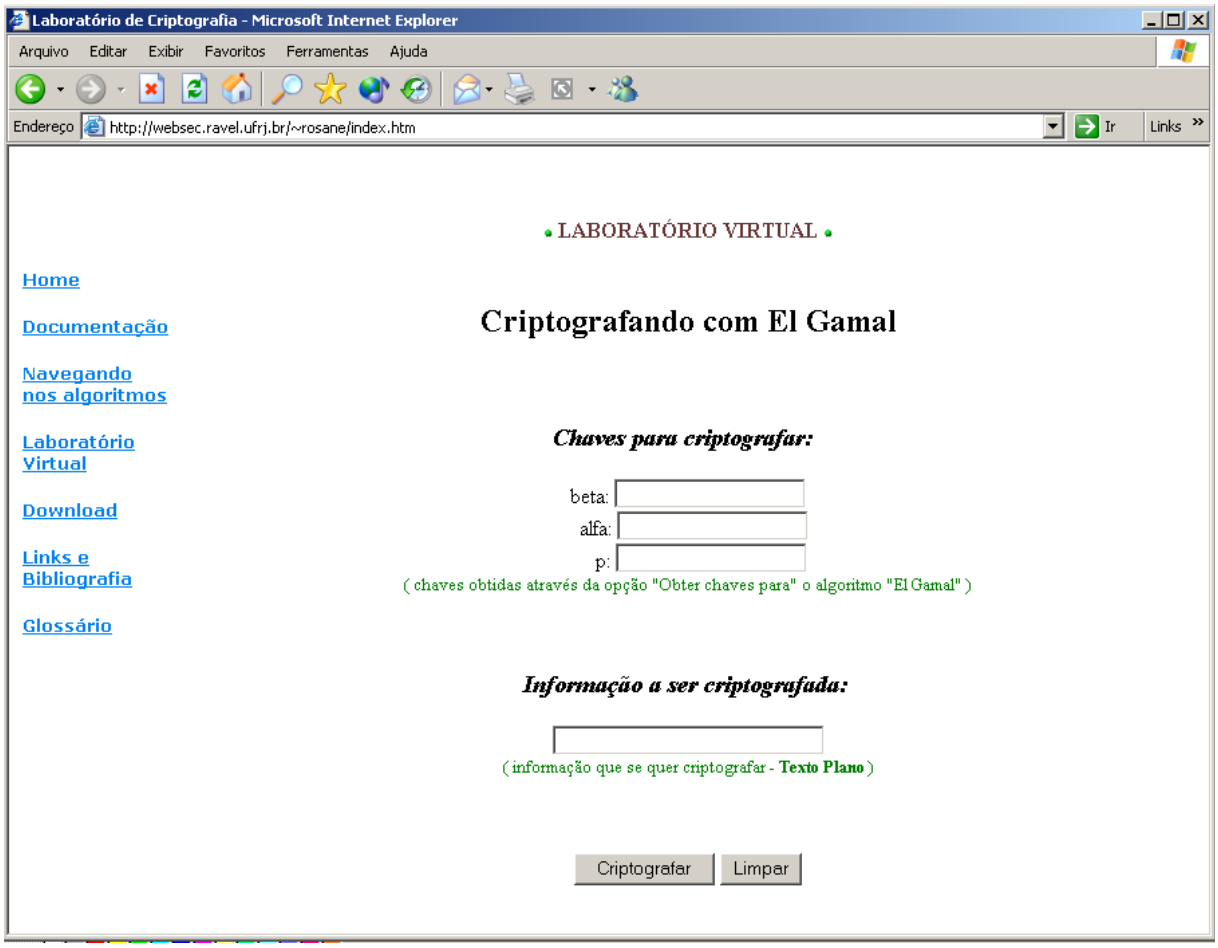


Figura 12.4- Inserindo os dados para cifrar

Após o pressionamento do botão "Criptografar" as informações serão processadas pelo Matlab. Este levará alguns segundos, dependendo do tamanho da informação a ser criptografada.

Após decorrido o tempo necessário, um Pop-Up será exibido para o usuário com o resultado da cifragem.

Neste Pop-Up, além da informação criptografada, serão exibidas também outras informações, como o texto plano, as chaves utilizadas bem como uma figura ilustrativa do processo de cifragem. Esta figura possui áreas clicáveis que revelará alguns detalhes do

processo, como por exemplo os blocos em que foram divididos o texto plano antes de ser criptografado, as subchaves geradas no processo, etc.

A figura a seguir mostra um Pop-Up resultante no qual foram utilizadas as seguites informações:

Chave Beta: 5517

Chave Alfa: 5

Chave P: 7177

Informação a ser Criptografada: Criptografando com El Gamal.

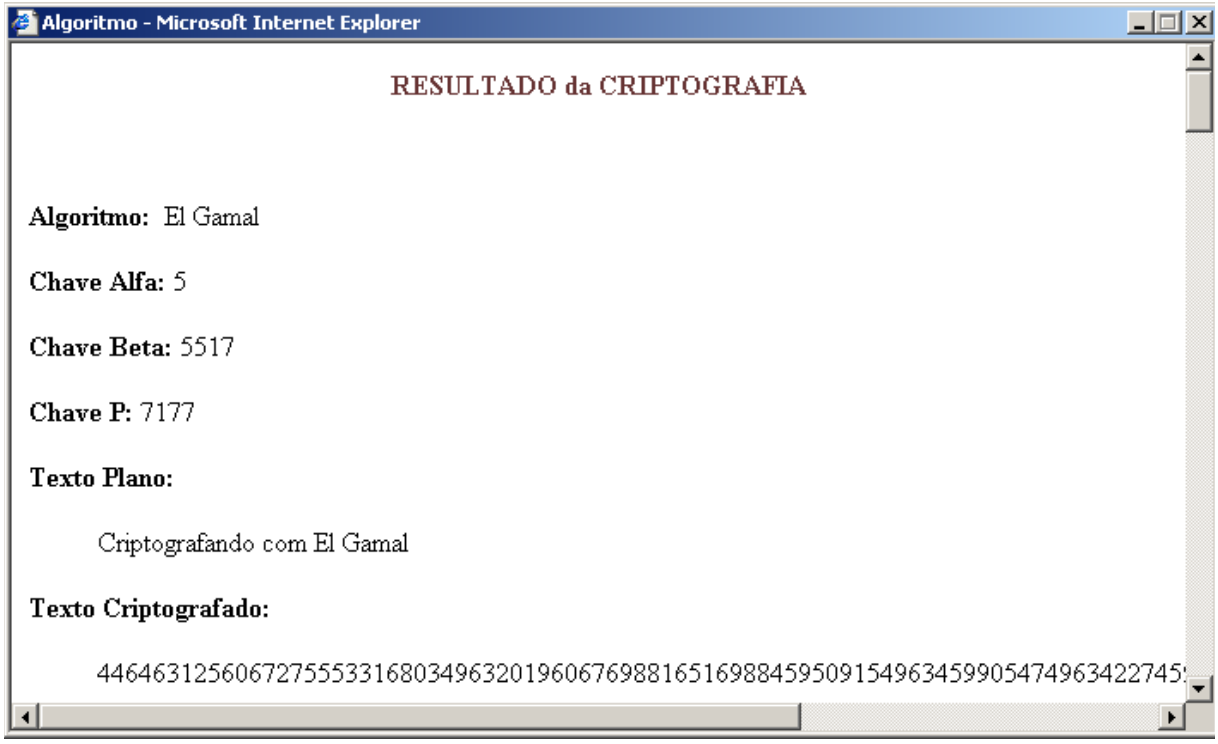


Figura 12.5- Resultado da cifragem

As demais informações do Pop-Up são apresentadas a seguir:

Desempenho:

O texto plano acima demorou 3.0337 segundos para ser criptografado

Segurança:

A forma mais perigosa de ataque contra o sistema El Gamal é resolver o logaritmo discreto que revele a chave A. Qual o número estimado de operações para resolver esse problema de logaritmo discreto? Levando-se em conta que a complexidade da operação é Lp(1/3,c) para uma constante c (< (64/9)1/3), o tempo para quebrar o El Gamal não é polinomial, mas exponencial!

Referência: Applied Cryptography, Second Edition: Protocols, Algorthms, and Source Code in C, by Bruce Schneier

Detalhes da cifragem:

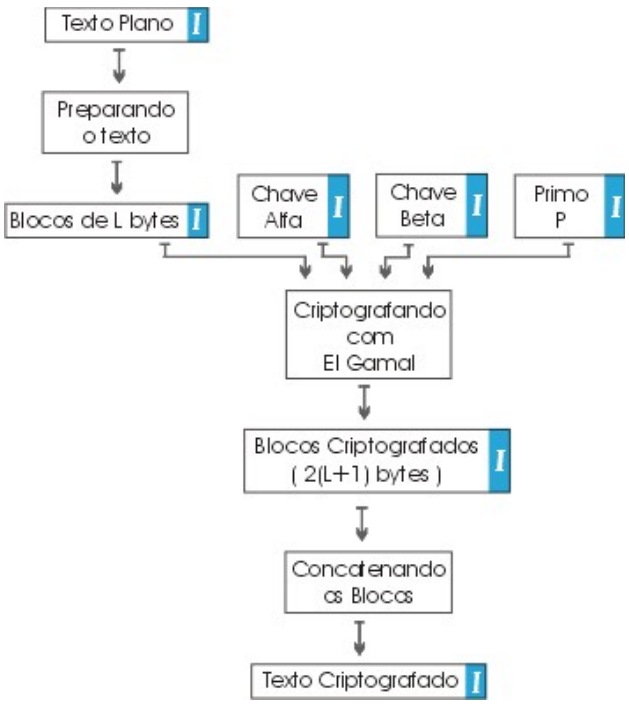


Figura 12.6- Diagrama da cifragem

Texto Plano:

Criptografando com El Gamal

voltar ao diagrama

Blocos a serem criptografados:

067
114
105
112
116
111
103
114
097
102
097
110
100
111
032
099
111
109
032
069
108
032
071
097
109
097
108

voltar ao diagrama

Chave Alfa:

5

voltar ao diagrama

Chave Beta:

5517

voltar ao diagrama

Chave P:

7177

voltar ao diagrama

Blocos Criptografados:

3125
6067
2755
5331
6803
4963
2019
6067
6988
1651
6988
4595
915
4963
4599
547
4963
4227
4599
3861
3859
4599
4597
6988
4227
6988
3859

voltar ao diagrama

Texto Criptografado:

44646312560672755533168034963201960676988165169884595091549634599054749634227459938613859459945976988422769883859

voltar ao diagrama

12.5.2 Decriptografando com El Gamal no Laboratório Virtual

Como já possuímos as chaves devemos passar direto à decifragem, para tanto devemos escolher "Decriptografar utilizando" o algoritmo "ElGamal", como mostra a figura abaixo:

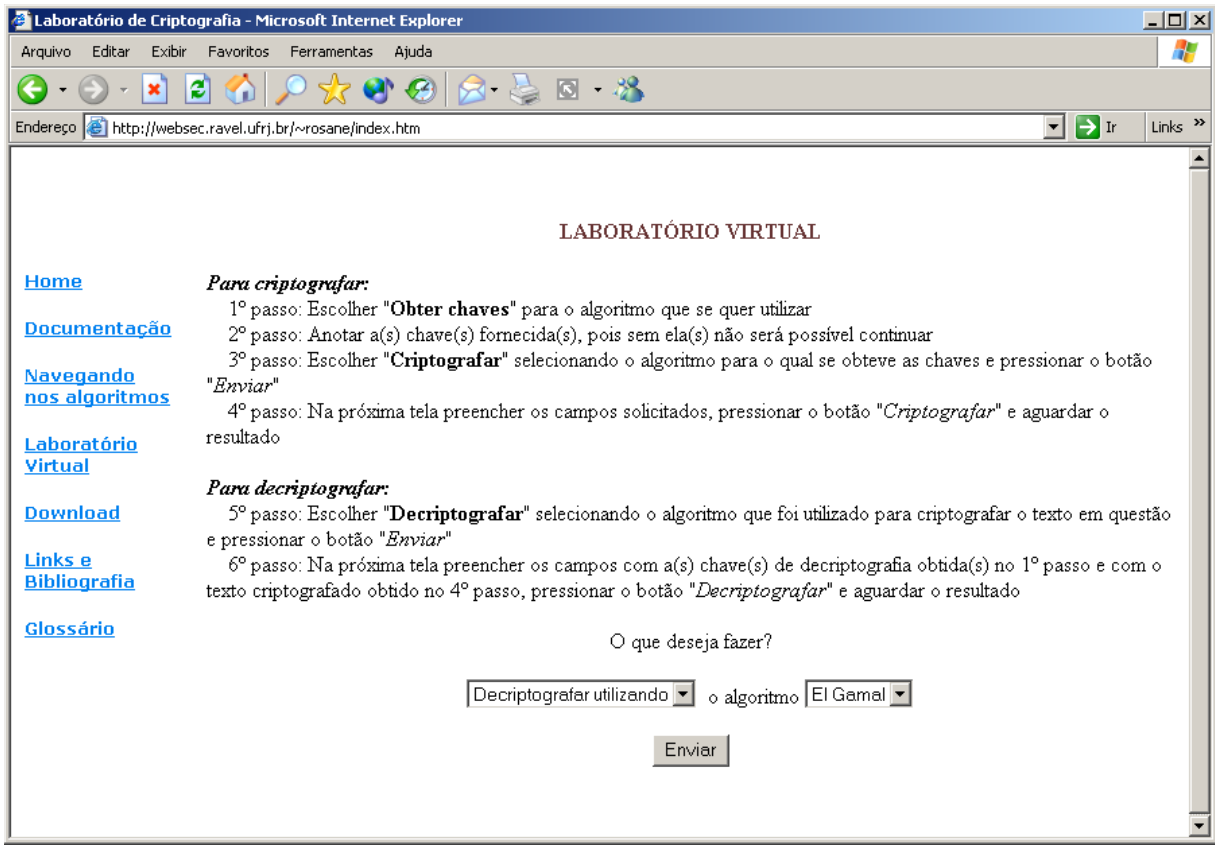


Figura 12.7- Optando pelo processo de decifragem

Semelhante ao processo de cifragem ao pressionarmos o botão Enviar uma nova tela se abrirá, como mostra a figura a seguir.

Nesta tela será possível digitar as Chaves a serem utilizadas para decifrar a informação (ATENÇÃO: como o RSA é um algoritmo assimétrico, as chaves utilizadas não são todas iguais, para cifrar utilizamos a Chave Alfa, a Chave Beta e a Chave P, para decifrar deveremos utilizar a Chave A e a Chave P).

Também será possível digitar a informação a ser decriptografada, ou seja, o texto ininteligível obtido através do processo de cifragem.

Semelhante ao processo de cifragem ao pressionarmos o botão Enviar uma nova tela se abrirá, como mostra a figura a seguir.

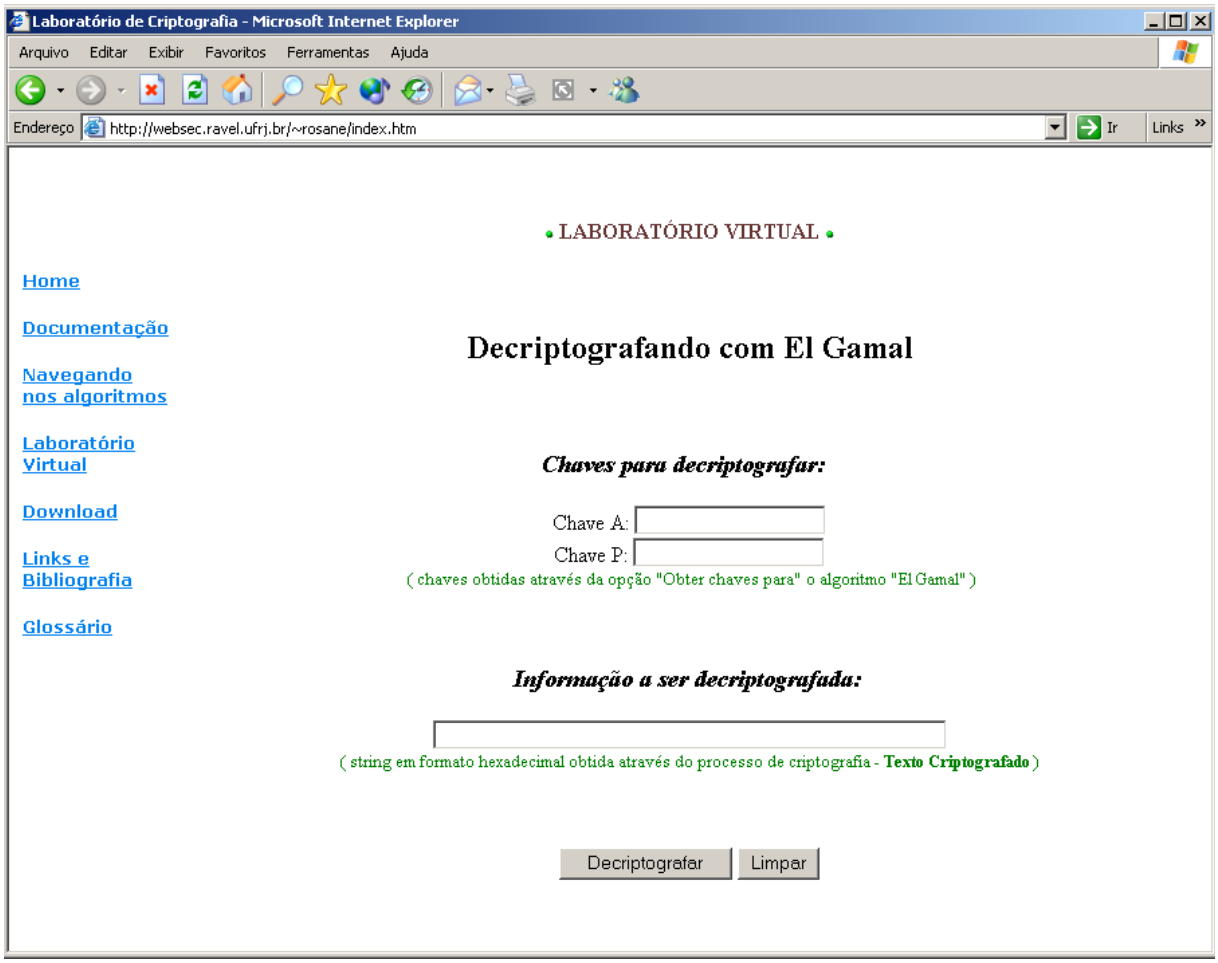


Figura 12.8- Inserindo dados para decifrar

Após o pressionamento do botão "Decriptografar" as informações serão processadas pelo Matlab. Este processo poderá levar alguns minutos, dependendo do tamanho da informação a ser decriptografada. Após decorrido o tempo necessário, um Pop-Up será exibido para o usuário com o resultado da decifragem.

Neste Pop-Up, além da informação decriptografada, serão exibidas também outras informações, como o texto criptografado, a chave utilizada bem como uma figura ilustrativa do processo de decifragem. Esta figura possui áreas clicáveis que revelará alguns detalhes do processo, como por exemplo os blocos em que foram divididos o texto criptografado antes de ser decriptografado, as subchaves geradas no processo, etc.

A figura a seguir mostra um Pop-Up resultante no qual foram utilizadas as seguites informações:

Chave A: 46

Chave P: 7177

Informação a ser Decriptografada:

44646312560672755533168034963201960676988165169884595091549634599054749634227459938613859459945976988422769883859

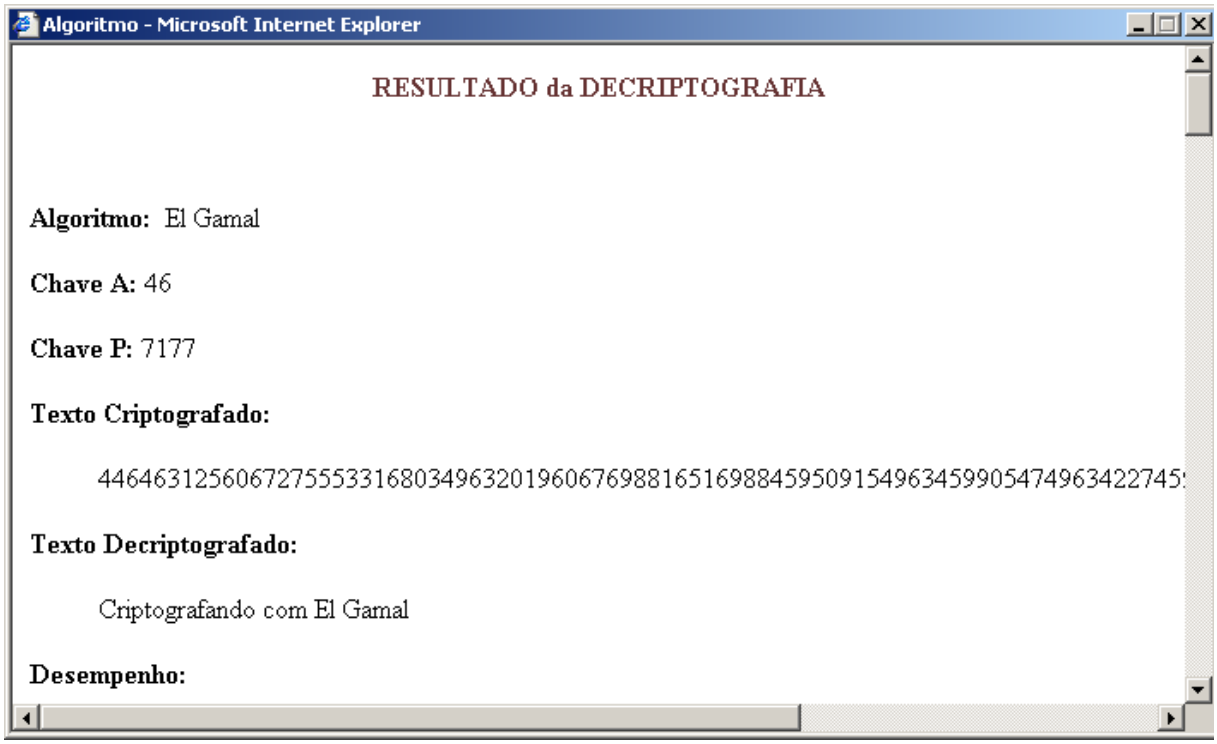


Figura 12.9- Resultado da decifragem

As demais informações do Pop-Up são apresentadas abaixo:

Desempenho:

O texto criptografado acima demorou 2.4201 segundos para ser decriptografado.

Segurança

A forma mais perigosa de ataque contra o sistema El Gamal é resolver o logaritmo discreto que revele a chave A. Qual o número estimado de operações para resolver esse problema de logaritmo discreto? Levando-se em conta que a complexidade da operação é Lp(1/

3,c) para uma constante c (< (64/9)1/3), o tempo para quebrar o El Gamal não é polinomial, mas exponencial!

Referência: Applied Cryptography, Second Edition: Protocols, Algorthms, and Source Code in C, by Bruce Schneier

Detalhes da cifragem:

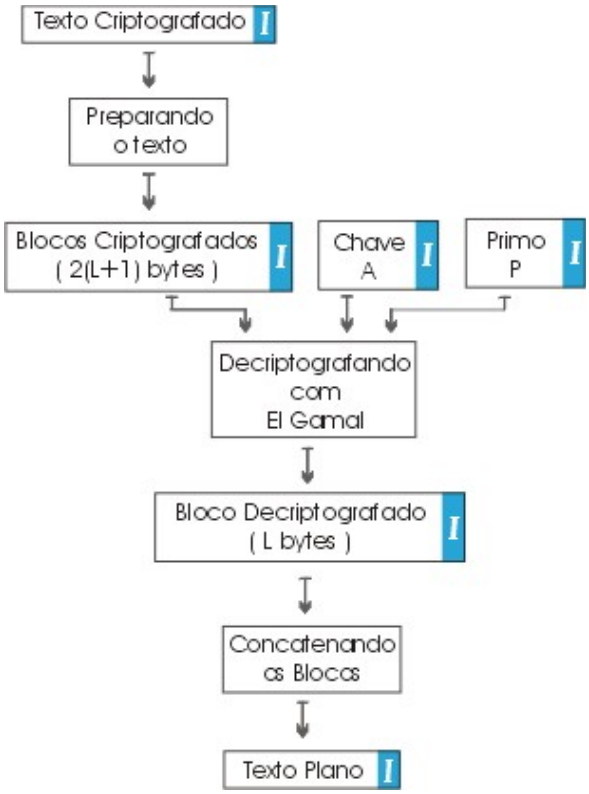


Figura 12.10- Diagrama da decifragem

Texto Criptografado:

44646312560672755533168034963201960676988165169884595091549634599054749634227459938613859459945976988422769883859

voltar ao diagrama

Blocos criptografados:

3125
6067
2755

5331
6803
4963
2019
6067
6988
1651
6988
4595
0915
4963
4599
0547
4963
4227
4599
3861
3859
4599
4597
6988
4227
6988
3859

voltar ao diagrama

Chave A:

46

voltar ao diagrama

Chave P:

7177

voltar ao diagrama

Blocos Decriptografados:

67
114
105
112
116
111

103
114
97
102
97
110
100
111
32
99
111
109
32
69
108
32
71
97
109
97
108

voltar ao diagrama

Texto Decriptografado:

Criptografando com El Gamal

voltar ao diagrama

13 Conclusão

13.1 Estrutura do WebSite

Home: Página de boas vindas do laboratório

Índice: Mapa do site

Documentação: O usuário encontra a documentação explicando um pouco da teoria e dos conceitos de criptografia bem como tem acesso à história e características de cada algoritmo.

Navegando nos algoritmos: Esta seção permite ao usuário navegar através da estrutura do algoritmo. Aqui cada algoritmo está apresentado através de diagrama em blocos. Clicando sobre cada bloco o usuário pode obter a descrição dos procedimentos, a definição de uma variável, ler o código fonte do programa em Matlab ou mesmo entrar no bloco para ver a sua estrutura interna.

Laboratório Virtual: Esta seção permite ao usuário experimetar a criptografia com cada algoritmo, ele pode escolher um algoritmo, digitar o texto e ver como ele fica criptografado, o caminho inverso também é permitido. Além do texto criptografado ou decriptografado, o usuário também tem acesso a alguns detalhes, como por exemplo as subchaves geradas no processo, como o texto foi dividido antes de criptografar, etc.

Downloads: Aqui estão disponibilizados todos os códigos do Matlab. O usuário pode baixá-los para estudar mais profundamente seu funcionamento e executá-lo em seu próprio computador. O código interpretado permite que o usuário entenda facilmente sua estrutura e consiga alterá-lo da forma que quiser para fazer ses próprios testes e chegar as suas próprias conclusões

Links e bibliografia: Referênicas utilizadas para a criação do trabalho.

Glossário: Uma lista de definições de termos utilizados em criptografia.

13.2 Conclusão

A proposta deste projeto foi a de combinar a facilidade do entendimento e alteração dos códigos interpretados gerados para o Matlab com a estrutura web para proporcionar um local com material para estudo dos algoritmos de criptografia DES, Triplo-DES, Blowfish, IDEA, RC5, RSA e El Gamal.

Para tanto foi construído um web site que possibilita aos internautas experimentarem a criptografia, interagindo como site, e fazerem o download dos arquivos para posterior execução no Matlab. Os arquivos contém as implementações dos algoritmos estudados. Ou seja, a toolbox de criptografia.

O fato da linguagem do Matlab ser interpretada possibilita ao estudante (após o download do código) não só a execução da toolbox, mas também a sua alteração com relativa facilidade. A partir de algumas alterações no arquivo .m, pode-se ter acesso por exemplo, às variáveis internas das funções.

O site disponibiliza também a documentação de cada algoritmo, explicando seu modo de funcionamento, suas características e um pouco de sua história

Paralelo a isto pode-se navegar através do diagrama em blocos dos algoritmos permitindo uma visão gráfica do fluxo de informações e de como a informação inicial é tratada e alterada por cada algoritmo antes de gerar o produto final.

Bibliografia

[APPLIED] Applied Cryptography: Protocols, Algorthms, and Source Code in C
Second Edition
by Bruce Schneier
Wiley Computer Publishing, John Wiley & Sons, Inc.
ISBN: 0471128457 Pub Date: 01/01/96

[BASIC] Basic Cryptanalysis, Field Manual 34-40-2
by Department of the Army Publicaiton Staff, Department of the Army
ISBN: 089412272X Pub Date: 03/97

[CITESEER] Paper sobre o algoritmo RC5
http://citeseer.nj.nec.com/rivest95rc.html

[COUNTERPANE] Paper original do Blowfish
http://www.counterpane.com/bfsverlag.html

[CRYPTOGRAPHY] Cryptography: Theory and Practice, Second Edition
by Douglas Stinson (1st edition)
ISBN: 0849385210 Pub Date: 03/95

[HANDBOOK] Handbook of Applied Cryptography
by Alfred J. Menezes, Paul C. Van Oorschot, Scott A. Vanstone
ISBN: 0849385237 Pub Date: 16/10/96

[KOC] Oregon State University
Çetin Kaya Koç
http://islab.oregonstate.edu/koc/ece679/notes/rsa1.pdf

A Modos de Operação

A.1 Introdução

Cifras em bloco são usadas para proteger dados em certos padrões chamados Modos de Operação.

Depois que o NIST introduziu o DES em 1977, ele produziu FIPS 81 (Federal Information Processing Standard), Modos de operação do DES, o qual define quatro modos para o algoritmo DES: Eletronic codebook (ECB), Cipher block chaining (CBC), cipher feedback (CFB) e o output feedback (OFB).

Esses quatro modos, informalmente conhecidos como os modos NiST, tem sido amplamente adotados pelos usuários de criptografia pelo mundo.

Apesar destes modos serem descritos no FIPS 81 para blocos de 64 bits e chaves de 56 bits do DES, eles podem ser extendidos para outros tamanhos de blocos e chaves.

A.2 Descrição

A.2.1 Modo ECB

Este modo apenas encripta o texto plano na forma do texto cifrado com o algoritmo escolhido e é a operação mais básica de todos os modos.

Quando o ECB encripta dados, para um mesmo texto plano de entrada teremos o mesmo texto cifrado de saída, assim padrões na entrada são revelados na saída.

A.2.1.1 Esquema gráfico do funcionamento na Encriptação

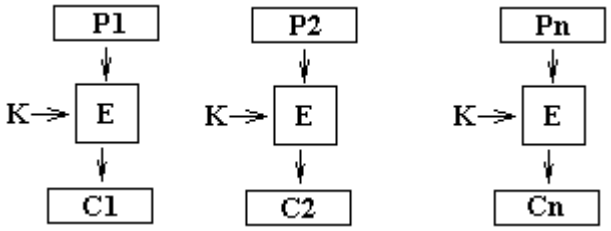


Figura A.1-Esquema gráfico da encriptação utilizando o modo de operação ECB

A.2.1.2 Esquema gráfico do funcionamento na Decriptação

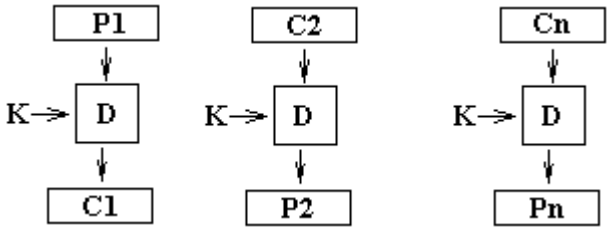


Figura A.2-Esquema gráfico da decriptação utilizando o modo de operação ECB

A.2.2 Modo CBC

É o modo no qual o bloco i de texto cifrado entra num XOR como bloco i+1 de texto plano antes de ser encriptado. Ao primeiro bloco faz-se um XOR com um vetor de inicialização (IV) que não precisa ser secreto.

O modo CBC esconde padrões de dados no texto cifrado e é mais seguro que o ECB.

O problema como modo CBC é que é estritamente serial, ou seja o bloco i deve estar criptografado antes de fazer a criptografia do bloco i+1. Isto limita a possibilidade de operações em paralelo. Um único bit errado na comunicação usualmente resulta em dois blocos inteiros corrompidos no texto plano.

A.2.2.1 Esquema gráfico do funcionamento na Encriptação:

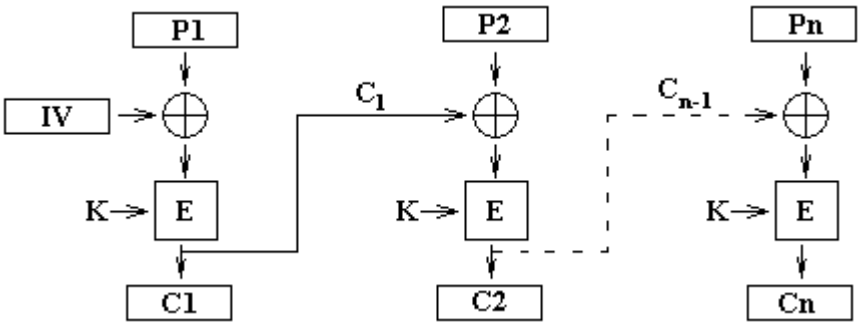


Figura A.3-Esquema gráfico da encriptação utilizando o modo de operação CBC

A.2.2.2 Esquema gráfico do funcionamento na Decriptação

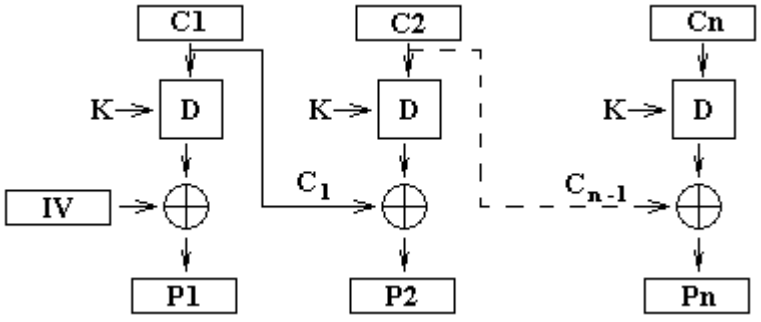


Figura A.4-Esquema gráfico da decriptação utilizando o modo de operação CBC

A.2.3 Modo OFB:

Nesse modo, a saída do algoritmo de encriptação é colocada novamente como entrada da encriptação e usada como entrada num XOR junto com os dados para gerar o bloco cifrado.

Erros na transmissão do texto cifrado não são expandidos no texto plano, mas se bits foresm perdidos ou inseridos no texto cifrado, a sincronização da cifragem é perdida.

A.2.2.1 Esquema gráfico do funcionamento na Encriptação

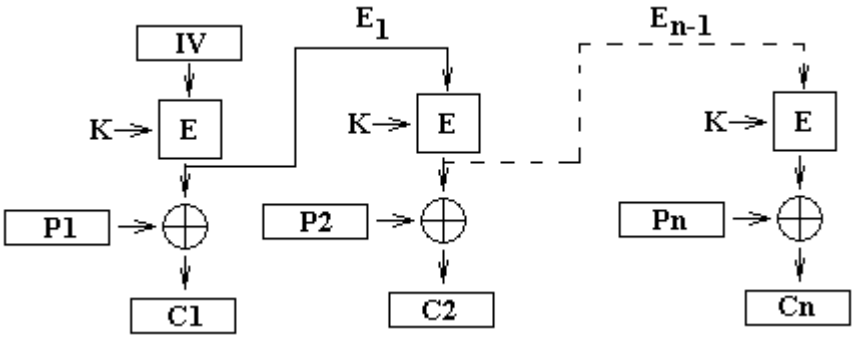


Figura A.5-Esquema gráfico da encriptação utilizando o modo de operação OFB

A.2.2.2 Esquema gráfico do funcionamento na Decriptação

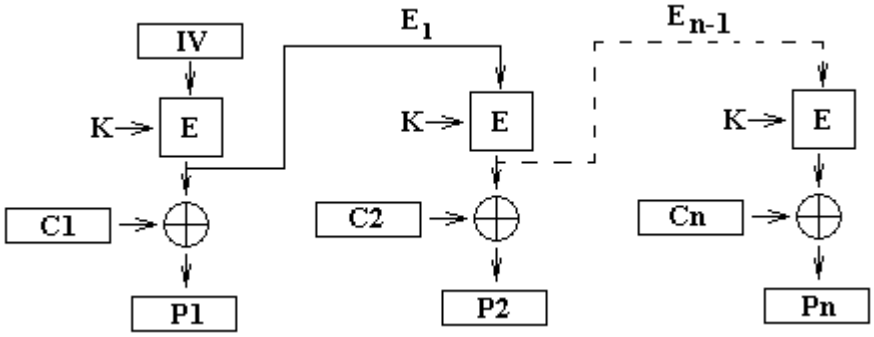


Figura A.6-Esquema gráfico da decriptação utilizando o modo de operação OFB

A.2.4 Modo CFB:

Aqui, o texto cifrado é encriptado e entra num XOR como se fosse uma chave juntamente com o texto plano. e o bloco criptografado é utilizado como entrada do algoritmo de criptografia para codificar o próximo bloco de texto plano (através do XOR).

Erros na transmissão de textos cifrados com o CFB são expandidos na saída do texto plano.

Em aguns casos essa expansão pode ser uma característica de segurança, pois evita que um adversário inverta seletivamente apenas um bit do texto plano.

A.2.2.1 Esquema gráfico do funcionamento na Encriptação

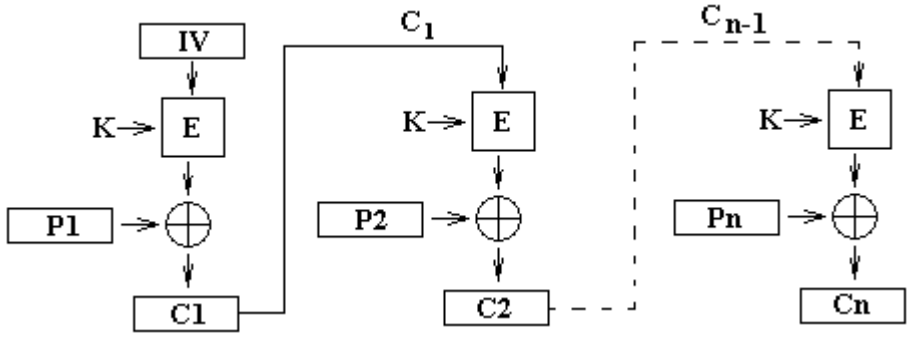


Figura A.7-Esquema gráfico da encriptação utilizando o modo de operação CFB

A.2.2.2 Esquema gráfico do funcionamento na Decriptação

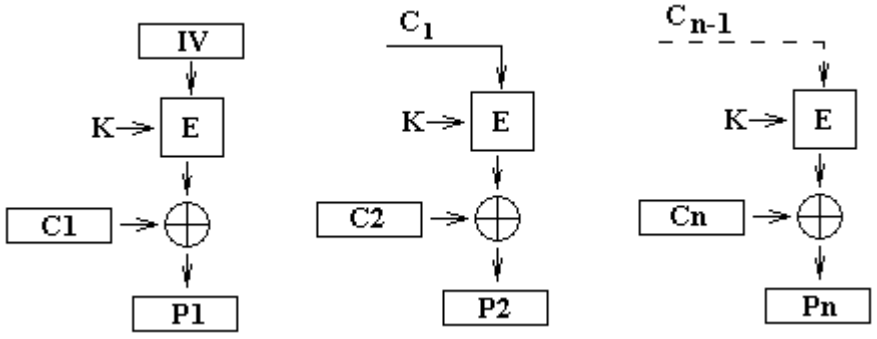


Figura A.8-Esquema gráfico da decriptação utilizando o modo de operação CFB

B História da criptografia

Cerca de 1900 a.C.

A história acontece numa vila egípcia perto do rio Nilo chamada Menet Khufu. Khnumhotep II era um arquiteto do faraó Amenemhet II. Ele construíu alguns monumentos para o faraó, os quais precisavam ser documentados. Nem é preciso dizer que estas informações, escritas em tabletes de argila, não eram para cair no domínio público.

O escriba de **Khnumhotep II** teve a idéia de substituir algumas palavras ou trechos de texto destes tabletes. Caso o documento fosse roubado, o ladrão não encontraria o caminho que o levaria ao tesouro - morreria de fome, perdido nas catacumbas da pirâmide.
Pode ser considerado o primeiro exemplo documentado da escrita cifrada.

1500 a.C.

A criptografia da Mesopotâmia ultrapassou a egípcia, chegando a um nível bastante moderno. O primeiro registro do uso da criptografia nesta região está numa fórmula para fazer esmaltes para cerâmica. O tablete que contém a fórmula tem apenas cerca de 8 cm x 5 cm e foi achado às margens do rio Tigre. Usava símbolos especiais que podem ter vários significados diferentes.

Nesta época, mercadores assírios usavam "intaglios", que são peças planas de pedra com símbolos entalhados para a sua identificação. O moderno comércio com assinaturas digitais estava inventado!

Esta também foi a época em que culturas como a do Egito, China, Índia e da Mesopotâmia desenvolveram a esteganografia:

* Tatuagens com mensagens na cabeça de escravos. Infelizmente era preciso esperar o cabelo crescer. A decifração era feita no barbeiro...

* Marcas na madeira de placas de cera. As marcas eram escondidas com cera nova. Para decifrar, bastava derreter a cera.

* Mensagens dentro do estômago de animais de caça.

Figura B.1-Marcas em madeira escondidas com cera nova

600 a 500 a.C.

Escribas hebreus, escrevendo o livro de Jeremias, usaram a cifra de substituição simples pelo alfabeto reverso conhecida como ATBASH. As cifras mais conhecidas da época são o ATBASH, o ALBAM e o ATBAH, as chamadas cifras hebraicas.

487 a.C.

Tucídides conta sobre ordens entregues ao príncipe e general espartano **Pasanius** em 475 a.C. através do que poderia ser o sistema de criptografia militar mais antigo, o scytale ou bastão de Licurgo.

Figura B.2-Scytale ou bastão de Licurgo

Como um dispositivo para esconder mensagens, o scytale consiste num bastão de madeira ao redor do qual se enrola firmemente uma tira de couro ou pergaminho, longa e estreita. Escreve-se a mensagem no sentido do comprimento do bastão, a tira é desenrolada e contém a mensagem cifrada.

Textos gregos antigos de **Enéas**, o Tático de **Políbio** e outros descrevem outros métodos de ocultar mensagens, mas nenhum deles parece ter sido efetivamente utilizado.

300 a.C.

Artha-sastra, um livro atribuído a **Kautilya**, foi escrito na Índia. Refere diversas cifras criptográficas e recomenda uma variedade de métodos de criptoanálise (o processo de quebrar códigos) para obter relatórios de espionagem. Os processos são recomendados para diplomatas.

Euclides de Alexandria foi um matemático grego que viveu aproximadamente de 330 a.C. a 270 a.C. Euclides compilou e sistematizou a geometria e a teoria dos números da sua época no famoso texto "Elementos".

A associação dos Elementos com a geometria é tão frequente que muitas vezes se esquece três dos treze livros. Os volumes VII, VIII e IX tratam apenas da teoria dos números. Nestes três livros Euclides define os números primos, desenvolve várias propriedades da divisibilidade, apresenta seu algoritmo para encontrar o máximo divisor comum de dois inteiros, mostra como encontrar um número perfeito par de um número primo (hoje denominado de Mersenne), prova que existe um número infinito de números primos e define uma versão do teorema fundamental da aritmética.

Figura B.3- Manuscrito dos Elementos

D'Orville 301, escrito no ano 888 (O scholium é do Vaticano)

Erastótenes de Cirene, filósofo e geômetra grego, viveu de 276 a.C. a 194 a.C. Conhecido pelo como criador de um método para identificar números primos, o crivo de Erastótenes, e por ter calculado o diâmetro da Terra com surpreendente precisão.

204 a 122 a.C.

O Código de Políbio - um telégrafo ótico (substituição poligrâmica)

±150 a.C.

Políbio, um historiador grego nascido em Megalópolis e que viveu de 204 a.C. a 122 a.C., escreveu vários livros sobre o Império Romano. A ele é atribuída a uma cifra de substituição que converte os caracteres da mensagem em cifras - o código de Políbio. Infelizmente Políbio não relatou qualquer utilização do seu sistema.

130 a.C.

Em Uruk, atualmente conhecido como Iraque, era comum os escribas transformarem seus nomes em números dentro do emblema dos seus trabalhos. A prática, provavelmente, era apenas para divertir os leitores e não estava relacionada à segurança.

50 a.C.

Júlio César usou sua famosa cifra de substituição para encriptar comunicações governamentais. Para compor seu texto cifrado, César alterou letras desviando-as em três posições; A se tornava D, B se tornava E, etc. Às vezes, César reforçava sua encriptação substituindo letras latinas por gregas.

O código de César é o único da antiguidade que é usado até hoje, apesar de representar um retrocesso em relação à criptografia existente na época. Atualmente denomina-se qualquer cifra baseada na substituição cíclica do alfabeto de código de César.

Figura B.4- O código de César

79 d.C.

Figura B.5-Coluna com o inscrito da fórmula Sator

A fórmula Sator ou quadrado latino é encontrado em escavações feitas em Pompéia, inscrito numa coluna. Ocorre também num amuleto de bronze, originário da Ásia Menor, datado do século V. As palavras rotas orepa tenet apero sator parecem ter o efeito mágico de nunca desaparecem... persistem até hoje como enigma de transposição.

```
R O T A S
O R E P A
T E N E T
A P E R O
S A T O R
```

Figura B.6-Fórmula de Sator

200 d.C.

O Papiro de **Leiden**, um trabalho detalhando como fazer poções especiais, possui texto encriptado nos trechos cruciais das receitas. Exemplos destas "receitas mágicas" são das que supostamente fazem com que um homem ame uma mulher ou que provoquem uma doença de pele incurável. Só para informar, as receitas não funcionam.

400 d.C.

Kama-Sutra, escrito por **Vatsayana**, classifica a criptografia como a 44ª e 45ª das 64 artes que as pessoas deveriam conhecer e praticar:

* A arte de saber escrever em cifras e de escrever palavras de uma forma peculiar.
* A arte de falar mudando as formas da palavra.

476 a 1453

Desde os idos do fim do Império Romano até perto da época do descobrimento do Brasil. Esta é a chamada Idade Média. Na Europa, entretanto, o período inicial desta época também foi chamado de "período das trevas", e a criptologia não escapou desta "recessão". Muito do conhecimento sobre o assunto foi perdido porque era considerado magia negra ou bruxaria.

Nesta época, a contribuição árabe-islâmica foi significativa, principalmente com a invenção da criptanálise para a substituição monoalfabética. A denominação "Cifra", "Chiffre", "Ziffer", etc, como também "zero", utilizado em muitas línguas, vem da palavra árabe "sifr", que significa "nulo".

A Itália foi a primeira a acordar, iniciando o movimento renascentista ao redor de 1300, sendo responsável pelos primeiros grandes avanços. Veneza criou uma organização especializada em 1452, cujo único objetivo era lidar com a criptologia. Eles possuíam três secretarias que solucionavam e criavam cifras que eram usadas pelo governo.

"Alguns documentos com textos cifrados do governo de **Ghaznavid** na Pérsia conquistada sobrevivem e um cronista relata que altos oficiais recebiam cifras pessoais antes de

serem enviados para ocupar novos postos. Mas a falta de continuidade dos estados islâmicos e a consequente falha em desenvolver um serviço civil e em criar embaixadas permanentes em outros países acabou por restringir o uso mais difundido da criptografia."

718-786

al-Khalil, cujo nome completo era **Abu Abd al-Rahman al-Khalil ibn Ahmad ibn Amr ibn Tammam al Farahidi al-Zadi al Yahmadi**, escreveu o livro Kitab al Mu'amma (O livro das mensagens criptográficas) sobre criptografia, em grego, para o imperador bizantino. Este livro infelizmente foi perdido. Ele decifrou um criptograma bizantino antigo. Sua solução baseava-se no início do texto original, que ele supôs corretamente como sendo "Em nome de Deus" - coisa comum na época. Este método criptanalítico tornou-se padrão, tendo sido usado até na decifração de mensagens Enigma durante a Segunda Guerra Mundial. É conhecido como o método da palavra provável.

801-873

al-Kindi, cujo nome completo era **Abu Yusuf Yaqub ibn Is-haq ibn as Sabbah ibn 'omran ibn Ismail Al-Kindi**, escreveu Risalah fi Istikhraj al Mu'amma (Escritos sobre a decifração de mensagens criptográficas). Este livro está conservado, sendo o mais antigo sobre Criptologia. Nele, o autor faz análises de frequência, portanto, pode-se considerar Al-Kindi como o bisavô da Matemática Estatística.

Figura B.7-Abu Yusuf Yaqub ibn Is-haq ibn as Sabbah ibn 'omran ibn Ismail Al-Kindi

855

Abu Bakr Ahmad ben Ali ben Wahshiyya an-Nabati publicou vários alfabetos de cifras, os quais eram tradicionalmente usados para mágicas.

1119-1311

Templários O Templo era uma ordem de monges combatentes fundada em 1119 pelos cavaleiros Ugo dei Pagani e Geoffrey de Saint-Omer para proteger os peregrinos na Terra Santa. Logo após a sua fundação em Jerusalém, **Balduíno II**, imperador de Constantinopla, concedeu-lhes um palácio nas proximidades do templo de Salomão, donde se originou o nome da ordem.

A ordem enriqueceu rapidamente graças a numerosas doações e se tornou uma organização internacional que, por muito tempo, teve uma influência notável, rivalizando com a do rei da França e a do próprio Papa. Em 1291 os templários foram obrigados a abandonar a Terra Santa, fugindo para a ilha de Chipre.

A organização cifrava suas letras de crédito utilizando um método próprio.

Em 1311, a ordem dos templários foi dissolvida por **Filipe, o belo**. Em sérias dificuldades financeiras, Filipe mandou prender e torturar os templários, fazendo com que lhe entregassem suas riquezas. Um ano mais tarde, em 1312, um decreto do Concílio de Viena aboliu a ordem.

1187-1229

Ibn DUNAINIR, **Ibrahim ibn Mohammad ibn Dunainir**, é autor do livro redescoberto em 1987 Maqasid al-Fusul al-Mutarjamah an Hall at-Tarjamah (Explicações claras para a solução de mensagens secretas). O livro contém uma inovação importante: cifras algébricas (substituição de letras por números e transformá-los aritmeticamente).

1187-1268

Ibn ADLAN, **Afif ad-Din ibn Adlan ibn Hammad ibn Ali al-Mousili an-Nahwi al-Mutarjim**, é autor do livro redescoberto em 1987 Al-Mu'allaf lil-Malik al-Ashraf (Escrito para o **Rei al-Ashraf**) que são explicações detalhadas de criptoanálise.

1226

Em 1226, uma criptografia política discreta apareceu nos arquivos de Veneza, onde "pontos e cruzes substituíam as vogais em algumas palavras esparsas".

±1250

O monge inglês Roger Bacon descreveu sete métodos de cifras e escreveu: "Um homem é louco se escrever um segredo de qualquer outra forma que não seja a de o dissimular do vulgar."

Figura B.8-Roger Bacon

1300

`Abd al-Rahman Ibn Khaldun escreveu o Muqaddimah, um importante relato da história que cita o uso de "nomes de perfumes, frutas, pássaros ou flores para indicar letras, ou [...] sobre formas diferentes das formas das letras aceitas" como uma código usado entre escritórios de impostos e militares. Ele também inclui uma referência à criptanálise, observando que "escritos conhecidos sobre o assunto estão em poder do povo".

1312-1361

Ibn AD-DURAIHIM, cujo nome completo era **Taj ad-Din Ali ibn Muhammad ibn Abdul'aziz ibn ad-Duraihim**, é autor do livro redescoberto em 1987 Miftah al-Kunuz fi Idah al-Marmuz (Chaves para a elucidação de mensagens secretas) contendo uma classificação de cifras, análise de frequências em várias línguas, uma tabela de Trithemius(Vigenère) e Grades.

1379

Em 1378, depois do Cisma de Avignon, o antipapa **Clemente VII** decide unificar o sistema de cifras da Itália Setentrional e designou tal tarefa a **Gabriele Lavinde**. Lavinde compilou uma coleção de cifras num manual, do qual o Vaticano conserva uma cópia de 1379.

Com seu alfabeto de substituição combinada (código/cifra) ele uniu a cifra de substituição com um código de listas de palavras, sílabas e nomes equivalentes. Este sistema foi amplamente utilizado por diplomatas e alguns civis europeus e americanos por mais de 450 anos.

1392

Geoffrey Chaucer, considerado o melhor poeta inglês antes de Shakespeare, no seu "The Equatorie of the Planetis", um suplemento do seu "Treatise on the Astrolabe", incluíu seis passagens escritas em cifras. O sistema de cifras consiste num alfabeto de símbolos de substituição.

Figura B.9-Geoffrey Chaucer

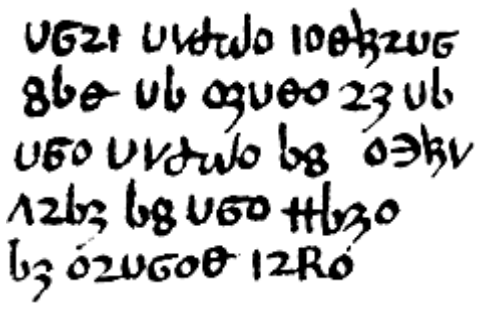


Figura B.10- Passagens cifradas em "The Equatorie of the Planetis"

A solução do criptograma mostrado acima é: "This table servith for to entre in to the table of equacion of the mone on either side."

1401

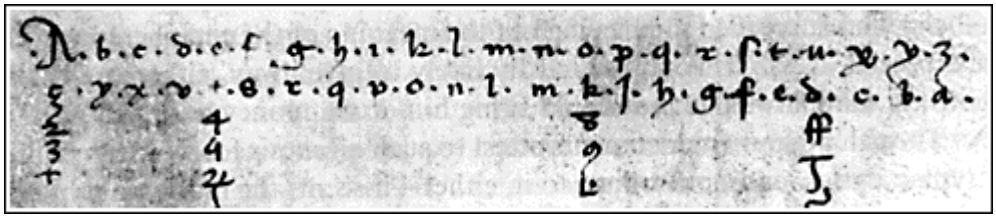


Figura B.11-Cifra homofônica criada por Simeone de Crema

Simeone de Crema usou uma chave na qual cada vogal do texto original possuía vários equivalentes. Isto comprova silenciosamente que, nesta época, o ocidente conhecia a criptanálise. Não pode haver outra explicação para o aparecimento destes múltiplos substitutos ou homófonos. O fato dos homófonos serem aplicados a vogais, e não apenas indiscriminadamente, indica no mínimo o conhecimento do esboço de uma análise de frequência.

O uso da análise de frequência pode levar a uma solução rápida de cifras de substituição monoalfabéticas simples. Acima, a chave da substituição homofonética de Crema.

1412

Shihab al-Din abu `l-`Abbas Ahmad ben `Ali ben Ahmad `Abd Allah al-Qalqashandi (1355-1418) em 1412 escreveu a Subh al-a `sha, uma enciclopédia de 14 volumes em Árabe, na qual incluíu uma seção de Criptologia. Esta informação foi atribuída a **Taj ad-Din `Ali ibn ad-Duraihim ben Muhammad ath-Tha`alibi al-Mausili**, que viveu de 1312 a 1361, cujos escritos sobre Criptologia foram perdidos. A lista de cifras nesta obra inclui tanto a substituição quanto a transposição e, pela primeira vez, uma cifra com múltiplas substituições para cada letra do texto original. Também é atribuída a **Ibn al-Duraihim** uma explicação com exemplo de criptanálise, inclusive o uso de tabelas de frequência de letras e conjuntos de letras que podem ocorrer juntas numa palavra.

1453 a 1789

Após o "período das trevas", a criptologia começou a progredir. Todos os governos da Europa Ocidental usaram a criptografia de uma ou de outra forma e a codificação começou a tornar-se mais popular. Cifras eram comumente utilizadas para manter contatos com embaixadores.

1466

Leon Battista Alberti era amigo de **Leonardo Dato**, um secretário pontificial o qual, provavelmente, introduziu Alberti na criptologia. Alberti inventou e publicou a primeira cifra polialfabética, criando um disco de cifragem (conhecido atualmente como "Captain Midnight Decoder Badge") para simplicar o processo. Ao que tudo indica, esta classe de cifra não foi quebrada até os anos de 1800. Alberti também tem muitos escritos sobre o estado da arte em cifras, além da sua própria invenção. Também fez uso do seu disco para obter código cifrado. Estes sistemas eram muito mais fortes que a nomenclatura usada por diplomatas da época e continuaram sendo por muitos séculos mais.

Figura B.12-"Captain Midnight Decoder Badge"

O "Trattati in cifra" de **Leone Battista Alberti** foi publicado em Roma em 1470. Se referia "especialmente a teorias e processos de cifragem, métodos de decifração e dados estatísticos".

1473-1490

"Um manuscrito [...] de **Arnaldus de Bruxella** usa cinco linhas de cifras para ocultar a parte crucial da operação que fazia a pedra filosofal."

1474

Em 1474, **Sicco Simonetta** publicou "Regulae ad extrahendum litteras zifferatas sine exemplo", um pequeno trabalho ressaltando "métodos de decifração e fornececendo dados estatísticos consideráveis". "A data do pequeno ensaio de Simonetta sobre cifras é importante porque se tratava de um período no qual a criptologia se tornou prática universal, quando cifras simples evoluíram para criptogramas complicados."

1518

Figura B.13-Johannes Trithemius

Johannes von Heydenberg aus Trittenheim/Mosel (1462-1516), ou **Johannes Trithemius**, escreveu o primeiro livro impresso de criptologia. Ele inventou uma cifra esteganográfica na qual cada letra era representada como uma palavra obtida de uma sucessão de colunas. A série de palavras resultantes seria uma oração legítima. Também descreveu cifras polialfabéticas na forma de tabelas de substituição retangulares que, na época, já tinham se tornado padrão. Introduziu a noção da troca de alfabetos a cada letra.

Johannes Trithemius escreveu, porém não publicou, sua Steganographia, a qual circulou como manuscrito por mais de cem anos, sendo copiada por muitas pessoas que desejavam extrair os segredos que se pensava que continha. A verdadeira história do feiticeiro que conjurava espíritos e praticava magia negra você encontra em "O Segredo do Terceiro Livro". Altamente interessante, recomendo a leitura.

A Polygraphiae libri sex de Trithemius, a qual incluía sua tabela de substituição tabula recta Caesar, foi publicada em 1518, apesar de haver dúvidas quanto à data correta. Foi

reimpressa em 1550, 1564, 1571 e 1600. Uma tradução em Francês apareceu em 1561 e em 1564.

1526

O Opus novum ... principibus maxime vtilissimum pro cipharis de **Jacopo Silvestri** é impresso. A obra discute seis métodos de cifras, inclusive a cifra de César, para a qual ele recomendava o uso de um disco de cifragem. Opum novum foi escrito para ser um manual prático de criptologia que "claramente pretendia alcançar um vasto círculo de leitores".

Figura B.14-Alfabeto chave de Silvestri

Na figura acima, observe que o alfabeto-chave de Silvestri não possuía as letras j, v, w e y. "No disco, as três marcas que sucedem o Z representam: & para et; um símbolo usado comumente no Latim medieval para significar us ou um no final de palavras (p.ex., plurib9 = pluribus), ou com, con, cum ou cun no início de palavras (p.ex., 9cedo = concedo); e um símbolo usado para rum, a terminação do genitivo plural latino (illo# = illorum). O zig-zag no centro da figura deve corresponder a uma pequena manivela para girar os discos móveis".

1533

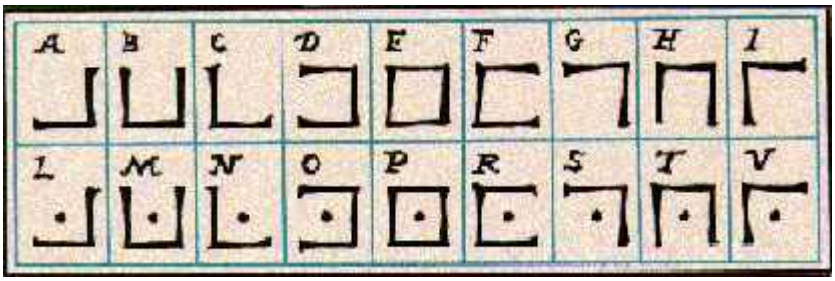


Figura B.15-Cifra Pig Pen

Heinrich Cornelius Agrippa von Nettelsheim publica o De occulta philosophia, em Colônia, na Alemanha. No livro 3, capítulo 30, descreve sua cifra de substituição monoalfabética, hoje conhecida como Cifra Pig Pen. A tradução literal do nome é Porco no Chiqueiro e vem do fato de que cada uma das letras (os porcos) é colocada numa "casa" (o chiqueiro).

Na época, a cifra parece ter sido de importância pois, alguns anos mais tarde, Vigenère a reproduz no seu Traicté des chiffres, ou secretes manieres d'escrire (Paris, 1586, f. 275 v). Aparentemente, esta cifra foi utilizada pela sociedade secreta dos franco-maçons.

1540

Giovanni Battista Palatino publicou seu Libro nvova d'imparare a scrivere ... Con vn breue et vtile trattato de le cifere. Foi reimpresso em 1545, 47, 48, 50, 53, 56, 61, 66, 78 e 1588. Uma versão revisada foi impressa em 1566, 78 e 88.

1550

Foi publicado o De subtilitate libri XXI de **Girolamo Cardano**. "Esta obra famosa, de um notável matemático, físico e filósofo contém... uma quantidade considerável de informações a respeito de processos de cifragem".

Figura B.16-Girolamo Cardano

Foi reimpressa em 1551, duas vezes em 54, em 59, outras duas vezes em 60, e em 80 e 82. Uma tradução francesa foi impressa em 1556.

Cardano inventou o primeiro procedimento com auto-chave, mas seu sistema era imperfeito. A grelha de Cardano consiste numa folha de material rígido onde se encontram, em intervalos ireegulares, pequenas aberturas retangulares da altura de uma linha de escrita e de comprimento variável. O remetente escreve o texto nas aberturas, depois retira a folha e completa os espaços vazios com letras quaisquer. O destinatário põe a mesma grelha sobre o texto cifrado para ler a mensagem.

Em 1556, Cardano publica De rerum varietate libri XVII, o qual continha informações criptográficas e era a continuação do seu popular De Subtilitate. Ambos os livros foram "traduzidos e pirateados por editores por toda a Europa". De rerum foi reimpresso em 1557, 58, 80 e 81.

1553

Giovanni Battista Bellaso, secretário do **cardeal Duranti** e do **cardeal Rodolfo Pio**, introduziu a noção do uso de uma senha como chave para uma cifra polialfabética.

La cifra del Sig. Giovan Battista Bellaso foi publicado em 1553, depois corrigido e reimpresso em 1557 e 1564.

Em 1564, Bellaso publicou uma cifra de auto-chave, melhorando o trabalho de Cardano, o qual parece ter sido o autor da idéia.

1556

A Espanha, sob a regência de **Filipe II**, adotou as mais modernas nomenclatura com homófonos (2 símbolos para as consoantes e 3 para as vogais) e listas para a substituição dos di- e trígrafos mais usados. Associados à nomenclatura e às listas, utilizaram códigos. O sistema foi utilizado até o século XVII e, a cada 3 a 5 anos, as códigos eram modificados.

1558

Philibert Babou, embaixador em Roma do **Rei Henrique II**, usa a substituição homofônica na correspondência oficial. Pode-se dizer que a cifra de Babou era uma substituição homofônica simplificada.

1563

O Magiae natvralis libri XX de **Giambattista Della Porta**, que no Livro XVI trata de decifração, foi publicado em 1558. Foi reimpresso em 1560, duas vezes em 61, em 62, 64, 67, 76, 85, 91, 97 e 1607. Uma tradução francesa anônima foi impressa em 1565, 67, 70, 71 e 84.

Figura B.17-Giambattista Della Porta

Em 1563, Della Porta escreveu um texto sobre cifras introduzindo a cifra digrâmica (ou digráfica). Ele classificou as cifras em cifras de transposição, de substituição e de substituição por símbolos (uso de alfabetos estranhos). Sugeriu o uso de sinônimos e erros ortográficos para confundir os criptanalistas. Aparentemente introduziu a noção de alfabeto misto numa tabela polialfabética.Foi publicado De fvrtivis literarvm notis, vvlgo de ziferis Libri IIII, do mesmo autor. No mesmo ano apareceu traduzido em Inglês sob o título de On secret notations for letters, commonly called chiphers. Seus quatro livros, tratando respectivamente de cifras arcaicas, cifras modernas, criptanálise e uma lista de peculiaridades linguísticas que ajudavam na solução, compilavam o conhecimento criptológico da época. Um conjunto rococó de discos de cifragem acompanhava os livros. A obra foi reimpressa em 1591, 93, duas vezes em 1602, em 1603 e 1606.

Em 1591, o De fvrtivis de **Della Porta** foi reimpresso por **John Wolfe** em Londres, o qual "aprimorou a edição original de 1563 tornando-a praticamente perfeita".

Em 1593, o De fvrtivis foi reeditado, sem permissão, como De occvltis literarvm notis e incluía o primeiro jogo de tabelas criptológicas sinópticas jamais publicadas. Foi reimpresso em 1603 e 1606.

1585

Blaise de Vigenère escreveu um livro sobre cifras, incluindo os primeiros sistemas autênticos de texto claro e texto cifrado com auto-chave, nos quais letras prévias do texto claro ou cifrado são usadas para a letra chave atual. Ambos foram esquecidos e reinventados no final do século XIX. A idéia da auto-chave sobrevive até os dias de hoje nos modos CBC e CFB do DES.

Figura B.18-Blaise de Vigenère

1586

Blaise de Vigenère publica seu Traicté des chiffres, um tratado de 600 páginas. Nele discute muitas cifras, inclusive o sistema da "auto-chave corrente", usada em algumas máquinas de cifragem modernas, e os assim chamado método "Vigenère tableau". Foi muito escrupuloso, dando o devido crédito para materiais de outros autores, destacando-os clara e inequivocamente.

1591

Matteo Argenti, sobrinho de **Della Porta**, publica um folheto de 135 páginas sobre criptologia. Ele utiliza uma chave mneumônica para misturar o alfabeto secreto onde deixa de lado as letras duplas.

1592

Julius Caesar Scaliger publicou seu Exotericarvm exercitationvm liber XV, de 1220 páginas. "Este tratado filosófico sobre o De subtilitate de Cardano... foi um livro de texto muito popular até a queda final da física de Aristóteles"

Foi reimpresso em 1557, 60 e 76.

Final do século XVI

A França começa a consolidar sua liderança na criptoanálise.

1623

Sir Francis Bacon (que se supõe, com grande probabilidade, ter sido William Shakespeare) é o inventor de um sistema de esteganografia que ele publicou em De dignitate et augmentis scientiarum. Denominou seu alfabeto de biliteral porque utiliza uma combinação das duas letras A e B em grupos de cinco.

Figura B.19-Sir Francis Bacon

A cifra é conhecida pelo seu nome, Cifra de Bacon, hoje em dia classificada como codificação binária de 5 bits.

1691

Antoine Rossignol e seu filho **Bonaventure** elaboraram a Grande Cifra de **Luís XIV**. Ela caiu em desuso após a morte dos seus inventores e suas regras precisas foram rapidamente perdidas. A Grande Cifra era muito robusta, tanto que só foi quebrada no final do século XIX (ao redor de 1890). Alguns autores afirmam foi quebrada por **Bazeries**, enquanto outros citam **Victor Gendron**, um seu contemporâneo.

1734

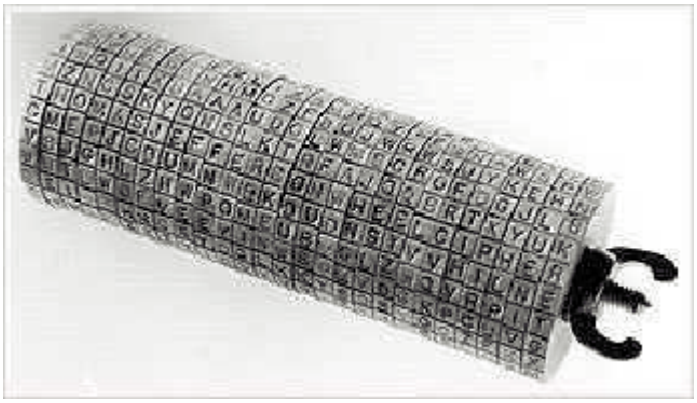
O belga **José de Bronckhorst**, **Conde de Gronsfeld**, era um homem de guerra e também um diplomata. Seu cargo exigia que guardasse certos segredos importantes. Ao redor de 1734 acabou criando seu próprio sistema de cifras: melhorou a Cifra de César introduzindo um deslocamento variável indicado por uma chave numérica. Na verdade, a cifra de Gronsfeld acaba sendo uma variante da Cifra de Vigenère, com uma diferença: a cifra de Vigenère permite até 26 deslocamentos enquanto que a Gronsfeld, atrelada aos dígitos de 0 a 9, tem apenas 10 deslocamentos possíveis.

1790 a 1900

A criptologia dissemina-se no mundo ocidental e tanto o interesse quanto a necessidade parecem ser cada vez maiores. Começam a surgir máquinas e dispositivos mais elaborados, um grande avanço no suporte à criptografia mecanizada. É uma época de grandes invenções e do aparecimento dos primeiros sistemas de comunicação à distância.

Os sistemas de comunicação à distância, por serem sistemas abertos, dão um novo impulso à criptografia. Por um lado, as enormes vantagens de uma comunicação rápida e eficiente; por outro lado, as mensagens ficam muito mais vulneráveis ao meio e também desprotegidas.

1795



Na época em que era secretário de estado de **George Washington**, **Thomas Jefferson**, futuro presidente dos Estados Unidos, criou um método simples, engenhoso e seguro de crifrar e decifrar mensagens: o cilindro cifrante.

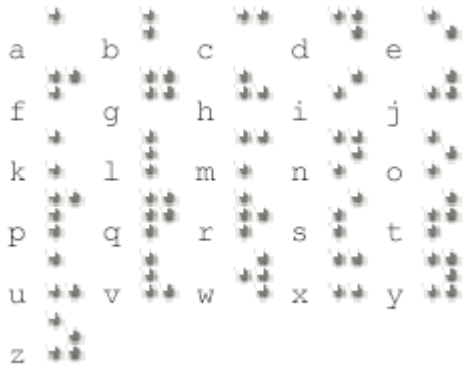
Figura B.20-Cilindro cifrante

Durante a revolução americana, Jefferson confiava cartas importantes a mensageiros que as entregavam pessoalmente porém, quando se tornou ministro americano para a França, os códigos assumiram grande importância na sua correspondência porque os agentes de correio europeus abriam e liam todas as cartas que passavam pelos seus comandos.

Apesar de, aparentemente, Jefferson ter abandonado o uso do cilindro cifrante em 1802, ele foi "re-inventado" um pouco antes da Primeira Guerra Mundial e foi usado pelo exército estadunidense e outros serviços militares.

O cilindro de Jefferson (Jefferson's wheel cipher em Inglês), na sua forma original, é composto por 26 discos de madeira que giram livremente ao redor de um eixo central de metal. As vinte e seis letras do alfabeto são inscritas aleatoriamente na superfície mais externa de cada disco de modo que, cada um deles, possua uma sequência diferente de letras. Girando-se os discos pode-se obter as mensagens.

1834



Louis Braille (1809-1852), educador francês, ficou cego aos 3 anos de idade. Interessou-se por um sistema de escrita, apresentado na escola Charles Barbier, no qual uma mensagem codificada em pontos era cunhada em papel-cartão. Aos 15 anos de idade trabalhou numa adaptação, escrita com um instrumento simples.

Figura B.21-Código Braile

O Código Braille consiste de 63 caracteres, cada um deles constituído por 1 a 6 pontos dispostos numa matriz ou célula de seis posições. Mais tarde adaptou este sistema para a notação musical. Publicou tratados sobre seu sistema em 1829 e 1837. O Sistema Braille é universalmente aceito e utilizado até os dias de hoje.

1840

Samuel Morse (1791-1872) desenvolve o código que recebeu o seu nome. Na verdade não é um código, mas sim um alfabeto cifrado em sons curtos e longos. Morse também foi o inventor de um dispositivo que chamou de telégrafo e, em 1844, enviou sua primeira mensagem com os dizeres "What hath God wrought".

A história de Morse é muito interessante porque, ao contrário do que se espera, a telegrafia deve-se a um artista e não a um cientista. A invenção do telégrafo altera profundamente a criptografia e torna a cifragem uma necessidade absoluta.

1901 a 1998

Os computadores são a expressão maior da era digital, marcando presença em praticamente todas as atividades humanas. Da mesma forma com que revolucionaram a informação, também causaram uma reviravolta na criptologia: por um lado ampliaram seus horizontes, por outro tornaram a criptologia quase que indispensável. Foi apenas há alguns anos que se reconheceu a criptologia como ciência.

1901

Guglielmo Marconi inicia a era da comunicação sem fio. Apesar da vantagem de uma comunicação de longa dist,ncia sem o uso de fios ou cabos, o sistema é aberto e aumenta o desafio da criptologia.

Figura B.22-Guglielmo Marconi

Inicialmente a telegrafia sem fio utilizava apenas o código Morse, acessível a todos que captassem os sinais. Impunha-se a necessidade de codificações que garantissem o sigilo das mensagens.

1913

O capitão **Parket Hitt** reinventou o cilindro cifrante, em forma de fita, abrindo caminho para o M-138-A da Segunda Guerra Mundial.
1916

O major **Joseph O. Mauborgne** passou a cifra de fita de Hitt novamente para a forma de cilindro, fortaleceu a construção alfabética e produziu o dispositivo que se transformaria no M-94.

Figura B.23-Joseph O. Mauborgne

Em 1918 aperfeiçoou a cifra de Vernam: o One-Time-Pad, cuja tradução livre para o Português seria Bloco Descartável.

1917

William Frederick Friedman, o homem que introduziu o termo "criptoanálise" e que posteriormente será chamado de "pai da criptoanálise dos EUA", começa a trabalhar como criptoanalista civil no Riverbank Laboratories, que também presta serviços ao governo dos EUA. Mais tarde Friedman cria uma escola de criptoanálise militar, inicialmente no Riverbank e depois em Washington.

Figura B.24-William Frederick Friedman

Um funcionário da AT&T, **Gilbert Sandford Vernam**, inventa uma máquina de cifragem polialfabética capaz de usar uma chave totalmente randômica e que nunca se repete. Esta máquina foi oferecida ao governo dos EUA para ser usada na Primeira Guerra Mundial, porém foi rejeitada. Foi colocada no mercado comercial em 1920.

Figura B.25-Gilbert Sandford Vernam

Vernam desenvolveu uma única cifra inviolável, baseada na cifra de Vigenère, que leva seu nome. Com o aperfeiçoamento feito por **Mauborgne**, nasce o One-Time-Pad.

1918

Os alemães começam a usar o sistema ADFGVX no final da Primeira Guerra Mundial. Era uma cifra baseada em substituição (através de uma matriz com chave), fracionamento e depois na transposição das letras fracionadas. Foi quebrada pelo criptoanalista francês, tenente **Georges Painvin**.

Arthur Scherbius patenteia uma máquina de cifragem e tenta vendê-la ao exército alemão, mas a máquina é rejeitada.

1919

Hugo Alexander Koch patenteia na Holanda uma máquina cifrante baseada em rotores. Em 1927, passa os direitos de patente para Arthur Scherbius, inventor e distribuidor da máquina Enigma desde 1923.

Arvid Gerhard Damm requer uma patente na Suécia para uma máquina cifrante de rotores mecânicos. Esta máquina, sob a direção de **Boris Caesar Wilhelm Hagelin**, evoluíu para uma família de máquinas cifrantes. À frente dos negócios, **Hagelin** foi o único criptógrafo comercial do seu tempo que teve sucesso no seu empreendimento. Após a guerra, uma lei sueca que permitia ao governo apropriar-se de inventos que considerasse importante serem defendidos, fez com que Hagelin se mudasse para Zug, na Suiça, onde foi incorporado pela Crypto AG. Esta empresa ainda está em atividade apesar de estar envolvida numa controvérsia: alega-se que tenha enfraquecido um produto cifrante para poder vendê-lo para o Irã.

1921

Edward Hugh Hebern funda a Hebern Electric Code, uma empresa produtora de máquinas de cifragem eletro-mecânicas baseadas em rotores que giram, no estilo de odômetros, a cada caracter cifrado.

1923

Arthur Scherbius funda um empreendimento, o Chiffriermaschinen Aktiengesellschaft, para construir e finalmente vender sua máquina Enigma para o exército alemão.

1924

Alexander von Kryha produz sua "máquina codificante" que foi utilizada até os anos 1950, inclusive pelo Corpo Diplomático alemão. Entretanto, era criptograficamente fraca por possuir um limite pequeno. Um criptograma de teste, com 1135 caracteres, foi decifrado em 2 horas e 41 minutos pelos criptoanalistas **Friedman**, **Kullback**, **Rowlett** e **Sinkov**. Mesmo assim, a máquina continuou a ser comercializada e usada, um sucesso de vendas e uma lição para os consumidores de dispositivos criptográficos.

1927 a 1933

O uso da criptografia não estava restrito a banqueiros, amantes ou pesquisadores. Cada vez mais os criminosos, especialmente os contrabandistas, usam a criptografia para os seus propósitos. "A maior era de contrabando internacional criou a maior era de criptografia criminosa". Nestes dias, o FBI instala um escritório de criptoanálise para lidar com o problema.

"Um comandante da Marinha Real, tenente aposentado, inventou os sistemas para as operações no Pacífico dos Consolidated Exporters. Seus grupos do Golfo e do Atlântico, contudo, desenvolveram seus próprios sistemas. Seu nome era desconhecido, porém sua capacidade criptográfica era evidente. Os sistemas dos contrabandistas tornaram-se gradativamente mais complexos."

"Alguns dos sistemas são de uma complexidade nunca antes aplicada por qualquer governo nas suas comunicações mais secretas" escreveu **Elizebeth Smith Friedman** num relatório em meados da década de 1930.

"Em nenhum momento durante a Segunda Guerra Mundial, quando os métodos de comunicação secreta alcançaram seu maior desenvolvimento, foram usadas ramificações tão intrincadas quanto as encontradas em algumas correspondências dos navios que transportavam rum na costa oeste".

Elizabeth Smith Friedman decifra os códigos dos contrabandistas de rum na vigência da lei seca.

1929

Lester S. Hill publica seu livro *Cryptography in an Algebraic Alphabet*, no qual um bloco de texto claro é cifrado através de uma operação com matrizes.

1930

A máquina SIGABA (M-134-C) é inventada nos EUA por **William F. Friedman**. Deavours atribui a idéia a **Frank Rowlett**, um dos primeiros a serem contratados por Friedman.

As invenções de rotores de **Hebern** e **Scherbius** foram aperfeiçoadas usando escalonamentos pseudo-randômicos de múltiplos rotores em cada passo da cifragem ao invés dos escalonamentos uniformes, do tipo odômetro, dos rotores da Enigma. Além disso, usava 15 rotores (10 para a transformação de caracteres e 5, provavelmente, para controlar os passos) no lugar dos 3 ou 4 rotores da Enigma.

A máquina inglesa TYPEX era uma imitação da Enigma comercial adquirida pelos britânicos em 1920 para estudos. Era uma máquina de 5 rotores, dos quais os dois primeiros eram "stators" (estáticos) cuja função era a mesma do painel de plugs da Enigma alemã.

1933 a 1945

A máquina Enigma não foi um sucesso comercial, porém foi aperfeiçoada até se transformar na ferramenta criptográfica mais importante da Alemanha nazista. O sistema foi quebrado pelo matemático polonês **Marian Rejewski** que se baseou apenas em textos cifrados interceptados e numa lista de três meses de chaves diárias obtidas através de um espião. **Alan Turing** e **Gordon Welchman** e outros, em Bletchley Park, Inglaterra, deram continuidade à criptoanálise do sistema Enigma.

1937

A Máquina Púrpura (Purple Machine) dos japoneses foi inventada a partir das revelações feitas por **Herbert O. Yardley** e seu sistema foi quebrado por uma equipe liderada por **William Frederick Friedman**. A Máquina Púrpura usava relés telefônicos escalonados ao invés de rotores, apresentando, portanto, permutações totalmente diferentes a cada passo ao invés das permutações relacionadas de um rotor em diferentes posições.

Os japoneses não foram capazes de quebrar os códigos dos EUA e imaginavam que seu próprio código também fosse inquebrável - não foram suficientemente cuidadosos.

1943

Colossus, um computador para quebrar códigos, é posto em ação no Bletchley Park.

1943 a 1980

O projeto criptográfico Venona, conduzido pela NSA (National Security Agency dos EUA), é o mais duradouro dos projetos deste tipo.

1948

Elwood Shannon, um dos primeiros criptólogos a introduzir a matemática na criptologia, publica seu livro *A Communications Theory of Secrecy Systems*.

Figura B.26-Elwood Shannon

Década de 1960

O Dr. **Horst Feistel**, liderando um projeto de pesquisa no IBM Watson Research Lab, desenvolve a cifra Lucifer. Alguns anos depois, esta cifra servirá de base para o DES e outros produtos cifrantes, criando uma família conhecida como "cifras Feistel".

1969

James Ellis desenvolve um sistema de chaves públicas e chaves privadas separadas.

1976

Em 1974 a IBM apresenta a cifra Lucifer ao NBS (National Bureau of Standards) o qual, após avaliar o algoritmo com a ajuda da NSA (National Security Agency), introduz algumas modificações (como as Caixas S e uma chave menor) e adota a cifra como padrão de encriptação de dados para os EUA o FIPS PUB-46, conhecido hoje como DES (Data Encryption Standard).

Na ocasião, **Diffie** e **Hellman** já lançaram dúvidas quanto à segurança do DES, apontando que não seria impossível obter a chave através da "força bruta", o que acabou acontecendo 20 anos mais tarde e com um custo 100 vezes inferior ao inicialmente estimado.

Hoje o NBS é chamado de National Institute of Standards and Technology, NIST.

Whitfield Diffie e **Martin Hellman** publicam seu livro *New Directions in Cryptography*, introduzindo a idéia de uma criptografia de chave pública. Também reforçaram a concepção da autenticação através de uma função de via única (one way function), agora usada no utilitário S/Chave pedido de senha/resposta. Terminaram o texto com a seguinte observação: "A habilidade de produzir criptoanálise esteve sempre acentuadamente do lado de profissionais mas a inovação, particularmente no desenvolvimento de novos tipos de sistemas criptográficos, veio basicamente de amadores."

1976

Em 1974 a IBM apresenta a cifra Lucifer ao NBS (National Bureau of Standards) o qual, após avaliar o algoritmo com a ajuda da NSA (National Security Agency), introduz algumas modificações (como as Caixas S e uma chave menor) e adota a cifra como padrão de encriptação de dados para os EUA o FIPS PUB-46, conhecido hoje como DES (Data Encryption Standard). Hoje o NBS é chamado de National Institute of Standards and Technology, NIST.

Na ocasião, **Diffie** e **Hellman** já lançaram dúvidas quanto à segurança do DES, apontando que não seria impossível obter a chave através da "força bruta", o que acabou acontecendo 20 anos mais tarde e com um custo 100 vezes inferior ao inicialmente estimado.

1977

Inspirados no texto publicado por **Diffie** e **Hellman** e como absolutos principiantes na criptografia, **Ronald L. Rivest**, **Adi Shamir** e **Leonard M. Adleman** começaram a discutir como criar um sistema de chave pública prático. Ron Rivest acabou tendo um grande idéia e a submeteu à apreciação dos amigos: era uma cifra de chave pública, tanto para confidencialidade quanto para assinaturas digitais, baseada na dificuldade da fatoração de números grandes. Foi batizada de RSA, de acordo com as primeiras letras dos sobrenomes dos autores. Confiantes no sistema, em 4 de Abril de 1970 os três entregaram o texto para **Martin Gardner** para que fosse publicado na revista Scientific American. O artigo apareceu na edição de Setembro de 1977 e incluía a oferta de enviar o relatório técnico completo para qualquer um que enviasse um

envelope selado com o próprio endereço. Foram recebidos milhares de pedidos provenientes dos quatro cantos do mundo.

Alguém da NSA (National Security Agency dos EUA) contestou a distribuição deste relatório para estrangeiros e, durante algum tempo, os autores suspenderam a correspondência. Como a NSA não se deu ao trabalho de informar a base legal desta proibição, solicitada pelos autores, os três voltaram a enviar os relatórios solicitados. Dois jornais internacionais, "Cryptologia" e "The Journal of Cryptology", foram fundados logo após esta tentativa da NSA de censurar publicações.

Rivest, **Shamir** e **Adleman**, não publicaram a cifra antes de patenteá-la, aliás, foi uma novidade conseguir patentear um algoritmo.

1978

O **algoritmo RSA** é publicado nas "Communication" da ACM.

1984-1985

A **cifra rot13** foi introduzida no software USENET News para permitir a cifragem de mensagens, prevenindo que olhos inocentes fossem assaltados por algum texto questionável. Que me conste, este é o primeiro exemplo de uma cifra amplamente conhecida que tenha funcionado.

1986

Criptografia de curva elíptica sugerida por **Miller**.

Década de 1990

Trabalhos com computadores quânticos e criptografia quântica. Trabalhos com biométrica para autenticação (impressões digitais, a íris, etc).

1990

Xuejia Lai e **James Massey** publicam na Suiça "A Proposal for a New Block Encryption Standard" ("Uma Proposta para um Novo Padrão de Encriptação de Bloco"), o assim chamado IDEA (International Data Encryption Algorithm), para substituir o DES. O IDEA utiliza uma chave de 128 bits e emprega operações adequadas para computadores de uso geral, tornando as implementações do software mais eficientes.

Charles H. Bennett, **Gilles Brassard** e colaboradores publicam seus resultados experimentais sobre Criptografia Quântica, a qual usa fotons únicos para transmitir um fluxo de bits chave para uma posterior cifragem Vernam da mensagem (ou outros usos). Considerando as leis que a mecânica quântica possui, a Criptografia Quântica não só oferece a possibilidade do segredo como também uma indicação positiva de interceptação e uma medida do número máximo de bits que possam ter sido interceptados. Uma desvantagem é que a Criptologia Quântica necessita de um cabeamento de fibra ótica entre as partes que se comunicam.

1991

Phil Zimmermann torna pública sua primeira versão de PGP (Pretty Good Privacy) como resposta ao FBI, o qual invoca o direito de acessar qualquer texto claro da comunicações entre cidadãos. O PGP oferece uma segurança alta para o cidadão comum e, como tal, pode ser encarado como um concorrente de produtos comerciais como o Mailsafe da RSADSI. Entretanto, o PGP é especialmente notável porque foi disponibilizado como freeware e, como resultado, tornou-se um padrão mundial enquanto que seus concorrentes da época continuaram absolutamente desconhecidos.

1993

A criptoanálise diferencial é desenvolvida por **Biham** e **Shamir**.

1994

O professor **Ron Rivest**, autor dos algoritmos RC2 e RC4 incluídos na biblioteca de criptografia BSAFE do RSADSI, publica a proposta do algoritmo RC5 na Internet. Este algoritmo usa rotação dependente de dados como sua operação não linear e é parametrizado de modo que o usuário possa variar o tamanho do bloco, o número de estágios e o comprimento da chave. Uma análise feita pelo RSA Labs, mostrada na CRYPTO '95, sugeriu que w=32 e r=12 proporcionam uma segurança maior que a do DES.

O algoritmo blowfish, uma cifra de bloco de 64 bits com uma chave de até 448 bits de comprimento, projetado por **Bruce Schneier**.

1997

O PGP 5.0 Freeware é amplamente distribuído para uso não comercial.

O código DES de 56 bits é quebrado por uma rede de 14.000 computadores.

1998

O código DES é quebrado em 56 horas por pesquisadores do Vale do Silício. Em 1999 é quebrado em apenas 22 horas e 15 minutos.

2000

O algoritmo Rijndael é selecionado como Advanced Encryption Standard para substituir o DES.

C Outros trabalhos utilizando MATLAB

C.1 "Controlo de equipamento laboratorial"

Universidade do Algarve / Unidade de Ciências Exactas e Humanas / Sector de Electrónica e Computação

Trecho do resumo:

"O Matlab é um pacote de software que devido às suas reconhecidas capacidades de cálculo, possui grandes potencialidades no âmbito da simulação. Por outro lado embora não permita fazer programação de equipamentos para controlo e aquisição de dados, permite o interface com a linguagem de programação C. Por estas razões foi escolhida esta plataforma para o desenvolvimento deste projecto: controlo remoto através de um computador do Gerador de Funções PHILIPS PM5139."

Referência:

http://w3.ualg.pt/~alima/public/proj/projecto.html

Trabalhos de Laboratório - Licenciatura/Mestrado em Engenharia Electrotécnica e de Computadores

Secção de Sistemas Digitais / Departamento de Engenharia Electrotécnica / Instituto Superior Técnico

C.2 "Simulação de protocolos MAC 802.11 com MATLAB"

Secção de Sistemas Digitais / Departamento de Engenharia Electrotécnica / Instituto Superior Técnico / Trabalhos de Laboratório - Licenciatura/Mestrado em Engenharia Electrotécnica e de Computadores

Descrição:

Desenvolvimento de uma aplicação em MATLAB para simular o protocolo MAC 802.11 cobrindo vários cenários de aplicação.

C.3 "Simulação de protocolos Bluetooth com MATLAB"

Secção de Sistemas Digitais / Departamento de Engenharia Electrotécnica / Instituto Superior Técnico / Trabalhos de Laboratório - Licenciatura/Mestrado em Engenharia Electrotécnica e de Computadores

Descrição:

Desenvolvimento de uma aplicação em MATLAB para simular protocolos Bluetooth cobrindo vários cenários de aplicação.

Referência:

http://digitais.ist.utl.pt/ec-cm/laboratorio.html

C.4 "Recepção de transmissões digitais acústicas"

Secção de Sistemas Digitais / Departamento de Engenharia Electrotécnica / Instituto Superior Técnico

Trechos do trabalho:

"Os vários algoritmos serão validados através de simulação, e posteriormente recorrendo aos sinais em banda de base sem distorção obtidos directamente na saída do emissor..."

"No âmbito deste trabalho deverá ser desenvolvido código em Matlab (e/ou Simulink) para desmodulação dos sinais gerados por um modem para comunicação acústica submarina. As arquitecturas de propostas para os receptores deverão permitir lidar com interferência intersimbólica moderada ou forte, que ocorre frequentemente em canais submarinos. Os algoritmos deverão ser testados em dados reais, e caracterizadas as condições em que o desempenho é (ou não) aceitável. Deverá ser elaborada uma base de dados de sinais devidamente segmentados e catalogados como suporte ao trabalho realizado, e a realizar no futuro. "

Referência:

http://digitais.ist.utl.pt/digitais/TFC/04/tfc_leec0405_199.html

C.5 "Sobre a aplicabilidade do modelo de tráfego pseudo auto-similar para a alocação de recursos em redes ATM"

Publicação apresentada no SBrT 2000 - XVIII Simpósio Brasileiro de Telecomunicações, 3 a 6 de Setembro, 2000, Gramado-RS

Trechos do trabalho:

"Embora o modelo pseudo auto-similar seja a tempo discreto, a aproximação foi implementada para os dois casos (discreto e contínuo) no MatLab."

"... uma implementação do estimador para o MatLab, que foi empregada na avaliação das amostras de tráfego FBM geradas pelo algoritmo RMD..."

"Nesta seção serão relatados os resultados preliminares do nosso estudo, comparando as probabilidades de perdas obtidas através da simulação do FBM (considerada aqui como uma representação mais fiel do tráfego real)..."

Referência:

http://www.nuperc.unifacs.br/suruagy/publicacoes/pseudoas.pdf

C.6 "Implementação e aperfeiçoamento do laboratório de comunicações ópticas do ENE."

Iniciação científica (Graduando em Engenharia Eletrica) - Fundação Universidade de Brasília, Conselho Nacional de Desenvolvimento Científico e Tecnológico/DF.

Descrição:

"Este projeto teve início em março de 1997, e tem como principal objetivo o aperfeiçoamento e a implementação do Laboratório de Comunicações Ópticas do Departamento de Engenharia Elétrica. Ele foi dividido em duas partes distintas: Consolidação dos Roteiros de experiências práticas do Laboratório; Implementação de modelos de componentes optoeletrônicos no programa SIMULAB. A primeira parte deste projeto foi concluída em julho de 1997. A segunda parte, que é o objetivo desta atual pesquisa, trata-se da implementação de modelos matemáticos de componentes optoeletrônicos, tais como fibra óptica, laser, fotodetectores e splices, no

programa simulab. A implementação dos modelos matemáticos serão feitos na linguagem C++, sendo que os modelos utilizados são de livros e de artigos científicos recentes, e são testados no programa MATLAB / SIMULINK.

Referência:

http://www.ene.unb.br/~leonardo/orienta.htm