REDES DE COMPUTADORES E A INTERNET 5ª edição
Uma Abordagem Top-Down

8.5. PROTEGENDO CONEXÕES TCP: SSL

Prof. Othon Marcelo Nunes Batista
Mestre em Informática

Capítulo 8: Esboço

REDES DE COMPUTADORES E A INTERNET 5ª edição

Uma Abordagem Top-Down

8.1 O que é segurança na rede?

8.2 Princípios de criptografia

8.3 Integridade de mensagem

8.4 Protegendo o e-mail

8.5 Protegendo conexões TCP: SSL

8.6 Segurança na camada de rede: IPsec

8.7 Segurança em LANs sem fio

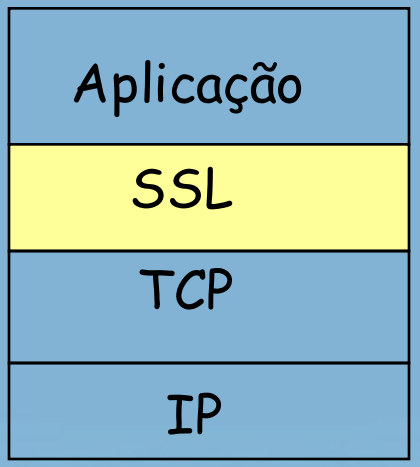
8.8 Segurança operacional: firewalls e IDS

SSL: Secure Sockets Layer

- protocolo de segurança bastante implantado
  - aceito por quase todos os navegadores e servidores Web
  - https
  - dezenas de bilhões de US$ gastos por ano sobre SSL
- originalmente projetado pela Netscape em 1993
- número de variações:
  - TLS: Transport Layer Security, RFC 2246
- oferece
  - Confidencialidade
  - Integridade
  - Autenticação
- objetivos originais:
  - teve em mente transações de comércio eletrônico na Web
  - criptografia (especialmente números de cartão de crédito)
  - autenticação de servidor Web
  - autenticação de cliente opcional
  - mínimo de incômodo ao fazer negócios com novos comerciantes
- disponível a todas as aplicações TCP
  - Interface Secure Socket

SSL e TCP/IP

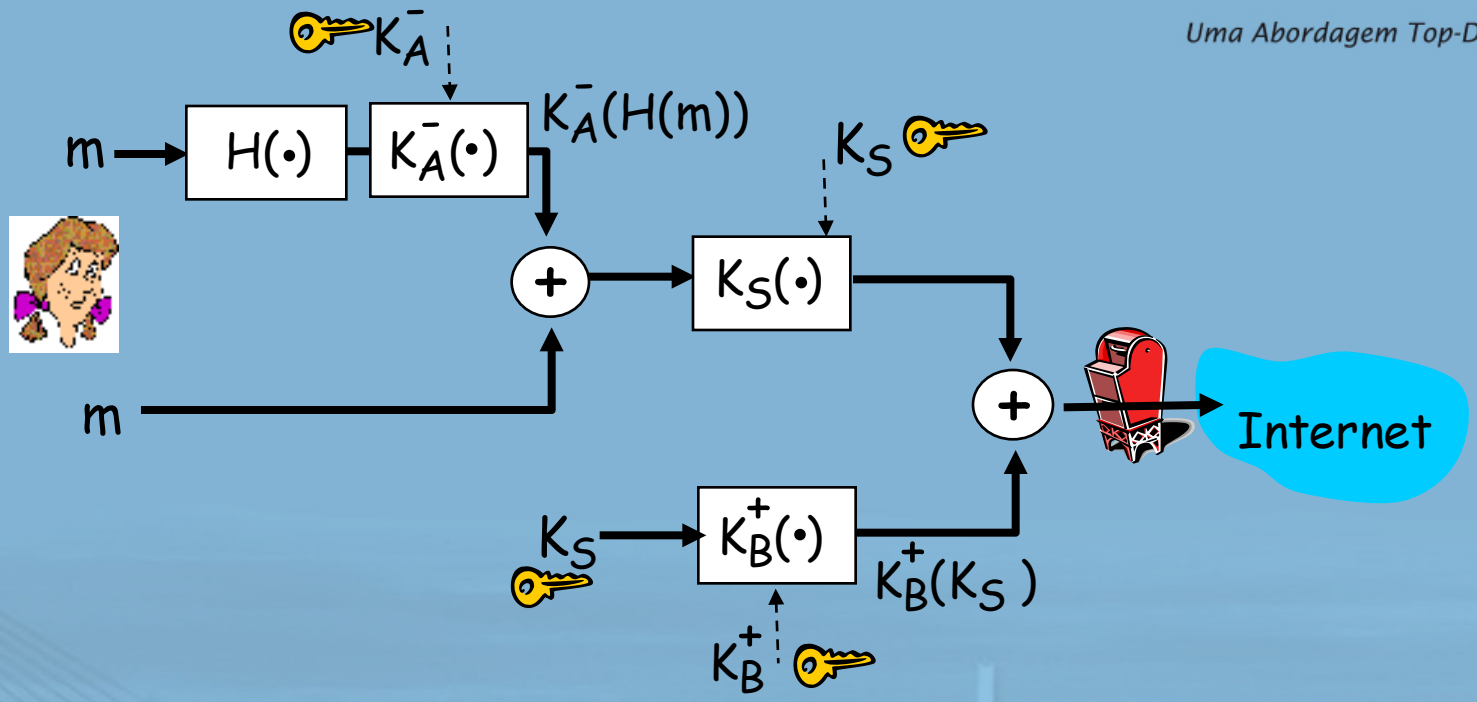
aplicação normal



aplicação com SSL

- SSL oferece interface de programação de aplicação (API) às aplicações
- bibliotecas/classes SSL em C e Java prontamente disponíveis

Poderia fazer algo como PGP:



- mas quer enviar fluxos de bytes & dados interativos
- quer um conjunto de chaves secretas para a conexão inteira
- quer parte de troca de certificado do protocolo: fase de apresentação (handshake)

SSL: um canal seguro simples

- apresentação: Alice e Bob usam seus certificados e chaves privadas para autenticar um ao outro e trocar segredo compartilhado
- derivação de chave: Alice e Bob usam segredo compartilhado para derivar conjunto de chaves
- transferência de dados: dados a serem transferidos são desmembrados em uma série de registros
- encerramento de conexão: mensagens especiais para encerrar conexão com segurança

Uma apresentação simples

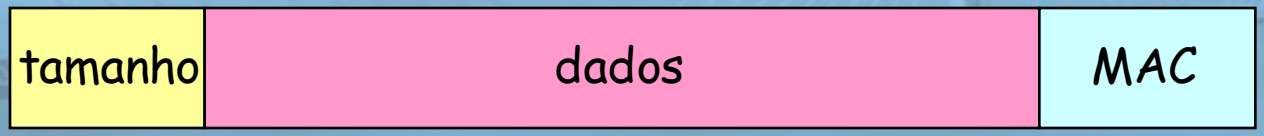
- MS = segredo mestre
- EMS = segredo mestre criptografado

Derivação de chave

- considerado ruim usar a mesma chave para mais de uma operação criptográfica
  - use chaves diferentes para código de autenticação de mensagem (MAC) e criptografia
- quatro chaves:
  - K_c = chave de criptografia para dados enviados do cliente ao servidor
  - M_c = chave MAC para dados enviados do cliente ao servidor
  - K_s = chave de criptografia para dados enviados do servidor ao cliente
  - M_s = chave MAC para dados enviados do servidor ao cliente
- chaves derivadas da função de derivação de chave (KDF)
  - toma segredo mestre e (possivelmente) alguns dados aleatórios adicionais e cria as chaves

Registros de dados

- Por que não criptografar dados em fluxo constante enquanto o escrevemos no TCP?
  - Onde colocaríamos o MAC? Se no final, nenhuma integridade de mensagem até todos os dados processados.
  - Por exemplo, com mensagens instantâneas, como podemos fazer verificação de integridade por todos os bytes enviados antes da exibição?
- Em vez disso, quebre fluxo em série de registros
  - cada registro transporta um mac
  - receptor pode atuar em cada registro quando ele chega
- Problema: no registro, receptor precisa distinguir MAC dos dados
  - quer usar registros de tamanho variável



Números de sequência

- invasor pode capturar e reproduzir registro ou reordenar registros
- solução: colocar número de sequência em MAC:
  - MAC = MAC(M_x, sequência||dados)
  - nota: sem campo de número de sequência
- invasor ainda poderia reproduzir todos os registros
  - use nonce aleatório

Informação de controle

- ataque por truncamento:
  - invasor forja segmento de encerramento de conexão TCP
  - um ou ambos os lados pensam que existem menos dados do que realmente existem.
- solução: tipos de registro, com um tipo para encerramento
  - tipo 0 para dados; tipo 1 para encerramento
- MAC = MAC(M_x, sequência||tipo||dados)

SSL: resumo

bob.com

olá

certificado, nonce

$K_B^+(MS) = EMS$

tipo 0, seq 1, dados

tipo 0, seq 2, dados

tipo 0, seq 1, dados

tipo 0, seq 3, dados

tipo 1, seq 4, encerra

tipo 1, seq 2, encerra

criptografado

SSL não é completo

- Qual é o tamanho dos campos?
- Quais protocolos de criptografia?
- sem negociação
  - permite que cliente e servidor admitam diferentes algoritmos de criptografia
  - permite que cliente e servidor escolham juntos algoritmo específico antes da transferência de dados

Cifras simétricas mais comuns em SSL

- DES - Data Encryption Standard: bloco
- 3DES - Força tripla: bloco
- RC2 - Rivest Cipher 2: bloco
- RC4 - Rivest Cipher 4: fluxo

Criptografia de chave pública

- RSA

Blocos de cifras SSL

- bloco de cifras
  - algoritmo de chave pública
  - algoritmo de criptografia simétrica
  - algoritmo MAC
- SSL admite uma série de blocos de cifras
- negociação: cliente e servidor devem combinar sobre bloco de cifras
- cliente oferece escolha; servidor escolhe uma

SSL real: Apresentação

Propósito

1. Autenticação do servidor
2. Negociação: concordar sobre algoritmos de criptografia
3. Estabelecer chaves
4. Autenticação do cliente (opcional)

SSL real: Apresentação

1. Cliente envia lista de algoritmos que admite, junto com nonce do cliente.
2. Servidor escolhe algoritmos da lista; envia de volta: escolha + certificado + nonce do servidor.
3. Cliente verifica certificado, extrai chave pública do servidor, gera pre_master_secret, criptografa com chave pública do servidor, envia ao cliente.
4. Cliente e servidor calculam independentemente chaves de criptografia e MAC a partir de pre_master_secret e nonces.
5. Cliente envia um MAC de todas as mensagens de apresentação.
6. Servidor envia um MAC de todas as mensagens de apresentação.

SSL real: Apresentação

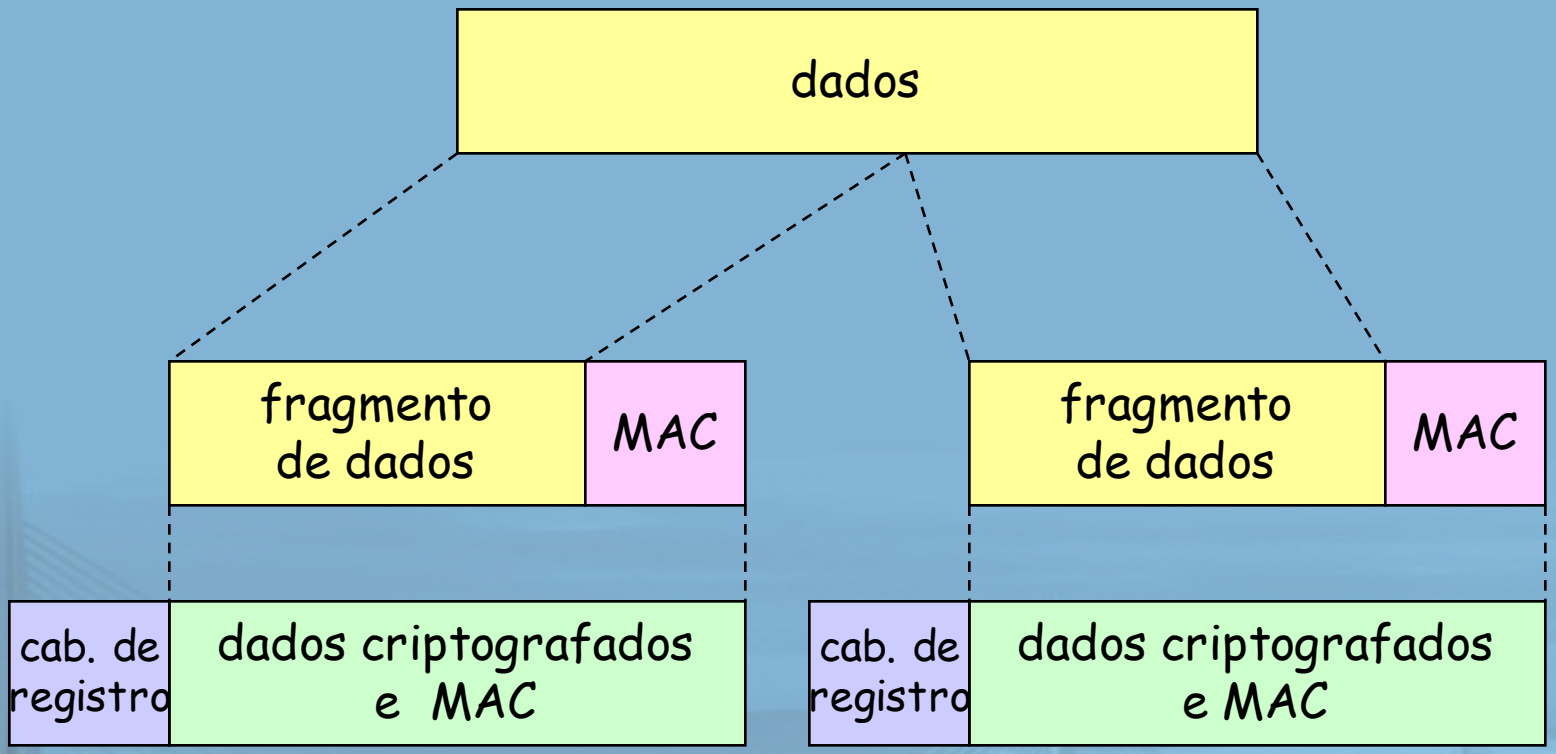
últimas 2 etapas protegem apresentação contra adulteração

- cliente normalmente oferece intervalo de algoritmos, alguns fortes, alguns fracos
- "homem do meio" poderia excluir da lista os algoritmos mais fortes
- últimas 2 etapas impedem isso
  - duas últimas mensagens são criptografadas

SSL real: Apresentação

- Por que os dois nonces aleatórios?
- Suponha que Trudy fareje todas as mensagens entre Alice & Bob.
- No dia seguinte, Trudy configura conexão TCP com Bob, envia a mesma sequência exata de registros.
  - Bob (Amazon) pensa que Alice fez dois pedidos separados para a mesma coisa.
  - Solução: Bob envia nonce aleatório diferente para cada conexão. Isso faz com que as chaves criptográficas sejam diferentes nos dois dias.
  - As mensagens de Trudy falharão na verificação de integridade de Bob.

Protocolo de registro SSL

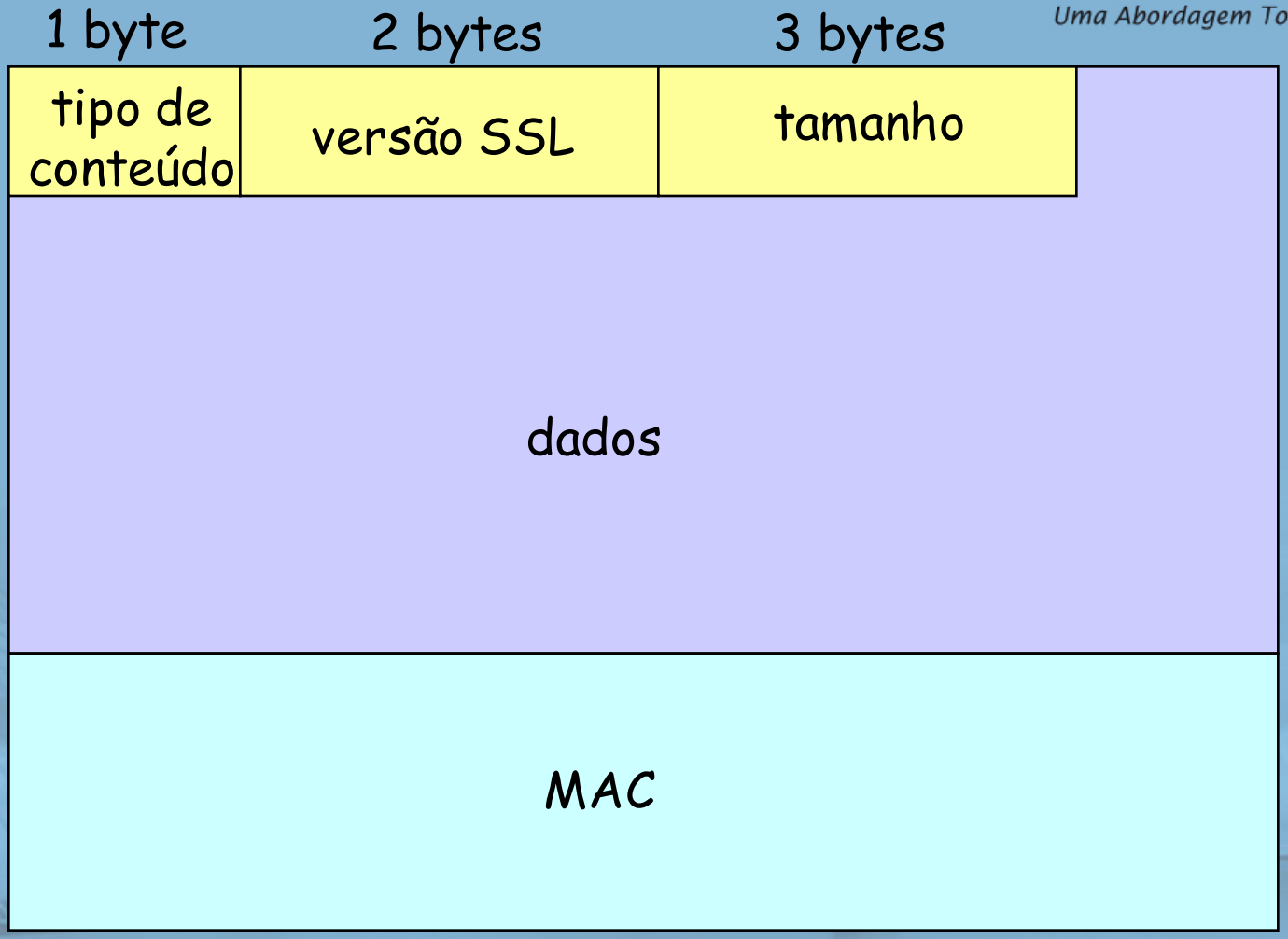


cabeçalho de registro: tipo de conteúdo; versão; tamanho

MAC: inclui número de sequência, chave MAC M_x

Fragmento: cada fragmento SSL 2^{14} bytes (~16 Kbytes)

Formato de registro SSL



Dados e MAC criptografados (algoritmo simétrico)

Conexão real

apresentação: ClientHello

apresentação: ServerHello

apresentação: Certificate

apresentação: ServerHelloDone

apresentação: ClientKeyExchange

ChangeCipherSpec

apresentação: Finished

ChangeCipherSpec

apresentação: Finished

application_data

application_data

Alerta: warning, close_notify

Tudo daqui para a frente é criptografado

TCP FIN em seguida

Derivação de chave

- nonce do cliente, nonce do servidor e segredo pre-master entram no gerador de número pseudoaleatório.
  - produz segredo mestre
- segredo mestre e novos nonces inseridos em outro gerador de número aleatório: "bloco de chaves"
  - devido à retomada: TBD
- bloco de chaves fatiado e dividido:
  - chave MAC do cliente
  - chave MAC do servidor
  - chave de criptografia do cliente
  - chave de criptografia do servidor
  - vetor de inicialização (IV) do cliente
  - vetor de inicialização (IV) do servidor