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 Бакиновский Михаил Николаевич \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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.

# СОДЕРЖАНИЕ

[СОДЕРЖАНИЕ 2](#_Toc140782669)

[ВВЕДЕНИЕ 3](#_Toc140782670)

[ОСНОВНАЯ ЧАСТЬ 4](#_Toc140782671)

[ГЛАВА 1. ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ НА ПРАКТИКУ 4](#_Toc140782672)

[ГЛАВА 2. ВЫПОЛНЕНИЕ ЗАДАНИЙ НА ПРАКТИКУ 4](#_Toc140782673)

[2.1. Решение задачи 4](#_Toc140782674)

[2.1.1. Теоретическая сводка. 4](#_Toc140782675)

[2.1.2. Анализ задачи 6](#_Toc140782676)

[2.1.3. Описание решения 7](#_Toc140782677)

[2.2 Верстка презентации 9](#_Toc140782678)

[2.3 Загрузка на github 10](#_Toc140782679)

[ЗАКЛЮЧЕНИЕ 11](#_Toc140782680)

[СПИСОК ЛИТЕРАТУРЫ 12](#_Toc140782681)

[Приложения 13](#_Toc140782682)

# 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: Сформировать и укрепить навыки владения системой верстки документов LaTeX, системой управления версиями git. Также укрепить навыки владения языком программирования python при решении поставленной задачи.

Помимо целей, представленных выше, важной задачей является сформировать умение применять полученные теоретические знания для решения практических задач.

Для того, чтобы достичь поставленных целей, были поставлены задачи:

1. Решить любую из задач в разделе CTF Archive на сайте cryptohack.org, используя любой удобный язык программирования, а также, основываясь на теоретических знаниях, полученных за период обучения.
2. Сформировать презентацию в системе верстки документов LaTeX с подробным описанием сути задачи и методах ее решения.
3. Создать открытый git-репозиторий с исходными кодами презентации и самого решения.

# ОСНОВНАЯ ЧАСТЬ

# ГЛАВА 1. ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ НА ПРАКТИКУ

**Задание 1.** Решить задачу broken oracle (HackTM CTF) на сайте cryptohack.org, используя язык программирования python

**Задание 2.** Сформировать презентацию, описывающую решение, с помощью системы LaTeX

**Задание 3.** Создать открытый git-репозиторий и загрузить в него исходные коды решения и презентации. Ссылку на репозиторий предоставить руководителю.

# ГЛАВА 2. ВЫПОЛНЕНИЕ ЗАДАНИЙ НА ПРАКТИКУ

# 2.1. Решение задачи

Дана программа, которая принимает на вход зашифрованное сообщение, на выход предоставляет enc(dec(сообщение)), где enc – функция шифрования, dec – функция расшифровки.

Исходный код условия задачи в приложении 1-4.

В задаче шифрование происходит в изменённой криптосистеме Рабина.

# 2.1.1. Теоретическая сводка.

Криптосистема Рабина - это один из первых публичных ключевых криптографических алгоритмов, предложенный Рональдом Линн Рабином в 1979 году. Основная идея криптосистемы Рабина заключается в использовании квадратного вычета для шифрования и расшифрования сообщений.

Для генерации ключей в криптосистеме Рабина выбираются два больших простых числа p и q, и вычисляется их произведение , которое используется в качестве открытого ключа. Для шифрования сообщения M используется формула: . Для расшифрования сообщения используется процесс нахождения корней квадратного уравнения . Однако, так как может существовать несколько корней, то криптосистема Рабина не обеспечивает уникальность расшифровки.

Криптосистема Рабина имеет несколько достоинств и недостатков. Достоинством является отсутствие необходимости в секретном обмене ключами и возможность использования открытого ключа для шифрования сообщений. Недостатком является отсутствие уникальности расшифровки, что делает криптосистему менее надежной в сравнении с другими криптографическими алгоритмами.

В изначальной задаче представлена изменённая криптосистема Рабина, а именно: Криптосистема Рабина с реципрокным[[1]](#footnote-1) методом дешифрования -публичный ключевой алгоритм шифрования, предложенный в статье "Public Key Cryptosystem Using a Reciprocal Number with the Same Intractability as Factoring a Large Number" (на которой основывается автор задачи). Он основан на использовании реципрокного числа и позволяет шифровать сообщения с помощью открытого ключа, который может быть распространен открыто, но для дешифрования сообщений требуется знание секретного ключа. В отличие от оригинальной криптосистемы Рабина, которая не гарантирует уникальность расшифровки, криптосистема Рабина с реципрокным методом дешифрования обеспечивает уникальность расшифровки. Это достигается путем добавления информации о корнях квадратного уравнения, которое используется при расшифровке сообщений.

Ключом шифрования является публичный ключ, который состоит из чисел n и c. Число n является произведением двух больших простых чисел p и q, а число c выбирается случайно таким образом, чтобы и , где - символ Якоби. Таким образом, c должно быть вычетом по модулю p и по модулю q.

Для шифрования сообщения m сначала вычисляется значение x, которое является квадратным корнем числа в конечном поле , где - мультипликативное обратное к в конечном поле , т.е. . Затем вычисляется символ Лежандра для числа по модулю . И наконец, вычисляется , равное 1, если , и 0 в противном случае.

Зашифрованное сообщение представляется тройкой чисел (r, s, t), где r равно . Для каждого зашифрованного сообщения (r, s, t) существует два возможных расшифрованных сообщения и , таких что , и символ Лежандра s равен 1 для одного из этих сообщений и -1 для другого.

Для расшифровки сообщения требуется закрытый ключ, который состоит из двух больших простых чисел p и q, таких что . Для каждого из двух возможных значений и , вычисляются корни и уравнения и корни и уравнения . Затем находятся два возможных значения m по формуле КТО (китайской теоремы об остатках) и , где crt - это функция, реализующая КТО. Из этих двух значений выбирается то, для которого символ Лежандра s равен тому значению, которое было вычислено при шифровании.

В статье, что используется автором задачи, говорится о том, как реципрокный метод дешифрования помогает выбрать из четырёх расшифрованных сообщений одно верное. Однако в изначальной задаче представлена криптосистема с «ошибкой»: автор оставляет лишь два значения, которые из-за «ошибок» в реализации кода – иногда дают одинаковые значения шифровки/расшифровки – этот принцип был упомянут ранее (перед теоретической сводкой).

# 2.1.2. Анализ задачи

В ходе выполнения задания были проведены следующие попытки атак на криптосистему:

1. Атака методом факторизации. Криптосистема Рабина основана на сложности факторизации больших простых чисел, и поэтому она уязвима к атакам методом факторизации. Также был опробован метод факторизации Ферма. Однако обе попытки были неудачны.
2. Атака методом квадратичного вычета. Криптосистема Рабина использует квадратичный вычет для шифрования и расшифрования сообщений, и поэтому она уязвима к атакам методом квадратичного вычета. Атака не увенчалась успехом.
3. Атака методом выбора открытого текста. Криптосистема Рабина не обеспечивает уникальности расшифровки, и поэтому она уязвима к атакам методом выбора открытого текста. Атака также не увенчалась успехом.
4. Атаки подбором (ключа, сообщения,…) – не увенчались успехом.
5. Атака методом анализа структуры сообщений. Криптосистема Рабина может быть уязвима к атакам, основанным на анализе структуры сообщений, таких как статистический анализ или анализ частоты символов. Атака не увенчалась успехом.

Также были опробованы атаки из курса обучения, такие, как: встреча посередине, атака Винера и атака повторным шифрованием.

Ни одна из приведённых выше атак не увенчалась успехом. Основные трудности для проведения всех перечисленных атак обусловлены тем, что берётся большой ключ (2048 бит), который меняется при каждом перезапуске программы. Все числа: n,p,q,c – всё берётся случайным образом и при перезапуске программы – выбирается новое случайным образом. Это усложняет как факторизацию, так и остальные виды атак, но, не исключает того, что атаки могли бы сработать в некоторых определённых, пусть и маловероятных, случаях.

# 2.1.3. Описание решения

Для поиска чисел p и q передаём оракулу несколько сообщений, записываем результат. Далее, по условиям криптосистемы нам необходимо, чтобы шифрованное сообщение соблюдало условия (было квадратичным невычетом), что в реализации алгоритма не проверяется. Таким образом, набираем несколько зашифрованных сообщений по этому принципу и вычисляем секретный ключ (p,q). А, уже зная секретный ключ, – можем расшифровать флаг.

Если сравнивать реализацию криптосистемы Рабина в задаче с тем, как она представлена в открытом доступе, можно заметить ошибку в решении систем уравнений ():

не имеет решений, когда

потому что

В функции decrypt возможен случай, когда число является квадратичным вычетом по модулю p, но не по модулю q. Пусть зашифрованное сообщение было расшифровано и результат равен . В этом случае в GF(p) пока в GF(q). Это означает, что является квадратом некоторого числа по модулю p, но не по модулю q. Поэтому, если обработать несколько - будет возможно восстановить p с помощью . Это также справедливо и для восстановления q.

В нашей криптосистеме роль открытого ключа играет с. Однако, конкретно в данной задаче с не доступно (так как это CTF). Пусть и , где r удовлетворяет уравнению

Если обработать , как , где путём изменения параметров s и t с тем же самым r. Как только следующие уравнения:

Можно решить достаточно просто:

Таким образом, возможно узнать c,m с помощью . Также можно восстановить публичный и секретный ключ, чтобы расшифровать флаг.

Для решения задачи был выбран язык программирования python, как самый простой и удобный в использовании, также, установка и подключение нестандартных библиотек осуществляется быстро и просто, при решении задачи использовались библиотеки ***gmpy2*** для генерации простых чисел, а также ***Crypto.Util.number*** для преобразования числа в байты. Условно программа делится на две части: работа оракула и попытка взлома, исходя из ответов оракула. Исходный код решения (дополнение к исходному коду) представлен в приложении 5.

# 2.2 Верстка презентации

Для верстки презентации можно было бы установить программное обеспечение, поддерживающее LaTeX, однако, было принято решение использовать отрытую платформу Overleaf. Ее главным преимуществом является бесплатность, удобность, поскольку не нужно устанавливать никакого дополнительного программного обеспечения, работа проходит в онлайн режиме.

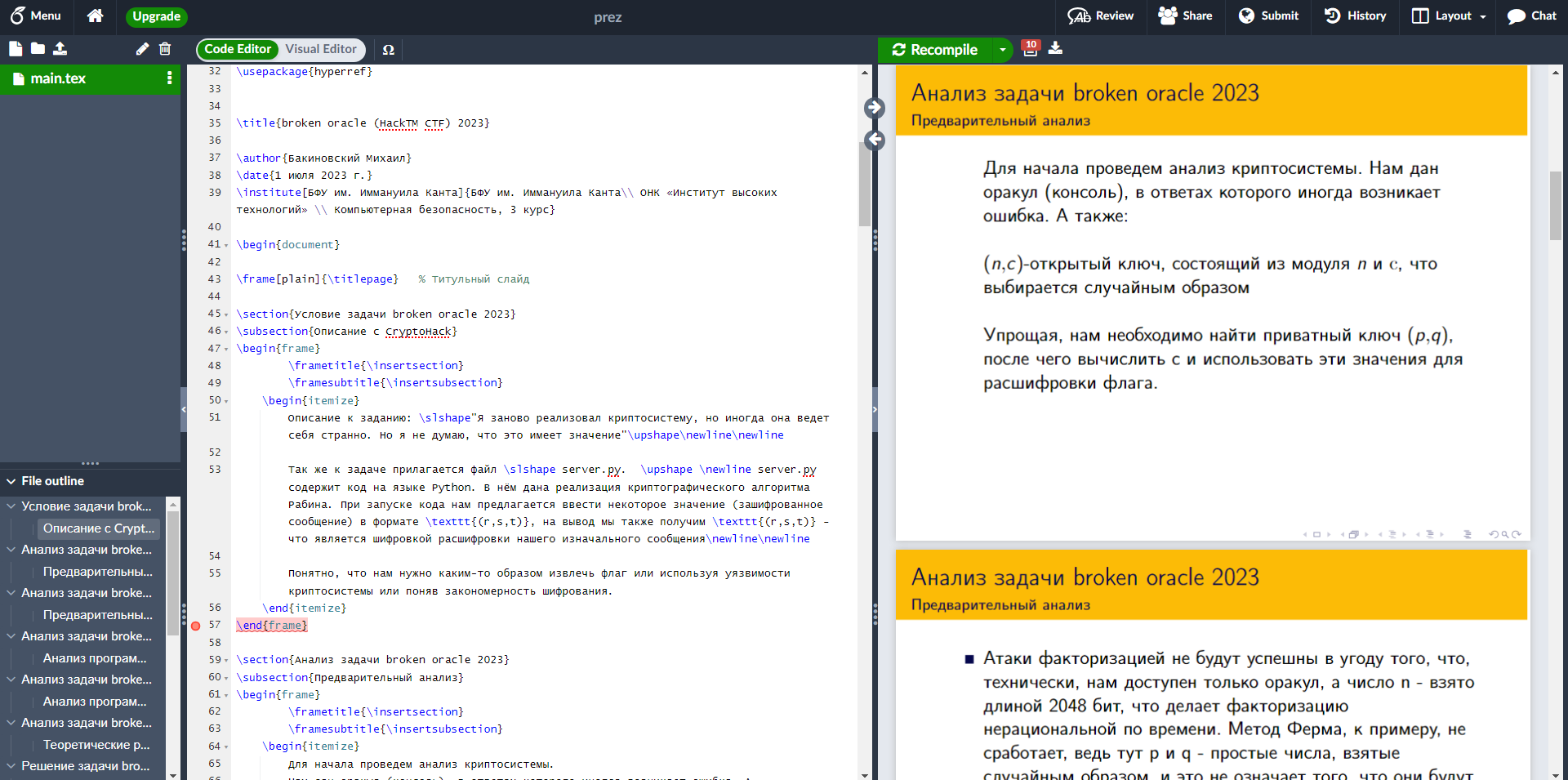


Рисунок 1. Интерфейс Overleaf

Также, помимо факторов, приведенных выше, у Overleaf есть собственное руководство с видео-уроками, из-за чего обучение протекало быстро и понятно.

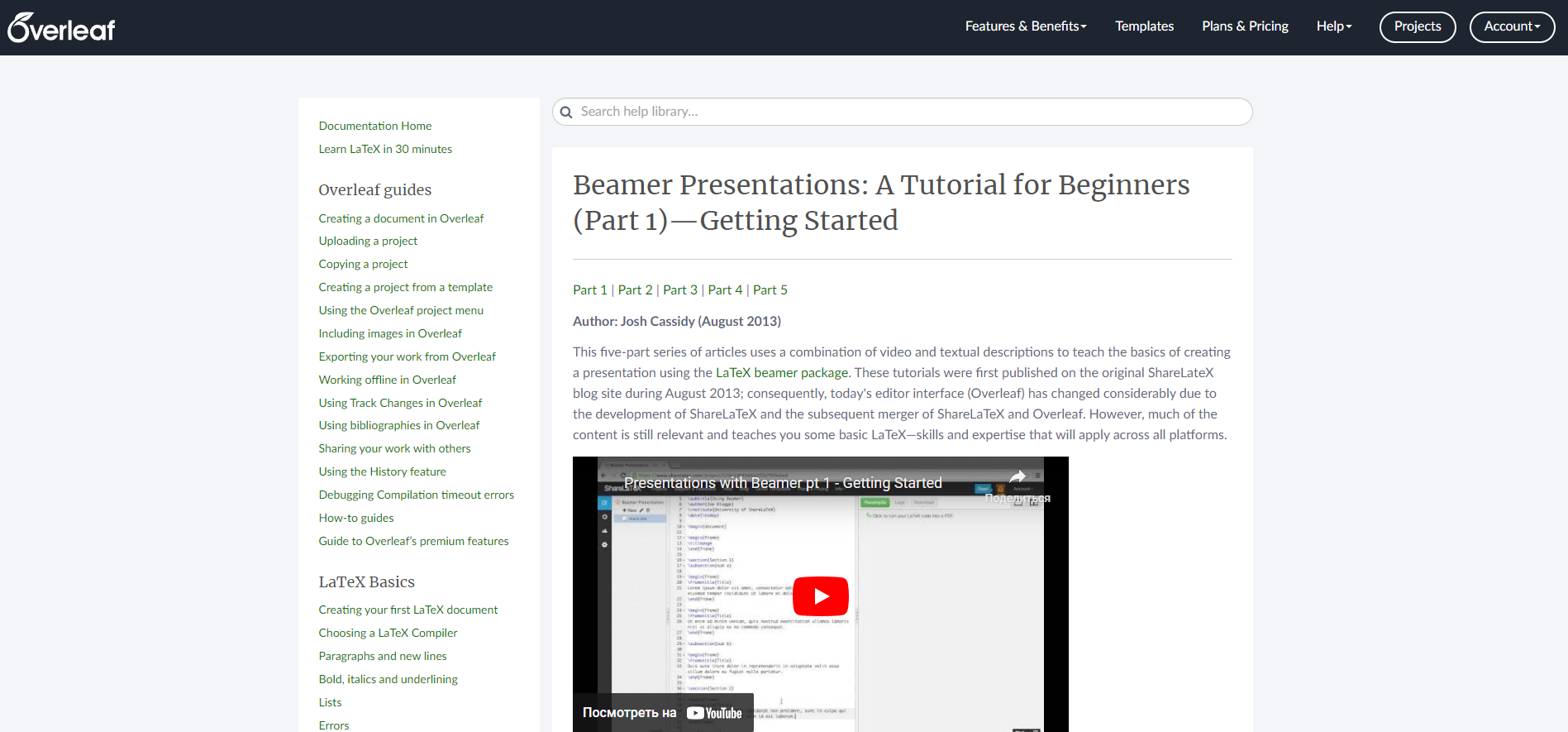


Рисунок 2. Руководство Overleaf

# 2.3 Загрузка на github

На сайте github.com создаю открытый репозиторий, после чего с помощью инструментов git (upload file) добавляю все необходимые файлы.

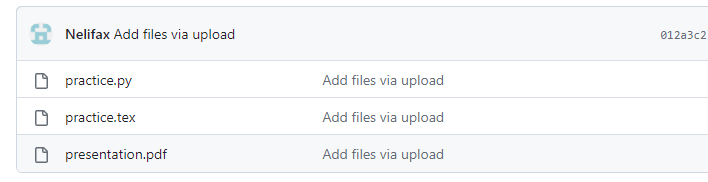


Рисунок 3. Онлайн репозиторий

Репозиторий доступен по ссылке: <https://github.com/Nelifax/Summer_practice2>

# ЗАКЛЮЧЕНИЕ

В течение учебной практики все поставленные задачи были выполнены, цели достигнуты.

Я укрепил свои навыки программирования на python. Также улучшил навыки работы с LaTeX, git и github.

Помимо этого, навыки исследовательской и научной деятельности были укреплены, удалось применить теоретические знания на практике, а также изучить новый криптоалгоритм.

# СПИСОК ЛИТЕРАТУРЫ

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

1. Официальный сайт Overleaf с видео-уроками - <https://www.overleaf.com/learn/latex/Beamer_Presentations%3A_A_Tutorial_for_Beginners_(Part_1)%E2%80%94Getting_Started>
2. Условие решаемой задачи - https://cryptohack.org/challenges/ctf-archive/
3. Официальный сайт git - [https://git-scm.com/](https://git-scm.com/%20)
4. Официальная документация git - <https://git-scm.com/docs>
5. Справочный материал к заданию от автора задания: https://www.cs.umd.edu/~gasarch/TOPICS/miscrypto/rabinwithrecip.pdf

# Приложения

Приложение 1

import os

import random

from dataclasses import dataclass

from math import gcd

from typing import List, Tuple

import gmpy2

from Crypto.Util.number import bytes\_to\_long, getPrime

@dataclass

class Pubkey:

n: int

c: int

@dataclass

class Privkey:

p: int

q: int

@dataclass

class Enc:

r: int

s: int

t: int

def \_\_repr\_\_(self) -> str:

return f"r = {self.r}\ns = {self.s}\nt = {self.t}"

def crt(r1: int, n1: int, r2: int, n2: int) -> int:

g, x, y = gmpy2.gcdext(n1, n2)

assert g == 1

return int((n1 \* x \* r2 + n2 \* y \* r1) % (n1 \* n2))

def gen\_prime(pbits: int) -> int:

p = getPrime(pbits)

while True:

if p % 4 == 3:

# return p

p = getPrime(pbits)

Приложение 2

def genkey(pbits: int) -> Tuple[Pubkey, Privkey]:

p, q = gen\_prime(pbits), gen\_prime(pbits)

n = p \* q

c = random.randint(0, n - 1)

while True:

if gmpy2.jacobi(c, p) == -1 and gmpy2.jacobi(c, q) == -1:

break

c = random.randint(0, n - 1)

pubkey = Pubkey(n=n, c=c)

privkey = Privkey(p=p, q=q)

return pubkey, privkey

def encrypt(m: int, pub: Pubkey) -> Enc:

assert 0 < m < pub.n

assert gcd(m, pub.n) == 1

r = int((m + pub.c \* pow(m, -1, pub.n)) % pub.n)

s = int(gmpy2.jacobi(m, pub.n))

t = int(pub.c \* pow(m, -1, pub.n) % pub.n < m)

enc = Enc(r=r, s=s, t=t)

assert s in [1, -1]

assert t in [0, 1]

return enc

def solve\_quad(r: int, c: int, p: int) -> Tuple[int, int]:

"""

Solve x^2 - r \* x + c = 0 mod p

See chapter 5.

"""

def mod(poly: List[int]) -> None:

"""Calculate mod x^2 - r \* x + c (inplace) """

assert len(poly) == 3

if poly[2] == 0:

return

poly[1] += poly[2] \* r

poly[1] %= p

poly[0] -= poly[2] \* c

poly[0] %= p

poly[2] = 0

Приложение 3

def prod(poly1: List[int], poly2: List[int]) -> List[int]:

"""Calculate poly1 \* poly2 mod x^2 - r \* x + c"""

assert len(poly1) == 3 and len(poly2) == 3

assert poly1[2] == 0 and poly2[2] == 0

res = [

poly1[0] \* poly2[0] % p,

(poly1[1] \* poly2[0] + poly1[0] \* poly2[1]) % p,

poly1[1] \* poly2[1] % p,

]

mod(res)

assert res[2] == 0

return res

# calculate x^exp mod (x^2 - r \* x + c) in GF(p)

exp = (p - 1) // 2

res\_poly = [1, 0, 0] # = 1

cur\_poly = [0, 1, 0] # = x

while True:

if exp % 2 == 1:

res\_poly = prod(res\_poly, cur\_poly)

exp //= 2

if exp == 0:

break

cur\_poly = prod(cur\_poly, cur\_poly)

# I think the last equation in chapter 5 should be x^{(p-1)/2}-1 mod (x^2 - Ex + c)

# (This change is not related to vulnerability as far as I know)

a1 = -(res\_poly[0] - 1) \* pow(res\_poly[1], -1, p) % p

a2 = (r - a1) % p

return a1, a2

def decrypt(enc: Enc, pub: Pubkey, priv: Privkey) -> int:

assert 0 <= enc.r < pub.n

assert enc.s in [1, -1]

assert enc.t in [0, 1]

mps = solve\_quad(enc.r, pub.c, priv.p)

mqs = solve\_quad(enc.r, pub.c, priv.q)

ms = []

for mp in mps:

for mq in mqs:

Приложение 4

m = crt(mp, priv.p, mq, priv.q)

if gmpy2.jacobi(m, pub.n) == enc.s:

ms.append(m)

assert len(ms) == 2

m1, m2 = ms

if m1 < m2:

m1, m2 = m2, m1

if enc.t == 1:

m = m1

elif enc.t == 0:

m = m2

else:

raise ValueError

return m

if \_\_name\_\_ == "\_\_main\_\_":

signal.alarm(300)

flag = os.environ.get("FLAG", "FAKEFLAG{THIS\_IS\_FAKE}").encode()

pbits = 1024

pub, priv = genkey(pbits)

while len(flag) < 255:

flag += os.urandom(1)

enc\_flag = encrypt(bytes\_to\_long(flag), pub)

print("encrypted flag:")

print(enc\_flag)

while True:

try:

r, s, t = map(int, input("r, s, t = ").split(","))

enc = Enc(r=r, s=s, t=t)

enc\_dec\_enc = encrypt(decrypt(enc, pub, priv), pub)

print("decrypt(encrypt(input)):")

print(enc\_dec\_enc)

except KeyboardInterrupt:

break

except Exception:

print("Something wrong...")

Приложение 5

def oracle(r,s,t,h=1): #имитация запросов к оракулу

RST=encrypt(decrypt(Enc(r=r,s=s,t=t),pub,priv),pub)

if h!=1: return RST

return RST.r, RST.s, RST.t

done=0

while done==0: #цикл сделан для того, чтобы избежать перезапусков программы для возникаюзих assertion error (не моя ошибка, умышленная проблема реализации алгоритма автором))

pbits = 1024 # переформирую также ключи, для того, чтобы программа нормально работала. Если при конкретных ключах вылезает assertion error - то надо перезапускать. (это из-за метода автора и подхода генерации случайных ключей при перезапуске программы)

pub, priv = genkey(pbits)

enc\_flag=encrypt(bytes\_to\_long("HackTM{h4v3\_y0u\_r34lly\_f0und\_4ll\_7h3\_bu65...?}".encode()),pub)

try:

res = []

for i in range(1, 21):

rst = oracle(i, 1, 1)

if rst[0] is None:

continue

res.append(rst[0] - i)

factors = set()

for i in range(len(res)):

if res[i] == 0:

continue

for j in range(i + 1, len(res)):

if res[j] == 0:

continue

tmp = gcd(res[i], res[j])

if tmp > 2\*\*100:

for pi in primerange(1000):

while True:

if tmp % pi == 0:

tmp //= pi

else:

break

factors.add(tmp)

Приложение 6

assert len(factors) == 2

p = int(factors.pop())

q = int(factors.pop())

n = p \* q

print("Recover p, q, n:\np = %d \nq = %d \nn = %d"%(p,q,n))

r = None

for i in range(100):

rst = oracle(i, 1, 1)

if rst[0] is None: continue

if gcd(rst[0] - i, n) == 1:

r = i

break

assert r is not None

rs = []

for s in [-1, 1]:

for t in [0, 1]:

rs.append(oracle(r, s, t)[0])

for i in range(4):

for j in range(i + 1, 4):

r1 = rs[i]

r2 = rs[j]

try:

m1 = (r2 \* r - r \*\* 2) \* pow(r1 - 2 \* r + r2, -1, n) % n

c = (r1 \* m1 - m1 \*\* 2) % n

print(long\_to\_bytes(decrypt(enc\_flag, Pubkey(n=n, c=c), Privkey(p=p, q=q)))[0:46].decode())

except Exception as e:

print(e)

continue

done=1

except: pass

input()

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

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

ОНК «Институт высоких технологий»

ДНЕВНИК

учебной практики по получению первичных

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

г. Калининград 2023 г.

**1.Информационная часть**

Бакиновский Михаил Николаевич студент очной формы обучения 3 курса группы 05\_КБ\_20\_О\_/ специальности 10.05.01 Компьютерная безопасность, специализация «Математические методы защиты информации» в соответствии с приказом №2218-ст от 06 июня 2023 г. направляется на учебную практику по получению первичных профессиональных умений и навыков, в том числе первичных умений и навыков научно-исследовательской деятельности в Высшую школу компьютерных наук и прикладной математики образовательно-научного кластера "Институт высоких технологий".

Период практики – с 26.062023 г. по 08.07.2023 г.

Руководитель практики от университета – доцент ОНК «ИВТ» Киршанова Елена Алексеевна.

ОНК «Институт высоких технологий»

Контактный номер телефона +7 (4012) 338 217

Первый заместитель

директора ОНК «ИВТ» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Шпилевой А.А.

**2. Программа практики**

**2.1. План работы**

|  |  |  |
| --- | --- | --- |
| **№ п.п.** | **Рабочее место практиканта, вид работы** | **Продолжительность**  **(в днях)** |
| **1.** | Компьютерный класс, ауд. 230 | **14** |
|  |  |  |
|  |  |  |

**2.2. Индивидуальное задание по профилю подготовки/специальности**

1. Пройти инструктаж по технике безопасности.
2. Ознакомиться и выполнить задачи на практику.
3. Написать отчет по практике.

Руководитель практики от университета

доцент ОНК «ИВТ» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Киршанова Елена Алексеевна

**3. Ход выполнения практики**

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п.п.** | **Дата** | **Описание выполненной работы** | **Отметки**  **руководителя практики от профильной организации** |
| 1 | 26.06.2023 | Пройден инструктаж по технике безопасности |  |
| 2 | 27.06.2023-30.06.2023 | Решение задачи. Написание программы на языке программирования Python |  |
| 3 | 06.07.2023 | Создание презентации на основе системы верстки LaTeX |  |
| 4 | 06.07.2023 | Создание открытой онлайн репозитории с исходным кодом презентации и решения |  |
| 5 | 07.07.2023 | Подготовка отчета по учебной практике |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**4. Отзыв руководителя практики**

Бакиновский Михаил Николаевич студент очной формы обучения 3 курса группы 05\_КБ\_20\_О\_/ специальности 10.05.01 Компьютерная безопасность, специализация «Математические методы защиты информации» в соответствии с приказом №2218-ст от 06 июня 2023 г. направляется на учебную практику по получению первичных профессиональных умений и навыков, в том числе первичных умений и навыков научно-исследовательской деятельности в Высшую школу компьютерных наук и прикладной математики образовательно-научного кластера "Институт высоких технологий".

Период практики – с 26.06.2023 г. по 08.07.2023 г.

Программа практики и индивидуальное задание на практику выполнены. Отчёт по практике сдан и защищён на отчётной конференции.

Студент Бакиновский Михаил Николаевич в процессе прохождения практики справился с поставленными задачами, приобрёл первичные профессиональные навыки и компетенции, в том числе:

* способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности;
* способность к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения.

Учебная практика оценена на оценку \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель практики от университета –

доцент ОНК «ИВТ» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Киршанова Елена Алексеевна

«08» июля 2023 г.

1. Реципрокный - это математический термин, который означает обратную величину или число, обратное данному числу. [↑](#footnote-ref-1)