УВАЖАЕМЫЕ СТУДЕНТЫ! Изучите теоретические сведения к лабораторной работе, выполните практическое задание, дайте ответы на контрольные вопросы.

Результаты работы, фотоотчет, предоставить преподавателю на e-mail: r.bigangel@gmail.com до 23.03.2023.

Требования к отчету:

Отчет предоставляется преподавателю в электронном варианте и должен содержать:

- **название работы, постановку цели, вывод;**
- **ответы на контрольные вопросы, указанные преподавателем.**

При возникновении вопросов по приведенному материалу обращаться по следующему номеру телефона: (072)111-37-59, (Viber, WhatsApp), vk.com: https://vk.com/daykini

ВНИМАНИЕ!!! При отправке работы, не забывайте указывать ФИО студента, наименование дисциплины, дата проведения занятия (по расписанию).

Лабораторная работа №37

Тема: Изучение стандартов шифрования AES и Rjndael.

Цель работы: освоить методы продукционного шифрования.

Описание методов продукционного шифрования

Как перестановочные, так и шифры замены имеют свои преимущества, поэтому безопасную криптосистему можно построить путем их объединения. *Продукционный шифр* — это такой шифр, в котором объединены две или несколько криптографических функции, выполняемых одна за другой. Примером такой системы является шифровальная машина военного времени “Enigma” («Загадка»), в которой ряд кодовых колес переставлял и преобразовывал символы сообщения в кодовые символы. Многие коммерческие шифры основаны на принципе выполнения достаточного количества относительно простых перестановок и преобразований, формирующих безопасную систему.

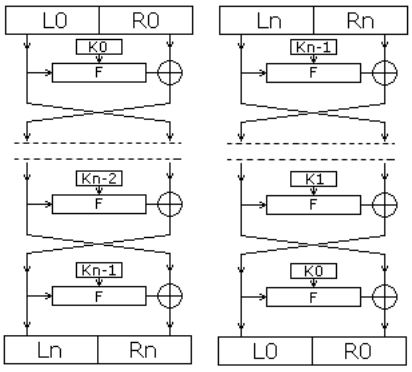
Хорошим примером системы этого типа является стандарт шифрования данных (Data Encryption Standard, DES), который использует последовательность из 16 преобразований и перестановок. Он имеет 56-разрядный ключ, который, хотя и восприимчив к нападениям "в лоб", но все еще очень популярен. Другие примеры продукционных шифров коммерческого использования — Blowfish Брюса Шнейера, TC4 Рональда Райвеста (соавтора криптосистемы RSA) и IDEA (International Data Encryption Algorithm) —

международный алгоритм шифрования данных, который использует 128-разрядный ключ и ряд, состоящий из 8 преобразований и перестановок.

Продукционные шифры обеспечивают очень хороший компромисс между защитой, сложностью и генерацией/распределением ключей. Они продолжают быть популярными и приняты Национальным институтом стандартов и технологий США (NIST, National Institute of Standards and Technology) для замены стандарта DES под названием Advanced Encryption Standard, AES (Расширенный стандарт шифрования).

Сети Фейстеля.

Сеть Фейстеля подразумевает разбиение обрабатываемого блока данных на несколько субблоков (чаще всего - на два), один из которых обрабатывается некоей функцией $f()$ и накладывается на один или несколько остальных субблоков.



а) *б)*

Рис. 1. Структура алгоритмов на основе сети Фейстеля
а) шифрования, б) дешифрования.

Блок открытого текста делится на две равные части (L_0, R_0). Для каждого раунда ($i = \overline{1,n}$ - номер раунда) вычисляется:

$$L_i = R_{i-1} \oplus f(L_{i-1}, K_{i-1})$$
$$R_i = L_{i-1}$$

где f — некоторая функция, а K_{i-1} — ключ i-го раунда.

Ключ раунда является результатом обработки ключа шифрования процедурой расширения ключа, задача которой - получение необходимого количества ключей K_i из исходного ключа шифрования относительно небольшого размера (в настоящее время достаточным для ключа симметричного шифрования считается размер 128 бит). В простейших случаях процедура расширения ключа просто разбивает ключ на несколько фрагментов, которые поочередно используются в раундах шифрования; существенно чаще процедура расширения ключа является достаточно сложной, а ключи K_i зависят от значений большинства бит исходного ключа шифрования.

Результатом выполнения n раундов является L_n, R_n. Но обычно в n-ом

раунде перестановка L_n и R_n не производится, что позволяет использовать ту же процедуру и для расшифрования, просто инвертировав порядок использования раундовой ключевой информации:

$$L_{i-1} = R_i \oplus f(L_i, K_{i-1})$$
$$R_{i-1} = L_i$$

Такая структура алгоритмов шифрования получила свое название по имени Хорста Фейстеля (Horst Feistel) - одного из разработчиков алгоритма шифрования Lucifer и разработанного на его основе алгоритма DES (Data Encryption Standard) - бывшего (но до сих пор широко используемого) стандарта шифрования США. Оба этих алгоритма имеют структуру, аналогичную показанной на рис. 1. Среди других алгоритмов, основанных на сети Фейстеля, можно привести в пример отечественный стандарт шифрования ГОСТ 28147-89, а также другие весьма известные алгоритмы: RC5, Blowfish, TEA, CAST-128 и т.д.

На сети Фейстеля основано большинство современных алгоритмов шифрования - благодаря множеству преимуществ подобной структуры, среди которых стоит отметить следующие:

Алгоритмы на основе сети Фейстеля могут быть сконструированы таким образом, что для зашифрования и расшифрования могут использоваться один и тот же код алгоритма - разница между этими операциями может состоять лишь в порядке применения ключей K_i; такое свойство алгоритма наиболее полезно при его аппаратной реализации или на платформах с ограниченными ресурсами; в качестве примера такого алгоритма можно привести ГОСТ 28147-89.

Алгоритмы на основе сети Фейстеля являются наиболее изученными - таким алгоритмам посвящено огромное количество криптоаналитических исследований, что является несомненным преимуществом как при разработке алгоритма, так и при его анализе.

Существует и более сложные структуры сети Фейстеля. При большом размере блоков шифрования (128 бит и более) реализация сети Фейстеля на 32-разрядных архитектурах может вызвать затруднения, поэтому применяются модифицированные варианты этой конструкции. Обычно используются сети с 4 ветвями. На рисунке показаны наиболее распространенные модификации. Также существуют схемы, в которых длины половинок L_0 и R_0 не совпадают. Они называются несбалансированными.

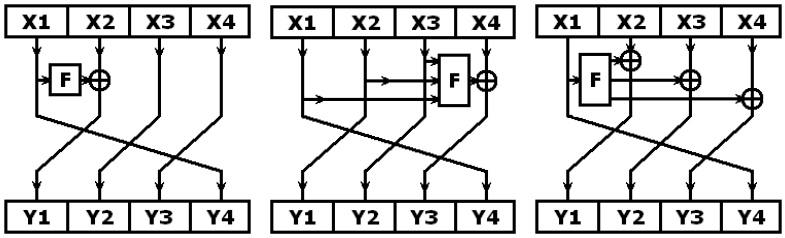


рис. 2. Модификации сети Фейстеля

DES

DES — это стандарт шифрования данных, созданный для американского правительства с целью использования в финансовых транзакциях. Он был разработан на базе криптосистемы Lucifer компании IBM. DES — нелинейный алгоритм. Это означает, что невозможно формировать правильные сообщения простым дополнением зашифрованных текстов (шифрограмм). DES включает только простые подстановки и дополнения, что делает его идеальным для реализации на интегральных схемах. Другое преимущество — для дешифровки DES использует то же самое оборудование, что и для шифрования, только с секциями подключей в обратном порядке.

Кроме шифрования передаваемых данных стандарт DES также используется для шифрования паролей в операционной системе UNIX и для проверки PIN-кодов на кэш-картах ATM.

DES-алгоритм состоит из 16 стандартных конструктивных блоков, которые переставляют и преобразуют 64-разрядные входные блоки в 64-разрядные выходные (рис.3, а). Каждый стандартный конструктивный блок работает с отдельным 48-разрядным ключом, который получается из первоначального ключа. Ключ состоит из 64 разрядов, но 8 из них — разряды проверки на четность, так что фактически ключ имеет длину 56 разрядов. До подачи в первый стандартный конструктивный блок, 64-разрядный ввод подвергается перестановке, а после прохождения через блоки — обратной перестановке.

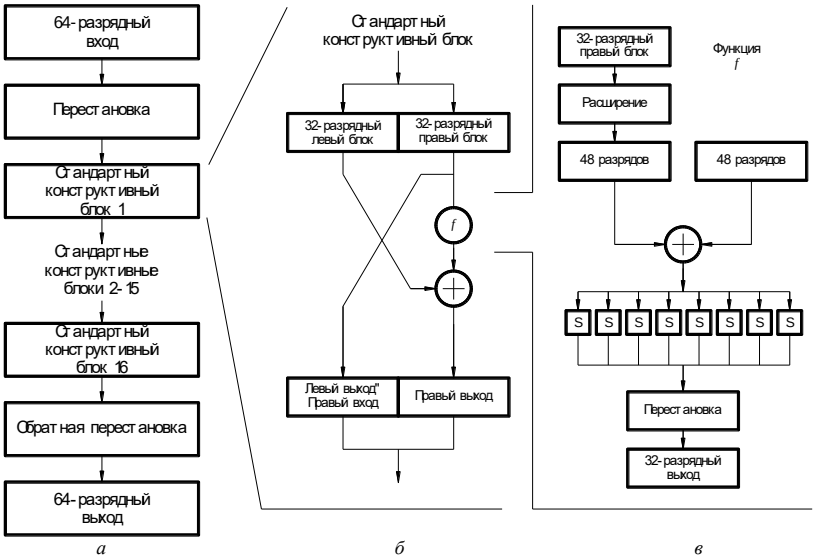


Рис.3. DES-шифрование

Каждый стандартный конструктивный блок преобразует левую и правую части 64 разрядного входа в 64-разрядный выход (рис.3, б). Правая часть входа передается напрямую и принимает участие в формировании левой половины выхода. Правая часть выхода формируется с помощью нелинейной функции из левой и правой частей входа и ключа, предназначенного для данного конкретного блока.

Нелинейная функция детализирована на рис.3,в. 32-разрядный правый блок входа расширяется до 48 битов путем двойного повторения половины его

разрядов и их перестановки. Затем к этому блоку добавляется специфический 48-разрядный ключ. Результат делится на восемь 6-разрядных блоков. Каждый из этих блоков используется как адрес для массива, состоящего из 8 S-элементов. Каждый S-элемент в этом массиве представляет число от 0 до 15, так что выходом каждого S-элемента является 4-разрядное число, поэтому функции S-элементов уменьшают выход до 32 разрядного числа. Функция S-элемента нелинейна, т.е. f(A) + f(B) ≠ f{A + B), а сами S-элементы различны. Перед добавлением к левому 32-разрядному блоку 32-разрядный выход S-элементов снова подвергается перестановке и затем используется для формирования нового 32-разрядного правого блока выхода.

16 секций подключей формируются из 56 значащих разрядов ключа (без учета разрядов проверки на четность) путем их расщепления на две секции по 28 разрядов. Затем эти 28-разрядные секции циклически сдвигаются на один или два разряда и между каждым этапом каждые 48 разрядов соответствующего этапа извлекаются и переставляются, формируя подключи K_1—K_{16}. Если подключи используются в прямом порядке (K_1, K_2, K_3,..., K_{16}), то выполняется прямое шифрование. Если же они используются в обратном порядке (K_{16}, K_{15},..., K_1), то в результате происходит инверсия функции шифрования, т.е. зашифрованный блок преобразуется обратно в соответствующий блок сообщения. Это означает, что для шифрования и дешифровки может использоваться одно и то же оборудование.

DES очень популярен. Его алгоритм опубликован, и машинный код его реализации доступен для многих языков программирования. Лучшей атакой на него остается полный перебор всех 2^{56} ключей, впрочем, поскольку шифрование дополнения сообщения с дополнением ключа дает дополнение полей шифрограммы, мы можем обойтись лишь половиной числа переборов (2^{55}).

Для работы с DES требуются весьма мощные компьютеры. В течение ряда лет полный перебор в течение нескольких часов был возможен только для очень богатых организаций, но прогресс в изготовлении микропроцессоров теперь вкладывает эти возможности в руки любой организации, позволяя экономить десятки миллионов долларов на использовании чужих секретов. Двойная шифровка сообщений приводит к небольшому увеличению защиты из-за возможности применения так называемого "встречающегося посередине" (meet in the middle) нападения. Однако шифрование с одним ключом, дешифрование — со вторым и повторное шифрование— с третьим, которое называется тройным DES-шифрованием, увеличивает число ключей, требующих проверки, приблизительно до 2^{80}, что снова можно считать безопасным.

Расширенный стандарт шифрования (AES)

Как отмечалось ранее, DES-стандарту присуща повышенная сложность обрабатывающего оборудования, что привело к практическим предложениям по его пересмотру. После открытого приглашения к разработке алгоритмов, на которое было получено 21 предложение от 11 стран, и двухлетней процедуры

их оценки Национальный институт стандартов и технологий США (NIST) выбрал замену для DES в форме Расширенного стандарта шифрования (AES). Для него был принят шифр, строящийся по алгоритму Райндала (Rijndael). Это сложный блочный продукционный шифр, который может быть эффективно реализован. Шифр Райндала был разработан фламандскими исследователями Джоан Даймен (Joan Daemen) и Винсентом Райменом (Vincent Rijmen). В стандарте определяется три размера ключей: 128, 192 и 256 разрядов.

Также как DES, алгоритм состоит из нескольких раундов, точное число которых зависит от размера ключа, состоящих из перестановок и добавления подключа. Хотя алгоритм Райндала может работать и на других размерах блоков, стандарт AES определяет размер блока, равным 128 разрядов. Этот блок разбивается на 168-разрядных байтов, размещаемых в массивах размером 4 строки × 4 столбца. Каждый раунд обработки состоит из четырех этапов: нелинейного преобразования S-элемента, которое реализуется как отображение одного байта в другой, перестановки строк, перемешивания столбцов и, наконец, добавления подключа текущего раунда.

Одним из первых приложений AES будет шифрование информации, посылаемой в эфир, в новой системе мобильного радио 3-го поколения CDMA2000.