Autoridade Certificadora para Acesso Seguro

Alessandro Martins
Laboratório RAVEL / COPPE / UFRJ
Última atualização em 02/02/2001 (ver. 0.1)

RESUMO

Com o crescimento do número e da variedade das transações de toda ordem na Internet, é cada vez mais forte a necessidade de se garantir a segurança na troca de dados entre os navegadores e os servidores *we*b. O protocolo mais utilizado para tal finalidade é o HTTP que não oferece as características de segurança necessárias para um transação comercial. Para contornar esse problema, usa-se o protocolo SSL *(Secure Sockets Laye*r) que tem como característica principal o estabelecimento de um canal privado (os dados são cifrados), autenticado (o servidor e o cliente podem ser autenticado) e confiável (o transporte de uma mensagem inclui uma verificação de integridade). A autenticação feita pelo SSL é realizada com a utilização de um Certificado Digital, que pode ser considerado como uma identidade digital. Estes Certificados Digitais são emitidos por entidades confiáveis conhecidas como Autoridades Certificadoras. Este trabalho envolve a construção da Autoridade Certificadora que emite, revoga e gerencia os Certificados Digitais para utilização em uma transação segura via *we*b.

1. INTRODUÇÃO

Com a crescente utilização dos computadores e das redes de computadores, especialmente da Internet, a segurança tornou-se um requisito essencial. Apesar de uma boa parte das informações transferidas na Internet ser para acesso público, há operações que requerem algum nível de segurança, acertado entre as partes, como por exemplo transações com número de cartões de crédito, dados de contas bancárias, além de acesso a informações privadas. Assim, aspectos como privacidade, autenticação, autorização e integridade da informação são todos elementos importantes da estratégia de segurança referente às transações via *web.*

A seguir serão ilustrados, através de um exemplo clássico, os problemas de segurança específicos das comunicações na *web* e os principais elementos ou módulos de segurança empregados na proteção dessas comunicações, de modo a explicar em linhas gerais a necessidade de cada um dos módulos no desenvolvimento de um ambiente de comunicação segura para transações *we*b.

O exemplo clássico apresentado neste capítulo envolve uma comunicação entre duas entidades, que podem corresponder a usuários humanos, ou *softwar*e, ou computadores. No exemplo, uma entidade, denominada Alice, realiza em conjunto com a outra, denominada Banco, uma operação ou transação, usando as técnicas da *web* e envolvendo informações que precisam ser absolutamente protegidas, no caso, valores financeiros, identificações das entidades, números de contas, etc. A transação realizada consiste de uma mensagem enviada por Alice ao seu Banco solicitando uma determinada transferência de dinheiro. Durante a apresentação do exemplo, as

necessidades de segurança são introduzidas, assim como os mecanismos de segurança que têm-se mostrado ou têm o potencial de responder efetivamente a tais necessidades. Tais mecanismos de segurança colocam em ação determinados algoritmos e protocolos específicos de segurança que são brevemente conceituados.

1.1 A necessidade de algoritmos de criptografia

No que se refere à mensagem de Alice para o seu Banco solicitando uma transferência de dinheiro, uma necessidade básica de segurança é garantir que tal mensagem seja privada, uma vez que inclui informações tais como o número da conta bancária e a quantidade de dinheiro a ser transferida. Por privacidade, entende-se que, ao transitar na mídia *we*b, a mensagem não poderá ser lida e compreendida por uma terceira entidade não autorizada, considerando que apenas Alice e o Banco têm autoridade de acesso sobre a mensagem. Para proteger essa mensagem, garantindo a privacidade, ou segredo, ou confidencialidade, um primeiro item da solução de segurança é o emprego, por Alice e pelo Banco, de um algoritmo de criptografia, capaz de transformar a mensagem original em uma mensagem cifrada, ou seja, pressupostamente não legível por uma terceira entidade. O algoritmo de criptografia escolhido é aplicado de uma forma particular, por Alice e pelo Banco, em função de um ou mais parâmetros pertencentes às duas entidades e compartilhados em parte, ou no todo, por elas duas. De acordo com o modo de funcionamento do algoritmo e, especificamente, do modo de utilização das chaves criptográficas, existem duas categorias de algoritmos de criptografia: a simétrica e a assimétrica:

- A criptografia simétrica, requer que o transmissor e o receptor compartilhem uma única chave: uma informação secreta, a chave, que deve ser usada para cifrar e decifrar uma mensagem. Se a chave permanece secreta, então ninguém mais, além do transmissor e do receptor, pode ler a mensagem. Se Alice e o Banco conhecem a mesma chave secreta e não a compartilham com mais ninguém, então eles podem enviar mensagens privadas um para o outro.

- Na criptografia assimétrica, também conhecida como criptografia de chave pública, é empregado um algoritmo que usa um par de chaves, uma pública e uma privada, que são utilizadas para cifrar e decifrar a mensagem. Se uma das chaves é usada para cifrar uma mensagem, então a outra deve ser usada para a decifração. Assim, o Banco escolhe duas chaves, guarda uma delas, a denominada chave secreta, e publica a outra. Desse modo, Alice pode cifrar uma mensagem usando a chave pública do Banco e enviar para este a mensagem cifrada, pois apenas ele, o Banco, que é o proprietário da chave privada, poderá decifrar a mensagem enviada.

A utilização do sistema de criptografia convencional requer que uma chave secreta seja compartilhada entre cada par de entidades com necessidades de comunicação, o que implica na necessidade de uma quantidade de chaves correspondente ao quadrado da quantidade de entidades comunicantes. Uma grande quantidade de chaves acarreta sérios problemas de distribuição, proteção e manutenção de chaves secretas, tornando a criptografia simétrica menos adequada para o ambiente *web* que aquela com chave pública, onde é suficiente que cada

entidade guarde, proteja e dê manutenção apenas à sua própria chave privada, divulgando amplamente a correspondente chave pública. Por essa razão, ainda que na *web* seja possível a utilização de criptografia simétrica, e que essa forma de criptografia seja efetivamente utilizada em determinados momentos da comunicação, na *web* a criptografia com chave pública tem plena aplicação.

1.2 Aplicação dos digestores de mensagens *(hash)*

Ainda que Alice possa cifrar sua mensagem tornando-a secreta, há outros riscos a serem considerados. Por exemplo, caso Alice e o Banco estejam usando criptografia por chave pública, uma terceira entidade pode interceptar ou substituir a mensagem enviada por Alice, de modo a transferir o dinheiro para essa terceira entidade. A questão de segurança nesse caso é a necessidade de garantir que a mensagem não seja modificada, em parte ou no todo, ao ser transferida, ou seja, trata-se da necessidade de garantir a integridade da mensagem.

Uma maneira de garantir a integridade da mensagem é fazer Alice calcular um resumo *(hash)* da mensagem e enviá-lo ao Banco anexado à própria mensagem. O receptor da mensagem, o Banco, ao receber a mensagem, também executa o cálculo de um resumo, comparando este último com aquele enviado por Alice. Se os dois resumos forem iguais, considera-se que a mensagem foi recebida intacta.

O resumo da mensagem é calculado usando um algoritmo que é chamado de digestor de mensagem, ou de *has*h. Os digestores de mensagens são usados para obter como resultado uma representação pequena e de tamanho fixo para as mensagens que podem ter tamanho variável. A probabilidade de se encontrar duas mensagens com o mesmo *hash* é muito pequena, por isso, pode se considerar que existe um resumo único para cada mensagem.

O problema na utilização dos resumos, é que Alice deve conseguir uma maneira de enviar cada resumo para o Banco seguramente, isto é, sem que o próprio resumo possa ser alterado por uma terceira entidade ao percorrer a mídia *we*b; quando isto é alcançado, a integridade da mensagem é garantida. Para proteger o próprio resumo, é necessário mais um elemento de proteção: a assinatura digital.

1.3 Aplicação da assinatura digital

Além da questão da integridade da mensagem, a assinatura digital é empregada também para responder a uma outra necessidade: o receptor precisa Ter uma garantia da origem da mensagem. Quando Alice envia sua mensagem para o Banco, este necessita garantir que a mensagem realmente provem de Alice, ou seja, que a mensagem é autêntica e que a operação financeira pode efetivamente ser realizada em nome de Alice.

A assinatura digital, criada por Alice e incluída na mensagem, consiste do cálculo de um resumo por um digestor de mensagem, e da cifração desse resumo, conjuntamente com outras informações, tal como um número de série, usando normalmente um algoritmo de chave pública, mas realizando a operação criptográfica usando a chave privada do transmissor. Como qualquer entidade pode decifrar o

resumo cifrado, bastando para tanto usar a chave pública do transmissor, e como apenas o transmissor assinante conhece a chave privada correspondente, há uma efetiva garantia de que apenas ele, o transmissor, pode ter assinado a mensagem.

Além disso, visto que o resumo só pode ser cifrado usando a chave privada do transmissor, uma terceira entidade, que não conhece tal chave, não poderá modificar a mensagem pois não terá como gerar um resumo cifrado correspondente à mensagem alterada. Desse modo, com a assinatura digital e presença de um resumo válido, há uma garantia da integridade e origem da mensagem.

1.4 Aplicação dos Certificados Digitais

Embora Alice possa ter enviado uma mensagem privada e assinada para o Banco, ela ainda necessita ter uma garantia de que realmente está se comunicando com o Banco.

Na verdade, ao utilizar uma chave pública, uma entidade precisa ter a garantia de que está usando uma chave que efetivamente correspondente à chave privada da outra entidade, haja visto que qualquer entidade tem a possibilidade de gerar um par de chaves correlacionadas e publicar uma delas.

Assim, o Banco também necessita verificar que a chave utilizada para assinar uma determinada mensagem pertence realmente a Alice. Para evitar que uma terceira entidade publique uma chave fazendo-se passar por Alice, faz-se necessário que Alice registre sua chave pública junto a alguma entidade confiável que tenha a possibilidade de divulgar um certificado garantindo a autenticidade da chave de Alice. Essa entidade confiável equivale a um cartório eletrônico onde se pode depositar uma chave pública, obtendo um certificado associando tal chave pública com o seu proprietário, ou seja, com a entidade que gerou e detém a chave privada correspondente. Assim, quando uma entidade apresenta um certificado registrado é como se esta entidade estivesse apresentando uma carteira de identidade digital cuja validade pode ser verificada junto à entidade confiável que gerou o certificado.

Nessas condições, em uma troca de mensagens entre duas entidades comunicantes, se cada parte tem um certificado assinado por uma autoridade confiável, então ambas as entidades têm alguma garantia de que estão efetivamente se comunicando com uma entidade bem identificada e autêntica. A terceira entidade confiável, que emite os certificados, é chamada de Autoridade Certificadora, e os certificados digitais são usados para autenticação de entidades comunicantes, para determinação de chaves a serem usadas nas comunicações e para registrar históricos das comunicações, de modo que as transações possam ser verificadas e comprovadas mais tarde. O registro do histórico das transações é denominado notarização.

1.5 O papel da Autoridade Certificadora

A autoridade certificadora (CA- *certificate authority*) deve garantir a Alice e ao Banco, através da assinatura de seus certificados, que tais entidades são realmente quem dizem ser. Então, a CA tem um papel básico de garantir a correspondência entre a identidade e a chave pública de uma determinada entidade, sabendo que tal chave

pública corresponde a uma chave privada que permanece sob guarda exclusiva dessa entidade.

Para tanto, a CA deve ser capaz de realizar todos os processos de emissão de certificados, verificação de validade, armazenamento, publicação ou acesso *on-lin*e, revogação e arquivamento para verificação futura. Em conseqüência, uma autoridade certificadora constitui-se de um sistema computacional completo, com capacidade de comunicação processamento e armazenamento. Além disso, tanto as comunicações envolvendo esse sistema, assim como o próprio sistema, devem ser também protegidos e a própria identidade do sistema deve ser garantida, necessidades esta que são atendidas por intermédio da publicação de uma chave pública pertencente à própria autoridade certificadora. Como tal chave deve também ser garantida com um certificado digital, então, em geral, uma autoridade certificadora deposita sua chave pública junto a outra autoridade certificadora, formando uma estrutura de certificação onde algumas CA funcionam como autoridades certificadoras para outras CAs .

2.AUTORIDADE CERTIFICADORA

Como mostrado anteriormente, um problema fundamental, o qual coloca em risco ambientes abertos com grande números de participantes que não se conhecem, é a questão da autenticidade da chave pública. Sem garantias adicionais, cada entidade usuária deverá desenvolver a sua verificação de autenticidade para cada chave pública das outras entidades com quem deseja comunicar-se antes de confiar nessas outras entidades. A complexidade desse problema pode ser reduzida pela certificação da chave pública de uma determinada entidade por intermédio de uma terceira parte em que ambos, emissor e receptor, confiem. Esta terceira parte, chamada de Autoridade Certificadora - CA, assina um certificado contendo a chave pública de uma entidade usuária, mais alguns dados adicionais tais como o nome da entidade , o período de validade do certificado, etc. A assinatura da CA é realizada com um algoritmo de assinatura digital usando a chave secreta da própria CA, de modo que tal assinatura pode ser verificada por qualquer entidade usando a chave pública da CA. O certificado assinado pela CA é chamado de Certificado Digital.

2.1 CA *root*

A necessidade de que uma CA tenha sua chave reconhecida ou assinada por uma outra CA leva à estruturação de uma hierarquia de CA para certificação de chaves públicas de criptografia. A CA que se encontra no topo de uma hierarquia tem uma característica especial: ela recorre a si mesma para certificar sua chave pública.

Vale observar que um certificado é gerado a partir de uma solicitação feita por alguma entidade a uma CA. A solicitação, após uma validação da identidade do solicitante através de algum processo com presença física ou com alguma troca de informação via rede, é atendida com a emissão do certificado contendo a assinatura da CA, assinatura esta que validará a identidade da entidade indicada no certificado.

No caso da emissão do certificado da CA de nível mais alto de uma hierarquia de CA *(CA roo*t), a própria autoridade assinará o seu certificado. Neste único caso, o

certificado é auto-assinado e, assim, o emissor do certificado é o mesmo que o receptor.

Algumas autoridades certificadoras e hierarquias de certificação encontram-se em operação para serviços de comunicação na Internet e especificamente para aplicações *web* que envolvem compras e pagamentos ou trocas de informações reservadas. Por essa razão, vários navegadores *web* são pré-configurados para confiar em autoridades certificadoras bem conhecidas, isto é , tais navegadores já vêm com os certificados de autoridades como a VeriSign, Thawte, dentre outras.

Por isso, ao se criar uma autoridade certificadora própria, faz-se necessário estabelecer um processo para enviar aos *browsers* o certificado dessa autoridade, habilitando os *browsers* para validar certificados assinados por tal autoridade certificadora.

2.2 Gerenciamento de chaves e certificados

Estabelecer uma autoridade certificadora é uma responsabilidade que requer assumir a administração de um banco de dados de certificados, estabelecer procedimentos técnicos e uma estrutura de gerenciamento de chaves. Autoridades certificadoras não apenas emitem certificados, mas também devem gerenciá-los, determinando a validade de cada certificado e as condições de renovação e armazenando uma lista de certificados que já foram emitidos mas não estão mais válidos, sendo uma lista desse tipo conhecida como uma Lista de Revogação de Certificados, ou CRL *(Certificate Revocation List*s).

A emissão de um certificado deve, ao menos em parte ocorrer de maneira *off-line* e não ser desenvolvida através de um mecanismo automático de solicitação/resposta. Antes da CA assinar um certificado, ela deve verificar os dados da solicitação, pois quando o certificado é gerado, a CA está garantindo que os dados do certificado são confiáveis.

É importante que a transferência de informação necessária à emissão do certificado para a autoridade certificadora não seja comprometida, e que a segurança física da CA seja garantida.

Enfim, para completar a gestão de certificados, uma operação de revogação deve ser provida pela CA. Os certificados devem ser revogados pela CA, gerando uma CRL *(Certificate Revocation Lis*t), ou seja, uma Lista de Revogação de Certificados, que justifica-se por vários motivos, dentro os quais se destacam :

- A chave secreta do usuário foi comprometida.
- Os dados pessoais do usuário foram modificados. Por exemplo, o usuário
mudou de organização.
- O usuário não deseja mais ser certificado por uma CA.
- O certificado da própria CA foi comprometido.
- O usuário violou a política de segurança da CA.

Deve-se observar que, depois de revogados, os certificados não deixam simplesmente de existir, mas apenas não podem mais ser utilizados, devendo ser arquivados para efeito de comprovação futura.

3. Certificados Digitais

O Certificado Digital pode ser considerado como a versão eletrônica (digital) de uma cédula de identidade. Ele associa uma chave pública com a identidade real de um indivíduo, de um sistema servidor, ou de alguma outra entidade .

No presente, a aplicação de certificados digitais segue os padrões da recomendação X.509, "*The Directory – Authentication Framework*", da ITU-T *(International Telecommunication Union – Telecommunications Standardization Sector*), recomendação esta que define uma estrutura de autenticação para sistemas abertos. Este faz uso da tecnologia de criptografia assimétrica, ou padrão de criptografia de chave pública PKCS *(Public Key Criptography Standard).*

3.1 Conteúdo do Certificado

Concretamente, um certificado é uma unidade de dados sub-dividida em vários campos de informação para armazenamento do conteúdo. Os seguintes campos são utilizados para representar um CertificadoX.509:

- **Número de versão *(version number*):** o campo número de versão é utilizado para indicar o tipo de formato do certificado em função da evolução dos padrões. O número de versão atual é 3.
- **Número de série (*serial number*):** o campo número de série fornece uma identificação única para cada certificado gerado por uma CA. Uma CA deve garantir que não existem dois certificados com o mesmo número de série. O número de série também é utilizado na revogação do certificado.
- **Assinatura *(signature*):** um valor de *hash* (resumo) do certificado é cifrado com a chave privada da CA para gerar a assinatura. O algoritmo de *hash* MD5 é bastante utilizado para tal fim.
- ***Nome (subject):*** o nome do proprietário do certificado está no formato de nome distinto, DN *(Distinguished Name*). O nome distinto é colocado na forma específica de um sistema de diretório X.500.

Uma autoridade certificadora deve definir uma política especifica, para utilização nos certificados emitidos por esta CA, quais os campos do DN que são opcionais, quais são obrigatórios, e quais devem ser iguais ao da CA. Os campos contidos em um DN são mostrados na Tab. 1 .

Tabela 1 Campos do DN X.500

Campo	Abreviação	Descrição
Nome Comum	CN	Nome sendo certificado
E-mail	E	E-mail
Organização	O	Nome da organização

Unidade Organizacional	OU	Unidade da organização
Localidade	L	Cidade
Estado ou Território	ST	Estado ou Território
País	C	Sigla do país (padrão ISO)

- **Nome do emissor *(issuer name*):** representação da identidade do emissor do certificado, na forma de DN. A identidade do emissor é usada para a validação do certificado tanto de cliente como do servidor, em uma transação segura na Internet.
- **Período de Validade *(validity perio*d):** um certificado tem a dada inicial e a data final do período de tempo em que pode ser utilizado. O intervalo de duração de certificados emitidos por uma determinada CA pode ser diferente de acordo com a necessidade do usuário e com as regras de política da Autoridade Certificadora.
- **Chave Pública:** um certificado tem um componente público (chave pública), que foi verificado por uma CA. Tal chave é calculada para utilização com um algoritmo de criptografia assimétrico, geralmente sendo utilizado o algoritmo de criptografia de chave pública RSA.

Além destes campos padrões existem outros campos chamados de campo de extensão. Estes campos podem ser utilizados como por exemplo para fornecer um nome alternativo para o proprietário do certificado, fornecer um comentário sobre a utilização do certificado, e etc. Estes campos são definidos pela Autoridade Certificadora.

4. IMPLEMENTAÇÃO DE UMA AUTORIDADE CERTIFICADORA PARA O CONTROLE DE ACESSO NA INTERNET

Para a implementação de uma autoridade certificadora que emitirá certificados digitais que serão utilizados no controle de acesso a páginas *web,* envolve a participação de 3 componentes principais: a estação cliente, que acessará o servidor *we*b, utilizando o protocolo HTTPS (este protocolo é uma implementação do protocolo HTTP sobre o protocolo SSL e serve-se, portanto, de certificados digitais para garantir conexões seguras, com privacidade, autenticidade de clientes e servidores, confiabilidade); o servidor *we*b, contendo as páginas que o cliente irá acessar; e o servidor da autoridade certificadora, contendo o banco de dados de certificados administrados por essa autoridade, além do conjunto de software que implementam os processos de emissão, armazenamento, validação, revogação, etc. relativos a tais certificados.

4.1 Autoridade Certificadora

Esta seção tem foco específico sobre os aspectos de implementação de uma autoridade certificadora e de certificados digitais para emprego nas transações entre os usuários via *we*b. Todo sistema foi desenvolvido utilizando o utilitário *openssl* , na plataforma OpenBSD 2.7, com *scripts* em C Shell e binários escritos em C++ quando a complexidade do processamento aumenta.

Para a implementação da CA, os seguintes passos podem ser seguidos:

- configurar o utilitário *openssl* de acordo com as regras da CA
- criar um banco de dados onde serão armazenados os certificados emitidos, os revogados, as CRLs e etc
- emitir um certificado auto-assinado para que a CA emita os seus certificados.

4.1.1 Configuração do *Openssl*

O arquivo de configuração do *Openssl* , ***openssl.cnf*** (/etc/ssl/openssl.conf), é utilizado para definir as características da CA, sendo divido em várias seções, como por exemplo:

• **Seção ca:** a seção ca define os diretórios de gerenciamento do banco de dados e alguns valores *default* para campos dos certificados:
• **Seção policy:** nesta seção é definida a política da autoridade certificadora, quais são os campos obrigatórios para a emissão dos certificados, quais são os campos do certificado que devem ser iguais aos da CA, etc
• **Seção req:** a seção req é usada quando da criação de um certificado e verificação de valores *defaults* e limites de comprimento para vários campos do *distinguished nam*e.

Além destas seções podem ser adicionadas outras para os campos de extensão.

4.1.2 Banco de dados da CA

O banco de dados da TesteCA, é o local onde estará armazenado todas as informações referentes a CA, certificados emitidos, CRLs, solicitações de certificados, o certificado da CA, sua chave privada e etc.

Este Banco de dados está dividido em diretórios. Alguns diretórios devem estar de acordo com os definidos no *openssl.cn*f, outros podem ser criados pelo próprio administrador da CA para o melhor gerenciamento dos usuários.

```
# ==========================================
# Criando os diretorios de gerenciamento
# ==========================================
CADIR=`pwd`/TesteCA
rm -rf ${CADIR}
mkdir ${CADIR}                          # dir da TesteCA
mkdir ${CADIR}/certs                    # dir dos certificados emitidos
mkdir ${CADIR}/crl                      # dir das CRLs
mkdir ${CADIR}/newcerts                 # default para certificados novos
mkdir ${CADIR}/private                  # a chave privada da TesteCA
echo "01" > ${CADIR}/serial             # o numero de serie do certificado
touch ${CADIR}/index.txt                # arquivo de indice do BD da TesteCA
mkdir ${CADIR}/request                  # solicitacoes de certificados
mkdir ${CADIR}/request/processed        # solicitacoes processadas
echo 1 > ${CADIR}/request/request.serial # o num. de serie das solicitacoes
```

4.1.3 Emissão do certificado da CA *root* TesteCA

Para a CA emitir certificados tanto dos clientes quanto dos servidores é necessário que ela possua um certificado digital. Neste caso, a própria TesteCA irá emitir seu certificado, um certificado auto-assinado.

```
openssl req -newkey 1024 -x509 -keyout ca.key -out ca.cert \
-config /etc/ssl/openssl.cnf –days 356
```

4.1.4 Instalação da TesteCA no navegador do cliente

O navegador da Netscape já vêm com os certificados de várias autoridades certificadoras bem conhecidas. Para que os certificados emitidos pela TesteCA possam ser utilizados em uma comunicação segura, o navegador tem que conhecer o certificado da TesteCA. De posse desse certificado, o navegador pode validar a assinatura nos certificados emitidos por tal CA.

O navegador importa certificados via protocolo HTTP. Há vários tipos de cabeçalhos MIME que indicam o tipo de certificado que está sendo importado.

Para o certificado da CA o tipo application/x-x509-ca-cert é utilizado.

```
#! /bin/sh
#
echo "Content-Type: application/x-x509-ca-cert"
echo
cat $DOCUMENT_ROOT/../TesteCA/cacert.pem
```

Para o *browser* da Netscape, o formato codificado do certificado da CA é PEM *(Privacy Enhaced Mail*) definido na RFC 1421 , codificado em base64, enquanto para o Internet Explorer da Microsoft, o formato é DER. *(Distinguished Encoding Rule*s) , que representa uma codificação única como *string* de *octets* (oito bits) para cada valor ASN.1, um padrão OSI.

Para passar um certificado no formato PEM em um no formato DER, pode utilizar o seguinte comando:

```
openssl base64 -d -in <nome do arquivo>.pem -out <nome do arquivo>.der
```

4.2 Autenticação e Configuração do Servidor *Web*

A autenticação do servidor *web* é realizada através da validação de seu certificado. Assim, para se realizar uma conexão segura pela URL https://nomededominio, é necessário obter um certificado para o servidor, e configurar as suas diretivas para utilizar este certificado.

4.2.1 Solicitação de um certificado para o Servidor Web

Os certificados do servidor e dos clientes sempre só podem ser emitidos após uma solicitação

```
openssl req -nodes -new -keyout newkey.pem \
-config /etc/ssl/openssl.cnf -days 365 -out newreq.pem
```

O resultado da solicitação do certificado é a geração de dois arquivos: o arquivo que contém a solicitação do certificado codificada e o arquivo com a chave secreta:

Arquivo com a solicitação do certificado codificada no formato PEM *(newreq.pe*m) :

```
-----BEGIN CERTIFICATE REQUEST-----
MIIBvjCCAScCAQAwfjELMAkGA1UEBhMCYnIxETAPBgNVBAgTCGJyYXNpbGlhMQsw
CQYDVQQHEwJkZjEMMAoGA1UEChMDdW5iMQwwCgYDVQQLEwNlbmUxHjAcBgNVBAMT
FWxvY2FsaG9zdC5sb2NhbGRvbWFpbjETMBEGCSqGSIb3DQEJARYEcm9vdDCBnzAN
BgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAyFGV0SGq4W6wvXGoT5mWnuOmKMOLJuzA
43bUK7DE8f63EIqJIZrQ15/XOlKFSmlGB90FhM4d7k3Qhtq1rnEQ5K7mWHOd1On7
Ku3SogwoYJPDdVIKpTW15toQ65oUHWzT7HVndYTDxg87ntcAFUxZ4LTh+hxHwv4t
/Gw2mGmcHrECAwEAAaAAMA0GCSqGSIb3DQEBBAUAA4GBACbQMO/QOoSY3bRVw+lX
iQffgs1i/fE8o/qIGuqhHv33M9R4qpCcKUeBr/HWLvgsuOga1J/0hM3hLyTn07q5
Bc+5yOex3ScicfI/HoXMY/MGemjBxxobqwXANOnuB3YgVT7wDA2cX7r9RsuqU7f1
71QV3VkaInXe2XVzNqe4isGP
-----END CERTIFICATE REQUEST-----
```

Arquivo com a chave secreta do servidor também no codificada no formato PEM *(newkey.pe*m):

```
-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQDIUZXRIarhbrC9cahPmZae46Yow4sm7MDjdtQrsMTx/rcQiokh
mtDXn9c6UoVKaUYH3QWEzh3uTdCG2rWucRDkruZYc53U6fsq7dKiDChgk8N1Ugql
NbXm2hDrmhQdbNPsdWd1hMPGDzue1wAVTFngtOH6HEfC/i38bDaYaZwesQIDAQAB
AoGBAJfwce9fVwqId9wpkl1WBM3dpBF1SF2s5B1j60h8WtWmutn8CkmH8PIsFgbx
wpIPSYP7XA1AfalcMlenGnkG36UMCv8izBz9BLbcF4CiwPtoGMEnrC7082wU+128
J5jd26WLf4MaJGw0tY06KTK/MBt0klNv/cNFT3kFlw508NOxAkEA+MoZA7SLsHHD
fxk65N0L57E1jE8qi/8HFB1i9XEMtIvH9JtX2bBNA94fZvgFb6p/IB89SvR7bI31
pdjvtiXkdQJBAM4f22UUYIB/7xF/X9wcnVE4Q9FtGocYIuasAyx54YCngKfugmyS
w05rS38ZHoBeD06JaxdT7qjKfx/PZ/yJ2c0CQEvlk3X+6TJ9si8e6kmRaaVX+UcB
i7BDr8wiHuNThpW5otlrXoSJdSU96QSHm1jygW+yOAeax+BY+lyK7q35p1UCQQC1
/dqwPHWX/10NjfENXxZnh84ZMsOhU+Y81NNYiaRY+4vUufnd8xcOPh3CcmND48iQ
BzTOvASUc3Yal/2wM+ZhAkBhDq/IpcUmkCtcWhmF2mHL9DCEFU8DuUaAeTB21Dt5
5H/ZhmMNIZn0f6aIf/ripU4+xme1NWccCrPT3veJmT/l
-----END RSA PRIVATE KEY-----
```

4.2.2 Emitindo um certificado para o Servidor *Web*

Um certificado é gerado a partir da assinatura da autoridade certificadora em uma solicitação feita anteriormente.

```
openssl ca -policy policy_anything -out newcert.pem \
-config /etc/ssl/openssl.cnf -days 365 -infiles newreq.pem
```

Resulta desse comando um certificado emitido codificado no formato texto, e em seguida no formato PEM (arquivo *newcert.pe*m):

```
issuer
:/C=BR/ST=Rio de Janeiro /L=RJ/O=COPPE/UFRJ/OU=Lab. Ravel/CN=RAVEL Root CA
/Email=ca_root@ravel.ufrj.br
subject:/C=BR/ST=Rio de Janeiro /L=RJ/O=COPPE/UFRJ/OU=Lab. Ravel/CN=Ravel Roor
CA/Email=root_ca@ravel.ufrj.br
root
serial :03

Certificate:
      Data:
            Version: 3 (0x2)
            Serial Number: 3 (0x3)
            Signature Algorithm: md5WithRSAEncryption
            Issuer: C=br, ST=brasilia, L=df, O=unb, OU=ene,
CN=CampusVirtualCA/Email=root@localhost.localdomain
      Validity
            Not Before: Apr 21 08:31:18 1999 GMT
            Not After : Apr 20 08:31:18 2000 GMT
      Subject: C=RJ, ST=Rio de Janeiro, L=RJ, O=COPPE/UFRJ, OU=Lab. Ravel,
CN=www.ravel.ufrj.br/Email=https@ravel.ufrj.br
      Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                  Modulus (1024 bit):
                        00:c8:51:95:d1:21:aa:e1:6e:b0:bd:71:a8:4f:99:
                        96:9e:e3:a6:28:c3:8b:26:ec:c0:e3:76:d4:2b:b0:
                        c4:f1:fe:b7:10:8a:89:21:9a:d0:d7:9f:d7:3a:52:
                        85:4a:69:46:07:dd:05:84:ce:1d:ee:4d:d0:86:da:
                        b5:ae:71:10:e4:ae:e6:58:73:9d:d4:e9:fb:2a:ed:
                        d2:a2:0c:28:60:93:c3:75:52:0a:a5:35:b5:e6:da:
                        10:eb:9a:14:1d:6c:d3:ec:75:67:75:84:c3:c6:0f:
                        3b:9e:d7:00:15:4c:59:e0:b4:e1:fa:1c:47:c2:fe:
                        2d:fc:6c:36:98:69:9c:1e:b1
                  Exponent: 65537 (0x10001)
            X509v3 extensions:
                  Netscape CA Revocation Url:
                        http://www.ravel.ufrj.br/crl/crl.pem
                  Netscape Comment:
                        This is a comment
                  Netscape Cert Type:
                        0x40
      Signature Algorithm: md5WithRSAEncryption
            7f:0a:38:e9:a8:63:f2:27:68:da:74:46:f5:31:ea:5c:b2:27:
```

```
            51:38:db:39:cf:db:3e:d9:bf:d7:eb:c2:ed:05:c6:e7:e1:f9:
            e7:02:4c:c6:e4:dd:2d:fa:e2:d0:9d:63:38:aa:b1:fc:e1:33:
            75:d3:17:d1:84:94:f7:f3:f9:ee:89:5d:ce:7e:93:61:d2:e2:
            99:7b:dd:7a:66:e1:f4:af:de:eb:f1:12:d5:f1:1f:31:96:46:
            e4:89:3d:b1:39:b5:3a:c8:14:d4:dd:59:88:20:9a:7e:b4:8b:
            8a:06:75:cf:d0:84:64:6b:b9:ce:53:3f:b5:c9:41:b4:27:03:
            45:13
-----BEGIN CERTIFICATE-----
MIIC7jCCAlegAwIBAgIBAzANBgkqhkiG9w0BAQQFADCBjjELMAkGA1UEBhMCYnIx
ETAPBgNVBAgTCGJyYXNpbGlhMQswCQYDVQQHEwJkZjEMMAoGA1UEChMDdW5iMQww
CgYDVQQLEwNlbmUxGDAWBgNVBAMTD0NhbXB1c1ZpcnR1YWxDQTEpMCcGCSqGSIb3
DQEJARYacm9vdEBsb2NhbGhvc3QubG9jYWxkb21haW4wHhcNOTkwNDIxMDgzMTE4
WhcNMDAwNDIwMDgzMTE4WjB+MQswCQYDVQQGEwJicjERMA8GA1UECBMIYnJhc2ls
aWExCzAJBgNVBAcTAmRmMQwwCgYDVQQKEwN1bmIxDDAKBgNVBAsTA2VuZTEeMBwG
A1UEAxMVbG9jYWxob3N0LmxvY2FsZG9tYWluMRMwEQYJKoZIhvcNAQkBFgRyb290
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDIUZXRIarhbrC9cahPmZae46Yo
w4sm7MDjdtQrsMTx/rcQiokhmtDXn9c6UoVKaUYH3QWEzh3uTdCG2rWucRDkruZY
c53U6fsq7dKiDChgk8N1UgqlNbXm2hDrmhQdbNPsdWd1hMPGDzue1wAVTFngtOH6
HEfC/i38bDaYaZwesQIDAQABo2swaTAyBglghkgBhvhCAQQEJRYjaHR0cDovL3d3
dy5jcnlwdHNvZnQuY29tL2NhLWNybC5wZW0wIAYJYIZIAYb4QgENBBMWEVRoaXMg
aXMgYSBjb21tZW50MBEGCWCGSAGG+EIBAQQEAwIAQDANBgkqhkiG9w0BAQQFAAOB
gQB/CjjpqGPyJ2jadEb1MepcsidRONs5z9s+2b/X68LtBcbn4fnnAkzG5N0t+uLQ
nWM4qrH84TN10xfRhJT38/nuiV3OfpNh0uKZe916ZuH0r97r8RLV8R8xlkbkiT2x
ObU6yBTU3VmIIJp+tIuKBnXP0IRka7nOUz+1yUG0JwNFEw==
-----END CERTIFICATE-----
```

O Protocolo SSL utiliza os dados que estão entre o `-----BEGIN CERTIFICATE-----` e o `-----END CERTIFICATE-----` , que nada mas é que todos os dados do certificado, como o DN, a chave pública, e etc., que estão no formato texto codificado no formato PEM. É comum armazenar apenas o certificado no formato PEM, acima está o certificado no formato texto com o certificado no formato PEM, apenas para ilustração.

4.2.3 Configuração do Servidor Web Apache

O servidor *web* Apache tem que ser configurado (arquivo httpd.conf) para utilizar o certificado do servidor na autenticação do Protocolo SSL. Estas são as diretivas configuradas:

```
# onde esta armazenado o certificado do servidor
SSLCertificateFile /etc/ssl/site.crt
# onde esta a chave privada do servidor
SSLCertificateKeyFile /etc/ssl/private/site.key
# o diretorio onde esta o certificado CA
SSLCACertificatePath /etc/ssl
# o arquivo onde esta o certificado da CA
SSLCACertificateFile /etc/ssl/ca.crt
# o diretorio onde o servidor verifica a CRL
SSLCARevocationPath /etc/ssl/crl
# o arquivo onde o servidor verifica a CRL
SSLCARevogationFile /etc/ssl/crl/ca.crl
```

Para gerar uma CRL que será utilizada para validação do certificado digital pelo servidor o *we*b, o comando e este:

```
openssl ca -gencrl -config /etc/ssl/openssl.cnf \
-key abcd1234 -out /etc/ssl/crl/ca.crl
```

4.3 Certificado Pessoal

Um Certificado Pessoal é usado na autenticação do cliente para o servidor. A emissão de um certificado é dividida em três etapas:

1. Preenchimento de um formulário de solicitação do certificado
2. Assinatura da CA e emissão do certificado.
3. Notificação para o usuário através do e-mail.
4. Recebimento do certificado pessoal.

As demonstrações seguintes só funcionam para os navegadores da Netscape.

4.3.1 Solicitação do Certificado Pessoal

Para solicitar o certificado, o cliente deve preencher um formulário em HTML com os dados necessários para a criação de um certificado. Estes dados são os que constituem um *Distinghished Name* (DN) e uma chave pública. Este formulário de solicitação é diferente para os *browsers* Netscape e Internet Explorer, pois o processo de geração do par de chaves é diferente. O par de chaves, pública (que será assinada pela autoridade) e a privada do cliente é gerado no Netscape pela TAG HTML KEYGEN, com a seguinte sintaxe:

```
<KEYGEN NAME=”nome” >
```

Esta nova TAG foi desenvolvida para ser usada em sistemas de gerenciamento de certificados na *we*b. Tal TAG deve estar dentro de um formulário HTML (TAG FORM). Ela automaticamente detecta o tamanho da chave que o browser suporta e exibe os tamanhos possíveis como uma lista, onde os valores são em geral: 512 bit, 768 bit, e 1024 bit. A TAG KEYGEN gera um par de chaves: a chave secreta, que é cifrada e armazenada no banco de dados local do *browse*r, e a chave pública enviada em uma variável ao submeter o formulário.

Os dados são recebidos pelo CGI e armazenados em arquivos que contém o par <nome da variavel> = <valor da variável> ,um por linha para futuro processamento.

4.3.2 Emissão de Certificados Pessoais

Quando uma autoridade certificadora assina uma solicitação de certificado, gerando um certificado, ela está em princípio garantindo que aquele usuário é realmente quem diz ser. Por isso, a emissão do certificado deve ser feita após a

verificação dos dados informados no formulário de solicitação, ou até mesmo exigindo a presença física e a assinatura da pessoa que está solicitando o certificado.

Com os dados obtidos do formulário de solicitação de certificado (gerados pela req-cert-ns.cgi), é necessário um comando ca para assinar a chave pública criada. O comando para a assinatura do certificado é:

```
openssl ca -spkac $request -keyfile $CAroot/private/cakey.pem \
-outdir $CAroot/certs -config /etc/ssl/openssl.cnf
```

Para que seja solicitado o certificado do cliente pela CGI, é necessário configurar o servidor *web* para que a diretiva SSLVerifyClient com o seguinte valor:

```
<Location /assina-ca.cgi>
      SSLVerifyClient require
      SSLRequireSSL
</Location>
```

A diretiva SSLVerifyClient pode ter os seguintes valores:

- **none:** o certificado do cliente não é requerido
- **optional:** o cliente pode apresentar um certificado válido.
- **require:** o cliente tem que apresentar um certificado válido.
- **optional_no_ca:** o cliente pode apresentar um certificado que não seja verificado com sucesso.

Na prática, apenas o nível **none** e **require** são realmente utilizados. O nível **opcional** não funciona em todos os *browsers* e o nível de **opcional_no_ca** é na realidade uma vulnerabilidade ao que se refere ao conceito de autenticação.

4.3.3 Instalação do Certificado Pessoal.

O cliente deve baixar o seu certificado utilizando o mesmo browser que fez a solicitação pois é nele que está a chave privada, par da chave pública que gerou o certificado. Para tanto, o cliente deve acessar uma URL emitida pela CA por e-mail na qual se encontrarão os elementos para captura dos dados do certificado.. A instalação é automática. Depois de instalado, é possível exportar o certificado de um *browser* e importar em um outro.

Para o Netscape, cria-se uma CGI que tenha como tipo MIME application/x-x509-user-cert. Assim o *browser* irá entender que o arquivo é um certificado pessoal que deve ser instalado.

4.3.4 Revogação de Certificados

Para revogar um certificado utiliza-se o comando:

```
openssl ca -revoke 04.pem \
-config /etc/ssl/openssl.cnf -key abcd1234
```

Assim que o certificado for revogado é necessário gerar uma nova CRL e atualizar a CRL que foi configurada no servidor *we*b, assim a próxima vez que o certificado revogado tentar acessar a página terá o seu acesso negado.

5. Conclusões

A segurança nas transferências de dados via Internet é uma efetiva necessidade hoje em dia, em consideração às transações com números de cartão de crédito, contas bancárias, e outros dados sensíveis, que estão sendo transportados via *we*b, em um número crescente.

A implementação de uma estrutura de comunicação segura via Internet, com a utilização do servidor *web* Apache e a biblioteca *openssl* sendo empregadas para a construção de uma autoridade certificadora, que emite certificados digitais para autenticação de clientes e servidores *we*b, mostrou-se uma ótima alternativa.

Referência:

HIRSCH, Frederick J., *Introducing SSL and Certificates using SSLeay,* 1997. Published in World Wide Web Journal, Summer .

ITU-T, *Information Technology Open Systems Interconnection the Directory: Authentication Framework – Recommendation X.509,* 1997.

RSA Laboratories. *A Layman's Guide to a Subset of ASN.1, BER, and DE*R, 1993.

RSA Laboratories. An *Overview of the PKCS Standard*s, 1993.

SCHNEIER, Bruce. *Applied cryptograph*y, 2 nd Edition – John Wiley & Sons, 1996.

STALLINGS, William. *Cryptography and Network Security Principles and Practice,* 2 nd Edition – Prentice Hall, 1999.

IETF, RFC 1321 - *The MD5 Message-Digest Algorithm,* 1992.

IETF, RFC 1421 - *Privacy Enhancement for Internet Eletronic Mail : Part I : Message Encryption and Authentication Procedures,* 1993.

IETF, RFC 1422 - *Privacy Enhancement for Internet Eletronic Mail : Part II : Certificate-Based Key Management,* 1993.

IETF, RFC 1423 - *Privacy Enhancement for Internet Eletronic Mail : Part III : Algorithms, Modes, and Identifiers,* 1993.

IETF, RFC 1424 - *Privacy Enhancement for Internet Eletronic Mail : Part IV : Key Certification and Related Services,* 1993.

IETF, RFC 1945 - *Hypertext Transfer Protocol – HTTP/1.0* , 1996.

IETF, *The SSL protocol,* Version 2.*0,* 1995.

IETF, *The SSL protocol,* Version 3.*0,* 1996.