Информационная безопасность

Обеспечение целостности данных на основе алгоритмов симметричного шифрования

(презентация к лекции 3)

Амелин Р. В. Информационная безопасность. Конспект лекций // Амелин Р.В. Лаборатория преподавателя [Электронный ресурс]. URL: rv-lab.ru (2017).

Коды аутентификации сообщений (MAC)

Message Authentication Codes

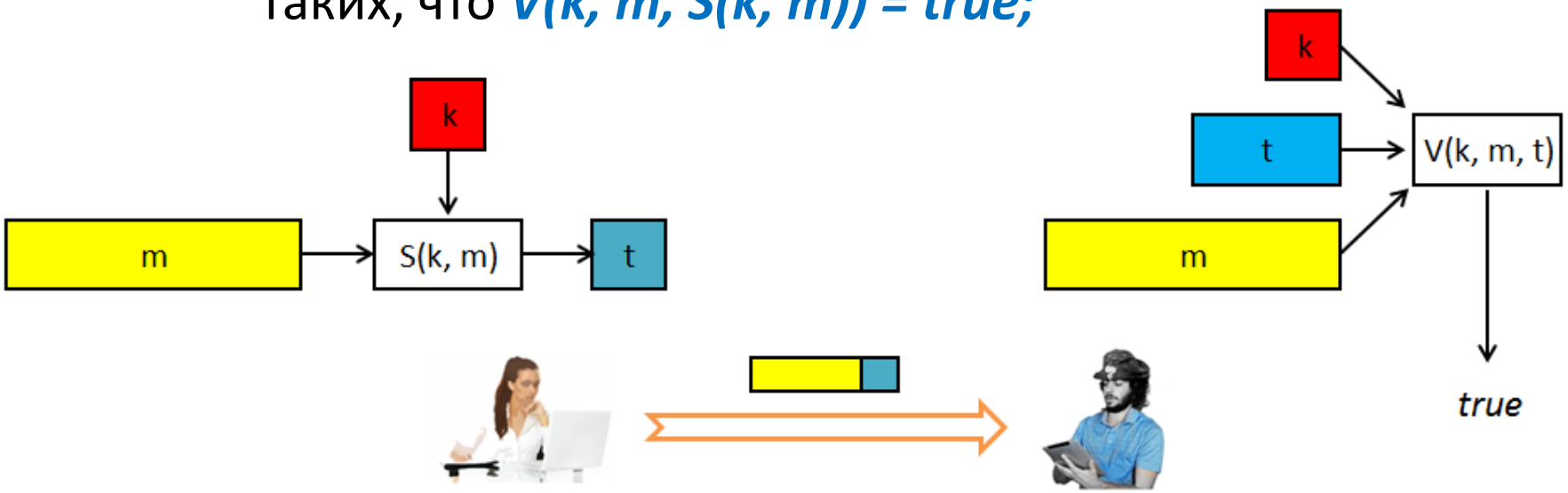
Задача: обеспечение целостности
Обычное шифрование не предназначено для этой цели

MAC представляет собой пару функций *I = (S, V)*

функция генерации тега $S: K \times X \rightarrow T$

функция проверки тега $V: K \times X \times T \rightarrow \{true, false\};$

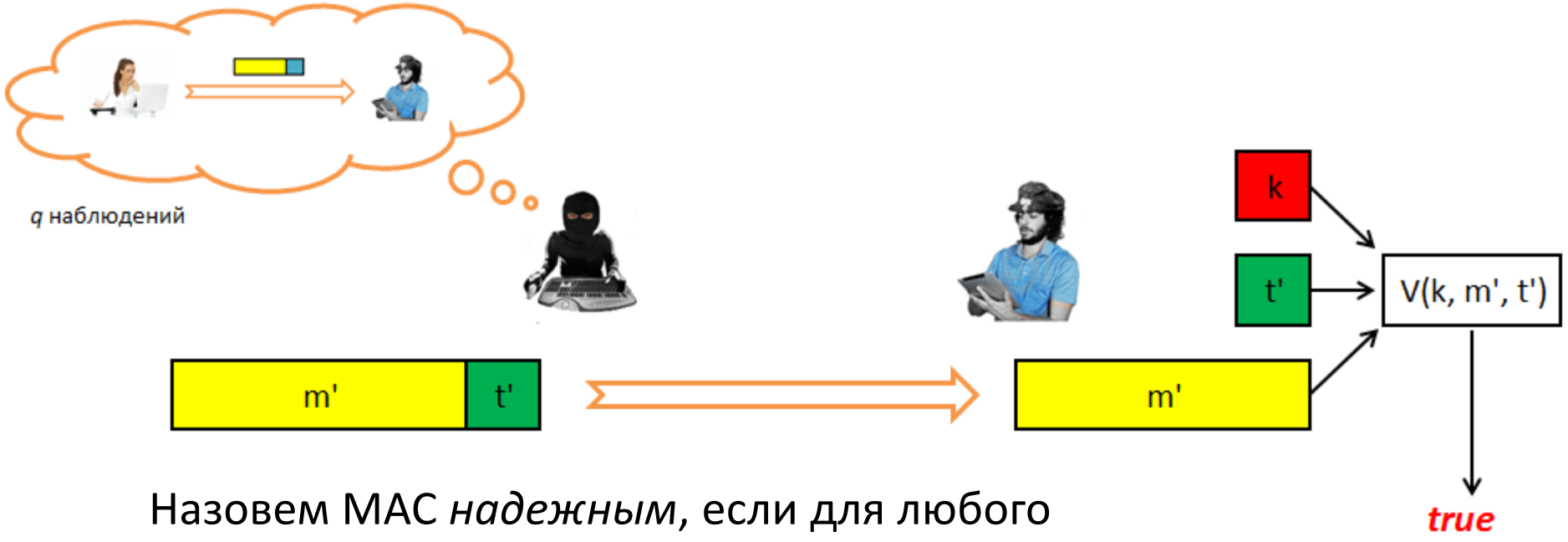
таких, что ***V(k, m, S(k, m)) = true;***



Цель противника – подделка MAC

Existential Forgery

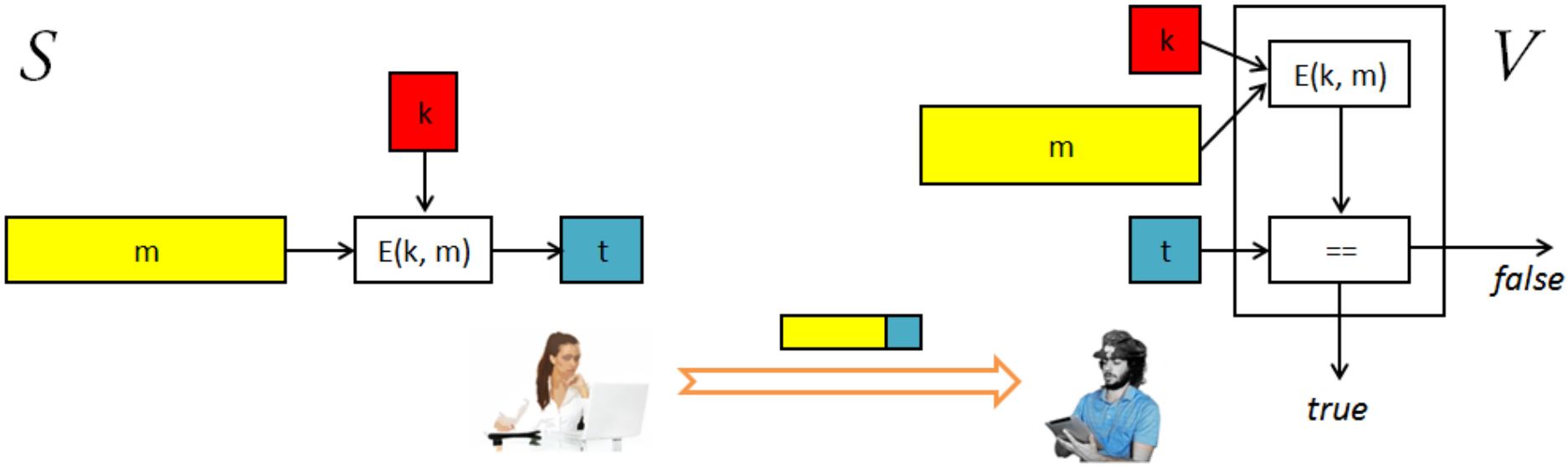
Подделка – это пара <m’, t’>, которая сгенерирована противником, не знающим ключа k и не наблюдавшего этой пары ранее (но наблюдавшим q других пар <m, S(k, m)>, такая что V (k, m’, t’) = true



Назовем MAC *надежным*, если для любого вычислительно эффективного алгоритма вероятность успешного создания подделки ничтожно мала

MAC на основе алгоритма шифрования

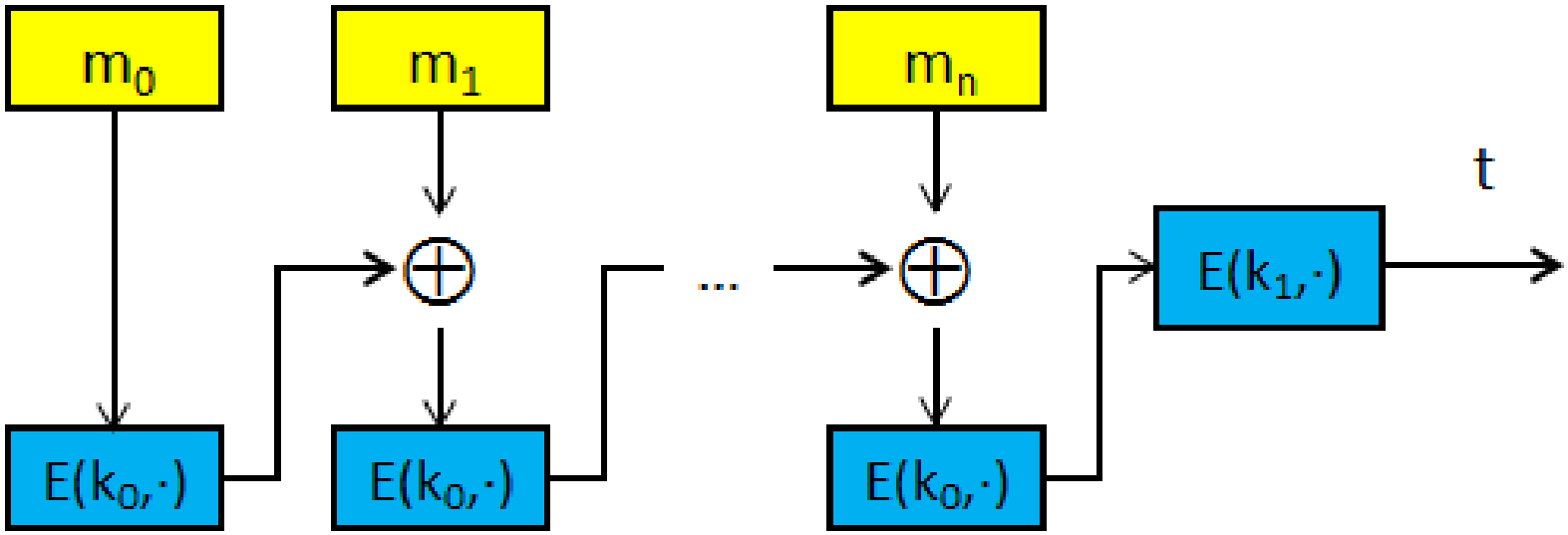
Пусть *E* – надежный алгоритм симметричного шифрования (например, AES).
Тогда в качестве S(k, m) возьмем E(k, m).
Функция проверки тега V(k, m, t) = (t == E(k, m)),
где «==» – оператор проверки на равенство



- Хорошо подходит для коротких сообщений (128 бит)
- Для сообщений произвольной длины не оптимален

ECBC-MAC

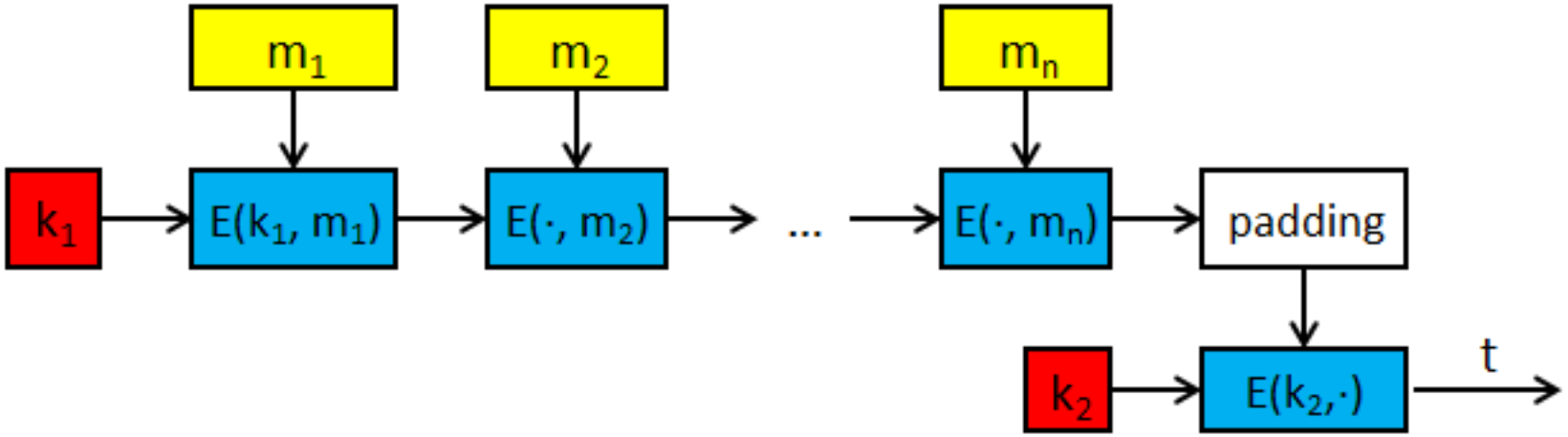
E – блочный шифр



Теорема. Если шифр E надежен, то ECBC-MAC на основе E также является надежным

NMAC

- E – блочный шифр, размер шифртекста должен совпадать с размером ключа
- Вместо E может использоваться псевдослучайная функция $F: X \times K \to K$



- В финале MAC шифруется отдельным подключом k_2, иначе противник может дописывать к перехваченному сообщению произвольное количество блоков и вычислять MAC, не зная ключа.

Хэш-функции

Хэш-функция H: X $\rightarrow$ T преобразует сообщение произвольной длины в битовую последовательность фиксированной длины |T| << |X|

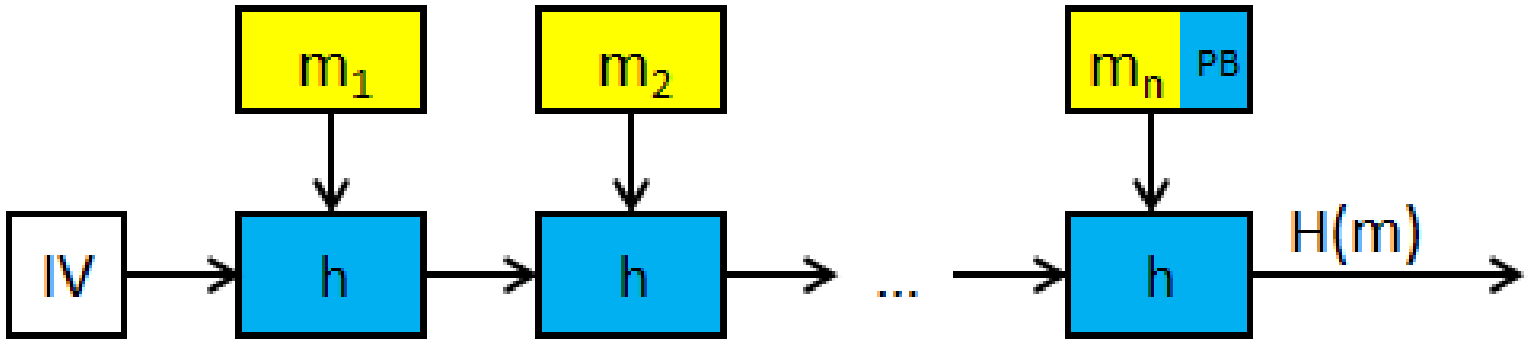
- не зависит от ключа и
- для одинаковых сообщений генерирует одинаковые хэши.

Ситуация, когда для двух разных сообщений $m_1 \neq m_2$ их хэши совпадают, т.е. $H(m_1) = H(m_2)$, называется *коллизией*.

Криптостойкие хэш-функции обладают следующими свойствами:

- *Односторонность (необратимость)*. Для любого *h* должно быть практически невозможно получить такое *x*, что H(*x*) = *h*.
- *Стойкость к коллизиям первого рода*. Для любого сообщения *x* должно быть практически невозможно получить другое сообщение *y*, такое что H(*x*) = H(*y*).
- *Стойкость к коллизиям второго рода*. Должно быть практически невозможно получить любую пару различных сообщений *x* и *y* для которых H(*x*) = H(*y*).

Структура Меркла-Дамгарда

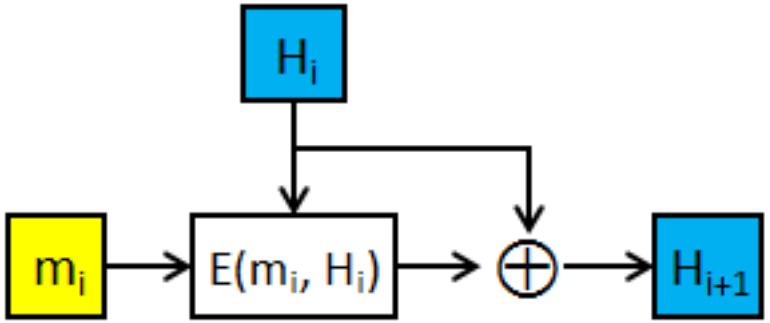


- **h: T × X → T** – функция сжатия, на основе которой строится хэш-функция
- **IV** – вектор инициализации (постоянный)
- **PB** – дополнение сообщения до длины, кратной |X|. Обычно берется последовательность 100…0 || длина сообщения (64 бита)

Функция Дэвиса-Мейера

Построение функции сжатия на основе алгоритма симметричного шифрования

$h(H, m) = E(m, H) \oplus H$



Пример: функция хэширования **SHA-256** построена на основе структуры Меркла-Дамгарда с использованием функции Дэвиса-Мейера (в качестве алгоритма шифрования используется **SHACAL-2**).

Доказано, что для идеального шифра E нахождение коллизии функции Дэвиса-Мейера потребует $O(2^{n/2})$ вычислений, где n – длина хэша. Это наилучший из возможных результатов, поскольку именно таких усилий требует подбор коллизии методом грубой силы – на основе **парадокса задачи о днях рожденья.**

HMAC

MAC на основе хэш-функции

$$\mathrm{S}(k, m) = \mathrm{H}(k \oplus \mathrm{opad} \ \| \ \mathrm{H}(k \oplus \mathrm{ipad} \| m))$$

- **opad и ipad –** два стандартных блока, которые накладываются на ключ для обеспечения лучшего лавинного эффекта
- Хэш-функция применяется дважды: сначала к комбинации ключа и сообщения, а затем к комбинации ключа и результата предыдущего применения
- Наиболее часто используемый MAC в интернет-протоколах