**Самостоятельная работа № 2**

Тема: Блочные шифры.

Ход работы:

Задание 1 Сложите по модулю 2:

* двоичные числа 10101100 и 11001010 ;
* десятичные числа 15 и 10;
* шестнадцатеричные числа 0В5 и 37.

Таблица 1 – Решение 1-ого пункта 1 задания.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |

Перевод числа из десятичного в двоичное 1 задания:

15 = 1111

10 = 1010

Таблица 2 – Решение 2-ого пункта

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |

Перевод числа из шестнадцатеричного в двоичное:

0В5 = 10110101

37 = 110111

Таблица 3 – Решение 3-ого пункта 1 задания.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

Задание 2. Сложите по модулю :

- двоичные числа 10101100 и 11001010;

- десятичные числа 155 и 100;

- шестнадцатеричные числа 0В5 и 37.

Таблица 4 – Решение 1-ого пункта 2 задания.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
|  | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |

Перевод числа из десятичного в двоичное:

155 = 10011011

100= 01100100

Таблица 5 – Решение 2-ого пункта 2 задания.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Перевод числа из шестнадцатеричного в двоичное:

0B5 = 10110101

37 = 00110111

Таблица 6 – Решение 3-его пункта 2 задания.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |

Задание 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.

Таблица 7 - Сдвиг числа 10101100 на 5 разрядов влево.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Исходное число | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| Сдвиг на 5 разрядов | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |

Перевод числа из шестнадцатеричной в двоичную:

9E = 10011110

Таблица 8 - Сдвиг числа 10011110 на 4 разряда вправо.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Исходное число | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |
| Сдвиг на 4 разряда | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |

Перевод числа из шестнадцатеричной в двоичную:

55 = 10101010

Таблица 9 - Сдвиг числа 10101010 на 2 разряда вправо

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Исходное число | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| Сдвиг на 2 разряда | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |

Задание 4 Пусть каждые три бита входного сообщения заменяются по следующей таблице замен:

Таблица 10 – Таблица входов/выходов.

|  |  |
| --- | --- |
| Вход | Выход |
| 000 | 011 |
| 001 | 101 |
| 010 | 000 |
| 011 | 111 |
| 100 | 010 |
| 101 | 110 |
| 110 | 001 |
| 111 | 100 |

0->3

1->5

2->0

3->7

4->2

5->6

6->1

7->4

Задание 5 Выполните разбиение исходного сообщения на блоки по три бита и произведите поблочную замену для следующих сообщений, представленных в цифровом виде:

- 1010 1100 1100(2)

- 2356(10)

- 0В57(16)

2356 = 1001 0011 0100

0B57 = 1011 0101 0111

Таблица 11 - Решение задания 5.

|  |  |  |
| --- | --- | --- |
| 101011001100 | 101 011 001 100 | 110 111 101 010 |
| 100100110100 | 100 100 110 100 | 010 010 001 010 |
| 101101010111 | 101 101 010 111 | 110 110 000 100 |

Контрольные вопросы:

1. Комбинация нескольких подряд примененных простых шифров, (например, перестановки или подстановки) дает в результате более сложное преобразование, называемое комбинированным (композиционным) шифром.
2. На стойкость шифра влияют такие факторы, как размер блока, размер ключа, количество раундов шифрования.
3. Одна из часто используемых операций – операция побитового сложения по модулю 2, обозначаемая XOR или. Принципы выполнения этой операции подробно рассмотрены в "Простейшие методы шифрования с закрытым ключом" . При сложении по модулю 2 операнды обрабатываются поразрядно. В разряде результата ставится единица, если в соответствующих разрядах операндов присутствует нечетное число единиц. Эта операция имеет очень удобное свойство: вычитание по модулю два есть то же самое, что и сложение, поэтому один из операндов может быть получен путем прибавления к сумме другого операнда. Также в блочных алгоритмах шифрования широко используется операция сложения по модулю 2 32 или по модулю 2 16. Эта операция представляет собой обыкновенное сложение двоичных чисел без учета переноса в старший 32-й или 16-й разряд результата.
4. В реальных шифрах также используется комбинация нескольких простейших операций над цепочками или блоками знаков. Для повышения криптостойкости эти операции выполняются циклически несколько раз, образуя раунды или шаги. На стойкость шифра влияют такие факторы, как размер блока, размер ключа, количество раундов шифрования. Современные шифры с закрытым ключом обрабатывают только двоичные данные, поэтому в них помимо обычных замены и перестановки применяются некоторые другие специфичные для двоичных чисел операции.
5. Алгоритмы симметричного шифрования могут обрабатывать исходный текст блоками или потоком. В зависимости от этого различают блочные алгоритмы симметричного шифрования и поточные. Блок текста рассматривается как неотрицательное целое число либо как несколько независимых неотрицательных целых чисел. Длина блока всегда выбирается равной степени двойки, например, 64, 128, 256 бит. Операции, используемые в блочных алгоритмах симметричного шифрования. Рассмотрим операции, используемые в большинстве алгоритмов симметричного шифрования. Будем при этом помнить, что рассматриваемые операции применяются к двоичным данным. Любая информация, например, изображения или текст, могут быть представлены в двоичном виде. Благодаря этому при шифровании не приходится задумываться о смысле передаваемых сообщений.
6. Требования к блочному алгоритму шифрования

* Алгоритм должен обеспечивать высокий уровень стойкости, и эта стойкость не должна основываться на сохранении втайне самого алгоритма.
* Незначительное изменение исходного сообщения должно приводить к существенному изменению зашифрованного сообщения даже при использовании одного и того же ключа.
* Алгоритм должен успешно противостоять атакам по выбранному тексту, то есть таким, чтобы нельзя было узнать ключ, даже зная достаточно много пар (зашифрованное сообщение, незашифрованное сообщение), полученных при шифровании с использованием данного ключа.
* Алгоритм шифрования должен иметь возможность быть реализованым на различных платформах, которые предъявляют различные требования. Для наиболее быстрых приложений используется специальная аппаратура. Несмотря на это, программные реализации применяются также достаточно часто. Поэтому алгоритм должен допускать эффективную программную реализацию на универсальных микропроцессорах. Алгоритм должен также работать на микроконтроллерах и других процессорах среднего размера.
* Алгоритм должен использовать простые операции, которые эффективны на микропроцессорах, т.е. исключающее или, сложение, табличные подстановки, умножение по модулю. Не должно использоваться сдвигов переменной длины, побитных перестановок или условных переходов.
* Алгоритм должен эффективно реализовываться на специализированной аппаратуре, предназначенной для выполнения операций шифрования и расшифрования, то есть реализация алгоритма в виде электронных устройств должна быть экономичной.
* Алгоритм шифрования должен быть применим во многих приложениях. Алгоритм должен быть эффективен при шифровании файлов данных или большого потока данных, при создании определенного количества случайных битов, а также должна быть возможность его использования для формирования односторонней хеш-функции.
* Алгоритм должен быть простым для написания кода, чтобы минимизировать вероятность программных ошибок. Также это дает возможность анализа и уменьшает закрытость алгоритма.
* Алгоритм должен допускать любую случайную строку битов нужной длины вкачестве возможного ключа (это называется иметь плоское пространство ключей ). Не должнобыть "слабых" ключей, облегчающих криптоанализ.
* Алгоритм должен легко модифицироваться для различных уровней безопасности и удовлетворять как минимальным, так и максимальным требованиям.

1. Блочный алгоритм шифрования должен иметь простую и понятную структуру по нескольким причинам: Проверяемость безопасности: Прозрачность структуры алгоритма позволяет криптографам и исследователям анализировать его на наличие слабых мест и уязвимостей. Если структура слишком сложная, это затрудняет проверку алгоритма на безопасность и делает его уязвимым для атак. Формальные доказательства безопасности: Для некоторых алгоритмов возможно формально доказать их стойкость на основе математических моделей и предположений (например, о сложности решения некоторых задач). Простая структура облегчает формулировку таких доказательств. Упрощение анализа атак: Сложные алгоритмы могут содержать неявные ошибки или возможности для использования их свойств злоумышленниками. Простой алгоритм легче анализировать с точки зрения различных типов атак, таких как дифференциальный и линейный криптоанализ. Упрощение реализации: Простая структура алгоритма снижает вероятность ошибок при его реализации. Это особенно важно для оборудования с ограниченными ресурсами (например, встроенных систем или IoT-устройств), где сложная реализация может оказаться неэффективной или даже небезопасной из-за ошибок программирования. Аудит и доверие: Простой алгоритм легче подвергнуть независимому аудиту, что повышает доверие к его безопасности со стороны сообществ и организаций. Чем более понятен и проверен алгоритм, тем выше вероятность его широкого принятия и использования. Оптимизация производительности: Простые алгоритмы легче оптимизировать для различных архитектур процессоров, что важно для производительности и масштабируемости, особенно при большом объеме данных.
2. Обычно под "высокой криптостойкостью" понимают, что шифр должен быть стоек по отношению к атаке по выбранному тексту. Это автоматически подразумевает его стойкость по отношению к атакам по шифротексту и по известному тексту. Однако известно, что при атаке по выбранному тексту шифр всегда может быть взломан путем перебора ключей. Поэтому требование стойкости шифра можно уточнить следующим образом: Шифр стоек (при атаке по выбранному тексту), если для него не существует алгоритма взлома, существенно более быстрого, чем прямой перебор ключей". Интересно, что по состоянию на сегодняшний день ни для одного используемого шифра не доказано строго соответствие этому определению стойкости.
3. Большой вклад в исследования принципов разработки блочных шифров внес американский ученый Х. Фейштель (Horst Feistel). Он, в частности, принимал участие в разработке системы шифрования "Люцифер" фирмы IBM. Фейштель предложил структуру, называемую в настоящее время сетью Фейштеля. Сети Фейштеля получили широкое распространение, так как, с одной стороны, они удовлетворяют всем требованиям к алгоритмам симметричного шифрования, а с другой стороны, достаточно просты и удобны в использовании. Раунд, организованный по сети Фейштеля имеет следующую структуру. Входной блок делится на несколько частей равной длины. Эти части блока называются ветвями. Так, например, если блок имеет длину 64 бита, используются две ветви по 32 бита каждая. Ветви обрабатываются по отдельности, после чего осуществляется циклический сдвиг всех ветвей влево. В случае двух ветвей каждый раунд имеет структуру.