Лабораторная работа 8

«Алгоритмы шифрования DES и AES»

Теоретическая часть

Основные сведения

Одной из наиболее известных криптографических систем с закрытым ключом является *DES – Data Encryption Standard*. Эта система первой получила статус государственного стандарта в области шифрования данных. Она разработана специалистами фирмы IBM и вступила в действие в США 1977 году. Алгоритм *DES* широко использовался при хранении и передаче данных между различными вычислительными системами; в почтовых системах, в электронных системах чертежей и при электронном обмене коммерческой информацией. Стандарт *DES* реализовывался как программно, так и аппаратно. Предприятиями разных стран был налажен массовый выпуск цифровых устройств, использующих *DES* для шифрования данных. Все устройства проходили обязательную сертификацию на соответствие стандарту.

Несмотря на то, что уже некоторое время эта система не имеет статуса государственного стандарта, она по-прежнему широко применяется и заслуживает внимания при изучении блочных шифров с закрытым ключом.

Длина ключа в алгоритме *DES* составляет 56 бит. Именно с этим фактом связана основная полемика относительно способности *DES* противостоять различным атакам. Как известно, любой блочный шифр с закрытым ключом можно взломать, перебрав все возможные комбинации ключей. При длине ключа 56 бит возможны 2^{56} разных ключей. Если компьютер перебирает за одну секунду 1 000 000 ключей (что примерно равно 2^{20}), то на перебор всех 2^{56} ключей потребуется 2^{36} секунд или чуть более двух тысяч лет, что, конечно, является неприемлемым для злоумышленников.

В 2001 году после специально объявленного конкурса в США был принят новый стандарт на блочный шифр, названный *AES (Advanced Encryption Standard)*, в основу которого был положен шифр *Rijndael*, разработанный бельгийскими специалистами. Этот шифр рассматривается в конце лекции.

Основные параметры *DES*: размер блока 64 бита, длина ключа 56 бит, количество раундов – 16. *DES* является классической сетью Фейштеля с двумя ветвями. Алгоритм преобразует за несколько раундов 64-битный входной блок данных в 64-битный выходной блок. Стандарт *DES* построен

на комбинированном использовании перестановки, замены и гаммирования. Шифруемые данные должны быть представлены в двоичном виде.

Шифрование

Общая структура *DES* представлена на рис. 4.1. Процесс шифрования каждого 64-битового блока исходных данных можно разделить на три этапа:

1. начальная подготовка блока данных;
2. 16 раундов "основного цикла";
3. конечная обработка блока данных.

На первом этапе выполняется *начальная перестановка* 64-битного исходного блока текста, во время которой биты определенным образом переупорядочиваются.

На следующем (основном) этапе блок делится на две части (ветви) по 32 бита каждая. Правая ветвь преобразуется с использованием некоторой функции F и соответствующего *частичного ключа*, получаемого из основного ключа шифрования по специальному алгоритму преобразования ключей. Затем производится обмен данными между левой и правой ветвями блока. Это повторяется в цикле 16 раз.

Наконец, на третьем этапе выполняется перестановка результата, полученного после шестнадцати шагов основного цикла. Эта перестановка обратна начальной перестановке.

Рис. 4.1. Общая схема DES

Рассмотрим более подробно все этапы криптографического преобразования по стандарту *DES*.

На первом этапе 64-разрядный блок исходных данных подвергается начальной перестановке. В литературе эта операция иногда называется "забеливание" – whitening. При начальной перестановке биты блока данных определенным образом переупорядочиваются. Эта операция придает некоторую "хаотичность" исходному сообщению, снижая возможность использования криптоанализа статистическими методами.

Одновременно с начальной перестановкой блока данных выполняется начальная перестановка 56 бит ключа. Из рис. 4.1. видно, что в каждом из раундов используется соответствующий 48-битный частичный ключ K_i. Ключи K_i получаются по определенному алгоритму, используя каждый из битов начального ключа по нескольку раз. В каждом раунде 56-битный ключ делится на две 28-битовые половинки. Затем половинки сдвигаются влево на один или два бита в зависимости от номера раунда. После сдвига определенным образом выбирается 48 из 56 битов. Так как при этом не только выбирается подмножество битов, но и изменяется их порядок, то эта операция называется "перестановка со сжатием". Ее результатом является набор из 48 битов. В среднем каждый бит исходного 56-битного ключа используется в 14 из 16 подключей, хотя не все биты используются одинаковое количество раз.

Далее выполняется основной цикл преобразования, организованный по сети Фейштеля и состоящий из 16 одинаковых раундов. При этом в каждом раунде (рис. 4.2) получается промежуточное 64-битное значение, которое затем обрабатывается в следующем раунде.

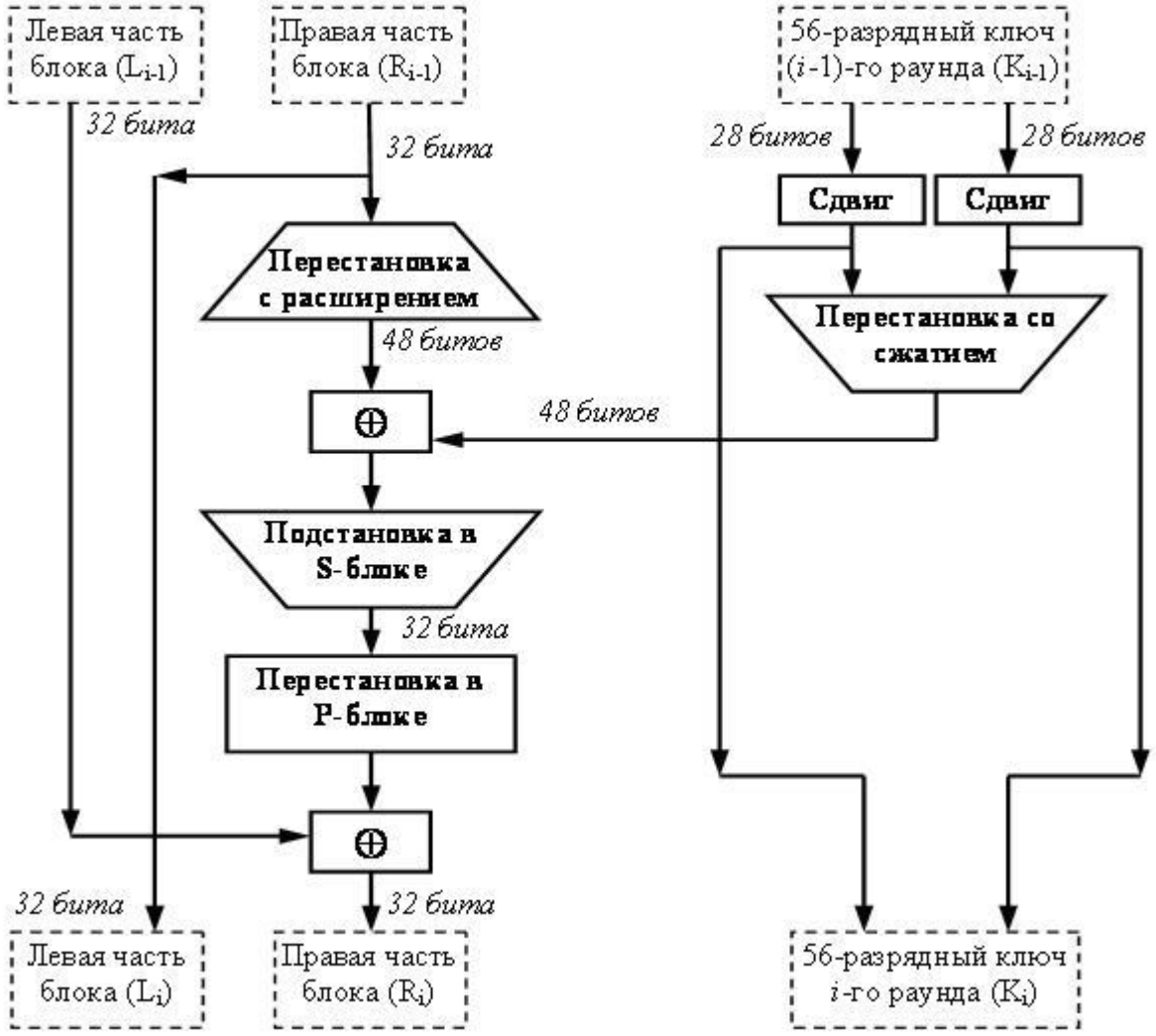

Рис. 4.2. Структура одного раунда DES

Левая и правая ветви каждого промежуточного значения обрабатываются как отдельные 32-битные значения, обозначенные L и R.

Вначале правая часть блока R_i расширяется до 48 битов, используя таблицу, которая определяет перестановку плюс расширение на 16 битов. Эта операция приводит размер правой половины в соответствие с размером ключа для выполнения операции XOR. Кроме того, за счет выполнения этой операции быстрее возрастает зависимость всех битов результата от битов исходных данных и ключа (это называется "лавинным эффектом"). Чем сильнее проявляется лавинный эффект при использовании того или иного алгоритма шифрования, тем лучше.

После выполнения перестановки с расширением для полученного 48-битного значения выполняется операция XOR с 48-битным подключом K_i. Затем полученное 48-битное значение подается на вход блока подстановки S (от англ. *Substitution* - подстановка), результатом которой является 32-битное значение. Подстановка выполняется в восьми блоках подстановки или восьми S-блоках (S-boxes). При выполнении этой операции 48 битов данных

делятся на восемь 6-битовых подблоков, каждый из которых по своей таблице замен заменяется четырьмя битами. Подстановка с помощью S-блоков является одним из важнейших этапом *DES*. Таблицы замен для этой операции специально спроектированы специалистами так, чтобы обеспечивать максимальную безопасность. В результате выполнения этого этапа получаются восемь 4-битовых блоков, которые вновь объединяются в единое 32-битовое значение.

Далее полученное 32-битовое значение обрабатывается с помощью перестановки P (от англ. *Permutation* – перестановка), которая не зависит от используемого ключа. Целью перестановки является максимальное переупорядочивание битов такое, чтобы в следующем раунде шифрования каждый бит с большой вероятностью обрабатывался другим S-блоком.

И, наконец, результат перестановки объединяется с помощью операции XOR с левой половиной первоначального 64-битового блока данных. Затем левая и правая половины меняются местами, и начинается следующий раунд.

После шестнадцати раундов шифрования выполняется конечная перестановка результата. Эта перестановка инверсна (обратна) начальной перестановке.

После выполнения всех указанных шагов блок данных считается полностью зашифрованным и можно переходить к шифрованию следующего блока исходного сообщения.

Расшифрование

Как известно, криптографическая система должна позволять не только зашифровать, но и расшифровать сообщения. Можно было бы ожидать, что процесс расшифрования по *DES* сильно запутан. Однако разработчики так подобрали различные компоненты стандарта, чтобы для зашифрования и расшифрования использовался один и тот же алгоритм. При расшифровании на вход алгоритма подается зашифрованный текст. Единственное отличие состоит в обратном порядке использования частичных ключей K_i. K_{16} используется на первом раунде, K_1 — на последнем раунде.

После последнего раунда процесса расшифрования две половины выхода меняются местами так, чтобы вход заключительной перестановки был составлен из R_{16} и L_{16}. Выходом этой стадии является незашифрованный текст.

Двухкратный DES и атака "встреча посередине"

В настоящее время основным недостатком DES считается маленькая длина ключа. Простейшим способом усложнения процесса криптоанализа является использование *двухкратного шифрования* с помощью одного и того же алгоритма с разными ключами. Если M – сообщение, K_1, K_2 – ключ, f –

процесс шифрования по *DES*, а E – зашифрованное сообщение, то можно записать

$E=f(f(M,K_1),K_2)$,

то есть сначала блок шифруется одним ключом, затем получившийся шифротекст шифруется вторым ключом. Расшифрование проводится в обратном порядке (f^{-1} – расшифрование по *DES*):

$E=f^{-1}(f^{-1}(E,K_2),K_1)$

В этом случае длина ключа равна 56 * 2 = 112 бит, поэтому для обнаружения двойного ключа, которым зашифрован блок, потребуется в общем случае 2^{112} попыток.

Исследовав эту проблему, американские ученые Меркл и Хеллман придумали способ проведения атаки по открытому тексту, который требует проведения не 2^{112} попыток, а 2^{57}. (Меркл и Хеллман предложили эту схему против *DES*, но можно сделать обобщение на все блочные алгоритмы.)

Этот вариант атаки называется *атака "встреча посередине"*. Он основан на следующем свойстве алгоритма. Мы имеем

$E=f(f(M,K_1),K_2)$

где M – сообщение, K_1, K_2 – ключ, f – процесс шифрования по *DES*, а E – зашифрованное сообщение.

Тогда $X=f(M,K_1)=f^{-1}(E,K_2)$.

Атака состоит в следующем. Требуется, чтобы атакующий знал несколько пар "незашифрованный текст - соответствующий ему зашифрованный текст" (M, E). В этом случае вначале шифруется M для всех возможных 2^{56} значений K_1. Этот результат запоминается в памяти ЭВМ. Запомненные данные упорядочиваются по значению X. Следующий шаг состоит в дешифровании E, с применением всех возможных 2^{56} значений K_2. Для каждого выполненного дешифрования ищется равное ему значение в первой таблице. Если такое значение найдено, то считается, что эти ключи могут быть правильными, и они проверяются для следующей известной пары "незашифрованный текст, зашифрованный текст". Максимальное количество попыток шифрования, которое, возможно, придется предпринять, равно $2*2^n$, или 2^{n+1} (где n – длина ключа в каждом из этапов шифрования; для *DES* n равно 56).

Название "встреча посередине" дано атаке по той причине, что с одной стороны выполняется шифрование, с другой – расшифрование, и полученные посередине результаты сравниваются.

Для проведения атаки "встреча посередине" нужен большой объем памяти: 2^n блоков (где n – длина ключа). Для *DES*, в котором используется 56-битовый ключ потребуется 2^{56} 64-разрядных блоков памяти. Это составляет 2^{62} байт

или 2^{22} Тбайт. Такой объем памяти достаточно трудно пока еще себе представить, кроме того, для проведения операций поиска в таком огромном массиве потребуется соответствующее время. Несмотря на это, двойное шифрование *DES* практически никогда не использовалась.

Трехкратный DES

В целях противодействия атаке "встреча посередине" было предложено использовать тройное шифрование с двумя ключами (рис. 4.3).

Рис. 4.3. Шифрование тройным DES с двумя ключами

Режимы работы блочных алгоритмов

Блочные шифры могут использоваться для выполнения различных задач. Поэтому для любого симметричного блочного алгоритма шифрования определено несколько режимов его применения. Каждый из режимов имеет свои особенности и сферы применения.

Пусть имеется некоторый блочный шифр, который выполняет преобразование f исходного блока данных X с помощью ключа K в зашифрованный блок Y:

Y=f(X,K)

Рассмотрим некоторые возможные режимы выполнения преобразования f.

Простейшим режимом является **режим простой поблочной замены**. Специалистами этот режим называется **ECB - Electronic CodeBook**, что переводится как "электронная кодовая книга". В этом режиме каждый блок исходных данных шифруется независимо от остальных блоков, с применением одного и того же ключа шифрования. Если сообщение длиннее, чем длина блока соответствующего алгоритма, то оно разбивается на блоки X_1, X_2,..., X_n соответствующей длины, причем последний блок дополняется в случае необходимости фиксированными значениями. Каждый блок шифруется блочным шифром:

$Y=f(X_i,K)$ для всех i от 1 до n

В результате шифрования всех блоков исходных данных Xi получается зашифрованное сообщение Y=Y1, Y2,..., Yn. Расшифрование выполняется по правилу

$X=f^{-1}(Y_i,K)$ для всех i от 1 до n

Из определения режима *ECB* следует, что расшифрование сообщения можно производить, выбирая блоки шифротекста в произвольном порядке. Такой режим удобен во многих реальных ситуациях, в частности для обработки файлов с произвольным доступом. Например, в режиме *ECB* можно работать с зашифрованной базой данных при условии, что каждая запись представляет собой отдельный блок данных и зашифрована отдельно от остальных.

Для устранения недостатков *ECB* при передаче нескольких блоков данных может использоваться режим **CBC (Chipher Block Chaining) – режим сцепления блоков шифра**.

Преобразование в режиме *CBC* выполняется следующим образом: каждый блок открытого текста складывается по модулю 2 с результатом шифрования предыдущего блока. Таким образом, результаты шифрования предыдущих блоков влияют на шифрование следующих блоков. Математически операция шифрования в режиме *CBC* описывается следующей формулой:

$$Y = f((X_i \oplus Y_{i-1}), K) \text{для всех } i \text{ от } 1 \text{ до } n$$

То есть перед шифрованием очередного блока над открытым текстом и результатом шифрования предыдущего блока выполняется операция "сумма по модулю 2". Когда блок открытого текста зашифрован, он сохраняется в памяти шифрующего устройства, например, в регистре обратной связи. Перед шифрованием следующего блока данных, он подвергается операции "сумма по модулю 2" вместе с регистром обратной связи и только после этого шифруется. Полученный зашифрованный блок снова сохраняется в регистре обратной связи и используется при шифровании следующего блока входных данных и так далее до конца сообщения. Блок Y_0 должен быть сформирован перед шифрованием первого блока исходных данных. Он называется вектором инициализации и используется для сложения по модулю 2 с первым блоком входных данных.

В результате использования обратной связи шифрование каждого блока зависит от всех предыдущих блоков.

Зашифрованное сообщение можно расшифровать следующим образом:

$$X_i = Y_{i-1} \oplus f^{-1}(Y_i, K) \text{для всех } i \text{ от } 1 \text{ до } n$$

Блок шифротекста сначала сохраняется в регистре обратной связи, а затем расшифровывается как обычно. Далее расшифровывается следующий блок и

подвергается операции "сумма по модулю 2" с регистром обратной связи. И так выполняется до конца сообщения.

Даже если все блоки исходных данных Xi идентичны, шифротекст будет состоять из различных блоков Y. Этот режим предпочтителен при шифровании сообщений, размер которых превышает размер блока. Однако два одинаковых сообщения будут шифроваться одинаково. Для того чтобы этого избежать, необходимо использовать разные векторы инициализации при каждом шифровании. Векторы инициализации необходимы также для расшифрования данных, поэтому их нужно или пересылать адресату вместе с зашифрованным сообщением, или договориться о каком-либо совместном формировании псевдослучайных векторов инициализации.

Расшифровать сообщение, зашифрованное в режиме CBC, можно только последовательно, начиная с первого блока.

Практическая часть

Задание для выполнения

1. Сложите по модулю 2:
   - двоичные числа 10101100 и 11001010 ;
   - десятичные числа 15 и 10 ;
   - шестнадцатеричные числа 0B5 и 37.

Примечание: десятичные и шестнадцатеричные числа необходимо сначала перевести в двоичный вид.

2. Сложите по модулю 2^8:
   - двоичные числа 10101100 и 11001010 ;
   - десятичные числа 155 и 100 ;
   - шестнадцатеричные числа 0B5 и 37.

Примечание: десятичные числа необходимо сначала перевести в двоичный вид.

3. Выполните операцию циклического сдвига:
   - влево на 5 разрядов для двоичного числа 10101100 ;
   - вправо на 4 разряда для шестнадцатеричного числа 9E ;
   - вправо на 2 разряда для шестнадцатеричного числа 55.

Примечание: шестнадцатеричные числа необходимо сначала перевести в двоичный вид.

Вопросы для защиты

1. Какой шифр называют комбинированным или композиционным шифром?

2. Какие факторы влияют на стойкость блочного алгоритма шифрования?

3. Что представляет собой сеть Фейштеля?

4. Перечислите основные параметры алгоритмов симметричного шифрования *DES, AES*.

5. Какие операции используются в блочных алгоритмах шифрования *DES, AES*?

6. В чем заключается атака "встреча посередине"?

7. Каков принцип использования тройного *DES*?

8. Каким образом используется блочный алгоритм шифрования в режиме простой замены (*ECB*)?

9. Каковы недостатки режима *ECB*?

10. Каковы особенности использования алгоритма шифрования в режиме сцепления блоков шифра (*CBC*)?

11. Каковы недостатки режима *CBC*?