

Московский физико-технический институт
(государственный университет)

Курс семинаров по предмету "Защита информации"
Эссе

Алгоритм Rijndael

Глаз Роман Сергеевич
Группа Б01-008а

Долгопрудный
2023

Содержание

1	Введение	1
2	Принцип работы	1
2.1	Краткое описание	1
2.2	Описание процедуры трансформации round	2
2.2.1	Про раундовые ключи	2
2.2.2	Описание процедуры	2
2.3	Описание вспомогательных процедур	2
2.3.1	Процедура <i>AddRoundKey</i>	2
2.3.2	Процедура <i>SubBytes</i>	2
2.3.3	Процедура <i>ShiftRows</i>	3
2.3.4	Процедура <i>MixColumns</i>	3
2.3.5	Алгоритм генерации раундовых ключей <i>KeyExpansion</i>	3
3	Список используемой литературы	3

1. Введение

Rijndael на данный момент является стандартом шифрования правительства США по результатам проведённого конкурса *Advanced Encryption Standard*, организованного Национальным институтом стандартов и технологий США.

Потребности принятия нового стандарта возникли из-за того, что предыдущий стандарт – *Data Encryption Standard* – имел ключ длиной всего в 56 бит, что позволяло взломать шифр простым перебором ключей.

Алгоритм *Rijndael* стал настолько популярным, что даже производители процессоров Intel и AMD ввели аппаратную поддержку инструкций, ускоряющих работу *Rijndael*.

2. Принцип работы

2.1. Краткое описание

Пусть имеется набор входных данных I и ключ K , а B – количество 32-битных слов, из которых состоят ключ и входные данные, то есть $I = (i_1, \dots, i_B, \dots, i_{4B})$ и $K = (k_1, \dots, k_V, \dots, k_{4V})$. Возможные значения B : 4, 5, 6, 7 и 8. Возможные значения V : 4, 5, 6, 7 и 8.

Rijndael сводится к следующей формальной процедуре: получить согласно некоторым правилам шифро-текст $C = (c_1, \dots, c_B, \dots, c_{4B})$.

Введём понятие S (state) – текущее состояние алгоритма, которое в начале соответствует входным данным I , в процессе применения алгоритма соответствует некоторому промежуточному представлению, а после применения алгоритма – шифро-тексту C . S является матрицей размером 4 x B .

Алгоритм состоит из следующих процедур:

1. Исходные данные помещаются в текущее состояние S по следующему правилу:

$$S = \begin{bmatrix} s_{11} & s_{12} & \dots & s_{1B} \\ \dots & & & \\ s_{41} & s_{42} & \dots & s_{4B} \end{bmatrix} = \begin{bmatrix} i_1 & i_2 & \dots & i_B \\ \dots & & & \\ i_{3B+1} & i_{3B+2} & \dots & i_{4B} \end{bmatrix} \quad (1)$$

2. К состоянию S применяется процедура трансформации – раунд (round) – $N_R - 1$ раз, где N_R может принимать значения от 10 до 14 включительно в зависимости от длины ключа K (10 раз соответствует минимальной длине ключа 128 бит и т.д.). Полное описание раунда изложено в главе 2.2.
3. К состоянию S применяется последний раунд N_R – он немного отличается от предыдущих (подробнее об этом позже, см главу 2.2.2).
4. Состояние S благополучно копируется в шифро-текст C :

$$C : \begin{bmatrix} c_1 & c_2 & \dots & c_B \\ \dots & & & \\ c_{3B+1} & c_{3B+2} & \dots & c_{4B} \end{bmatrix} = \begin{bmatrix} s_{11} & s_{12} & \dots & s_{1B} \\ \dots & & & \\ s_{41} & s_{42} & \dots & s_{4B} \end{bmatrix} \quad (2)$$

2.2. Описание процедуры трансформации round

2.2.1. Про раундовые ключи

Для каждого раунда генерируется собственный раундовый ключ W_r размером B 32-битных слов, где r – номер раунда. Вместе с исходным шифро-ключом $W_0 = K$ имеем массив ключей размером $B \cdot (N_R + 1)$ 32-битных слов: $W = (W_0, \dots, W_{N_R})$.

Процедура генерации раундовых ключей W_r называется "Расширение ключа" (*KeyExpansion, подробнее в главе 2.3.5*).

2.2.2. Описание процедуры

Процедура round при $0 \leq r < N_R$ над текущим состоянием S состоит из следующих этапов:

1. Применение процедуры "Сложить S с ключом раунда W_r " (*AddRoundKey, подробнее в главе 2.3.1*).
2. Применение процедуры "Использовать нелинейную таблицу замен для S " (*SubBytes, подробнее в главе 2.3.2*).
3. Применение процедуры "Сдвинуть строки в S " (*ShiftRows, подробнее в главе 2.3.3*).
4. Применение процедуры "Перемножить колонки S с полиномом" (*MixColumns, подробнее в главе 2.3.4*).

Процедура round при $r = N_R$, как уже было сказано, слегка отличается от предыдущих:

1. Применение процедуры "Сложить S с ключом раунда W_{N_R} " (*AddRoundKey, подробнее в главе 2.3.1*).
2. Применение процедуры "Использовать нелинейную таблицу замен для S " (*SubBytes, подробнее в главе 2.3.2*).
3. Применение процедуры "Сдвинуть строки в S " (*ShiftRows, подробнее в главе 2.3.3*).
4. Применение процедуры "Сложить S с ключом раунда" (*AddRoundKey, подробнее в главе 2.3.1*).

2.3. Описание вспомогательных процедур

2.3.1. Процедура *AddRoundKey*

TBD

2.3.2. Процедура *SubBytes*

TBD

2.3.3. Процедура *ShiftRows*

TBD

2.3.4. Процедура *MixColumns*

TBD

2.3.5. Алгоритм генерации раундовых ключей *KeyExpansion*

TBD

3. Список используемой литературы

- TBD
- TBD