Лабораторная работа № 4

Режимы работы блочных шифров. Схемы кратного шифрования

Цель работы

Изучить и реализовать режимы работы блочных шифров и схемы кратного шифрования для симметричных алгоритмов шифрования DES и ГОСТ 28147-89.

Режимы работы блочных шифров

Режимами шифрования называют различные алгоритмы обработки данных, построенные на основе базового режима ECB. Криптографическая стойкость этих алгоритмов определяется в основном стойкостью базового режима. Однако особенности различных режимов шифрования позволяют использовать блочный шифр для решения различных криптографических задач.

ECB – режим электронной шифровальной книги (простая замена)

Простейшим режимом является ***режим электронной шифровальной книги*** (*ECB – Electronic Codebook*), когда открытый текст обрабатывается блоками по 64 бита и каждый блок шифруется с одним и тем же ключом [1]. Термин ***шифровальная книга*** объясняется тем, что при заданном ключе каждый 64-битный блок открытого текста представляется уникальным блоком шифрованного текста. Если длина сообщения превышает 64 бита, то оно разделяется на 64-битные блоки с добавлением при необходимости заполнителей к последнему блоку.

Уравнение шифрования режима ECB: $\boxed{C_i = E_k(P_i)}$, где $i = \overline{1,N}$, P_i, – входные блоки открытого текста, C_i – соответствующий шифрованный текст, E_k – алгоритм шифрования с использованием ключа k.

Уравнение дешифрования режима ECB: $\boxed{P_i = D_k(C_i)}$, где $i = \overline{1,N}$, $D_k = E_k^{-1}$ – алгоритм дешифрования с использованием ключа k.

Режим ECB имеет следующие особенности:

1. Замены и перестановки отдельных блоков в шифротексте не нарушают корректности расшифрования остальных блоков текста.
2. Шифрование на одном ключе одинаковых блоков открытого текста даёт одинаковые блоки шифротекста.
3. В силу хорошего перемешивания информации шифрующими подстановками (это общее свойство блочных шифров) каждый из 64 бит выходного блока P_i может быть искажён с вероятностью $\frac{1}{2}$ при искажении лишь одного случайно выбранного бита входного блока C_i. Единичная ошибка в блоке C_i порождает, в среднем, n ошибок внутри шифроблока P_i. На следующие шифроблоки ошибка не распространяется.

При искажении бит выходного блока P_i соответствующий блок C_i расшифровывается некорректно, а остальные блоки расшифровываются верно. Но если бит шифротекста случайно потерян или добавлен, то в силу произошедшего сдвига весь последующий текст расшифровывается некорректно. Для локализации последствий сдвига следует предусмотреть средство контроля границ блоков.

Первые две особенности позволяют активному противнику, контролирующему линию связи, защищаемую шифром в режиме ECB, наблюдать частоты появления отдельных блоков и сообщений. В определённых условиях он может генерировать ложные сообщения, не зная ни ключа, ни алгоритма шифрования, даже если сообщения содержат метки времени. В силу этих серьезных недостатков режим ECB не используется для шифрования длинных сообщений. В этом режиме шифруются лишь короткие сообщения вспомогательного характера: пароли, сеансовые ключи и т.п.

CBC – режим сцепления шифрованных блоков

Технология, свободная от недостатков режима ECB, должна в случае повторения в сообщении уже встречавшегося ранее блока открытого текста генерировать блок шифрованного текста, отличный от сгенерированного ранее. Проще всего добиться этого с помощью ***режима сцепления шифрованных блоков*** (*CBC – Cipher Block Chaining*, см. рисунок 1) [1].

Уравнение шифрования режима CBC: $\boxed{C_i = E_k(P_i \oplus C_{i-1})}$, $i = \overline{1,N}$, где $C_0 = IV$ – вектор инициализации (начальный вектор, синхропосылка).

Уравнение дешифрования режима CBC: $\boxed{P_i = D_k(C_i) \oplus C_{i-1}}$, $i = \overline{1,N}$.

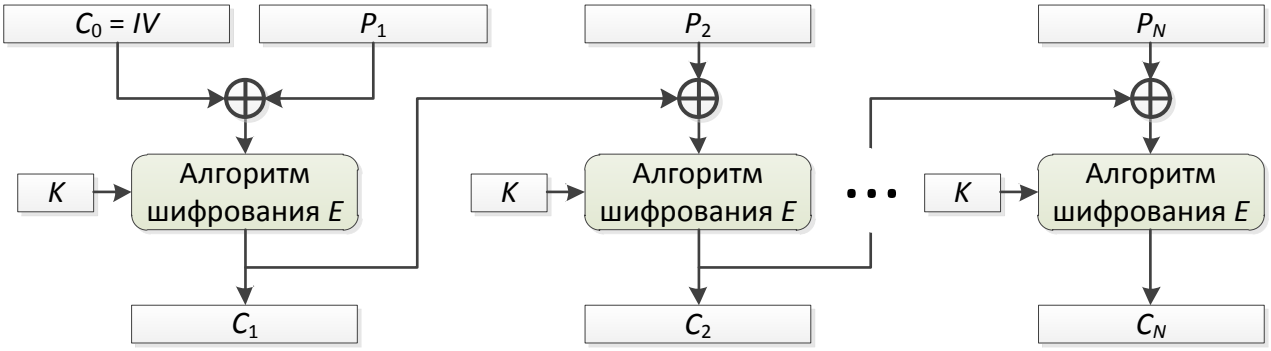

Рисунок 1 – Схема шифрования в режиме CBC

Начальный вектор может передаваться в линию связи как в открытом, так и в шифрованном виде (в частности, с помощью режима ECB). Таким образом, если исходный текст состоял из N блоков, то результат шифрования в режиме CBC будет содержать $N + 1$ блоков. Однако важно избегать повторения синхропосылки в разных сообщениях, шифруемых одинаковым ключом. Это затрудняет атаку на шифротекст, основанную на наличии стандартов в начале сообщения. В качестве синхропосылки используется некоторая строка случайных байт либо метка времени.

Искажение одного бита в блоке P_i влечёт искажение примерно половины бит во всех блоках шифротекста, начиная с C_i. Для расшифрования это несущественно, так как восстановленный текст содержит ту же единственную ошибку.

Искажение k-го бита в блоке C_i, (такие искажения возможны из-за шумов в линиях связи или сбоев в устройствах хранения) влечёт искажение около половины бит в блоке P_i, и k-го бита в блоке P_{i+1}. Следующие блоки расшифровываются корректно (самовосстанавливаются). В то же время, режим CBC совершенно неустойчив к ошибкам синхронизации.

Симметричный алгоритм шифрования в режиме CBC используется для поблочной передачи данных общего назначения и аутентификации.

CFB – обратная связь по шифротексту (гаммирование с обратной связью)

В некоторых практических ситуациях требуется шифровать символы поступающего потока, не дожидаясь, когда сформируется целый блок данных. В таких случаях удобен режим CFB (*Cipher Feedback*), в котором блоки открытого и шифрованного текста имеют длину j бит, где j – параметр режима, $1 \leq j \leq 64$. Обозначим такой режим шифрования CFB-j [1]. В связи с байтным представлением информации параметр j выбирается, как правило, равным восьми.

На входе функции шифрования размещается 64-битный регистр сдвига, в котором изначально размещается некоторое значение инициализационного вектора (IV). Крайние слева (старшие) j бит этого значения связываются операцией XOR с первой порцией открытого текста P_1, в результате чего получается первая порция шифрованного текста C_1, который подаётся на линию передачи данных. Содержимое регистра сдвига смещается влево на j бит, а в крайние справа (младшие) j бит помещается значение C_1. Затем весь процесс повторяется до тех пор, пока не будут зашифрованы все элементы открытого текста (см. рисунок 2) [3].

Алгоритм шифрования в режиме CFB:

1. $I_1 = IV$.
2. $O_i = E_k(I_i), \quad i = \overline{1,N}$.
3. $C_i = P_i \oplus St_j(O_i), \quad i = \overline{1,N}$.
4. $I_i = Ml_{64-j}(I_{i-1}) \,||\, C_{i-1}, \quad i = \overline{2,N}$,

где $St_j(X)$ – крайние слева (старшие) j бит блока X, $Ml_j(X)$ – крайние справа (младшие) j бит блока X, $||$ – операция конкатенации битовых строк.

Алгоритм дешифрования в режиме CFB:

1. $I_1 = IV$.
2. $O_i = E_k(I_i), \quad i = \overline{1,N}$.
3. $P_i = C_i \oplus St_j(O_i), \quad i = \overline{1,N}$.
4. $I_i = Ml_{64-j}(I_{i-1}) \,||\, C_{i-1}, \quad i = \overline{2,N}$.

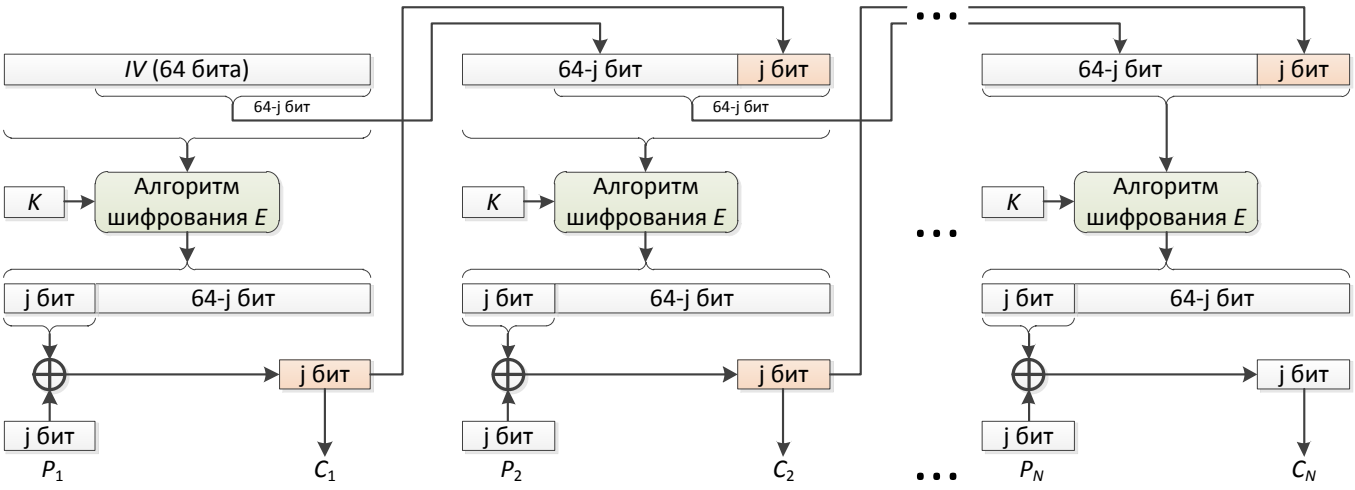

Рисунок 2 – Схема шифрования в режиме CFB

Как и в режиме CBC, синхропосылка может передаваться в линию связи в открытом виде. Однако необходимо исключить повторение синхропосылки в различных сообщениях, шифруемых одинаковым ключом.

Искажение одного бита в блоке P_i влечёт искажение одного бита в C_i и в среднем половины бит во всех блоках шифротекста, начиная с C_{i+1}, но при расшифровании получается открытый текст с той же единственной ошибкой.

Искажение k-го бита в блоке C_i влечёт искажение k-го бита в блоке P_i. Затем ошибка поступает в регистр состояний и искажает в среднем половину бит в каждом из следующих l блоков, где $\left[\frac{64}{j}\right] \leq l$. В дальнейшем блоки расшифровываются корректно.

Режим CFB самостоятельно восстанавливается после ошибок синхронизации.

Симметричный алгоритм шифрования в режиме CFB используется для потоковой передачи данных общего назначения и аутентификации.

OFB – обратная связь по выходу (гаммирование или внутренняя обратная связь)

Блочный шифр в режиме OFB можно рассматривать как синхронный шифр гаммирования, обрабатывающий j-битные блоки открытого и шифрованного текста (обозначим такой режим OFB-j) [1].

Режим обратной связи по выходу (*OFB – Output Feedback*) во многом подобен режиму CFB. В режиме OFB в регистр сдвига подаётся значение, получаемое на выходе функции шифрования (см. рисунок 3), а в режиме CFB в этот регистр подаётся порция шифрованного текста [3].

Алгоритм шифрования в режиме OFB:

1. $I_1 = IV$.
2. $O_i = E_k(I_i), \qquad i = \overline{1,N}$.
3. $C_i = P_i \oplus St_j(O_i), \qquad i = \overline{1,N}$.
4. $I_i = Ml_{64-j}(I_{i-1}) \,||\, St_j(O_{i-1}), \quad i = \overline{2,N}$,

где $St_j(X)$ – крайние слева (старшие) j бит блока X, $Ml_j(X)$ – крайние справа (младшие) j бит блока X, $||$ – операция конкатенации битовых строк.

Алгоритм дешифрования в режиме OFB:

1. $I_1 = IV$.
2. $O_i = E_k(I_i), \qquad i = \overline{1,N}$.
3. $P_i = C_i \oplus St_j(O_i), \qquad i = \overline{1,N}$.
4. $I_i = Ml_{64-j}(I_{i-1}) \,||\, St_j(O_{i-1}), \quad i = \overline{2,N}$.

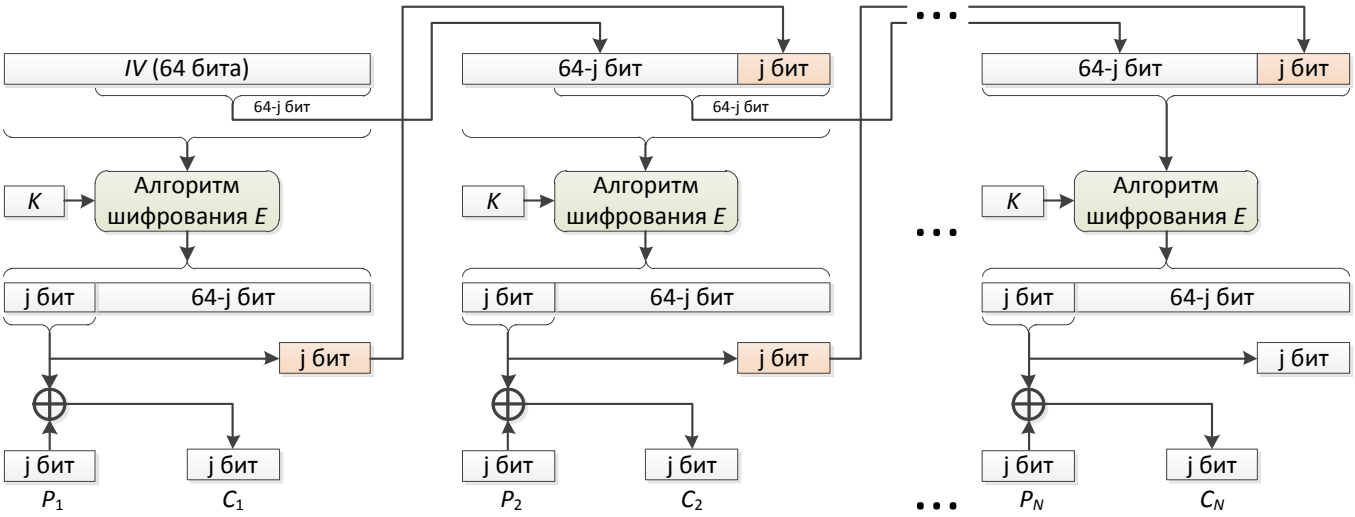

Рисунок 3 – Схема шифрования в режиме OFB

Синхропосылка может передаваться в линию связи в открытом виде, но необходимо исключить повторение синхропосылки в разных сообщениях, шифруемых одинаковым ключом.

При использовании режима OFB чрезвычайно важно сохранять синхронизацию. Для этого необходимо предусмотреть средство контроля над синхронизацией и средство восстановления синхронизации в случае её потери.

В режиме OFB ошибки не распространяются, что является позитивным при передаче оцифрованных речевых сигналов или видеоизображений.

Симметричный алгоритм шифрования в режиме работы OFB используется для потоковой передачи данных по каналам с помехами (например, по спутниковой связи).

Другие режимы шифрования

Разработка других режимов шифрования стимулировалась стремлением устранить некоторые недостатки четырёх основных режимов [2].

BC – сцепление блоков. Этот режим задаётся следующим равенством:

$$\boxed{C_i = E_k\left(P_i \oplus \sum_{j=0}^{i-1} C_j\right)}, \qquad i = \overline{1,N},$$

где $C_0 = IV$ – вектор инициализации. Основной недостаток режима BC заключается в том, что при расшифровании единичная ошибка в шифротексте влечёт некорректное расшифрование всех последующих блоков шифротекста.

PCBC – сцепление блоков шифротекста с распространением ошибки. Режим задаётся следующим равенством: $\boxed{C_i = E_k\left(P_i \oplus C_{i-1} \oplus P_{i-1}\right)}$, $i = \overline{1,N}$, где $C_0 = IV$ и $P_0 = IV'$ – векторы инициализации.

Этот режим используется в протоколе Kerberos 4 для выполнения за один проход и шифрования, и проверки целостности. В режиме PCBC единичная ошибка в шифротексте влечёт некорректное расшифрование всех последующих блоков шифротекста, что используется для проверки целостности сообщений. Однако если проверка целостности охватывает лишь завершающий отрезок текста, то могут остаться незамеченными перестановки пары шифроблоков в начале текста. Это подозрительное свойство заставило разработчиков отказаться от данного режима в пользу CBC в следующей версии протокола Kerberos.

OFBNLF – нелинейная обратная связь по выходу. Этот режим наследует некоторые свойства режимов OFB и ECB: $\boxed{C_i = E_{k_i}(P_i),\ k_i = E_k(k_{i-1})}$, $i = \overline{1,N}$, где k_0 – вектор инициализации.

Единичная ошибка в шифротексте распространяется только на один блок открытого текста, однако требуется поддержка синхронизации.

Скорость обработки информации определяется не только скоростью шифрования базовым алгоритмом, но и скоростью обновления текущего значения ключа.

Кратное шифрование

Криптоаналитикам не удалось разработать практически приемлемый метод дешифрования алгоритма DES, который был бы лучше метода полного перебора ключей. Вместе с тем короткая длина ключа алгоритма DES не позволяла рассматривать его как надёжное средство защиты информации. Это стимулировало криптографов заняться построением блочного шифра с длинным ключом, использующего в качестве базового элемента алгоритм DES.

Один из методов – многократное шифрование с использованием базового алгоритма. Этот метод применим к любому симметричному блочному шифру, однако его использование приводит к уменьшению скорости шифрования (либо требуется больше аппаратных ресурсов) в соответствующее число раз. Кроме того, важно, чтобы множество шифрующих подстановок не являлось группой (для алгоритма DES это доказано), иначе кратное шифрование сводится к однократному.

Двойное шифрование

Простейшая схема кратного шифрования [1] – это ***двойное шифрование*** с помощью двух шифрующих подстановок с независимыми ключами: $\boxed{C = E_{k_2}\left(E_{k_1}(P)\right)}$.

Заметим, что при последовательном использовании алгоритма DES дважды с двумя разными ключами будет получено отображение, отличное от всех тех, которые получаются при однократном применении алгоритма DES, т.е. $\neg\exists k_3 : E_{k_2}\left(E_{k_1}(P)\right) = E_{k_3}(P)$. Здесь следует заметить, что, несмотря на многочисленные и вполне очевидные аргументы в пользу такого утверждения, оно было строго доказано только в 1992 году.

Итак, повторное применение алгоритма DES задаёт отображение, неэквивалентное отображениям, получаемым при однократном использовании этого алгоритма. Но для криптоанализа этой схемы существует и другой путь, независящий от конкретных особенностей алгоритма DES, а, наоборот, подходящий для всех блочных шифров.

Соответствующий алгоритм получил название ***метода двусторонней атаки*** (*meet-in-the-middle attack*). Он основан том факте, что из $C = E_{k_2}\left(E_{k_1}(P)\right)$ следует, что $X = E_{k_1}(P) = D_{k_2}(C)$. При наличии известной пары $(P,\ C)$ анализ с помощью данного метода выполняется по следующей схеме. Сначала проводится шифрование P с каждым из 2^{56} возможных значений k_1. Полученные результаты сохраняются в таблице, а сама таблица сортируется по значениям X. Затем выполняется дешифрование C для каждого из 2^{56} возможных значений k_2 с одновременным сравнением получаемых результатов с результатами, представленными в таблице. Если обнаруживается совпадение, то соответствующие два ключа проверяются на следующей известной паре открытого и шифрованного текста. Если оказывается, что и с новой парой эти два ключа порождают правильный шифрованный текст, они принимаются как истинные ключи.

Оказывается, что оценка для сложности задачи криптоанализа "двойного" алгоритма DES с известным открытым текстом, несмотря на размер ключа в 112 битов, имеет порядок 2^{56}, что ненамного превышает оценку 2^{55}, имеющую место для обычного алгоритма DES.

Другой способ двойного шифрования, называемый ***методом Дэвиса-Прайса***, построен на идеях режима шифрования CBC: $\boxed{C_i = E_{k_2}\left(P_i \oplus E_{k_1}(C_{i-1})\right)}$, $i = \overline{1,N}$.

Метод двусторонней атаки позволяет определить ключи и в этом случае, характеристики сложности метода примерно такие же.

Тройное шифрование

Более стойкие схемы используют тройное шифрование. Очевидным решением проблемы противодействия методам двусторонней атаки является тройное шифрование с использованием трёх разных ключей [1]. Это увеличивает сложность задачи криптоанализа с известным открытым текстом до 2^{112}, что выходит за рамки как сегодняшних реальных возможностей, так и возможностей обозримого будущего. Однако недостаток такого подхода – требование использовать ключ размером $56 \times 3 = 168$ бит, который можно считать уж слишком громоздким.

Схему тройного шифрования Тачмена с парой независимых ключей k_1 и k_2 называют ***режимом EDE*** (зашифрование-расшифрование-зашифрование): $\boxed{C = E_{k_1}\left(D_{k_2}\left(E_{k_1}(P)\right)\right)}$.

Тройной алгоритм DES с двумя ключами стал довольно популярной альтернативой стандартному алгоритму DES, рекомендованной стандартами управления ключами ANSI X9.17 и ISO 8732.

При одинаковых ключах эта схема равносильна однократному шифрованию, что позволяет совместно использовать эти схемы в общей сети. Несмотря на чередование ключей, исключающее стандартный метод согласования, Меркл и Хеллман разработали оригинальный вариант метода согласования памяти и времени, требующий выполнения порядка 2^{56} операций и некоторого количества подобранных блоков открытого текста.

Хотя данные методы криптоанализа и оказываются несостоятельными с практической точки зрения, использование тройного DES с двумя ключами может вызывать некоторые опасения относительно его криптоаналитической стойкости. Поэтому многие специалисты склоняются к мысли, что более надёжной альтернативой является ***тройной алгоритм DES с тремя ключами***. Последний использует ключ, эффективная длина которого составляет 168 бит, и выполняет преобразование, задаваемое формулой $\boxed{C = E_{k_3}\left(D_{k_2}\left(E_{k_1}(P)\right)\right)}$.

Тройной алгоритм DES с тремя ключами реализован во многих приложениях, ориентированных на работу в сети Internet, в том числе в PGP и S/MIME.

Рассмотренные схемы кратного шифрования могут сочетаться с различными режимами шифрования.

Задание

I. Реализовать приложение для шифрования, позволяющее выполнять следующие действия:
   1. Шифровать данные с использованием заданного в варианте режима шифрования, применённого для того симметричного алгоритма, который был реализован в предыдущей лабораторной работе:
      1) шифруемый текст должен храниться в одном файле, ключ шифрования – в другом, а вектор инициализации – в третьем;
      2) зашифрованный текст должен сохраняться в файл;
      3) в процессе шифрования предусмотреть возможность просмотра и изменения ключа, вектора инициализации, шифруемого и зашифрованного текстов в шестнадцатеричном и символьном виде.
   2. Шифровать данные по заданной в варианте схеме кратного шифрования.

3. Исследовать лавинный эффект:
    1) приложение может самостоятельно строить необходимые графики либо графики можно строить в стороннем ПО, но тогда приложение для шифрования должно сохранять в файл необходимую для построения графиков информацию.

II. Реализовать приложение для дешифрования, позволяющее выполнять следующие действия:
1. Дешифровать данные с использованием заданного в варианте режима шифрования, применённого для того симметричного алгоритма, который был реализован в предыдущей лабораторной работе:
    1) зашифрованный текст должен храниться в одном файле, ключ – в другом, а вектор инициализации – в третьем;
    2) расшифрованный текст должен сохраняться в файл;
    3) в процессе дешифрования предусмотреть возможность просмотра и изменения ключа, вектора инициализации, зашифрованного и расшифрованного текстов в шестнадцатеричном и символьном виде.
2. Дешифровать данные по заданной в варианте схеме кратного шифрования.

III. С помощью реализованных приложений выполнить следующие задания:
1. Протестировать правильность работы разработанных приложений.
2. Исследовать лавинный эффект для реализованного режима шифрования (рассматривать текст из трёх блоков):
    1) построить графики зависимости числа изменённых бит в блоках C_1, C_2, C_3 от позиции изменившегося бита в открытом тексте (3 отдельных графика или 3 зависимости на 1 графике);
    2) построить графики зависимости числа изменённых бит в блоках C_1, C_2, C_3 от позиции изменившегося бита в ключе (3 отдельных графика или 3 зависимости на 1 графике);
    3) построить графики зависимости числа изменённых бит в блоках C_1, C_2, C_3 от позиции изменившегося бита в векторе инициализации (3 отдельных графика или 3 зависимости на 1 графике);
    4) построить графики зависимости числа изменённых бит в блоках P_1, P_2, P_3 от позиции изменившегося бита в зашифрованном тексте (3 отдельных графика или 3 зависимости на 1 графике).
3. Исследовать лавинный эффект для реализованной схемы кратного шифрования (рассматривать текст из 1 блока).
4. Сделать выводы о проделанной работе.

Дополнительные критерии оценивания качества работы

1. Наглядность приложений:
    1 – приложения позволяют просматривать и изменять ключи, шифруемый и зашифрованный тексты во всех предусмотренных заданием представлениях;
    0 – приложения позволяют просматривать ключи, шифруемый и зашифрованный тексты только в каком-то одном представлении;
    л.р. не принимается – иначе.
2. Построение графиков:
    1 – программа сама строит графики лавинного эффекта;
    0 – программа только выгружает необходимые для построения графиков данные;
    л.р. не принимается – программа не строит графики и не выгружает данные.

Варианты

При выполнении лабораторной работы используется тот же симметричный алгоритм шифрования, что был в предыдущей лабораторной работе.

№ варианта	Режим шифрования	Схема кратного шифрования
1	CFB	Схема с тремя ключами
2	BC	Метод Дэвиса-Прайса
3	PCBC	EDE
4	OFB	Простая двойная
5	CBC	Схема с тремя ключами
6	OFBNLF	Метод Дэвиса-Прайса
7	OFB	Схема с тремя ключами
8	CFB	EDE
9	CBC	Метод Дэвиса-Прайса
10	BC	Простая двойная
11	OFB	EDE
12	PCBC	Схема с тремя ключами
13	OFBNLF	Метод Дэвиса-Прайса
14	CFB	Простая двойная
15	CBC	EDE

Вопросы для защиты

1. Режим электронной шифровальной книги (ECB).
2. Режим сцепления шифрованных блоков (CBC).
3. Режим обратной связи по шифротексту (CFB).
4. Режим обратной связи по выходу (OFB).
5. Режим сцепления блоков (BC).
6. Режим сцепления блоков шифротекста с распространением ошибки (PCBC).
7. Режим нелинейной обратной связи по выходу (OFBNLF).
8. Двойное шифрование.
9. Метод двусторонней атаки.
10. Тройное шифрование.

Список литературы

1. Столлингс, В. Криптография и защита сетей: принципы и практика : Пер. с англ. / В. Столлингс. – 2-е изд. – М. : Издательский дом "Вильямс", 2001. – 672 с.
2. Шнайер, Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си / Б. Шнайер. – М. : Триумф, 2002. – 816 с.
3. DES Modes of Operation : FIPS Publication 81. – Gaithersburg : National Institute of Standards and Technology, 1980.