Ежегодная международная научно-практическая конференция
«РусКрипто’2019»

Принципы построения отечественных криптонаборов для TLS 1.3

Докладчик:
Евгений Алексеев,
Начальник отдела криптографических исследований,
ООО «КРИПТО-ПРО»

Протокол TLS

- Протокол установления аутентифицированного защищенного канала связи
- Работает поверх протокола TCP
- Имеет двухуровневую структуру: над TCP работает Record, над Record работают все остальные протоколы
- Основная криптографическая составляющая:
  - протокол **Record** – протокол обеспечения защищенного двустороннего канала связи
  - протокол **Handshake** – протокол аутентифицированной выработки общего ключа.

Протокол TLS

- Определяется документами IETF (rfc)
- Версии:
  - TLS 1.0 – rfc 2246 (1999 год)
  - TLS 1.1 – rfc 4346 (2006 год)
  - TLS 1.2 – rfc 5246 (2008 год)
  - TLS 1.3 – rfc 8446 (2018 год) + контрольные примеры rfc 8448 («Example Handshake Traces for TLS 1.3», 2019 год)
- Фиксированный каркас, возможность уточнять порядок работы с различными криптопримитивами в рамках криптонаборов (ciphersuites): rfc 5289 определяет криптонабор на основе AES, режима GCM и хэш-функции SHA256.

Криптонаборы TLS 1.2 с российскими криптоалгоритмами

- В 2019 году, криптонаборы для TLS 1.2, основанные на российских криптоалгоритмах получили официальные номера IANA!
- IANA Transport Layer Security (TLS) Parameters:
  - TLS_GOSTR341112_256_WITH_KUZNYECHIK_CTR_OMAC – 0xC1, 0x00
  - TLS_GOSTR341112_256_WITH_MAGMA_CTR_OMAC – 0xC1, 0x01
  - TLS_GOSTR341112_256_WITH_28147_CNT_IMIT – 0xC1, 0x02

Криптонаборы TLS 1.2 с российскими криптоалгоритмами

- Преимущества наличия номеров IANA:
  - Устранение возможности блокировки TLS с ГОСТ-ми из-за захвата номеров другими криптонаборами;
  - Легитимизация поддержки российских криптонаборов в свободном ПО (например, в OpenSSL);
  - Снижение вероятности конфликтов с эволюционными изменениями в TLS;
  - Поддержка сторонним по отношению к реализациям TLS ПО (например, Wireshark).

TLS 1.3

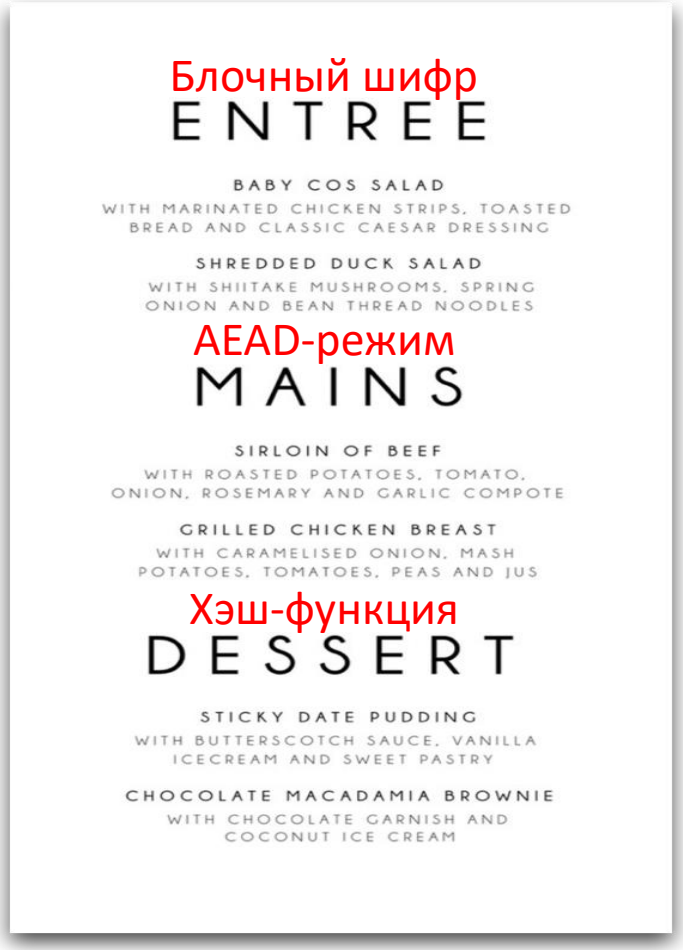
- Существенно переработан по сравнению с предыдущими версиями
- Учтены современные подходы к построения безопасных протоколов (возможность построения сведений к стойкости базовых криптопримитивов):

> - The key derivation functions have been redesigned. The new design allows easier analysis by cryptographers due to their improved key separation properties. The HMAC-based Extract-and-Expand Key Derivation Function (HKDF) is used as an underlying primitive.

- Особое внимание уделено обеспечению свойства Forward-Secrecy:

> - Static RSA and Diffie-Hellman cipher suites have been removed; all public-key based key exchange mechanisms now provide forward secrecy.

Вариативность в TLS 1.3



Криптонаборы в TLS 1.3

Криптонаборы в TLS 1.2

TLS 1.3 – строение и основные особенности

3 режима работы протокола Handshake:

- ECDHE-only
- PSK-only
- PSK-ECDHE

TLS 1.3 – строение и основные особенности

Схема обмена сообщениями в режиме ECDHE-only протокола Handshake.

Клиент		Сервер
1-я фаза ключевого обмена		
ClientHello: • supported_versions • signature_algorithms • supported_groups • key_share	------->	
		ServerHello: supported_versions • key_share •
		EncryptedExtensions supported_groups* •
		CertificateRequest*: signature_algorithms •
		Certificate
		CertificateVerify
		Finished
	<-------	Application Data*
Certificate*		
CertificateVerify*		
Finished	------->	
Application Data	<------>	Application Data
	<-------	NewSessionTicket*
KeyUpdate*	<------>	KeyUpdate*

Используемая система обозначений:

"*"	– опциональные данные;
	– сообщения, защищенные на ключах, выработанных из секретного значения [*sender*]_*handshake_traffic_secret* (см. подробнее 8.4);
	– сообщения, защищенные на ключах, выработанных из секретного значения [*sender*]_*application_traffic_secret_N* (см. подробнее 8.4).

TLS 1.3 – строение и основные особенности

Схема обмена сообщениями в режиме PSK-only протокола Handshake.

Клиент		Сервер
ClientHello • supported_versions • [signature_algorithms]* • [key_share]* • [supported_groups]* • pre_shared_key • psk_key_exchange_modes	------->	
		ServerHello supported_versions • pre_shared_key •
		EncryptedExtensions
		Finished
		Application Data*
Finished	------->	
Application Data	<------>	Application Data
	<-------	NewSessionTicket*
KeyUpdate*	<------>	KeyUpdate*

Используемая система обозначений:

*	– опциональные данные;
[]	– опциональное расширение, посылаемое клиентом в режиме восстановления соединения для обеспечения возможности перехода к Full Handshake;
	– сообщения, защищенные на ключах, выработанных из секретного значения [*sender*]_*handshake_traffic_secret* (см. подробнее 8.4);
	– сообщения, защищенные на ключах, выработанных из секретного значения [*sender*]_*application_traffic_secret_N* (см. подробнее 8.4).

TLS 1.3 – строение и основные особенности

Схема обмена сообщениями в режиме PSK-ECDHE протокола Handshake.

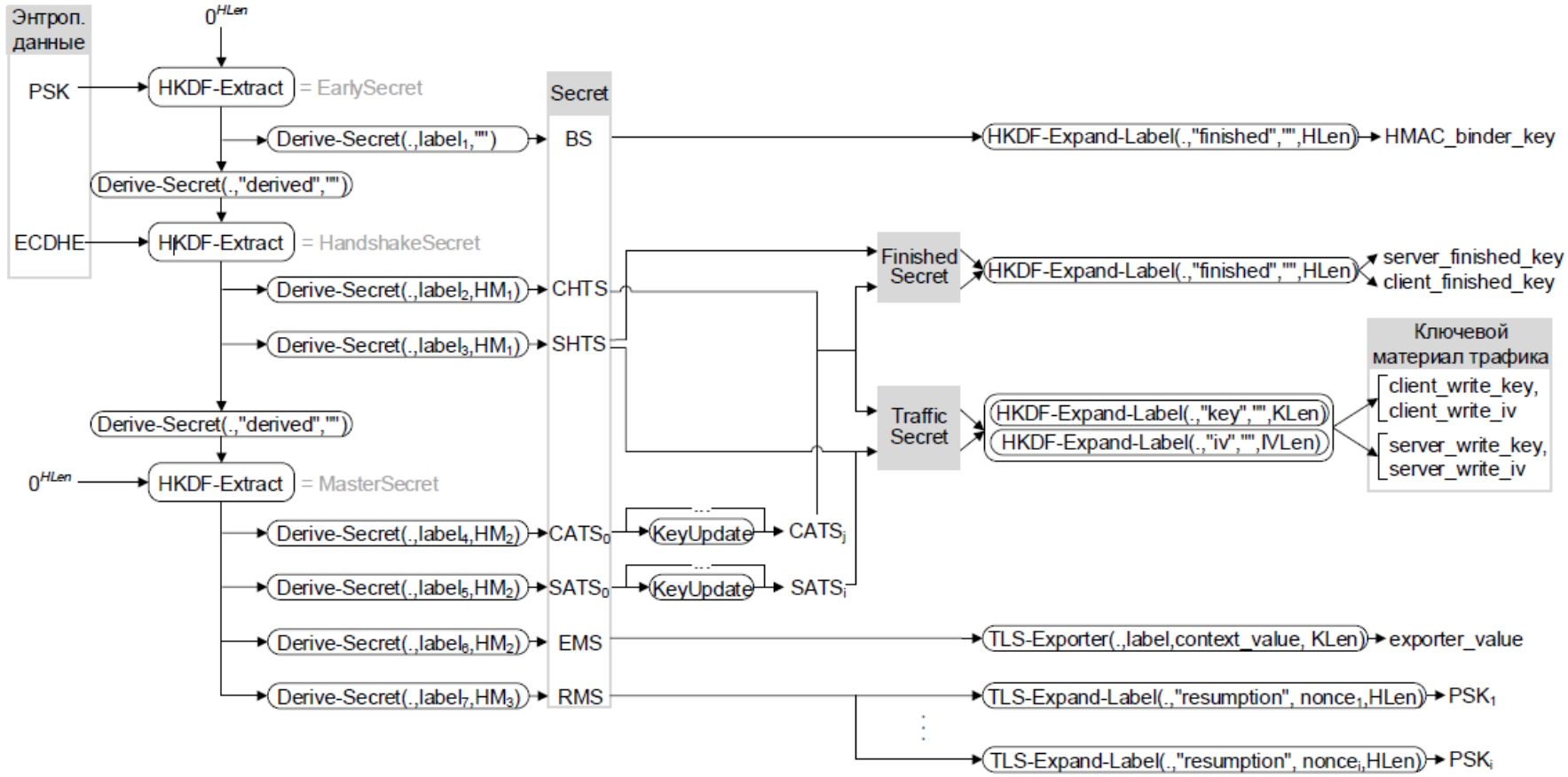
Клиент		Сервер
ClientHello: • supported_version • [signature_algorithms]* • key_share • supported_groups • pre_shared_key • psk_key_exchange_modes	------->	
		ServerHello: supported_versions • key_share • pre_shared_key •
		EncryptedExtensions: supported_groups* •
		Finished
	<-------	NewSessionTicket*
Finished	------->	
Application Data	<------>	Application Data
	<-------	NewSessionTicket*
KeyUpdate*	<------>	KeyUpdate*

Используемая система обозначений:

*	– опциональные данные;
[]	– опциональное расширение, посылаемое клиентом в режиме восстановления соединения для обеспечения возможности перехода к Full Handshake;
	– сообщения, защищенные на ключах, выработанных из секретного значения [sender]_handshake_traffic_secret (см. подробнее 8.4);
	– сообщения, защищенные на ключах, выработанных из секретного значения [sender]_application_traffic_secret_N (см. подробнее 8.4).

TLS 1.3 – строение и основные особенности

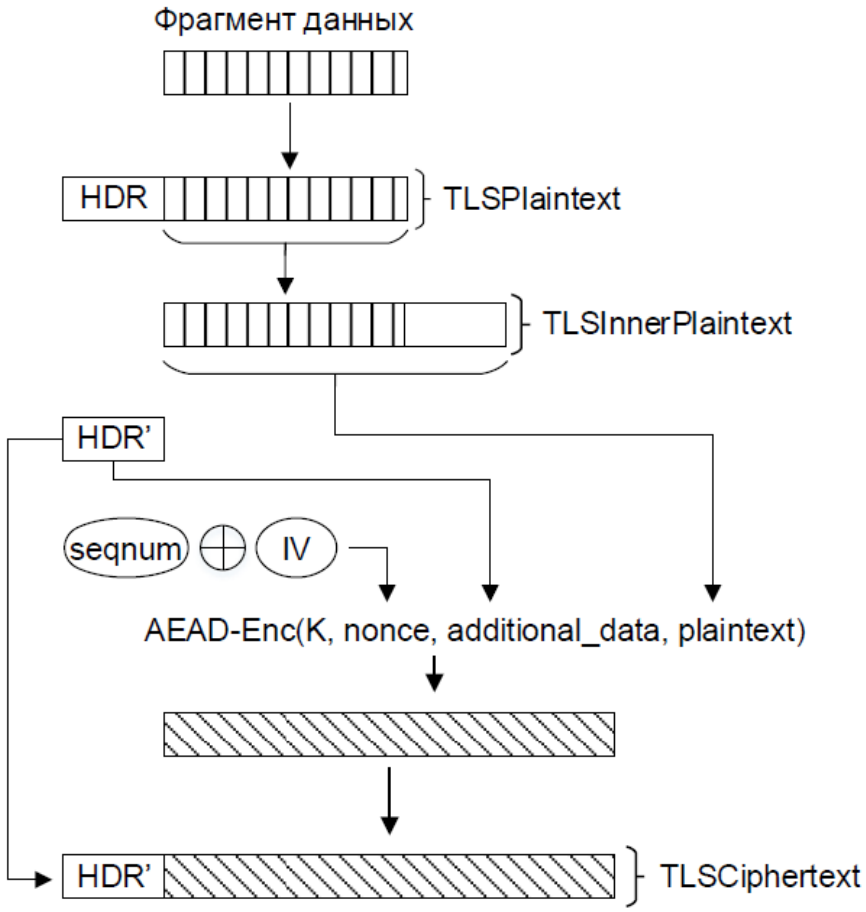
Дерево ключевого расписания в протоколе TLS 1.3:



Множество	Содержимое
HM_1	$\{ClientHello, \dots, ServerHello\}$
HM_2	$\{ClientHello, \dots, Finished$ со стороны сервера$\}$
HM_3	$\{ClientHello, \dots, Finished$ со стороны клиента$\}$

TLS 1.3 – строение Record

Схема формирования защищенной записи в протоколе **Record**:



TLS 1.3 – основные свойства Handshake

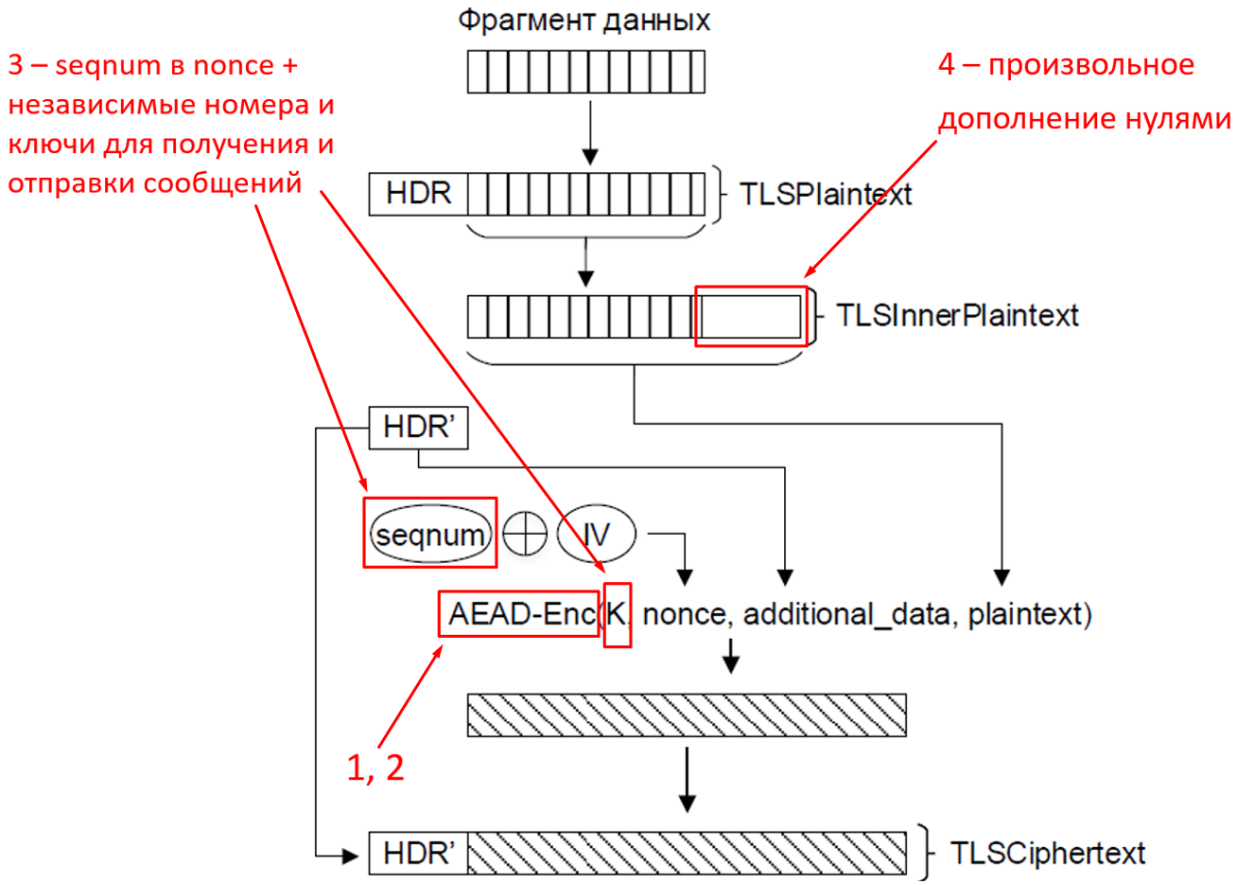
1. Установление одинаковых сессионных ключей
2. Секретность сессионных ключей
3. Аутентификация сторон
4. Уникальность и независимость ключей
5. Безопасность ранее установленных сессий при компрометации долговременных ключей (Forward Secrecy)
6. Стойкость в условиях KCI
7. Приватность

TLS 1.3 – основные свойства Handshake

- 1, 2, 3 – стандартное свойство AKE-протоколов
- 4 – обеспечивается за счет структуры ключевого дерева («improved key separation properties»)
- 5, 6 – обеспечивается за счет использования долговременных ключей только для аутентификации вырабатываемых ключей (при подписи данных)
- 7 – обеспечивается за счет шифрование всех сообщений кроме ClientHello, ServerHello и HelloRetryRequest

TLS 1.3 – основные свойства Record

1. Конфиденциальность
2. Целостность на уровне сообщений
3. Целостность на уровне потоков сообщений
4. Защита длины сообщений



TLS 1.3 с российскими криптоалгоритмами

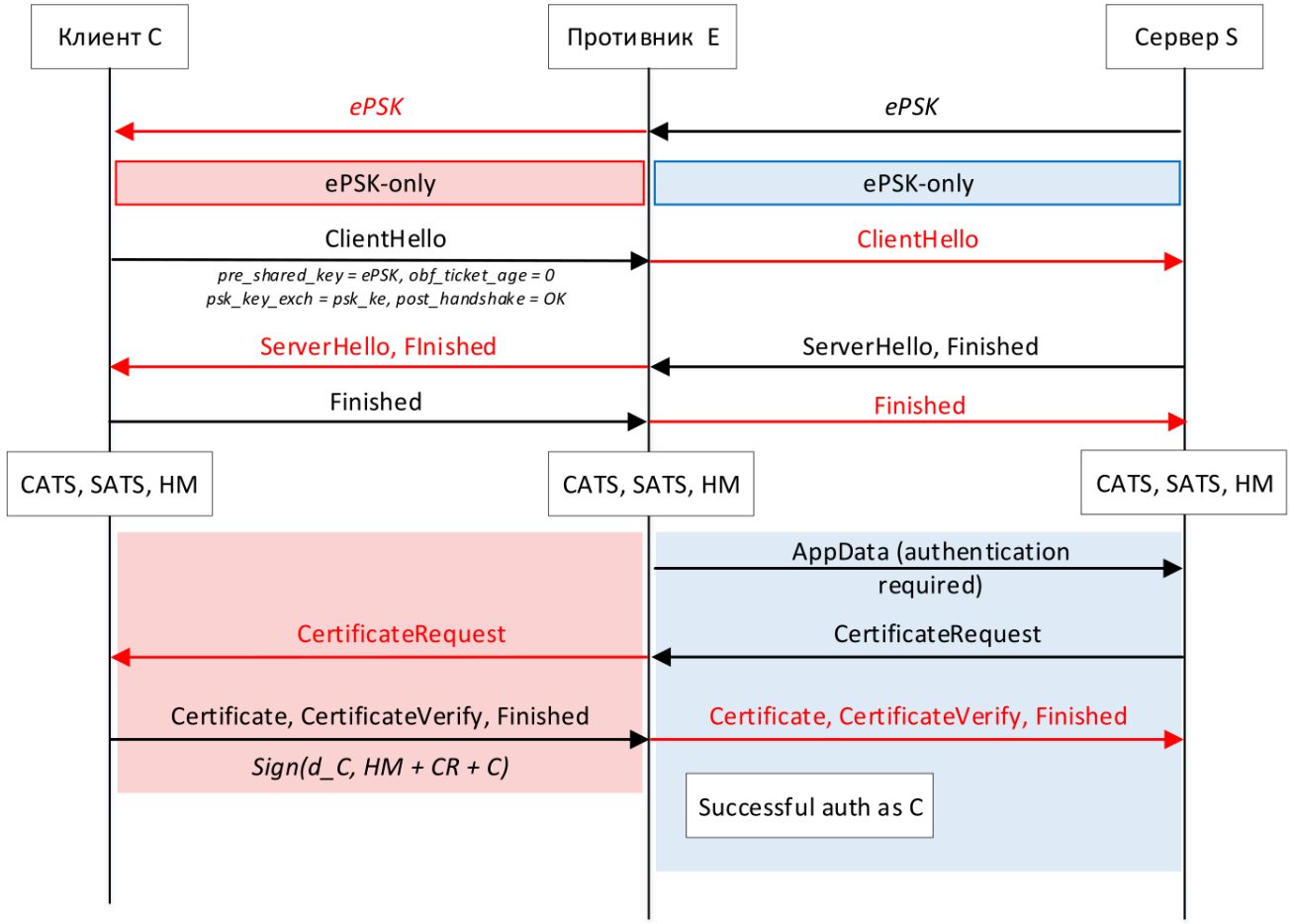
- Документ, определяющий криптонаборы на основе российских криптоалгоритмов, разрабатывается силами рабочей группы № 2.1 по сопутствующим криптографическим алгоритмам и протоколам (СКАиП) ТК26
- Планируемая дата окончания работы над документом: сентябрь 2019 года
- После завершения работы над документом планируется начать работу над rfc с целью получения номеров IANA и для этих криптонаборов

RTLS 1.3 – основные особенности

- Использование внешнего PSK (ePSK) только в режиме с ECDHE
- Ограничение возможности пересылки прикладных данных клиентом и сервером до завершения протокола Handshake
- Ограничение возможностей «продления жизни» соединений с помощью вырабатываемых PSK (iPSK)
- Порядок защиты данных в протоколе Record

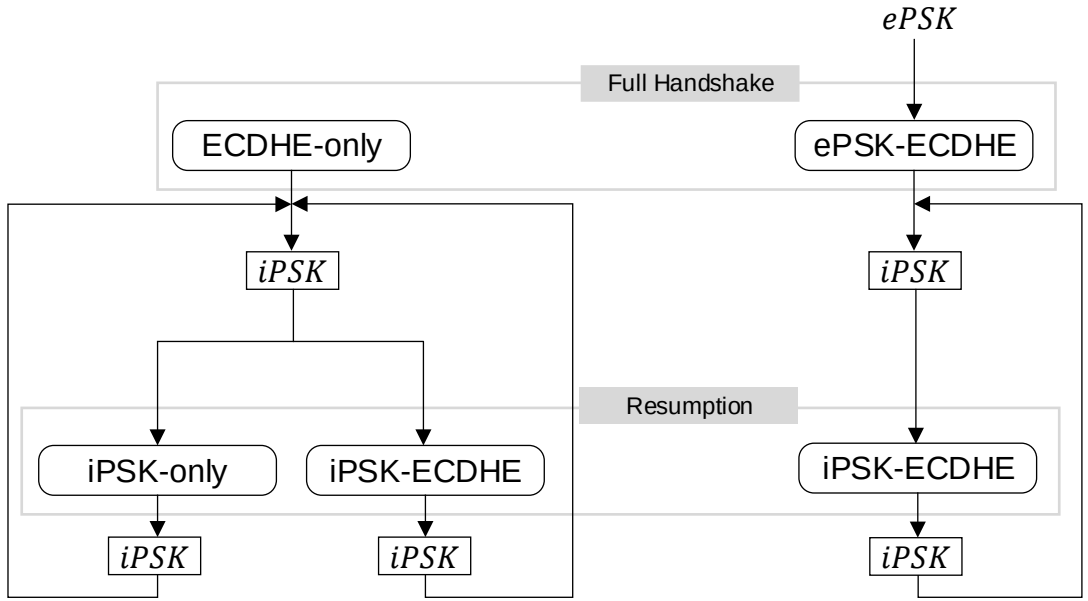
RTLS 1.3 – ePSK только с ECDHE

- Возможность осуществления ложной аутентификации при некоторых условиях с использованием ePSK

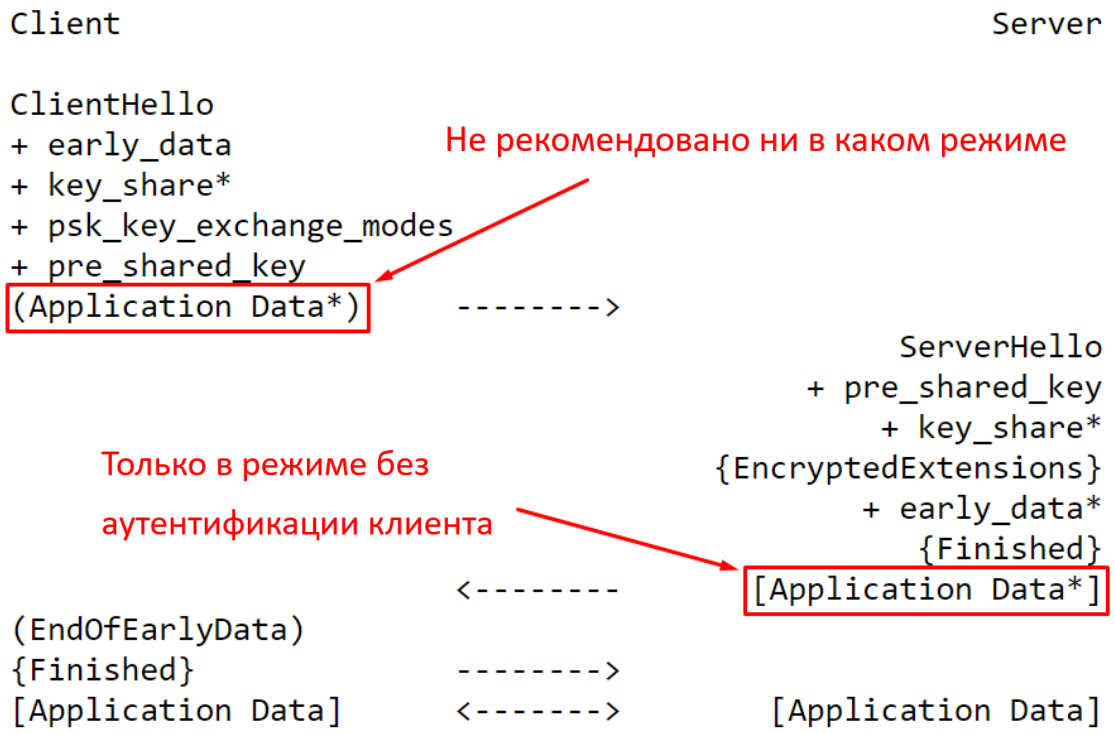


RTLS 1.3 – ePSK только с ECDHE

Особенности взаимосвязи режимов работы протокола Handshake в RTLS



RTLS 1.3 – прикладные данные до завершения Handshake



RTLS 1.3 – жизнь соединения после ECDHE

- Режимы с PSK – аналог Session Resumption старых версий TLS
- Вводимые ограничения:
  - Один iPSK можно использовать только в рамках одного соединения
  - Сервер не должен отправлять более 1024 тикетов в рамках одного соединения
  - Срок жизни iPSK не должен превышать минимума из следующих значений:
    - 604800 секунд (7 дней) в соответствии с RFC 8446;
    - количество секунд с момента формирования тикета до момента, определяющегося датой, превышающей дату окончания действия сертификата на 604800 секунд (+ 7 дней);
    - количество секунд, продиктованное другими локальными политиками сервера. Например, если к моменту создания тикета сертификат клиента, с помощью которого устанавливалось базовое соединение, отозван, то сервер может принять решение о запрете создания новых тикетов.

Модели для оценки стойкости подпротокола Record

- Эволюция моделей противника для конфиденциальности и целостности:
  - Конфиденциальность: LOR-CPA, ROR-CPA, IND-CPA, CPNA, SEM-CPA, …
  - Целостность: EU-CMA, UU-CMA, SU-CMA, IND-CMA, …
  - Совмещенные модели: DAE, IND-CCA, IND-CCA3, …
  - Модели для защищенного канала: IND-sfCCA3, IND-sfCCSA

Модель IND-sfCCSA для оценки стойкости подпротокола Record

$\underline{\mathbf{Exp}^{\text{IND-sfCCSA}-b}_{\text{sfAEAD}}(A)}$

$K \xleftarrow{\$} \text{sfAEAD.K}()$
$u \leftarrow 0,\ v \leftarrow 0$
$sent \leftarrow \emptyset$
$st \leftarrow A$
$(st_E,\ st_D) \leftarrow \text{sfAEAD.Init}(st)$
$b' \leftarrow A^{\text{Encrypt}^b, \text{Decrypt}^b}$
return b'

$\underline{\text{Oracle Encrypt}^b(ad, m)}$

if $b = 0$ **then**
  $m \xleftarrow{\mathcal{U}} \{0,1\}^{|m|}$
$c \leftarrow \text{sfAEAD.E}(K, ad, m, st_E)$
$sent \leftarrow sent \cup (ad, c, u)$
$st_E \leftarrow \text{sfAEAD.Upd}(st_E)$
$u \leftarrow u + 1$
return c

$\underline{\text{Oracle Decrypt}^1(ad, c)}$

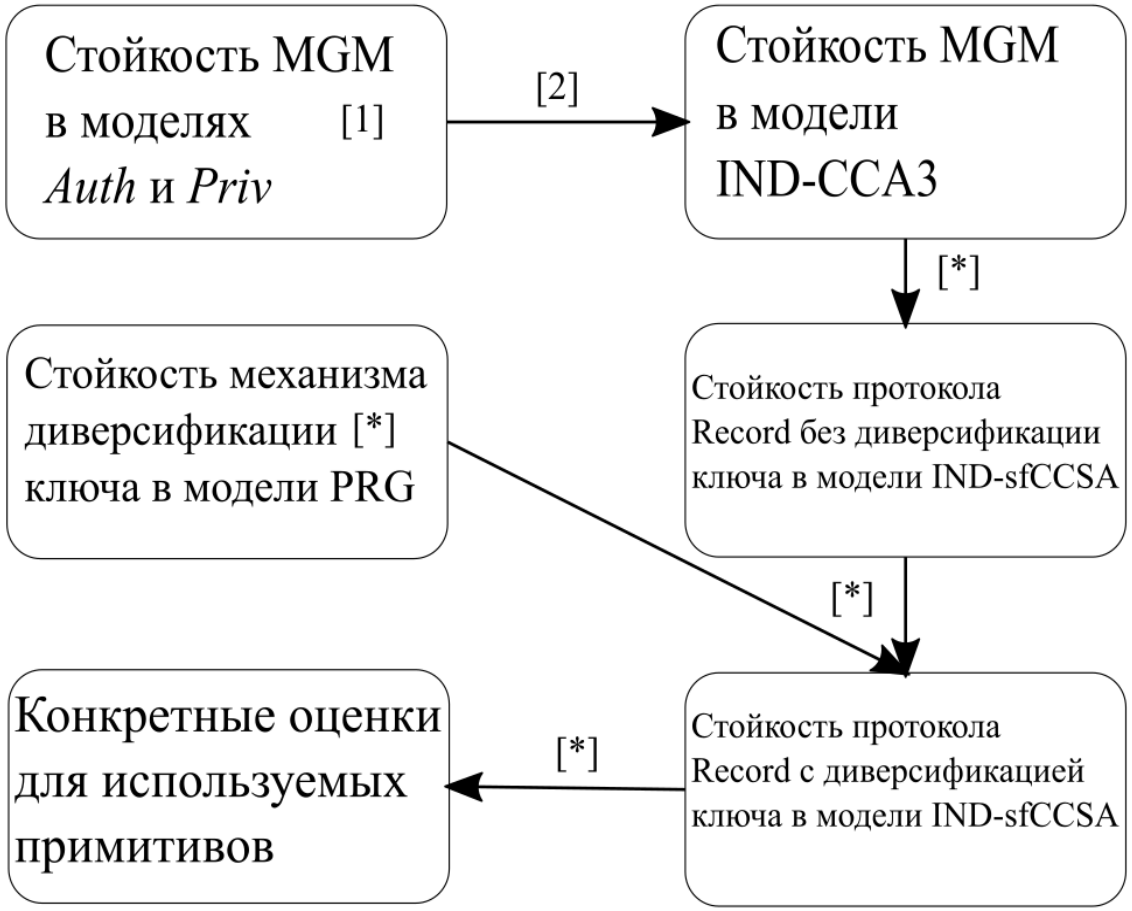
$m \leftarrow \text{sfAEAD.D}(K, ad, c, st_D)$
if $(m \neq \perp)$ **then**
  if $((ad, c, v) \in sent)$ **then**
    $m \leftarrow \perp$
  $st_D \leftarrow \text{sfAEAD.Upd}(st_D)$
  $v \leftarrow v + 1$
return m

$\underline{\text{Oracle Decrypt}^0(ad, c)}$

return $\perp$

$$\text{Adv}^{\text{IND-sfCCSA}}_{\text{sfAEAD}}(A) = \Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-1}_{\text{sfAEAD}}(A) \to 1\right] - \Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-0}_{\text{sfAEAD}}(A) \to 1\right]$$

Схема построения сведения



- [1] Akhmetzyanova L., Alekseev E., Karpunin G., Nozdrunov V. *«Security of Multilinear Galois Mode (MGM)»*, 2019.
- [2] Shrimpton T. *«A characterization of authenticated-encryption as a form of chosen ciphertext security»*, 2004.

- [*] Доказано в ходе оценки стойкости Record.

Доказанные теоремы

- Теорема о стойкости Record без диверсификации ключа:

$$\mathsf{Insec}^{\text{IND-sfCCSA}}_{\mathsf{Record}_{\mathsf{MGM}}}(t,q,q_D,l) \leqslant \mathsf{Insec}^{\text{IND-CCA3}}_{\mathsf{MGM}}(t,q,q_D,l) \leqslant \frac{3(ql+4q)^2}{2^n} + q_D \cdot \frac{3(ql+4q+l+3)^2+2}{2^n}$$

- Теорема о стойкости схемы с диверсификацией ключа:

$$\mathsf{Insec}^{\text{IND-sfCCSA}}_{(\mathsf{Record}_{\mathsf{MGM}},ExtKeyGen13)_h}(t,q,q_D,l) \leqslant 2 \cdot \mathsf{Insec}^{\text{PRG}}_{ExtKeyGen13}\left(t,\left\lceil\frac{q}{h}\right\rceil\right) + \left\lceil\frac{q}{h}\right\rceil \cdot \mathsf{Insec}^{\text{IND-sfCCSA}}_{\mathsf{Record}_{\mathsf{MGM}}}(t,h,q_D,l)$$

Оценка стойкости в модели IND-sfCCSA

Блочный шифр Кузнечик. Для данного алгоритма $l = 2^{10} + 1, n = 128$. Тогда можно получить следующую оценку:

$$InSec^{\text{IND-sfCCSA}}_{(\text{Record}_{\text{MGM}}, ExtKeyGen13)_h}(t, q, 1, l) \leqslant \text{Insec}^{\text{PRG}}_{ExtKeyGen13}\left(t, \left\lceil \frac{q}{h} \right\rceil\right) + \\ + \left\lceil \frac{q}{h} \right\rceil \cdot \frac{3176523h^2 + 3173436h + 1585177}{170141183460469231731687303715884105728} \leqslant \\ \leqslant qh \cdot 2^{-105} \quad (1)$$

Блочный шифр Магма. Для данного алгоритма $l = 2^{11} + 1, n = 64$. Тогда можно получить следующую оценку:

$$InSec^{\text{IND-sfCCSA}}_{(\text{Record}_{\text{MGM}}, ExtKeyGen13)_h}(t, q, 1, l) \leqslant \text{Insec}^{\text{PRG}}_{ExtKeyGen13}\left(t, \left\lceil \frac{q}{h} \right\rceil\right) + \\ + \left\lceil \frac{q}{h} \right\rceil \cdot \frac{12644427h^2 + 12638268h + 6316057}{92233720368547758 08} \leqslant \\ \leqslant qh \cdot 2^{-39} \quad (2)$$

Спасибо за внимание! Вопросы?

Контактная информация

Авторы доклада:

Алексеев Евгений

Ахметзянова Лилия

Седов Григорий

Смышляева Екатерина

Электронная почта:

alekseev@cryptopro.ru

lah@cryptopro.ru

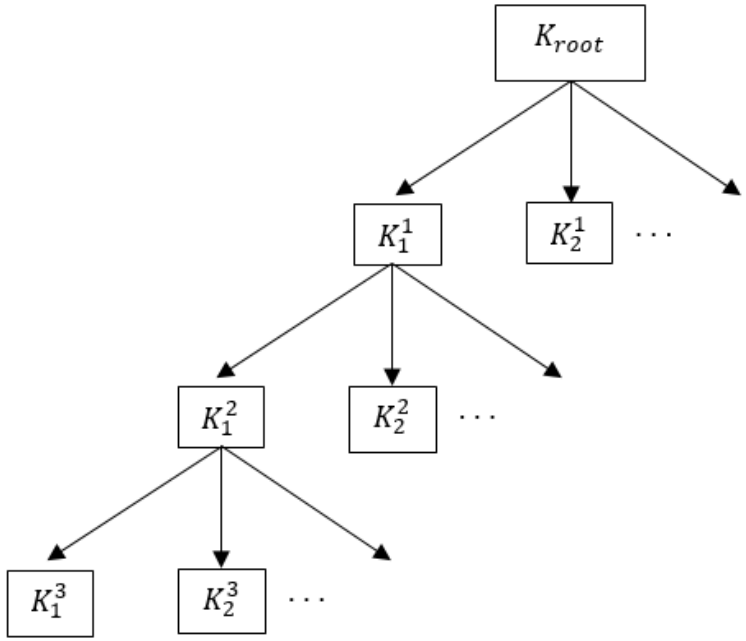
sedovgk@cryptopro.ru

ess@cryptopro.ru

Сайт:

www.cryptopro.ru

Ключевое дерево в Record



$$TLSTREE(K_{root}, i) = Divers_3(Divers_2(Divers_1(K_{root}, i \,\&\, C_1), i \,\&\, C_2), i \,\&\, C_3)$$