Segurança na Internet
partes 3 & 4

Prof. Elias P. Duarte Jr., *Ph.D.*
DInfo – UFPR
Itaipu 11/07/2003

Criptografia com Chave Pública

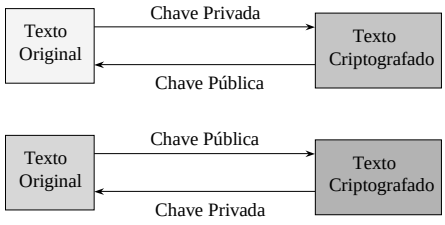
Distribuição da Chave Secreta

- O maior problema é a chave DEIXAR de ser secreta
- Risco: como compartilhar e distribuir a chave?
- Se a chave for descoberta por um invasor, por melhor que seja o algoritmo, é possível descriptografar tudo

Que Tal Usar Chave PÚBLICA?

- Todos conhecem uma das chaves! É distribuída publicamente
- Em 1976, Diffie & Hellman propuseram a ***criptografia com chave pública***
- O sistema usa duas chaves **diferentes**: uma para criptografar, outra para descriptografar
- Todos conhecem uma das chaves! É distribuída publicamente

Duas Chaves Diferentes



Chave Pública: Aplicações

- Bob criptografa uma mensagem para Alice com a chave pública da Alice
- Apenas Alice consegue descriptografar, com sua chave privada
- Bob criptografa mensagem para Alice com a chave privada dele próprio (de Bob)
- Ao usar a chave pública de Bob para descriptografar a mensagem, Alice confirma sua origem

Todo Mundo Conhece a Chave

- Critptografia com chave pública é também chamada de *criptografia assimétrica*
- Pois há duas chaves envolvidas, uma diferente da outra
- Criptografia com chave secreta é também chamada de *criptografia simétrica*
- Uma única chave é usada para criptografar E descriptografar

Criptografia Assimétrica

- $Descript_{K1}(Cript_{K2}(P)) = P$
- Deve ser extremamente difícil obter uma das chaves a partir da outra
- Como uma das chaves é pública, e o algoritmo também é público, o intruso pode tentar realizar vários experimentos para descobrir a chave privada

Um Pouco de Matemática

- Aritmética Modular
- Módulo *n*
- x **mod** *n* é o resto da divisão inteira de x por n
- Inclui apenas os números inteiros menores ou iguais a *n*

Soma Módulo *n*

- Basta tirar o módulo *n* do resultado
- Exemplos módulo 10:
  - 5 + 5 = 0
  - 3 + 9 = 2
  - 2 + 2 = 4
  - 9 + 9 = 8
- Na verdade o método de César usa soma módulo 26 considerando o alfabeto

Subtração Módulo *n*

- Seja -*n* o número que você tem que somar a *n* para obter 0
- Considere o número 7; -7 = 3

Um Método com Duas Chaves!

- A soma módulo *n* faz o mapeamento:
  - 0 1 2 3 4 5 6 7 8 9 0
- Com chave 8:
  - 8 9 0 1 2 3 4 5 6 7 8
- Para descriptografar basta somar -8=2
  - 0 1 2 3 4 5 6 7 8 9 0
- 8 é a chave usada para criptografar, 2 é a chave usada para descriptografar!

Multiplicação Módulo *n*

- Como fica a tabela de múltiplicação módulo 10 quando multiplicamos por *0*?
  - 0 1 2 3 4 5 6 7 8 9
  - 0 0 0 0 0 0 0 0 0 0
- Quando multiplicamos por 1?
  - 0 1 2 3 4 5 6 7 8 9
  - 0 1 2 3 4 5 6 7 8 9

Multiplicação Módulo *n*

- Agora vamos multiplicar por 3
  - 0 1 2 3 4 5 6 7 8 9
  - 0 3 6 9 2 5 8 1 4 7
  - Mapeamento perfeito!!
- Multiplicando por 6:
  - 0 1 2 3 4 5 6 7 8 9
  - 0 6 2 8 4 0 6 2 8 4
  - Neste caso cada dois números são mapeados para a mesma saída
  - Há perda de informação

Prepare a Tabela Completa

- 0 1 2 3 4 5 6 7 8 9
- 0 -> 0 0 0 0 0 0 0 0 0 0
- 1 -> 0 1 2 3 4 5 6 7 8 9
- 2 -> 0 2 4 6 8 0 2 4 6 8
- 3 -> 0 3 6 9 2 5 8 1 4 7
- 4 -> 0 4 8 2 6 0 4 8 2 6
- 5 -> 0 5 0 5 0 5 0 5 0 5
- 6 -> 0 6 2 8 4 0 6 2 8 4
- 7 -> 0 7 4 1 8 5 2 9 6 3
- 8 -> 0 8 6 4 2 0 8 6 4 2
- 9 -> 0 9 8 7 6 5 4 3 2 1

Quais Chaves Podemos Usar?

- Apenas {1, 3, 7, 9}
- Por que?
- Estes números e o 10 são *primos entre si*
- O único divisor comum entre estes números e o 10 é 1

Inverso Multiplicativo

- Considere o número x
- Qual outro número eu devo multiplicar por x para obter 1?
- Este é o *inverso multiplicativo* de x (x^{-1})
- $x * x^{-1} = 1$
- Apenas os números {1, 3, 7, 9} têm inversos multiplicativos módulo 10

Inversos Módulo 10

- Confira sua tabela!
- O inverso de 9 de 1, o inverso de 7 é 3
- Assim temos DUAS CHAVES!
- É claro que {1,9} não são um bom par de chaves...
- Entretanto {3,7} fazem um mapeamento com boa aparência!

Obtendo o Inverso

- Considerando os números de 1 dígito é fácil testar todas as possibilidades
- Considerando números GRANDES, por exemplo de 100 dígitos, é difícil descobrir o inverso usando a força bruta
- O algoritmo de Euclides é usado para descobrir o número y tal que $x*y \bmod n = 1$
- O algoritmo de Euclides é eficiente!

O Algoritmo de Euclides

- Para calcular o MDC de dois números i,j
- Publicado no livro Elementos, de Euclides, há mais de 2000 anos!

```
while (i>0) {
  if (i<j)
    {t=i; i=j; j=t; }
  i = i-j;
}
```

Exponenciação Módulo *n*

- Usa os mesmos princípios da adição e multiplicação
- após efetuar a operação a^b é obtido o mod *n*
- Em alguns casos é possível obter o inverso da exponenciação

O Algoritmo RSA

- Proposto por Rivest, Shamir e Adleman na década de 1970
- Este algoritmo é baseado nos seguintes princípios:
  - Selecione dois números primos grandes (p,q)
  - Calcule n = p*q, z = (p-1)*(q-1)
  - Escolha d, sendo d,z primos entre si
  - Descubra e tal que e*d = 1 mod z

RSA: Crip & Descriptografando

- Para criptografar a mensagem P
- Faça $C = P^e \pmod n$
- Para descriptografar
- Faça $P = C^d \pmod n$
- Desta forma, o RSA é baseado em exponenciação modulo n
- d é o inverso de e na exponenciação

RSA: Chaves

- Para criptografar é necessário saber (e,n)
- Para descriptografar é necessário saber (d,n)
- Chave pública (e,n)
- Chave privada (d,n)
- A dificuldade em descobrir d a partir de (e,n) está em fatorar números grandes (ex. 100 dígitos)

Um Exemplo

- Vamos criptografar a letra **S (cod 19)**
- Vamos usar dois primos pequenos:
  - **p=3, q=11** desta forma **n=p*q=33**
  - **z=(p-1)(q-1)=20**
- Podemos escolher **d=7** pois (7,20) são primos entre si
- Escolhemos **e=3**, pois **7e=1 mod 20**

Continução do Exemplo

- *Criptografando:* $C = 19^3 \bmod 33 = 28$
- *Descriptografando:* $P = 28^7 \bmod 33 = 19$
- Neste exemplo é fácil fatorar n=33 e obter p,q e então z
- Conhecendo z, e conseguimos obter d usando o algoritmo de Euclides

Fatorando Números Grandes

- Considerando um tempo médio por instrução de 1 microsegundo:
- Para fatorar um número de 200 dígitos seriam necessários 4 bilhões de anos
- 500 dígitos -> 10^{25} anos!
- Mesmo considerando tempos por instrução ordens de magnitude menor (1 nanosegundo é a realidade de hoje) é necessário muito tempo para fatorar

Outros Algoritmos de Chave Pública

- Existem vários outros algoritmos
- O primeiro algoritmo de chave pública era baseado no problema da mochila, proposto por Merkle and Hellman
- A lista de todos os objetos para colocar na mochila é pública, os pesos dos objetos selecionados também
- Entretanto a seleção de objetos é secreta

Outros Algoritmos - cont.

- O inventor do algoritmo estava tão certo da sua segurança que ofereceu publicamente US$100 a quem conseguisse quebrá-lo
- Isso foi feito imediatamente por Adi Shamir (o “S” do RSA)
- O inventor propôs modificações e ofereceu US$1000 a quem conseguisse quebrá-lo
- Rivest (o “R” do RSA) levou o prêmio!

Hash ou *Message Digest*

O que é um hash?

- Uma função que faz o mapeamento de uma entrada **A** em uma saída **B**
- Mas *não* faz o mapeamento da saída **B** na entrada **A**
- Em outras palavras: é difícil descobrir a entrada, dada a saída
- ***One-way function***
- Também chamada de RESUMO DIGITAL *message digest*

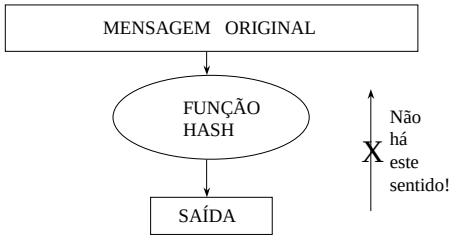
Exemplo de Aplicação

- Passwords - quando o usuário digita o sistema simplesmente calcula o hash do string
- Uma observação: como não há volta da função, é possível *perder informação no processo*

Hash: Entradas & Saídas

- As entradas comuns são
  - arquivos
  - strings
- As saídas são, em geral, números entre
  - 128 e
  - 256 bits

Hash ou *Message Digest*



Características Básicas

- A entrada tem tamanho arbitrário em número de bits, a saída tem número fixo de bits
- Deve ser computacionalmente impossível encontrar a entrada, dada a saída
- Da mesma forma, deve ser impossível encontrar duas mensagens para o mesmo *hash*

Hash *Parece* Aleatório

- Não deve ser possível prever como será um bit da saída, dada a entrada
- Cada saída deve ter, idealmente, metade dos bits setados em 1, a outra metade em 0
- Dadas 1000 saídas, um bit específico deve ser 1 em aproximadamente metade delas
- Mesmo que duas entradas sejam muito parecidas, as saídas devem ser totalmente diferentes

Hash: Características

- Cada bit da saída é influenciado por cada bit da entrada
- Se um bit da entrada é trocado, cada um dos bits da saída tem 50% de chance de ser trocado
- Dado um arquivo de entrada e seu correspondente resultado de hash, deve ser impossível obter um segundo arquivo com o mesmo resultado

Os Hashes Mais Usados Hoje

- **MD2**: Message Digest #2; desenvolvido por Ronald Rivest - produz hashes de 128 bits, é a mais segura das funções de Rivest, mas a mais lenta...
- **MD4**: Message Digest #4; também desenvolvida por Ron Rivest - e também produz hashes de 128 bits, mas é rápida, e mostrou-se insegura... (foi publicada forma de encontrar dois arquivos com mesmo hash)

Hashes Populares - cont.

- **MD5**: Message Digest #5; mesmo autor das anteriores; resultado também é de 128 bits, **a mais usada hoje**, é uma melhorias de MD4, mas já foram publicadas algumas "colisões"
- **SHA**: Secure Hash Algorithm; produzida pelo NSA; resultado de 160 bits, fraquezas obrigaram a produção de **SHA-1**

Qual a aplicação prática dos hashes?

- *Não* são usados para criptografia pura
- E sim para:
  - **criação de assinaturas digitais**
  - **códigos de autenticação de mensagens**
  - **criação de chaves de criptografia a partir de senhas**
- É uma excelente ferramenta para verificar se houve uma *pequena mudança* num arquivo

MD5 em Funcionamento

- Por exemplo, com a entrada:
  - MD5(A chave está escondida no tapete)
- A saída seria
  - 05f8cfc03f4e58cbee731aa4a14b3f03
- Se fizermos uma pequena alteração:
  - MD5(A chave está escondida no tapete!)
- A saída muda *completamente!*
  - d6dee11aae89661a45eb9d21e30d34cb

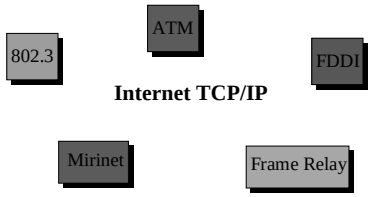
TCP/IP & Segurança
Panorama de Ferramentas

Conteúdo

- A Internet
- Túneis de Segurança & IPsec
- VPN: Virtual Private Networks
- Firewalls
- Sistemas Cliente-Servidor Seguros
- Kerberos
- Aplicações Seguras

A Internet

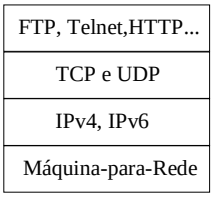
- Um pouquinho de história...60s..70s..80s..90s
- Internet: Rede de Redes



Pilha de Protocolos TCP/IP

Aplicação
Transporte
InterRede
Maq/Rede

Pilha de Protocolos TCP/IP



IPsec

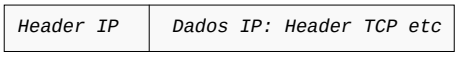
- Arquitetura de segurança para o IP
- Tanto IPv4 como IPv6
- Componentes do IPsec:
  - Protocolos de Segurança, AH - *Authentication Header* & ESP - *Encapsulating Security Payload*
  - Associações de Segurança
  - Gerência de Chaves de Critpografia
  - Algoritmos para criptografia & autenticação

IPsec: Serviços

- Acrescenta vários serviços de segurança para tráfego na camada IP:
  - integridade
  - autenticação
  - sigilo
  - ...
- O IPsec permite a *convivência* de dispositivos com/sem seus mecanismos de segurança

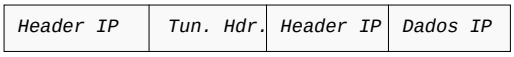
Encapsulamento

- Dentro do pacote IP: outro pacote IP



Encapsulamento

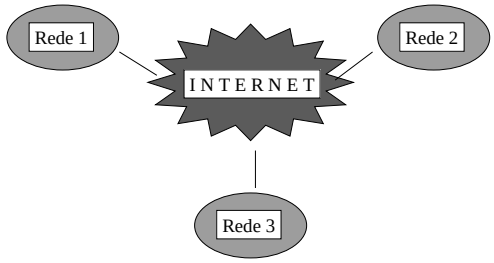
- Dentro do pacote IP: outro pacote IP
- Entre eles o Tunnel Header:



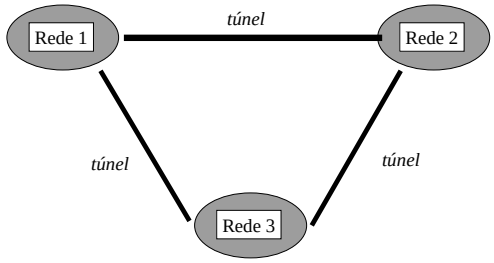
Túneis IP

- Usados para interligar porções da Internet que têm funcionalidades diferentes
- Multicasting, IPv6, IPsec

Túneis de Segurança



Túneis de Segurança



Dois Tipos de Túnel

- *Todo* o tráfego IP entre dois roteadores consiste em um túnel de segurança
- IPsec pode definir um túnel de segurança separado *para cada* conexão TCP
  - Neste caso *não* há dois headers IP, mas um header AH ou ESP
- Além disso: entre um par de dispositivos é possível especificar diferentes categorias de tráfego seguro

Base de Dados de Segurança

- Tanto hosts como roteadores podem implementar os serviços de segurança
- Quando chega um pacote IP, alguns bits especiais no header são examinados (bits *seletores)*
- O padrão dos bits seletores indexa uma base de dados de segurança (SPD), que indica o que deve ser feito com o pacote (AH...)

AH - *Authentication Header*

- Permite autenticação de origem & verificação de integridade
- Calcula um certificado de autenticação, podendo usar uma variedade de algoritmos
- Diferentemente do ESP, permite autenticar o hearder IP - tanto quanto possível
  - Alguns campos mudam seu valor durante a transmissão

ESP - *Encapsulating Security Payload*

- Permite a transmissão de dados criptografados
- *Não* criptografa o header externo ;-)
- Pode ser usado em conjunto com AH

Gerência de Chaves

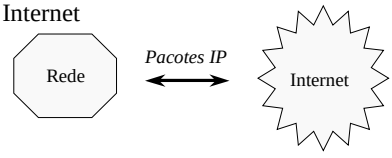
- IKE: Internet Key Exchange (RFC2409)
- ISKAMP: arquitetura para autenticação de chaves
- Oakley: define protocolos seguros para troca de chaves
- SKEME

VPN: *Virtual Private Networks*

- Redes de longa distância de organizações descentralizadas
- Tradicionalmente são baseadas em tecnologias específicas (Frame Relay, SMDS, satélite...)
- Muitas vezes: redes privadas
- VPN: uso da própria Internet como infra-estrutura de comunicação SEGURA usando IPsec

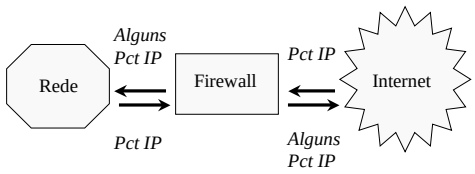
Conectando uma Rede à Internet

- Todos os pacotes vindos de fora chegam à rede
- Pacotes produzidos na rede saem para a Internet



Firewall

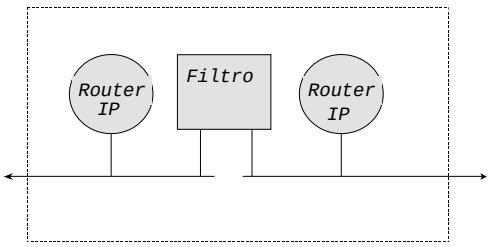
- Filtros
- Todo tráfego de/para a Internet passa pelo firewall



Para que serve um firewall?

- A idéia é proteger uma rede específica de ataques oriundos de outras redes através da Internet
- Da mesma forma, o firewall controla os pacotes que fluem *da rede interna para a Internet*
- Serve também para diminuir a funcionalidade e incomodar usuários :(

Por Dentro do Firewall



firewall ex.

Firewall: Componentes

- O(s) roteador(es) é (são) configurado(s) para examinar cada pacote que chega (ou sai) da rede
- O pacote segue caminho se cumprir os critérios de segurança (ex. IP + porta)
- O filtro pode trabalhar também com o protocolo de aplicação - examinando cabeçalhos, tamanho da mensagem, até o conteúdo

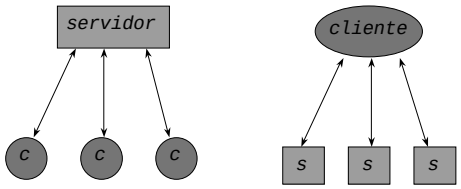
Firewall: Consequências

- Parcialmente isola uma rede da Internet
- Aplicações desabilitadas (ex. ftp em um ou dois sentidos) ou parcialmente desabilitadas (e-mail apenas de mensagens curtas)
- Não é **a** solução 100% perfeita
- Integração a um sistema de gerência para emissão de alarmes - e monitoração do próprio firewall

Sistemas Cliente-Servidor

- Dois processos de aplicação, em dois computadores ligados em rede, desejam se comunicar - como iniciar a comunicação?
- Solução: um dos processos inicia, com uma requisição; outro deve estar na escuta, a espera de requisições e preparado para atendê-las;
- Cliente & Servidor

Clientes & Servidores



Cliente-Servidor: Ameaças

- Os servidores são os portais de entrada que o mundo usa para o sistema da organização
- É necessário autenticar & autorizar acessos
- Um pequeno bug (ou mesmo uma *porta dos fundos*) pode levar a transtornos
- Em seguida veremos algumas ferramentas voltadas para sistemas cliente-servidor

TCP & UDP

- Protocolos para transportar as aplicações
- TCP: confiável, orientado à conexão
- UDP: não-confiável, não-orientado à conexão
- Socket: Interface para o programador de aplicações utilizar TCP ou UDP

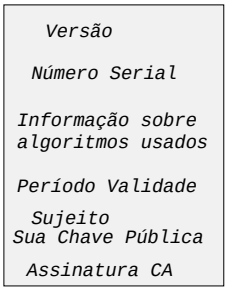
SSL: Secure Socket Layer

- Uma “nova”camada, entre o *TCP* e a aplicação
- Permite que o fluxo bidirecional de dados da conexão seja criptografado
- Sigilo, autenticação, integridade
- Proposta pela Netscape (94), hoje adotada por diversas empresas/organizações

SSL: Características

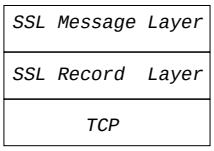
- Flexibilidade: permite o estabelecimento de sessões com autenticação sem sigilo, etc.
- Usa criptografia com chave pública apenas no estabelecimento da sessão
- Os algoritmos a serem usados são determinados para cada sessão
- SSL usa certificados digitais X.509 v3 para prover autenticação de cliente/servidor

Certificados X.509 V3



SSL: O Protocolo, Vers 3.0

- Estruturado em duas camadas:
  - camada de mensagens SSL
  - camada de registros
- As duas camadas ficam sobre o TCP



SSL: A Camada de Registros

- Responsável pelo envio de registros contendo dados entre cliente/servidor
- Cada registro pode conter até 16.383 octetos
- Contém informações de controle & dados
- Os dados podem estar opcionalmente criptografados & comprimidos
- Integridade: Código de Autenticação

SSL: A Camada de Mensagens

- Além de mensagens contendo dados, há 3 tipos de mensagem:
  - Mensagens Handshake
  - Mensagens ChangeCipherSpec
  - Mensagens de Alerta
- Todas são transmitidas pela camada de registros

SSL: Mensagens de Handshake

- Usado para que cliente e servidor possam se autenticar mutuamente
- Uma *chave-mestra* é gerada a partir das chaves públicas, e
- É usada para gerar as chaves secretas para envio de dados cliente-servidor & servidor cliente

SSL: Mensagens ChangeCipherSpec

- Os algoritmos de criptografia e as chaves são determinados no estabelecimento da sessão (ex. RSA, IDEA,...)
- Mas, podem ser alterados posteriormente

SSL: Mensagens de Alerta

- Existem vários tipos de alerta:
  - bad_record_mac
  - handshake_failure
  - certificate_expired....etc...etc
- Os alertas podem ser de 2 níveis:
  - Fatais - terminam a sessão SSL
  - Warnings - indicam problemas

SSL: Considerações

- Performance: na inicialização o uso de criptografia com chave pública reduz taxa de transmissão
- Durante a transmissão de dados - criptografia chave secreta oferece baixo impacto
- A chave-mestra pode ser armazena em cache
- SSLeay: freeware disponível pelo mundo

Aplicações Seguras

- Os próprios protocolos de aplicação podem ter extensões de segurança
- Exemplo: e-mail seguro com PGP, DNS seguro
- Kerberos um sistema para autenticação de sistemas cliente-servidor
- Sistemas Integrados de Gerência de Redes

Conclusões

- A Internet foi inicialmente projetada sem mecanismos de segurança
- A partir do final dos anos 80 problemas graves começam a ocorrer
- Explosão de aplicações & usuários provoca um aumento na vulnerabilidade
- Novas estratégias de segurança têm surgido: IPsec, firewalls, SSL, aplicações seguras