Criptografia Aplicada

Introdução a protocolos e TLS

Sumário

- Contextualização
- Protocolo TLS
- HTTPS
- Segurança do protocolo
- TLS na prática

Contextualização

- Os tópicos de criptografia estudados até o momento são fundamentais para a garantia de segurança de comunicações
- Mas eles sozinhos não resolvem todos os problemas
  - Precisamos de uma combinação de primitivas criptográficas para atingir os objetivos desejados
  - Confidencialidade, Integridade, Autenticação, Anonimato, Temporalidade, Não-repúdio
- Existe uma variedade muito grande de ataques que precisam ser levados em consideração
  - como escutas (*eavesdropping*), roubos de dados, alterações de dados (*tampering*) e personificação (*impersonation*).
  - Criptografia por si só não protege de todos esses ataques

Introdução aos Protocolos de Segurança

- Protocolos de segurança são fundamentais para estabelecer confiança em comunicações via redes de computadores
- Definem um conjunto de procedimentos para assegurar que os dados transmitidos sejam seguros
  - confidencialidade, integridade, autenticidade, etc.
- Através deles, podemos:
  - negociar algoritmos a serem utilizados;
  - autenticar os participantes da comunicação;
  - fazer um acordo seguro de chaves (de sessão, de cifragem, etc.);
  - estabelecer uma comunicação segura;
  - entre outros.
- Exemplos:
  - SSL/TLS, IPsec, SSH, Kerberos.

Conceitos básicos de criptografia

- Criptografia Simétrica
  - mesma chave (simétrica) para cifragem e decifragem
  - garante confidencialidade
  - ex: AES
- Criptografia Assimétrica
  - par de chaves: pública e privada
  - provê cifragem/decifragem, assinaturas digitais e troca de chaves
  - ex: RSA, DH, ECDSA
- Certificados digitais
  - provê a autenticação da identidade do dono de uma chave pública
  - assinado por uma autoridade certificadora (AC)

Sumário

- Contextualização
- **Protocolo TLS**
- HTTPS
- Segurança do protocolo
- TLS na prática

TLS

- **História**:
  - Transport Layer Security (TLS) é o sucessor do Secure Sockets Layer (SSL)
  - Desenvolvido inicialmente pela Netscape em 1995
  - Versões 1.0, 1.1, 1.2 e 1.3
- **O que é:**
  - Serviço de propósito geral implementado como um conjunto de protocolos
  - Garante confidencialidade, integridade e autenticidade das comunicações
  - Fica entre a camada de transporte (TCP) e a camada de serviços
- **Importância:**
  - HTTPS, VPNs, e-mail seguro, etc.
  - Essencial para transações online e proteção de dados

Imagem:
https://www.cloudflare.com/learning/ssl/what-is-ssl/

LabSEC
Laboratório de Segurança em Computação

cpqd

TLS - Overview

- Cliente e servidor se introduzem e decidem os algoritmos criptográficos que serão utilizados
- Servidor se autentica para o cliente com seu certificado digital
- Cliente verifica a validade do certificado
- Alice e Bob começam o processo de acordo de chaves para dar início a uma comunicação cifrada

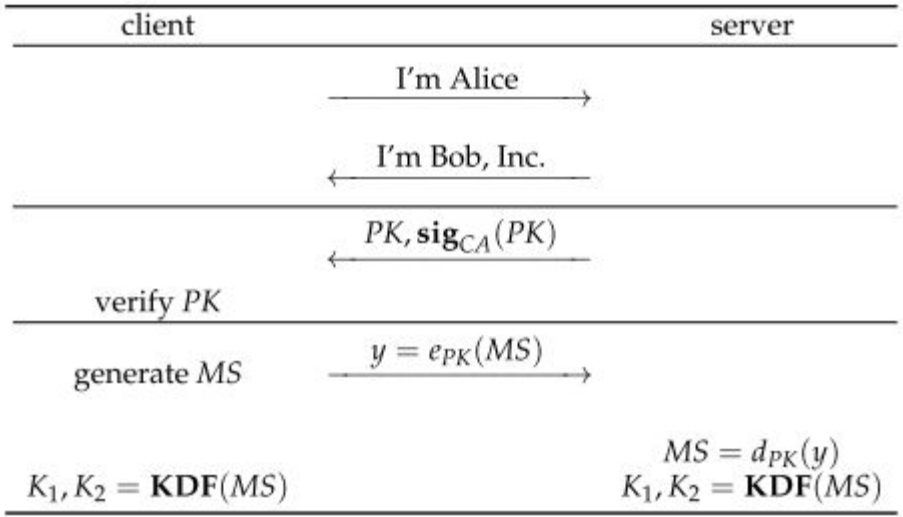


FIGURE 12.1: Setting up a TLS Session

Imagem: D. Stinson e M. Paterson. *Cryptography: Theory and Practice*. 4a edição. Capítulo 12.1.

LabSEC
Laboratório de Segurança em Computação

cpqd

Conjunto de protocolos do TLS

- Record Protocol
- Change cipher spec
- Alert protocol
- Handshake protocol

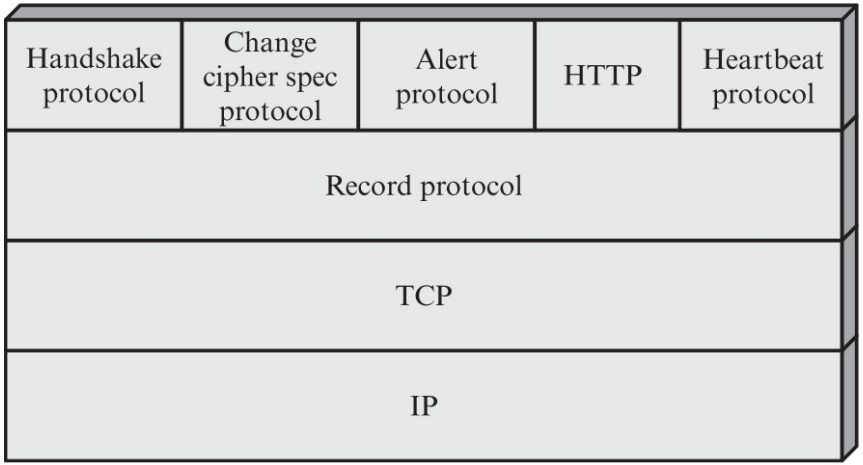


Figure 17.2 TLS Protocol Stack

Imagem: W. Stallings. *Cryptography and network security*. Cap 17.2

LabSEC
Laboratório de Segurança em Computação

cpqd

Conjunto de protocolos do TLS

Record Protocol: provê confidencialidade e integridade para conexões TLS

- confidencialidade através da cifragem
- integridade utilizando MACs
- Fragmenta a mensagem em blocos, protege os blocos e transmite o resultado
- Verifica, decifra e combina os dados recebidos, depois encaminha para o usuário

Change cipher spec: uma mensagem de 1 byte que sinaliza que o conjunto de cifras a ser usado nesta conexão deve ser atualizado.

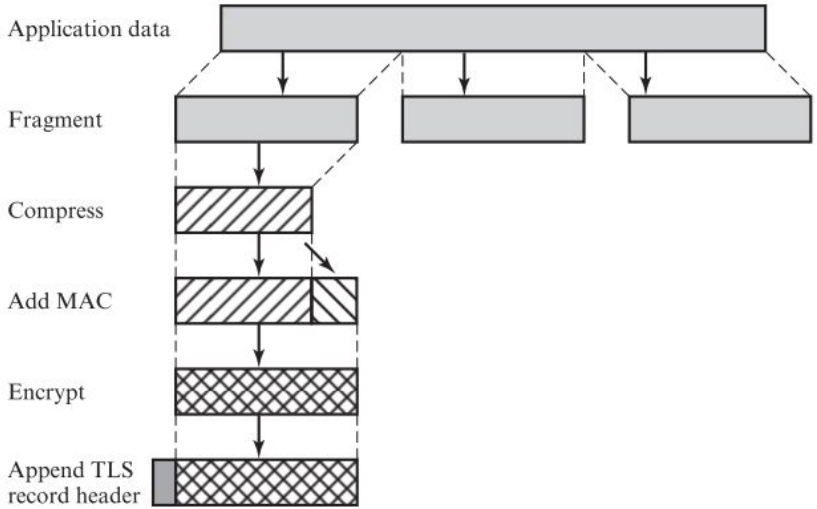


Figure 17.3 TLS Record Protocol Operation

Imagem: W. Stallings. *Cryptography and network security*. Cap 17.2

LabSEC
Laboratório de Segurança em Computação

cpqd

Conjunto de protocolos do TLS

Alert protocol: Usado para notificar erros ou eventos importantes durante a sessão TLS.

- alertas são comprimidos e cifrados de acordo com o estado atual da comunicação
- **níveis**: *warning* e *fatal*
- **classes**:
  - *closure alerts*: indicam que a conexão será finalizada/cancelada
  - *error alerts*: enviado pela parte que detectou o erro. Ambas as partes devem imediatamente finalizar a conexão
- Ver mais detalhes em: RFC8446

```
enum { warning(1), fatal(2), (255) } AlertLevel;

enum {
    close_notify(0),
    unexpected_message(10),
    bad_record_mac(20),
    record_overflow(22),
    handshake_failure(40),
    bad_certificate(42),
    unsupported_certificate(43),
    certificate_revoked(44),
    certificate_expired(45),
    certificate_unknown(46),
    illegal_parameter(47),
    unknown_ca(48),
    access_denied(49),
    decode_error(50),
    decrypt_error(51),
    protocol_version(70),
    insufficient_security(71),
    internal_error(80),
    inappropriate_fallback(86),
    user_canceled(90),
    missing_extension(109),
    unsupported_extension(110),
    unrecognized_name(112),
    bad_certificate_status_response(113),
    unknown_psk_identity(115),
    certificate_required(116),
    no_application_protocol(120),
    (255)
} AlertDescription;

struct {
    AlertLevel level;
    AlertDescription description;
} Alert;
```

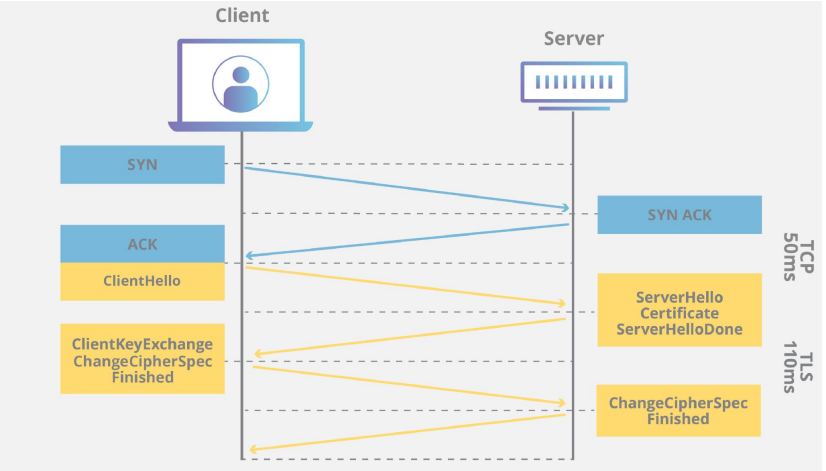
LabSEC
Laboratório de Segurança em Computação

cpqd

Conjunto de protocolos do TLS

Handshake protocol: negociação dos parâmetros de segurança de uma conexão

- Utilizado antes de qualquer transmissão de dados
- Consiste em uma série de mensagens trocadas entre cliente e servidor
- Cliente e servidor se autenticam um para o outro e definem os algoritmos criptográficos que serão usados
- Cada mensagem trocada tem os campos como tipo, tamanho e conteúdo

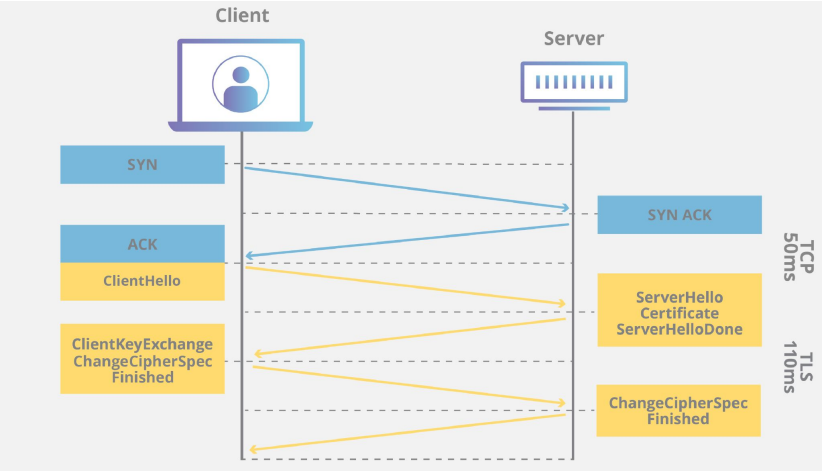


LabSEC
Laboratório de Segurança em Computação

cpqd

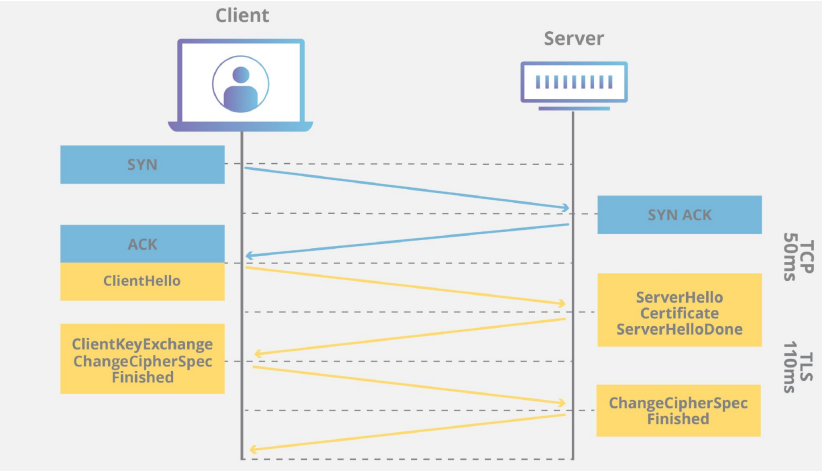
Handshake protocol

- Client hello:
  - versão do TLS compatível
  - conjuntos de algoritmos criptográficos compatíveis
    - grupos suportados para EC
    - algoritmos de assinatura
  - string de bytes aleatórios
- Server hello:
  - certificado SSL do servidor
  - algoritmos criptográficos escolhidos
  - string de bytes aleatórios
- Autenticação
  - cliente verifica o certificado SSL do servidor com a AC que o emitiu
  - cliente garante a interação com o verdadeiro proprietário do domínio



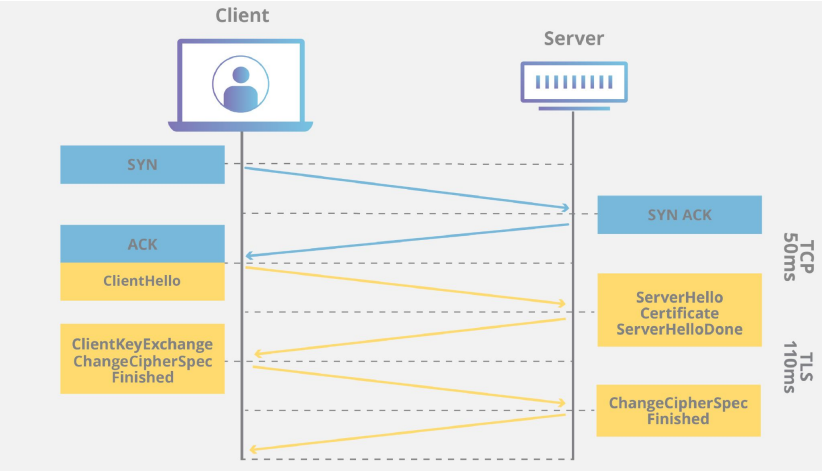
Handshake protocol

- Acordo de chaves
  - cliente envia uma pré-*master secret* cifrada com a chave pública do servidor
  - a pré-*master secret* é utilizada pelo cliente e servidor para gerarem a mesma chave de sessão
- Comunicação é cifrada a partir daqui



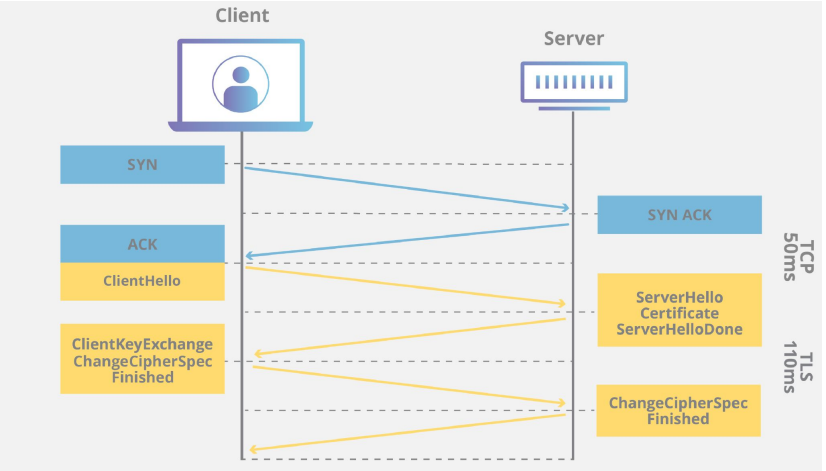
Handshake protocol

- Alternativa: ephemeral Diffie-Hellman
  - servidor envia o seu certificado e junto dele a assinatura de todas as mensagens até aqui
  - cliente verifica a assinatura e certificado
  - cliente envia parâmetros DH para para o servidor
  - cliente e servidor calculam a chave de sessão usando DH



Modo 0-RTT

- Ocorre caso cliente e servidor já tenham se conectado antes
- derivam outra chave secreta a partir das informações da primeira sessão
- não há necessidade de comunicação para handshake



Cipher suites

- Combinações de algoritmos de criptografia utilizados pelo TLS para garantia de uma coneção segura
- Cifragem:
  - TLS_AES_128_GCM_SHA256
  - TLS_AES_256_GCM_SHA384
  - TLS_CHACHA20_POLY1305_SHA256
- Assinatura digital
  - rsa_pkcs1_sha256
  - rsa_pss_rsae_sha256
  - ecdsa_secp256r1_sha256

Imagem:
https://www.cloudflare.com/learning/ssl/what-is-ssl/

Resumindo o TLS

- Tem por objetivo de garantir a privacidade, autenticação e integridade de dados nas comunicações.
- A identidade das partes é **autenticada** usando criptografia de chave pública. Pode ser mútua ou pelo menos pelo lado do servidor
- **Conexão privada** ao compartilhar uma chave de sessão simétrica e utilizá-la para criptografar os dados transmitidos
- É possível verificar a **integridade** de cada mensagem transmitida.
- Com todas essas etapas, garantimos uma conexão segura entre cliente e servidor.

Sumário

- Contextualização
- Protocolo TLS
- **HTTPS**
- Segurança do protocolo
- TLS na prática

HTTPS

- HTTPS (HTTP over SSL) se refere à combinação de SSL com HTTP para a implementação de uma comunicação segura entre browser e servidor web.
- HTTP utiliza a porta 80 enquanto o HTTPS utiliza a 443
- Ao utilizar HTTPS, as seguintes informações são cifradas:
  - URL
  - conteúdo da comunicação entre cliente e servidor
  - dados preenchidos pelo usuário
  - cookies
  - cabeçalhos HTTP
- Mais informações: RFC2818

Imagem:
https://www.cloudflare.com/learning/ssl/what-is-ssl/

Sumário

- Contextualização
- Protocolo TLS
- HTTPS
- **Segurança do protocolo**
- TLS na prática

Segurança do protocolo

- Desde a criação do SSL e depois do TLS, inúmeros ataques foram encontrados contra esses protocolos
  - ataques no protocolo de *handshake*
  - ataques no protocolo *record* e nos dados de aplicação
    - BEAST (Browser Exploit Against SSL/TLS) - 2011
  - ataques na validação dos certificados
  - etc.
- Cada ataque iniciou um processo de mudança no protocolo, como a troca dos algoritmos utilizados ou aspectos de implementação.
- TLS 1.3 eliminou a compatibilidade com algoritmos vulneráveis, tornado o protocolo mais resistente à ataques.

Sumário

- Contextualização
- Protocolo TLS
- HTTPS
- Segurança do protocolo
- **TLS na prática**

Atividade: conexão TLS com openssl

- Podemos utilizar o openssl para analisar e depurar conexões TLS
- Conectar com um servidor remoto usando TLS:

```
openssl s_client -connect <endereço_do_servidor>:<porta>
```

- Escolha um servidor/site e identifique:
  - a versão TLS utilizada
  - a cadeia de certificação até o certificado do servidor
  - as cifras utilizadas
- Documentação

Atividade: análise de pacotes com wireshark

- O Wireshark é um analisador de protocolos
- Com ele, você poderá analisar o tráfego de rede em tempo real
- Você pode examinar os pacotes, dados transmitidos, entre outras informações
- Atividade: utilizar o Wireshark para analisar os pacotes de dados do TLS
  - escolha um website qualquer que utilize https
  - identifique a troca de mensagens entre cliente e servidor
  - identifique o protocolo de handshake
  - identifique a versão do TLS utilizada, algoritmos disponíveis e utilizados na comunicação
- Tutorial para uso do wireshark com TLS aqui

Referências

- W. Stallings. *Cryptography and network security*. 7a edição. Capítulos 17.2 e 17.3.
- D. Stinson e M. Paterson. *Cryptography: Theory and Practice*. 4a edição. Capítulo 12.1.
- RFC8446: The Transport Layer Security (TLS) Protocol Version 1.3
- RFC2818: HTTP Over TLS
- TLS handshake