3.3. Алгоритм AES (Rijndael)

Название AES присвоено алгоритму Rijndael — победителю конкурса AES. Конкурс AES подробно описан в разд. 2.1, а здесь рассмотрим структуру ал­горитма и результаты его криптоанализа.

Структура алгоритма

Алгоритм AES представляет блок данных в виде двумерного байтового мас­сива размером 4x4. Все операции производятся над отдельными байтами массива, а также над независимыми столбцами и строками.

В каждом раунде алгоритма выполняются следующие преобразования (рис. 3.7) [132, 153]:

1. Операция SubBytes, представляющая собой табличную замену каждого байта массива данных согласно Таблице замен алгоритма AES (№ рисунка)

Таблица меняет входное значение 0 на 63 (шестнадцатеричное значение), 1 — на 7С, 2 — на 77 и т. д.

Вместо данной табличной замены можно выполнить эквивалентную ей комбинацию двух операций:

* вычисление мультипликативной обратной величины от входного зна­чения в конечном поле GF(28); обратной величиной от 0 является 0;
* выходное значение Ь вычисляется следующим образом:

bj — flj- © ^t44mod8 ©^('+5mod8 ® ®t + 6mod8 ® a/47mod8 ® ^i\*

где:

0 л, обозначает i-й бит величины л;

0 a — результат предыдущей операции;

0 с — шестнадцатеричная константа 63.

1. Операция ShiftRows, которая выполняет циклический сдвиг влево всех строк массива данных, за исключением нулевой (рис. 3.9). Сдвиг /-й стро­ки массива (для i = 1,2,3) производится на i байтов.
2. Операция MixColumns. Выполняет умножение каждого столбца массива данных (рис. 3.10), который рассматривается как полином в конечном поле GF( 28), на фиксированный полином а(х):

а(х) = Зх3 + х1 + х + 2.

Умножение выполняется по модулю х\* + 1.

1. Операция AddRoundKey. Выполняет наложение на массив данных мате­риала ключа. А именно, на i-й столбец массива данных (/ = 0...3) побито­вой логической операцией «исключающее или» (XOR) накладывается оп­ределенное слово расширенного ключа WArVt, где г— номер текущего раунда алгоритма, начиная с 1 (процедура расширения ключа будет опи­сана далее). Операция AddRoundKey представлена на рис. 3.11.

Зависимость количества раундов алгоритма R от размера ключа приведена в табл. 3.2.

Таблица 3.2

|  |  |
| --- | --- |
| Размер ключа в битах | Количество раундов |
| 128 | 10 |
| 192 | 12 |
| 256 | 14 |

Перед первым раундом алгоритма выполняется предварительное наложение ма­териала ключа с помощью операции AddRoundKey, которая производит наложе­ние на открытый текст первых четырех слов расширенного ключа Vy0...W3.

Последний же раунд отличается от предыдущих тем, что в нем не выполня­ется операция MixColumns.

Процедура расширения ключа

Алгоритм AES использует ключи шифрования трех фиксированных разме­ров: 128, 192 и 256 битов. В зависимости от размера ключа конкретный вари­ант алгоритма AES может обозначаться как AES-128. AES-192 и AES-256 соответствен но [ 153].

Задача процедуры расширения ключа состоит в формировании нужного ко­личества слов расширенного ключа для их использования в операции AddRoundKey. Как было сказано выше, под «словом» здесь понимается 4-байтный фрагмент расширенного ключа, один из которых используется в первичном наложении материала ключа и по одному — в каждом раунде алгоритма. Таким образом, в процессе расширения ключа формируется 4 \* (/? + 1) слов.

Расширение клдлча выполняется в два этапа, на первом из которых произво­дится инициализация слов расширенного ключа (обозначаемых как №)): первые Nk (Nk — размер исходного ключа шифрования К в словах, т. е. 4, б или 8) слов Wj , т. е. i = 0...(Nk -1), формируются их последовательным за­полнением байтами ключа (рис. 3.12).

Последующие слова Wj формируются такой последовательностью операций для каждого i = Nk...{4 \* (if +1) -1):

1. Инициализируется временная переменная Т:

T = Wi.i.

1. Эта переменная модифицируется таким образом:

• если i кратно Nk, то:

Т = SubWord(RotWord(T))© RCllNk;

операции SubWord и RotWord будут описаны далее, а константы RC„ представляют собой слова, в которых все байты, кроме первого, являются нулевыми, а первый байт имеет значение 2"“ mod 256;

* если Nk = 8 и (i modNk) = 4 , то:

Т = SubWord(T);

* в остальных случаях модификация переменной Т не выполняется.

1. Формируется i-е слово расширенного ключа:

W,=W,\_Nk®T.

Операция SubWord выполняет над каждым байтом входного значения таб­личную замену, которая была описана выше — см. операцию SubBytes.

Операция RotWord побайтно вращает входное слово на 1 байт влево.

Как видно, процедура расширения ключа является достаточно простой по сравнению со многими другими современными алгоритмами шифрования. Процедура расширения ключа имеет также несомненное достоинство в том, что расширение ключа может быть выполнено «на лету» (on-the-fly), т. е. па­раллельно с зашифровыванием данных.

Авторы алгоритма в [132] пишут также, что не следует задавать напрямую расширенный ключ — программная или аппаратная реализация алгоритма должна именно получать исходный ключ шифрования К и выполнять про­цедуру расширения ключа. Здесь стоит снова вспомнить алгоритм DES — известно, что DES с независимо задаваемыми ключами раундов оказался слабее против некоторых атак, чем исходный алгоритм DES (см. разд. 3.15).

Расшифровывание

Расшифровывание выполняется применением обратных операций в обратной последовательности. Соответственно, перед первым раундом расшифровы­вания выполняется операция AddRoundKey (которая является обратной са­мой себе), накладывающая на шифртскст четыре последних слова расширен­ного ключа, т. е. И,4Я...И/4/г+з.

Затем выполняется R раундов расшифровывания, каждый из которых осуще­ствляет следующие преобразования (рис. 3.13):

1. Операция InvShiftRows производит циклический сдвиг вправо трех по­следних строк массива данных на то же количество байтов, на которое выполнялся сдвиг операцией ShiftRows при зашифровывании.
2. Операция InvSubBytes производит побайтно обратную табличную замену, которая приведена в табл. 3.3.

Такую табличную замену можно выполнить, применив к входному байту преобразование, обратное второму действию альтернативной операции SubBytes (см. описание SubBytes), после чего вычислить мультипликатив­ную обратную величину от результата предыдущей операции в конечном поле GF( 28).

1. Операция AddRoundKey, как и при зашифровывании, выполняет наложе­ние на обрабатываемые данные четырех слов расширенного ключа lV4r...W4rt3 . Однако нумерация раундов г при расшифровывании произво­дится в обратную сторону — от (R -1) до 0.
2. Операция InvMixColumns выполняет умножение каждого столбца массива данных аналогично прямой операции MixColumns, однако, умножение производится на полином а"'(л), определенный следующим образом:

a‘,U) = Bx-3 + Dx2 + 9\* + Е.

Аналогично зашифровыванию, последний раунд расшифровывания не со­держит операцию InvMixColumns.

Отличия AES

от исходного алгоритма Rijndael

Алгоритм Rijndael позволяет шифровать данные не только 128-битными бло­ками. но и блоками по 192 или 256 битов. Таким образом, алгоритм AES, фактически, имеет лишь одно принципиальное отличие от Rijndael: он пре­дусматривает использование только 128-битных блоков данных. Рассмотрим изменения в приведенном выше описании алгоритма AES, связанные с дру­гими размерами блоков.

* Обрабатываемые данные могут представляться не только в виде массива размером 4 х 4, но и 4 х 6 или 4x8 для 192- и 256-битных блоков соответ­ственно.
* Количество раундов R алгоритма Rijndael определяется табл. 3.4 в зависи­мости не только от размера ключа, но и от размера блока.

Таблица 3.4

|  |  |  |  |
| --- | --- | --- | --- |
| Размер ключа в битах | Размер блока в битах | | |
| 128 | 192 | 256 |
| 128 | 10 | 12 | 14 |
| 192 | 12 | 12 | 14 |
| 256 | 14 | 14 | 14 |

О Количество битов сдвига строк таблицы также зависит от размера блока (табл. 3.5).

Таблица 3.5

|  |  |  |  |
| --- | --- | --- | --- |
| Номер строки | Количество битов сдвига | | |
| 128 | 192 | 256 |
| 1 | 1 | 1 | 1 |
| 2 | 2 | 2 | 3 |
| 3 | 3 | 3 | 4 |

* Поскольку для 192- и 256-битного блоков увеличивается количество столбцов массива данных до 6 и 8 соответственно, в операции AddRoundKey участвуют уже 6 или 8 слов расширенного ключа вместо четырех. Следовательно, в г-м раунде алгоритма выполняется наложение слов расширенного ключа IVvjv- ^wvj■ гДе — количество столбцов массива данных.
* В связи с вышесказанным, изменяется и процедура расширения ключа, однако изменение состоит лишь в том, что эта процедура должна вырабо­тать Nb\*(R +1), а не 4\*(Л + 1) слов расширенного ключа (что, впрочем, остается справедливым для 128-битного блока).

Первичная оценка криптостойкости алгоритма Rijndael

Первичная оценка криптостойкости алгоритма Rijndael была приведена авто­рами алгоритма в его спецификации [132], представленной на конкурс AES. Согласно оценкам авторов, Rijndael не подвержен следующим видам крип­тоаналитических атак:

* у алгоритма отсутствуют слабые ключи, а также возможности его вскры­тия с помощью атак на связанных ключах (описание криптоаналитических атак, упомянутых здесь и далее, приведено в разд. 1.5-1.9);

О к алгоритму не применим дифференциальный криптоанализ;

О алгоритм не атакуем с помощью линейного криптоанализа и усеченных дифференциалов;

* Square-атака (специфичная атака на алгоритмы со структурой «квадрат», к которым относится и AES; подробно описана в [130]) также не приме­нима к алгоритму Rijndael;
* алгоритм не вскрывается методом интерполяции.

Криптоанализ алгоритма Rijndael в рамках конкурса AES

Сначала рассмотрим те результаты криптоанализа, которые были учтены экспертами при выборе алгоритм — победителя конкурса AES. Прежде все­го, было найдено несколько атак на усеченные (с уменьшенным количеством раундов) версии алгоритма Rijndael [283]:

* в упомянутой выше первичной оценке криптостойкости были приведены некоторые атаки на усеченные версии Rijndael [132], из которых стоит от­метить Square-атаку на 6-раундовую версию алгоритма, для которой необ­ходимо 232 выбранных открытых текстов и 272 операций шифрования.
* Square-атака на 6-раундовую версию Rijndael была незначительно усилена Эли Бихамом и Натаном Келлером (Nathan Keller) [73], которые также предложили атаку методом невозможных дифференциалов на 5-раундовую версию алгоритма; для нее требуется 229,5 выбранных открытых текстов и 231 операций; атака с помощью невозможных дифференциалов рядом корейских специалистов была распространена на 6-раундовую версию Rijndael [117]: для данной атаки требуется 291'5 выбранных открытых тек­стов и 2122 операций шифрования;
* впоследствии Square-атака была расширена на 7 раундов алгоритма Rijndael; однако данная атака весьма непрактична: для ее реализации тре­буется 232 выбранных открытых текстов и от 2184 до 2208 операций шиф­рования [244];

О авторы алгоритма Twofish (финалиста конкурса AES) с участием ряда других специалистов продолжили усиление Square-атаки против алгорит­ма Rijndael: новая Square-атака позволяла вскрыть 6-раундовый Rijndael выполнением 244 операций, а 7-раундовый— от 2155 до 2172 операций шифрования при том же требуемом количестве выбранных открытых тек­стов [147]; кроме того, новая атака позволяет вскрыть 7- и 8-раундовые (последнюю — только при использовании 256-битного ключа) версии Rijndael при наличии от 2119 до 2128 выбранных открытых текстов вы­полнением, соответственно, 2120 или 2204 операций;

* та же команда криптологов предложила атаку на связанных ключах на 9- раундовую версию Rijndael с 256-битным ключом, которой необходимо 277 выбранных открытых текстов, зашифрованных на 256 связанных ключах, и 2224 операций шифрования.

С учетом того, что в алгоритме Rijndael выполняется, как минимум, 10 раун­дов, запас криптостойкости алгоритма экспертами был признан адекватным [283]. С этим, однако, согласны далеко не все специалисты. В частности, в материалах [51,215, 348, 350] достаточным запасом криптостойкости счи­тается двукратное увеличение количества раундов алгоритма по сравнению с максимально атакуемым (однако стоит отметить, что далеко не все атаки на Rijndael с уменьшенным количеством раундов являются осуществимыми на практике [372]). В [51] достаточно убедительно утверждается, что алгоритм, выбранный в качестве стандарта AES, должен оставаться криптографически стойким до 2100 г. Ясно, что за почти сто лет будет сделано огромное количе­ство попыток вскрытия AES («AES будет наиболее привлекательной мише­нью для лучших криптоаналитиков мира» [350]), некоторые из которых при­ведут к увеличению количества вскрываемых раундов. Судя по последнему десятилетию, появятся и принципиально новые виды криптоаналитических атак. Поэтому, считая формально, согласно [51] и [348], Rijndael должен вы­полнять, как минимум, 18 раундов.

Авторы [348] утверждают, что в алгоритме Rijndael сделан чрезмерный акцент на быстродействие в ущерб криптостойкости, и рекомендуют даже 24 раунда, а не 18. Ясно, что в этом случае Rijndael по быстродействию будет заметно проигрывать другим алгоритмам — финалистам конкурса AES. Они же считают, что успешные атаки на алгоритм Rijndael происходят именно благодаря его чересчур простой структуре. Стоит сказать, что документ [348] также написан авторами «конкурирующего» с Rijndael алгоритма Twofish; в нем настолько убедительно доказывается преимущество Twofish над ос­тальными финалистами конкурса (аналогичные доказательства приведены и в работе [350]), что после его прочтения выбор Rijndael (а не Twofish) в ка­честве стандарта AES начинает удивлять.

В рамках исследований в течение первого раунда конкурса AES было также отмечено [284], что Rijndael имеет ряд потенциальных уязвимостей при реа­лизации данного алгоритма в смарт-картах.

* Rijndael может быть подвержен атаке по потребляемой мощности, наце­ленной на его процедуру расширения ключа [80]. что, однако, не доказано авторами [80] Эли Бихамом и Эди Шамиром.
* Весьма интересное исследование [114] на данную тему было проведено специалистами исследовательского центра компании IBM. Они выполни­ли реализацию алгоритма Twofish для типичной смарг-карты с CMOS- архитектурой и проанализировали возможность атаки на данный алгоритм с помощью дифференциального анализа потребляемой мощности (DPA — см. разд. 1.7). Оказалось, что атаке подвержена процедура входного отбе­ливания алгоритма Twofish, в результате чего можно вычислить ключ шифрования данного алгоритма. Атака была (теоретически) распростра­нена на остальные алгоритмы — участники конкурса AES. Аналогично алгоритму Twofish, с помощью DPA атакуется предварительное наложе­ние материала ключа, выполняемое в алгоритме Rijndael, после чего вы­числяется ключ шифрования целиком. Авторы [114] утверждают, что сре­ди алгоритмов — финалистов конкурса AES алгоритмы MARS и RC6 подвержены этой атаке несколько менее, чем Twofish, Rijndael и Serpent.

Эти погенциальные уязвимости не повлияли на выбор алгоритма Rijndael в качестве стандарта AES, поскольку, во-первых, остальные алгоритмы- финалисты не принципиально лучше в данном контексте, а во-вторых, суще­ствуют различные методы противодействия атакам по потребляемой мощно­сти (в частности, описаны в [ 114]), которые, однако, усложняют реализацию и ухудшают быстродействие алгоритма.

Среди других исследований можно отметить работу [277], в которой утвер­ждается, что алгоритм Rijndael не обеспечивает достаточное рассеивание (распространение влияния одного бита открытого текста на несколько битов шифртекста) данных (это указано и в [350], где предложены конкретные ме­ры усиления алгоритма), однако конкретных атак в [277] предложено не бы­ло. Дискуссия между авторами [277] и авторами Rijndael продолжилась [133, 276], однако эта потенциальная проблема также не повлияла на выбор алго- ритма-победителя.

Криптоанализ алгоритма после конкурса AES

Как и предполагали эксперты, после принятия алгоритма Rijndael в качестве стандарта AES попытки вскрытия этого алгоритма существенно усилились.

Можно сказать, что криптоанализ алгоритма AES стал развиваться, в основ­ном, в следующих четырех направлениях.

Во-первых, были предприняты попытки усиления «классических» атак или применения других известных атак к данному алгоритму. Например, работа [84] описывает атаку методом бумеранга на 6-раундовую версию алгоритма со 128-битным ключом, для выполнения которой требуется 239 выбранных открытых текстов, 271 шифртекстов с адаптивным выбором и 271 операций шифрования.

Во-вторых, имело место применение различных методов криптоанализа на связанных ключах, в частности:

* в работе [406] предложено несколько вариантов атак на связанных ключах на 7- и 8-раундовый AES-192 с использованием невозможных дифферен­циалов;
* комбинация метода бумеранга и связанных ключей предложена в работе

[70]: 9-раундовый AES-192 атакуется при наличии 279 выбранных откры­тых текстов, каждый из которых шифруется на 256 связанных ключах, выполнением 2125 операций шифрования; для атаки на 10-раундовый AES-256 требуется 2114,9 выбранных открытых текстов (включая зашиф- ровывание на 256 связанных ключах) и 2 ' операций; данная атака ис­

пользует слабость процедуры расширения ключа, состоящую в ее недос­таточной нелинейности;

О эта атака была усилена в работе [202], в которой, в частности, предлагает­ся атака на 10-раундовый алгоритм AES-192; для новой атаки требуется 2125 выбранных открытых текстов (на 256 связанных ключах) и 2146'7 операций.

Несмотря на то, что предложенные атаки на связанных ключах являются весьма непрактичными, настораживает тот факт, что атаке подвержены уже 10 из 12 раундов алгоритма AES-192 (и это после всего 5 лет после принятия стандарта AES!)— возникает опасение, что эксперты (указывающие на не­достаточность раундов в алгоритме Rijndael) были правы и полнораундовый алгоритм AES будет вскрыт существенно раньше, чем предполагали экспер­ты института NIST.

В-третьих, многие исследования были посвящены алгебраической структуре алгоритма Rijndael, например:

О в работе [156] найдены линейные соотношения в таблице замен Rijndael (т. е. в единственном нелинейном элементе алгоритма); однако, как и в дру­гих аналогичных работах, каких-либо практических возможностей ис­пользования этого свойства не предложено;

О как показано в работе [149], зашифровывание с помощью Rijndael можно выразить относительно (особенно по сравнению с другими «серьезными» алгоритмами шифрования) простой формулой; авторы не нашли практи­ческого применения данной формулы, но предположили, что она будет использована в реальных атаках в течение ближайших примерно 20 лет;

* в работе [275] показано, что вскрытие алгоритма AES эквивалентно реше-

g

нию системы квадратичных уравнений в конечном поле GF( 2 ).

Попытки использования алгебраических свойств алгоритма для его вскрытия были названы «алгебраическими атаками» [121]. Стоит отметить, что были и работы с попытками доказательства того факта, что простая структура ал­горитма AES не ухудшает его криптостойкости, например, [361].

В-четвертых, больше всего исследований было посвящено атакам, исполь­зующим информацию, полученную по побочным каналам:

* во многих работах содержатся примеры успешного вскрытия различных реализаций полнораундового алгоритма AES с помощью атак по времени выполнения (например, [49]), потребляемой мощности (например, [248]) и атак на основе сбоев (например, [299]); автор [49] (эта работа описывает успешное вскрытие сервера OpenSLL, использующего для шифрования алгоритм AES), в частности, считает, что эксперты NIST при выборе Rijndael в качестве AES допустили весьма серьезную ошибку, посчитав, что время выбора значения из таблицы замен является константной вели­чиной в конкретной реализации; вывод автора таков: выбор Rijndael, ско­рее всего, был ошибочным;
* не меньше исследований (например. [124, 349]) посвящено безопасным (т. е. защищенным от утечки данных по побочным каналам) программным или аппаратным реализациям AES.

Довольно большой список работ, посвященных side-channel-атакам на AES и методам защиты от них, можно найти, например, на ресурсе [373], посвя­щенном криптоанализу AES.

Заключение

Итак, на май 2007 г., по прошествии всего 6 лет после принятия алгоритма Rijndael в качестве стандарта AES, криптоаналитики весьма серьезно про­двинулись во вскрытии данного алгоритма:

* предложена теоретическая атака уже на 10 раундов (из 12) алгоритма AES-192;
* существует множество примеров вскрытия реализаций алгоритма AES с помощью side-channel-атак.

Не правда ли, все это производит впечатление, что эксперты NIST могли ошибиться в выборе алгоритма — победителя конкурса AES?