Таблица 1 Степень экспоненциальной сложности ключевой системы АШД DSt2005.

Тип ключа шифрования	Степень экспоненциальной сложности
АШД DSt2005 – 128	
K_{sch} и K_{so} массива сеансового ключа шифрования	2^{80}
Первый массив этапных ключей k_{e1}	2^{128}
Второй массив этапных ключей k_{e2}	2^{128}
Ключевая система в целом k_{se}	2^{336}
АШД DSt2005 – 256	
K_{sch} и K_{so} массива сеансового ключа шифрования	2^{160}
Первый массив этапных ключей k_{e1}	2^{256}
Второй массив этапных ключей k_{e2}	2^{256}
Ключевая система в целом k_{se}	2^{672}

Литература

1. Столлингс В. Криптография и защита сетей: принципы и практика. – М. , Изд. дом «Вильямс», 2001. – 672 стр.

2. Венбо Мао. Современная криптография. Теория и практика. – Москва–Санкт-Петербург–Киев: Лори Вильямс, 2005. –768 стр.

3. Алферов А.П., Зубов А.Ю., Кузьмин А.С., Черемушкин А.В. Основы криптографии: Учебное пособие, 1-2-е изд. –М.: Гелиос АРВ, 2002.-480 с

О.Ш. Узаков

ОЦЕНКА ЭКСПОНЕНЦИАЛЬНОЙ СЛОЖНОСТИ СТАНДАРТА АЛГОРИТМА ШИФРОВАНИЯ ГОСТ 28147-89

(Каршинский филиал Ташкентского университета информационных технологий имени Мухаммада аль-Хоразмий)

Оценка экспоненциальной сложности дешифрования шифр текстов позволит дать ответ на вопрос, “каким должен быть крипто стойкий алгоритм шифрования”, чтобы дешифрование шифр текста не было практически возможным без исходного ключа шифрования. С развитием сетевых технологий и с ростом объемов передачи данных по сетям телекоммуникациям возникает проблема защиты информации, где активным способом защиты выступает применение криптографических методов, т.е. фактически возникает проблема создания современного стойкого алгоритма шифрования. Для этого необходимо знать степень полиномиальной сложности шифрования и экспоненциальной сложности дешифрования алгоритма шифрования.

Стандарт блочного шифрования данных ГОСТ 28147-89 является в России установленным единым алгоритмом криптографического преобразования данных для систем обработки информации в сетях ПЭВМ, отдельных вычислительных комплексах и ПЭВМ. Этот алгоритм шифрования предназначен для аппаратной и программной реализации, удовлетворяет необходимым криптографическим требованиям по стойкости и, следовательно, не накладывает ограничений на степень секретности защищаемой информации. В основе стандарта алгоритма шифрования ГОСТ 28147-89 лежит конструкция сети Фейстеля, приведенная так же на рис. 1.

Алгоритм реализует шифрование 64-битовых блоков данных с помощью 256-битового ключа, схема алгоритма шифрования приведена на рис. 1. Ниже вычислим оценку экспоненциальной сложности стандарта алгоритма шифрования ГОСТ 28147-89.

В начале осуществляется разделение выходного шифрованного блока данных 64 бита, на левую L_{32} и правую R_{32} части не дает сложности, причем $R_{31} = L_{32}$, а $R_{32} = L_{31} \oplus f(R_{31}, k_{32})$ известны. Для нахождения L_{31} и $f(R_{31}, k_{32})$ раундовой функции необходимо 2^{32} операций полного перебора – степень экспоненциальной сложности данного преобразования.

Далее осуществляется циклический сдвиг вправо, что никакой сложности не составляет – преобразование однозначно обратимо.

Секретность S-блоков таблица 2. увеличивает стойкость алгоритма шифрования. В каждом S-блоке, в интервале $0 \le S_j^i \le 15$ (i =1,2,…,8; j =0,1,2, …,15) имеется 16 не повторяющихся чисел в каждой строке, причем полная перестановка S_j^i равна 16!, и так как из общего количество S-блоков выбирается только восемь S-блоков, тогда количество операций для нахождения $[c_{1+4(t-1)}(t)c_{2+4(t-1)}(t)c_{3+4(t-1)}(t)c_{4+4(t-1)}(t)]_2$ $= [s_{1+4(t-1)}(t)s_{2+4(t-1)}(t)s_{3+4(t-1)}(t)s_{4+4(t-1)}(t)]_2$ равно: $C_{16!}^8 = (16!)!/8!(16!-8)!$, что и составляет степень экспоненциальной сложности данного преобразования.

Далее мы имеем значение $R_{31} = L_{32}$ и значение однозначно восстановленной входной последовательности S-блока, равную C_t. Для восстановления значения раундового ключа k_{32}, который складывается по операции сложения по модулю 2^{32} с правая часть R_{31}: $C_t = (R_{31} \boxplus k_{31}) \bmod 2^{32}$ (генерация раундовых ключей подробно рассматриваться не будет), где t = 1,2,…,32 – число раундов, необходимо 2^{32} операций полного перебора – степень экспоненциальной сложности данного преобразования.

Данная итерация нахождения L_{i-1} и k_i осуществляется еще тридцать один раз. В итоге мы получим значение L_0 (32 бита), R_0 (32 бита) и k_1 (32 бит).

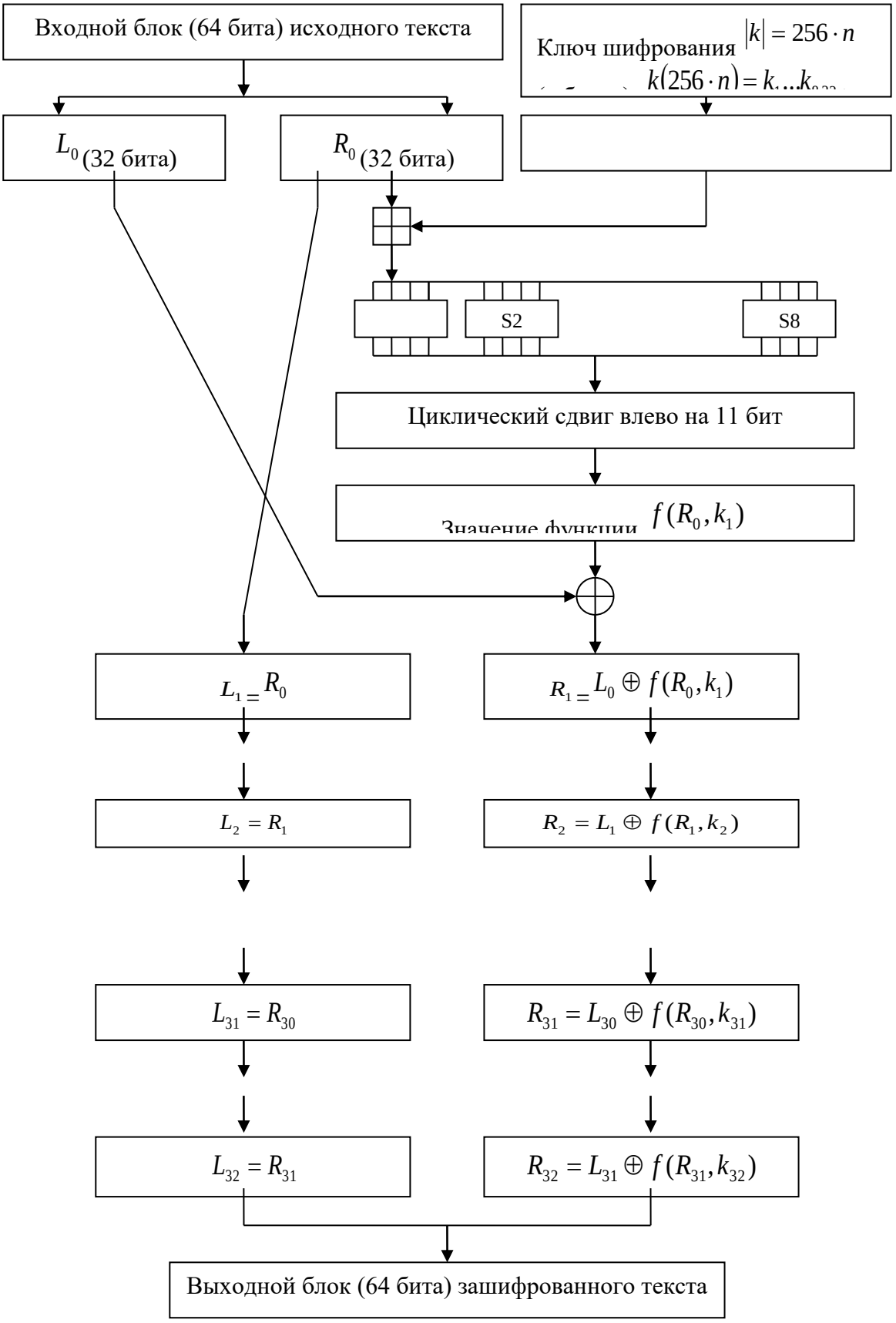

Рис. 1. Схема стандарта алгоритма шифрования ГОСТ 28147-89

Таблица 1. Таблица восьми S-блоков алгоритма шифрования ГОСТ 28147-89

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S_0^1	S_1^1	S_2^1	S_3^1	S_4^1	S_5^1	S_6^1	S_7^1	S_8^1	S_9^1	S_{10}^1	S_{11}^1	S_{12}^1	S_{13}^1	S_{14}^1	S_{15}^1
S_0^2	S_1^2	S_2^2	S_3^2	S_4^2	S_5^2	S_6^2	S_7^2	S_8^2	S_9^2	S_{10}^2	S_{11}^2	S_{12}^2	S_{13}^2	S_{14}^2	S_{15}^2
S_0^3	S_1^3	S_2^3	S_3^3	S_4^3	S_5^3	S_6^3	S_7^3	S_8^3	S_9^3	S_{10}^3	S_{11}^3	S_{12}^3	S_{13}^3	S_{14}^3	S_{15}^3
S_0^4	S_1^4	S_2^4	S_3^4	S_4^4	S_5^4	S_6^4	S_7^4	S_8^4	S_9^4	S_{10}^4	S_{11}^4	S_{12}^4	S_{13}^4	S_{14}^4	S_{15}^4
S_0^5	S_1^5	S_2^5	S_3^5	S_4^5	S_5^5	S_6^5	S_7^5	S_8^5	S_9^5	S_{10}^5	S_{11}^5	S_{12}^5	S_{13}^5	S_{14}^5	S_{15}^5
S_0^6	S_1^6	S_2^6	S_3^6	S_4^6	S_5^6	S_6^6	S_7^6	S_8^6	S_9^6	S_{10}^6	S_{11}^6	S_{12}^6	S_{13}^6	S_{14}^6	S_{15}^6
S_0^7	S_1^7	S_2^7	S_3^7	S_4^7	S_5^7	S_6^7	S_7^7	S_8^7	S_9^7	S_{10}^7	S_{11}^7	S_{12}^7	S_{13}^7	S_{14}^7	S_{15}^7
S_0^8	S_1^8	S_2^8	S_3^8	S_4^8	S_5^8	S_6^8	S_7^8	S_8^8	S_9^8	S_{10}^8	S_{11}^8	S_{12}^8	S_{13}^8	S_{14}^8	S_{15}^8

Затем из конкатенации левой (L_0) и правой (R_0) части по 32 бита осуществляется однозначное восстановление входного блока данных 64 бита.

Таким образом, мы однозначно восстановили открытый текст и исходный ключ шифрования в целом.

Расчет степени экспоненциальной сложности 32-х раундов алгоритма ГОСТ 28147-89 дешифрования шифр текста приведем в виде таблицы 2.

Таблица 2. Степень экспоненциальной сложности алгоритма шифрования ГОСТ

Тип преобразования	Вид преобразования	Степень экспоненциальной сложности
Сложение по *mod* 2^{32}	$C_t = (L_t + K_t) \bmod 2^{32}$	2^{32}
Табличное преобразование S-блоков	$[c_{1+4(t-1)}(t)c_{2+4(t-1)}(t)c_{3+4(t-1)}(t)c_{4+4(t-1)}(t)]_2 = [s_{1+4(t-1)}(t)s_{2+4(t-1)}(t)s_{3+4(t-1)}(t)s_{4+4(t-1)}(t)]_2$	$C_{16!}^8 = (16!)!/8!(16!-8)! \approx 2^{337}$
Побитное сложение по модулю 2: $f(R_{i-1}, k_i)$ и R_i	$L_{i-1} = R_i \oplus f(R_{i-1}, k_i)$	2^{32}
Степень экспоненциальной сложности 1-го раунда	$2^{337}2^{32}2^{32}$	2^{401}
Степень экспоненциальной сложности 32-х раундов	$[2^{32}2^{32}]^{32}2^{337}$	2^{2385}

С использованием в данной методов оценки экспоненциальной сложности можно дать оценку стойкости других алгоритмов шифрования.

Литература

1. Столлингс В. Криптография и защита сетей: принципы и практика. – М. , Изд. дом «Вильямс», 2001. – 672 стр.
2. Венбо Мао. Современная криптография. Теория и практика. – Москва–Санкт-Петербург–Киев: Лори Вильямс, 2005. –768 стр.
3. Алферов А.П., Зубов А.Ю., Кузьмин А.С., Черемушкин А.В. Основы криптографии: Учебное пособие, 1-2-е изд. –М.: Гелиос АРВ, 2002.-480 с

Н.А. Филатов

УПРАВЛЕНИЯ СЕТЕВОЙ БЕЗОПАСНОСТЬЮ В ПРОГРАММНО-ОПРЕДЕЛЯЕМЫХ СЕТЯХ

(Самарский университет)

Сегодня контроллеры OpenFlow [1], [2] работают, преимущественно, как некоторые точки координации, которые передают правила потока от приложений, отправляют запросы конфигурации и исследуют совокупность данных, чтобы получить информацию о состоянии. Так как контроллер взаимодействует со всеми коммутаторами в своей сети или сетевом срезе, он предоставляет средства для распределения скоординированного набора правил потока по сети для оптимизации маршрутов потока и переадресации, а также баланса трафика для повышения эффективности сети.

В области сетевой безопасности OpenFlow может предложить уникальный контроль над любым потоком (или участником потока), считающимся опасным. Приложение OpenFlow, направленное на безопасность может осуществить более сложную логику управления потоками, чем просто остановить или перенаправить поток. Подобные приложения могут включать логику составления правил потока с отслеживанием состояния для реализации сложных процедур помещения на карантин производителя потока, или они могут переносить вредоносное соединение в специальное приложение-ловушку для дальнейшего анализа вредоносной деятельности способом, незаметным для атакующего.

Тем не менее, существуют также и существенные проблемы безопасности, возникающие в OpenFlow и SDN. Например, то, какая политика сетевой безопасности осуществляется в коммутаторах OpenFlow, во всем зависит от того, как текущие приложения OpenFlow отвечают на поступающие запросы потока.

В данных тезисах рассматриваются проблемы определения уровня безопасности посредничества между прикладным уровнем OpenFlow (где должны существовать как приложения безопасности, так и приложения для управления трафиком) и плоскостью данных (где коммутаторы реализуют политики потоков, реализуемые правилами потоков, создаваемых приложениями OpenFlow). В качестве контроллера используется SE-Floodlight. SE-Floodlight расширяет