МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования «Балтийский федеральный университет имени Иммануила Канта»

ОНК «Институт высоких технологий»

ОТЧЁТ

о прохождении учебной практики по получению первичных

профессиональных умений и навыков, в том числе первичных умений и навыков научно-исследовательской деятельности

на базе Высшей школы компьютерных наук и прикладной математики образовательно-научного кластера "Институт высоких технологий"

Выполнил Нецветайлов Андрей Александрович \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

студент очной формы обучения 3 курса

специальности 10.05.01 Компьютерная безопасность

специализация «Математические методы защиты информации»

Руководитель практики от университета

доцент ОНК «ИВТ» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Киршанова Е.А.

г. Калининград 2023 г.

Оглавление

[Введение 3](#__RefHeading___Toc561_2911844422)

[Глава 1. Наше первое задание. 4](#__RefHeading___Toc556_2911844422)

[Глава 2. Наше второе задание 6](#__RefHeading___Toc567_2911844422)

[Глава 3. Описание AES-256 7](#__RefHeading___Toc569_2911844422)

[Глава 4. Режимы шифрования 9](#__RefHeading___Toc571_2911844422)

[Глава 5.Решние 11](#__RefHeading___Toc573_2911844422)

[Заключение 13](#__RefHeading___Toc1726_3202987432)

[Список литературы и источники 14](#__RefHeading___Toc575_2911844422)

## ***Введение***

Вид практики – учебная практика по получению первичных профессиональных умений и навыков, в том числе первичных умений и навыков научно-исследовательской деятельности, далее Учебная практика.

Цель учебной практики: изучит ранее неизвестную криптосистему, улучшить навыки командной работы, усовершенствовать навыки работы с LaTeX.

Задачи учебной практики:

* Выбрать задание на сайте https://cryptohack.org/challenges/ctf-archive/
* Изучить криптосистему, используемую в задании
* Изучить программу, приложенную к заданию и найти в ней уязвимость
* Написать программу для получения флага и составить презентацию-отчёт в LaTeX.

Приступая к выполнению задания, мы хотели узнать что-то новое для себя, что не проходили по программе в университете и отточить это на практике. Также мы хотели сделать более качественную презентацию, по сравнению с предыдущем годом.

## ***Глава 1. Наше первое задание***

Изначально мы выбрали задачу «unrandom DSA (HackTM CTF)». В этой задаче нужно было взломать систему электронной подписи DSA. Для решения задачи мы решили сначала изучить криптосистему DSA, стандарт цифровой подписи DSS и хэш-функцию SHA-256.

DSS – это стандарт США, описывающий алгоритм цфировой подписи. август 1991

DSA – это Digital Signature Algorithm. В переводе с английского «алгоритм цифровой подписи». Он очень похож на RSA. Основное отличие заключается в том, что DSA шифрует не само сообщение, а хэш-функцию сообщения.

(p, q, g, y) – открытый ключ;

(x) – закрытый ключ

Сначала выбирает просто число **q** длиной 256. Затем выбирает простое число p. Оно должно быть длинной 2048 бит и удовлетворять выражению: **q | p-1.** Дальше выбирается число **g.**  Оно должно быть таким, что его **мультипликативный порядок по модулю p равен q**. Секретный ключ **x** должен быть случайным и меньше **q**.Открытый ключ **y = gx mod p**.

Алгоритм SHA-256 принадлежит семейству SHA-2, семейству однонаправленных хэш-функций. Хеш-функции предназначены для создания «отпечатков» или «дайджестов» для сообщений произвольной длины. Cемейство SHA-256 включает в себя в себя алгоритмы SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/256 и SHA-512/224.

Хэш-функция должна удовлетворять следующим требованиям:

* На выходе должна быть строка определённой длины, которая не зависит от длины входных данных.
* Функция должна быть однонаправленной. То есть зная функцию и зная хэш, не получится восстановить исходное сообщение.
* Уникальный хэш и полное или практически полное отсутствие коллизий. Это означает, что два разных набора входных данных не должны иметь одинаковый хэш. Однако, все хэш-функции имеют коллизии, но их теоретическая частота появления стремиться к нулю.
* Изменение одного байта должно приводить к сильному изменению значения хэша.

Благодаря вышеперечисленным требованиям к хэш-функциям, незначительные изменения сообщения приведут к значительному изменению хэша. Это сообщит адресату, что полученное сообщение не является той информацией, которую отправил отправитель. Благодаря устойчивости к коллизиям и шифрованию, подделать значение хэша на практике невозможно.

Эта задача показалась нам несколько скучной, потому что напоминала RSA.

## ***Глава 2. Наше второе задание***

На этот раз мы решили выбрать задание «Cipher Mode Picker (HKCERT CTF)». В этом задании используется криптосистема AES-256. Она является симметричной системой. В университете мы говорили только об ассиметричном шифровании. Единственное исключение это семинар с сибиряками. И то, речь там шла о криптоанализе AES, а не о самом шифровании. Поэтому такое задание нам показалось весьма интересным и в какой-то степени экзотическим. Сначала мы обратили своём внимание непосредственно на алгоритм и на математику, которая в него заложена; заострили своё внимание на режимах шифрования и потом приступили к рещению.

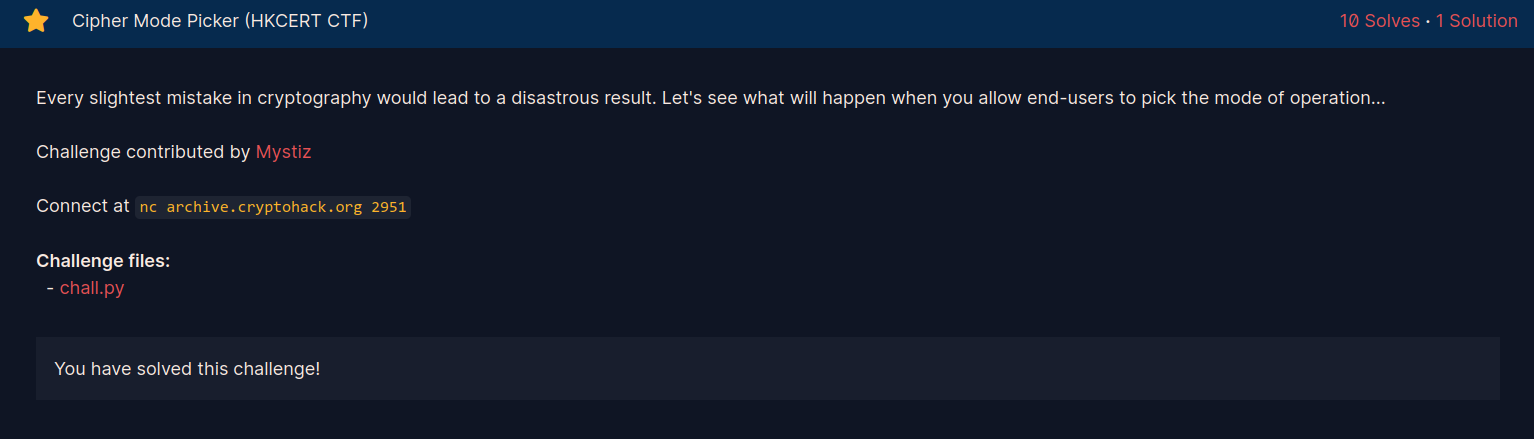


Рис 1. Задание наcryptohack.org с принятым решением

## ***Глава 3. Описание AES-256***

AES-256 это симметричный алгоритм блочного шифрование, который является стандартом в США и был принят по результатам конкурса AES в 2000 году. 26 мая 2002 года криптосистема официально стала стандартом.

AES пришёл на смену DES, который, в связи с развитием ЭВМ, стал ненадёжным. На вход подаются блоки длинной 128 бит и ключ длинной 128, 192 или 256 бит, в зависимости от модификации. В нашем случае длинна ключа 256 бит.

Высосуровневое описание:

1. KeyExpansion — раундовые ключи получаются из ключа шифрования с использованием «расписания ключей AES» . AES-256 требует отдельного 256-битного блока ключей для каждого раунда плюс еще один.
2. Добавление ключа начального раунда:
   1. AddRoundKey — каждый байт состояния объединяется с байтом раундового ключа с помощью побитового исключающего «или».
3. 13 раундов:
   1. SubBytes — шаг нелинейной подстановки, при котором каждый байт заменяется другим в соответствии с таблицей поиска.
   2. ShiftRows — шаг транспонирования, при котором последние три строки состояния циклически сдвигаются на определенное количество шагов.
   3. MixColumns — операция линейного смешивания, которая работает со столбцами состояния, объединяя четыре байта в каждом столбце.
   4. AddRoundKey
4. Последний раунд:
   1. SubBytes
   2. ShiftRows
   3. AddRoundKey

Окончательные стандарты указаны в следующих документах:

* FIPS PUB 197: Advanced Encryption Standard (AES)
* FIPS PUB 197: Advanced Encryption Standard (AES)

## *Глава 4. Режимы шифрования*

Режимы шифрования являются неотъемлемой частью криптосистемы AES. В условии нашей задачи используются «Вектор инициализации IV» ECB, CBC, CFB, OFB, CTR. Их мы и рассмотрим.

**Вектор инициализации IV:**

Представляет собой блок битов, котор и для создания различных шифр-текстов.

ый используется другими режимами для рандомизации шифрования и для создания различных шифр-текстов.

**ECB (Режим электронной кодовой книги):**

Самый просто и неиспользуемый на практике режим шифрования, потому что плохо скрывает шаблоны, так как шифрует шифрует идентичные блоки входных данных в идентичные блоки зашифрованных данных.

**CBC (Режим сцепления блоков шифртекста):**

В этом режиме каждый блок входных данных подвергается операции исключающего «или» с предыдущем блоком зашифрованных данных. Таким образом шифруемый блок зависит от всех предыдущих блоков. В этом режиме к первому блоку применяется вектор инициализации, чтобы обеспечить уникальность последующих блоков.

**CFB (Режим сцепления блоков шифртекста):**

Этот режим шифруемый блок складывается по модулю 2 с зашифрованным предидущем блоке.

**OFB (Режим обратной связи по выходу):**

Чтобы получить зашифрованный текст этот режим превращает блочный шифр в синхронный шифр потока: генерирует ключевые блоки, которые являются результатом сложения с блоками открытого текста.

**CTR (Режим счётчика):**

В данном режиме предполагается возврат на вход соответствующего алгоритма блочного шифрования значения некоторого счётчика, накопленного с момента старта. Он генерирует потоковый шифр из блочного и применяет к потоку операцию XOR с текстом сообщения.

## *Глава 5.Решние*

Из приложенного файла можно сделать следующие выводы:

* Длина флага 80 символов.Реализовано пять режимов шифрования: ’ECB’, ’CBC’, ’CFB’, ’OFB’, ’CTR’
* key и вектор инициализации IV используются для каждого соединения.
* Реализовано пять режимов шифрования: «ECB», «CBC», «CFB», «OFB», «CTR».

Мы можем дать серверу данные для шифрования, позволить серверу шифровать флаг в течении 5 минут за одно подключение. Но можем использовать только один режим шифрования за одно подключение. Выше мы рассмотрели функции CFB и OFB. Они практически идентичны, поэтому мы их и будем использовать. Введём на вход нули и применим CFB, он вернёт шифрование каждого блока. Затем запросим зашифрованный флаг с режимом OFB. Объедем их вместе и получим флаг.

Код для CFB:

C0 = IV

Ci = Ek (Ci−1 ) ⊕ Pi

P = 0 ⇒ Ci = Ek (Ci−1 )

Код для OFB:

O0 = IV

Oi = Ek (Oi−1 )

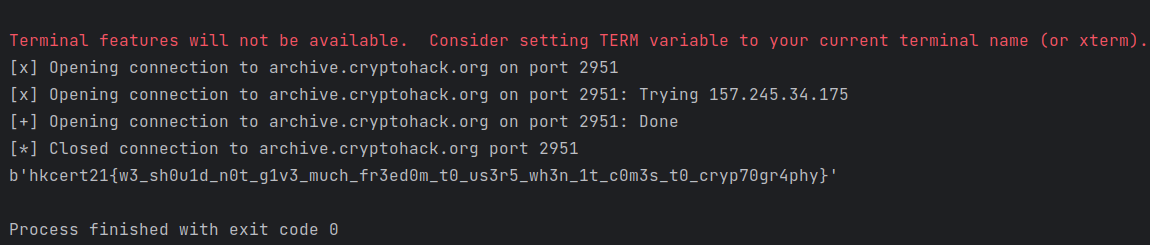
Ci = Ek (Oi−1 ) ⊕ Pi

В основной программе, написанной на языке Python использовались две библиотеки: PyCryptodome и pwntools.

Код программы:

from pwn import \*  
from Crypto.Util.number import long\_to\_bytes  
  
def xor(s1, s2):  
 return ''.join([str(int(a) ^ int(b)) for a,b in zip(s1,s2)])  
  
r = remote("archive.cryptohack.org", 2951)  
r.recvuntil(b'> ')  
r.sendline(b'cfb data '+ b'0'\*160)  
s1 = r.recvline()[:-1]  
r.recvuntil(b'> ')  
r.sendline(b'ofb flag')  
s2 = r.recvline()[:-1]  
r.close()  
t = xor(bin(int(s1,16))[2:],bin(int(s2,16))[2:])  
#t = int(t,2)  
print(t)  
print(long\_to\_bytes(int(t, 2)), 'itf-8')

Программа выводит следующее:

 Рис 2. Скриншот окна вывода IDE PyCharm

Флаг: hkcert21{w3\_sh0u1d\_n0t\_g1v3\_much\_fr3ed0m\_t0\_us3r5\_wh3n\_1t\_c0m3s\_t0\_cryp70gr4phy}

## ***Заключение***

Во время работы мы познакомились с двумя криптосистемами: DSA и AES. Помимо этого ознакомились со стандартом DSS, семейством хэш-функций SHA-2. Изучили и подробно разобрали AES-256, ознакомились с режимами шифрования, библиотеками для Python «pwntools» и «PyCryptodome», а так же закрепили навыки работы в LaTeX и продуктивно поработали в команде.

В течение практики все задачи были выполнены, а цели достигнуты.

## *Список литературы и источники*

**Перечень ресурсов информационно-телекоммуникационной сети «Интернет»**

1. Wikipedia: Advanced Encryption Standard. – режим доступа: https://en.wikipedia.org/wiki/Advanced\_Encryption\_Standard
2. Wikipedia: AES (стандарт шифрования). – режим доступа: https://ru.wikipedia.org/wiki/AES\_(%D1%81%D1%82%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%82\_%D1%88%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)
3. Wikipedia: Block cipher mode of operation – режим доступа: https://en.wikipedia.org/wiki/Block\_cipher\_mode\_of\_operation
4. Wikipedia: Режим шифрования – режим доступа: https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B6%D0%B8%D0%BC\_%D1%88%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F
5. Видеолекция на YouTube «Криптосистема AES» – режим доступа: https://www.youtube.com/watch?v=-lDezsnaU24