Защита информации

Павел Юдаев

МГТУ им. Баумана, Кафедра ИУ-9

Москва, 2014

Раздел 9 - Заверенное шифрование

Шифрование и обеспечение целостности

authenticated encryption

Уже умеем:

- Семантическая стойкость против атаки с выбором открытого текста - тайна сообщения от того, кто подслушивает. Но может менять шифротекст.

- Обеспечение целостности открытого текста против атаки с выбранным (открытым) текстом

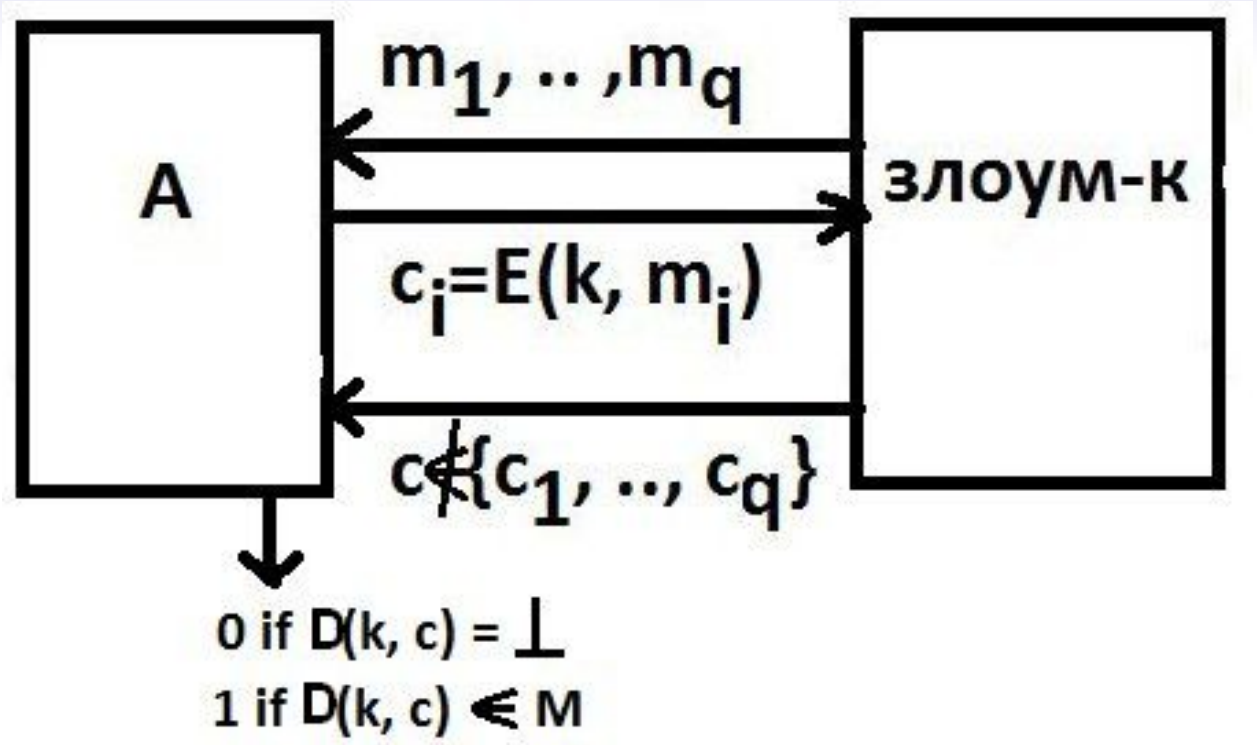
Цель: одновременно конфиденциальность и целостность.

Зачем это нужно?

- я отправил в банк платежное поручение (фиксированный формат), чтобы в нем не поменяли получателя или сумму.

- в протоколе IPSec (рас-)шифрование происходит на сервере, на сетевом (третьем) уровне модели OSI для всех его пользователей, и далее по номеру порта каждый получает свои сообщения. Изменим шифротекст так, чтобы изменился номер порта - получим чужое сообщение.

Пусть шифр (E,D) - это набор из двух алгоритмов
$E : K \times M(\times N) \to C$ и
$D : K \times C(\times N) \to M \bigcup \perp$
N - множество nonce, опционально.
$\perp \notin M$ - отказ от приема данного шифротекста.



Опр.

шифр (E,D) *обеспечивает целостность шифротекста* (ciphertext integrity), если
$Adv_{CI}[A, E, D] =$
$P(\text{злоум-к созд. нов. ш/текст, кот. примет система}) < \varepsilon(n)$

Опр.

Шифр (E,D) - *шифр с заверенным шифротекстом* (authenticated encryption, AE), если он
1) семантически стойкий к атакам с выбором открытого текста
2) обеспечивает целостность шифротекста

Пример

Ни один из рассмотренных ранее шифров не является шифром с заверенным шифротекстом. Потому что расшифрование всегда дает какое-то сообщение.

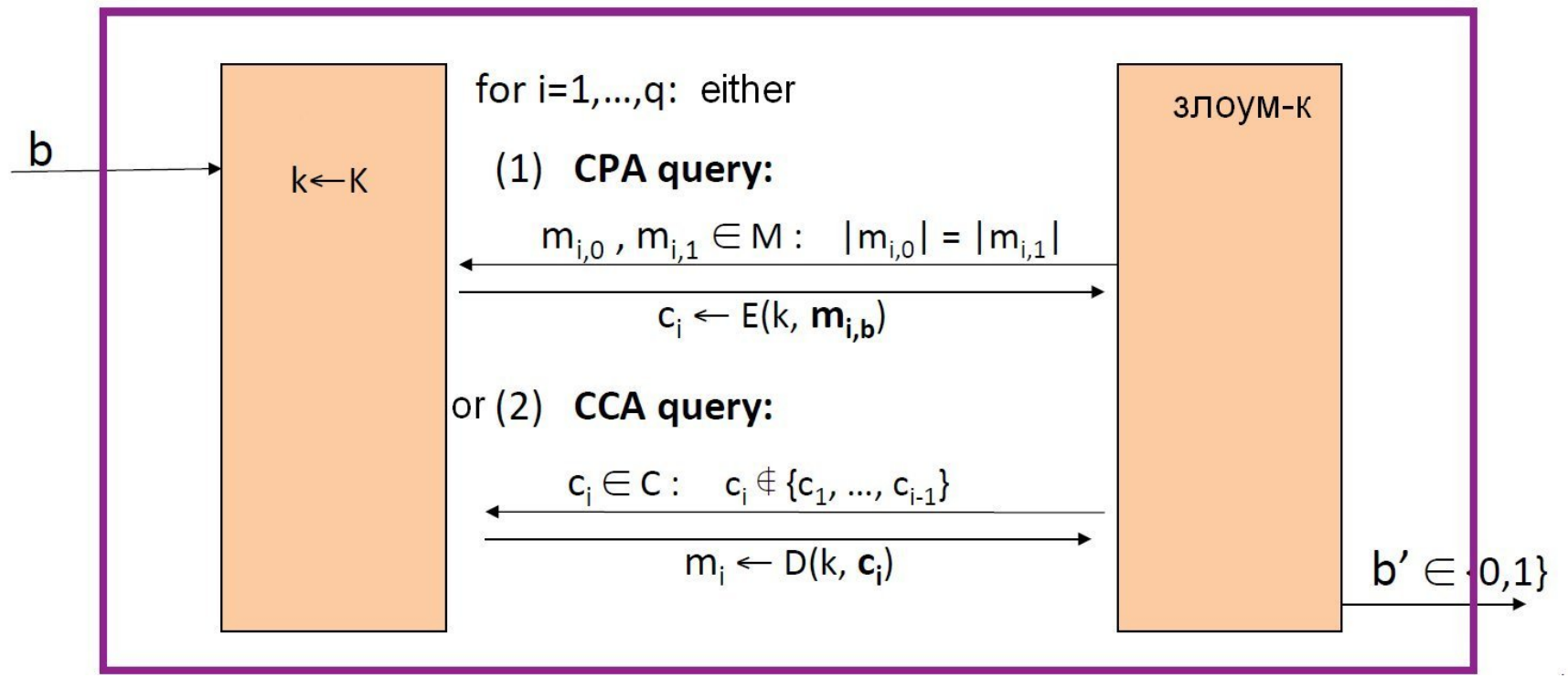
Стойкость заверенного шифрования к атаке с выбранным шифротекстом.

Модель атаки:

- злоум-к может получать шифротексты для любых сообщений (выбор открытого текста)

- злоум-к может пытаться получать по шифротекстам их открытый текст (с ограничениями, см. далее)

Цель: нарушить семантическую стойкость шифра.



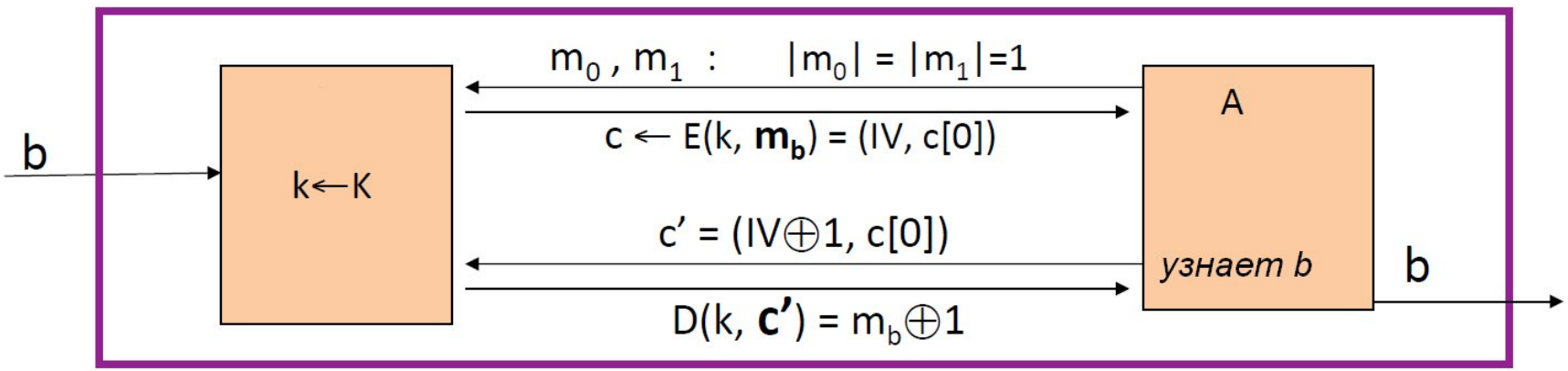
1) система случайно выбирает $b = 0$ или 1 и случайный ключ k. Значение b фиксировано. Ключ многоразовый, он фиксирован.
2) Злоум-к делает по очереди q запросов любого вида.
3) Злоум-к пытается угадать значение b.

Опр.

(E,D) *стойкий к атаке с выбранным шифротекстом*, если
$Adv_{CCA}[A, E, D] = |P(b' = 1|b = 1) - P(b' = 1|b = 0)| < \varepsilon(n)$

Пример

Режим CBC со случайным IV не стойкий к этой атаке.



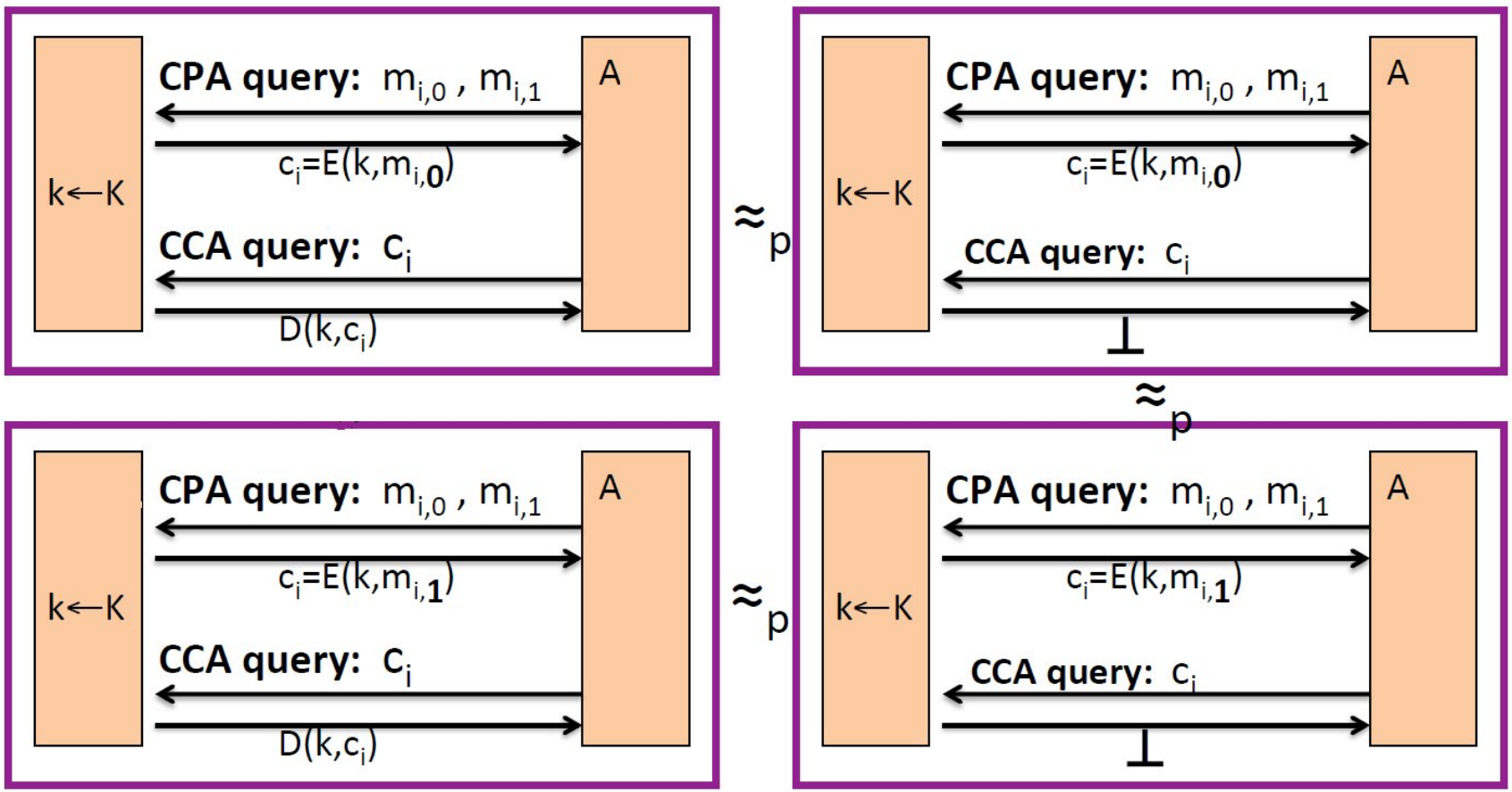
Утверждение

Пусть (E,D) - шифр с заверенным шифротекстом.
Тогда (E,D) стойкий к атаке с выбранным шифротекстом.
Причем $\forall$ алгоритма (злоум-ка) $A \in \mathrm{PPT}$ сущ. алгоритмы $B_1,\ B_2 \in \mathrm{PPT}$:

$$Adv_{CCA}[A, E, D] \leq 2q \cdot Adv_{CI}[B_1, E, D] + Adv_{CPA}[B_2, E, D]$$

Док-во

схема в картинках: исп. целостность шифротекста, стойкость к атаке с выбранным открытым текстом, снова целостоность шифротекста.



Вывод:
Заверенное симметричное шифрование (authenticated symmetric encryption) гарантирует:

- защиту от злоум-ка, который слушает канал передачи данных и может менять передаваемые сообщения. Принимающая сторона уверена, что автор сообщения знал ключ k.

- стойкость к атаке с выбранным шифротекстом

Не гарантирует:

- стойкость к атаке воспроизведения. Поэтому многие протоколы исп. метки времени и/или требуют точное системное время, напр. сеть i2p. Или счетчик сообщений от начала сессии.

- стойкость к атакам по вторичным каналам (напр., по врем. выполнения)

- можно сфабриковать сообщение от контрагента, т.к. у них общий секретный ключ.

Раздел 9 - Заверенное шифрование

Сочетание шифрования и MAC.

k_E, k_I - ключи для шифрования и MAC.

1) MtE, MAC-then-Encrypt.
$m \to m||S(k_I, m) \to E(k_E, S(k_I, m)||m)$.
Пример: TLS/SSL

2) EtM, Encrypt-then-MAC.
$m \to E(k_E, m) \to E(k_E, m)||S(k_I, E(k_E, m))$.
Пример: IPSec

3) EaM, Encrypt-and-MAC. $m \to E(k_E, m) \to E(k_E, m)||S(k_I, m)$.
Пример: SSH

Теорема 1

Если шифр обесп. стойкость к атаке с открытым текстом, а MAC - целостность шифротекста, EtM всегда дает авторизованное шифрование, независимо от реализаций шифра и MAC.

Остальные два варианта могут быть уязвимы к атакам с выбором шифротекста.

EaM: x/ф может раскрывать частичную инф. о тексте, напр. $H(m)[0] = f(m)$

Детали реализации протоколов SSH (EaM) и TLS (MtE)
$\Rightarrow$ заверенное шифрование.

SSH (с уязвимостями!) появилась в 1995 году. До этого - rlogin и т.д.

Стандартные режимы заверенного шифрования:

GCM: шифрование в CTR режиме и к нему GMAC.

CCM: CBC-MAC, затем шифрование в CTR режиме: MtE. (WiFi 802.11i) Доказано: заверенное шифрование.

EAX: шифрование в CTR режиме и к нему CMAC.

Реализация: библиотека OpenSSL.

Режим AEAD (Authenticated Encryption with Associated Data):

Шифроваться может часть сообщения, но MAC - всегда ко всему! Открытые данные - сетевые заголовки.

Производительность (AMD Opteron, 2.2 GHz, Linux, Crypto++ 5.6) (*)

алгоритм	размер исходного кода	скорость, МБ/с
AES/GCM	большой (Intel - одна инстр-я)	108
AES/CCM	малый	61
AES/EAX	малый	61
-	-	-
AES-CTR		139
AES-CBC		109
-	-	-
HMAC(SHA1)		147

Раздел 9 - Заверенное шифрование

Атаки на протокол TLS ранних версий

TLS:

1) согласование симметричных ключей (рассм. позже). Рез-т: A,B оба имеют ключи k_{cs}, k_{sc}

2) работа с симм. шифром.

Каждая сторона хранит состояние - два 64-битных счетчика ctr_{cs}, ctr_{sc} Счетчики сбрас. в 0 по инициализациии сессии, ++ при каждом сообщении.

Рез-т: защита от атаки воспроизведения.

Шифрование (E): MtE.
Напр., HMAC-SHA1, CBC-AES-128.

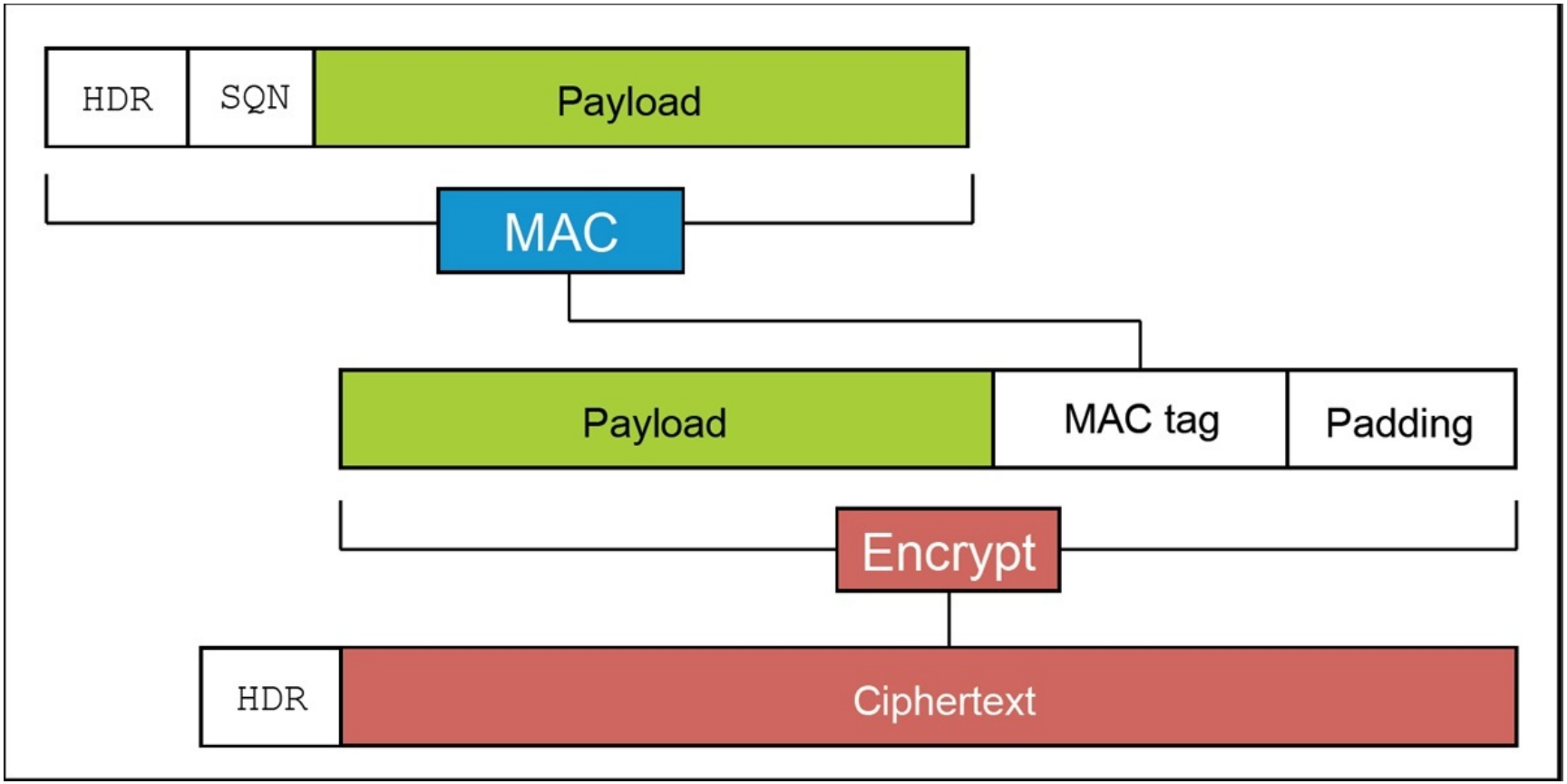
$k_{cs} = (k_I, k_E)$

header: тип, версия протокола; длина сообщения (без заголовка). ctr не посылается.

1) $payload := compress(data)$
2) $tag = S(k_I, (++ ctr_{cs} || header || payload))$
3) $pad =$ дополнение до длины блока $(payload || tag)$
4) $E(IV, k_E, payload || tag || pad)$ с новым случайным IV.
5) приписать в начало сообщения открытый header

Сообщение $= header || E(payload || tag || pad)$

(см. спецификацию в RFC-5246,
http://tools.ietf.org/html/rfc5246#section-6)


HDR
SQN
Payload
MAC
Payload
MAC tag
Padding
Encrypt
HDR
Ciphertext

Расшифрование (D):

1) Расшифровать с k_{cs}.

2) Проверить *pad*. Если не правильный, ответить bad_record_mac

3) Проверить *tag* для $(++ctr_{cs}||header||payload)$. Если не правильный, ответить bad_record_mac
Увеличить счетчик входящих сообщений.

4) Расшифровать, разархивировать *payload*.

(E,D) обесп. заверенное шифрование для *payload*.

BEAST: атака на CBC детерминированным IV

TLS 1.0 (1999): IV = последний блок предыдущего сообщения.

BEAST: javascript отправляет нужные запросы от клиента на нужный адрес. Запросы содержат шифрованный cookie автооризации. Злоум-к слушает канал. Атака с выбранным открытым текстом.

Позволяет узнать сообщение. В частности, узнать (украсть) cookie авторизации.

Доп. лит-ра (*): T.Duong, J.Rizzo, “Here Come The $\oplus$ Ninjas”, `https://bug665814.bugzilla.mozilla.org/attachment.cgi?id=540839`

Раздел 9 - Заверенное шифрование

CBC padding oracle attack

Расшифрование в TLS 1.0:

1) Расшифровать с k_{cs}.

2) Проверить pad. Если не правильный, ответить decryption_failed.

3) Проверить tag для $(++ctr_{cs}||header||payload)$. Если не правильный, ответить bad_record_mac

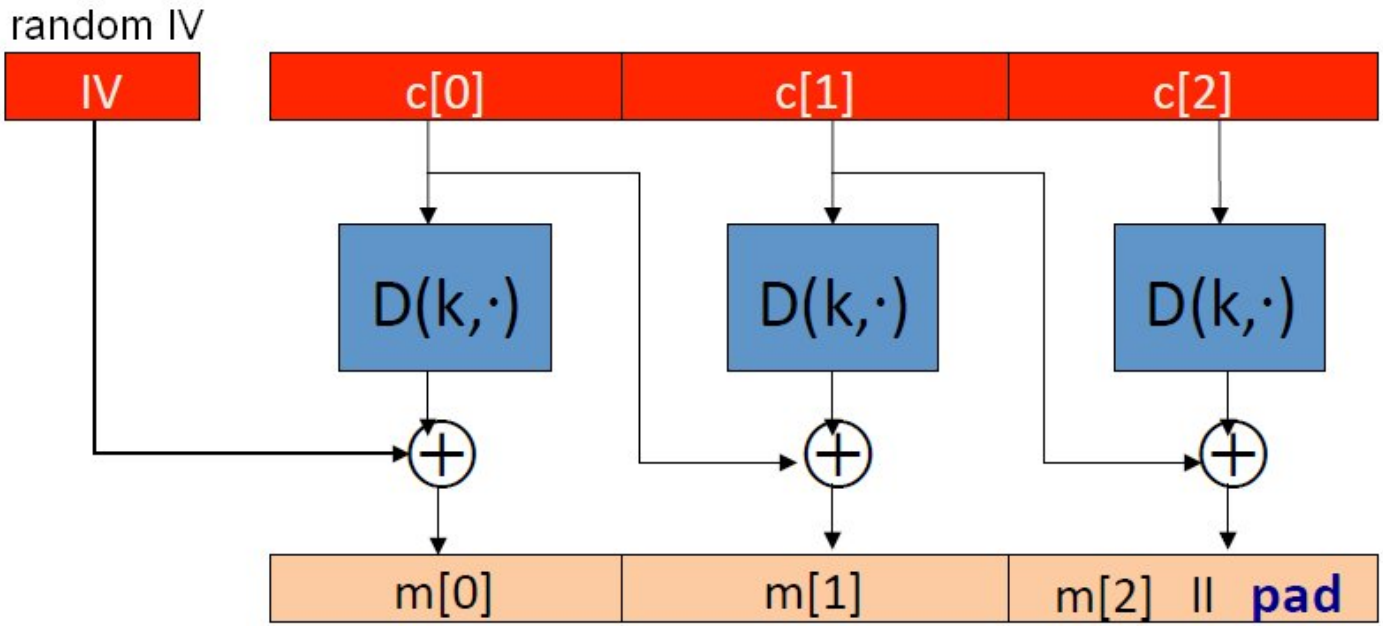
Упс, утекает информация об открытом тексте:
Злоум-к посылает шифротекст и узнает, последние байты открытого текста - правильное дополнение или нет.

Также по времени отклика:
а) bad_pad 21ms,
б) good_pad, bad_mac 23 ms

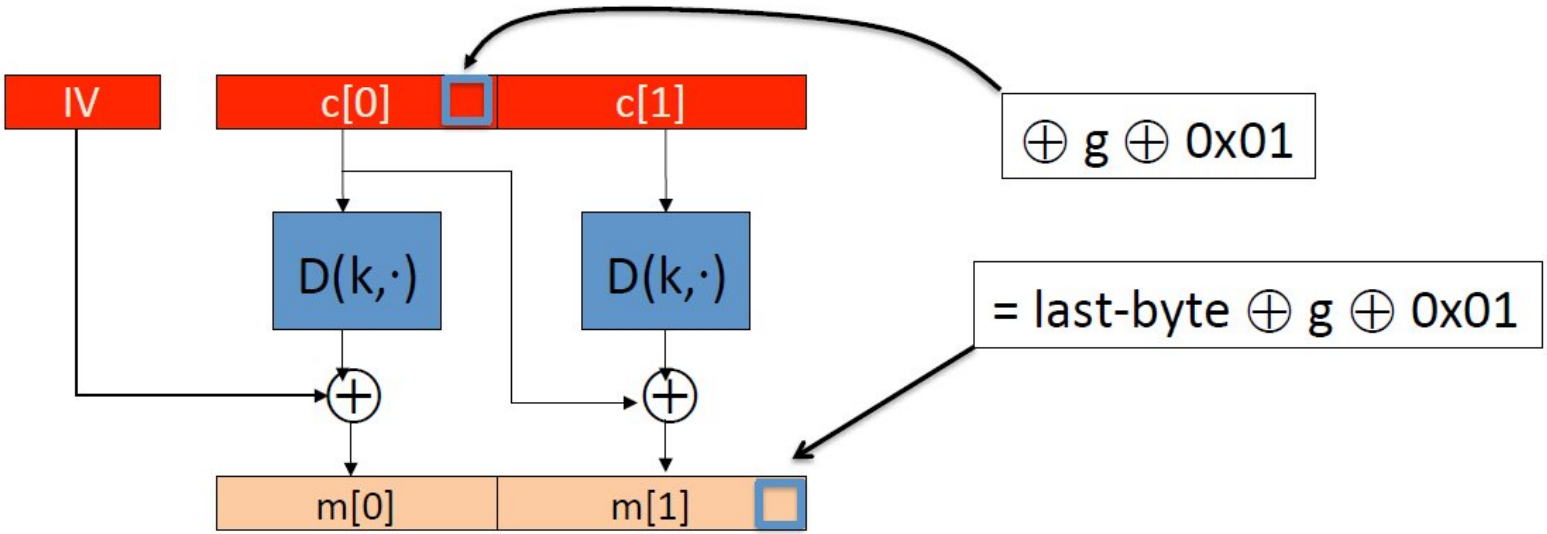
TLS 1.2 пересогласовывает ключ после ошибки, атака не пройдет.

Padding oracle attack на практике

Пусть злоум-к подслушал шифротекст c = (c[0] c[1] c[2]) и хочет узнать m[1].



Предположим, что последний байт m[1] равен g. Тогда пошлем такой шифротекст (IV, c'[0], c[1]):



Если последний байт $m[1]$ равен g, pad верный. Иначе - неверный.

Проделаем это для $g = 0, 1, ..., 255$ - узнаем последний байт m[1].

Потом - для второго с конца байта: последний байт равен 0x02, предпоследний $\oplus(g \oplus 0x02)$. И т.д.

Вывод:

MAC-then-CBC подвержен этой атаке.

Encrypt-then-MAC позволил бы полностью избежать этой уязвимости.

MAC-then-CTR не подвержен: не нужно дополнять до блока.

Литература к лекции

1*. Bellare, M.; Namprempre, C. (2000), "Authenticated Encryption: Relations among notions and analysis of the generic composition paradigm" , in T. Okamoto, *Extended abstract in Advances in Cryptology: Asiacrypt 2000 Proceedings*

2*. Полный текст статьи из 1*:
`http://cseweb.ucsd.edu/~mihir/papers/oem.html` (2007)

3*. Документы NIST - описание режимов GCM, CCM, EAX.

4*. T.Duong, J.Rizzo, "Here Come The $\oplus$ Ninjas". (BEAST attack)
`https://bug665814.bugzilla.mozilla.org/attachment.cgi?id=540839`