BandBBBTec

Serviços de Rede

Protocolos

SSH / TELNET

TELNET – TELecommunication NETwork

O protocolo **TELNET**, originalmente definido na **RFC 854**, permite o acesso a um computador remoto na forma de emulação de terminal.

Por isso que, normalmente, os clientes **TELNET** são denominados de “emuladores de terminal”. Isso se deve aos primórdios dos computadores onde o **TELNET** era utilizado para que um terminal “burro” (dumb terminal) pudesse se conectar a um computador central (mainframe). Isso permite a um cliente remoto interagir com um computador central (servidor) como se estivesse localmente conectado ao mesmo.

No processo de conexão, é utilizado um usuário e senha para conexão. Muitos dispositivos de rede (roteadores, comutadores, repetidores, impressoras, etc) utilizam esse protocolo para permitir acesso e configuração remota. O daemon do **TELNET** possui os seguintes padrões de funcionamento:

Camada: Aplicação
Protocolo: TCP
Porta Padrão: 23
Modelo de Operação: Cliente-Servidor

Por outro lado, o **TELNET** pode ser usado como uma ferramenta de testes de conectividade pois permite a alteração d aporta de conexão remota de modo que se pode testar se uma porta está repondendo (**LISTEN**) ou não.

Sintaxe do TELNET: **telnet <endereço IP ou hostname do ativo de rede> <porta de conexão>**

Comandos do TELNET: close, display, environ, logout, mode, open, quit, send, set, unset, ?

Pontos Positivos: simples, eficiente, payload leve, fácil utilização

Pontos Negativos: sem criptografica, utiliza /etc/passwd. Senha transmitida em texto puro

Criptografia

"Se uma chave não for requerida, você não terá criptografia, terá uma codificação"

Criptografia é a ciência e arte de escrever mensagens em forma cifrada ou em código. É parte de um campo de estudos que trata das comunicações secretas, usadas, dentre outras finalidades, para:

- ✓ autenticar a identidade de usuários;
- ✓ autenticar e proteger o sigilo de comunicações pessoais e de transações comerciais e bancárias;
- ✓ proteger a integridade de transferências eletrônicas de fundos

Uma mensagem codificada por um método de criptografia deve ser **privada**, ou seja, somente aquele que enviou e aquele que recebeu devem ter acesso ao conteúdo da mensagem. Além disso, uma mensagem deve poder ser **assinada**, ou seja, a pessoa que a recebeu deve poder verificar se o remetente é mesmo a pessoa que diz ser e ter a capacidade de identificar se uma mensagem pode ter sido modificada

Os métodos de criptografia atuais são seguros e eficientes e baseiam-se no uso de uma ou mais **chaves**. A chave é uma seqüência de caracteres, que pode conter letras, dígitos e símbolos (como uma senha), e que é convertida em um número, utilizado pelos métodos de criptografia para codificar e decodificar mensagens

Atualmente, os métodos criptográficos podem ser subdivididos em duas grandes categorias, de acordo com o tipo de chave utilizada: a criptografia de chave única e a criptografia de chave pública e privada

Criptografia de Chave Única (SIMETRICA)

A criptografia de chave única utiliza a mesma chave tanto para codificar quanto para decodificar mensagens.

Apesar deste método ser bastante eficiente em relação ao tempo de processamento, ou seja, o tempo gasto para codificar e decodificar mensagens, tem como principal desvantagem a necessidade de utilização de um meio seguro para que a chave possa ser compartilhada entre pessoas ou entidades que desejem trocar informações criptografadas

Criptografia de Chaves Pública e Privada (Assimetrica)

A criptografia de chaves pública e privada também é conhecida como **asymmetric cryptography** ou **public key cryptography (PKI)** e utiliza duas chaves distintas, uma para codificar e outra para decodificar mensagens. Neste método cada pessoa ou entidade mantém duas chaves: uma pública, que pode ser divulgada livremente, e outra privada, que deve ser mantida em segredo pelo seu dono. As mensagens codificadas com a chave pública só podem ser decodificadas com a chave privada correspondente

Seja o exemplo, onde José e Maria querem se comunicar de maneira sigilosa. Então, eles terão que realizar os seguintes procedimentos:

- ❖ José codifica uma mensagem utilizando a chave pública de Maria, que está disponível para o uso de qualquer pessoa;
- ❖ Depois de criptografada, José envia a mensagem para Maria, através da Internet;
- ❖ Maria recebe e decodifica a mensagem, utilizando sua chave privada, que é apenas de seu conhecimento;
- ❖ Se Maria quiser responder a mensagem, deverá realizar o mesmo procedimento, mas utilizando a chave pública de José

Apesar deste método ter o desempenho bem inferior em relação ao tempo de processamento, quando comparado ao método de criptografia de chave única, apresenta como principal vantagem a livre distribuição de chaves públicas, não necessitando de um meio seguro para que chaves sejam combinadas antecipadamente. Além disso, pode ser utilizado na geração de assinaturas digitais

Exemplos de algoritmos de chave simétrica

Cifra	Autor	Tamanho da Chave (bits)	Comentários
Blowfish	Bruce Schneier	1 a 448	Velho e lento
DES	IBM	56	Muito fraco para usar agora
IDEA	Massey e Xuejia	128	Bom, mas patenteado
RC4	Ronald Rivest	1 a 1024	Algumas chaves são fracas
RC5	Ronald Rivest	128 a 256	Bom, mas patenteado
Rijndael	Daemen e Rijmen	128 a 256	Melhor escolha
Serpent	Anderson, Biham, Knudsen	128 a 256	Muito forte
DES triplo	IBM	168	Segunda melhor escolha
Twofish	Bruce Schneier	128 a 256	Muito forte, amplamente utilizado

Fonte: Tanenbaum

Intermediários Confiáveis

Problema chave simétrica:

Como duas entidades estabelecem uma chave secreta compartilhada pela rede?

Solução:

Centro de distribuição de chaves confiável (KDC - Key Distribution Center) atuando como intermediário entre as entidades.

Problema chave pública:

Quando Alice obtém a chave pública de Bob (a partir de um *web site*, e-mail, disquete), qual é a garantia que essa é realmente a chave pública de Bob, não a de Trudy?

Solução :

Autoridade de certificação confiável (CA - Certification Authority)

Assinatura Digital

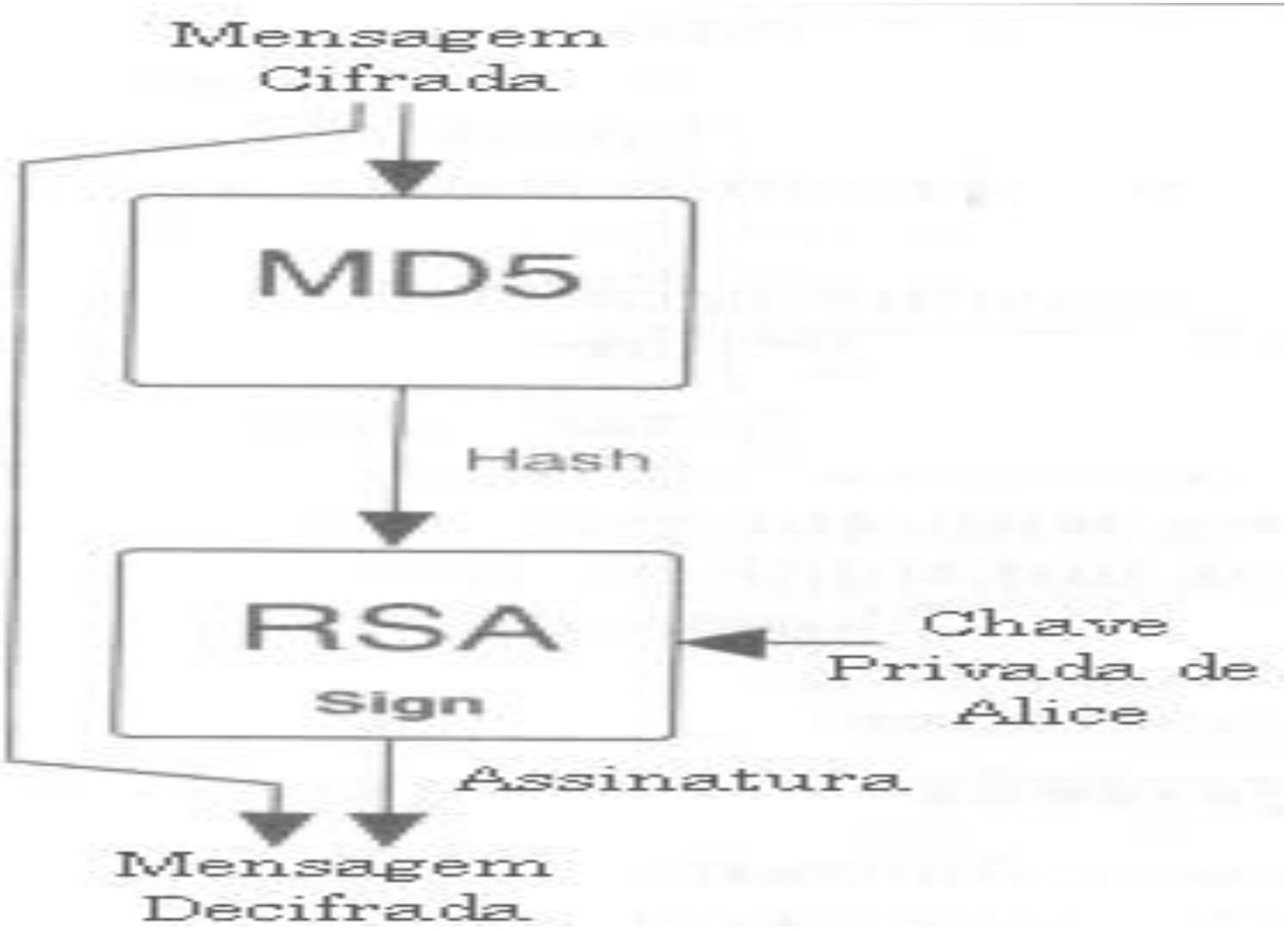
A assinatura digital consiste na criação de um código, através da utilização de uma chave privada, de modo que a pessoa ou entidade que receber uma mensagem contendo este código possa verificar se o remetente é mesmo quem diz ser e identificar qualquer mensagem que possa ter sido modificada

Desta forma, é utilizado o método de criptografia de chaves pública e privada, mas em um processo inverso ao apresentado no slide anterior

Se José quiser enviar uma mensagem assinada para Maria, ele codificará a mensagem com sua chave privada. Neste processo será gerada uma assinatura digital, que será adicionada à mensagem enviada para Maria. Ao receber a mensagem, Maria utilizará a chave pública de José para decodificar a mensagem. Neste processo será gerada uma segunda assinatura digital, que será comparada à primeira. Se as assinaturas forem idênticas, Maria terá certeza que o remetente da mensagem foi o José e que a mensagem não foi modificada

É importante ressaltar que a segurança do método baseia-se no fato de que a chave privada é conhecida apenas pelo seu dono. Também é importante ressaltar que o fato de assinar uma mensagem não significa gerar uma mensagem sigilosa. Para o exemplo anterior, se José quisesse assinar a mensagem e ter certeza de que apenas Maria teria acesso a seu conteúdo, seria preciso codificá-la com a chave pública de Maria, depois de assiná-la

Assinatura Digital



Exemplos

Exemplos que combinam a utilização dos métodos de criptografia de chave única e de chaves pública e privada são as conexões seguras, estabelecidas entre o *navegador (browser)* de um usuário e um *site*, em transações comerciais ou bancárias via *Web*

Estas conexões seguras via *Web* utilizam o método de criptografia de chave única, implementado pelo protocolo SSL (*Secure Socket Layer*) transmitido via camada de Transporte via HTTPS. O *browser* do usuário precisa informar ao *site* qual será a chave única utilizada na conexão segura, antes de iniciar a transmissão de dados sigilosos

Para isto, o *browser* obtém a chave pública do certificado da instituição que mantém o *site*. Então, ele utiliza esta chave pública para codificar e enviar uma mensagem para o *site*, contendo a chave única a ser utilizada na conexão segura. O *site* utiliza sua chave privada para decodificar a mensagem e identificar a chave única que será utilizada

A partir deste ponto, o *browser* do usuário e o *site* podem transmitir informações, de forma sigilosa e segura, através da utilização do método de criptografia de chave única. A chave única pode ser trocada em intervalos de tempo determinados, através da repetição dos procedimentos descritos anteriormente, aumentando assim o nível de segurança de todo o processo

Algoritmos de Assinatura Digital e Chaves

RSA

Foi desenvolvido por Ron **R**ivest, Adi **S**hamir e Len **A**dleman (1978). É baseado na dificuldade computacional de se fatorar um número inteiro muito grande em seus fatores primos

É o algoritmo mais popular e fácil de se implementar. Sua segurança é baseada na dificuldade de fatoração do produto de dois números primos extremamente grandes, da ordem de 100 dígitos decimais, em seus fatores originais

O par de chaves usadas neste algoritmo é gerado com base em dois números primos aleatórios grandes, donde se calculam as chaves privada e pública. Depois que o par de chaves é gerado, os dois números iniciais são descartados

Para encriptar uma mensagem usa-se uma das chaves, sendo que a decriptação somente poderá ser feita pela outra chave. A segurança deste algoritmo reside na dificuldade computacional de fatoração em seus fatores primos de um número muito grande, acima de 2048 bits ou 200 dígitos decimais
1024
Este algoritmo é largamente empregado em vários softwares que provêm criptografia assimétrica, e tem-se provado bastante seguro e facilmente escalável para uso de chaves cada vez maiores.

Além de codificar chaves, o algoritmo de chave pública RSA pode ser também usado para gerar assinaturas digitais. As matemáticas são as mesmas se você utiliza o RSA para administração de chaves ou assinaturas digitais: existe uma chave pública e privada, e a segurança do sistema está baseado na dificuldade de fatorar números primos grandes

DSA

O DSA é um algoritmo de assinatura digital (Digital Signature Algorithm). (Também existe um Padrão de Assinatura Digital (Digital Signature Standard), ou DSS. O padrão implementa o algoritmo - realmente é só uma diferença na terminologia.) é um algoritmo de chave pública, mas só pode ser usado para assinaturas digitais. O DSA foi inventado no NSA e é patenteado pelo governo norte-americano. O NIST aprovou o DSA como um padrão de assinatura digital federal. Quando o DSA foi primeiramente proposto, a Indústria RSA de Segurança de Dados e companhias que já tinham autorizado o algoritmo RSA lançaram uma campanha contra o DSA. Eles aludiram uma possível "porta de armadilha" que permitiria ao governo quebrar o DSA. Eles se queixaram da velocidade do algoritmo, o fato que não pode ser usado para cifragem e o tamanho da chave.

O único ponto de segurança é o tamanho da chave. Quando o padrão foi primeiramente proposto, o tamanho da chave era de 512 bits, o que era muito pequeno. O padrão final permite chaves de até 1024 bits o que é mais que suficiente para necessidades de segurança de qualquer um. O DSA obtém sua segurança do problema de logaritmo discreto. As matemáticas são muito diferentes do RSA, mas a segurança é semelhante para chaves semelhantes em tamanho. Embora exceções são teoricamente possíveis, é provável que qualquer grande avanço na quebra do RSA também implique um avanço semelhante na quebra do DSA, e vice-versa. Não existe nenhuma vantagem na segurança usando um algoritmo em cima do outro. Ocasionalmente comparações entre a velocidade e eficiência entre os dois algoritmos são publicadas. Não surpreendentemente, os publicados pela Indústria RSA de Segurança de Dados mostram que o RSA é melhor que o DSA e os publicados pelo NIST mostram que o DSA é melhor que o RSA. Com o RSA, leva-se mais tempo para assinar uma mensagem que verifica uma assinatura. Com o DSA, ambas operações de assinatura e de verificação levam a mesma quantia de tempo. Em realidade, estas diferenças são secundárias; com a finalidade da segurança de um programa de correio eletrônico elas são desprezíveis. Atualmente nenhum programa de segurança de correio eletrônico utiliza o DSA

X.509

Refere-se ao IETF's PKI Certificado e LCR Perfil dos padrões certificados X.509 versão 3, como determinado na RFC 3280 especificações

Um certificado digital X.509 versão 3 tem três principais variáveis:

- ✓ o certificado
- ✓ o algoritmo de assinatura
- ✓ a assinatura

O certificado é descrito por atributos como a versão, algoritmo de identificação, número de série, emitente do título, assunto, validade, informações da chave pública, extensões e várias outras informações, como o proprietário e identificador do emitente

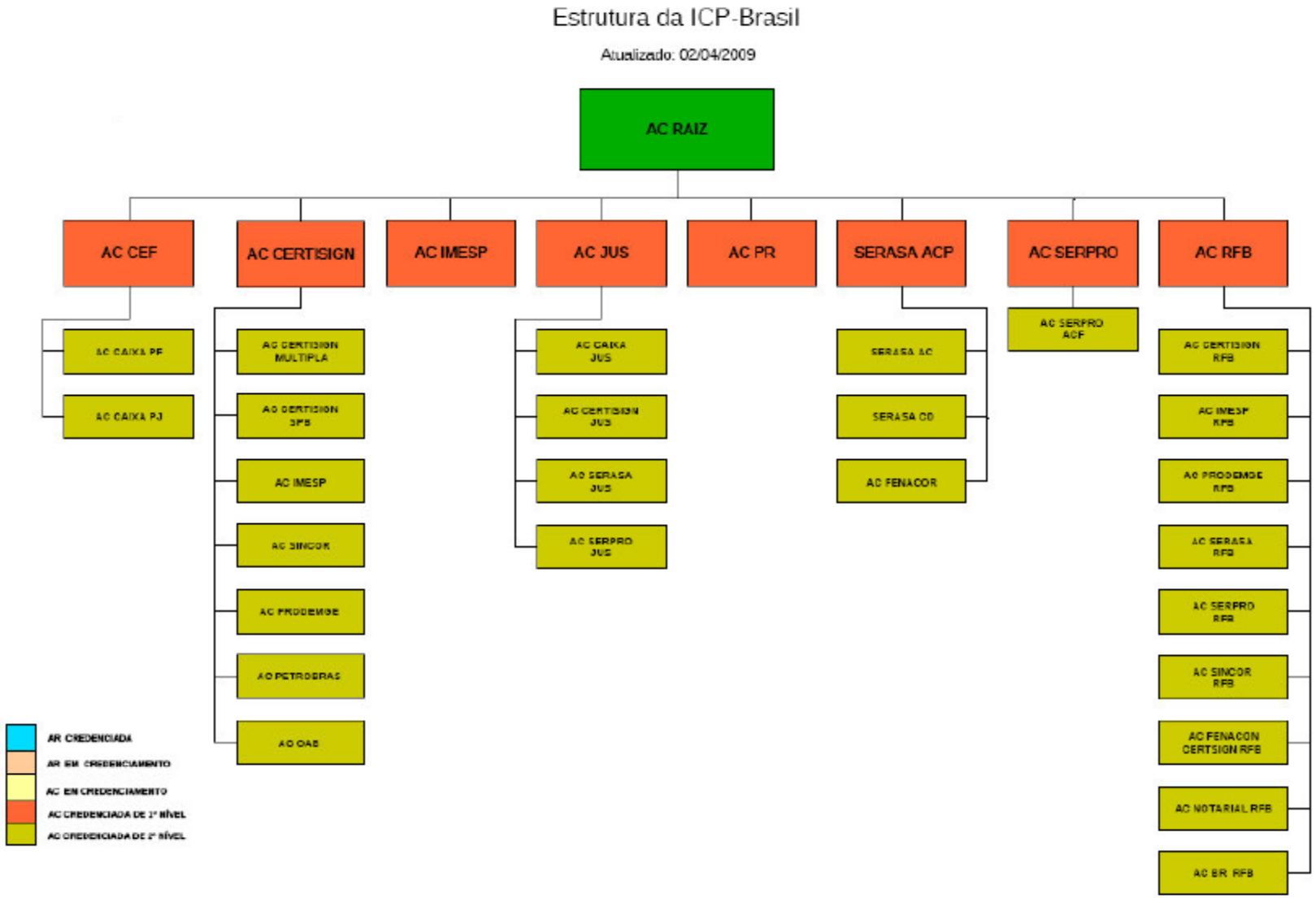
ICP Brasil – Infraestrutura de Chaves Públicas

A ICP-Brasil oferece duas categorias de certificados digitais: A e S, sendo que cada uma se divide em quatro tipos: A1, A2, A3 e A4; S1, S2, S3 e S4. A categoria A é direcionada para fins de identificação e autenticação, enquanto que o tipo S é direcionado a atividades sigilosas. Vejas as características que tornam as versões de ambas as categorias diferentes entre si:

✓ A1 e S1: geração das chaves é feita por software; chaves de tamanho mínimo de 1024 bits; armazenamento em dispositivo de armazenamento (como um HD); validade máxima de um ano;
✓ A2 e S2: geração das chaves é feita por software; chaves de tamanho mínimo de 1024 bits; armazenamento em cartão inteligente (com chip) ou token (dispositivo semelhante a um pendrive); validade máxima de dois anos;
✓ A3 e S3: geração das chaves é feita por hardware; chaves de tamanho mínimo de 1024 bits; armazenamento em cartão inteligente ou token; validade máxima de três anos;
✓ A4 e S4: geração das chaves é feita por hardware; chaves de tamanho mínimo de 2048 bits; armazenamento em cartão inteligente ou token; validade máxima de três anos

Os certificados A1 e A3 são os mais utilizados, sendo que o primeiro é geralmente armazenado no computador do solicitante, enquanto que o segundo é guardado em cartões inteligentes (smartcards) ou tokens protegidos por senha

ICP Brasil – Estrutura



Tamanho de Chaves

Os métodos de criptografia atualmente utilizados, e que apresentam bons níveis de segurança, são publicamente conhecidos e são seguros pela robustez de seus algoritmos e pelo tamanho das chaves que utilizam

Para que um atacante descubra uma chave ele precisa utilizar algum método de força bruta, ou seja, testar combinações de chaves até que a correta seja descoberta. Portanto, quanto maior for a chave, maior será o número de combinações a testar, inviabilizando assim a descoberta de uma chave em tempo hábil. Além disso, chaves podem ser trocadas regularmente, tornando os métodos de criptografia ainda mais seguros

Atualmente, para se obter um bom nível de segurança na utilização do método de criptografia de chave única, é aconselhável utilizar chaves de no mínimo 128 bits. E para o método de criptografia de chaves pública e privada é aconselhável utilizar chaves de 2048 bits, sendo o mínimo aceitável de 1024 bits. Dependendo dos fins para os quais os métodos criptográficos serão utilizados, deve-se considerar a utilização de chaves maiores: 256 ou 512 bits para chave única e 4096 ou 8192 bits para chaves pública e privada

Função de HASH

Uma função de hash de uma só via converte uma mensagem de tamanho arbitrário em um hash de tamanho fixo. Este é um dos outros truques de criptografia. Como um algoritmo de cifragem, uma função de hash de uma só via converte uma mensagem de texto plano em palavras sem nexo. Porém, distinto de um algoritmo de cifragem, não existe nenhuma maneira de voltar para trás com uma função de hash de uma só via. Com a chave correta, a pessoa pode decifrar um texto cifrado codificado com um algoritmo de cifragem; é impossível inverter uma função de hash de uma só via para adquirir a entrada original do valor da saída

Esta é uma diferença importante: Um algoritmo de cifragem não destrói nenhuma informação. Para qualquer determinado texto cifrado (e uma chave), existe somente um texto plano correto que pode ter resultado aquele texto cifrado. Uma função de hash de uma só via destrói a informação. Para qualquer saída resultante de uma função de hash de uma só via, várias mensagens podem ter produzido aquela saída. Outra diferença entre algoritmos de cifragem e funções de hash de uma só via é que as funções de hash de uma só via não têm uma chave. Nenhum segredo é envolvido na função de hash de uma só via; a segurança está na falta da habilidade de ir para outro modo. Esta propriedade faz disso uma maneira útil para identificar uma mensagem. Pense em uma função de hash de uma só via como uma impressão digital. Da mesma maneira que uma impressão digital identifica exclusivamente um indivíduo, uma função de hash de uma só via identifica exclusivamente uma mensagem de tamanho arbitrário. Pelo menos, essa é a idéia. Tecnicamente isso é uma mentira.

Algoritmos de Funções Hash

MD5 (RFC 1321):

- ✓ Fornece resumos de 128 bits computados em quatro passos.
- ✓ Dada uma seqüência arbitrária de 128 bits x, é difícil construir uma mensagem m cujo hash MD5 seja igual a x.

SHA-1:

- ✓ Padrão nos EUA [NIST, FIPS PUB 180-1].
- ✓ Resumos de 160 bits.

Exemplo de Comunicação

Suponha que Alice queira enviar uma mensagem assinada digitalmente para Bob. Estes são os passos que ela tem que seguir:

1. Alice escreve a mensagem
2. Alice gera um hash de uma só via da mensagem e usa uma função de hash de uma só via, como o MD5
3. Alice assina o valor de hash com um algoritmo de assinatura digital de chave pública, como o RSA e a chave privada dela
4. Alice concatena a mensagem e a assinatura para adquirir uma nova mensagem, agora assinada digitalmente
5. Alice envia por e-mail esta mensagem assinada para Bob

Do lado de Bob, ele pode ler a mensagem sem fazer qualquer esforço. Mas ele quer verificar a assinatura de Alice. Estes são os passos ele tem que seguir:

1. Bob separa a mensagem da assinatura
2. Usando uma função de hash de uma só via, Bob computa o valor de hash da mensagem
3. Bob adquire a chave pública de Alice
4. Usando um algoritmo de assinatura digital de chave pública e a chave pública de Alice, Bob decifra a assinatura de Alice
5. Bob compara a assinatura decifrada de Alice com o valor de hash da mensagem. Se eles são o mesmo, Bob verificou a assinatura de Alice e aceitou a mensagem como genuína. Se eles são diferentes, Bob rejeita a assinatura

Certificado Digital

O certificado digital é um arquivo eletrônico que contém dados de uma pessoa ou instituição, utilizados para comprovar sua identidade. Este arquivo pode estar armazenado em um computador ou em outra mídia, como um *token* ou *smart card*

Exemplos semelhantes a um certificado digital são o CNPJ, RG, CPF e carteira de habilitação de uma pessoa. Cada um deles contém um conjunto de informações que identificam a instituição ou pessoa e a autoridade (para estes exemplos, órgãos públicos) que garante sua validade. Algumas das principais informações encontradas em um certificado digital são: dados que identificam o dono (nome, número de identificação, estado, etc); nome da Autoridade Certificadora (AC) que emitiu o certificado; o número de série e o período de validade do certificado; a assinatura digital da AC. O objetivo da assinatura digital no certificado é indicar que uma outra entidade (a Autoridade Certificadora) garante a veracidade das informações nele contidas

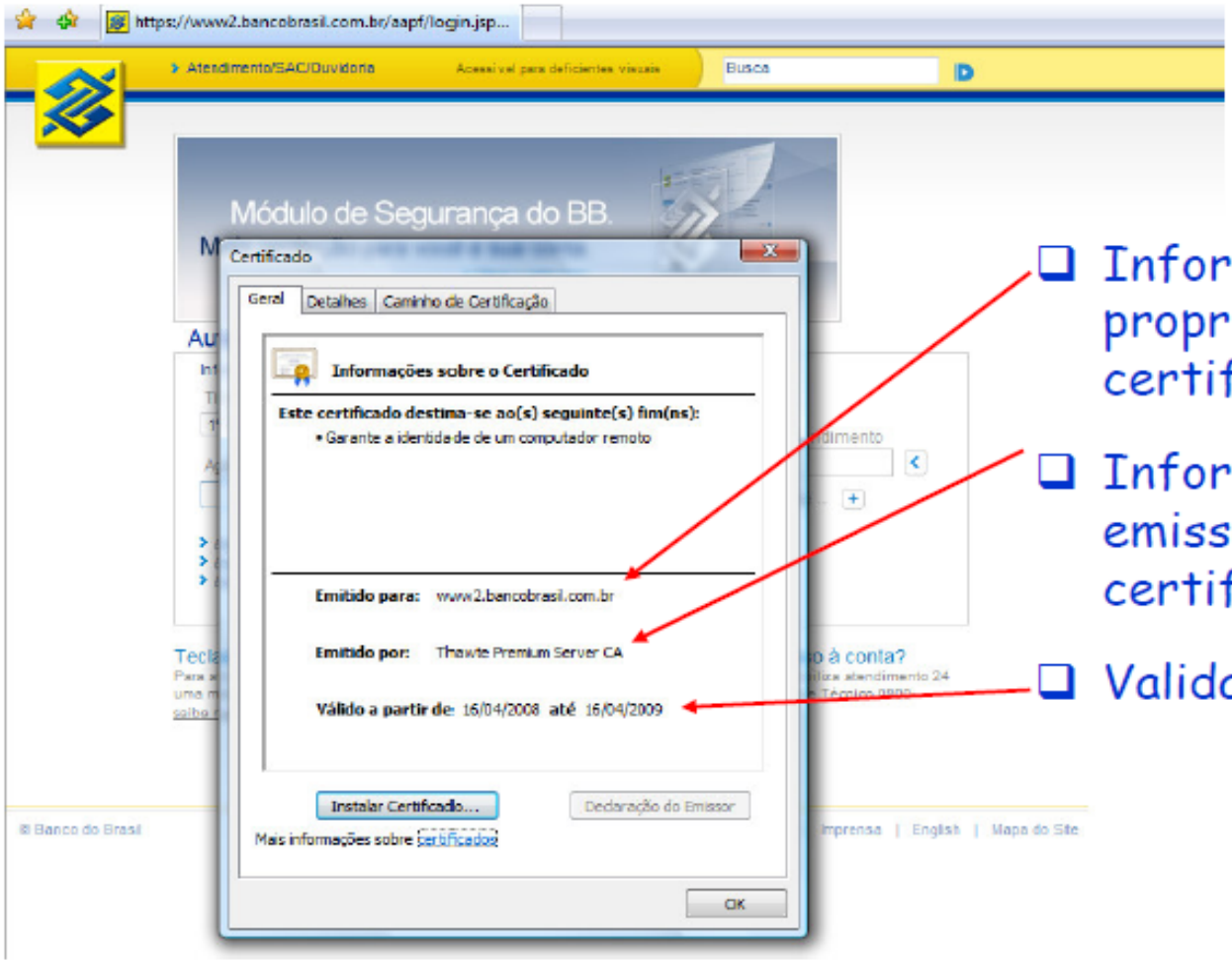
Autoridade Certificadora (AC) é a entidade responsável por emitir certificados digitais. Estes certificados podem ser emitidos para diversos tipos de entidades, tais como: pessoa, computador, departamento de uma instituição, instituição, etc. Os certificados digitais possuem uma forma de assinatura eletrônica da AC que o emitiu. Graças à sua idoneidade, a AC é normalmente reconhecida por todos como confiável, fazendo o papel de "Cartório Eletrônico".

Alguns exemplos típicos do uso de certificados digitais são:

✓ Quando você acessa um site com conexão segura, como por exemplo o acesso a sua conta bancária pela Internet (vide Parte IV: Fraudes na Internet), é possível checar se o site apresentado é realmente da instituição que diz ser, através da verificação de seu certificado digital;
✓ Quando você consulta seu banco pela Internet, este tem que se assegurar de sua identidade antes de fornecer informações sobre a conta;
✓ Quando você envia um e-mail importante, seu aplicativo de e-mail pode utilizar seu certificado para assinar "digitalmente" a mensagem, de modo a assegurar ao destinatário que o e-mail é seu e que não foi adulterado entre o envio e o recebimento

Certificado Digital

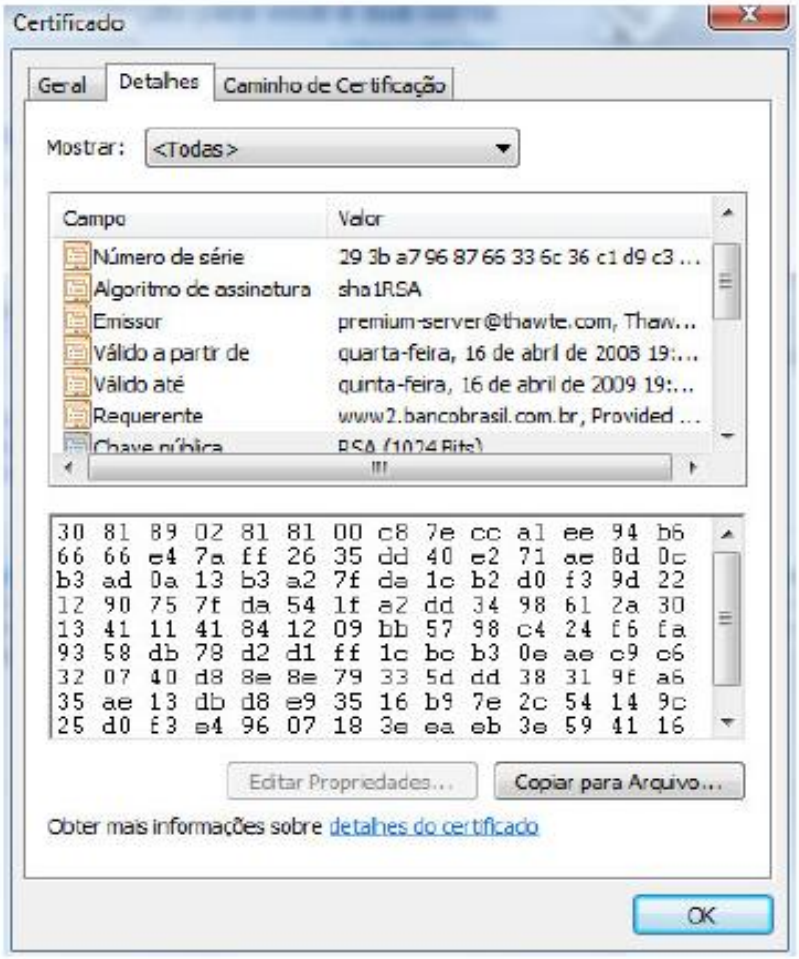
Um certificado contém:

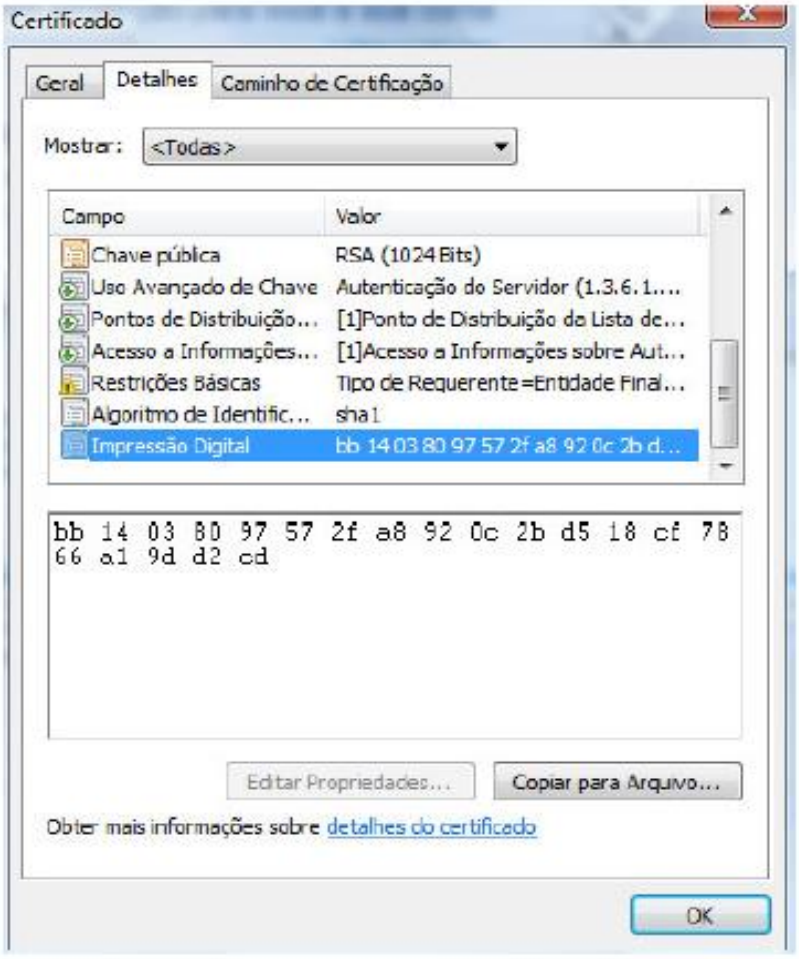


- Informações do proprietário do certificado
- Informações do emissor do certificado
- Validade

Certificado Digital

Um certificado contém:





SSH – Secure Shell

O protocolo SSH, também foi idealizado para permitir o acesso a um computador remoto como substituição ao protocolo TELNET. O SSH permite que os dados sejam trocados entre os sistemas conectados num canal seguro (criptografado). A criptografia utilizada no SSH foi idealizada para prover confidencialidade e integridade de dados através de redes inseguras como a Internet, por exemplo. O SSH também subsitui a necessidade de outros protocolos inseguros como: rlogin, rsh e rcp

O SSH utiliza criptografia de chave-pública para se autenticar a um sistema remoto e assim permitir a este sistema autenticar o usuário. Qualquer pessoa pode gerar seu próprio par de chaves diferenciadas (pública e privada). A chave-publica é distribuída e armazenada em qualquer computador que o usuário deseja ter o acesso garantido. A chave-privada sempre deve ser mantida em segredo e fora do alcance. Embora o SSH também permita a autenticação através de senha (password) essa funcionalidade deve ser evitada sempre que possível para impedir ataques do tipo MITM

Atualmente na versão 2 (SSH-2) o SSH ainda possibilita o tunelamento de conexões com protocolos diferentes; a transferência de arquivos via SCP ou SFTP; o encaminhamento (forwarding) de portas TCP e conexões via padrão X11

Camada: Aplicação
Protocolo: TCP
Porta Padrão: 22
Modelo de Operação: Cliente-Servidor

Sintaxe do SSH: ssh user@host [opções]

Arquivos Principais do SSH: /etc/ssh/ (sshd_config)

Daemon: sshd

OpenSSH – Secure Shell

O OpenSSH é uma suíte open-source de ferramentas e serviços com características de SSH, muito utilizado por administradores de rede. A maioria dos usuários nem imagina que os utilitários telnet, rlogin e ftp que utilizam diariamente são um dos elos mais fracos da Internet e o motivo de inúmeras invasões e roubo de informações ocorrerem, pois esses utilitários transmitem os dados e a autenticação do usuário (senha) em texto claro que pode ser facilmente capturado

Já a suíte OpenSSH encripta e criptografa todo o tráfego incluindo as senhas, o que elimina de forma efetiva a possibilidade de capturar as informações trafegadas mantendo assim a confifencialidade e integridade das informações. Adicionalmente, a suíte SSH provê recursos de tunelamento seguro e vários métodos de autenticação, suportando todas as versões do protocolo SSH e suas características de segurança

A suíte OpenSSH substitui com eficiência o rlogin e o telnet pelo ssh; o rcp pelo scp e o ftp pelo sftp. Também inclui o daemon sshd (o daemon ou servidor de ssh) e outros utilitários importantes como: ssh-add, ssh-agent, ssh-keysign, ssh-keyscan, ssh-keygen e o sftp-server

A suíte OpenSSH é desenvolvida e mantida pelo Projeto OpenBSD. É desenvolvido em países que permitem a exportação de criptografia e é mantido sob código aberto (open source) sob licenciamento BSD

Gerando relação de confiança via SSH entre servidores. Login sem utilização de senha e sim chave SSH

Instalação do pacote OpenSSH

Geração de chave DSS (DSA) ou RSA => **ssh-keygen -t rsa -b 1024**

Nessa fase é solicitada uma senha (passphrase) que é opcional e pode ser alterada posteriormente via comando: ssh-keygen -p [-P old_passphrase] [-N new_passphrase] [-f keyfile]

Serão gerados dois arquivos em **/home/<user>/.ssh** => **id_rsa** e **id_rsa.pub**

O conteúdo do arquivo id_rsa.pub deve ser enviado, de forma segura, para o(s) outro(s) servidor(es). Exemplo:

cat id_rsa.pub | ssh username@host 'cat >> .ssh/authorized_keys‘

ou

ssh-copy-id -i ~/.ssh/id_rsa.pub username@host

O arquivo **authorized_keys** é portanto o arquivo aonde as chaves públicas de servidores confiáveis são armazenadas

Outro arquivo importante é o **known_hosts** que armazena a chave pública de um servidor que está se conectando a outro via ssh na primeira conexão que o mesmo realiza. Esse arquivo também reside no diretório **/home/<user>/.ssh**

A permissão do diretório **.ssh** deve estar setada em **0700** e a permissão dos arquivos em **0640**

```
a@A:~> ssh-keygen -t rsa -b 1024
Generating public/private rsa key pair.
Enter file in which to save the key (/home/a/.ssh/id_rsa):
Created directory '/home/a/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/a/.ssh/id_rsa.
Your public key has been saved in /home/a/.ssh/id_rsa.pub.
The key fingerprint is:
3e:4f:05:79:3a:9f:96:7c:3b:ad:e9:58:37:bc:37:e4 a@A
```

Now use ssh to create a directory ~/.ssh as user b on B. (The directory may already exist, which is fine):

```
a@A:~> ssh b@B mkdir -p .ssh
b@B's password:

a@A:~> cat .ssh/id_rsa.pub | ssh b@B 'cat >> .ssh/authorized_keys'
b@B's password:
```

From now on you can log into B as b from A as a without password:

```
a@A:~> ssh b@B hostname
B
```

Depending on your version of SSH you might also have to do the following changes:
Put the public key in .ssh/authorized_keys2 Change the permissions of .ssh to 700 Change the permissions of .ssh/authorized_keys2 to

Outra utilização do OpenSSH é para a geração de certificados digitais do tipo **CSR (Certificate Signing Request)** usado para que pessoa física ou jurídica tenham um certificado autenticado por uma entidade certificadora (CA) reconhecida mundialmente. Outro tipo de certificado que pode ser produzido são ou **self-signed certificates (certificados auto-assinados)** que podem ser utilizados para testes ou utilização interna em ambientes de Intranet, por exemplo

Gerando uma Private Key

openssl genrsa -des3 -out server.key 1024

Gerando um certificado CSR (Certificate Signing Request)

openssl req -new -key server.key -out server.csr

Removendo a senha (passphrase) do arquivo gerado

cp server.key server.key.org
openssl rsa -in server.key.org -out server.key

Gerando um Self-Signed Certificate

openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt

ou

openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout <arquivo .pem> -out <arquivo .pem>

Checando a consistência de um arquivo de certificado gerado

openssl req -in <arquivo certificado> -noout -text -verify

Dúvidas ???