Лабораторная работа №2
Блочно-итерационные криптосистемы

Дедлайн: 27.11.2024

1 Введение

В данной лабораторной работе вам необходимо программно реализовать одну из 15 блочно-итерационных криптосистем, основанных на сети Фейстеля:

1. ГОСТ 28147-89
2. Blowfish
3. Camellia
4. CAST-128
5. CAST-256
6. CLEFIA
7. DES
8. IDEA
9. MARS
10. RC5
11. RC6
12. Triple DES
13. Twofish

Дополнительную информацию вы можете получить в книгах по криптографии.

2 Условие лабораторной работы

Ваша задача: выбрать одну из приведенных во введении криптосистем и выполнить ее реализацию, которая позволяет зашифровывать и расшифровывать бинарные файлы с данными произвольной длины для заданного ключа для следующих режимов шифрования: ECB, CBC, PCBC, CFB, OFB, CTR.

При этом может использоваться один из следующих режимов дополнения последнего блока: *ANSI X.923*, *ISO 10126*, *PKCS7*, *ISO/IEC 7816-4*.

Особенности реализации лабораторной работы

- Под реализацией следует понимать программу, написанную на одном из языков программирования: {C/C++, C#, Java, Python}.
- (При реализации на языке отличном от C++) При реализации только криптосистемы с режимом шифрования ECB и без корректной обработки последнего блока максимальный балл, который вы можете получить, составит 45 баллов.
- (При реализации на C ++) Допускается свободное использование примера с реализацией шифра DFC. При этом максимальный балл, который вы можете получить, составит 40 баллов. В таком случае вам достаточно изменить функцию Фейстеля и генерацию тактовых ключей. **Обратите внимание**, даже при неправильной реализации, как функции, так и расписания ключей расшифрование зашифрованного текста останется корректным. В связи с этим, для сдачи так же потребуется подробный вывод на экран всех этапов вычисления тактовых ключей и функции Фейстеля.
- (При реализации на языке отличном от C++) Дополнительная реализация каждого из 5 остальных режимов шифрования увеличит максимальный балл, который вы можете получить за лабораторную работу, на 7.

- (При реализации на языке отличном от C++) Дополнительная реализация каждого из 4 режимов дополнения последнего блока увеличит максимальный балл, который вы можете получить за лабораторную работу, на 5.
- Одну и ту же криптосистему могут реализовывать не более трех студентов.

Входные данные программы

- файл in.bin с открытым текстом / шифртекстом;
- файл key.bin, содержащий ключ;
- файл sync.bin, содержащий синхропосылку;
- указание, нужно зашифровывать или расшифровывать;
- указание, какой режим шифрования нужно использовать;
- указание, какой режим дополнения нужно использовать.

Выходные данные программы

- файл out.bin с шифртекстом / открытым текстом.

Замечания

- Если не реализованы дополнительные режимы шифрования, то файл sync.bin и указание, какой режим шифрования нужно использовать, на вход подавать не требуется.
- Если не реализованы режимы дополнения, то указание, какой режим дополнения нужно использовать, на вход подавать не требуется.
- При реализации режима счетчика (CTR) вы должны использовать синхропосылку (IV) для инициализации счетчика:$\mathrm{Ctr}_0 = \mathrm{E}_k(\mathrm{IV})$. Для вычисления значения счетчика Ctr_i для i-го блока воспользуйтесь формулой $\mathrm{Ctr}_i = (\mathrm{Ctr}_{i-1} + 1) \bmod 2^{128}$.
- Следует обратить внимание на использование режима дополнения: дополнение сообщения байтами осуществляется всегда, то есть даже в том случае, если длина исходных данных кратна размеру блока.
- Данный подход позволяет однозначно определить размер исходных данных.
- Файлы in.bin, key.bin, sync.bin и out.bin должны быть представлены в бинарном виде. Указания могут поступать на вход любым способом, в том числе из консоли, из файла и через оконный интерфейс.

3 Бонусные задания

3.2 Бонусное задание №1

Реализовать оконную программу, которая представляет собой менеджер паролей. Функциональные возможности программы определяются ее автором самостоятельно. При необходимости можно обратиться к преподавателю за консультацией.

3.3 Бонусное задание №2

Реализовать оконную программу, которая представляет собой аналог программы TrueCrypt. Функциональные возможности программы определяются ее автором самостоятельно. При необходимости можно обратиться к преподавателю за консультацией.

3.4 Особенности бонусных заданий

- За бонусные задания можно получить от 20 до 50 баллов.
- Бонусные задания можно выполнить и сдать до конца семестра.

4 Порядок сдачи лабораторной работы

Работа сдается лично преподавателю. Компиляция (интерпретация) программы должна выполняться из явно указанного исходного файла (или базового файла проекта) без ошибок. При наличии любых частей кода выполненных не самостоятельно (в т.ч. если код написан самостоятельно, но по его структуре консультировались с другими студентами), преподаватель должен быть уведомлен заблаговременно, и соответствующие комментарии !обязаны! быть указаны в исходном коде.

• **Обычный порядок сдачи**: в качестве срока сдачи используется дата в которую вы сдаете лабораторную работу преподавателю.

• **Досрочный порядок сдачи**: при помощи программы, предоставленной преподавателем (HashCalc), для исходного файла вычисляется контрольная характеристика. В группу Telegram отправляется сообщение в формате #TaskDone <имя файла с расширением либо директории> <значение контрольной характеристики>. (Пример: #TaskDone lab2_Ivanov_project3 K3aSsaHCTjC4q8r/X/ZhZIIkMxqGYZtzhMjHHeq5c/o=)
Работа сдается очно, при соблюдении всех условий (в том числе совпадении значения контрольной характеристики), в качестве даты сдачи используется дата сообщения в Telegram.
Сообщения с пометкой “редактировано” не принимаются!