ЛЕКЦИЯ 2. РЕЖИМЫ ШИФРОВАНИЯ И КРИПТОГРАФИЯ С ОТКРЫТЫМ КЛЮЧОМ

Сергей Николенко

1. Режимы шифрования

Первая тема этой лекции относится к тому, как на практике применять криптографию с закрытым ключом. Большинство алгоритмов кодирования с закрытым ключом (такие, как DES, AES, IDEA и т.п.) кодируют только один блок сообщения небольшой длины (порядка 256-1024 бит). На практике же нужно уметь надёжно передавать большие сообщения, состоящие из большого числа таких блоков.

Режим шифрования — это метод применения блочного шифра, позволяющий преобразовать последовательность блоков открытых данных в последовательность блоков зашифрованных данных. При этом для шифрования одного блока могут использоваться данные другого блока. Обычно режимы шифрования используются для модификации процесса шифрования так, чтобы результат шифрования каждого блока был уникальным вне зависимости от шифруемых данных и не позволял сделать какие-либо выводы об их структуре. Существует несколько стандартных режимов шифрования.

1.1. **Electronic Code Book (ECB).** В режиме ECB сообщение делится на блоки, и каждый блок шифруется отдельно (независимо от других, на одном и том же ключе). Этот режим называется режимом электронной кодовой книги, так как существует возможность создать книгу, в которой каждому блоку открытого текста будет сопоставлен блок зашифрованного текста.

У метода ECB есть ряд недостатков, из-за которых применять его на практике не рекомендуется. Во-первых, одинаковые блоки открытого текста шифруются в одинаковые же блоки зашифрованного текста при одном и том же ключе. Таким образом, увидев два одинаковых шифра, противник поймёт, что на этих местах были две одинаковых части сообщения. Во-вторых, блоки могут пропадать или появляться независимо от других блоков. Злоумышленник может перехватить блок и продублировать его или исключить из сообщения, и декодирующая сообщение сторона ничего не заметит.

1.2. **Cipher Block Chaining (CBC).** В этом методе результат шифрования предыдущих блоков используется для шифрования текущего блока; таким образом в шифровании появляется механизм обратной связи. Это означает, что любой блок шифра зависит не только от исходного текста, но и от всех предыдущих блоков текста. В Cipher Block Chaining (CBC) перед шифрованием текст XOR'ится с предыдущим зашифрованным блоком. Декодирование проводится аналогично.

Законспектировала Тузова Екатерина.

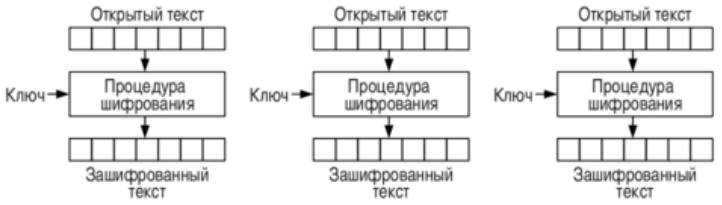

Рис. 1. Процесс шифрования в режиме ECB.

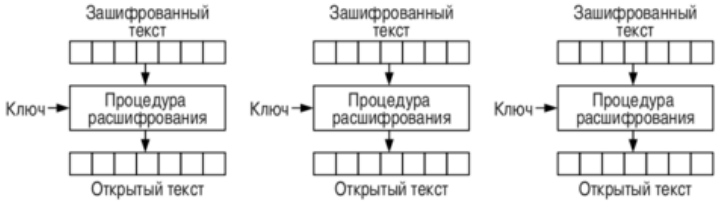

Рис. 2. Процесс дешифрования в режиме ECB.

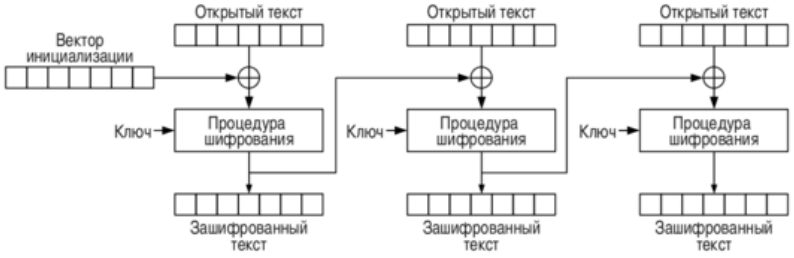

Рис. 3. Процесс шифрования в режиме CBC.

При этом в начале кодирования используется вектор инициализации для того, чтобы любое сообщение было по-настоящему уникальным. Вектор инициализации должен быть случайным числом; его не обязательно хранить в секрете, можно передавать его вместе с сообщением в открытом виде.

CBC более надёжен, чем ECB, и часто применяется в реальных приложениях. Важный его недостаток заключается в том, что кодирование обязательно получается последовательным, его невозможно распараллелить: нужно дождаться окончания кодирования предыдущего блока, прежде чем начинать кодировать следующий. Декодирование, с другой стороны, распараллелить можно: результат декодирования блока зависит только от него самого и одного его соседа.

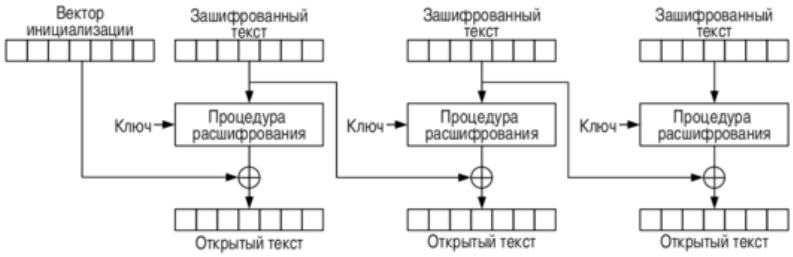

Рис. 4. Процесс дешифрования в режиме CBC.

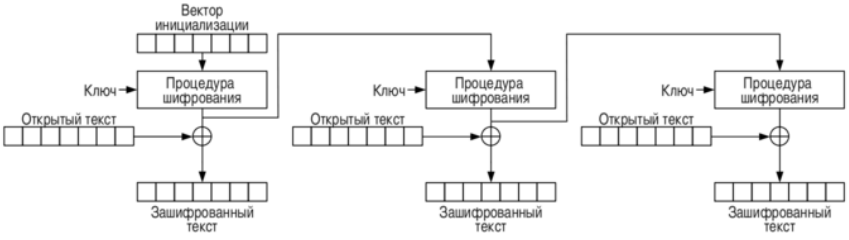

Рис. 5. Процесс шифрования в режиме OFB.

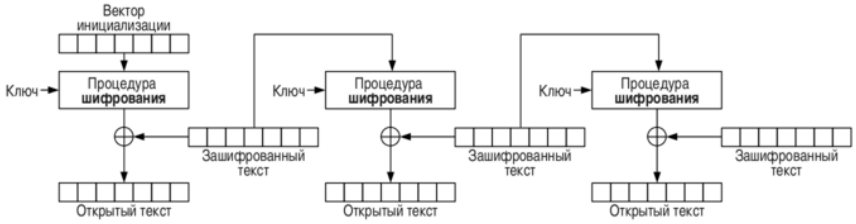

Рис. 6. Процесс дешифрования в режиме OFB.

1.3. **Output Feedback Mode (OFB).** Режим обратной связи вывода (OFB) генерирует ключевые блоки, каждый из которых является результатом шифрования предыдущего ключевого блока. Каждая операция блочного шифра обратной связи вывода зависит от всех предыдущих.

Основное преимущество режима OFB состоит в том, что если при передаче произошла ошибка, то она не распространяется на следующие зашифрованные блоки, и тем самым сохраняется возможность дешифрования последующих блоков. Например, если появляется ошибка в i-том блоке, это приведет только к невозможности дешифрования этого блока. Дальнейшая последовательность блоков будет расшифрована корректно.

1.4. **Counter Mode (CTR).** Шифрование происходит аналогично OFB, только теперь каждый новый ключевой блок является результатом шифрования

`RSA-подпись`:

(1) Получить открытый текст M.

(2) Создать цифровую подпись σ, с помощью своего секретного ключа (d, n):

$$\sigma = S_A(M) = M^d \mod n.$$

(3) Передать пару (M, σ), состоящую из сообщения и подписи.

`Проверка RSA-подписи`:

(1) Получить на вход пару (M, σ), а также открытый ключ (e, n) стороны A.

(2) Проверить подлинность подписи: если

$$P_A(\sigma) = \sigma^e \equiv M \mod n,$$

то подпись верная.

Рис. 7. Схема цифровой подписи RSA.

исходного вектора с добавлением единицы. Таким образом можно начинать кодировать и декодировать сообщение с любого места, что очень важно, например, при передаче больших файлов (представьте, что вам нужно заглянуть в середину большого зашифрованного видеофайла). Математически говоря, если r — исходный вектор инициализации, то $r_i = K(r + i - 1)$.

Главным недостатком CTR является то, что для расшифровки всего сообщения достаточно расшифровать один из $K(r_i)$, дальше всё получится автоматически.

2. Криптография с открытым ключом

Как строить криптосистемы с открытым ключом? Мы уже знаем, что Боб должен суметь зашифровать сообщение, но потом никто не должен иметь возможности расшифровать его — кроме Алисы, конечно. Говоря математическим языком, функция шифрования должна легко вычисляться, а вот вычисление обратной к ней должно быть «практически неосуществимым» (из-за большого, желательно экспоненциально большого, объёма вычислений). Для этого нужно придумать вычислительно сложную задачу, обратную к которой было бы легко вычислить (одностороннюю функцию и даже более того — функцию с секретом).

Большинство современных алгоритмов криптографии с открытым ключом в качестве таких сложных вычислительных задач используют либо разложение чисел на множители (легко перемножить два числа, трудно разложить число на множители), либо задачу дискретного логарифма (легко возвести число в степень в конечной группе, трудно определить эту степень по результату возведения и исходному числу). Мы сейчас кратко изложим основные криптографические примитивы, основанные на этих задачах.

2.1. **Разложение на множители.** Одной из первых криптосистем с открытым ключом была система RSA, названная так по именам создателей: Ривеста

Кодирование:

(1) На вход подаётся публичный ключ (y, p, q, g), где

$$y = g^x \mod p.$$

(2) Боб кодирует сообщение m парой (c_1, c_2), где

$$\begin{aligned} c_1 &= g^r \mod p, \\ c_2 &= y^r m \mod p. \end{aligned}$$

Декодирование:

(1) Алиса получает (c_1, c_2) и вычисляет

$$y^r = g^{xr} = c_1^x \mod p.$$

(2) Теперь $m = c_2(y^r)^{-1}$.

Рис. 8. Криптосистема ElGamal.

(Rivest), Шамира (Shamir) и Адлемана (Adleman). Принцип работы RSA основан на сложности задачи разложения на множители.

Секретный ключ Алисы в RSA — это два простых числа p и q. Алиса публикует произведение этих чисел $N = pq$. Ключевое предположение, на котором основана RSA, — то, что большое число трудно разложить на множители, то есть N можно распространять, а p и q всё равно никто не узнает.

Кроме собственно $N = pq$, в открытый ключ входит также число e (экспонента), которое должно быть меньше числа $\varphi(N) = (p-1)(q-1)$ и взаимно просто с ним. $\varphi(N)$ — это функция Эйлера; в общем случае функция Эйлера от n равна количеству чисел, меньших n и взаимно простых с ним. Секретный же ключ RSA — такое число d, что

$$de \equiv 1 \mod \varphi(N).$$

Алиса может легко вычислить d (делить числа по простому модулю несложно — например, алгоритмом Евклида), но никто другой на это не способен — ведь если трудно разложить N на множители, то трудно и подсчитать функцию Эйлера. Боб теперь, желая зашифровать свое сообщение m, вычисляет

$$y \equiv m^e \mod N$$

и передаёт его Алисе (и всем желающим). Алиса может расшифровать его, вычислив

$$y^d \equiv m^{ed} \equiv m \mod N$$

(последнее сравнение выполняется благодаря малой теореме Ферма).

Несложно модифицировать RSA-идею так, чтобы получить алгоритм криптографической подписи, основанный на сложности задачи разложения чисел на множители. Он представлен на рис. 7.

2.2. **Дискретный логарифм.** Задача дискретного логарифма в общем виде ставится в произвольной абелевой группе. Рассмотрим некоторую коммутативную группу G и её элемент $g \in G$. Требуется по g и g^a, где $g^a = g \cdot g \cdot \ldots \cdot g$ (a раз) найти a. В нескольких последующих лекциях нашей целью будет изучение начальных сведений об эллиптических кривых, чтобы научиться строить

`Создание подписи`:

(1) На вход подаётся текст сообщения M, а также параметры схемы (публичный ключ) (y, p, q, g), где $y = g^x \mod p$.

(2) Вычислить:

$$\begin{aligned} T &= g^r && \mod p, \\ t &= T && \mod q, \\ S &= r^{-1}(M + xt) && \mod q. \end{aligned}$$

(3) Выдать пару (t, S).

`Проверка подписи`:

(1) На вход подаётся пара (t, S) и параметры (y, p, q, g), где $y = g^x \mod p$.

(2) Вычислить:

$$\begin{aligned} r &= S^{-1}(M + xt) && \mod q, \\ g^r &= g^{S^{-1}(M+xt)} && \mod p, \\ T &= g^{S^{-1}M} g^{S^{-1}tx} = g^{S^{-1}M} y^{S^{-1}t} && \mod p. \end{aligned}$$

(3) Если $T = t \mod q$, подпись верная.

Рис. 9. Схема цифровой подписи ElGamal.

криптографические примитивы, в которых групповой операцией будет сложение двух точек на эллиптической кривой. А на этой лекции мы ограничимся более простым случаем: $G = \mathbb{Z}_p^*$ с обычным умножением по модулю p.

Первым криптографическим примитивом с открытым ключом был протокол согласования ключа Диффи-Хеллмана (Diffie-Hellman key agreement scheme). Суть протокола согласования ключа заключается в том, чтобы перед началом общения, зашифрованного закрытым ключом, посредством криптографии с открытым ключом договориться о закрытом ключе. Иными словами, цель не в том, чтобы передать сообщение от одного участника к другому, а в том, чтобы у обоих участников в конце концов обнаружилось одно и то же число, которое можно будет использовать как секретный ключ.

В протоколе Диффи-Хеллмана это делается так: Алиса и Боб выбирают простое число p, по модулю которого будут проводиться все вычисления, и основание g, которое они потом будут возводить в степень. Затем Алиса выбирает число a (никому его не показывая) и передаёт Бобу $g^a \mod p$, а Боб выбирает число b и передаёт Алисе $g^b \mod p$.

Теперь Алисе достаточно возвести полученное число в свою секретную степень a, а Бобу — в свою секретную степень b, и у них обоих будет один и тот же ключ, так как

$$g^{ab} \mod p = g^{ba} \mod p.$$

Такая схема будет работать над произвольной абелевой группой G, и именно в этом месте приходится по существу использовать коммутативность.

Стойкость этого протокола согласования ключа связана с вычислительной трудностью операции дискретного логарифмирования — зная $g^a \mod p$, g и p

(все эти числа в принципе доступны противнику), вычислить a. Нужно, впрочем, отметить, что стойкость системы Диффи-Хеллмана на самом деле основана на немного другой задаче (которая так и называется Diffie-Hellman problem, DHP): по g, g^a и g^b найти g^{ab}. Очевидно, что, умея решать задачу дискретного логарифма, легко решить и DHP, но обратное неизвестно, поэтому DHP может оказаться проще.

На основе протокола Диффи-Хеллмана были разработаны «настоящая» криптосистема и алгоритм цифровой подписи. Они оба носят имя египетского криптографа Тахира Эль-Гамаля, который разработал их в середине 1980-х. Алгоритмы, их реализующие, изображены на рис. 8 и 9.