Desenvolvimento de um laboratório virtual baseado em PHP para estudo de criptografia RSA em ambiente de internet

Fabricio da Silva Valadares Xavier

Ceará, 2005
(Edição do Autor)

FUNDAÇÃO COMUNITÁRIA TRICORDIANA DE EDUCAÇÃO

Decretos Estaduais n.º 9.843/66 e n.º 16.719/74 e Parecer CEE/MG n.º 99/93

UNIVERSIDADE VALE DO RIO VERDE DE TRÊS CORAÇÕES

Decreto Estadual n.º 40.229, de 29/12/1998

Pró-Reitoria de Pós-Graduação, Pesquisa e Extensão.

Desenvolvimento de um laboratório virtual baseado em PHP para estudo de criptografia RSA em ambiente de internet

Três Corações

2005

FABRICIO DA SILVA VALADARES XAVIER

Desenvolvimento de um laboratório virtual baseado em PHP para estudo de criptografia RSA em ambiente de internet

Projeto de Conclusão de Curso apresentado ao Programa de Graduação em Ciência da Computação da Universidade Vale do Rio Verde, como requisito obrigatório da disciplina Estágio Supervisionado VI.

Orientadores

Prof. Marcos Paulo Valadares de Oliveira
Prof. Ms. Luiz Eduardo da Silva

Três Corações

2005

Aos pais.

E a todas as pessoas que estiveram envolvidas, direta e indiretamente.

OFEREÇO

AGRADECIMENTOS

A Deus, por dar-me força nesta conquista.

Aos pais, pelo apoio e incentivo para vencer mais esta etapa.

Aos irmãos, pela confiança transmitida.

Ao orientador, Marcos Paulo Valadares de Oliveira, pelos ensinamentos passados, pela amizade, pela compreensão e pela brilhante orientação.

Ao co-orientador, Ms. Luiz Eduardo da Silva, pela amizade, pelo incentivo e ensinamentos transmitidos desde a iniciação científica.

A Professora Ms. Jocyare Cristina Pereira de Souza, pelas sugestões e correções.

Aos amigos, pelo convívio de vários anos, pelas palavras carinhosas de incentivo e ajuda na correção deste trabalho.

À Universidade Vale do Rio Verde (UNINCOR) e a todos os colegas professores.

A todos que, de alguma forma, contribuíram para o meu êxito profissional.

"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".

Declaração Universal dos Direitos Humanos, art. 19

RESUMO

XAVIER, Fabrício da Silva Valadares. ***Desenvolvimento de um laboratório virtual baseado em PHP para estudo de criptografia RSA em ambiente de internet***. 2005. (Projeto de Conclusão de Curso – Ciência da Computação). Universidade Vale do Rio Verde – UNINCOR – Três Corações – MG[1]

Com a crescente utilização dos computadores em 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 por meio de procedimentos matemáticos, genericamente conhecidos como algoritmos de criptografia. Este trabalho visa implementar uma ferramenta em linguagem PHP e HTML de fácil manuseio, em linguagem interpretada e não compilada, onde será possível além de aprender sobre o algoritmo RSA de criptografia, 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 algoritmo de criptografia assimétrica. O algoritmo assimétrico foi o RSA. Na Criptografia assimétrica também conhecida como criptografia de chave pública é empregado dois algoritmos que usam cada um uma chave diferente, uma chamada pública e a outra privada.

[1]Comitê Orientador: Marcos Paulo Valadares de Oliveira – UFMG (Orientador), Luiz Eduardo da Silva – UNINCOR (Orientador)

ABSTRACT

XAVIER, Fabrício da Silva Valadares. ***Development of a based virtual laboratory in PHP for analysis of criptography RSA in Internet environment.*** 2005. (Project of Conclusion of Course - Computer science). Universidade Vale do Rio Verde – UNINCOR – Três Corações – MG[2]

With the increasing use of the computers the residential level, the computer networks in the companies and especially of the Internet, the security guard became an essential requirement. Although a great part of the transferred information way Internet to be of public access, exists a set of operations that require some level of security to be made right between the parts, as for example, transactions with number of credit cards or data of banking accounts, beyond access the private information using the Internet. These aspects of the security can be gotten through mathematical procedures, generically known as cryptography algorithms. This work aims at to implement a tool in language PHP and manuscript HTML easy, in interpreted and not compiled language, where it will be possible besides learning on cryptography algorithm RSA, to interact with the functions that will be implemented making possible one high degree of adjustment for each use case. The project is concentrated in the implementation of algorithm of anti-symmetrical criptography. The anti-symmetrical algorithm was the RSA. The anti-symmetrical cryptography also known as cryptography of public key is used two algorithms that use each one a different key, a public call and to another private one.

[2]Orienting committee: Marcos Paulo Valadares de Oliveira – UFMG (Orienting professor), Luiz Eduardo da Silva – UNINCOR (Orienting professor)

SUMÁRIO

LISTA DE ILUSTRAÇÕES

1. INTRODUÇÃO

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 se imagina 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 definir 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 o anonimato e fornece provas de identidade de pessoas. Pode, ainda, impedir vândalos de alterarem sua página na Internet e concorrentes 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 os usuários e evitar que a transação possa ser decifrada por pessoas não autorizadas, especialmente transações bancárias e de compras.

2. REVISÃO DE LITERATURA

Não custa lembrar que as centenas de milhões de computadores que estão por aí, na face da terra, rodam programas. E, um programa nada mais é do que a materialização de um algoritmo. Assim, o conceito de algoritmo adquiriu alguma importância no nosso dia-a-dia. Eis a razão pela qual voltar para a história da vida daquele que é considerado o criador do conceito.

2.1 O Criador dos Algoritmos

Como diz o autor Pedro Luis Kantek Garcia Navarro:

> Leonard Euler, nascido na Basiléia em 1707, filho do pastor calvinista Paul Euler. O pai já havia escolhido a profissão do filho: Teologia. Embora dono de um talento impressionante para a matemática o filho obedientemente foi estudar teologia e hebraico na Universidade da Basiléia. Lá travou contato com dois irmãos Daniel e Niklauss Bernouilli, que vêm a ser membros da famosa família de matemáticos. Nada menos que 8 componentes deste clã deixaram seu nome na matemática em espaço inferior a 100 anos. A família Bernouilli era famosa: Daniel contava que um dia recebera o maior elogio de sua vida. Viajando incógnito, durante um passeio, travou conversa com um desconhecido. No meio do papo, humildemente, apresentou-se: *"Eu sou Daniel Bernouilli",* ao que o conhecido fez cara de zombaria e respondeu cheio de pompa "E eu, sou Isaac Newton". Nos dias de hoje se diria "e eu, sou a Madonna".
>
> Pois, voltando aos dois irmãos, logo depois de terem conhecido Euler, chegaram à conclusão de que valia a pena a humanidade perder um pregador medíocre em troca de um grande matemático. Foram convencer Paul Euler a que liberasse o filho. O velho Paul, que fora contemporâneo na escola de Jakob Bernouilli, o pai de todos, e tinha pelos Bernouilli muito respeito, aceitou relutantemente que o filho deixasse a Teologia.
>
> Euler passou a se interessar por quantos problemas passassem perto dele: estudou a navegação, as finanças, acústica, irrigação, entre outras questões. Cada novo problema levava Euler a criar matemática inovadora e engenhosa. Conseguia escrever diversos trabalhos em um único dia e contava-se que entre a primeira e a segunda chamada para o jantar, era capaz de rascunhar cálculos dignos de serem publicados. Euler tinha memória e intuição e com eles trabalhava. Era capaz de realizar um cálculo completo de cabeça, sem pôr o lápis no papel. Foi conhecido ainda em vida como "a encarnação da análise". (NAVARRO, 2001)

O primeiro algoritmo de que se tem notícia, trabalhado por Euler, é a previsão das fases da lua. Relembrando, a terra atrai a lua e a lua atrai a terra, e o conhecimento deste fato com precisão ajuda a criar tabelas de navegação, fundamentais para um navio descobrir onde está. Não esquecendo que nesta época, século XVIII, não existia os GPS. O cálculo do comportamento da lua seria quase trivial se não aparecesse na história o sol. É o assim chamado "problema dos 3 corpos". Euler não achou uma solução, que de resto até hoje não existe, mas trabalhou num algoritmo que permitia calcular um primeiro valor aproximado

para a posição da lua. Reintroduzindo essa primeira posição no mesmo algoritmo, era possível obter um novo valor melhor, e agindo sucessivamente dessa forma poder-se-ia chegar ao valor com a precisão desejada. O Almirantado Britânico pagou 300 libras (um dinheirão) a Euler pelo algoritmo. Lembrando que, Euler deve ter sido o primeiro programador profissional da história do mundo.

> Quando passou pela Rússia, Euler foi convidado pela czarina Catarina (a grande) a ajudá-la a resolver um imenso problema. Andava pela corte Denis Diderot, francês famoso e ateu convicto. Diderot passava seu tempo tentando convencer as pessoas de que Deus não existia, o que deixava a religiosa Catarina furiosa. Euler, disse *"deixa comigo"* e imediatamente proclamou ter uma prova algébrica da existência de Deus. Rapidamente, Catarina convidou toda a corte para assistir o dilema teológico entre Euler e Diderot.
> No grande dia, Euler levantou-se, pigarreou, dirigiu-se à lousa e escreveu:
> "Senhor, (a+bn)/n = x, portanto Deus existe, refute!". O pobre do Diderot que era uma nulidade matemática não conseguiu dizer nada e humilhado, deixou a corte. Não é necessário explicar, que o argumento do Euler é um baita facão. Ele deve ter sido um ótimo jogador de truco.
> Outro problema estudado, pelo qual é atribuído a Euler a paternidade da Teoria dos Grafos, é o famoso problema das pontes de Königsberg (atual Kaliningrado). Explica-se: corta a cidade o Rio Pregel, formando o seguinte padrão de 4 regiões (margem esquerda, direita, ilha pequena e ilha grande) e 7 pontes.
> Desde a idade média desconfiava-se que não era possível sair de um lugar qualquer, atravessar as 7 pontes uma única vez cada uma e retornar ao ponto de partida. Euler conseguiu mostrar que para este caminho existir cada ponto deve ser ligado por um número par de pontes (ou deve haver apenas 2 lugares com pontes ímpares, se estes lugares forem à saída e a chegada e forem diferentes entre si). Até hoje, na teoria dos grafos, um caminho que goze desta propriedade é chamado caminho Euleriano.
> É de Euler a primeira contribuição importante para a solução do Último Teorema de Fermat (não existe n tal que $x^n + y^n = z^n$, para $n>2$). De fato ele provou que o teorema era verdadeiro para n=4. (NAVARRO, 2001)

2.2 A evolução dos Algoritmos de Primalidade e seu impacto na Criptografia

Existem duas grandes vertentes de algoritmos de criptografia: os simétricos e os assimétricos. Ambos são assim chamados em função de possuírem uma ou duas chaves, ou seja, os algoritmos simétricos dependem de uma única chave para criptografar e descriptografar uma mensagem, enquanto os assimétricos possuem uma chave pública para o processo de cifragem e uma privada para o processo de decifragem da mensagem.

Ambas as classes de algoritmos possuem aplicações distintas: algoritmos simétricos costumam ser mais rápidos, sendo úteis em aplicações que requerem uma implementação em hardware, tal como celulares e outros tipos de comunicação privadas, em que ambas as partes se confiam mutuamente. Agora, quando você precisar receber uma

mensagem cifrada de alguém que nem mesmo sabe quem será, é necessário lançar mão de um algoritmo de chave pública. Tal chave tem apenas o poder de cifrar, e pode ser distribuída livremente a qualquer pessoa interessada em enviar-lhe uma mensagem confidencialmente.

Munido da respectiva chave privada, você tem garantia de ser a única pessoa capaz de ler a sua "correspondência" secreta.

Dentro de cada uma destas classes de algoritmos de criptografia, existem diversas outras subclasses. “O projeto de algoritmos simétricos costuma ter sua segurança baseada na Teoria da Informação de Shannon, e seguem os mesmos princípios estatísticos enunciados nesta obra.” (Navarro, 2001).

2.3 Protocolo TCP/IP

O autor Douglas E. Comer demonstra no livro Interligação em Redes com TCP/IP uma introdução e visão geral que contém: A motivação para a interligação em redes, A interligação em reder TCP/IP, Serviços de interligações em redes, História e escopo da Internet, O conselho de arquitetura da Internet, A reorganização do IAB, a Internet Society, RFCs da Internet, Protocolos e padronização da Internet, Expansão e tecnologia, que vem descreve:

> Uma interligação em redes consiste em um conjunto de redes conectadas que agem como um todo coordenado. A principal vantagem é que ela proporciona uma interconexão universal, ao mesmo tempo em que permite que grupos de indivíduos utilizem qualquer hardware de rede que melhor atenda às suas necessidades. Examinaremos os princípios que orientam a comunicação de interligação em redes em geral e os detalhes de uma pilha de protocolos dessa interconexão, em particular. Também discutiremos como esses protocolos são utilizados em uma interconexão desse tipo. A tecnologia que utilizamos em nosso exemplo, denominada TCP/IP em virtude de seus dois protocolos principais, foi desenvolvida pela ARPA – Advanced Research Projects Agency. Ela fornece a base para Internet global, uma interligação em redes ampla e operacional que conecta universidades, organizações e departamentos do governo em muitos países em todo mundo. A Internet global está passando por uma rápida expansão. (COMER,1998)

Esse arquivo mostra como funciona o TCP/IP, assim como suas camadas e o que cada uma delas vem a ser e seu funcionamento são elas:

Camada de Apresentação:

> É formada pelos protocolos utilizados pelas diversas aplicações do modelo TCP/IP. Esta camada não possui um padrão comum. O padrão é estabelecido por cada aplicação. Isto é, o FTP possui seu próprio protocolo, assim como o TELNET, SMTP, POP3, DNS e etc. (SANTOS, 2002).

Camada de Transporte:

> É formada pelos protocolos utilizados pelas diversas aplicações do modelo TCP/IP. Esta camada não possui um padrão comum. O padrão é estabelecido por cada aplicação. Isto é, o FTP possui seu próprio protocolo, assim como o TELNET, SMTP, POP3, DNS e etc. (SANTOS, 2002).

Camada Internet (IP):

> Essa camada é a primeira normatizada do modelo. Também conhecida como camada Internet, é responsável pelo endereçamento, roteamento e controle de envio e recepção. Ela não é orientada à conexão, se comunica através de datagramas. (SANTOS, 2002).

Camada de Rede (hardware):

> Camada de abstração de hardware tem como principal função a interface do modelo TCP/IP com os diversos tipos de redes (X.25, ATM, FDDI, Ethernet, Token Ring, Frame Relay, PPP e SLIP). Por causa da grande variedade de tecnologias de rede, ela não é normatizada pelo modelo, o que provê a possibilidade de interconexão e interoperação de redes heterogêneas. (SANTOS, 2002).

Traz uma nota dizendo:

> A grande flexibilidade e interoperabilidade fornecidas pela arquitetura TCP/IP, atraíram os fabricantes e fornecedores de recursos e o mercado de informática como um todo, pois, esta arquitetura, permite interconectar ambientes heterogêneos de forma eficiente e, com isso, todos passaram a usar esta tecnologia em larga escala. (SANTOS, 2002).

2.4 Arquitetura Cliente / Servidor

Pesquisando o livro Redes de Computadores dos autores Luiz Fernando Gomes Soares, Guido Lemos e Sérgio Colcher, foi encontrado no capítulo 16 - Sistemas Operacionais de Redes a Arquitetura Cliente/Servidor explicando cada uma das funções dos módulos como: modulo do cliente,

> Vimos que a solução adotada foi à introdução de um redirecionador que manteve inalterada a interface usada pelas aplicações, aproveitando-as para permitir, inclusive, o acesso aos dispositivos remotos. Uma outra foi manter a utilização dos recursos (memória, processador etc.) dos clientes em níveis tais que o usuário não fosse atrapalhado em suas tarefas locais. (SOARES, LEMOS, COLCHER, 1995)

o módulo servidor,

> Estações que possuem tais módulos chamamos por simplificação de servidores, são distinguidas das outras estações pelo software de suporte ao serviço e algum hardware especial que contenham. Entre os serviços mais oferecidos podemos citar: armazenamento de arquivos, a gerência de banco de dados, o suporte para impressão, a tradução de nomes simbólicos em endereços físicos, a concentração de terminais, a monitoração de redes, a criptografia, o correio eletrônico e os serviços de comunicação. (SOARES, LEMOS, COLCHER, 1995).

e este ainda se subdivide em Servidor de Arquivos, Servidor de Banco e Dados, Servidor de Impressão, Servidor de Comunicação, Servidor de Gerenciamento e outros Servidores.

2.5 Linguagem de Programação PHP

No site da linguagem PHP no setor de documentação está escrito: “PHP (um acrônimo recursivo para "PHP: Hypertext Preprocessor") é uma linguagem de script Open Source (gratuita) de uso geral, muito utilizada especialmente para o desenvolvimento de aplicações Web dentro do HTML.” (PHP, 2005)

2.5.1 Um exemplo introdutório

O PHP como sendo uma linguagem de programação WEB pode-se ser inserido dentro do HTML, modo que o processamento seja unificado, como pode ser visto na figura 1.

```
<html>
<head>
<title>Exemplo</title>
</head>
<body>
    <?php
    echo "Olá, Eu sou um script PHP!";
    ?>
</body>
</html>
```

Figura 1 - Exemplo de PHP interagindo com HTML

O PHP é diferente de scripts CGI escritos em outras linguagens como Perl ou C - ao invés de escrever um programa com vários comandos para imprimir HTML, escreve-se um arquivo HTML com algum código inserido para fazer alguma coisa (no caso, imprimir um texto). O código PHP é delimitado por tags iniciais e finais que lhe permitem pular pra dentro e pra fora do "modo PHP".

> O que distingui o PHP de algo como Javascript no lado do cliente é que o código é executado no servidor. Se você tivesse um script em seu servidor, o cliente receberia os resultados da execução desse script, sem nenhum modo de determinar como é o código fonte. Você pode inclusive configurar seu servidor para processar todos os seus arquivos HTML como PHP, e então não haverá nenhum modo dos usuários descobrirem que se você usa essa linguagem ou não. (PHP, 2005)

A melhor coisa em usar PHP está no fato de ele ser extremamente simples para um iniciante, mas oferece muitos recursos para o programador profissional.

Apesar do desenvolvimento do PHP ser focado nos scripts do lado do servidor, pode-se fazer muito mais com ele.

2.5.2 Características

O PHP é focado para ser uma linguagem de script do lado do servidor, portanto, pode-se fazer qualquer coisa que outro programa CGI pode fazer, como: coletar dados de formulários, gerar páginas com conteúdo dinâmico ou enviar e receber informações. Mas o PHP pode fazer muito mais.

> Esses são os maiores campos onde os scripts PHP podem se utilizados: Script no lado do servidor (server-side). Este é o mais tradicional e principal campo de atuação do PHP. Você precisa de três coisas para seu trabalho. O interpretador do PHP (como CGI ou módulo), um servidor web e um browser. Basta rodar o servidor web conectado a um PHP instalado. Você pode acessar os resultados de seu programa PHP com um browser, visualizando a página PHP através do servidor web. Script de linha de comando. Você pode fazer um script PHP funcionar sem um servidor web ou browser. A única coisa necessária é o interpretador. Esse tipo de uso é ideal para script executados usando o cron (cronômetro de execuções, Linux) ou o Agendador de Tarefas (no Windows). Os scripts podem ser usados também para rotinas de processamento de texto. Escrevendo aplicações GUI (Grafical user interface) no lado do cliente (client-side). O PHP não é (provavelmente) a melhor linguagem para produção de aplicações com interfaces em janelas, mas o PHP faz isso muito bem, e se você deseja usar alguns recursos avançados do PHP em aplicações no lado do cliente poderá utilizar o PHP-GTK para escrever esses programas. E programas escritos desta forma ainda serão independentes de plataforma. O PHP-GTK é uma extensão do PHP, não disponível na distribuição oficial. (PHP, 2005)

O PHP pode ser utilizado na maioria dos sistemas operacionais, incluindo Linux, várias variantes Unix (incluindo HP-UX, Solaris e OpenBSD), Microsoft Windows, Mac OS X, RISC OS, e provavelmente outros. O PHP também é suportado pela maioria dos servidores web atuais, incluindo Apache, Microsoft Internet Information Server, Personal Web Server, Netscape and iPlanet Servers, Oreilly Website Pro Server, Caudium, Xitami, OmniHTTPd, e muitos outros. O PHP pode ser configurado como módulo para a maioria dos servidores, e para os outros como um CGI comum.

> Com o PHP, portanto, um programador tem a liberdade para escolher o sistema operacional e o servidor web. Do mesmo modo, o programador pode escolher entre utilizar programação estrutural ou programação orientada a objeto, ou ainda uma mistura deles. Mesmo não desenvolvendo nenhum recurso padrão de OOP (Object Oriented Programming, Programação Orientada a Objetos) na versão atual do PHP, muitas bibliotecas de código e grandes aplicações (incluindo a biblioteca PEAR) foram escritos somente utilizando OOP. Com PHP limita-se a gerar somente HTML. As habilidades do PHP incluem geração de imagens, arquivos PDF e animações Flash (utilizando libswf ou Ming) criados dinamicamente. Criar-se facilmente qualquer padrão texto, como XHTML e outros arquivos XML. O PHP pode gerar esses padrões e os salvar no sistema de arquivos, em vez de imprimi-los, formando um cache dinâmico de suas informações no lado do servidor. (PHP, 2005)

Talvez a mais forte e mais significativa característica do PHP é seu suporte a uma ampla variedade de banco de dados. Escrever uma página que consulte um banco de dados é incrivelmente simples.

Também foi implementada uma abstração de banco de dados DBX permitindo a você utilizar qualquer banco de dados transparentemente com sua extensão. Adicionalmente, o PHP suporta ODBC (Open Database Connection, ou Padrão Aberto de Conexão com Bancos de Dados), permitindo que se utilize qualquer outro banco de dados que suporte esse padrão mundial.

> O PHP também tem suporte para comunicação com outros serviços utilizando protocolos como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (em Windows) e incontáveis outros. Pode-se abrir sockets de rede e interagir diretamente com qualquer protocolo. O PHP também suporta o intercâmbio de dados complexos WDDX, utilizado em virtualmente todas as linguagens de programação para web. Falando de comunicação, o PHP implementa a instanciação de objetos Java e os utiliza transparentemente como objetos PHP. Pode-se ainda usar sua extensão CORBA para acessar objetos remotos. O PHP é extremamente útil em recursos de processamento de texto, do POSIX Estendido ou expressões regulares Perl até como interpretador para documentos XML. Para acessar e processar documentos XML, são suportados os padrões SAX e DOM. Você ainda pode usar nossa extensão XSLT para transformar documentos XML. Utilizando o PHP no campo do e-commerce, um programador poderá usar as funções específicas para Cybescash, CyberMUT, Verysign, Payflow Pro e CCVS, práticos sistemas de pagamento on-line. (PHP, 2005)

Por último, mas longe de terminar, temos também outras extensões interessantes: funções para o search engine mnoGoSearch, funções para Gateway IRC, vários utilitários de compressão (gzip, bz2), calendário e conversões de datas, tradução.

2.6 Criptologia e 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 de 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, "oi" em português equivale a "hola" em espanhol ou "hi" em inglês. A maioria dos códigos funciona com um "livro de códigos" onde estão relacionadas às equivalências, como se fosse um dicionário. Já a palavra cifra vem do hebraico saphar, que significa "dar número". A maioria das cifrações é 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. (SCHNEIER, WILEY, WILEY, 1996)

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 transformar 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. (SCHNEIER, WILEY, WILEY, 1996)

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

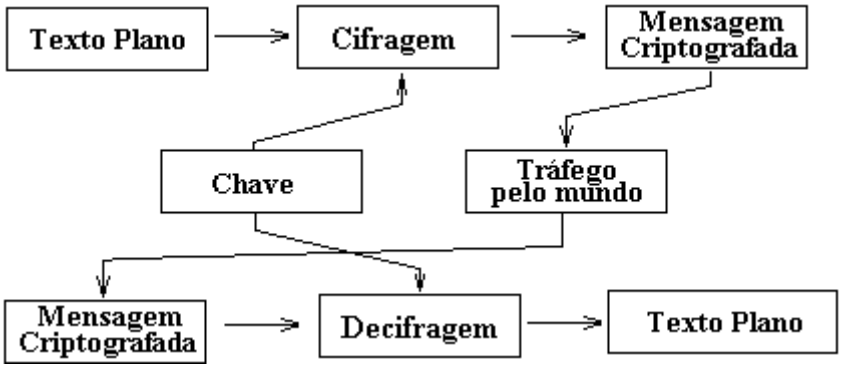


Figura 2 - Criptografia e transmissão de dados

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.

Pode-se 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 descriptografia, 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/descriptografar. 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 descriptografia (D). Em muitos casos não tem uma ordem específica de utilização, ou seja, uma mensagem criptografada com uma das chaves pode ser descriptografada pela outra, não importando qual será usada para criptografar, desde que a outra seja usada para descriptografar. (ou seja, se encriptar 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 emprega-se 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. (STINSON, 1995)

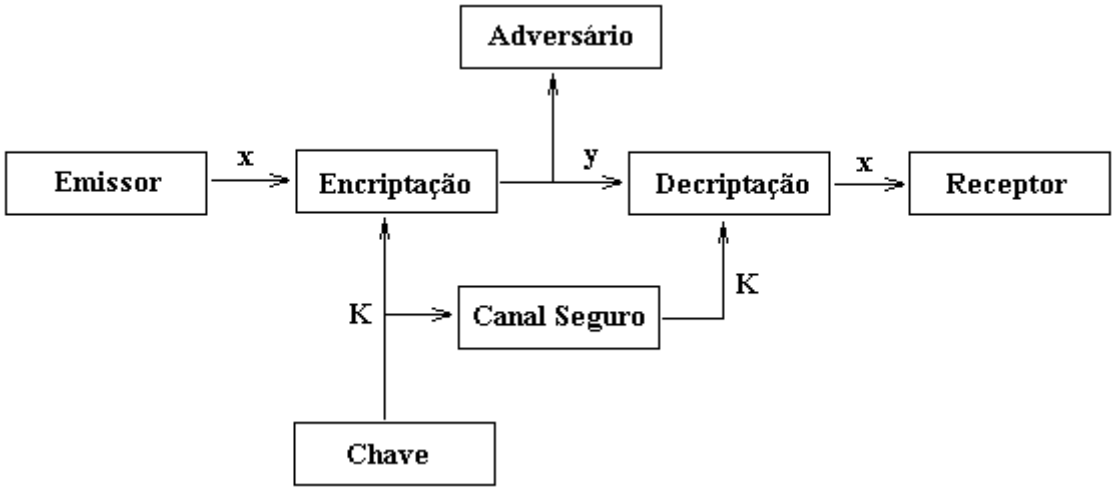


Figura 3 - Criptografia e ataques

> 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 descriptografar) é o processo inverso,

> ou seja, transformar os dados criptografados na sua forma original, inteligível. Para cifrar ou decifrar uma mensagem, necessita-se 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 descriptografar 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 o algoritmo ou função de decifragem (D) devendo, naturalmente. (SINGH, 2004)

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

As chaves devem definir univocamente o criptograma.

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

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.

> Nos 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. (STINSON, 1995)

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 K) de tal forma que 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, 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 aplicá-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.

2.6.1 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.

2.6.1.1 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. (DAPS, 1997)

2.6.1.2 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 nas quais 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. (DAPS, 1997)

2.6.1.3 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 à violação de autenticidade).

2.6.1.4 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 à negação do acesso a um serviço ou informação, como por exemplo: bloqueamento do canal de comunicação ou do acesso a servidores de dados. (DAPS, 1997)

2.6.1.5 Algoritmos de criptografia em blocos

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

2.6.1.6 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.

2.6.1.7 Robustez criptográfica

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

- O segredo da chave.
- A dificuldade de adivinhar a chave ou as árduas possibilidades de buscar a chave.
- A dificuldade de inverter o algoritmo de encriptação sem conhecer a chave de encriptação.
- 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 uma porção 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 são envolvidas em tais desenvolvimentos.

2.6.1.8 Tamanho criptográfico

Quando dizemos que um método de criptografia utiliza 512, 1024 ou 2048 bits estamos dizendo que o número de combinações que alguém precisa fazer para tentar decifrar nossa mensagem são 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 (equações matemáticas, polinômios) complexos que, em sua fórmula, utilizam como parte integrante o próprio texto a ser criptografado e uma "chave de criptografia".

2.6.1.9 Expiração de chaves

Para se precaver de ataques fatoriais em longo prazo, cada chave deve ter uma data de expiração 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 expiração.

A data de expiração de uma chave acompanha a chave pública em uma Identificação Digital. O programa de verificação de assinatura deve verificar a data de expiração, e não 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 não 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 expiração, 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 não foi comprometida. A Autoridade Certificadora Digital gerará então uma nova Identificação Digital para a mesma chave, e todas

as novas assinaturas farão menção à nova Identificação 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.

2.7 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.

2.7.1 Tipos de Ataques

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).

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).

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.

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.

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.

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. (DAPS, 1997)

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.

2.8 Algoritmos Simétricos

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, demonstrado na figura 4.

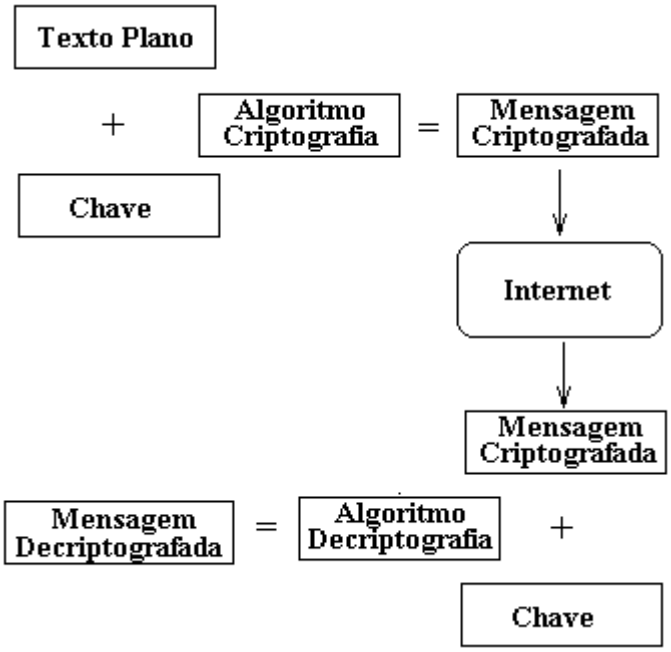


Figura 4 - Esquema para cifragem e decifragem com chaves simétricas

A ilustração acima, demonstra 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 seqüência 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 são denominados 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 terem 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). (MENEZES, OORSCHOT, VANSTONE, 1996)

A encriptação 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).

2.9 Algoritmos Assimétricos

Para fins de transações comerciais virtuais, os algoritmos simétricos se tornam pouco prático e inseguro, 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 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 usá-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 probabilisticamente 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 é 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. (SINGH, 2004)

O método de encriptação por chave publica resolve o problema de transmitir uma mensagem totalmente segura através de um canal inseguro (sujeito a observação, "grampo" etc.), como criar assinaturas digitais, correio eletrônico, verificar a origem dos dados e integridade. O receptor da mensagem cria duas chaves que são relacionadas entre si, uma pública e uma privada. A chave pública pode e deve ser distribuída livremente. Quem envia a

mensagem tem que utilizar a chave publica do receptor para encriptá-la. Uma vez encriptada, esta mensagem só pode ser desencriptada pela chave do receptor, visto na figura 5.

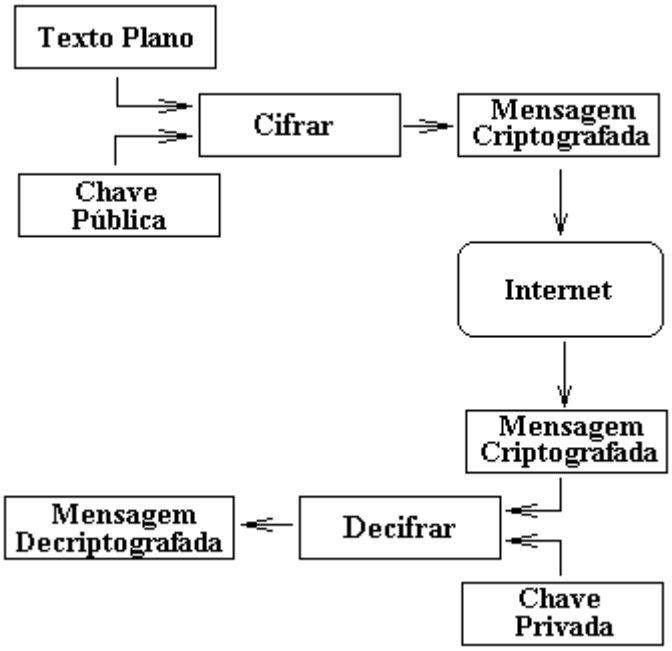


Figura 5 - Encriptação por chave pública

O conjunto de chaves públicas, uma para cada usuário, pode ser colocado numa lista acessível a todos os membros da rede.

2.9.1 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 decifrar 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. (Singh, 2004)

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 como segue nas figuras 6, 7 e 8.

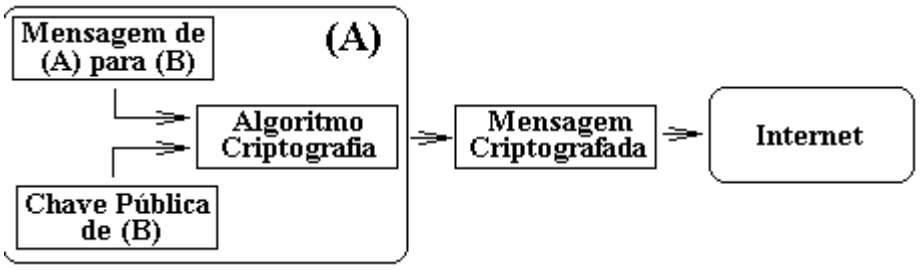


Figura 6 - Pessoa (A) criptografando uma mensagem para pessoa (B)

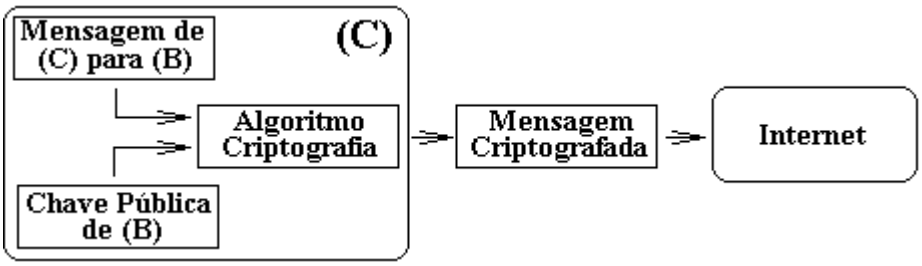


Figura 7 - Pessoa (C) criptografando uma mensagem para pessoa (B)

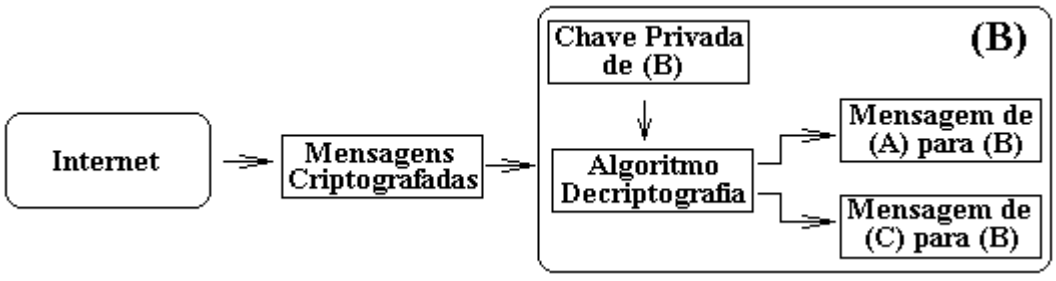


Figura 8 - Pessoa (B) descriptografando as mensagens recebidas de (A) e (C)

2.9.2 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 a chave secreta é transmitida usando a chave pública do destinatário, não se pode 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.

3. JUSTIFICATIVA

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 informações confidenciais de forma segura. As informações 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.

4. OBJETIVOS

4.1 Objetivos Gerais

Disponibilizar um material palpável, baseado na internet, para que profissionais, alunos e outras pessoas possam aprender como ocorre o processo de criptografia RSA passo-a-passo, já que não há muito material disponível em idioma Português (Brasil).

Desenvolver um sistema web de ensino da criptografia mais utilizada no momento em meios de transações e certificados digitais o RSA.

4.2 Objetivos Específicos

- Estudar o método e segurança de criptografia
- Estudo da criptografia RSA
- Desenvolver uma ferramenta de método didático para o estudo da criptografia citada
- Ensinar por meio da internet o método RSA de criptografia
- Disponibilizar gratuitamente a ferramenta para propagação da criptografia e do método

5 MATERIAIS E MÉTODOS

5.1 RSA (Rivest, Shamir e Adelman)

A encriptação por chaves públicas foi desenvolvida por Diffie and Hellman que criou uma forma aceitável de algoritmos para dois sistemas trocarem chaves criptográficas de forma segura (DES ou 3DES).

A primeira reunião para discutir o algoritmo RSA e suas características foi feita em 1977 por Rivest, Shamir e Adelman, sendo atualmente 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.

5.1.1 Características

O RSA possui várias características, mas, vela a pena mostrar algumas das mais importantes no seu estudo atual:

- Rivest-Shamir-Adleman são seus criadores
- Baseado na dificuldade de fatorar números primos grandes (Teoria dos Números)
- Maduro e bastante estudado por matemáticos e curiosos
- Segurança estrutural é uma idéia 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

5.1.2 Funcionamento do Algoritmo

A fim de obter as duas chaves, uma para cifrar e outra para decifrar, devemos proceder como segue os tópicos abaixo:

5.1.2.1 Obtenção das Chaves

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) x (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

e) 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:7
0: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:f
5:37:33:0c:3d:30:53:45:51:e9:4d:cf:b7:31:5a:c8:a1:a9:3b:80:92:58:8b:a6:0e:a9:8
3:16:83:91:3a:3f:99:72:23:5f:8a:dc:a1:1e:34:73:5f:10:a9:fa:f0:d9:d4:ad

Expoente: 65537 (0x10001)

5.1.2.2 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 e **C** representa o bloco criptografado seguindo como, por exemplo, se desejarmos encriptar a mensagem:

Criptografando

b) Transformamos em código ascii:

067114105112116111103114097102097110100111

c) Geramos as chaves (para este exemplo usaremos primos de valores pequenos). Por exemplo, para p = 6703 e q = 4909 temos:

e = 5303;
d = 16852991;
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:

28270723 28335380 12961910 3360204 16647278 17491042 25978915 28335380 22437584 11823588 22437584 5581343 19643582 17491042

f) Assim, a mensagem encriptada é:

28270723283353801296191033602041664727817491042259789152833538022437584118235882243758455813431964358217491042

5.1.2.3 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:

28270723283353801296191033602041664727817491042259789152833538022437584118235882243758455813431964358217491042

b) Primeiramente devemos separá-la em blocos de mesmo número de algarismos de n:

8270723 28335380 12961910 3360204 16647278 17491042 25978915 28335380 22437584 11823588 22437584 5581343 19643582 17491042

c) E aplicar a fórmula de decifragem a cada bloco, gerando o texto descriptografado:

0671141 0511211 6111103 1140971 0209711 0100111

d) Convertendo os códigos ascii:

Criptografando

Temos novamente a mensagem original.

5.2 Implementação do sistema

Toda a implementação foi voltada para o modo acadêmico, ou seja, é totalmente instrucional, de modo que não há requisito mínimo de conhecimento.

5.2.1 Requisitos de software

- Windows 2003 Server
- IIS 6
- PHP 5.0.4
- Internet Explorer 6.0
- Dreamweaver 8.0

5.2.2 Requisitos de hardware

- Athlon XP 2.4GHz
- HD 120GB
- 512MB-RAM
- Placa Mãe Asus A7V8X-X
- Monitor 17’’
- Mouse
- Teclado
- Modem ADSL D-LINK 502G

5.2.3 Descrição das funções principais

O RSA composto de várias funções, os quais devem estar no arquivo de trabalho “RSA.PHP”. As funções principais são: generate_chaves( ), rsa_encrypt( ) e

rsa_decrypt( ).

5.2.3.1 Função Generate_chaves

Este módulo é o responsável pela geração das chaves públicas (E e N) e da chave privada (D) utilizadas pelo algoritmo caso não haja nenhum primo definido a função escolherá aleatoriamente dentro de um array.

Sintaxe:
generate_chaves ($p,$q)

Onde:
$p e $q são os dois primeiros números primos escolhidos
Retorna as chaves

5.2.3.2 Função rsa_encrypt

É o responsável pela geração do criptograma (texto criptografado).

Sintaxe:
rsa_encrypt ($m, $e, $n)

Onde:
$m é texto que se quer cifrar
$e e $**n** são as chaves públicas
Retorna o texto criptografado

5.2.3.3 Função rsa_decrypt

É o responsável pela decifragem, restaurando assim o texto plano original.

Sintaxe:
rsa_decrypt ($c, $d, $n, $e)

Onde:
$c é texto cifrado
$e é o expoente
$d é a chave privada
$**n** é a chave pública
Retorna o texto original

5.2.4 Descrição das funções auxiliares

Os módulos acima são formados de diversas funções necessários a sua implementação, agora veremos em detalhes sua composição.

5.2.4.1 Auxiliares da função Generate_chaves

- mo( ) - Função Modular baseada em aritmética modular ou aritmética do relógio. Sintaxe: *mo($g,$l)*;

- gcd( ) - Calcula o máximo divisor comum entre dois números, se gcd = 1 os números são primos relativos. Sintaxe: *GCD($e,$pi)*;

- extend( ) - Calcula o inverso modulo n segundo o "Extended Euclidean Algorithm", ou seja, b-1 mod n. Sintaxe: *extend ($e, $pi)*;

- tofindE( ) - função para calcular E sob circunstâncias: GCD(N, e) = 1 e 1<E<N se cada teste E for principal, será muito menos loop´s, significando menos cálculos. Sintaxe: *tofindE($pi)*;

5.2.4.2 Auxiliares da função rsa_encrypt

- powmod( ) - Função para a exponenciação. Sintaxe: *powmod($base, $exp, $modulus)*;

5.2.4.3 Auxiliares da função rsa_dencrypt

- powmod( ) - Função para a exponenciação. Sintaxe: *powmod($base, $exp, $modulus)*;

5.2.5 Considerações gerais para implementação

Segue algumas considerações essenciais utilizados na implementação das funções e cálculos do projeto.

5.2.5.1 Verificação de primos – critério de Rabin Miller

Como visto em **APPLIED Cryptography: Protocols, Algorthms, and Source Code in C Second Edition, Inc.** (SCHNEIER, WILEY, WILEY, 1996)

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

5.2.5.2 Geração de Primos

Como visto em **APPLIED Cryptography: Protocols, Algorthms, and Source Code in C Second Edition, Inc.** (SCHNEIER, WILEY, WILEY, 1996)

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.

5.2.5.3 Exponenciação Modular (Método binário)

Como visto em **RSA Implementation** (KOÇ, 2002), 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

5.2.5.4 Máximo divisor comum (GCD – greatest common divisor)

Como visto em **Handbook of Applied Cryptography** (MENEZES, OORSCHOT, VANSTONE, 1996), 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)

5.2.5.5 Inversor Modular

Como visto em **Cryptography: Theory and Practice, Second Edition** (STINSON, 1995), 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 $b_0 \neq$ then
    b não tem inverso módulo n
    else
    b^{-1} = t mod n

5.3 Laboratório Virtual

Primeiramente digita-se um texto e escolhe dois números primos de tamanho de 4 bits, como ilustrado na Figura 9 abaixo:

Figura 9 - Tela principal do sistema em branco

Seguindo também podendo somente digitar o texto e deixar o sistema gerar automaticamente os números primos clicando no botão "Gerar nºs primos" – veja Figura 10.

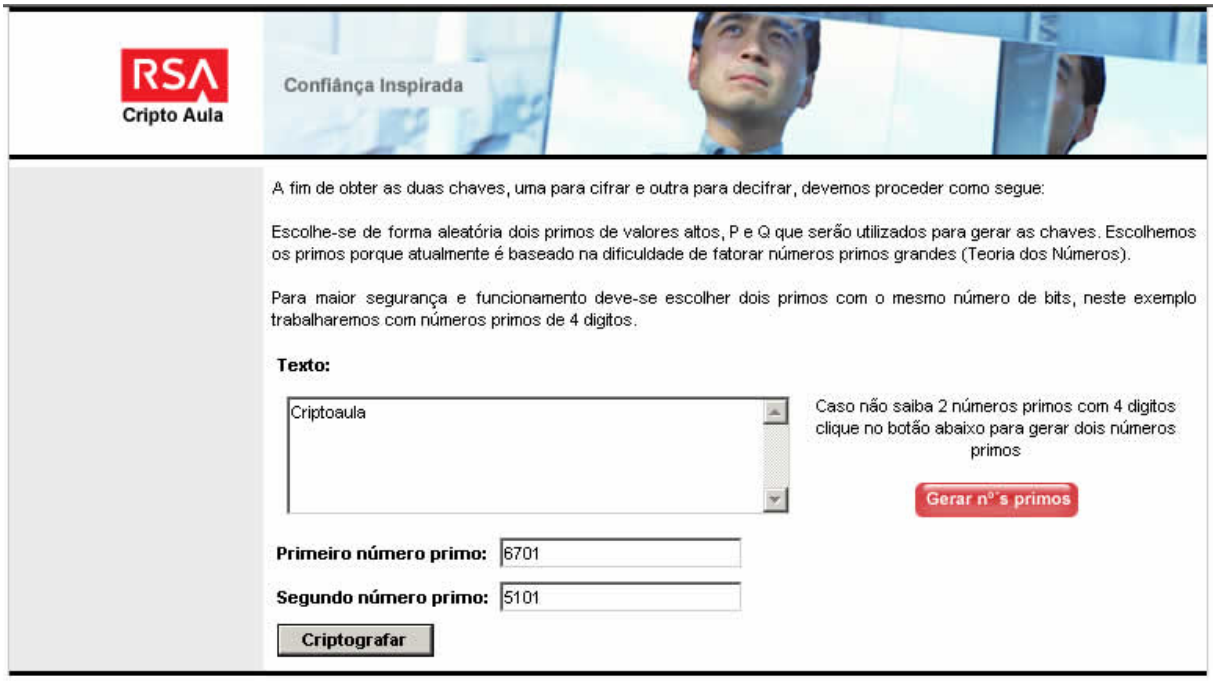


Figura 10 – Tela principal após preenchimento e geração automática dos primos

Logo após ter escolhido os números primos e digitado a mensagem clica-se no botão criptografar que segue para a obtenção de chaves de criptografia.

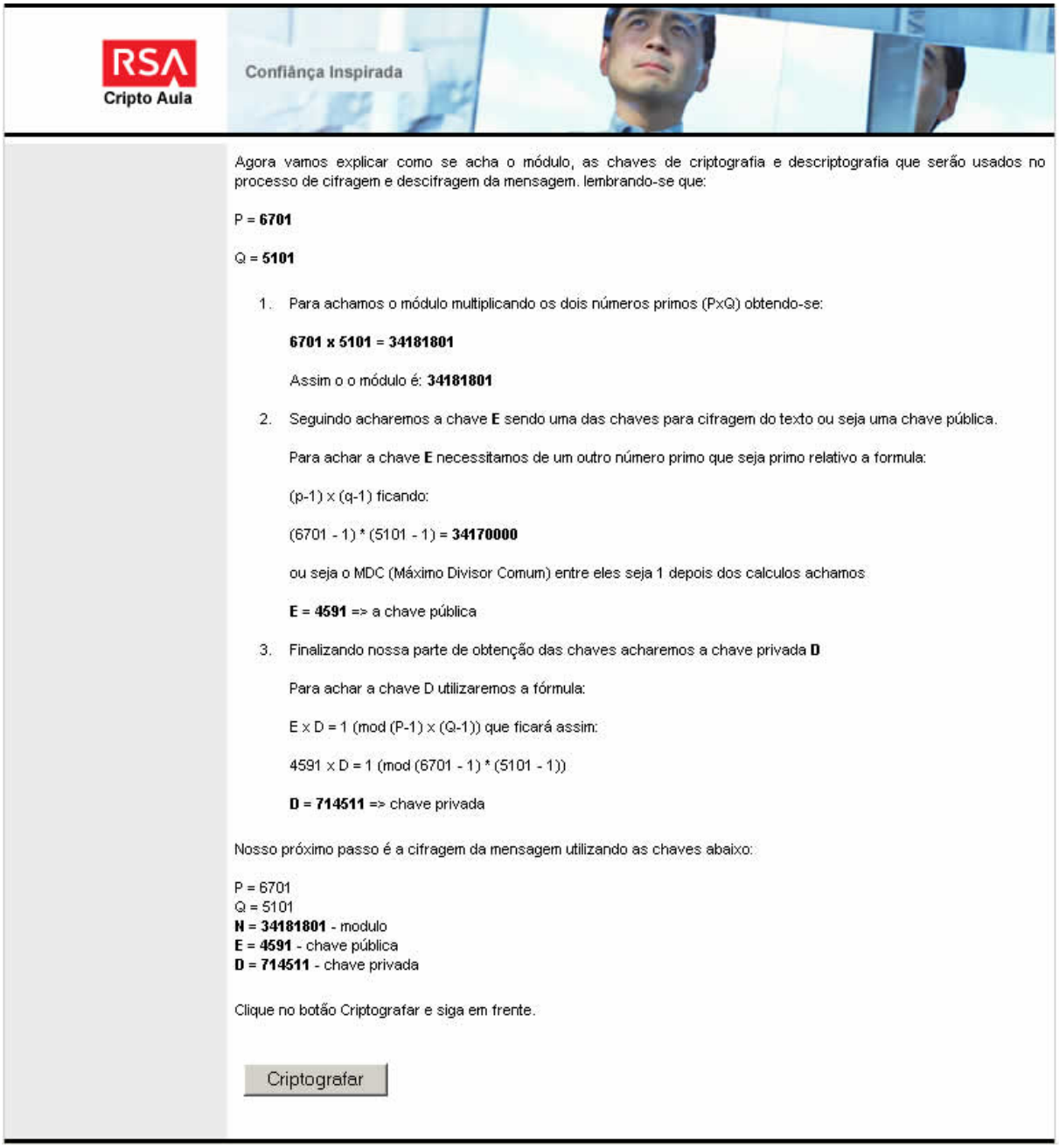


Figura 11 – Tela explicativa de obtenção de chaves

Nesta fase mostra-se na Figura 11 como é obtida as chaves pública, privada e o módulo além dos cálculos necessários para criptografia. O sistema escolherá aleatóriamente um terceiro número primo, que no caso será a chave pública a partir de uma base de primos interna.

Após a leitura do texto explicativo clica-se no botão criptografar para agora explicar o método da criptografia em si.

Na parte da criptografia propriamente dita explica-se o cálculo modular(aritmética do relógio) utilizada na obtenção de chaves e crifragem ou descifragem da mensagem.

Calcular diretamente em uma calculadora não é simples, porque o mostrados não pode exibir números tão grandes. Entretanto há um truque fácil para calcular exponenciais em aritmética modular. Como podemos ver no exemplo abaixo baseado nos primos P=17, Q=11 e E=7 crifrando apenas a letra X na Figura 12:

$$88^7 \pmod{187} = |88^4 \pmod{187}\ 88^2 \pmod{187} \times 88^1 \pmod{187}| \pmod{187}$$

$$88^1 \pmod{187} = 88 = 88 \pmod{187}$$

$$88^2 \pmod{187} = 7.744 = 77 \pmod{187}$$

$$88^4 \pmod{187} = 59.969.536 = 132 \pmod{187}$$

$$88^7 \pmod{187} = 88^1 \times 88^2 \times 88^4 = 88 \times 77 \times 132 = 894.432 = 11 \pmod{187}$$

Texto cifrado, C= 11 para ser enviado

Figura 12 – Cálculo exponencial em aritmética modular simplificado

Logo após o conhecimento da aritmética modular(aritmética do relógio) na mesma tela apresenta-se o texto quebrado em letras com seus respectivos valores na tabela ASCII e o calculo sendo feito letra a letra, finalizando o sistema apresenta o texto ou mensagem cifrada – veja Figura 13. Clicando no botão “Descriptografar” segue-se para a ultima tela que explica a decifragem do texto ou mensagem.

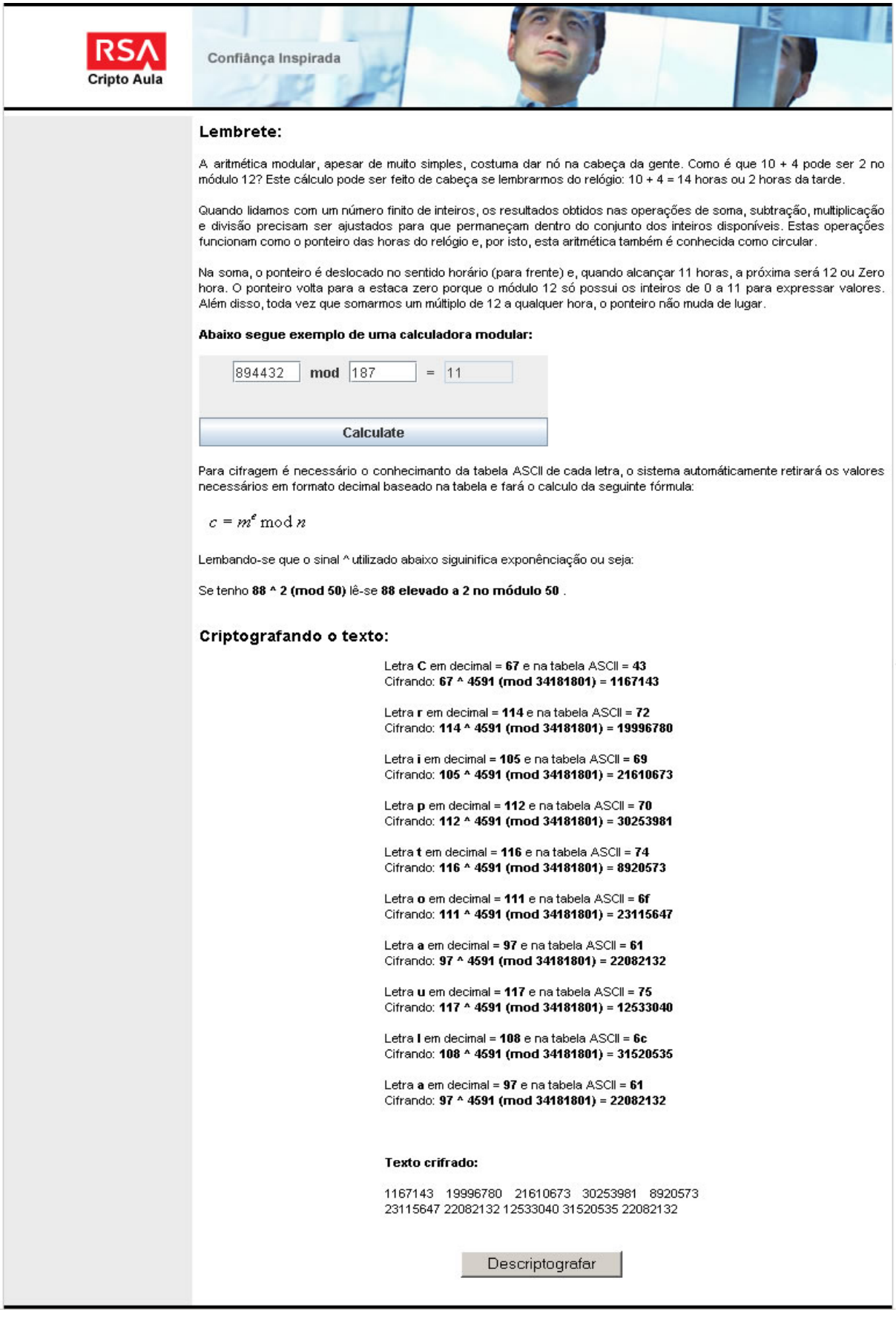


Figura 13 – Tela explicativa de aritmética modular e cifragem da mensagem

Clicando no botão “Descriptografar” segue-se para a última tela como abaixo na Figura 14 que explica a decifragem do texto ou mensagem. Na tela de decifragem apresenta-se um explicativo da fórmula utilizada, a decifragem com o cálculo e a letra resultante, finalizando faz-se a conferência com o texto utilizado para se saber se foi obtido a descifragem correta ou não e um link de reinício para se houver alguma dúvida possa voltar e recomeçar o processo de aprendizagem.

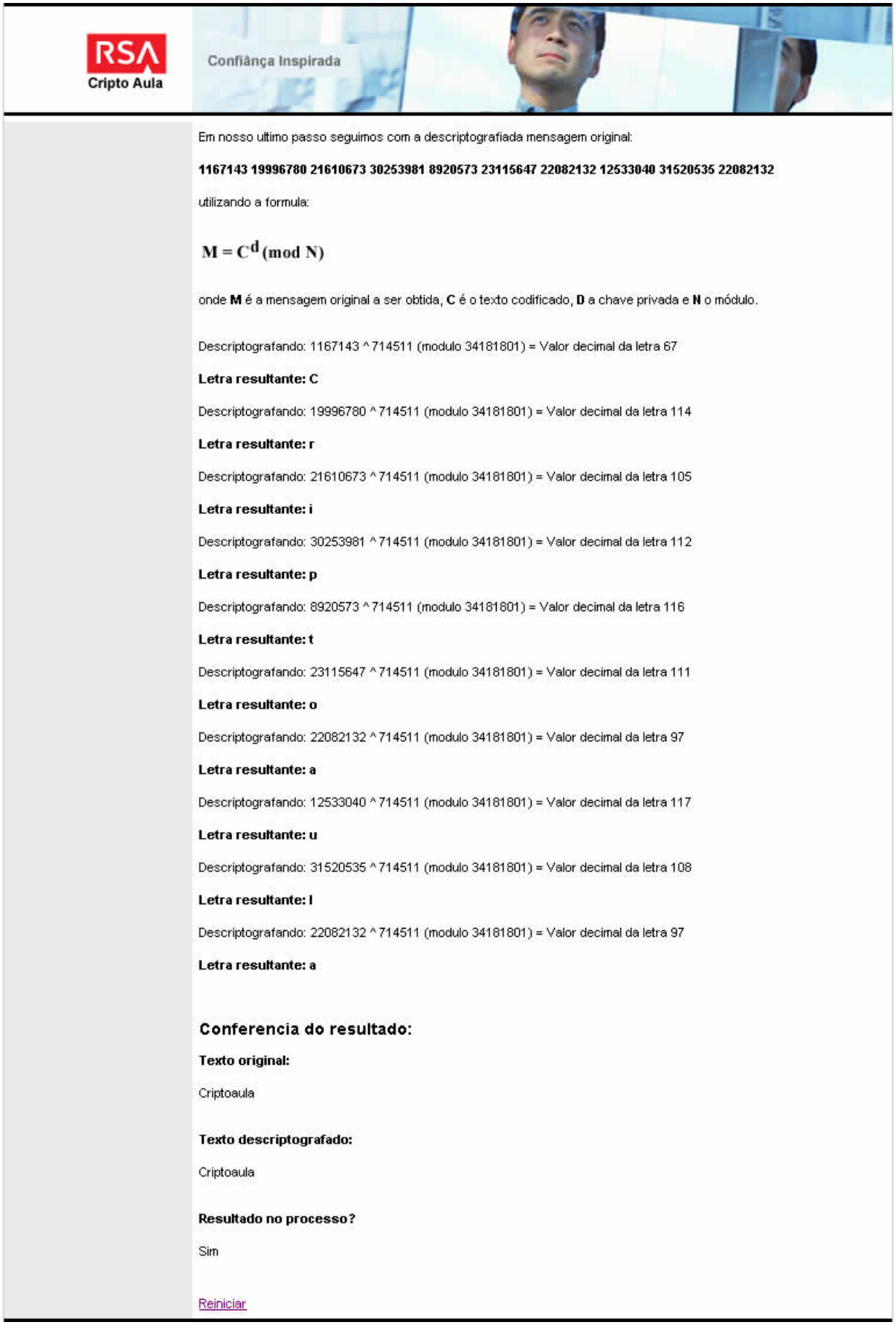


Figura 14 – Tela explicativa do processo de decifragem e conferencia do resultado

6 RESULTADOS E DISCUSSÃO

A proposta deste projeto foi a de combinar a facilidade do entendimento e alteração dos códigos interpretados gerados para o PHP com a estrutura web visando proporcionar um local com material de estudo do algoritmo de criptografia RSA.

O fato da linguagem PHP, ser interpretada, possibilita ao estudante não só a execução da aplicação, mas também a sua alteração com relativa facilidade. A partir de algumas alterações nos arquivos php, pode-se ter acesso, por exemplo, às funções e suas variáveis internas.

A aplicação do sistema amplia os meios de aprendizado sobre segurança (em mais detalhamento a criptografia RSA) demonstrando sua técnica, algoritmo e facilidade de aplicação seguindo as seguintes considerações de aprendizado dos autores:

- Benefício à sociedade com melhores condições para o estudo do método de criptografia RSA;
- Pesquisa e aplicação de programação em PHP, Configuração de Servidor Windows 2003 Server, Programação HTML, Algoritmos Estruturados, segurança da informação e métodos criptográficos;
- Disciplinas de estudo e pesquisa como Matemática Computacional, Algoritmos Estruturados, Linguagens de Programação, Segurança.

7 CONCLUSÃO

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.

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.

8 SUGESTÕES PARA TRABALHOS FUTUROS

Com a amplitude de vários tipos criptográficos hoje existentes, observando desde o meio de substituição simples e podendo chegar até a criptografia quântica. Tem-se vários caminhos de pesquisa e desenvolvimento para o aperfeiçoamento da ferramenta apresentada.

Primeiramente será a construção de um website que possibilite aos internautas experimentarem a criptografia, interagindo com o mesmo, podendo fazer download dos arquivos para posterior execução em modo off-line (sem necessidade da internet). Os arquivos do projeto contêem as implementações, os algoritmos estudados, seus métodos matemáticos, estatísticos e programáveis. O site disponibilizará também a documentação do algoritmo, explicando seu modo de funcionamento, suas características e um pouco de sua história.

Pode-se também ocorrer expansão do site com novos meios de criptografia, visando o estudo e desenvolvimento de outras ferramentas como a apresentada.

9 REFERÊNCIAL BIBIOGRÁFICO

A. J. MENEZES, P. C. V. OORSCHOT, S. A. VANSTONE, **Handbook of Applied Cryptography,** ISBN: 0849385237, data de publicação: 16/10/96.

COMER, Douglas E. **Interligação em Redes com TCP/IP** – Rio de Janeiro: Campos, 1998.

DAPS - Department of the Army Publicaiton Staff, Department of the Army. **Basic Cryptanalysis, Field Manual**, paginas 34-40-2. ISBN: 089412272X, data de publicação: 03/97.

KOÇ, Çetin Kaya. **RSA Implementation,** Oregon State University. Disponivel em <http://islab.oregonstate.edu/koc/ece679/notes/rsa1.pdf>. Data de acesso: 11/09/2005

NAVARRO, Pedro Luis Kantek Garcia. **O Criador dos Algoritmos –** data de publicação: 2001. Disponível em: <http://www.pr.gov.br/batebyte/edicoes/2001/bb116/criador.htm>. Data de acesso: 11/09/2005

SANTOS, Luiz Carlos dos. **Como funciona o TCP/IP ? –** data de publicação: 18/11/2002. Disponível em: <http://www.clubedasredes.eti.br/rede0007.htm>. Data de acesso: 11/09/2005

SCHNEIER B., WILEY C. P., WILEY J. S., **APPLIED Cryptography: Protocols, Algorthms, and Source Code in C Second Edition, Inc.** ISBN: 0471128457 data de publicação : 01/01/96

SINGH, Simon. **O Livro dos Códigos: A ciência do sigilo – do antigo Egito à criptografia quântica,** 4º Edição, Editora Record, Janeiro 2004

SOARES, Luiz Fernando Gomes; LEMOS, Guido; COLCHER, Sérgio. **Redes de Computadores: das LANs, MANs e WANs às Redes ATM -** Rio de Janeiro: Campos, 1995.

STINSON, Douglas. **Cryptography: Theory and Practice, Second Edition,** 1995.

10 GLOSSÁRIO

Abaixo segue um glossário com significados de algumas palavras encontrada neste estudo, o mesmo encontra-se em ordem alfabética.

Letra 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.

Letra B

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

Letra 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.

Letra 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 EUA que combina DSA e SHA-1 para especificar um formato para assinatura digital.

Letra 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.

Letra 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.

Letra 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.

Letra 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.

Letra 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.

Letra 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".

Letra 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.

Letra 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.

Letra 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".

Letra 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 - Seqüência de três letras consecutivas. Exemplo: ong, tio, ...

Letra W

WATERMARKING - Veja "Marca d'água".

11 APÊNDICES

Apêndice A

Segue o código fonte de todos os arquivos utilizador no sistema, incluindo os HTML, JAVASCRIPT e PHP.

RSA.PHP - Arquivo com funções principais e auxiliares

```
<?
/* somente gerador aleatório */
mt_srand((double)microtime()*1000000);

/* Tabela de números primos gravada em uma base array */
$primos = array (4507, 4513, 4517, 4519, 4523, 4547, 4549, 4561, 4567,
4583, 4591, 4597, 4603, 4621, 4637, 4639, 4643, 4649, 4651, 4657, 4663,
4673, 4679, 4691, 4703, 4721, 4723, 4729, 4733, 4751, 4759, 4783, 4787,
4789, 4793, 4799, 4801, 4813, 4817, 4831, 4861, 4871, 4877, 4889, 4903,
4909, 4919, 4931, 4933, 4937, 4943, 4951, 4957, 4967, 4969, 4973, 4987,
4993, 4999, 5003, 5009, 5011, 5021, 5023, 5039, 5051, 5059, 5077, 5081,
5087, 5099, 5101, 5107, 5113, 5119, 5147, 5153, 5167, 5171, 5179, 5189,
5197, 5209, 5227, 5231, 5233, 5237, 5261, 5273, 5279, 5281, 5297, 5303,
5309, 5323, 5333, 5347, 5351, 5381, 5387, 5393, 5399, 5407, 5413, 5417,
5419, 5431, 5437, 5441, 5443, 5449, 5471, 5477, 5479, 5483, 5501, 5503,
5507, 5519, 5521, 5527, 5531, 5557, 5563, 5569, 5573, 5581, 5591, 5623,
5639, 5641, 5647, 5651, 5653, 5657, 5659, 5669, 5683, 5689, 5693, 5701,
5711, 5717, 5737, 5741, 5743, 5749, 5779, 5783, 5791, 5801, 5807, 5813,
5821, 5827, 5839, 5843, 5849, 5851, 5857, 5861, 5867, 5869, 5879, 5881,
5897, 5903, 5923, 5927, 5939, 5953, 5981, 5987, 6007, 6011, 6029, 6037,
6043, 6047, 6053, 6067, 6073, 6079, 6089, 6091, 6101, 6113, 6121, 6131,
6133, 6143, 6151, 6163, 6173, 6197, 6199, 6203, 6211, 6217, 6221, 6229,
6247, 6257, 6263, 6269, 6271, 6277, 6287, 6299, 6301, 6311, 6317, 6323,
6329, 6337, 6343, 6353, 6359, 6361, 6367, 6373, 6379, 6389, 6397, 6421,
6427, 6449, 6451, 6469, 6473, 6481, 6491, 6521, 6529, 6547, 6551, 6553,
6563, 6569, 6571, 6577, 6581, 6599, 6607, 6619, 6637, 6653, 6659, 6661,
6673, 6679, 6689, 6691, 6701, 6703, 6709, 6719, 6733, 6737, 6761, 6763,
6779, 6781, 6791, 6793, 6803, 6823, 6827, 6829, 6833, 6841, 6857, 6863,
6869, 6871, 6883, 6899, 6907, 6911, 6917, 6947, 6949, 6959, 6961, 6967,
6971, 6977, 6983, 6991, 6997, 7001, 7013, 7019, 7027, 7039, 7043, 7057,
7069, 7079, 7103, 7109, 7121, 7127, 7129, 7151, 7159, 7177, 7187, 7193,
7207, 7211, 7213, 7219, 7229, 7237, 7243, 7247, 7253, 7283, 7297, 7307,
7309, 7321, 7331, 7333, 7349, 7351, 7369, 7393, 7411, 7417, 7433, 7451,
7457, 7459, 7477, 7481, 7487, 7489, 7499, 7507, 7517, 7523, 7529, 7537,
7541, 7547, 7549, 7559, 7561, 7573, 7577, 7583, 7589, 7591, 7603, 7607,
7621, 7639, 7643, 7649, 7669, 7673, 7681, 7687, 7691, 7699, 7703, 7717,
7723, 7727, 7741, 7753, 7757, 7759, 7789, 7793, 7817, 7823, 7829, 7841,
7853, 7867, 7873, 7877, 7879, 7883, 7901, 7907, 7919, 7927, 7933, 7937,
7949, 7951, 7963, 7993, 8009, 8011, 8017, 8039, 8053, 8059, 8069, 8081,
8087, 8089, 8093, 8101, 8111, 8117, 8123, 8147, 8161, 8167, 8171, 8179,
8191, 8209, 8219, 8221, 8231, 8233, 8237, 8243, 8263, 8269, 8273, 8287,
8291, 8293, 8297, 8311, 8317, 8329, 8353, 8363, 8369, 8377, 8387, 8389,
8419, 8423, 8429, 8431, 8443, 8447, 8461, 8467, 8501, 8513, 8521, 8527,
8537, 8539, 8543, 8563, 8573, 8581, 8597, 8599, 8609, 8623, 8627, 8629,
```

```
8641, 8647, 8663, 8669, 8677, 8681, 8689, 8693, 8699, 8707, 8713, 8719,
8731, 8737, 8741, 8747, 8753, 8761, 8779, 8783, 8803, 8807, 8819, 8821,
8831, 8837, 8839, 8849, 8861, 8863, 8867, 8887, 8893, 8923, 8929, 8933,
8941, 8951, 8963, 8969, 8971, 8999, 9001, 9007, 9011, 9013, 9029, 9041,
9043, 9049, 9059, 9067, 9091, 9103, 9109, 9127, 9133, 9137, 9151, 9157,
9161, 9173, 9181, 9187, 9199, 9203, 9209, 9221, 9227, 9239, 9241, 9257,
9277, 9281, 9283, 9293, 9311, 9319, 9323, 9337, 9341, 9343, 9349, 9371,
9377, 9391, 9397, 9403, 9413, 9419, 9421, 9431, 9433, 9437, 9439, 9461,
9463, 9467, 9473, 9479, 9491, 9497, 9511, 9521, 9533);

$maxprimos = count($primos) - 1;

/* Função para gerar chaves qunado não se digita os numeros primos.
Disposição do retorno onde
$array[0] -> modulo N
$array[1] -> Chave pública E
$array[2] -> Chave pública D
O par de chaves públicas é N e E
O par de chaves privadas é N e D
*/
function generate_chaves(){

     global $primos, $maxprimos;
     while (empty($e) || empty($d)) {

          /* encontrar 2 números primos $p e $q sendo que $p e $q deve ser
diferente e com mesmo número de caracteres */
          $p = $primos[mt_rand(0, $maxprimos)];
          while (empty($q) || ($p==$q)) $q = $primos[mt_rand(0,
$maxprimos)];
          //segunda parte dos pares públicos e confidenciais - N
          $n = $p*$q;

          //$pi (nós o necessitamos calcular D e E)
          $pi = ($p - 1) * ($q - 1);

          // Chave pública  E
          $e = tofindE($pi, $p, $q);

          // Chave privada D
          $d = extend($e,$pi);

          //Criação de uma array e armazenando valores
          $chaves = array ($n, $e, $d, $p, $q, $pi);
     }
     //Retorno de valores em array
     return $chaves;
}

/*Função para gerar chaves qunado se digita os numeros primos.
Disposição do retorno onde
$array[0] -> modulo N
$array[1] -> Chave pública E
$array[2] -> Chave pública D
O par de chaves públicas é N e E
O par de chaves privadas é N e D
*/
function generate_chaves2($p,$q){
          $primos = $p;
          $maxprimos = $q;
          global $primos, $maxprimos;
```

```
        //Primeira parte dos pares públicos e confidenciais - N
        $n = $p*$q;

        //$pi (nós o necessitamos para calcular D e E)
        $pi = ($p - 1) * ($q - 1);

        // Chave pública  E
        $e = tofindE($pi, $p, $q);

        // Chave privada D
        $d = extend($e,$pi);

        //Criação de uma array e armazenando valores
        $chaves = array ($n, $e, $d, $p, $q, $pi);

        //Retorno de valores em array
        return $chaves;
}

/*
Função Modular baseada em aritimética modular ou aritimética do relógio
*/
function mo ($g, $l) {
      $retorno = $g - ($l * floor ($g/$l));
      return $retorno;
      }

/*
* Método padrão para calcular D
* D = E-1 (mod N)
* Presume-se que D estará encontrado em menos de 16 iterações
*/
function extend ($Ee,$Epi) {
     $u1 = 1;
     $u2 = 0;
     $u3 = $Epi;
     $v1 = 0;
     $v2 = 1;
     $v3 = $Ee;
     while ($v3 != 0) {
          $qq = floor($u3/$v3);
          $t1 = $u1 - $qq * $v1;
          $t2 = $u2 - $qq * $v2;
          $t3 = $u3 - $qq * $v3;
          $u1 = $v1;
          $u2 = $v2;
          $u3 = $v3;
          $v1 = $t1;
          $v2 = $t2;
          $v3 = $t3;
          $z = 1;
     }
     $uu = $u1;
     $vv = $u2;
     if ($vv < 0) {
          $inverse = $vv + $Epi;
     } else {
          $inverse = $vv;
     }
     return $inverse;
```

```
}

/*
Este é o MDC do retorno da função para os números $e comuns e $pi os
maiores
*/
function GCD($e,$pi) {
     $y = $e;
     $x = $pi;
     while ($y != 0) {
          $w =  mo($x , $y);
          $x = $y;
          $y = $w;

     }
     return $x;
}

/*
função para calcular E sob circunstâncias:  GCD(N, e) = 1 e 1<E<N se
cada teste E for principal, lá será muito menos laços e significando menos
cálculos
*/
function tofindE($pi) {
     global $primos, $maxprimos;
     $great = 0;
     $cc = mt_rand (0,$maxprimos);
     $startcc = $cc;
     while ($cc >= 0) {
          $se = $primos[$cc];
          $great = GCD($se,$pi);
          $cc--;
          if ($great == 1) break;
     }
     if ($great == 0) {
          $cc = $startcc + 1;
          while ($cc <= $maxprimos) {
               $se = $primos[$cc];
               $great = GCD($se,$pi);
               $cc++;
               if ($great == 1) break;
          }
     }
     return $se;
}

/*
*  ENCRYPT retornos da função
* X = M^E (mod N)
* Cada letra na mensagem é representada como seu número de código do ASCII
- 30
* 3 letras em cada bloco com o 1 no começo e na extremidade.
* Para o exemplo
* AAA
* tornar-se-á
* 13535351 (A = ASCII 65-30 = 35)
* 4507^2 = 20313049
* Isto significa que:
* 1. Modulo N a vontade seja sempre < 19999991
* 2. Letras > ASCII 128 obrigação para não ocorrer na mensagem correta do
texto
```

```
*/

function rsa_encrypt ($m, $e, $n) {
     $asci = array ();
     $asc2 = array ();

     for ($i=0; $i < strlen($m); $i+=3) {
          $tmpasc2=" ";
          for ($h=0; $h<3; $h++) {
               if ($i+$h < strlen($m)) {
                    $letra = substr ($m, $i+$h, 1);
                    $tmpstr = ord (substr ($m, $i+$h, 1)) - 30;
                    if (strlen($tmpstr) < 2) {
                         $tmpstr ="0".$tmpstr;
                    }
               } else {
                    break;
               }

               //imprime resultado
               $tmpasc2 = $tmpstr + 30;
               array_push($asc2, $tmpasc2);
          }

     }
     for ($i=0; $i < strlen($m); $i+=3) {
          $tmpasci="1";
          for ($h=0; $h<3; $h++) {
               if ($i+$h < strlen($m)) {
                    $tmpstr = ord (substr ($m, $i+$h, 1)) - 30;
                    if (strlen($tmpstr) < 2) {
                         $tmpstr ="0".$tmpstr;
                    }
               } else {
                    break;
               }
               $tmpasci .=$tmpstr;
          }
          array_push($asci, $tmpasci."1");
     }
     //E o número do ach crifrado então usando a fórmula de RSA:  mod N do
^E do bloco
     for ($k=0; $k < count ($asci); $k++) {
          $resultmod = powmod($asci[$k], $e, $n);

          //imprime resultado
          $coded .= $resultmod." ";
     }

     for ($k2=0; $k2 < count ($asc2); $k2++) {
          $resultmod2 = powmod($asc2[$k2], $e, $n);
          $trs = chr($asc2[$k2]);

          //imprime resultado
          echo "<div align=\"justify\" class=\"texto\">Letra <strong>". $trs
."</strong> em decimal = <strong>";
          echo $asc2[$k2];
          echo "</strong> e na tabela ASCII = <strong>";
          echo bin2hex($trs);
          echo "</strong>";
          echo "<BR>";
```

```
          echo "Cifrando: <strong>". $asc2[$k2]. " ^ ". $e ." (mod ". $n .")
= ". $resultmod2. "</strong>";
          echo "<BR><BR>";
          $coded2 .= $resultmod2." ";
     }

     echo "<br><br><strong>Texto crifrado:</strong> <br><br>";
     echo $coded2;
     echo "</div>";
     $codificado = array();
     $codificado[0] = trim($coded);
     $codificado[1] = trim($coded2);
     return $codificado;
}

/*Método para a exponenciação */
function powmod ($base, $exp, $modulus) {
     $accum = 1;
     $i = 0;
     $basepow2 = $base;
     while (($exp >> $i)>0) {
          if ((($exp >> $i) & 1) == 1) {
               $accum = mo(($accum * $basepow2) , $modulus);
          }
          $basepow2 = mo(($basepow2 * $basepow2) , $modulus);
          $i++;
     }
     return $accum;
}

/*
ENCRYPT retornos da função M = X^D (mod N)
*/
function rsa_decrypt ($c, $d, $n, $e) {
     //Descarte os espaços em branco do texto criptografado e armazenam-no
em uma disposição
     $decryptarray = split(" ", $c);
     for ($u=0; $u < count ($decryptarray); $u++) {
          if ($decryptarray[$u] == "") {
               array_splice($decryptarray, $u, 1);
          }
     }
     //Cada número descriptografado usando a fórmula de RSA:  modificação N
do ^D do bloco
     for ($u=0; $u < count($decryptarray); $u++) {
          $resultmod = powmod($decryptarray[$u], $d, $n);
          //remova os dígitos '1' conduzindo e arrastando
          $deencrypt.= substr ($resultmod,1,strlen($resultmod)-2);
     }
     //Cada número de código do ASCII + 30 na mensagem é representado como
sua letra
     for ($u=0; $u < strlen($deencrypt); $u+=2) {
          $tmp22 = substr ($deencrypt, $u, 2);
          $resultd .= chr(substr ($deencrypt, $u, 2) + 30);
          $resultado = chr(substr ($deencrypt, $u, 2) + 30);

          $ttt = substr ($deencrypt, $u, 2);
          $bases = $ttt + 30;
          $res = powmod($bases, $e, $n);
          //imprime resultado
          echo "<div align=\"justify\" class=\"texto\"><BR>";
```

```
            echo "Descriptografando: ". $res. " ^ ". $d ." (modulo ". $n .") =
Valor decimal da letra ";
            echo $tmp22 + 30;
            echo "<BR>";
            //echo $tmp22 + 30;
            echo "<br><strong>Letra resultante: ";
            echo $resultado;
            echo "</strong><br></div>";
        }
        return $resultd;
}
?>
```

GERADO.PHP - Arquivo de geração de números primos

```
<?
mt_srand((double)microtime()*1000000);

$primos = array (4507, 4513, 4517, 4519, 4523, 4547, 4549, 4561, 4567,
4583, 4591, 4597, 4603, 4621, 4637, 4639, 4643, 4649, 4651, 4657, 4663,
4673, 4679, 4691, 4703, 4721, 4723, 4729, 4733, 4751, 4759, 4783, 4787,
4789, 4793, 4799, 4801, 4813, 4817, 4831, 4861, 4871, 4877, 4889, 4903,
4909, 4919, 4931, 4933, 4937, 4943, 4951, 4957, 4967, 4969, 4973, 4987,
4993, 4999, 5003, 5009, 5011, 5021, 5023, 5039, 5051, 5059, 5077, 5081,
5087, 5099, 5101, 5107, 5113, 5119, 5147, 5153, 5167, 5171, 5179, 5189,
5197, 5209, 5227, 5231, 5233, 5237, 5261, 5273, 5279, 5281, 5297, 5303,
5309, 5323, 5333, 5347, 5351, 5381, 5387, 5393, 5399, 5407, 5413, 5417,
5419, 5431, 5437, 5441, 5443, 5449, 5471, 5477, 5479, 5483, 5501, 5503,
5507, 5519, 5521, 5527, 5531, 5557, 5563, 5569, 5573, 5581, 5591, 5623,
5639, 5641, 5647, 5651, 5653, 5657, 5659, 5669, 5683, 5689, 5693, 5701,
5711, 5717, 5737, 5741, 5743, 5749, 5779, 5783, 5791, 5801, 5807, 5813,
5821, 5827, 5839, 5843, 5849, 5851, 5857, 5861, 5867, 5869, 5879, 5881,
5897, 5903, 5923, 5927, 5939, 5953, 5981, 5987, 6007, 6011, 6029, 6037,
6043, 6047, 6053, 6067, 6073, 6079, 6089, 6091, 6101, 6113, 6121, 6131,
6133, 6143, 6151, 6163, 6173, 6197, 6199, 6203, 6211, 6217, 6221, 6229,
6247, 6257, 6263, 6269, 6271, 6277, 6287, 6299, 6301, 6311, 6317, 6323,
6329, 6337, 6343, 6353, 6359, 6361, 6367, 6373, 6379, 6389, 6397, 6421,
6427, 6449, 6451, 6469, 6473, 6481, 6491, 6521, 6529, 6547, 6551, 6553,
6563, 6569, 6571, 6577, 6581, 6599, 6607, 6619, 6637, 6653, 6659, 6661,
6673, 6679, 6689, 6691, 6701, 6703, 6709, 6719, 6733, 6737, 6761, 6763,
6779, 6781, 6791, 6793, 6803, 6823, 6827, 6829, 6833, 6841, 6857, 6863,
6869, 6871, 6883, 6899, 6907, 6911, 6917, 6947, 6949, 6959, 6961, 6967,
6971, 6977, 6983, 6991, 6997, 7001, 7013, 7019, 7027, 7039, 7043, 7057,
7069, 7079, 7103, 7109, 7121, 7127, 7129, 7151, 7159, 7177, 7187, 7193,
7207, 7211, 7213, 7219, 7229, 7237, 7243, 7247, 7253, 7283, 7297, 7307,
7309, 7321, 7331, 7333, 7349, 7351, 7369, 7393, 7411, 7417, 7433, 7451,
7457, 7459, 7477, 7481, 7487, 7489, 7499, 7507, 7517, 7523, 7529, 7537,
7541, 7547, 7549, 7559, 7561, 7573, 7577, 7583, 7589, 7591, 7603, 7607,
7621, 7639, 7643, 7649, 7669, 7673, 7681, 7687, 7691, 7699, 7703, 7717,
7723, 7727, 7741, 7753, 7757, 7759, 7789, 7793, 7817, 7823, 7829, 7841,
7853, 7867, 7873, 7877, 7879, 7883, 7901, 7907, 7919, 7927, 7933, 7937,
7949, 7951, 7963, 7993, 8009, 8011, 8017, 8039, 8053, 8059, 8069, 8081,
8087, 8089, 8093, 8101, 8111, 8117, 8123, 8147, 8161, 8167, 8171, 8179,
8191, 8209, 8219, 8221, 8231, 8233, 8237, 8243, 8263, 8269, 8273, 8287,
8291, 8293, 8297, 8311, 8317, 8329, 8353, 8363, 8369, 8377, 8387, 8389,
8419, 8423, 8429, 8431, 8443, 8447, 8461, 8467, 8501, 8513, 8521, 8527,
8537, 8539, 8543, 8563, 8573, 8581, 8597, 8599, 8609, 8623, 8627, 8629,
8641, 8647, 8663, 8669, 8677, 8681, 8689, 8693, 8699, 8707, 8713, 8719,
8731, 8737, 8741, 8747, 8753, 8761, 8779, 8783, 8803, 8807, 8819, 8821,
8831, 8837, 8839, 8849, 8861, 8863, 8867, 8887, 8893, 8923, 8929, 8933,
8941, 8951, 8963, 8969, 8971, 8999, 9001, 9007, 9011, 9013, 9029, 9041,
9043, 9049, 9059, 9067, 9091, 9103, 9109, 9127, 9133, 9137, 9151, 9157,
9161, 9173, 9181, 9187, 9199, 9203, 9209, 9221, 9227, 9239, 9241, 9257,
9277, 9281, 9283, 9293, 9311, 9319, 9323, 9337, 9341, 9343, 9349, 9371,
9377, 9391, 9397, 9403, 9413, 9419, 9421, 9431, 9433, 9437, 9439, 9461,
9463, 9467, 9473, 9479, 9491, 9497, 9511, 9521, 9533);

$maxprimos = count($primos) - 1;

global $primos, $maxprimos;
$p = $primos[mt_rand(0, $maxprimos)];
while (empty($q) || ($p==$q)) $q = $primos[mt_rand(0, $maxprimos)];
?>
```

INDEX.PHP – Arquivo de estrutura de todo o sistema

```
<?php
include("rsa.php");
$fases = $_POST['fase'];
switch ($fases) {
     case 0:
       $inclusao = ("formulario.php");
        break;
     case 1:
       $inclusao = ("formulario2.php");
        break;
     case 2:
       $inclusao = ("formulario3.php");
        break;
     case 3:
       $inclusao = ("formulario4.php");
        break;
}

?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title>Cripto-Aula RSA</title>
<style type="text/css">
<!--
body {
     margin-left: 0px;
     margin-top: 0px;
     margin-right: 0px;
     margin-bottom: 0px;
}
.borda {
     border-top-width: 1px;
     border-right-width: 1px;
     border-bottom-width: 1px;
     border-left-width: 1px;
     border-top-style: none;
     border-right-style: solid;
     border-bottom-style: none;
     border-left-style: solid;
     border-top-color: #CCCCCC;
     border-right-color: #CCCCCC;
     border-bottom-color: #CCCCCC;
     border-left-color: #CCCCCC;
}
.texto {
     font-family: Arial, Helvetica, sans-serif;
     font-size: 11px;
     font-weight: normal;
     color: #000000;
     text-decoration: none;
}
.textotitu {
     font-family: Arial, Helvetica, sans-serif;
     font-size: 11px;
     font-weight: bold;
```

```
     color: #000000;
     text-decoration: none;
}
.textog {
     font-family: Arial, Helvetica, sans-serif;
     font-size: 14px;
     font-weight: normal;
     color: #000000;
     text-decoration: none;
}
-->
</style>
</head>

<body>
<table width="764" border="0" align="center" cellpadding="0"
cellspacing="0" class="borda">
  <tr>
    <td colspan="2"><img src="imagens/blackline.gif" width="764"
height="3"></td>
  </tr>
  <tr>
    <td width="162"><img src="imagens/rsa_logo.gif" width="162"
height="91"></td>
    <td width="602"><img src="imagens/confidence.jpg" width="602"
height="91"></td>
  </tr>
  <tr>
    <td colspan="2"><img src="imagens/blackline.gif" width="764"
height="3"></td>
  </tr>
  <tr>
    <td colspan="2"><img src="" width="5" height="5" alt=""></td>
  </tr>
</table>
<table width="764" border="0" align="center" cellpadding="0"
cellspacing="0" class="borda">
  <tr>
    <td width="162" valign="top" bgcolor="#ebebeb"><div align="justify">
      <table width="100%"  border="00" cellspacing="0" cellpadding="6">
        <tr>
          <td> </td>
        </tr>
      </table>
    </div></td>
    <td width="602" valign="top"><div align="center">
      <? include($inclusao); ?>
    </div></td>
  </tr>
  <tr>
    <td colspan="2"><img src="imagens/blackline.gif" width="764"
height="3"></td>
  </tr>
</table>
</body>
</html>
```

FORMULARIO.PHP – Arquivo de digitação do texto e números primos

```
<? require_once("gerado.php"); ?>
<script type="text/JavaScript">
<!--
function abrir_janela_popup(theURL,winName,features) { //v2.0
  window.open(theURL,winName,features);
}
function retorno() {
document.primos.primo1.value='<? echo $p;?>';
document.primos.primo2.value='<? echo $q;?>';
}
//-->
</script>
<form name="primos" method="post" action="index.php">
  <table width="100%"  border="0" align="left" cellpadding="6"
cellspacing="0">
    <tr>
      <td class="texto"><div align="justify">A fim de obter as duas chaves,
uma para cifrar e outra para decifrar, devemos  proceder como segue:<br />
          <br />
          Escolhe-se de forma aleat&oacute;ria dois primos de valores
altos, P e Q que ser&atilde;o utilizados para gerar as chaves. Escolhemos
os primos porque atualmente &eacute; baseado na dificuldade de fatorar
n&uacute;meros primos  grandes (Teoria dos N&uacute;meros).<br />
          <br />
      Para maior seguran&ccedil;a e funcionamento deve-se escolher dois
primos  com o  mesmo n&uacute;mero de bits, neste exemplo trabalharemos com
n&uacute;meros primos de 4 digitos.</div></td>
    </tr>
    <tr>
      <td><table width="100%"  border="0" align="left" cellpadding="3"
cellspacing="0">
        <tr >
          <td colspan="2" class="texto"><div
align="left"><strong>Texto:</strong></div></td>
        </tr>
        <tr >
          <td colspan="2"><div align="left">
              <table width="100%" border="0" cellspacing="0"
cellpadding="6">
                <tr>
                  <td width="54%"><textarea name="msn" cols="60" rows="5"
class="texto"></textarea></td>
                  <td width="46%" class="texto"><div align="center">Caso
n&atilde;o saiba 2 n&uacute;meros primos com 4 digitos clique no
bot&atilde;o abaixo para gerar dois n&uacute;meros primos<br />
                      <br />
                      <img src="imagens/gerador2.gif" width="104"
height="22" onclick="retorno()" /></div></td>
                </tr>
              </table>
          </div></td>
        </tr>
        <tr >
          <td width="24%" class="texto"><div align="left"><strong>Primeiro
n&uacute;mero primo:</strong></div></td>
          <td width="76%"><div align="left">
              <input name="primo1" type="text" class="texto" size="25"
maxlength="5" />
```

```
          </div></td>
        </tr>
        <tr >
          <td class="texto"><div align="left"><strong>Segundo n&uacute;mero
primo:</strong></div></td>
          <td><div align="left">
              <input name="primo2" type="text" class="texto" size="25"
maxlength="5" />
            </div>
              <input name="fase" type="hidden" value="1" /></td>
        </tr>
        <tr >
          <td colspan="2"><div align="left">
              <input name="Submit" type="submit" class="textotitu"
value="Criptografar" />
          </div></td>
        </tr>
      </table></td>
    </tr>
  </table>
</form>
```

FORMULARIO2.PHP – Arquivo de geração de chaves

```
<?
$xv = $_POST['primo1'];
$xv2 = $_POST['primo2'];
$msn = $_POST['msn'];
if ($xv == "") {
$chaves = generate_chaves();
} else {
$msn = $_POST['msn'];
$chaves = generate_chaves2($xv,$xv2);
}
?>
<form name="form1" method="post" action="index.php">
  <table width="100%" border="0" cellspacing="0" cellpadding="6">
    <tr>
      <td><div align="justify" class="texto">Agora vamos explicar como se
acha o m&oacute;dulo, as chaves de criptografia e descriptografia que
ser&atilde;o usados no processo de cifragem e descifragem da mensagem.
lembrando-se que:<br />
          <br />
          <? echo "
              P = <strong>$chaves[3]</strong><br><br>
              Q = <strong>$chaves[4]</strong>";
              ?><br />
          <br />
          <ol><li>Para
            achamos o m&oacute;dulo multiplicando os dois n&uacute;meros
primos (PxQ) obtendo-se: <br />
            <br />
            <strong><? echo "$chaves[3] x $chaves[4] = <b>$chaves[0]</b>";
?></strong><br />
            <br />
            Assim o
            o m&oacute;dulo &eacute;: <strong><? echo "<b>$chaves[0]</b>";
?></strong><br />
            <br />
          </li>
          <li>Seguindo acharemos a chave <strong>E</strong> sendo uma das
chaves para cifragem do texto ou seja uma chave p&uacute;blica.<br />
            <br />
            <span>Para achar a chave <strong>E</strong> necessitamos de um
outro n&uacute;mero primo que seja primo relativo a formula: <br />
              <br />
              (p-1) x (q-1) ficando: <br />
              <br />
              (<?echo "$chaves[3]"; ?> - 1) * (<?echo "$chaves[4]"; ?> - 1)
=
              <?$relativo = (($chaves[3] -1)*($chaves[4]-1));
echo "<b>$relativo</b>"; ?>
              <br />
              <br />
              ou seja o MDC (M&aacute;ximo Divisor Comum) entre eles seja 1
depois dos calculos achamos<br />
              <br />
              <?echo "<b>E = $chaves[1]</b> => a chave p&uacute;blica";
?><br />
              <br />
            </span></li>
          <li>Finalizando nossa parte de obten&ccedil;&atilde;o das chaves
```

```
acharemos a chave privada <strong>D</strong><br />
            <br />
            Para achar a chave D utilizaremos a f&oacute;rmula:<br />
            <br />
            E x D = 1 (mod (P-1) x (Q-1)) que ficar&aacute; assim:<br />
            <br />
            <?echo "$chaves[1]"; ?> x D = 1 (mod (<?echo "$chaves[3]"; ?> -
1) * (<?echo "$chaves[4]"; ?> - 1))<br />
            <br />
            <? echo "<b>D = $chaves[2]</b> => chave privada<p>"; ?></li>
          </ol>
          Nosso pr&oacute;ximo passo &eacute; a cifragem da mensagem
utilizando as chaves abaixo:<br />
          <br />
          <?
              echo "
              P = $chaves[3]<br>
              Q = $chaves[4]<br>
              <b>N = $chaves[0]</b> - modulo<br>
              <b>E = $chaves[1]</b> - chave p&uacute;blica<br>
              <b>D = $chaves[2]</b> - chave privada<p>"; ?>
          Clique no bot&atilde;o Criptografar e siga em frente.<br />
          <br />
      </div></td>
    </tr>
    <tr>
      <td><table width="100%" border="0" cellpadding="6" cellspacing="0">
        <tr>
          <td><div align="left">
              <input type="hidden" name="mod" value="<? echo $chaves[0];?>"
/>
              <input type="hidden" name="pub" value="<? echo $chaves[1];?>"
/>
              <input type="hidden" name="priv" value="<? echo
$chaves[2];?>" />
              <input type="hidden" name="msn" value="<? echo $msn;?>" />
              <input type="hidden" name="fase" value="2" />
              <input type="submit" name="Submit" value="Criptografar" />
          </div></td>
          </tr>
      </table></td>
    </tr>
  </table>
</form>
```

FORMULARIO3.PHP – Arquivo de arquivo de criptografia

```
<?
$mod = $_POST['mod'];
$pub = $_POST['pub'];
$priv = $_POST['priv'];
$msn = $_POST['msn'];

?>
<form name="form1" method="post" action="index.php">
  <table width="100%" height="150"  border="0" cellpadding="6"
cellspacing="0">
    <tr class="style5">
      <td width="100%" class="texto"><p align="justify"><span
class="textog"><strong>Lembrete:</strong></span><br />        <br />
        A aritm&eacute;tica modular, apesar de muito simples, costuma dar
n&oacute; na cabe&ccedil;a da gente. Como &eacute; que 10 + 4 pode ser 2 no
        m&oacute;dulo 12? Este c&aacute;lculo pode ser feito de
cabe&ccedil;a se lembrarmos do rel&oacute;gio: 10 + 4 = 14 horas ou 2 horas
da tarde.<br />
        <br />
        Quando lidamos com um n&uacute;mero finito de inteiros, os
resultados obtidos nas opera&ccedil;&otilde;es de soma,
subtra&ccedil;&atilde;o, multiplica&ccedil;&atilde;o e
        divis&atilde;o precisam ser ajustados para que permane&ccedil;am
dentro do conjunto dos inteiros dispon&iacute;veis. Estas
opera&ccedil;&otilde;es
        funcionam como o ponteiro das horas do rel&oacute;gio e, por isto,
esta aritm&eacute;tica tamb&eacute;m &eacute; conhecida como circular.<br
/>
        <br />
        Na soma, o ponteiro &eacute; deslocado no sentido hor&aacute;rio
(para frente) e, quando alcan&ccedil;ar 11 horas, a pr&oacute;xima
ser&aacute; 12 ou
        Zero hora. O ponteiro volta para a estaca zero porque o
m&oacute;dulo 12 s&oacute; possui os inteiros de 0 a 11 para expressar
valores.
        Al&eacute;m disso, toda vez que somarmos um m&uacute;ltiplo de 12 a
qualquer hora, o ponteiro n&atilde;o muda de lugar.<br />
        <br />
        <strong>Abaixo segue exemplo de uma calculadora modular:
</strong></p>      </td>
    </tr>
    <tr class="style5">
      <td class="texto"><div align="left">
        <applet code="Modular.class" codebase = "modulo" width="300"
height="82">
        </applet>
      </div></td>
    </tr>
    <tr class="style5">
      <td class="texto"><p align="justify">Para cifragem &eacute;
necess&aacute;rio o conhecimanto da tabela ASCII de cada letra, o sistema
autom&aacute;ticamente retirar&aacute; os valores necess&aacute;rios em
formato decimal baseado na tabela e far&aacute; o calculo da seguinte
f&oacute;rmula:<br />
        <br />
        <img width="99" height="24"
src="imagens/formulario3_clip_image002.gif" /><br />
        <br />
```

```
        Lembando-se que o sinal ^ utilizado abaixo siguinifica
expon&ecirc;ncia&ccedil;&atilde;o ou seja:<br />
        <br />
        Se tenho <strong>88 ^ 2 (mod 50) </strong>     l&ecirc;-se
<strong>88 elevado a 2 no m&oacute;dulo 50 </strong>.<br />
        <br />
      </p>
      </td>
    </tr>
    <tr class="style5">
      <td class="textog"><div align="left"><strong>Criptografando o texto:
</strong></div></td>
    </tr>
    <tr class="style5">
      <td><div align="justify" class="texto">
        <blockquote>
          <blockquote>
            <blockquote>
              <blockquote>
                <p>
                  <?
                      $encoded = array();
                      $encoded = rsa_encrypt ($msn, $pub, $mod); ?>
                </p>
              </blockquote>
            </blockquote>
          </blockquote>
        </blockquote>
      </div>      </td>
    </tr>
    <tr class="style5">
      <td><div align="center" class="style5">
          <input type="hidden" name="mod" value="<? echo $mod;?>">
           <input type="hidden" name="pub" value="<? echo $pub;?>">
           <input type="hidden" name="priv" value="<? echo $priv;?>">
          <input type="hidden" name="msn" value="<? echo $encoded[0];?>">
           <input type="hidden" name="msn1" value="<? echo $encoded[1];?>">
          <input type="hidden" name="msn2" value="<? echo $msn;?>">
           <input type="hidden" name="fase" value="3">
          <input type="submit" name="Submit" value="Descriptografar">
      </div></td>
    </tr>
  </table>
</form>
```

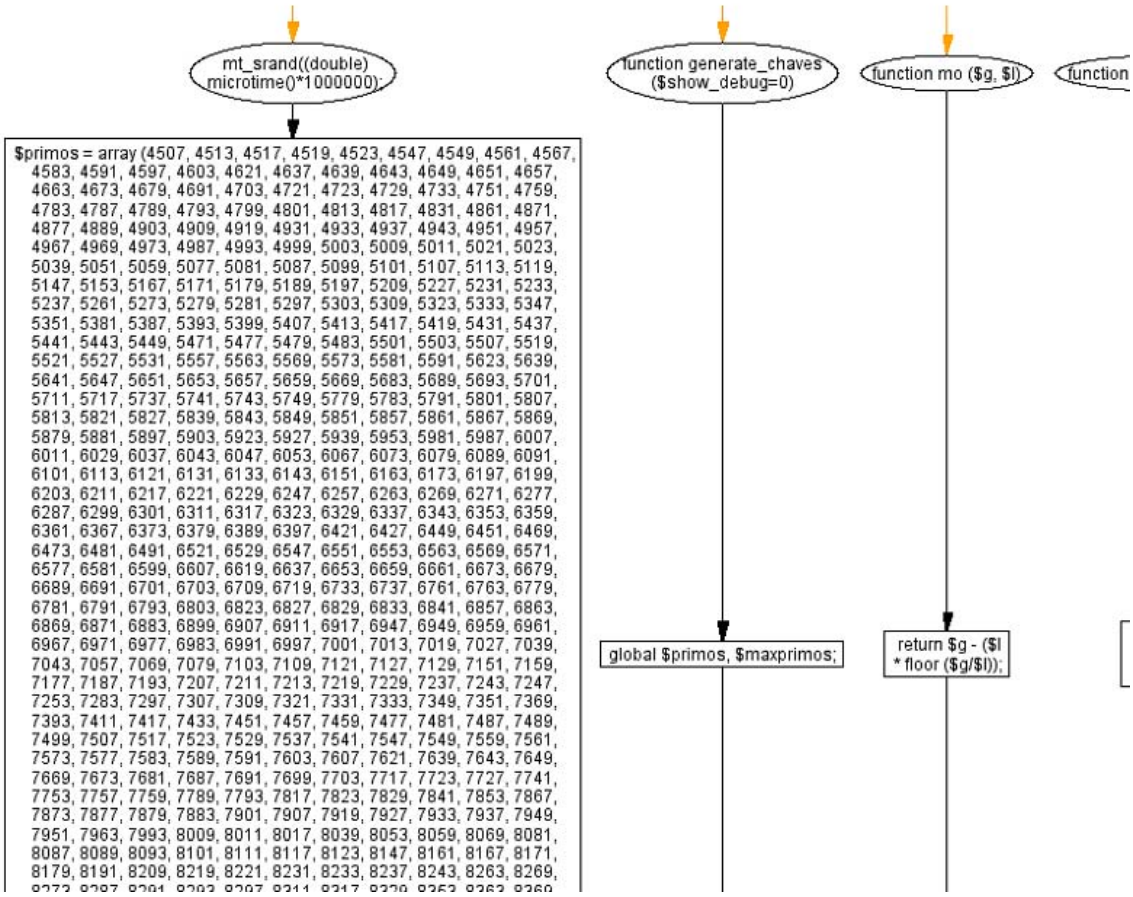
FORMULARIO4.PHP – Arquivo de arquivo de descriptografia

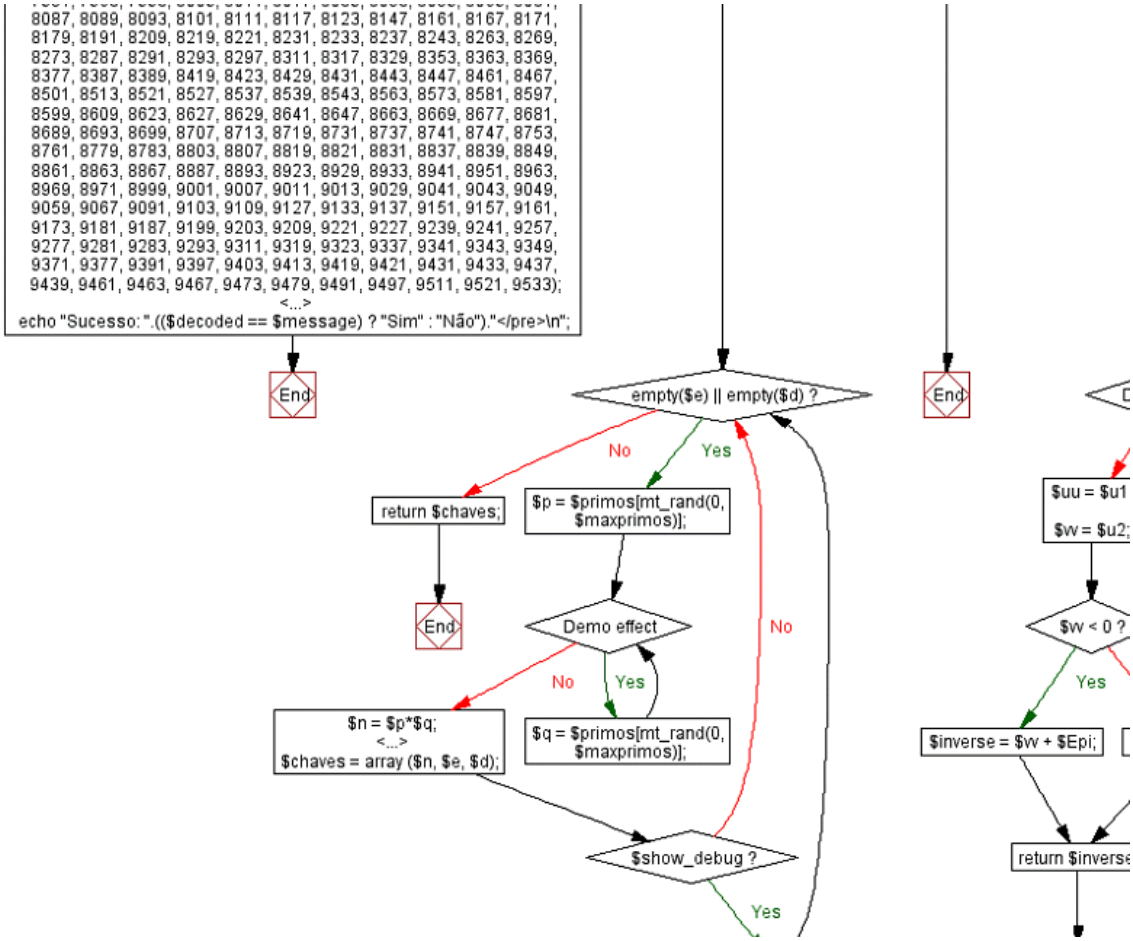
```
<?
$mod = $_POST['mod'];
$pub = $_POST['pub'];
$priv = $_POST['priv'];
$msn = $_POST['msn'];
$msn1 = $_POST['msn1'];
$msn2 = $_POST['msn2'];

?>
<table width="100%" height="150" border="0" cellpadding="6"
cellspacing="0">
          <tr class="texto">
            <td width="100%"><p align="justify">Em nosso ultimo passo
seguimos com a descriptografiada mensagem original:<br />
              <br />
              <strong><?php echo $msn1; ?></strong><br />
              <br />
            utilizando a formula:</p>
            <p align="justify"><img src="imagens/formula_desci.gif"
width="112" height="32" /></p>
            <p align="justify">onde <strong>M</strong> &eacute; a mensagem
original a ser obtida, <strong>C</strong> &eacute; o texto codificado,
<strong>D</strong> a chave privada e <strong>N</strong> o m&oacute;dulo.
</p></td>
          </tr>
          <tr class="texto">
            <td><? $decoded = rsa_decrypt($msn, $priv, $mod, $pub); ?>
              <strong><br />
            </strong><strong><br />
            </strong></td>
          </tr>
          <tr class="texto">
            <td><div align="justify" class="textog"><strong>Conferencia do
resultado: </strong></div></td>
          </tr>
          <tr class="texto">
            <td><div align="justify"><strong>Texto original:<br />
            </strong>              <br />
              <? echo $msn2; ?><br />
              <br />
            </div></td>
          </tr>
          <tr class="texto">
            <td><div align="justify"><strong>Texto descriptografado:<br />
            </strong>                  <br />
                  <? echo $msn2; ?><br />
                  <br />
            </div></td>
          </tr>
          <tr class="texto">
            <td><div align="justify">

              <strong>Resultado no processo?<br />
              </strong>            <br />
            <? echo (($decoded == $msn2) ? "Sim" : "Não")."</pre>\n";
              ?></div></td>
          </tr>
          <tr class="texto">
```

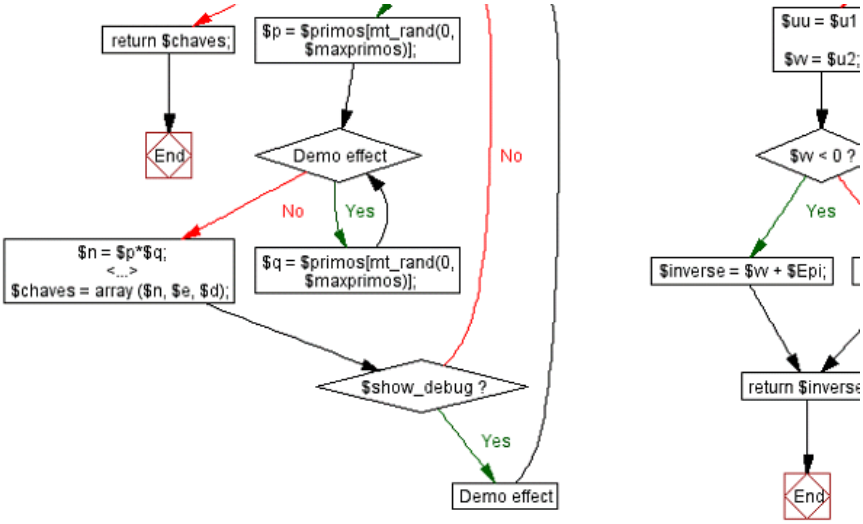
```
            <td><div align="justify" class="style5"><a href="index.php"
class="style5">Reiniciar</a>

            </div></td>
          </tr>
</table>
```

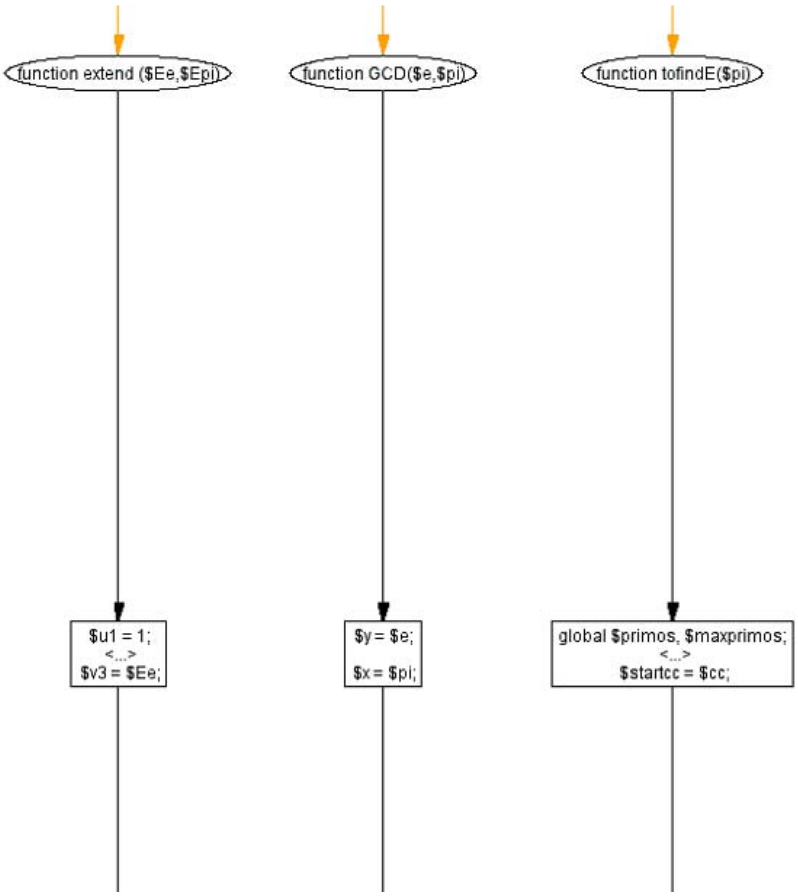
Apêndice B

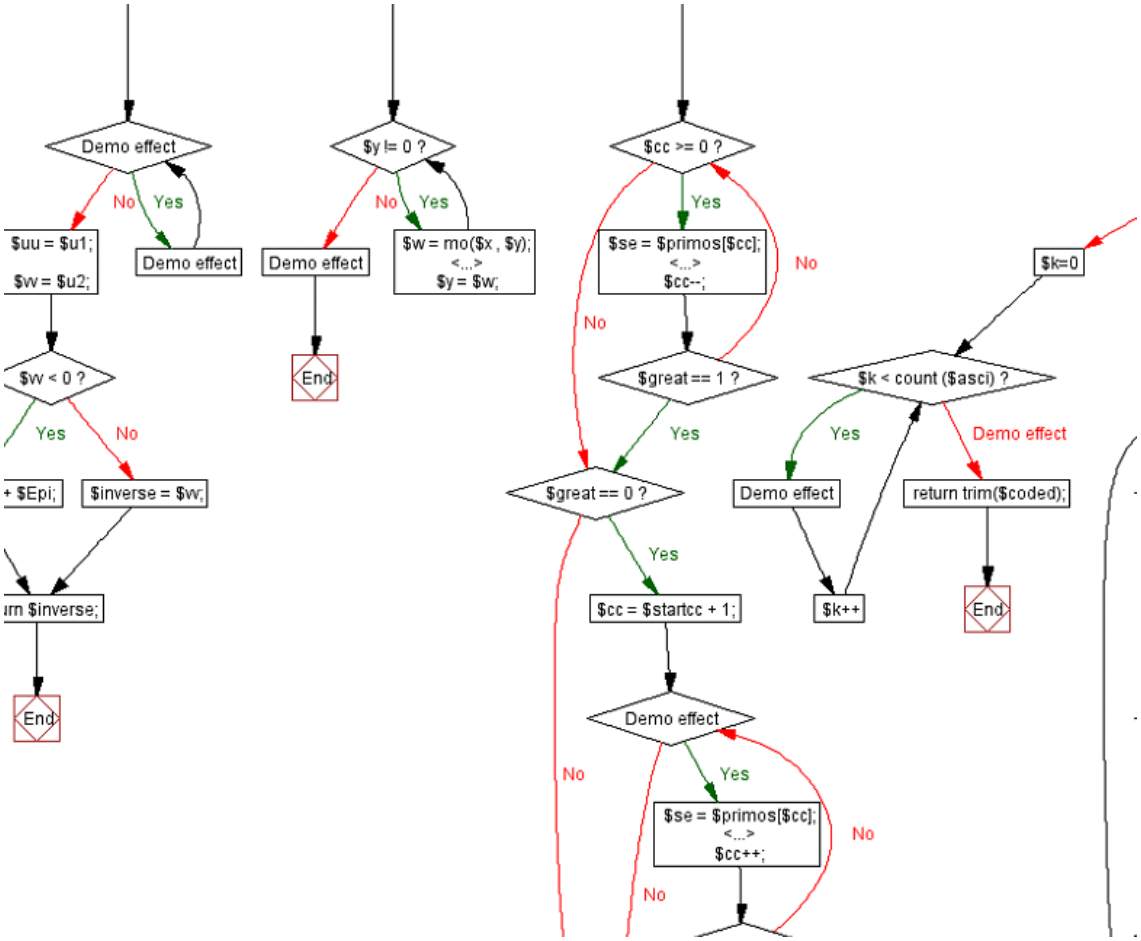
Diagrama de fluxo do sistema completo e expandido

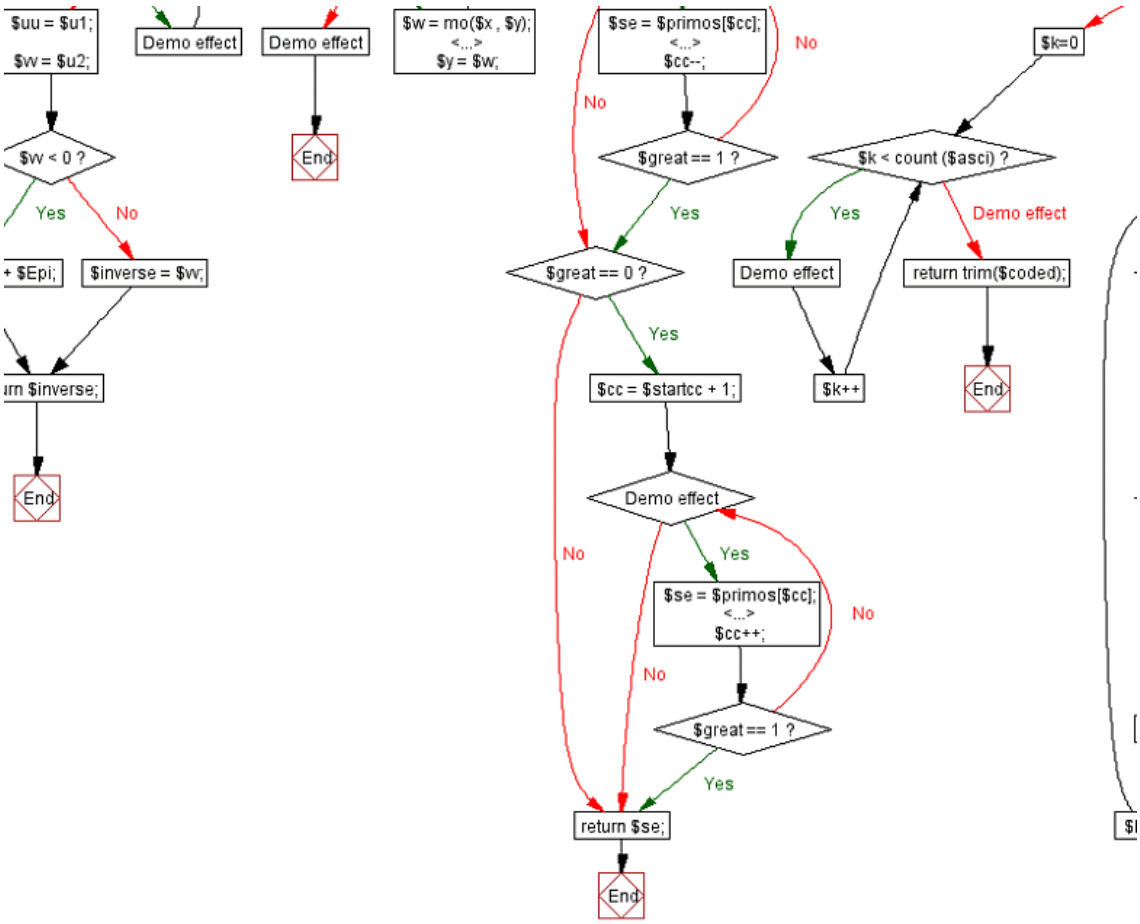


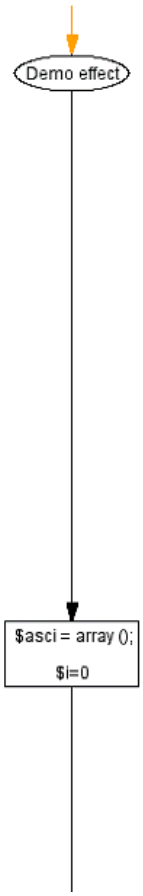

8087, 8089, 8093, 8101, 8111, 8117, 8123, 8147, 8161, 8167, 8171,
8179, 8191, 8209, 8219, 8221, 8231, 8233, 8237, 8243, 8263, 8269,
8273, 8287, 8291, 8293, 8297, 8311, 8317, 8329, 8353, 8363, 8369,
8377, 8387, 8389, 8419, 8423, 8429, 8431, 8443, 8447, 8461, 8467,
8501, 8513, 8521, 8527, 8537, 8539, 8543, 8563, 8573, 8581, 8597,
8599, 8609, 8623, 8627, 8629, 8641, 8647, 8663, 8669, 8677, 8681,
8689, 8693, 8699, 8707, 8713, 8719, 8731, 8737, 8741, 8747, 8753,
8761, 8779, 8783, 8803, 8807, 8819, 8821, 8831, 8837, 8839, 8849,
8861, 8863, 8867, 8887, 8893, 8923, 8929, 8933, 8941, 8951, 8963,
8969, 8971, 8999, 9001, 9007, 9011, 9013, 9029, 9041, 9043, 9049,
9059, 9067, 9091, 9103, 9109, 9127, 9133, 9137, 9151, 9157, 9161,
9173, 9181, 9187, 9199, 9203, 9209, 9221, 9227, 9239, 9241, 9257,
9277, 9281, 9283, 9293, 9311, 9319, 9323, 9337, 9341, 9343, 9349,
9371, 9377, 9391, 9397, 9403, 9413, 9419, 9421, 9431, 9433, 9437,
9439, 9461, 9463, 9467, 9473, 9479, 9491, 9497, 9511, 9521, 9533);
<...>
echo "Sucesso: ".(($decoded == $message) ? "Sim" : "Não")."</pre>\n";
End
empty($e) || empty($d) ?
No
Yes
return $chaves;
$p = $primos[mt_rand(0, $maxprimos)];
End
Demo effect
No
Yes
No
$n = $p*$q;
<...>
$chaves = array ($n, $e, $d);
$q = $primos[mt_rand(0, $maxprimos)];
$show_debug ?
Yes
End
$uu = $u1
$vv = $u2;
$vv < 0 ?
Yes
$inverse = $vv + $Epi;
return $inverse

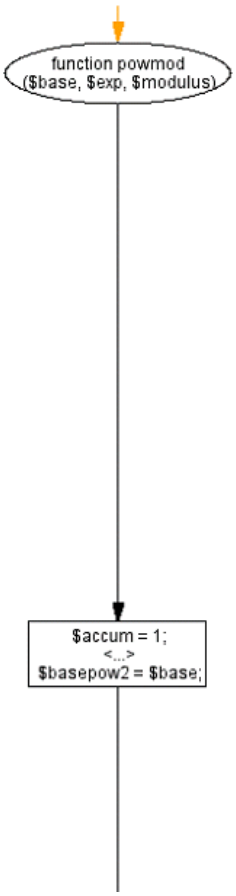

return $chaves;
End
$p = $primos[mt_rand(0, $maxprimos)];
Demo effect
No
Yes
$n = $p*$q;
<...>
$chaves = array ($n, $e, $d);
$q = $primos[mt_rand(0, $maxprimos)];
No
$show_debug ?
Yes
Demo effect
$uu = $u1
$vv = $u2;
$vv < 0 ?
Yes
$inverse = $vv + $Epi;
return $inverse
End

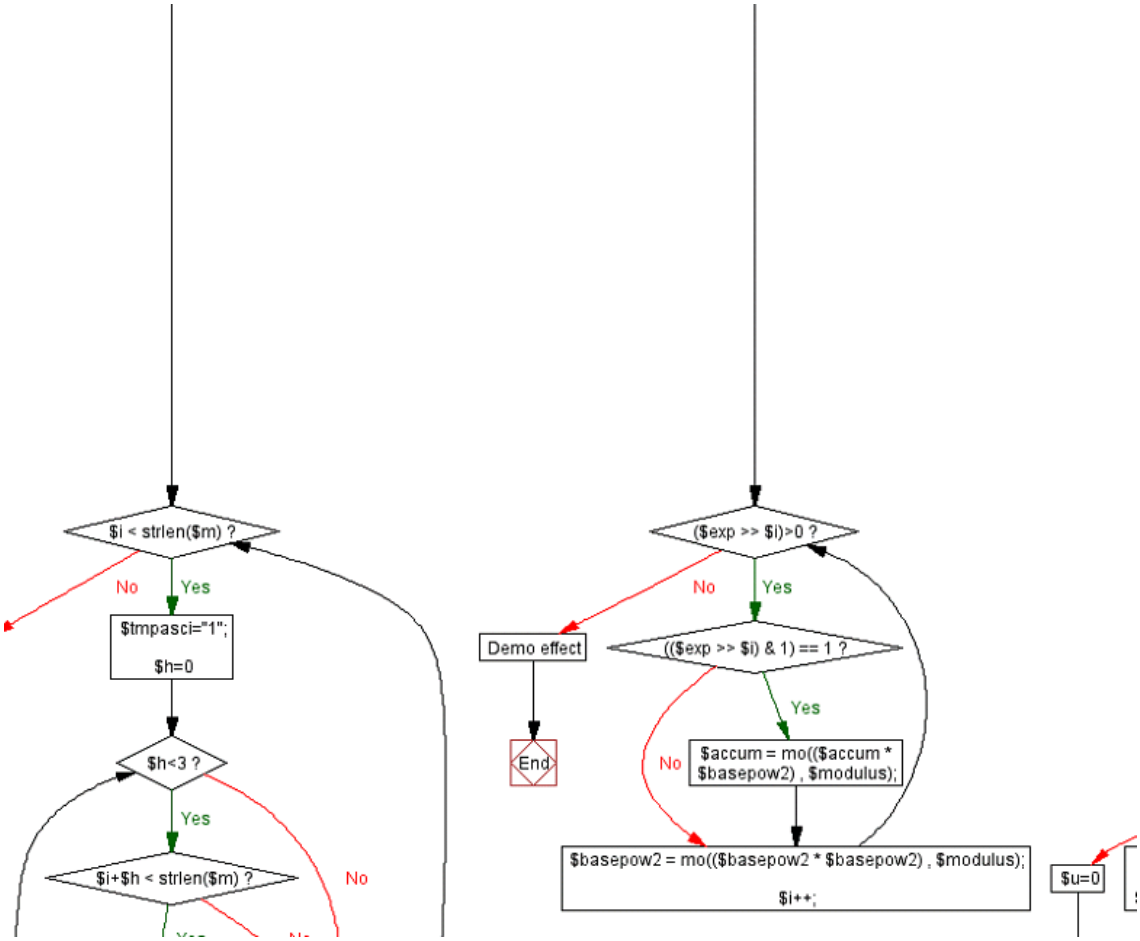

function extend ($Ee,$Epi)
function GCD($e,$pi)
function tofindE($pi)
$u1 = 1;
<...>
$v3 = $Ee;
$y = $e;
$x = $pi;
global $primos, $maxprimos;
<...>
$startcc = $cc;

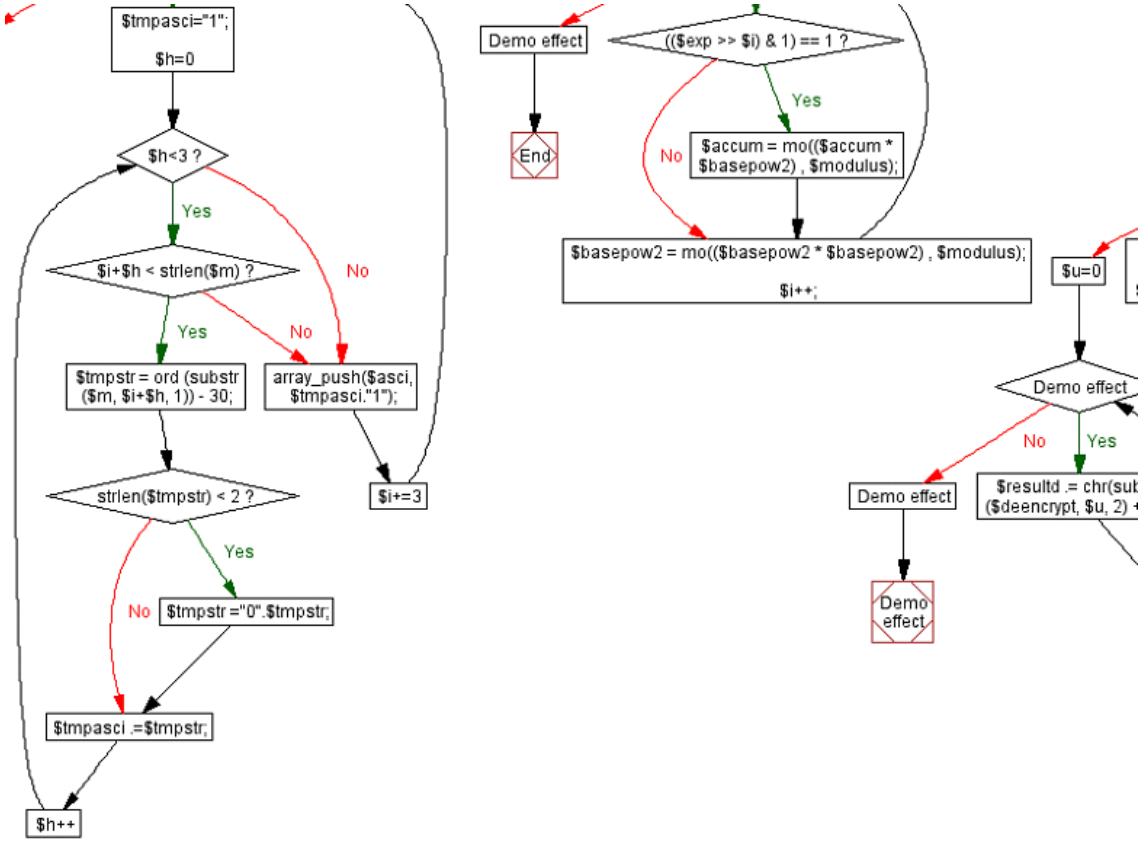

Demo effect
No
Yes
Demo effect
$uu = $u1;
$vv = $u2;
$w < 0 ?
Yes
No
+ $Epi;
$inverse = $w;
rn $inverse;
End
$y != 0 ?
No
Yes
Demo effect
$w = mo($x , $y);
<...>
$y = $w;
End
$cc >= 0 ?
Yes
$se = $primos[$cc];
<...>
$cc--;
No
No
$great == 1 ?
Yes
$great == 0 ?
Yes
$cc = $startcc + 1;
Demo effect
No
Yes
$se = $primos[$cc];
<...>
$cc++;
No
No
$k=0
$k < count ($asci) ?
Yes
Demo effect
Demo effect
return trim($coded);
$k++
End

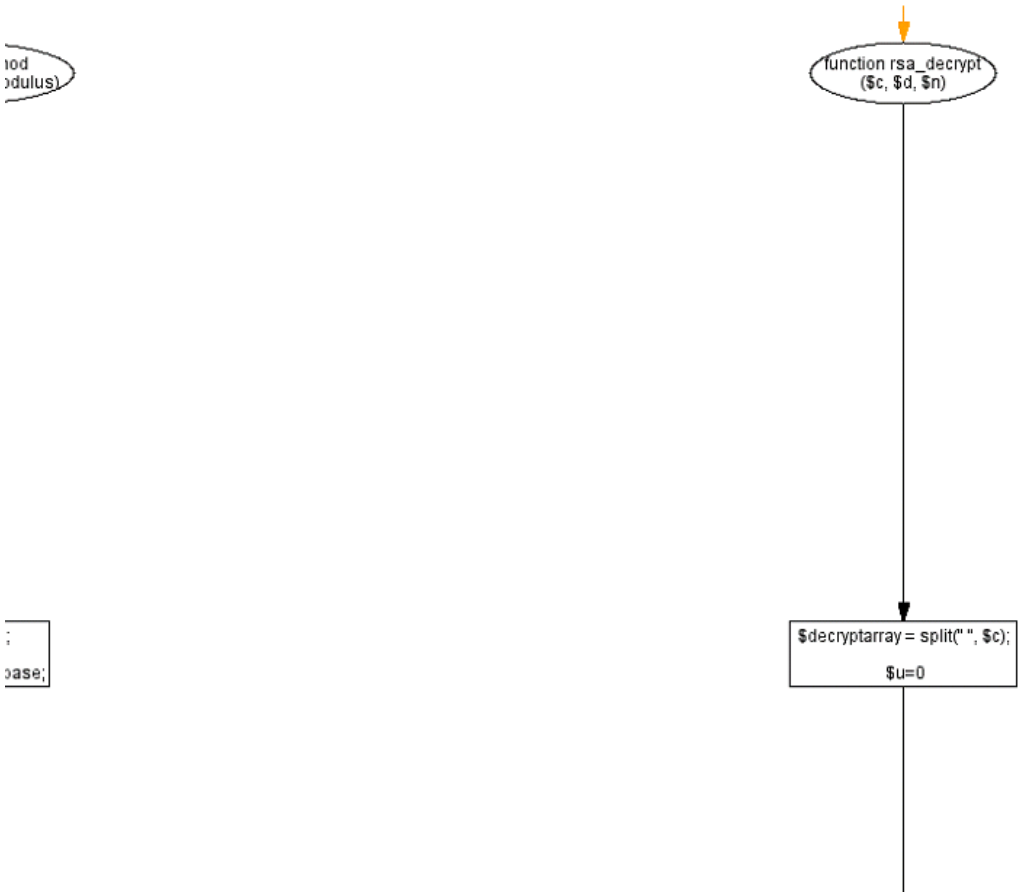

$uu = $u1;
$vv = $u2;
$w < 0 ?
Yes
No
+ $Epi;
$inverse = $w;
rn $inverse;
End
Demo effect
Demo effect
End
$w = mo($x , $y);
<...>
$y = $w;
$se = $primos[$cc];
<...>
$cc--;
No
No
$great == 1 ?
Yes
$great == 0 ?
Yes
$cc = $startcc + 1;
Demo effect
No
Yes
$se = $primos[$cc];
<...>
$cc++;
No
No
$great == 1 ?
Yes
return $se;
End
$k=0
$k < count ($asci) ?
Yes
Demo effect
Demo effect
return trim($coded);
$k++
End

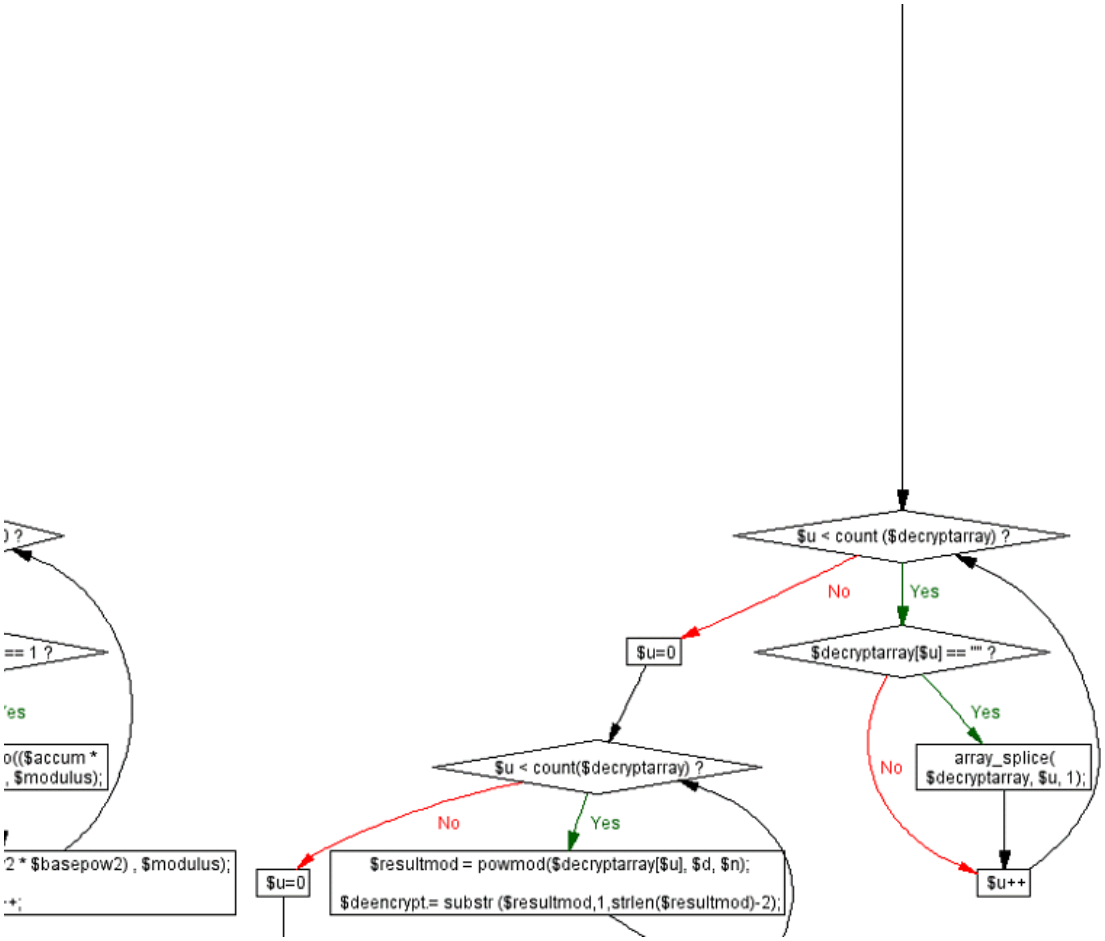

Demo effect
$asci = array ();
$i=0

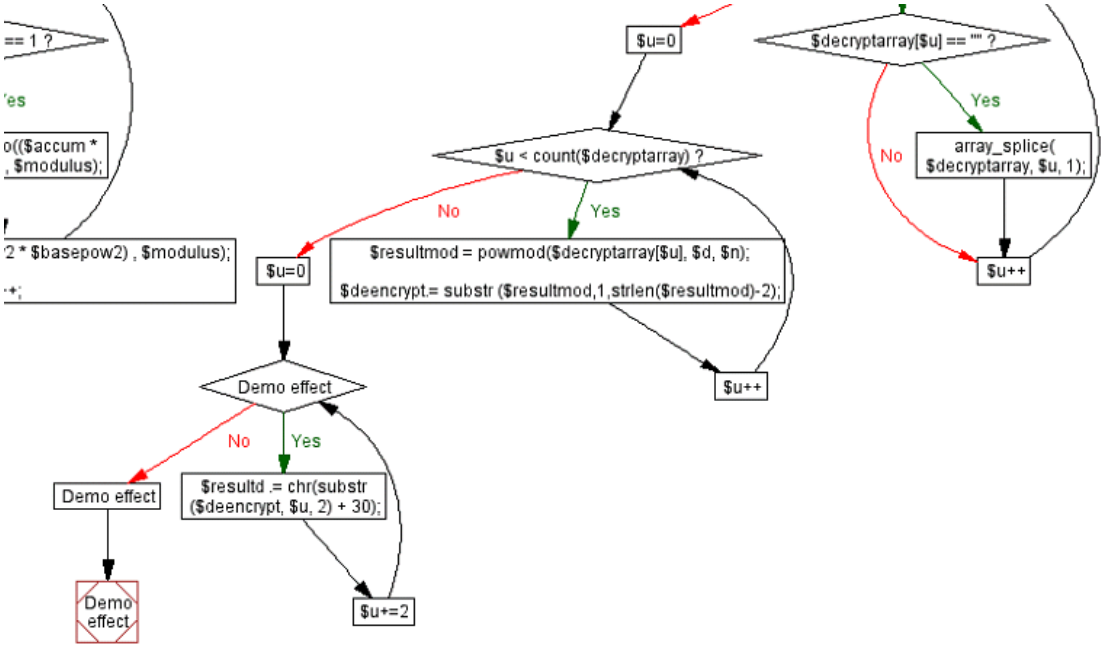

function powmod
($base, $exp, $modulus)
$accum = 1;
<...>
$basepow2 = $base;


$i < strlen($m) ?
No
Yes
$tmpasci="1";
$h=0
$h<3 ?
Yes
No
$i+$h < strlen($m) ?
($exp >> $i)>0 ?
No
Yes
Demo effect
End
((($exp >> $i) & 1) == 1 ?
Yes
No
$accum = mo(($accum * $basepow2) , $modulus);
$basepow2 = mo(($basepow2 * $basepow2) , $modulus);
$i++;
$u=0


$tmpasci="1";
$h=0
$h<3 ?
Yes
No
$i+$h < strlen($m) ?
Yes
No
$tmpstr = ord (substr ($m, $i+$h, 1)) - 30;
array_push($asci, $tmpasci."1");
$i+=3
strlen($tmpstr) < 2 ?
Yes
No
$tmpstr ="0".$tmpstr;
$tmpasci .=$tmpstr;
$h++
Demo effect
End
(($exp >> $i) & 1) == 1 ?
Yes
No
$accum = mo(($accum * $basepow2) , $modulus);
$basepow2 = mo(($basepow2 * $basepow2) , $modulus);
$i++;
$u=0
Demo effect
No
Yes
Demo effect
Demo effect


od
dulus)
function rsa_decrypt
($c, $d, $n)
;
base;
$decryptarray = split(" ", $c);
$u=0


$u < count ($decryptarray) ?
No
Yes
$u=0
$decryptarray[$u] == "" ?
Yes
No
array_splice(
$decryptarray, $u, 1);
$u++
$u < count($decryptarray) ?
No
Yes
$u=0
$resultmod = powmod($decryptarray[$u], $d, $n);
$deencrypt.= substr ($resultmod,1,strlen($resultmod)-2);
== 1 ?
es
o(($accum *
, $modulus);
2 * $basepow2) , $modulus);


== 1 ?
es
o(($accum *
, $modulus);
2 * $basepow2) , $modulus);
$u=0
$u < count($decryptarray) ?
No
Yes
$resultmod = powmod($decryptarray[$u], $d, $n);
$deencrypt.= substr ($resultmod,1,strlen($resultmod)-2);
$u++
$decryptarray[$u] == "" ?
Yes
No
array_splice(
$decryptarray, $u, 1);
$u++
$u=0
Demo effect
No
Yes
Demo effect
$resultd .= chr(substr
($deencrypt, $u, 2) + 30);
Demo
effect
$u+=2

12 ANEXOS

Anexo A

A Tabela ASCII (American Standard Code for Information Interchange) é usada pela maior parte da industria de computadores para a troca de informações. Cada caracter é representado por um código de 8 bits (um byte). Abaixo mostramos a tabela ASCII de 7 bits juntamente com a tabela extendida para 8 bits que incluem os caracteres acentuados.

TABELA ASCII											
Decimal	Hexa	Caracter	Decimal	Hexa	Caracter	Decimal	Hexa	Caracter	Decimal	Hexa	Caracter
0	00	NUL	64	40	@	128	80	€	192	C0	À
1	01	SOH	65	41	A	129	81		193	C1	Á
2	02	STX	66	42	B	130	82	‚	194	C2	Â
3	03	ETX	67	43	C	131	83	*ƒ*	195	C3	Ã
4	04	EOT	68	44	D	132	84	„	196	C4	Ä
5	05	ENQ	69	45	E	133	85	…	197	C5	Å
6	06	ACK	70	46	F	134	86	†	198	C6	Æ
7	07	BEL	71	47	G	135	87	‡	199	C7	Ç
8	08	BS	72	48	H	136	88	ˆ	200	C8	È
9	09	HT	73	49	I	137	89	‰	201	C9	É
10	0A	LF	74	4A	J	138	8A	Š	202	CA	Ê
11	0B	VT	75	4B	K	139	8B	‹	203	CB	Ë
12	0C	FF	76	4C	L	140	8C	Œ	204	CC	Ì
13	0D	CR	77	4D	M	141	8D		205	CD	Í
14	0E	SO	78	4E	N	142	8E		206	CE	Î
15	0F	SI	79	4F	O	143	8F		207	CF	Ï
16	10	DLE	80	50	P	144	90		208	D0	Đ
17	11	DC1	81	51	Q	145	91	‘	209	D1	Ñ

18	12	DC2	82	52	R	146	92	’	210	D2	Ò
19	13	DC3	83	53	S	147	93	“	211	D3	Ó
20	14	DC4	84	54	T	148	94	”	212	D4	Ô
21	15	NAK	85	55	U	149	95	•	213	D5	Õ
22	16	SYN	86	56	V	150	96	–	214	D6	Ö
23	17	ETB	87	57	W	151	97	—	215	D7	×
24	18	CAN	88	58	X	152	98	˜	216	D8	Ø
25	19	EM	89	59	Y	153	99	™	217	D9	Ù
26	1A	SUB	90	5A	Z	154	9A	š	218	DA	Ú
27	1B	ESC	91	5B	[	155	9B	›	219	DB	Û
28	1C	FS	92	5C	\	156	9C	œ	220	DC	Ü
29	1D	GS	93	5D	]	157	9D		221	DD	Ý
30	1E	RS	94	5E	^	158	9E	ž	222	DE	Þ
31	1F	US	95	5F	_	159	9F	Ÿ	223	DF	ß
32	20	Sp	96	60	`	160	A0		224	E0	à
33	21	!	97	61	a	161	A1	¡	225	E1	á
34	22	"	98	62	b	162	A2	¢	226	E2	â
35	23	#	99	63	c	163	A3	£	227	E3	ã
36	24	$	100	64	d	164	A4	¤	228	E4	ä
37	25	%	101	65	e	165	A5	¥	229	E5	å
38	26	&	102	66	f	166	A6	¦	230	E6	æ
39	27	'	103	67	g	167	A7	§	231	E7	ç
40	28	(	104	68	h	168	A8	¨	232	E8	è
41	29	)	105	69	i	169	A9	©	233	E9	é
42	2A	*	106	6A	j	170	AA	ª	234	EA	ê
43	2B	+	107	6B	k	171	AB	«	235	EB	ë
44	2C	,	108	6C	l	172	AC	¬	236	EC	ì
45	2D	-	109	6D	m	173	AD		237	ED	í

46	2E	.	110	6E	n	174	AE	®	238	EE	î
47	2F	/	111	6F	o	175	AF	¯	239	EF	ï
48	30	0	112	70	p	176	B0	°	240	F0	ð
49	31	1	113	71	q	177	B1	±	241	F1	ñ
50	32	2	114	72	r	178	B2	²	242	F2	ò
51	33	3	115	73	s	179	B3	³	243	F3	ó
52	34	4	116	74	t	180	B4	´	244	F4	ô
53	35	5	117	75	u	181	B5	µ	245	F5	õ
54	36	6	118	76	v	182	B6	¶	246	F6	ö
55	37	7	119	77	w	183	B7	·	247	F7	÷
56	38	8	120	78	x	184	B8	¸	248	F8	ø
57	39	9	121	79	y	185	B9	¹	249	F9	ù
58	3A	:	122	7A	z	186	BA	º	250	FA	ú
59	3B	;	123	7B	{	187	BB	»	251	FB	û
60	3C	<	124	7C	|	188	BC	¼	252	FC	ü
61	3D	=	125	7D	}	189	BD	½	253	FD	ý
62	3E	>	126	7E	~	190	BE	¾	254	FE	þ
63	3F	?	127	7F	DEL	191	BF	¿	255	FF	ÿ

Tabela ASCII (American Standard Code for Information Interchange)