TLS and Signal

Transport Layer Security (TLS)

> O navegador da Netscape foi um navegador extremamente popular em 1990 (com 90% dos usuários usando-o), mas que no entanto perdeu a “guerra do primeiro navegador” para o Internet Explorer. Hoje em dia o Firefox é considerado o sucessor do Netscape. Os serviços do Netscape foram encerrados em 2004 (mais ou menos), com menos de 0.4% dos usuários utilizando-o.

O browser Netscape foi de extrema importância histórica por ser responsável pelo desenvolvimento de protocolos importantes como o SSL (Secure Socket Layer).

Diffie Hellman Autenticado

Hoje em dia é evidente as vantagens de se usar este tipo de protocolo, não só por conta da eficiência (**uso de curvas elípticas**), mas também por conta do **perfect forward secrecy,** que diz que se uma chave de longa duração for comprometida, então as sessões anteriores não podem ser, porque cada sessão possui um **Diffie Hellman** independente. As chaves de longa duração são utilizadas apenas para assinar.

TLS implica PKI

O servidor autentica a troca de DH, mas a chave do cliente é opcional.

O passos são:

- O cliente estabelece a ligação
- O servidor envia o certificado

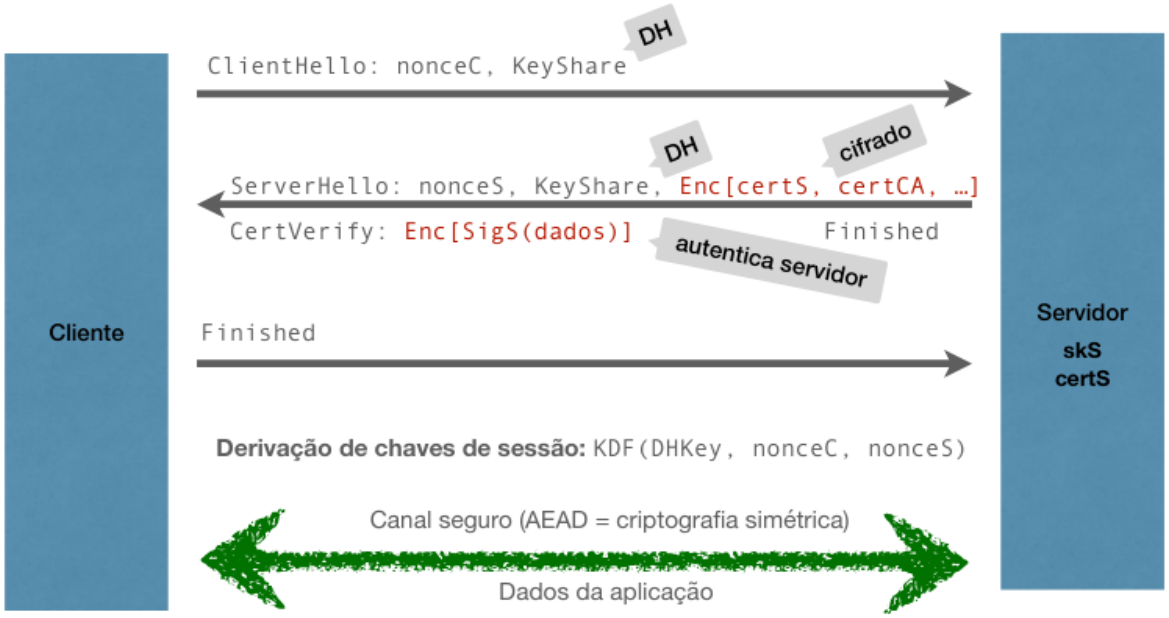
- O cliente valida através dos certificados já instalados por default. O nome do domínio faz match com subject do certificado.

O cliente não é autenticado.

Handshake

> An overview of the SSL or TLS handshake
>
> The SSL or TLS handshake enables the SSL or TLS client and server to establish the secret keys with which they communicate.
>
> https://www.ibm.com/docs/en/ibm-mq/7.5?topic=ssl-overview-tls-handshake

A simple explanation of how these schema below works.



Encripta certificados para garantir anonimidade do domínio acedido.

0 - RTT

O TLS 1.3 tenta maximar o anonimato e o problema é que algumas aplicações tentam transmitir uma **0-RTT (dados)** que foram previamente partilhados.

Integração TLS/HTTP

As mensagens HTTP são enviadas escondidas no payload do TLS. Algumas aplicações sofrem com isto. Um web proxy, por exemplo, precisa saber o cabeçalho HTTP para estabelecer uma ligação.

Problemas:

- O cliente deve informar ao proxy para qual site as mensagens devem ser enviadas. O proxy não deseja ver nossas mensagens e é exatamente por isso que não se sabe para onde deve-se enviar a informação.
- Ainda temos o problema de **virtual hosts,** são sites com o mesmo IP (ou seja, vários sites na mesma máquina) e com múltiplos DNS. Como sabe o servidor que certificado devolver?
  A solução antiga é o client-hello incluir o nome de domínio do servidor. O TLS 1.3, por outro lado, tenta preservar a privacidade do nome de domínio (certificado cifrado). A solução futura é incluir no helloClient o nome do domínio com quem se quer falar, mas cifrado com uma chave pública.

HTTPS para todo o tráfego

Antigamente havia muita crítica ao uso do HTTPS por conta da performance. Mas hoje em dia o hardware é adaptado para isto e o uso de curvas elípticas acelerou muito o processo.

HTTPS nos browsers

Até que ponto valorizamos se o aloquete está totalmente verde?

Alguem pode convencer a clicar num link http://www.paypal.com. O servidor do paypal vai nos redirecinar para uma comunicação `https`, mas pode estar alguem no meio da comunicação que se comunica com o paypal usando `https`, mas se comunica com a vítima em http.

Este se chama um ataque SSL Strip Attack. Por isso os browsers avisam quando há um elemento não https.

Ainda é possível deixar na cache do browser uma flag chamada (Strict Transport Security) que vai exigir com que o browser fale com domínio daí pra frente sempre com

https, mas isto desaparece com a limpeza da cache. Isto é uma boa prática nas aplicações web.

Problemas TLS

O TLS ainda revela muita informação.

Isto também se aplica a tráfego transferido via TOR.

Perguntas

1) Como é feita a ligação do Diffie-Hellman autenticado.