Criptografia

Módulo III – Aplicações Correntes da Criptografia

M. B. Barbosa
mbb@di.uminho.pt

Departamento de Informática
Universidade do Minho

2005/2006

Introdução

- Os riscos associados à utilização indevida de um certificado revogado podem não ser aceitáveis.
- Em alternativa ou adição à consulta de uma CRL, pode ser necessária informação actual sobre o estado de revogação de um certificado.
- O OCSP permite a uma aplicação determinar o estado de um certificado com maior frescura temporal, ou mesmo para obter informação adicional sobre o certificado.
- O Cliente OCSP emite um pedido a um Responder OCSP (Servidor) e suspende a aceitação do certificado até que este forneça uma resposta.
- A extensão `AuthorityInfoAccess` permite indicar num certificado que a CA suporta este serviço.

Pedidos OCSP

- Um pedido OCSP contém a seguinte informação, podendo ser assinados:
  - Versão do protocolo
  - Identificação do serviço requisitado
  - Identificador do certificado alvo
  - Extensões opcionais
- O processamento de um pedido pelo Responder passa pelas seguintes fases:
  - Verificação do formato da mensagem.
  - Verificação de que o servidor está configurado para fornecer o serviço requisitado.
  - Verfificação de que o pedido contém toda a informação necessária.
  - Construção da resposta.

- Num pedido OCSP, os certificados para os quais é solicitada a informação de revogação são indicados numa lista de estruturas ASN.1 `CertID` :

```
CertID          ::=     SEQUENCE {
    hashAlgorithm       AlgorithmIdentifier,
    issuerNameHash      OCTET STRING,
                         -- Hash of Issuer's DN
    issuerKeyHash       OCTET STRING,
                         -- Hash of Issuers public key
    serialNumber        CertificateSerialNumber }
```

- A CA emissora de cada certificado é indicada através de valores de hash do seu *Distinguished Name* e da sua chave pública.

Respostas OCSP

- As respostas OCSP consistem
  - num **identificador do tipo da resposta** e
  - numa **sequência de bytes com seu conteúdo**.
- As respostas OCSP são assinadas digitalmente por uma das seguintes entidades:
  - A CA que emitiu o certificado em questão.
  - Um Trusted Responder, i.e. um Responder em cuja chave pública o cliente confie.
  - Um CA Designated Responder i.e. um Responder autorizado pela CA que emitiu o certificado em questão (detentor de um certificado com uma extensão específica para este fim – `extendedKeyUsage` – emitido pela mesma CA).

- O conteúdo da resposta consiste nos seguintes itens:
  - A versão da sintaxe da resposta.
  - O nome do Responder.
  - **Respostas para cada um dos certificados contidos no pedido**.
  - Uma assinatura digital da resposta.
- Para cada certificado é enviada a seguinte informação:
  - O identificador do certificado alvo.
  - O estado do certificado: **good**, **revoked** ou **unknown**.
  - Período de validade da resposta.
- Uma resposta **good** significa apenas que não foi encontrado nenhum registo de revogação. Não indica que o certificado sequer exista!

- A indicação da validade da resposta pode ser feita de três formas diferentes:
  - **thisUpdate** Hora a que o Responder sabe que a resposta é correcta.
  - **nextUpdate** Hora a que o Responder sabe que vai ser capaz de fornecer uma resposta actualizada. Se não for indicada indica que qualquer nova resposta conterá informação actualizada.
  - **producedAt** Hora a que o Responder assinou a resposta.
- A validação de uma resposta OCSP por parte de um cliente implica verificar que:
  - há correspondência entre os certificados do pedido e da resposta.
  - a assinatura da resposta é válida e provém de um agente autorizado.
  - a validade da resposta é suficientemente recente.

- No serviço OCSP básico a resposta usa a seguinte estrutura ASN.1:

```
ResponseData ::= SEQUENCE {
   version              [0] EXPLICIT Version DEFAULT v1,
   responderID              ResponderID,
   producedAt               GeneralizedTime,
   responses                SEQUENCE OF SingleResponse,
   responseExtensions   [1] EXPLICIT Extensions OPTIONAL }

SingleResponse ::= SEQUENCE {
   certID                       CertID,
   certStatus                   CertStatus,
   thisUpdate                   GeneralizedTime,
   nextUpdate     [0]    EXPLICIT GeneralizedTime OPTIONAL,
   ...   }
```

Serviços Extendidos

- O OCSP é standardizado no RFC2560. Neste documento é apenas definido o serviço básico descrito anteriormente.
- A IETF publicou entretanto um Internet Draft definindo o OCSP V2, uma extensão à funcionalidade básica que define três serviços:
  - Online Revocation Status (ORS)
  - Delegated Path Validation (DPV)
  - Delegated Path Discovery (DPD)
- O serviço é indicado numa extensão incluída nos pedidos OCSP no campo `requestExtensions` .
- O ORS corresponde à funcionalidade básica e é o caso default i.e. quando nenhum serviço é indicado trata-se de uma invocação do ORS.

Delegated Path Validation

- Permite a uma aplicação transferir o processamento da validação de cadeias de certificados complexas para um servidor central.
- Isto permite simplificar as aplicações cliente, reduzindo ao mínimo a funcionalidade de validação de certificados a implementar.
- O protocolo permite que a aplicação cliente controle os aspectos essenciais do comportamento do servidor, por exemplo:
  - Restringir as cadeias de certificação aceitáveis.
  - Se o servidor se deve basear em CRLs e/ou OCSP para fazer a validação.
  - Quais as políticas de certificação que são relevantes para a aplicação.

Delegated Path Discovery

- Permite a uma aplicação que processe certificados obter do servidor a informação disponível sobre um certificado e a sua revogação: cadeias de certificados, CRLs, respostas de outros servidores OCSP, etc.
- A aplicação pode validar certificados obtendo informação de todos os pontos de acesso à PKI disponíveis no servidor OCSP e.g. X.500, LDAP, HTTP, FTP, etc.
- Tal como no serviço anterior, a aplicação pode controlar diversos aspectos da operação do servidor:
  - Restringir as cadeias de certificação aceitáveis.
  - Quais as políticas de certificação relevantes.
  - Mecanismo interactivo de selecção de uma cadeia de validação aceitável para o cliente em termos de políticas de certificação.

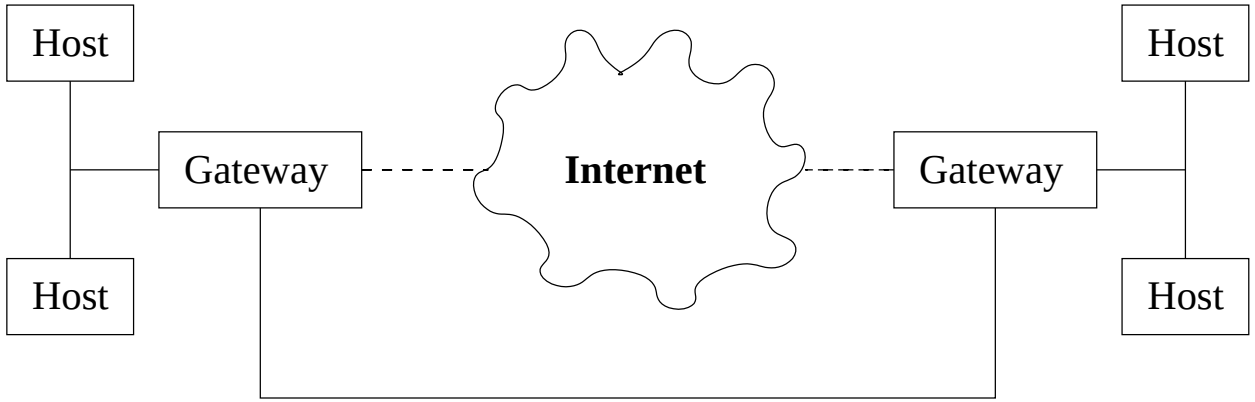
Introdução

- O IP (Internet Protocol) está ao nível da camada de rede do Modelo OSI. Fornece serviços de encaminhamento de pacotes através de redes heterogéneas.
- A maior parte das infraestruturas de comunicação na Internet são baseadas neste protocolo, conjuntamente com o TCP (TCP/IP).
- O IPsec fornece o mesmo conjunto de serviços, mas inclui funcionalidade extra ao nível da seguraça.
- Estes serviços são oferecidos ao nível da camada de rede, oferecendo protecção não só a esse nível, mas também a todas as camadas superiores.
- O IPsec está definido nas especificações RFC2401, e seguintes, da IETF.

- O IPsec oferece os seguintes serviços seguros ao nível da camada de rede:
  - Controlo de acessos
  - Integridade ao nível do pacote
  - Autenticação da origem de dados
  - Protecção contra pacotes repetidos
  - Confidencialidade
  - Confidencialidade de parte do tráfego
- Estes serviços permitem proteger ligações de rede entre nós IP, entre gateways seguras, ou entre um nó IP e uma gateway segura.
- Não substituem os serviços IP. São módulos adicionais que podem ser implementados e utilizados consoante o contexto e as necessidades das aplicações.

Revisão do Protocolo IP

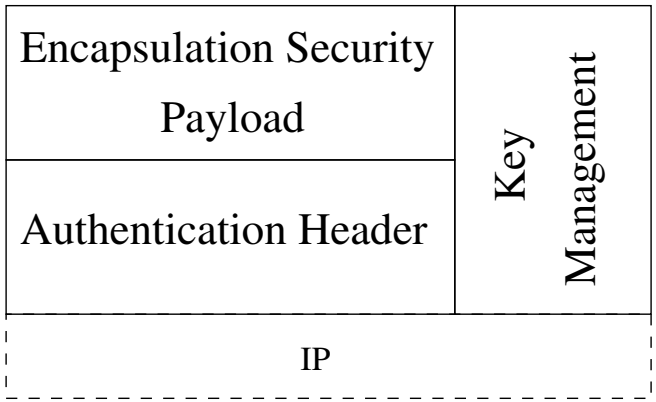
- Funcionamento:



  - Dentro de uma rede local, cada nó constroi um pacote IP incluindo os endereços de origem e destino no cabeçalho.
  - A comunicação com redes remotas é feita passando os pacotes a uma gateway: o endereço da gateway encapsula o verdadeiro.
  - A gateway substitui o encapsulamento reencaminhando o pacote. A gateway da rede remota retira o encapsulamento.

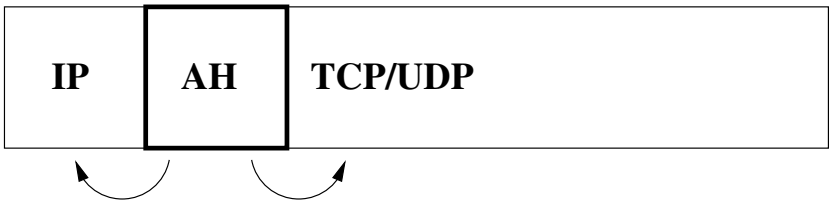
Estrutura do IPsec

- O IPsec está estruturado em duas sub-camadas:



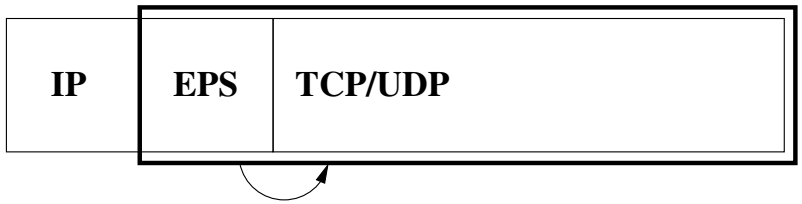
- As duas sub-camadas são apoiadas por procedimentos e protocolos de gestão de chaves criptográficas (manuais ou automáticos).
- Os protocolos estão especificados de forma a serem independentes de algoritmos criptográficos. No entanto, alguns destes algoritmos estão pré-definidos.

IP Authentication Header



- A sub-camada **IP Authentication Header** (AH) inclui serviços de integridade ao nível dos pacotes, autenticação da origem de dados e, opcionalmente, protecção contra a repetição de pacotes.
- Recorre-se ao AH quando se pretende autenticação da informação correspondente à camada de rede (cabeçalho IP), ou quando a confidencialidade não é necessária (ou permitida).

Encapsulating Security Payload



- A sub-camada **Encapsulating Security Payload** (ESP) fornece confidencialidade (cifragem) da totalidade ou de apenas parte do tráfego.
- Como opções, o ESP oferece autenticação, verificação de integridade, e protecção contra pacotes repetidos. No entanto, esta funcionalidade abrange apenas informação correspondente às camadas de transporte e superiores (não trata o cabeçalho IP).
- No entanto pode ser, geralmente é, usado isoladamente.

Modos de Funcionamento

- Consoante o tipo de nós envolvidos, pode funcionar em:
  - **Transport Mode** – Sobre os cabeçalhos IP originais.
    - Apenas serve para ligações host-host
    - Com ESP não há protecção dos cabeçalhos IP (interferiria com a infraestrutura IP).
    - Com AH, há uma protecção parcial desses cabeçalhos.
  - **Tunnel Mode** – Sobre cabeçalhos IP encapsulados.
    - Corresponde a um túnel IP (caminho virtual entre nós).
    - Típicamente utilizado em ligações com/entre gateways
    - A protecção alcança todo o pacote original.
    - Diferenças entre AH e ESP mantêm-se, mas apenas para o cabeçalho exterior.
    - Para as camadas superiores, estas ligações aparecem como interfaces de rede adicionais.

Security Associations e SADs

- A gestão do IPsec baseia-se em **Security Associations**.
- Uma SA é uma ligação simplex identificada por três parâmetros incorporados nos cabeçalhos IPsec:
  - **IP Destination** Define o endereço de destino dos pacotes.
  - **IPsec Protocol** O protocolo (AH ou ESP) utilizado pela SA.
  - **Security Parameter Index** (SPI) Número de 32 bits que distingue SAs do mesmo tipo.
- Associados a este identificador estão todos os parâmetros de funcionamento necessários para a codificação e descodificação dos pacotes enviados através da ligação segura representada pela SA.
- Cada nó IPsec regista esta informação numa Security Association Database (SAD).

Security Policy Database

- O IPsec funciona num nó (máquina ou gateway) de uma rede IP. Cada pacote que passa num nó IPsec pode ser tratado de acordo com uma de três políticas:
  - proteger o pacote com segurança IPsec,
  - enviar o pacote com IP simples, ou
  - descartar o pacote (no caso de gateways que limitam o tráfego entre redes).
- A política aplicada a cada pacote específico está armazenada numa Security Policy Database (SPD) que é gerida por um utilizador ou administrador do sistema.
- A pesquisa na tabela baseia-se em informação contida nos cabeçalhos de rede e transporte do pacote em questão: endereços IP de origem e destino, protocolo e portas de transporte (TCP/UDP).

Configuração do IPsec

- O IPsec permite grande flexibilidade na configuração:
  - dos serviços seguros que são utilizados e em que combinações;
  - da granularidade com que um determinado serviço é aplicado; e
  - dos algoritmos criptográficos utilizados em cada serviços.
- A configuração da granularidade de um serviço consiste em definir com que detalhe se distinguem os pacotes. Por exemplo:
  - podem-se cifrar todos os pacotes entre duas gateways, criando um canal seguro para todas as máquinas que utilizem essa ligação, ou . . .
  - podem cifrar-se apenas os pacotes que circulam entre duas máquinas protegendo apenas essa ligação.

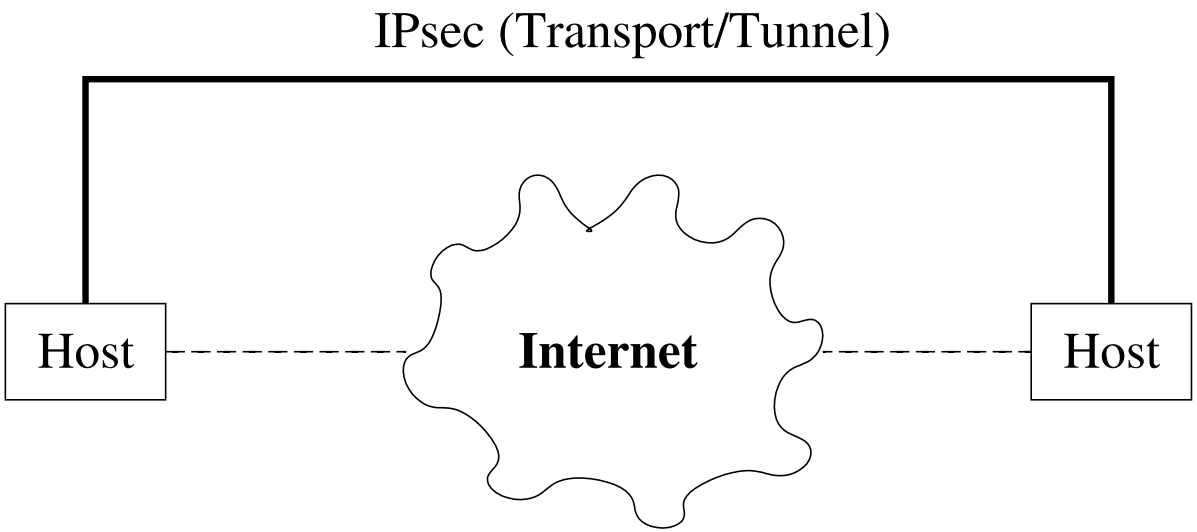
Gestão de Chaves e Algoritmos Criptográficos

- O IPsec permite o funcionamento baseado em técnicas manuais de gestão de chaves.
- No entanto, este tipo de operação não é adequado em sistemas com muitos nós, onde um sistema de criação dinâmica de SAs é preferível.
- Este tipo de funcionamento implica um sistema automático de gestão de algoritmos e chaves criptográficos.
- O sistema recomendado é o **Internet Key Exchange** (IKE), especificado nos RFCs 2407, 2408,2409 e 2412.
- O IKE permite a construção automática de SAs com negociação de parâmetros de comunicação e segurança, autenticação e protocolos de geração e acordo de chaves.

Tratamento de Pacotes Recebidos

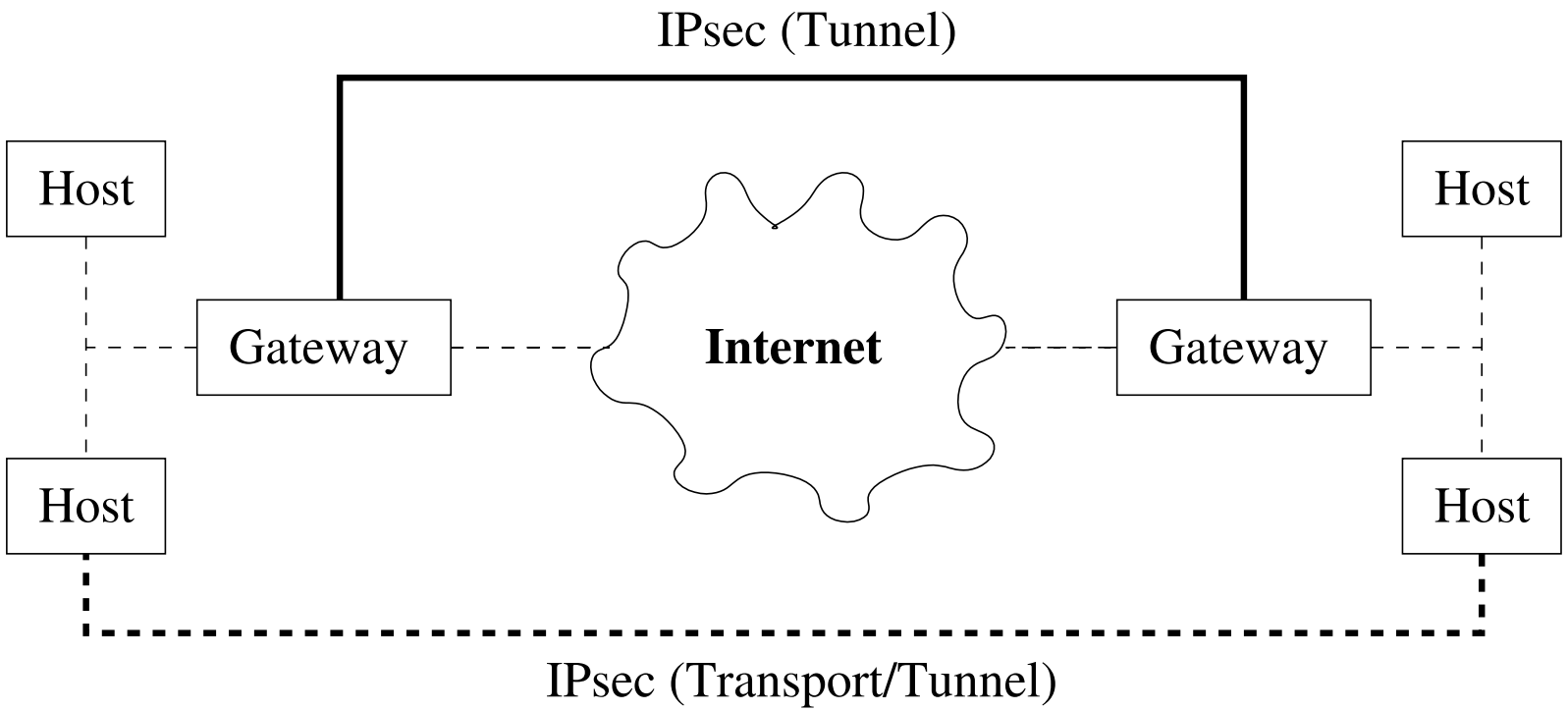
1. A informação contida no cabeçalho IPsec indica uma entrada na SAD:
2. Se essa entrada não existir desencadeia-se, se possível, a negociação de chaves para configurar uma nova SA. Se isto não for possível, o pacote é descartado.
3. Com base na informação associada à SA, processam-se os cabeçalhos e descodifica-se o pacote.
4. Consulta-se outra vez a SPD para saber se o pacote foi processado de acordo com a política correcta.
5. Se o processo decorrer sem problemas passa-se o pacote à camada de transporte.

Ligações Host-to-Host



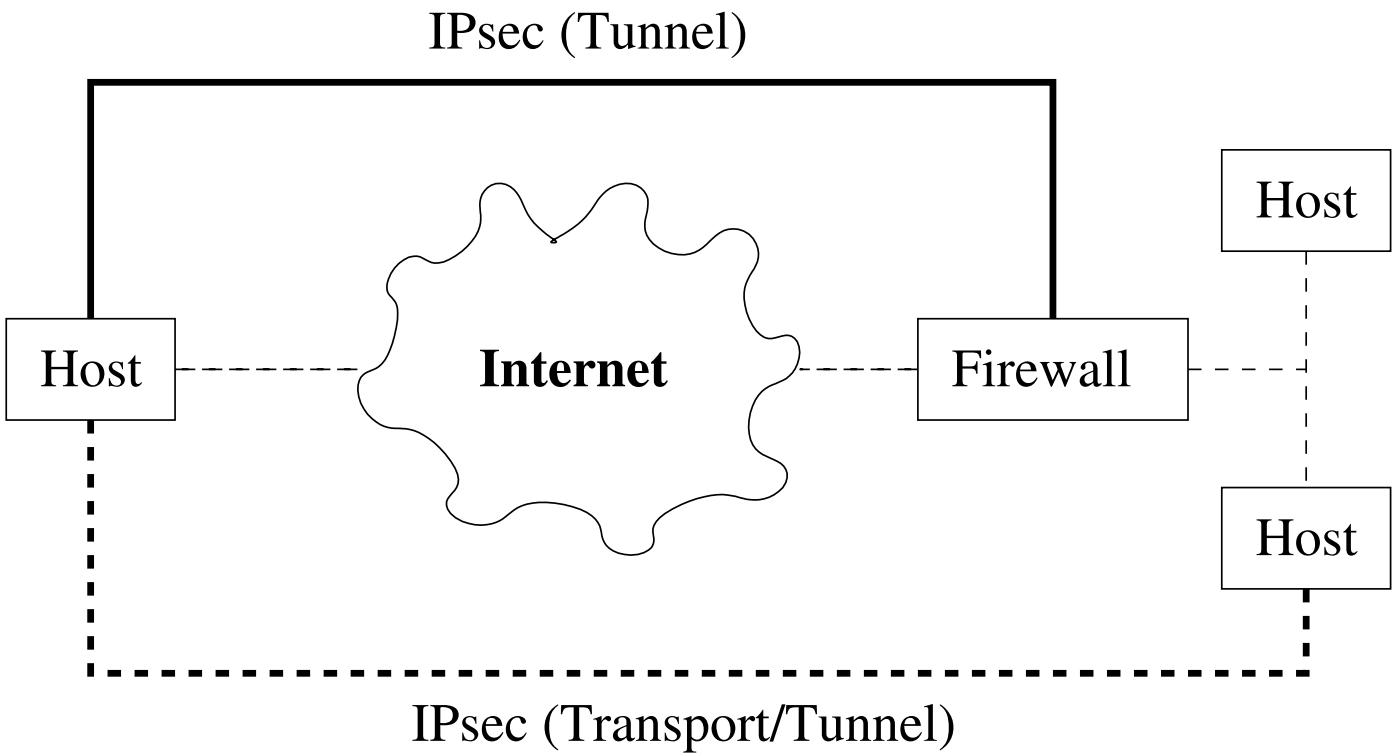
- Os pacotes trocados entre as duas máquinas são protegidos com IPsec, com base em Transport SAs.
- Pode ser utilizado o modo Tunnel, mas isso é redundante: o cabeçalho externo terá os mesmos endereços que o interno.

Virtual Private Networks (VPN)



- As gateways tornam a troca de pacotes entre redes totalmente transparente. Todo o tráfego é protegido.

Ligações remotas seguras (Road Warrior)



- Depois de a máquina externa se validar perante a firewall, passa a funcionar como se estivesse dentro da rede.

Ficha Técnica

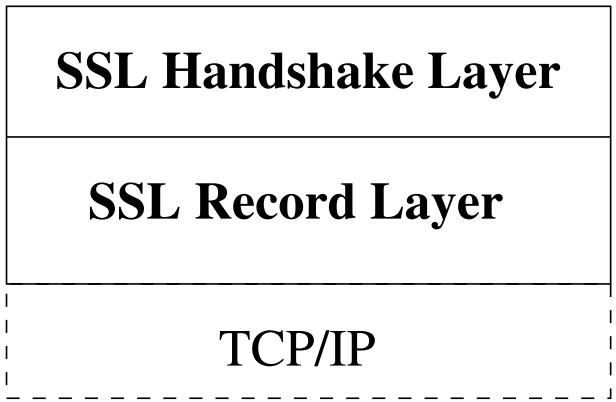
- Algoritmos obrigatórios:
  - **Cifras Simétricas**: DES (CBC)
  - **Funções de Hash Criptográficas**: SHA-1, MD-5
  - **Message Authentication Codes**: HMAC
- Os protocolos são especificados de forma independente dos algoritmos, e há uma grande flexibilidade na utilização de outros algoritmos criptográficos.
- Está prevista por exemplo a utilização de diversas cifras simétricas (3-DES, IDEA, Blowfish), do RSA, do DSA, etc.

Introdução

- A Secure Sockets Layer está para o TCP como o IPsec está para o IP. É um *upgrade* da camada de transporte para incluir segurança nas comunicações.
- O SSL foi desenvolvido pela Netscape, e a sua versão 3 foi adoptada pela IETF sob a designação **Transport Layer Security** (TLS). O TLS está definido no RFC2246.
- Os serviços fornecidos pelo SSL incluem:
  - Confidencialidade baseada em cifras simétricas.
  - Autenticação baseada em criptografia de chave pública.
  - Integridade baseada em Message Authentication Codes.

Estrutura do SSL

- O SSL está estruturado em duas sub-camadas:



- A **Handshake Layer** permite a autenticação mutua entre clientes e servidores, e a negociação de algoritmos e chaves criptográficos antes de se iniciar a troca de dados através da Record Layer.
- A **Record Layer** encapsula a informação correspondente às camadas superiores.

Sessões SSL

- O funcionamento do SLL baseia-se em **sessões** estabelecidas entre um **cliente** e um **servidor**.
- Cada sessão SSL pode incluir várias ligações seguras, e cada nó pode manter diversas sessões SSL. Durante o seu estabelecimento e operação, as sessões e ligações SSL atravessam uma sequência de estados.
- Cliente e Servidor mantêm uma máquina de estados para cada sessão e ligação. A camada de Hanshake sincroniza os estados no cliente e no servidor.
- As transições entre estados efectuam-se em duas fases:
  - Primeiro constrói-se/negoceia-se um **pending state**.
  - Depois substitui-se o **operating state** pelo pending state.

Estado de uma Sessão SSL

- **Session identifier** Uma sequência arbitrária de bytes escolhida pelo servidor para identificar a sessão.
- **X509 certificate of the peer** Certificado do interlocutor.
- **Compression method** Algoritmo de compressão da informação antes de ser cifrada.
- **Cipher spec** Algoritmo de cifra simétrica (e algoritmo de hash criptográfico para utilização em MACs).
- **Master secret** Chave secreta partilhada por Cliente e Servidor e da qual são derivados todos os segredos utilizados na sessão (chaves e IVs).
- **Is resumable** Indica se a sessão pode ser utilizada para novas ligações.

Estado de uma Ligação SSL

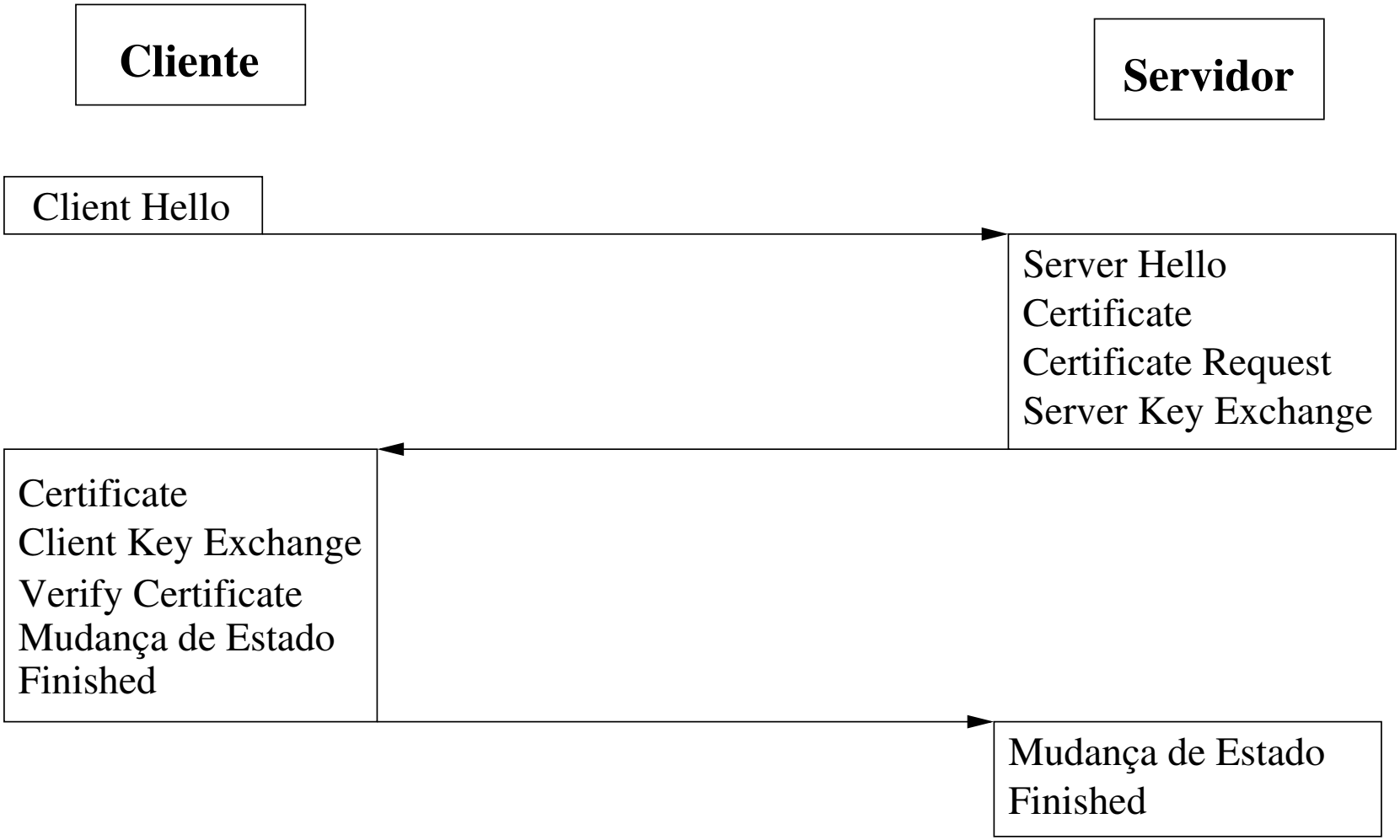
- **Server/Client random** Números aleatórios escolhidos por Cliente e Servidor para estabelecimento da ligação.
- **Server/Client write MAC secret** Chaves utilizadas por Cliente e Servidor para efectuar MACs sobre dados transmitidos.
- **Server/Client write key** Chaves utilizadas por Cliente e Servidor para cifrar dados transmitidos.
- **Initialization vectors** Vectores de inicialização (IV) para os modos de cifra simétrica que os utilizam.
- **Sequence numbers** Contadores sequenciais das mensagens enviadas e recebidas.

Record Layer

- Recebe informação arbitrária das camadas superiores, em blocos de dados de tamanho variável.
- Os dados são **fragmentados** em blocos com um máximo de 2^{14} bytes denominados **SSL Plaintext**.
- Os blocos SSL Plaintext são comprimidos com o algoritmo da sessão, originando blocos **SSL Compressed**.
- Os dados SSL Compressed são protegidos com a cifra e algoritmo de MAC definidos na CipherSpec da sessão (o MAC é calculado antes da cifragem). O resultado é um bloco do tipo **SSL Ciphertext**.
- Estes blocos são trocados entre Cliente e Servidor que têm de reverter estas transformações para obter o texto limpo.

Handshake Layer

- Os parâmetros de sessão e ligação utilizados pela Record Layer são estabelecidos pela Hanshake Layer.
- As mensagens da Handshake Layer viajam elas próprias sob o controlo da Record Layer. Inicialmente não há qualquer protecção: é utilizada uma *cipher spec* nula até que a primeira negociação seja concluida.
- Uma negociação é iniciada pelo Cliente com uma mensagem **Client Hello**. O Servidor deve responder com uma mensagem equivalente. Ficam acordados:
  - A versão do protocolo SSL a utilizar
  - O identificador da sessão e os números aleatórios.
  - Os algoritmos criptográficos a utilizar (os mais fortes dos suportados).
  - O algoritmo de compressão a utilizar


Cliente
Servidor
Client Hello
Server Hello
Certificate
Certificate Request
Server Key Exchange
Certificate
Client Key Exchange
Verify Certificate
Mudança de Estado
Finished
Mudança de Estado
Finished

- Caso seja utilizada autenticação do Servidor, este envia o seu certificado X.509 ao Cliente, que o valida. Além da validação habitual, o Cliente assegura-se de que o nome de domínio do Servidor, indicado no certificado, está correcto.
- Parâmetros do Servidor específicos para acordo de chaves podem também ser enviados nesta fase (**Server Key Exchange**), se o seu certificado não incluir informação suficiente para esta funcionalidade.
- Caso o Servidor autentique o Cliente, solicita o certificado correspondente (**Certificate Request**). Este pedido inclui um desafio para ser utilizado na autenticação do cliente.
- O Servidor termina esta fase da negociação enviando uma mensagem **Server Hello Done**.

- Caso tenha recebido um pedido de certificado, o Cliente tem de envia-lo ou a negociação falha.
- Conjuntamente com o certificado o Cliente tem de enviar uma assinatura digital do desafio que recebeu, comprovando assim a posse da chave privada associada ao certificado.
- Finalmente, o Cliente envia os seus parâmetros para acordo de chaves (**Client Key Exchange**), altera o seu estado de sessão, e envia uma primeira mensagem cifrada que indica o seu estado de prontidão (**finished**).
- O Servidor efectua o mesmo procedimento e a negociação termina tendo sido acordado o Master Secret da sessão.

- A autenticação do servidor fica implicita pelo sucesso da comunicação cifrada nas mensagens **finished**, ou não?
- De facto, o servidor só fica autenticado se o protocolo de acordo de chaves implicar a utilização da sua chave privada.
- Isto acontece sempre:
  - No protocolo **RSAKeyExchange** o cliente gera um segredo e cifra-o com a chave pública do servidor. Para gerar o Master Secret, o servidor tem de decifrar este segredo com a sua chave privada.
  - Nos outros protocolos, os parâmetros públicos do servidor utilizados no protocolo de acordo de chave são assinados com a sua chave privada.

Segurança

- A versão 3 do SSL é considerada um sistema seguro. É uma evolução em relação às versões anteriores, colmatando falhas de segurança importantes.
- Um dos problemas mais conhecidos na versão 2 do SSL era a vulnerabilidade ao ataque "ciphersuit rollback":
  - Um intruso podia editar as mensagens de **hello** trocadas entre Cliente e Servidor de forma a que ambos pensassem que o outro apenas conseguia funcionar com um nível de segurança reduzido.
  - O resto da negociação decorria sem alterações e estabelecia-se uma ligação com um nível de segurança reduzido, mais vulnerável a ataques por parte do intruso.
- Este ataque era possível porque as mensagens de handshake não eram autenticadas!

- A versão 3 do SSL resolveu este problema obrigando a que todas as mensagens de handshake fossem utilizadas para gerar o valor cifrado nas mensagens **finished**.
- "Change cipher spec dropping" é outro ataque possível sobre uma implementação pouco cuidada:
  - Quando a sessão a ser negociada inclui apenas autenticação, i.e. não inclui cifragem, é possivel eliminar das mensagens **finished** a informação de autenticação.
  - Interceptando as mensagens **change cipher spec**, impede-se a activação da autenticação. Fornecendo a Cliente e Servidor mensagens **finished** alteradas, estabelece-se uma sessão sem protecção.
- A solução para este ataque consiste em exigir uma mensagem de **change cipher spec** antes de uma mensagem **finished** nestas situações.

Ficha Técnica

- **Cifras Simétricas**: DES, 3-DES, RC4
- **Algoritmos de Compressão**: ZLIB
- **Funções de Hash Criptográficas**: SHA-1, MD5
- **Cifras Assimétricas**: RSA
- **Assinaturas Digitais**: RSA, DSA
- **Acordo de Chaves**: Fortezza, Diffie-Hellmann, distribuição RSA.