Лекция №6

Криптографические протоколы. TLS

Елена Киршанова

Асимметричная+ Симметричная криптография

Гибрид

Обмен ключами /KEM, безопасные против активных атакующий +
Симметричное шифрование с аутентификацией –

центральное комбо в современных протоколах.

TLS: Transport Layer Security (Протокол защиты транспортного уровня)

TLS – протокол установления и поддержки безопасного соединения между клиентов и серверов по Интернету.

I. SSL = Secure Socket Layer

- SSLv1 (1994) — не опубликован
- SSLv2 (1995) — взломан
- SSLv3 (1996) — поддерживается

II. TLS = Transport Layer Security

- TLS 1.0 (1999) — RFC 2246
- TLS 1.1 (2006) — RFC 4346
- TLS 1.2 (2008) — RFC 5246
- TLS 1.3 (2018) — RFC 8448

`https://datatracker.ietf.org/doc/rfc8446/`

Стандарты IETF

RFC = Request for Comments
IETF = Internet Engineering Task Force

Структура протокола TLS

`Клиент` — `Сервер`

Фаза 1 Handshake

выбор примитивов, параметров

аутентификация (как мин. сервера)

генерация общего ключа

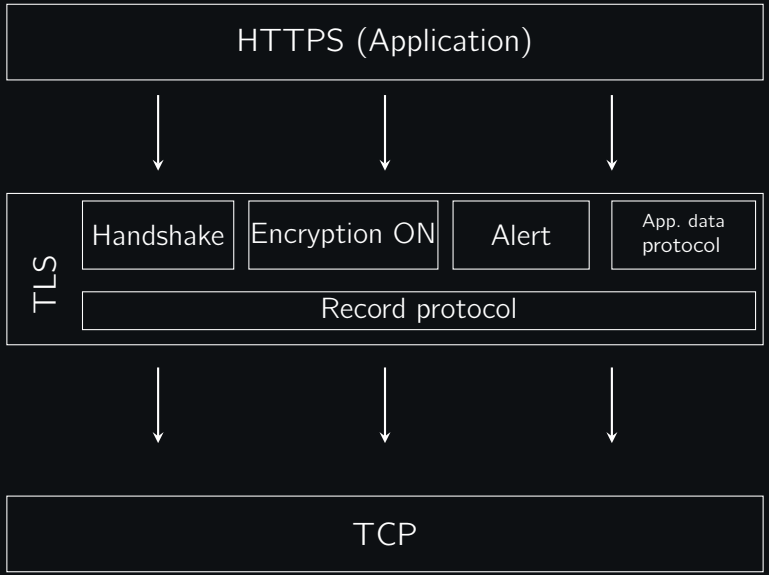
$\downarrow k$

Фаза 2 TLS record protocol

шифрование данных с помощью AEAD с ключом k

TLS живёт на транспортном уровне TCP/IP, т.е., пакеты приходят в корректном порядке.

Где живёт TLS



Слайд взят из презентации K.Patterson `https://www.isg.rhul.ac.uk/~kp/TLSwheredowestand.pdf`

Фаза1: TLS Handshake

Клиент

Сервер

$\text{pk}_c = g^a$, Nonce N_c, offer

offer: список шифров клиентов

Фаза1: TLS Handshake

Клиент

$\text{pk}_c = g^a$, Nonce N_c, offer

offer: список шифров клиентов

Сервер

1. Выбор шифра
(Enc. scheme, hash)
2. Вычисляет $k_{\text{shared}} = g^{ab}$
k_{sh} – ключ шифрования сервера
k_{sm}– ключа MAC'а сервера
k_{ch} - ключ шифрования клиента
k_{cm} -ключа MAC'а клиента

$\text{pk}_s = g^b$, Nonce N_s, режим

mode: chosen cipher suits

Фаза1: TLS Handshake

Клиент

Сервер

$\text{pk}_c = g^a$, Nonce N_c, offer

offer: список шифров клиентов

1. Выбор шифра
(Enc. scheme, hash)
2. Вычисляет $k_{\text{shared}} = g^{ab}$
k_{sh} – ключ шифрования сервера
k_{sm}– ключа MAC'а сервера
k_{ch} - ключ шифрования клиента
k_{cm} -ключа MAC'а клиента

$\text{pk}_s = g^b$, Nonce N_s, режим

mode: chosen cipher suits

3. Вычисляет $k_{\text{shared}} = g^{ab}$
$k_{\text{sh}}, k_{\text{sm}}, k_{\text{ch}}, k_{\text{cm}}$

Фаза1: TLS Handshake

Клиент

Сервер

$\mathsf{pk}_c = g^a$, Nonce N_c, offer

offer: список шифров клиентов

1. Выбор шифра
(Enc. scheme, hash)
2. Вычисляет $k_{\mathsf{shared}} = g^{ab}$
k_{sh} – ключ шифрования сервера
k_{sm}– ключа MAC'а сервера
k_{ch} - ключ шифрования клиента
k_{cm} -ключа MAC'а клиента

$\mathsf{pk}_s = g^b$, Nonce N_s, режим

mode: chosen cipher suits

3. Вычисляет $k_{\mathsf{shared}} = g^{ab}$
$k_{\mathsf{sh}}, k_{\mathsf{sm}}, k_{\mathsf{ch}}, k_{\mathsf{cm}}$

$c_1 = \mathsf{Enc}(k_{\mathsf{sh}},\ \text{Cert. request})$
$c_2 = \mathsf{Enc}(k_{\mathsf{sh}},\ \text{Cert. Server})$
$c_3 = \mathsf{Enc}(k_{\mathsf{sh}}, \mathsf{Sign}(\text{transcript}))$
$c_4 = \mathsf{Enc}(k_{\mathsf{sh}}, \mathsf{MAC}(k_{\mathsf{sm}}, \text{transcript}))$

Фаза1: TLS Handshake

Клиент

Сервер

$\text{pk}_c = g^a$, Nonce N_c, offer

offer: список шифров клиентов

1. Выбор шифра
(Enc. scheme, hash)
2. Вычисляет $k_{\text{shared}} = g^{ab}$
k_{sh} – ключ шифрования сервера
k_{sm}– ключа MAC'а сервера
k_{ch} - ключ шифрования клиента
k_{cm} -ключа MAC'а клиента

$\text{pk}_s = g^b$, Nonce N_s, режим

mode: chosen cipher suits

3. Вычисляет $k_{\text{shared}} = g^{ab}$
$k_{\text{sh}}, k_{\text{sm}}, k_{\text{ch}}, k_{\text{cm}}$

$c_1 = \text{Enc}(k_{\text{sh}},\ \text{Cert. request})$
$c_2 = \text{Enc}(k_{\text{sh}},\ \text{Cert. Server})$
$c_3 = \text{Enc}(k_{\text{sh}}, \text{Sign}(\text{transcript}))$
$c_4 = \text{Enc}(k_{\text{sh}}, \text{MAC}(k_{\text{sm}}, \text{transcript}))$

$c_5 = \text{Enc}(k_{\text{ch}},\ \text{Cert. Client})$
$c_6 = \text{Enc}(k_{\text{ch}}, \text{Sign}(\text{transcript}))$
$c_7 = \text{Enc}(k_{\text{ch}}, \text{MAC}(k_{\text{cm}}, \text{transcript}))$

$(k_{c\to s}, k_{s\to c}) = \mathcal{H}(\text{transcript})$

$(k_{c\to s}, k_{s\to c}) = \mathcal{H}(\text{transcript})$

Фаза2: TLS Record Layer

Данные = $[m_1, \ldots, m_s]$

Клиент

$k_{c \to s}$

$k_{s \to c}$

Сервер

$k_{c \to s}$

$k_{s \to c}$

$$\underbrace{[\texttt{Meta data} || \, m_i \, || \, \texttt{Nonce}]}_{\text{AES-GCM-AEAD}(k_{c \to s})} \longrightarrow$$

Безопасность

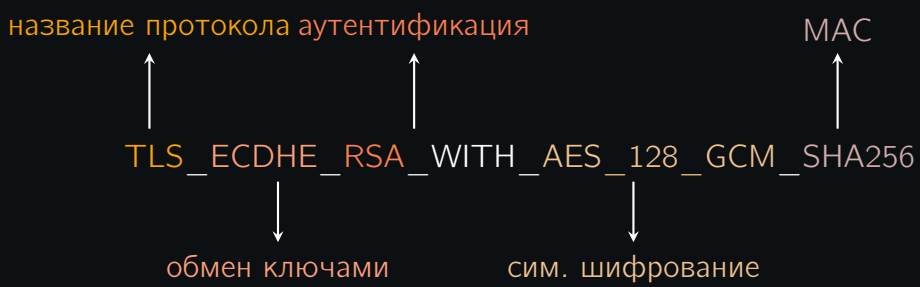
- Протокол Alert ответственен за обработку ошибок, предупреждений и окончания сессий
- Существуют формальные доказательства безопасности TLS 1.3
- Обновление ключа: получив сообщение `KeyUpdate` Клиент и Сервера обновляют $k_{c \to s}, k_{s \to c}$
- Возобновление сессии (Pre-shared key handshake): более эффективная фаза Handshake Phase, если между клиентом и сервером ранее уже были установлены сессии
- Forward secrecy: если злоумышленник получает общий ключ, *предыдущие* сообщения остаются конфиденциальными.

Наборы алгоритмов в TLS 1.3

Обмен ключами	Сертификаты	Сим. шифрование	Хэш-функции
ECDHE	ECDSA	AES_256_GCM	(H)SHA_384
DHE	RSA	CHACHA20_Poly1350	(H)SHA_256
RSA		AES_128_GCM	(H)SHA1
		AES_256_CBC	
		AES_128_CBC	
		3DES_CBC	

Декодирование названия шифра

TLS 1.2



TLS 1.3 наборы шифров по умолчанию:

TLS_AES_256_GCM_SHA384

TLS_CHACHA20_POLY1305_SHA256

TLS_AES_128_GCM_SHA256

Данные Вашего браузера / сервера

Используйте

`https://www.ssllabs.com/index.html`

для получения поддерживаемых версий SSL/TLS Вашего браузера или сервера