



Universidade Estadual de Maringá
Centro de Tecnologia
Departamento de Informática
Curso de Especialização:
Desenvolvimento de Sistemas para Web

ESP – WEB
Uso de SSL em Sites

Uso de SSL em Sites

Marco Antonio de Barros

Universidade Estadual de Maringá
Centro de Tecnologia
Departamento de Informática

Marco Antonio de Barros
Uso de SSL em Sites

Trabalho de Pós Graduação apresentado ao Curso Especialização em Desenvolvimento de
Sistemas para Internet, do Centro de Tecnologia, da Universidade Estadual de Maringá.

Orientador: Prof. Flávio Arnaldo B. da Silva

Maringá – PR

Brasil

2005

Marco Antonio de Barros

Uso de SSL em Sites

Este exemplar corresponde à redação final da monografia aprovada como requisito parcial para obtenção do grau de Especialista em Desenvolvimento de Sistemas para Internet da Universidade Estadual de Maringá, pela comissão formada pelos professores:

Orientador: Prof. Msc. Flávio Arnaldo B. da Silva
Departamento de Informática, CTC, DIN

Prof. MEng. Carlos Antonio Pizo
Departamento de Informática, CTC, DIN

Prof. Dr. David Calhau Jorge
Departamento de Informática, CTC, DIN

Maringá, Fevereiro de 2005

Universidade Estadual de Maringá

Departamento de Informática

Av. Colombo, 5790, Maringá – PR

CEP 87020-900

Tel: (44) 226-2727 R. 219/324 Fax: (44) 223-2676

Dedicatória

Eu gostaria de Dedicar este trabalho a minha família, meus amigos e aqueles que contribuíram direta e indiretamente com este trabalho.

AGRADECIMENTOS

Agradeço a Deus;

Agradeço a minha família;

Agradeço aos meus amigos;

Agradeço aos meus professores, em especial, agradeço o meu professor orientador;

Agradeço a Universidade.

RESUMO

Em todos os sistemas computacionais existe a necessidade de assegurar que as informações que trafeguem pela rede estejam em um meio seguro. Com a chegada da Internet e o aumento do uso pela rede, entretanto, as informações passaram a ter maior valor. Hoje em dia, não apenas as aplicações bancárias e de comércio eletrônico são as responsáveis pela necessidade de novas tecnologias de segurança na troca de informações, mas toda e qualquer aplicação que necessite de o mínimo sigilo pode contar com diversos recursos da computação atual. Os principais meios conhecidos são: VPN (*Virtual Private Network*) e SSL (*Secure Socket Layer*). Neste trabalho, apresentaremos uma solução prática do uso de SSL em sites de Internet.

ABSTRACT

In all the computation systems, there are necessities to assure the information that come to pass through by the network, been in secure middle. Therefore, with arrive of Internet and the increase of use by network, the information had passed to have more value. Nowadays, not only the banking applications and electronic commerce they are the responsible ones by necessities of new technologies of security in the exchange of information, but, all and any application that needs of minimal secrecy, it can count on diverse resources of current computation. The main known ways currently are: VPN (Virtual Private Network) e SSL (Secure Socket Layer). In this work, we will go to present a practice solution of use of SSL in the Internet sites.

LISTA DE ILUSTRAÇÕES

Figura 1. Processo de criptografia.....	3
Figura 2. Processo de criptografia Simétrica	5
Figura 3. Processo de criptografia Assimétrica	7
Figura 4. Processo de troca entre cliente e servidor HTTP	14
Figura 5. Arquivo de configuração do Apache	16

SUMÁRIO

Dedicatória.....	5
AGRADECIMENTOS.....	6
RESUMO	7
ABSTRACT	8
LISTA DE ILUSTRAÇÕES	9
SUMÁRIO.....	10
CAPÍTULO 1 – INTRODUÇÃO	1
CAPÍTULO 2 – CRIPTOGRAFIA.....	3
2.1. Tipos de Criptografia.....	4
2.2. Criptografia de Chave Simétrica	5
2.3 CRIPTOGRAFIA DE CHAVE ASSIMÉTRICA	6
CAPÍTULO 3 – <i>SECURE SOCKET LAYER</i> (SSL).....	8
3.1 ALGORITMOS PARA TROCA DE CHAVES DE SESSÃO DURANTE O HANDSHAKE.....	8
3.2 ALGORITMOS PARA DEFINIÇÃO DE CHAVE DE ENCRIPTAÇÃO	8
3.3 ALGORITMOS QUE IMPLEMENTAM A FUNÇÃO DE HASH PARA DEFINIÇÃO DO MAC	9
3.4 TIPOS DE CERTIFICADOS.....	9
CAPÍTULO 4 – <i>TRANSPORT LAYER SECURITY</i> (TLS).....	10
CAPÍTULO 5 – ASSINATURA DIGITAL	11
CAPÍTULO 6 – CERTIFICADO DIGITAL.....	12
CAPÍTULO 7 – IMPLEMENTAÇÃO	13
7.1 SERVIDOR.....	13
7.2 CLIENTE.....	13
7.3. LINUX.....	14
7.4. APACHE.....	14
7.5. OPENSSEL	15
7.6. INSTALAÇÃO	15
CAPÍTULO 8 – CONCLUSÃO	20
REFERÊNCIAS BIBLIOGRÁFICAS.....	21

CAPÍTULO 1 – INTRODUÇÃO

Houve uma época em que as pessoas não se preocupavam em trocar informações por meios eletrônicos, e independentemente da forma comunicativa ninguém esperava que outrem pudesse interceptar a comunicação e obter dados alheios, fossem dados sigilosos ou quaisquer outros tipos de dados. Com o passar dos anos, os computadores foram se tornando cada vez mais populares e acessíveis, não apenas a grandes empresas como também às pequenas empresas e às pessoas simples, acarretando a dependência cada vez maior destas empresas e pessoas com a estrutura informatizada, e, conseqüentemente, passaram a armazenar seus dados, importantes ou não, nos computadores. Inicialmente, eram grandes computadores, rápidos para a época, e, por manter os dados em um único local, não havia muito risco desses dados serem interceptados sem alguma dificuldade de acesso físico à rede de computadores na qual os dados estavam armazenados. Como houve a necessidade de interligar computadores em locais distantes, foram criadas redes espalhadas por locais distantes, entre eles a Internet. Com o advento da interligação desses computadores por redes como a Internet, os dados passaram a ser trocados por uma quantidade razoavelmente grande de pessoas, e agora esses dados não estavam mais concentrados em apenas um determinado local, não havia aquela proteção física dos dados que eram trocados pelas redes, o que levou as pessoas a se preocupar com a segurança e o sigilo de suas informações. Ao longo dos anos, sistemas de proteção foram criados e implementados para garantir a segurança da informação, e a maior preocupação centrava-se não somente em proteger os dados, mas fazer com que o acesso aos documentos protegidos pudesse ser trocado pelas pessoas certas, com segurança e agilidade. Northcutt (2001) assinala que para a maioria das indústrias a diferença entre uma empresa e sua concorrência se resume a alguns segredos corporativos ou processos proprietários. Uma das maneiras de garantir que os dados sejam trocados com segurança é transformar os dados corretos em informações ilegíveis, de difícil reconhecimento; a esse tipo de recurso denominamos criptografia. Segundo o dicionário Aurélio, criptografia é a arte de escrever em cifra ou em código, ou ainda, o conjunto de técnicas que permitem criptografar informações (como mensagens escritas, dados armazenados ou transmitidos por computador, etc.). Sem o uso de criptografia, torna-se arriscada e perigosa a realização de transações via Internet, como compras, nas quais o número de cartão de crédito, CPF e/ou informações pessoais são solicitadas pelo site. Para transações on-line, o uso de SSL (*Security Socket Layer*) é

imprescindível, pois garante que todos os dados trocados entre o cliente e o fornecedor (site de compras) está sendo criptografado, ou seja, está sendo trocado de forma segura. Neste trabalho, apresentaremos como tomar uso de SSL em sites da Internet, mostrando como implementar na prática um sistema de SSL.

CAPÍTULO 2 – CRIPTOGRAFIA

Como já mencionamos no capítulo primeiro, na introdução deste trabalho, criptografia é definida como a arte de escrever em cifras, e cifra, em conformidade com o dicionário Aurélio, é a explicação ou chave de uma escrita enigmática ou secreta. A palavra criptografia vem do grego *Kryptos*, que significa oculto, e *graphen*, que significa escrever (SILVA, 2004). Qualquer tipo de dado que pode ser lido e entendido é chamado de texto plano ou texto puro (PGP.COM, 2004). Em um processo criptográfico, os dados são submetidos a uma função matemática que transforma as informações em dados ininteligíveis, tornando a informação encriptada. Após o dado estar encriptado, só é possível a sua leitura retornando o dado a sua informação original. Ao processo inverso da encriptação chama-se decriptação. O processo de encriptação/decriptação é mostrado na figura a seguir.

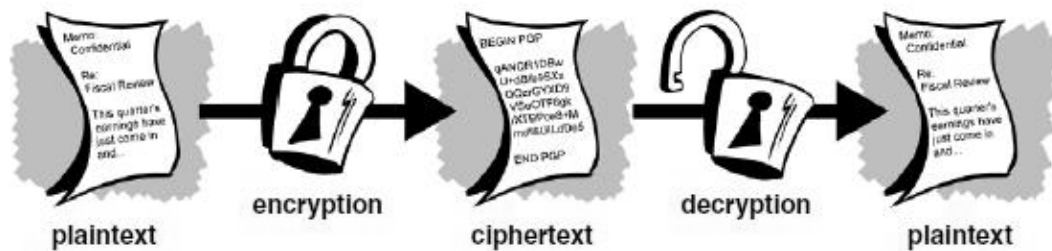


Figura 1. Processo de criptografia.

Enquanto a criptografia é a ciência que estuda os meios, as formas para a ocultação das informações, criptoanálise é a ciência que estuda e explora as formas para a quebra da segurança (PGP.COM, 2004). Criptologia, assim, é a ciência que estuda tanto a criptografia quanto a criptoanálise.

2.1. Tipos de Criptografia

Os sistemas criptográficos são vários, desde os mais simples e rápidos como os mais complexos e difíceis de serem quebrados e descobertos. Acredita-se que o primeiro sistema criptográfico criado no mundo tenha sido o sistema de César (SILVA, 2004). Há mais de dois mil anos, o imperador romano Júlio César, para ter certeza de que suas mensagens não seriam descobertas por seus inimigos e espiões, criou um sistema de criptografia baseado na transposição de letras. O sistema consistia na substituição de letras do alfabeto por outras letras também do alfabeto. Nesse sistema, César trocava a letra A pela letra D, a letra B pela letra E, e assim sucessivamente. Temos um exemplo a seguir:

Leve este comunicado ao general.

Esse mesmo texto, após ser submetido à função criptográfica criada por César, ficaria assim:

Ohyh hvwh frpxqlfdgr dr jhqhudo.

Como não havia na época o uso desse tipo de artifício, isto funcionou bem porque tanto o imperador quanto quem receberia a mensagem conhecia o algoritmo, a função e a técnica utilizada para a cifragem e a decifragem da mensagem. Caso alguém mais tivesse acesso à forma como a mensagem era cifrada, seria possível reverter toda a mensagem que fosse enviada ao imperador ou aos que receberiam a mensagem. Ao longo dos anos, os sistemas foram sendo criados, quebrados, aperfeiçoados e abandonados. Enfatizamos que o relevante não é o algoritmo de criptografia em si, mas sim a chave empregada para embaralhar as informações. Os melhores sistemas de criptografia são aqueles conhecidos de todos, pois, assim sendo, são submetidos a rigorosos testes, sendo analisados por engenheiros, matemáticos e cientistas. Para um sistema de criptografia ser considerado seguro, o mesmo deve se garantir por longos anos, mesmo submetido a um ataque por força bruta ao maior e mais rápido computador ou ao conjunto de computadores, formando computação paralela. Há, basicamente, dois tipos de criptografia, a saber: criptografia com chave simétrica e criptografia com chave assimétrica.

2.2. Criptografia de Chave Simétrica

O sistema de criptografia intitulado Chave Simétrica foi criado em 1972 pela IBM, sendo também chamado de Lúçifer Cipher, e mais tarde, em 1977, foi reavaliado pelo *National Institute of Standards* (NIST), *Federal Information Processing Standards* (FIPS) e *American National Standards Institute* (ANSI), recebendo o nome de *Data Encryption Standards* (DES). Em criptografia simétrica, o emissor aplica a sua chave, também chamada de chave secreta, ao texto que deseja enviar, também chamado de texto plano ou texto puro. O resultado da chave + texto plano é o texto cifrado ou encriptado. Quando a mensagem chegar ao seu destino, o processo a ser realizado pelo receptor é o inverso, ou seja, de posse da chave simétrica ou chave secreta, e do texto encriptado, este aplica a chave + texto encriptado e o resultado é o texto plano, texto original nas mãos, daí o nome simétrica, pois a chave que fecha é a chave que abre.

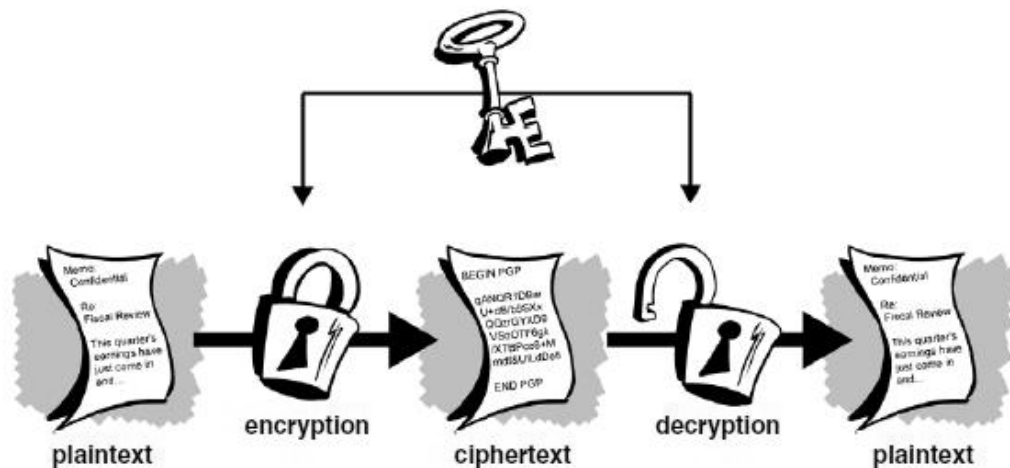


Figura 2. Processo de criptografia Simétrica.

2.2.1 Data Encryption Standards (DES)

O DES é um padrão de criptografia de chaves simétricas utilizado pelos Estados Unidos. É um padrão de criptografia de bloco que usa chaves de 64 bits de comprimento, sendo 56 bits para a chave e 8 de paridade (SILVA, 2004). Durante décadas o padrão DES tem sido analisado por críticos criptógrafos a respeito de sua confiabilidade, porém até hoje não se

encontraram falhas no algoritmo de encriptação dos dados, e apenas os testes acadêmicos de quebra por força bruta tiveram algum êxito. Para aumentar o grau de segurança do algoritmo DES basta aumentar o tamanho da chave.

Inicialmente, o governo americano impôs limite ao tamanho máximo da chave do DES, visto que segundo o governo essa ação é uma medida de segurança. O sistema de criptografia por chave simétrica é mais rápido que o sistema de criptografia assimétrica.

2.2.2 Triplo Data Encryption Standards (Triplo DES)

Com o passar dos anos e com o aumento do poder computacional dos computadores, o algoritmo DES já não era mais tão seguro, foi preciso que um novo algoritmo fosse desenvolvido para poder suprir a necessidade do mercado que carecia de um sistema de criptografia mais seguro. O 3DES (Triplo DES) é muito parecido com seu antecessor DES, todavia, possui uma particularidade: nesse algoritmo, a informação é criptografada três vezes. De acordo com Silva, o 3DES usa três chaves combinadas para criptografar a mensagem, isto é, são três chaves de 56 bits, totalizando 168 bits, para garantir uma maior segurança.

2.3 CRIPTOGRAFIA DE CHAVE ASSIMÉTRICA

O sistema de criptografia conhecido como criptografia por chave assimétrica surgiu em 1975, quando dois pesquisadores da Universidade de Standford, Withfield Diffie e Martin Hellman, escreveram um artigo pressupondo que havia uma forma de criptografar um documento com uma chave e decriptografar com outra chave, sem ter relação com a primeira chave. O sistema de criptografia de chave assimétrica também é conhecido por sistema de chave pública, haja vista que são criadas duas chaves, uma denominada chave pública, usada para criptografar, e a outra chamada de chave privada, usada para decriptografar. Nesse sistema, uma das chaves, a chave pública, fica livre para ser usada por qualquer pessoa que queira proteger a informação, e a outra, a chave privada, é usada para realizar o processo inverso, ou seja, para desproteger a informação e voltá-la a sua forma original. A chave privada não é compartilhada, visto que apenas com essa chave é possível decriptografar os dados. Um sistema de chave simétrica é mais simples do que

um modelo de chave assimétrica, porque no sistema de chave simétrica a preocupação se encontra em modelos matemáticos capazes de embaralhar as informações de tal forma que elas se tornem mais confiáveis. Em um modelo de chave assimétrica, a preocupação vai além de simplesmente embaralhar os dados, está em criar um sistema que consiga fatorar dois números primos que não tenham relação um com o outro, e que um deles possa ser distribuído por uma rede.

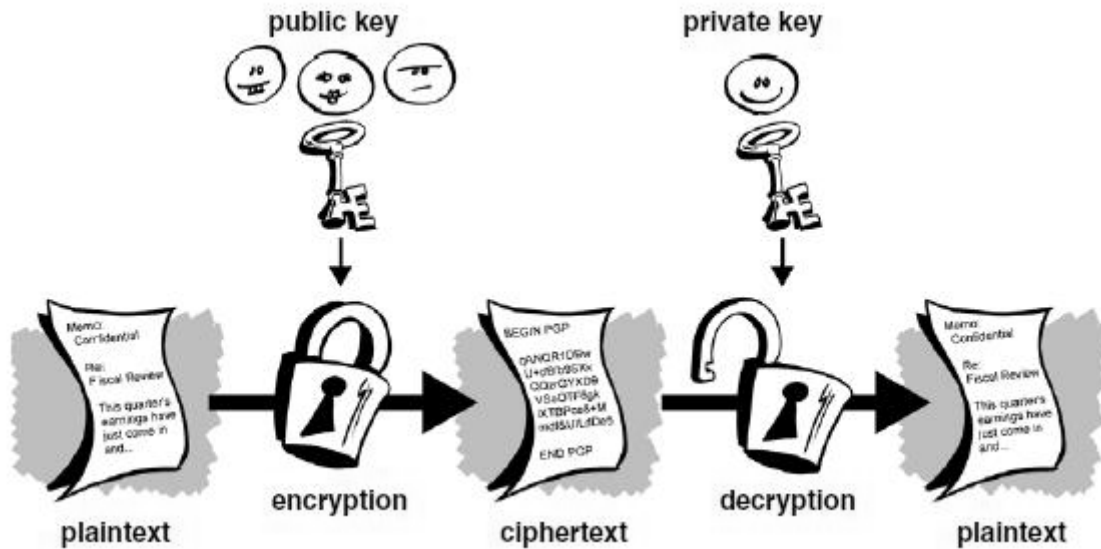


Figura 3. Processo de Criptografia Assimétrica.

CAPÍTULO 3 – SECURE SOCKET LAYER (SSL)

O SSL é um protocolo de comunicação de dados que implementa um canal seguro em aplicações de redes, principalmente a Internet, de forma simples e transparente para o usuário. O SSL foi desenvolvido pela *Netscape Corporation* no ano de 1994, tendo sua versão alterada de 1.0 para 2.0 no ano de 1995. Daí para frente, os *browsers* mais comuns, como *Netscape Communication* e *Internet Explorer* e servidores de páginas como Apache e IIS, passaram a trazer integrados a si suporte para o uso de SSL. O SSL também é usado na maioria dos sites que realizam transações comerciais, como os de compras e bancos. O uso de SSL teve grande impulso com a chegada dos certificados digitais, pois essa chegada permitiu que o uso de logins e senhas possa, em alguns casos, ser substituído pelo seu uso.

É importante ressaltarmos que o SSL não é um algoritmo de criptografia; mas o responsável por implementar uma via segura de troca de informações. Esse sistema não influencia na escolha de um determinado padrão de criptografia ou certificado digital; para isso, podemos escolher uma gama de protocolos que desempenham essa função.

Apresentamos, a seguir, uma lista de protocolos de criptografia, assinatura e certificado digital que podem ser usados em conjunto com o SSL v3.

3.1 ALGORITMOS PARA TROCA DE CHAVES DE SESSÃO DURANTE O HANDSHAKE

NULL, RSA, Diffie-Hellman RSA, Diffie-Hellman DSS, DHE_DSS, DHE_RSA, DH_anonymous, Fortezza/DMS

3.2 ALGORITMOS PARA DEFINIÇÃO DE CHAVE DE ENCRIPTAÇÃO

NULL, RC2, RC4, IDEA, DES, 3DES, Fortezza

3.3 ALGORITMOS QUE IMPLEMENTAM A FUNÇÃO DE HASH PARA DEFINIÇÃO DO MAC

NULL , SHA , MD5

3.4 TIPOS DE CERTIFICADOS

X.509 v1,X.509 v2,X.509 v3

O SSL trabalha entre as camadas de Transporte (TCP) e aplicação, é um protocolo que independe de protocolos como http, ftp entre outros.

CAPÍTULO 4 – *TRANSPORT LAYER SECURITY (TLS)*

O TLS é uma versão *free* do SSL; embora o TLS v1 seja muito parecido com o SSL v3, eles não se comunicam, sendo necessário que o TLS aja de forma a usar compatibilidade retroativa. O protocolo TLS fornece um serviço seguro e confiável *End-to-End* para as camadas mais altas do TCP. O TLS é uma atualização da versão atual do SSL v3. Esse protocolo foi desenvolvido por um grupo de trabalho do IETF e a sua definição está descrita na RFC 2246, sendo atualizada em outra publicada recentemente, a RFC 3546. Outras RFCs complementam a especificação atual, são as RFC 2712, 2817, 2818 e 3268 [5]. O protocolo TLS opera nas camadas de sessão e transporte, entre a camada de aplicação (por exemplo: HTTP) e o protocolo de transporte TCP (Figura 3), fornecendo as seguintes opções de segurança: dupla autenticação; confidencialidade; garantia de integridade na troca das mensagens e não-repúdio (BERIM, 2003).

CAPÍTULO 5 – ASSINATURA DIGITAL

A assinatura digital assegura se uma determinada informação foi ou não enviada por uma determinada pessoa. Por exemplo, caso um determinado usuário envie um email para outro usuário, este deve assinar o email para garantir que o destinatário receberá o conteúdo do email de forma segura – o destinatário recebe o email e precisa ter certeza de que o email é realmente da pessoa que diz ser. Para que essa etapa funcione corretamente, o emissor utiliza a sua chave privada para encriptar as informações e gerar, assim, a assinatura; e o destinatário usa a chave pública do emissor para realizar o processo inverso, isto é, decriptografar as informações. Desta forma, é possível a quem estiver recebendo a mensagem garantir que um determinado usuário enviou ou não aquela informação. Caso o destinatário receba a informação, mas a assinatura não condiz com a chave do possível emissor, a mesma é tida como falsificação ou fraude na comunicação. Em conformidade com Silva (2004), a assinatura não pode ser repudiada, isto faz com que a origem da informação tenha como ser validada, garantindo que não haja problemas como fraude.

CAPÍTULO 6 – CERTIFICADO DIGITAL

O certificado digital é a forma mais segura de garantir que a chave pública utilizada é realmente a chave do emissor ou do receptor. A utilização de certificados digitais se dá por meio de uma terceira parte, também chamada de Autoridade Certificadora (AC). Pela AC temos a garantia de uma entidade idônea que intermediará a negociação entre emissor e receptor. Há vários tipos de certificados digitais, entre eles o X.509 (o mais conhecido), SPKI/SDSI, SET, PGP etc. (SILVA, 2004). O protocolo X.509, mais precisamente em sua terceira versão (2002), é o padrão escolhido para ser utilizado pela ICP-Brasil, a principal AC do Brasil. O certificado no padrão X.509 possui vários campos contendo informações do usuário como: versão do protocolo, chave pública, algoritmo utilizado para encriptar as informações, assinatura, entre outros campos.

CAPÍTULO 7 – IMPLEMENTAÇÃO

Como já apregoa o título deste estudo, visamos efetuar a preparação prática de um ambiente de uso de SSL em sites. Para a realização deste trabalho, será necessária a criação de um sistema servidor de páginas utilizando o serviço de criptografia SSL em uma máquina que utilize um sistema operacional de redes. Do lado do cliente será necessário também o uso de um aplicativo denominado *browser*, o qual fará acesso ao servidor de redes que estará executando um sistema servidor de páginas juntamente com o serviço de criptografia SSL.

O cliente, ao acessar uma página com recursos de página segura (SSL) no servidor, será notificado de que está navegando em um site que possui recursos de segurança, e após aceitar a instalação do certificado necessário do lado do servidor, aí sim visualizará a página de forma segura. Demonstramos, a seguir, a lista dos componentes utilizados para a implementação desse teste:

7.1 SERVIDOR

Microcomputador Intel Celeron 1.7 Ghz

Memória de 512 MB RAM

Sistema operacional Linux (Slackware 9.1)

Sistema servidor de Páginas (Apache v1.3.28)

Sistema de encriptação SSL (OpenSSL v0.9.7b)

7.2 CLIENTE

Microcomputador Intel Celeron 2.5 Ghz

Memória de 256MB RAM

Sistema operacional Windows (Windows XP Professional com SP2)

Browser de navegação (Internet Explorer 6)

7.3. LINUX

Para o desenvolvimento prático de implementação deste trabalho será utilizado, como sistema operacional servidor, o Linux. Entretanto, há outros sistemas operacionais que possuem como suporte a implementação de SSL, como o Windows NT, Windows 2000, Windows XP, Windows 2003, entre outros. O Linux apresenta diversas vantagens: primeiramente, é um software Livre, isto significa que, além de não ser necessário pagar pela sua utilização, nos dá total liberdade para alterações no código fonte por pessoas habilitadas. Em segundo lugar, é um software estudado por muitos programadores ao redor do mundo, já que o código pode ser visto e alterado por qualquer pessoa, pode também ser analisado à procura de falhas de segurança e outros tipos de falhas, como instabilidade.

7.4. APACHE

O uso de SSL ocorre em vários segmentos da computação, como email, Proxy, http entre outros, mas o principal uso se dá em navegação de páginas http. Quando o cliente acessa uma determinada página em dado servidor, ocorre uma série de processos de troca de informações entre ambos. Após o cliente solicitar o carregamento da página, ocorre o processo de transferência de hipertexto para o cliente, onde é executado.

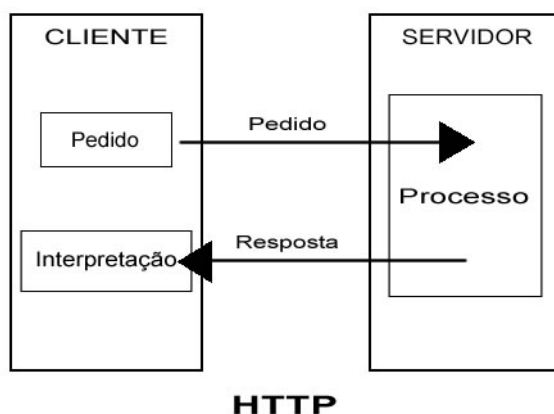


Figura 4. Processo de troca entre cliente e servidor http.

Do lado do cliente, o software *browser* faz a solicitação, e do lado servidor, um software denominado servidor http responde à solicitação do cliente. Assim como o sistema operacional Linux, o Apache também é, além de gratuito, código aberto, ou seja, qualquer programador pode verificar o código e procurar novas falhas e/ou implementar novos recursos. Há outros servidores de páginas, como o *Internet Information Services* (IIS) da Microsoft e outros servidores de empresas como IBM, Novell, Netscape etc. O Apache tem a grande vantagem de ser multiplataforma, ou seja, funciona não apenas no ambiente Linux como também em outros Unix e até mesmo no Windows.

7.5. OPENSSL

Denominamos OpenSSL a um conjunto de utilitários, bibliotecas e ferramentas utilizado para encriptação de dados em vários tipos de programas. O modelo OpenSSL também é um software gratuito e código aberto, isto é, qualquer um pode utilizar e/ou modificar de acordo com suas necessidades. O conjunto de bibliotecas do OpenSSL é utilizado quando, no momento da negociação entre cliente e servidor, é solicitada uma comunicação segura por SSL. O servidor de páginas solicita e transfere dados encriptados baseado em informações como chave pública e privada. Essas mesmas chaves podem ser criadas utilizando o conjunto de ferramentas OpenSSL.

7.6. INSTALAÇÃO

Para a implementação da solução de criação de um sistema servidor de páginas com o uso de SSL foi necessário a instalação do software Linux em sua distribuição Slackware 9. O Slackware foi instalado no módulo completo, com todos os programas (pacotes). Primeiramente, é necessário habilitar o servidor de páginas Apache a aceitar o uso de SSL. Para ativarmos o acesso ao protocolo HTTPS (http seguro), devemos editar o arquivo */etc/apache/httpd.conf*, localizar a linha ***Include /etc/apache/mod_ssl.conf***, provavelmente ela estará comentada, ou seja, desabilitada. Para habilitar a configuração, devemos remover o símbolo do jogo da velha (#) do início da linha, salvar e sair.

```
# ==> mod_php configuration settings <==
#
# PACKAGES REQUIRED:  openssl-solibs (A series) and/or openssl (N series),
#                   mysql (AP series), gmp (L series), and apache (N series)
#
#Include /etc/apache/mod_php.conf

# ==> mod_ssl configuration settings <==
#
# PACKAGES REQUIRED:  apache (N series) and openssl (N series)
#
#MAB
Include /etc/apache/mod_ssl.conf
```

Figura 5. Arquivo de configuração do Apache.

Depois, devemos editar o arquivo */etc/apache/mod_ssl.conf* e adicionar as seguintes configurações:

```
SSLCertificateFile /etc/apache/ssl/server.crt
SSLCertificateKeyFile /etc/apache/ssl/server.key
SSLCACertificateFile /etc/apache/ssl/ca.crt
```

Após adicionar as configurações a seguir, devemos salvar e sair.

Essas linhas supracitadas são responsáveis pelas configurações do certificado do servidor, da chave privada do servidor e do certificado da autoridade certificadora.

Depois de realizadas as alterações nos arquivos do Apache, é hora de criarmos os certificados. Para tanto, devemos criar um diretório chamado “SSL”, onde ficarão os certificados.

```
mkdir /etc/apache/ssl
cd /etc/apache/ssl
```

Após criarmos e entrarmos no diretório, precisaremos gerar a chave que será usada para criar o certificado e, de posse da chave, criarmos o certificado.

```
openssl genrsa -out server.key 1024
openssl req -new -key server.key -out server.csr
```

Agora é hora de criarmos a chave e o certificado da CA.

```
openssl genrsa -out ca.key 1024
```

```
openssl req -new -x509 -days 365 -key ca.key -out ca.crt
```

Para finalizar, é necessário realizar a assinatura do certificado. Para isso, devemos ir até o site oficial da ModSSL, cujo nome está inscrito a seguir, baixá-lo e descompactá-lo.

```
http://www.modssl.org/source/mod_ssl-2.8.19-1.3.31.tar.gz
```

```
tar xzfv mod_ssl-2.8.19-1.3.31.tar.gz
```

Ao copiarmos o seguinte arquivo: `mod_ssl-2.8.19-1.3.31/pkg.contrib/sign.sh` para um diretório qualquer referenciado na variável de ambiente `PATH` como o `/usr/local/bin` para que possa ser invocado, estando a partir de qualquer lugar dentro da árvore de diretórios.

```
sign.sh server.csr
```

Para testar o funcionamento, devemos digitar a seguinte linha e abrir um *browser* qualquer e disparar contra a própria máquina.

```
/usr/sbin/apachectl startssl
```

```
https://127.0.0.1
```

Caso não seja possível baixar o arquivo de assinatura do site, apresentamos a seguir o conteúdo do mesmo:

Conteúdo do arquivo sign.sh

```
#!/bin/sh
##
## sign.sh -- Sign a SSL Certificate Request (CSR)
## Copyright (c) 1998-2001 Ralf S. Engelschall, All Rights
## Reserved.
##

# argument line handling
CSR=$1
if [ $# -ne 1 ]; then
    echo "Usage: sign.sign <whatever>.csr"; exit 1
fi
if [ ! -f $CSR ]; then
    echo "CSR not found: $CSR"; exit 1
fi
case $CSR in
    *.csr ) CERT=`echo $CSR | sed -e 's/\.csr/.crt/'` ;;
    * ) CERT="$CSR.crt" ;;
esac

# make sure environment exists
if [ ! -d ca.db.certs ]; then
    mkdir ca.db.certs
fi
if [ ! -f ca.db.serial ]; then
    echo '01' >ca.db.serial
fi
if [ ! -f ca.db.index ]; then
    cp /dev/null ca.db.index
fi

# create an own SSLeay config
cat >ca.config <<EOT
[ ca ]
default_ca          = CA_own
[ CA_own ]
dir                 = .
certs               = \${dir}
new_certs_dir       = \${dir}/ca.db.certs
database            = \${dir}/ca.db.index
serial              = \${dir}/ca.db.serial
RANDFILE            = \${dir}/ca.db.rand
certificate          = \${dir}/ca.crt
private_key         = \${dir}/ca.key
default_days        = 365
default_crl_days    = 30
default_md           = md5
preserve            = no
policy              = policy_anything
```

```
[ policy_anything ]
countryName          = optional
stateOrProvinceName = optional
localityName         = optional
organizationName     = optional
organizationalUnitName = optional
commonName           = supplied
emailAddress         = optional
EOT

# sign the certificate
echo "CA signing: $CSR -> $CERT:"
openssl ca -config ca.config -out $CERT -infiles $CSR
echo "CA verifying: $CERT <-> CA cert"
openssl verify -CAfile ca.crt $CERT

# cleanup after SSLeay
rm -f ca.config
rm -f ca.db.serial.old
rm -f ca.db.index.old

# die gracefully
exit 0
```

CAPÍTULO 8 – CONCLUSÃO

Na atualidade, a Internet passou a ser um dos meios mais utilizados para a troca de informações e para a realização de negócios. O potencial de negócios a ser realizado dentro dessa grande rede gera milhões de oportunidades diariamente, não apenas em instituições financeiras, mas também em muitos outros ramos de atividades. A grande preocupação com esse crescimento da troca de informações entre pessoas e/ou empresas na Internet se dá em relação à segurança da informação que é trocada em sistemas computacionais.

O uso de SSL em sites da Internet traz não apenas segurança, mas também alívio para quem precisa realizar qualquer tipo de transação na Internet, seja pagamento, troca de informações confidenciais ou outro tipo qualquer de transação via Web.

Neste trabalho, realizamos uma abordagem prática do uso de SSL em sites da Internet, mostrando que não é necessariamente obrigatório que o programador tenha sólidos conhecimentos de segurança para implementar um meio seguro de troca de informações, mas basta que o servidor esteja preparado para o fornecimento de um canal seguro, pois, através da implementação de SSL, toda troca de informação entre emissor e receptor é realizada de forma confiável.

Mediante a criação de um ambiente computacional seguro foi possível atingirmos os objetivos implementar o SSL em sites da Internet em ambiente Linux e por meio de comandos SSL.

REFERÊNCIAS BIBLIOGRÁFICAS

LARGURA, Luiz Aristides Rios. **Monografia sobre SSL para o Curso de Extensão Segurança em Redes de Computadores** (5ª. turma). Brasília, 2000.

SILVA, Lino Sarlo. **Public Key Infrastructure – PKI**. São Paulo, Editora Novatec, 2004.

TLS, Transport Layer Security. Disponível em: <http://www.ietf.org/rfc/rfc2246.txt>. Acessado em 12 de janeiro de 2005.

SSL, Secure Sockets Layer. Disponível em: <http://wp.netscape.com/eng/ssl3/draft302.txt>. Acessado em 12 de janeiro de 2005.

JUNIOR, Antonio Bizeti. **Certificação Digital – Um Estudo de Ferramentas e Ambientes**. Maringá, 2003.

SILVA, Bruno de Melo. **Uma abordagem de infra-estrutura de chaves públicas para ambientes corporativos**. Brasília, 2004.