Протокол SSL/TLS

SSL/TLS

Расшифровка аббревиатур:

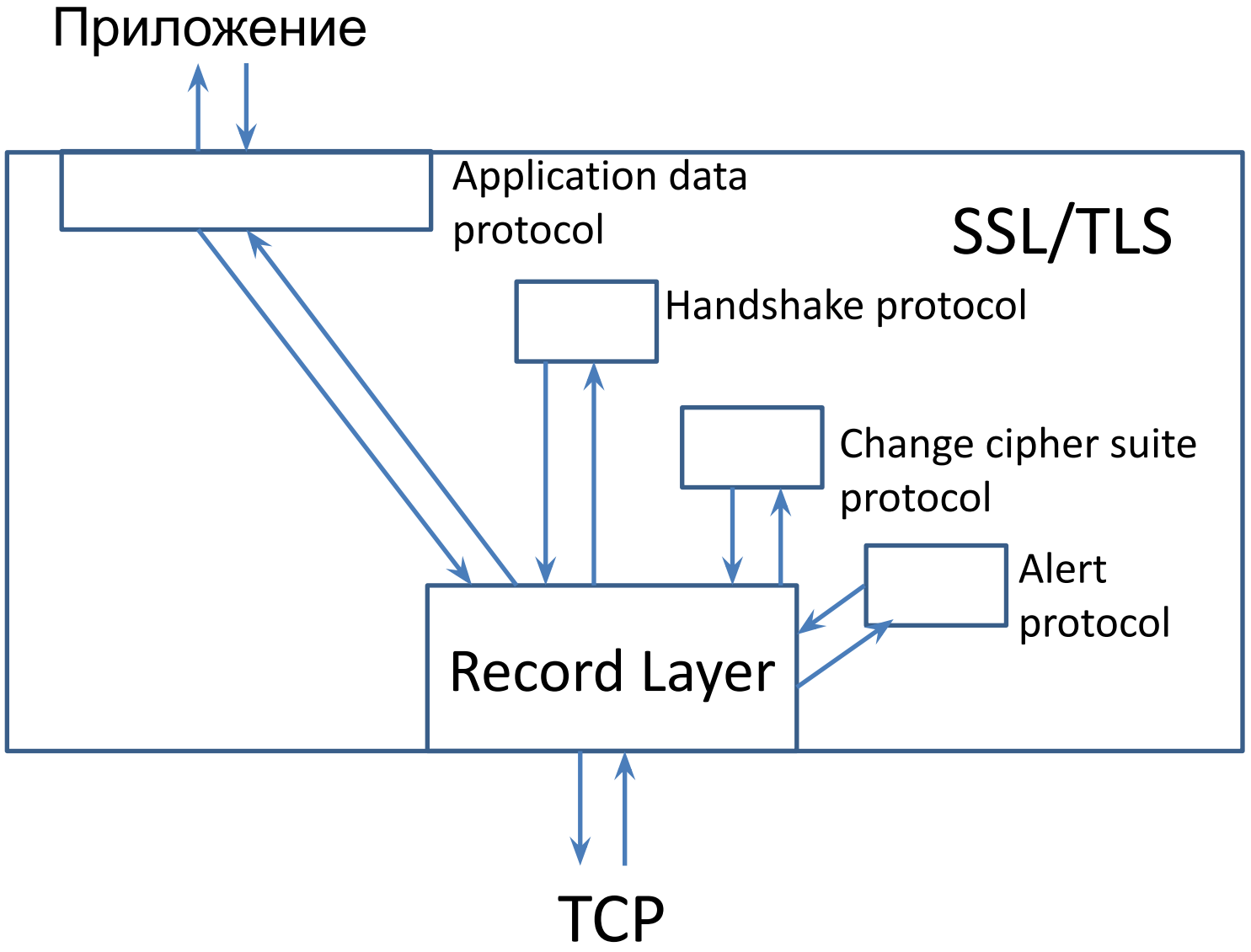
- Secure Socket Layer
- Transport Layer Secure

Версии протокола

Протокол	Документы	Цифровой идентификатор версии
SSL 3.0	Внутренние спецификации Netscape Corp. Internet-draft (черновик RFC) Informational RFC 6101 (2011)	0x0300
TLS 1.0	RFC 2246 (1999)	0x0301
TLS 1.1	RFC 4346 (2002)	0x0302
TLS 1.2	RFC 5246 (2006)	0x0303
TLS 1.3	RFC 8446 (2018)	0x0303

Положение в стеке протоколов

Архитектура протокола



Основные характеристики протокола

- Аутентификация сторон на основе сертификатов X.509
  - ЭЦП RSA, DSA
  - По умолчанию сервер аутентифицируется, а клиент – нет
- Создание общего секрета (RSA, DH, DHE (Ephemeral – одноразовые ключи)
- Шифрование симметричными алгоритмами, режим CBC, AEAD (TLS 1.2, 1.3)
- Контроль подлинности и целостности данных HMAC

Основные характеристики протокола - алгоритмы

- RSA, DH, DSS (DSA), RC2, RC4, IDEA, DES, 3DES, MD5, SHA-1
- Kerberos (опц.) – с TLS 1.1
- AES, SHA-2 (256/384/512), отмена RC2, DES, IDEA – с TLS 1.2
- TLS 1.3: отмена почти всех перечисленных алгоритмов и введение новых

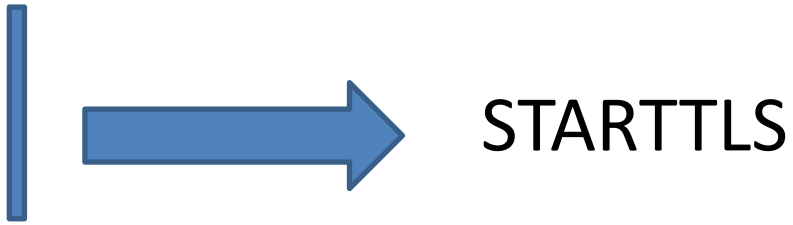
Основные характеристики протокола – алгоритмы в TLS 1.3

- Общий секрет: DH, ECDH
- ЭЦП: RSA, ECDSA, EDDSA (еще один вариант алгоритма на эллиптических кривых)
- Шифрование данных: AES, CHACHA20
  - AEAD вместо CBC
- Хэш: SHA-2 (256/384/512)

Основные характеристики протокола - транспорт

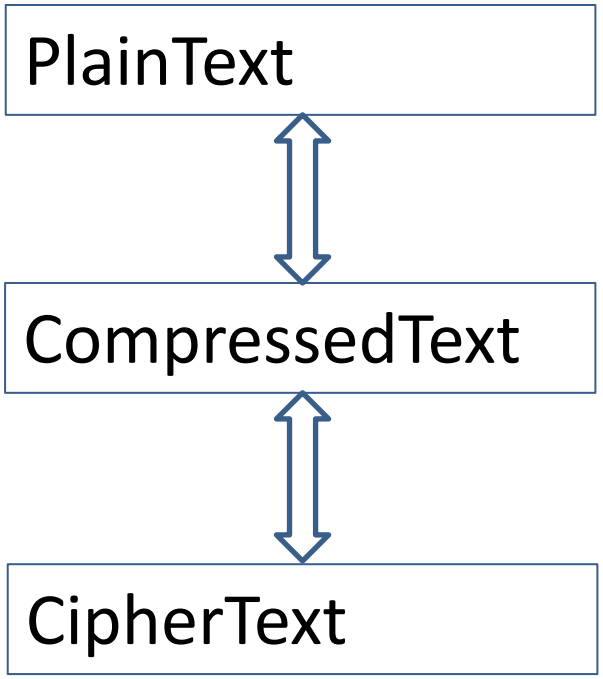
- Работает поверх TCP
- Дополнительные защищенные TCP-соединения (одновременно или после основного) в рамках одного сеанса протокола
- Порты:

  https: 443
  почтовые протоколы:
  SMTPS: 465
  IMAP4S: 993
  POP3S: 995



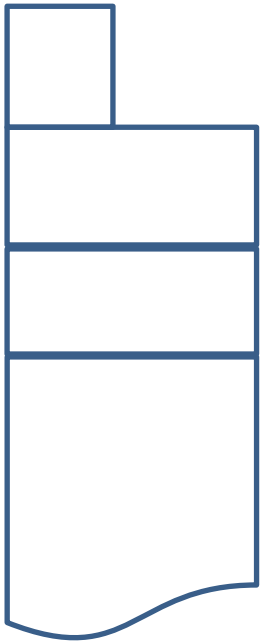
- Защита произвольных соединений (stunnel, s_client, s_server)

Record Layer



Record Layer

PlainText
CompressedText



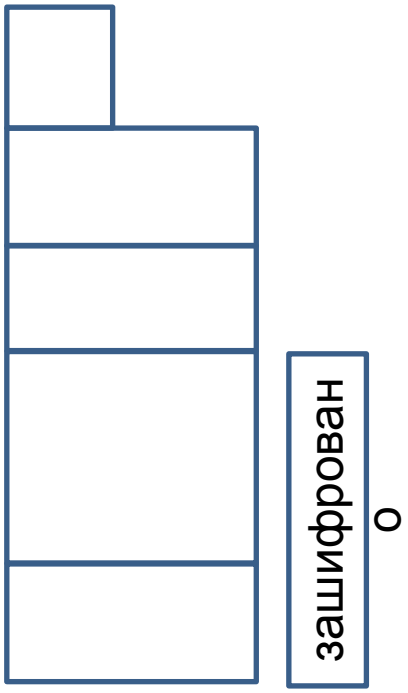
ContentType
ProtocolVersion
Length
Data

```
enum {
 change_cipher_spec(20),
  alert(21),
  handshake(22),
  application_data(23),
  (255)
} ContentType;
```

Record Layer

CipherText

Потоковые

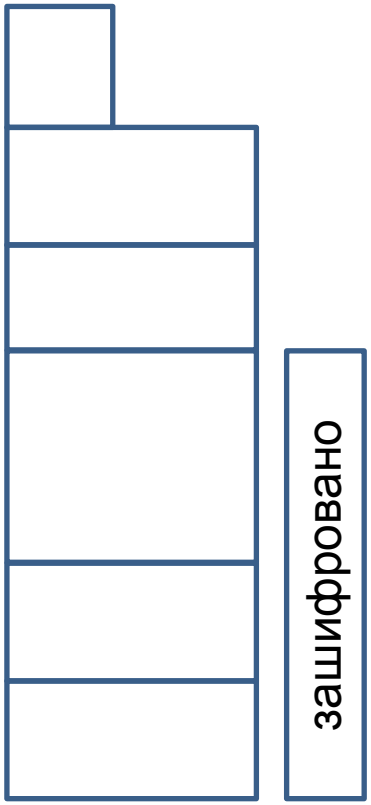


ContentType
ProtocolVersion
Length
Data

MAC

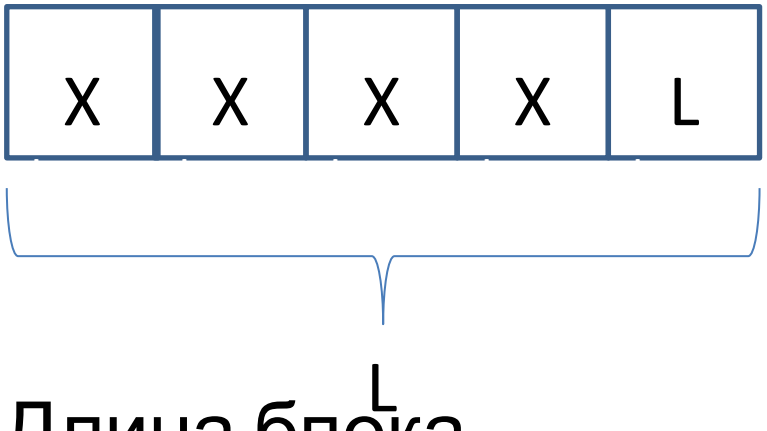
PAD

Блочные



Record Layer

Pad (SSL 3.0)



1 <= L <= Длина блока

X определяется реализацией,

обычно

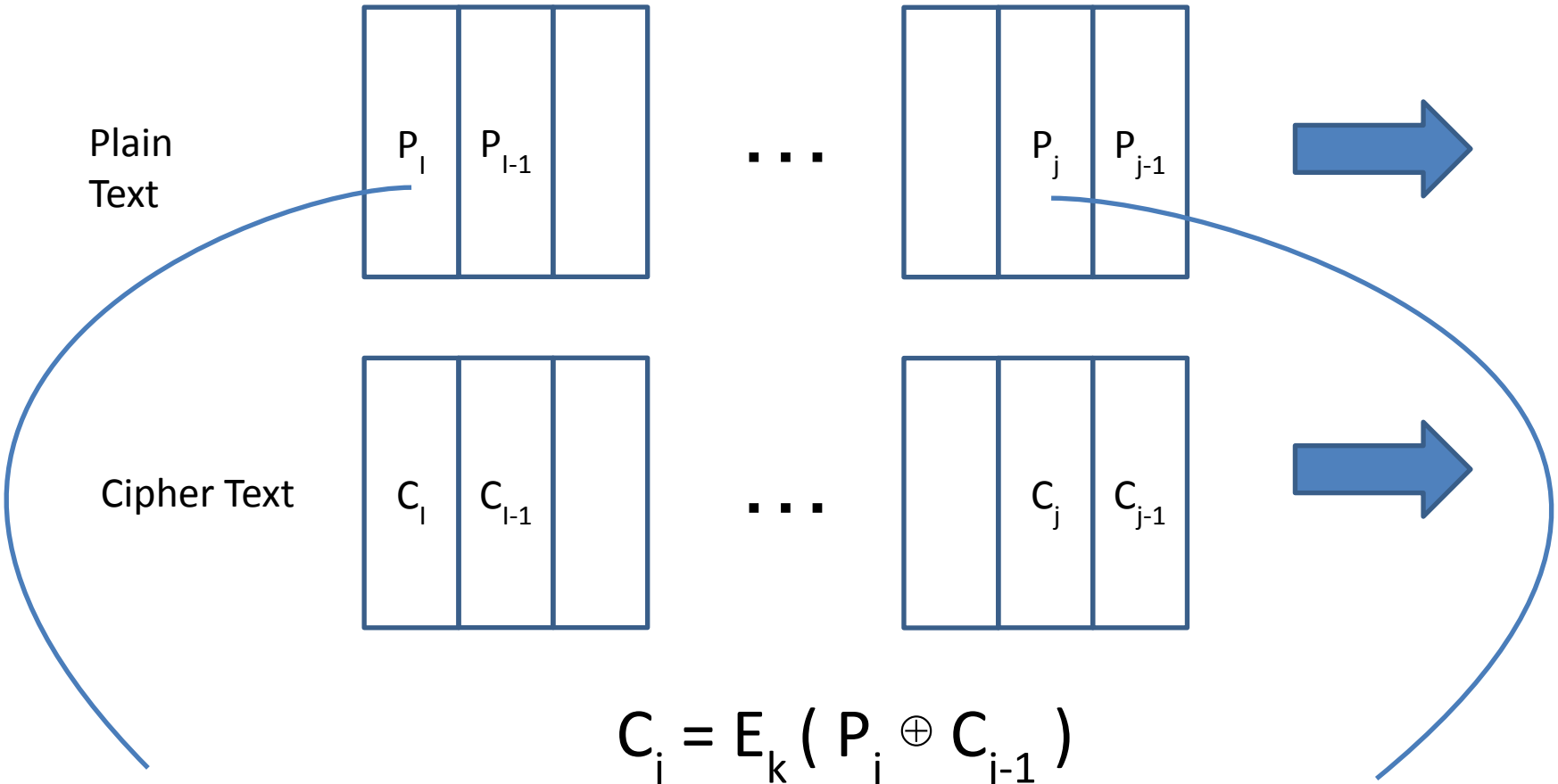
X = 0

Атаки на SSL/TLS – первые ласточки

Wei Dai.
An attack against SSH2 protocol.
2002.
(«Chosen Plaintext»)

Serge Vaudenay.
Security Flaws Induced by CBC Padding Application to SSL, IPSEC, WTLS. **2002**.
(«Padding Oracle»)

«Chosen Plaintext» by Wei Dai



$$C_j = E_k (P_j \oplus C_{j-1})$$

атакующий может задать

расшифровываем

«Chosen Plaintext» by Wei Dai (2)

Проверяем гипотезу $P_j = X$

Задаем $P_i = X \oplus C_{i-1} \oplus C_{j-1}$

Тогда $C_i = E_k (P_i \oplus C_{i-1}) = E_k (X \oplus C_{i-1} \oplus C_{j-1} \oplus C_{i-1}) = E_k (X \oplus C_{j-1})$

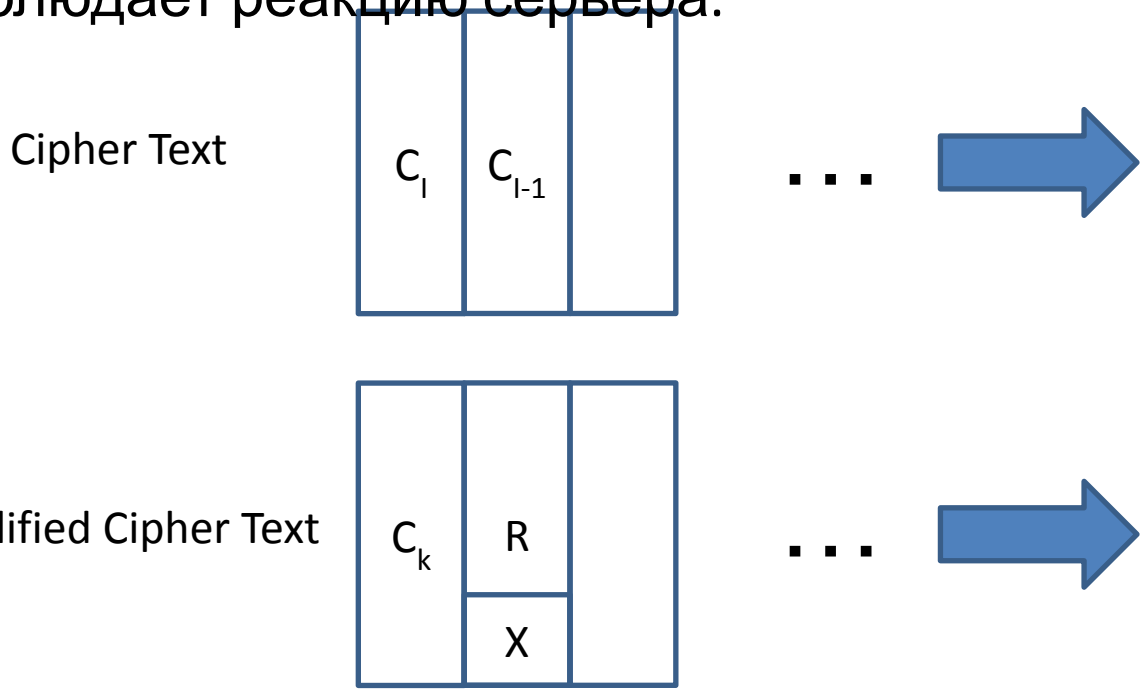
Если $P_j = X$, то $C_i = C_j$

Перебрать все X, пока не будет достигнуто равенство $C_i = C_j$.

Требует 2^N попыток, где N – длина блока (min N = 64).

«Padding Oracle» by Serge Vaudenay

Злоумышленник перехватывает сообщение, затем многократно модифицирует его, отправляет серверу и наблюдает реакцию сервера.



C_k – блок, который надо расшифровать
R – случайный блок
X – подбираемое значение

«Padding Oracle» by Serge Vaudenay

- Если в результате получился некорректный PAD, сервер отвечает сообщением протокола Alert decryption_failed (21)
- Если получился корректный PAD, сервер отвечает сообщением протокола Alert bad_record_mac (20)
- В этом случае (наиболее вероятно) что последний байт PAD равен 1, и тогда
  P_k[последний байт] = 1 $\oplus$ x $\oplus$ C_{k-1}[последний байт]
- Для определения последнего байта P_k требуется 256 попыток.
- Зная последний байт, по аналогичной схеме определяется предпоследний, и далее все байты блока. Затем расшифровывается следующий блок.
- Для расшифровки всего шифротекста требуется попыток: 256 x (длина шифротекста)

Атаки на SSL/TLS – ускорение

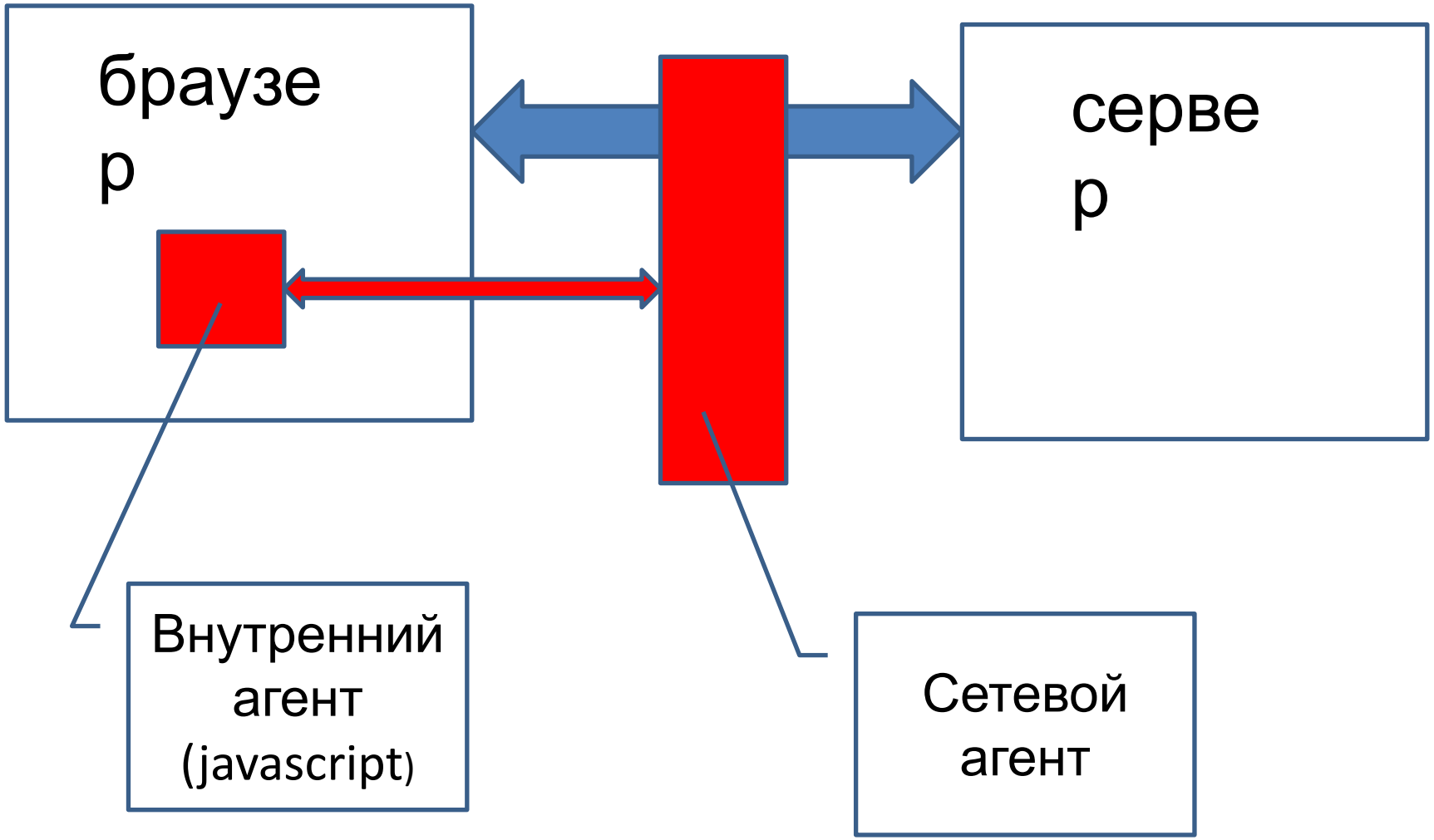
Wei Dai.
An attack against SSH2 protocol.
2002.
(«Chosen Plaintext»)

Duong, Rizzo
Атака BEAST
2011

Serge Vaudenay.
Security Flaws Induced by CBC Padding Application to SSL, IPSEC, WTLS. **2002**.
(«Padding Oracle»)

Duong, M**öller, Kotiwiz, et all.**
Атака POODLE
2014

BEAST & POODLE



BEAST & POODLE

Структура HTTP-сообщения:

POST <URL> HTTP/1.1

Headers: …

Cookies: …

<body>

Злоумышленник управляет составом <URL> и <body> и охотится за Headers и Cookie. Он может установить дешифруемый байт на любую удобную позицию, а также управлять длиной

BEAST & POODLE

Сложность вскрытия:

- BEAST: 256 попыток на каждый байт сообщения - против 2^N на каждый N-битный блок в «Chosen Plaintext» by Wei Dai
- POODLE: 256 попыток на каждый байт сообщения - так же, как и в «Padding Oracle» by Serge Vaudenay, но технически реализация значительно проще.
- На практике вскрытие Headers и Cookie занимает 5 минут.

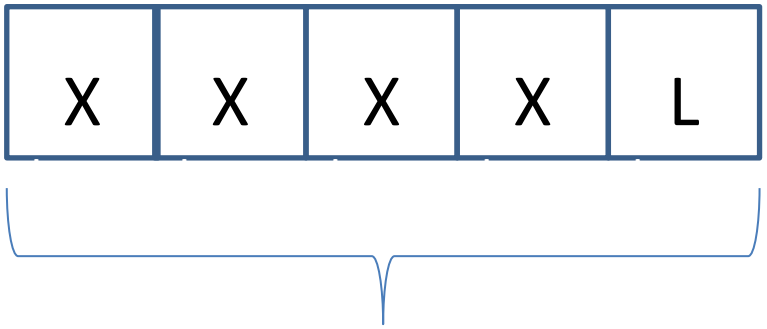
POODLE

Состав атаки

1. Атака против Handshake protocol, заставляющая стороны переключиться на SSL 3.0
2. Собственно алгоритм Padding Oracle с оптимизацией за счет особенностей HTTP.

Защита от Padding Oracle

Pad (TLS 1.0)



1 <= L <= **255**

X = L

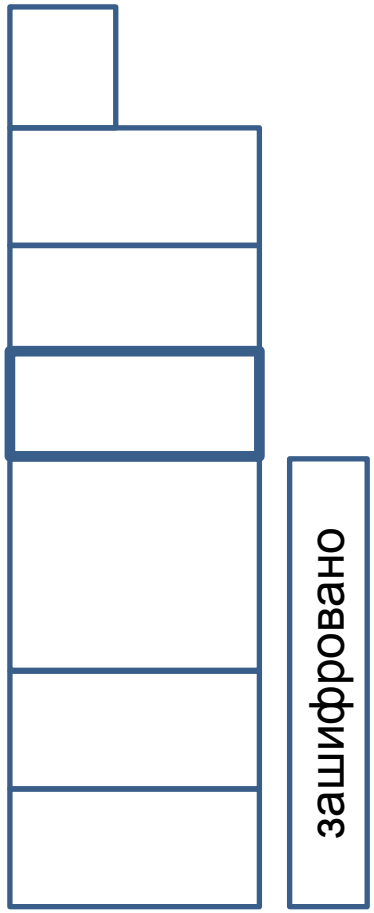
Никогда не возникает ошибка decryption_failed (21), сервер больше не работает как Padding Oracle

Защита от «Chosen Plaintext», BEAST

CipherText (TLS 1.1)

Блочные

ContentType
ProtocolVersion
Length
IV
Data
MAC
PAD



Атаки на протокол Handshake

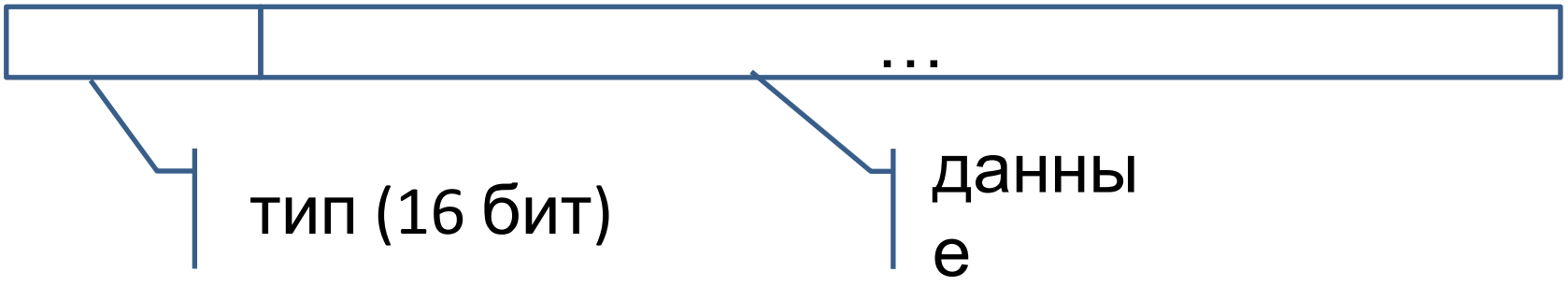
- POODLE (часть 1) – переключение на SSL 3.0
  Защита: дополнение протоколов TLS 1.0-1.3, RFC 7507
- M.Ray, S.Dispensa. The SSL authentication gap. 2009. вставка произвольного текста в начало SSL-сеанса. «Renegotiation» (Сервер запрашивает сертификат клиента.) Защита: дополнение протоколов TLS 1.0-1.2, RFC RFC 5746. Защита в TLS 1.3???

Hello Extensions

Extensions – дополнительные, необязательные поля, которые могут быть добавлены к сообщениям протокола Handshake ClientHello и ServerHello

- Предложены в RFC 3546 (2003)
- Вошли как опция в спецификацию TLS 1.2
- Стали основным механизмом согласования параметров сеанса в TLS 1.3

Hello Extensions

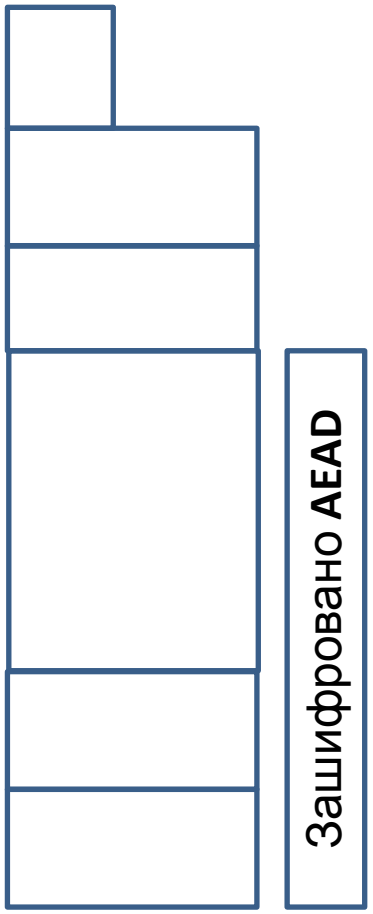


Примеры:

- server_name (для поддержки вирт. хостов)
- max_fragment_length
- client_certificate_url
- status_request (OCSP перепоручается серверу)
- …

Record Layer в TLS 1.3

CipherText (TLS 1.3)



ContentType = 23 (app.data)

ProtocolVersion = 0x0303 (TLS 1.2)

Length

Data

ContentType (реальный)

PAD

Record Layer в TLS 1.3 – шифрование AEAD

AEAD –
Authenticated Encryption with Associated Data

- Режим применения взамен CBC в TLS 1.2 и более ранних
- Обеспечивает одновременно
  - Шифрование
  - Контроль подлинности и целостности.
- Специфицирован в RFC 5116 (2008)

Record Layer в TLS 1.3 – шифрование AEAD (2)

Режим применения AEAD

- Разрешен как опция в TLS 1.2
- Стал единственным режимом в TLS 1.3

Режим применения AEAD определен для алгоритмов шифрования:

- AES - RFC 5116 (2008), RFC 6655 (2012)
- CHACHA20 – RFC 8439 (2018)