Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Лабораторная работа №7
по курсу "Защита иформации"

Тема Электронная подпись

Студент Нисуев Н. Ф.

Группа ИУ7-72Б

Преподаватель Руденкова Ю.С.

Москва — 2025 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ **3**

1 Аналитическая часть **4**

1.1 Электронная цифровая подпись 4

1.2 Алгоритм RSA 5

1.3 Алгоритм SHA1 6

2 Ответы на вопросы **8**

2.1 Схемы алгоритмов 8

3 Технологическая часть **10**

3.1 Средства реализации 10

3.2 Реализация алгоритма 10

3.3 Пример работы программы 12

ЗАКЛЮЧЕНИЕ **13**

ВВЕДЕНИЕ

Шифрование информации — занятие, которым человек занимался ещё до начала первого тысячелетия, занятие, позволяющее защитить информацию от посторонних лиц.

Криптографический алгоритм RSA — алгоритм, разработанный в 1977 году и положивший основу первой системе, пригодной как для шифрования, так и для цифровой подписи

Хеширование — процесс преобразования набора данных произвольной длины в выходной набор данных установленной длины, выполняемый определённым алгоритмом.

Целью данной работы является реализация в виде программы, позволяющую создать и проверить электронную подпись для документа с использованием алгоритма RSA и алгоритма хеширования SHA1.

Для достижения поставленной цели необходимо выполнить следующие задачи:

— изучить криптографический алгоритм RSA и алгоритм хеширования SHA1 или MD5;

— реализовать криптографический алгоритм RSA в виде программы, обеспечив возможности создания и проверки подлинности электронной подписи для документа с использованием алгоритма SHA1;

— протестировать разработанную программу, показать, что удаётся создавать и проверять электронные подписи.

1 Аналитическая часть

В этом разделе будут рассмотрен криптографический алгоритм RSA, алгоритм хеширования SHA1 и MD5, понятие электронной подписи и принципы её получения и проверки с использованием алгоритмов RSA, SHA1 и MD5.

1.1 Электронная цифровая подпись

Электронная (цифровая) подпись — ЭЦП — позволяет подвердить авторство электронного документа. Она связана не только с автором документа, но и с самим документов (при помощи криптографических методов) и не может быть подделана при поммощи обычного копирования.

Существует **три** основных вида электронной подписи: простая (ПЭП), неквалифицированная (НЭП) и квалифицированная (КЭП). Они различаются по уровню защиты и юридической силе, а КЭП признается равной собственноручной подписи без дополнительных условий.

— **Простая электронная подпись (ПЭП)** — Подтверждает, что документ подписал конкретный человек, но не гарантирует его неизменность.

— **Неквалифицированная электронная подпись (НЭП)** — Гарантирует, что документ не был изменен после подписания, но для придания ему юридической силы нужно заключить дополнительное соглашение между сторонами.

— **Квалифицированная электронная подпись (КЭП)** — Это самый защищенный вид подписи, который имеет силу собственноручной подписи без каких-либо дополнительных соглашений.

Создание ЭП с использованием криптографического алгоритма RSA и алгоритма хеширования SH1/MD5 происходит следующим образом:

1) происходит хеширование сообщения при помощи SH1/MD5, сообщение — файл, который неообходимо подписать;

2) происходит шифрование с использованием закрытого ключа RSA последовательности 128/160 бит, полученных на предыдущем этапе;

3) значение подписи — результат шифрования.

Проверка ЭП с использованием криптографического алгоритма RSA и алгоритма хеширования MD5 происходит следующим образом:

1) происходит хеширование сообщения при помощи SH1/MD5, сообщение — файл, подпись которого необходимо проверить;

2) происходит расшифровка подписи с использованием открытого ключа RSA;

3) происходит побитовая сверка значений, полученных на предыдущих этапах, если они одинаковы, подпись считается подлинной.

1.2 Алгоритм RSA

RSA (аббревиатура от фамилий Rivest, Shamir и Adleman) — ассиметричный алгоритм с открытым ключом, основывающийся на вычислительной сложности задачи факторизации больших полупростых чисел. В алгоритме RSA используется 2 ключа — открытый (публичный) и закрытый (приватный).

В ассиметричной криптографии и алгоритме RSA, в частности, открытый и закрытый ключи являются двумя частями одного целого и неразрывны друг с другом. Для шифрования информации используется открытый ключ, а для её расшифровки закрытый.

Криптосистема RSA стала первой системой, пригодной и для шифрования, и для цифровой подписи. Алгоритм используется в большом числе криптографических приложений, включая PGP, S/MIME, TLS/SSL, IPSEC/IKE и других.

RSA ключи генерируются следующим образом:

1) выбираются два отличающихся друг от друга случайных простых числа p и q, лежащие в установленнном диапазоне;

2) вычисляется их произведение $n = p \cdot q$, называемое модулем;

3) вычисляется значение функции Эйлера от числа n: $\phi(n) = (p-1) \cdot (q-1)$;

4) выбирается целое число e $(1 < e < \phi(n))$, взаимно простое со значением $\phi(n)$, оно называется открытой экспонентой;

5) вычисляется число $d = e^{-1} mod(\phi(n))$, оно называется закрытой экспонентой.

Пара (e, n) публикуются в качестве открытого ключа RSA, а пара (d, n) — в виде закрытого ключа.

Шифрование сообщения m $(0 < m < n - 1)$ в зашифрованное сообщение c производится по формуле $c = E(m, k_1) = E(m, n, e) = m^e mod(n)$.

Расшифрация: $m = D(c, k_2) = D(c, n, d) = c^d mod(n)$

У данного принципа имеется следующие минусы:

1) если $m = 0$, то и $c = 0$;

2) если $m_1 = m_2$, то и $c_1 = c_2$.

Из-за этого RSA используется для передачи ключей других шифров.

1.3 Алгоритм SHA1

SHA1 (англ. *Secure Hash Algorithm 1*) — алгоритм криптографического хеширования. Для входного сообщения произвольной длины алгоритм генерирует 160-битное (20 байт) хеш-значение, называемое также дайджестом сообщения, которое обычно отображается как шестнадцатеричное число длиной в 40 цифр.

Алгоритм получает на входе сообщение максимальной длины 264 бит и создает в качестве выхода дайджест сообщения длиной 160 бит.

Алгоритм состоит из следующих шагов:

— *Добавление недостающих битов.* Сообщение добавляется таким образом, чтобы его длина была кратна 448 по модулю 512 (длина $\equiv 448 \mod 512$). Добавление осуществляется всегда, даже если сообщение уже имеет нужную длину. Таким образом, число добавляемых битов находится в диапазоне от 1 до 512.

— *Расширение.* Результатом первых двух шагов является сообщение, длина которого кратна 512 битам. Расширенное сообщение может быть представлено как последовательность 512-битных блоков $Y_0, Y_1, ..., Y_{L-1}$, так что общая длина расширенного сообщения есть $L * 512$ бит. Таким образом, результат кратен шестнадцати 32-битным словам.

— *Инициализация SHA-1 буфера.* Используется 160-битный буфер для хранения промежуточных и окончательных результатов хэш-функции. Буфер может быть представлен как пять 32-битных регистров A, B, C, D и E. Эти регистры инициализируются следующими шестнадцатеричными числами: A = 67452301, B = EFCDAB89, C = 98BADCFE, D = 10325476, E = C3D2E1F0.

— *Обработка сообщения в 512-битных (16-словных) блоках.* Основой алгоритма является модуль, состоящий из 80 циклических обработок, обозначенный как H_{SHA}. Все 80 циклических обработок имеют одинаковую структуру.

— *Выход.* После обработки всех 512-битных блоков выходом L-ой стадии является 160-битный дайджест сообщения.

2 Ответы на вопросы

В этом разделе будут представлена схема алгоритма RSA и SHA1.

2.1 Схемы алгоритмов

Схема алгоритма RSA представлена на рисунке 2.1.

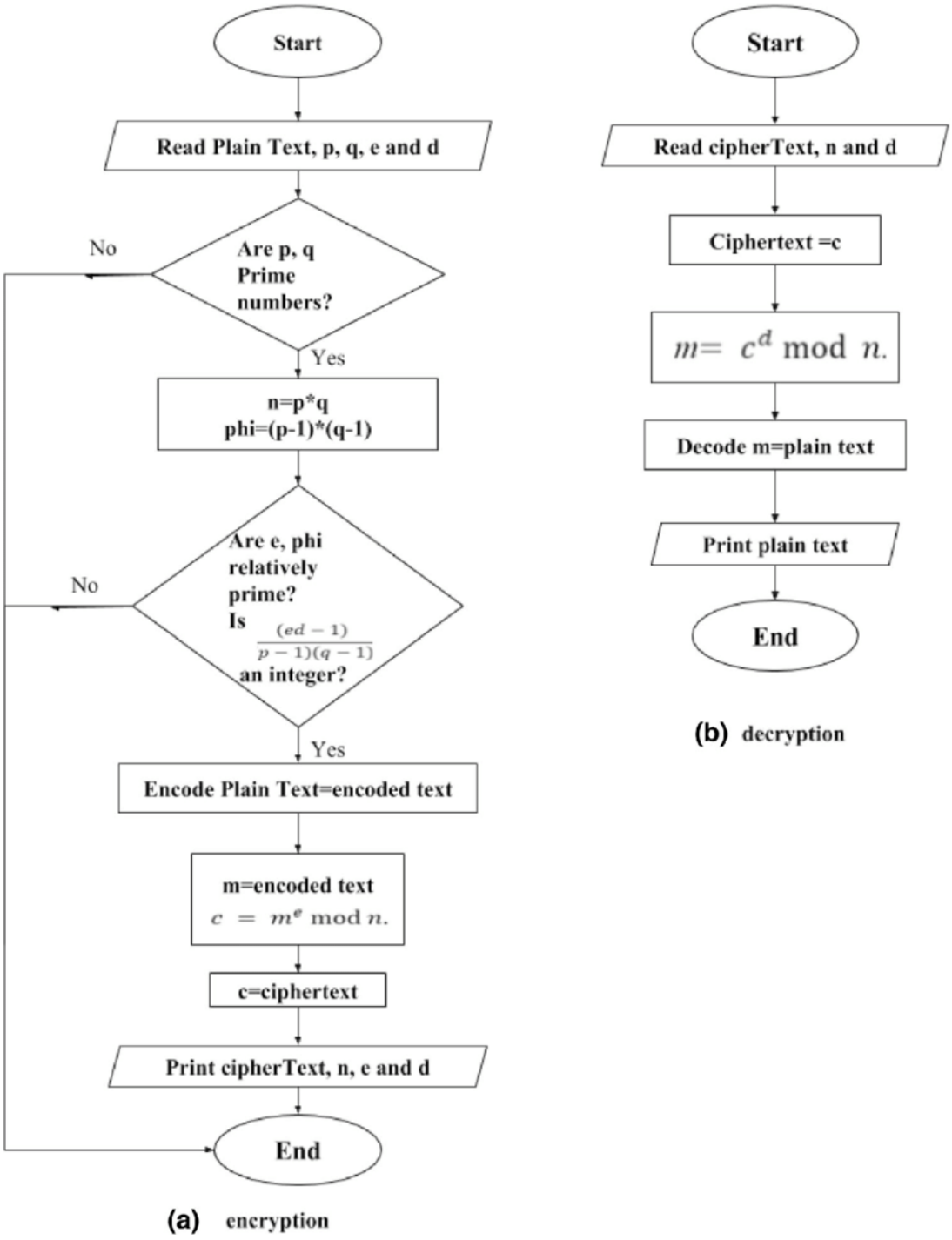

Рисунок 2.1 – Схема алгоритма шифрования RSA

Схема алгоритма SHA1 представлена на рисунке 2.2.

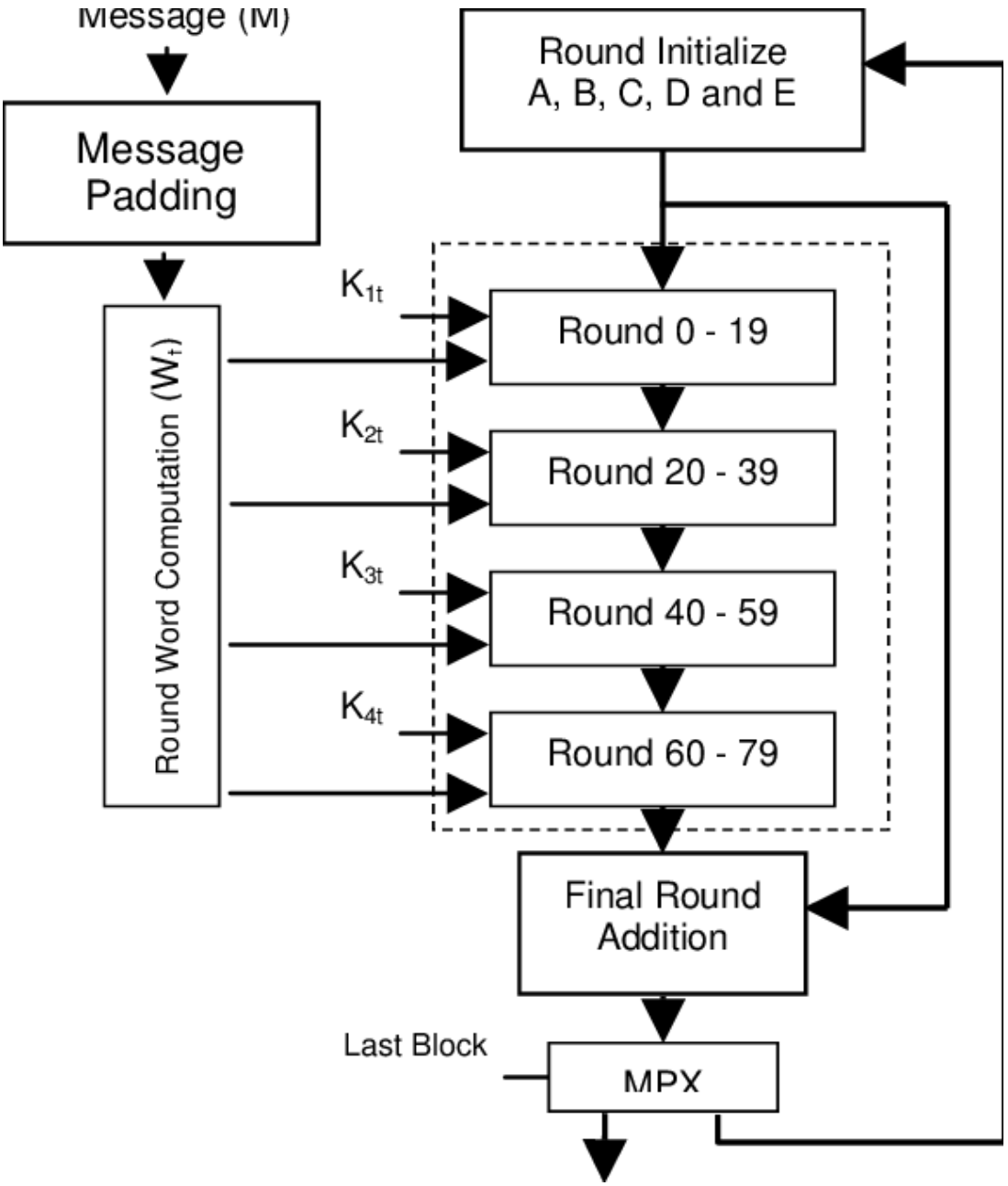

Рисунок 2.2 – Схема алгоритма SHA1

3 Технологическая часть

3.1 Средства реализации

Для программной реализации электронной подписи был выбран язык программирования *Python*. В данном языке есть все требующиеся инструменты для данной лабораторной работы.

3.2 Реализация алгоритма

В листингах 3.1 представлена реализация алгоритма электронной подписи.

Листинг 3.1 – Реализация электронной подписи

```
def generate_keys(private_key_path, public_key_path):
    key = RSA.generate(2048)
    private_key = key.export_key()
    public_key = key.publickey().export_key()

    with open(private_key_path, "wb") as f:
        f.write(private_key)
    with open(public_key_path, "wb") as f:
        f.write(public_key)

    print(f"Ключи сгенерированы:\n - Приватный: {private_key_path}\n - Публичный: {public_key_path}")

def sign_file(input_file, private_key_path, signature_file):
    with open(private_key_path, "rb") as f:
        private_key = RSA.import_key(f.read())

    with open(input_file, "rb") as f:
        data = f.read()

    h = SHA256.new(data)
    signature = pkcs1_15.new(private_key).sign(h)

    with open(signature_file, "wb") as f:
        f.write(signature)
```

```
25
26      print(f"Подпись сохранена в {signature_file}")
27
28  def verify_signature(input_file, public_key_path, signature_file):
29      with open(public_key_path, "rb") as f:
30          public_key = RSA.import_key(f.read())
31
32      with open(input_file, "rb") as f:
33          data = f.read()
34      with open(signature_file, "rb") as f:
35          signature = f.read()
36
37      h = SHA256.new(data)
38      try:
39          pkcs1_15.new(public_key).verify(h, signature)
40          print("Подпись действительна.")
41      except (ValueError, TypeError):
42          print("Подпись недействительна!")
```

3.3 Пример работы программы

На рисунке 3.1 представлен пример работы программы на текстовом фале.

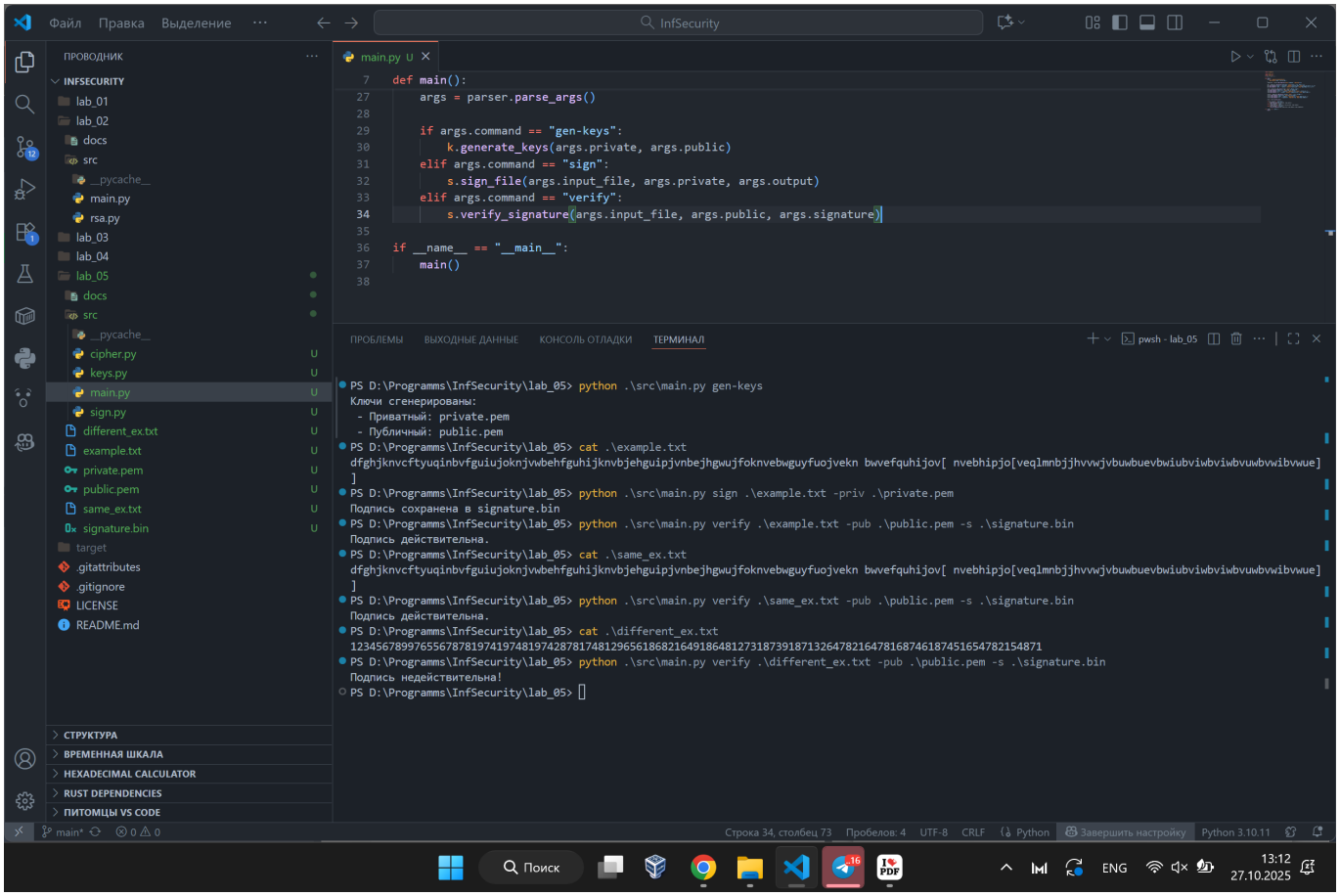

Рисунок 3.1 – Пример работы программы на текстовом файле

ЗАКЛЮЧЕНИЕ

В результате лабораторной работы был изучен алгоритма шифрования AES и разработана программная реализация. Были решены следующие задачи:

— изучены криптографический алгоритм RSA и алгоритм хеширования SHA1 или MD5;

— реализован криптографический алгоритм RSA в виде программы, обеспечивающий возможности создания и проверки подлинности электронной подписи для документа с использованием алгоритма SHA1;

— протестирована разработанная программа, показано, что удаётся создавать и проверять электронные подписи.