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

ФАКУЛЬТЕТ ИУ «Информатика и системы управления»

КАФЕДРА ИУ-7 «Программное обеспечение ЭВМ и информационные технологии»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4

по дисциплине «Защита информации»

«Электронная цифровая подпись. Алгоритм RSA с хешированием SHA1 и MD5»

Студент группы **ИУ7-76Б** _____ **В. М. Мансуров**
(Подпись, дата) (И.О. Фамилия)

Руководитель _____ **И. С. Чиж**
(Подпись, дата) (И.О. Фамилия)

2023 г.

СОДЕРЖАНИЕ

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 шифрованя, так и для цифрвой подпис

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

Целью данной работы является реализация в виде программы на языке программирования C или C++, позволяющую создать и проверить электронную подпись для документа с использованием алгоритма RSA и алгоритма хеширования SHA1 или MD5.

Для достижения поставленной цели необходимо выполнить следующие задачи:

1) изучить криптографический алгоритм RSA и алгоритм хеширования SHA1 или MD5;
2) реализовать криптографический алгоритм RSA в виде программы, обеспечив возможности создания и проверки подлинности электронной подписи для документа с использованием алгоритма SHA1 или MD5;
3) протестировать разработанную программу, показать, что удаётся создавать и проверять электронные подписи;
4) описать и обосновать полученные результаты в отчёте о выполненной лабораторной работе.

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.

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.

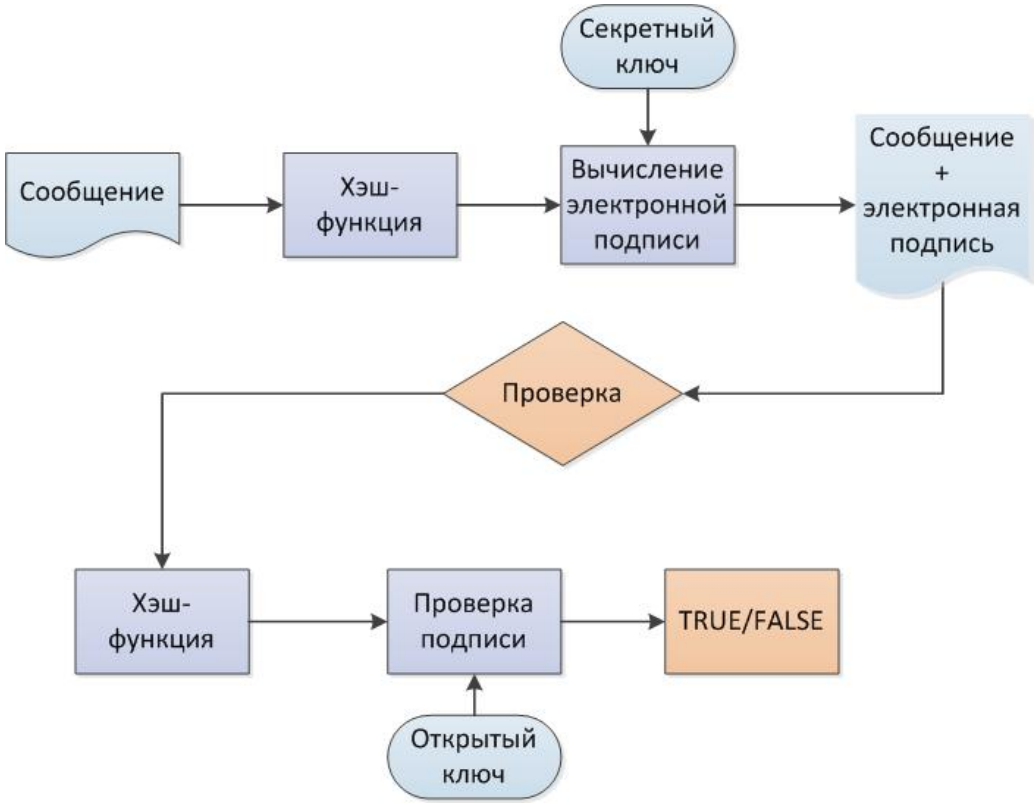

Рисунок 1.1 – RSA

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.

1.4 Алгоритм MD5

MD5 (англ. *Message Digest 5*) — алгоритм хеширования, предназначенный для получения последовательнсти длиной 128 бит, используемой для последующей проверки подлинности сообщений произвольных длины.

На вход алгоритма поступает последовательность бит произвольной

длины L, хеш которй нужно найти. Алгоритм MD5 состоит из 4 следующих этапов

1) выравнивание потоков;
2) добавление длины сообщения;
3) инициализация буфера;
4) вычисления в цикле.

Выравнивание потоков представляет из себя добавление некоторого числа нулевых бит такое, чтобы новая длина последовательности L' стала сравнима с 448 по модулю 512. Выравнивание происходит в любом случае, даже если длина исходного потока уже сравнима с 448

Под добавлением длины сообщения представляет из себя добавление 64 битов в последовательность: сначала младшие 4 байта, потом старшие 4 байта. После этого длина потока станет кратной 512. Вычисления будут основываться на представлении этого потока данных в виде массива слов по 512 бит.

После этого происходит инициализация буфера, состоящего из 4-х переменных размерностью 32 бита, начальные значения которых задаются шестнадцатеричными числами. В этих переменных будут храниться результаты промежуточных вычислений.

Далее в цикле каждый блок длиной 512 бит проходит 4 этапа вычислений по 16 раундов. Для этого блок представляется в виде массива X из 16 слов по 32 бита. Все раунды однотипны и имеют вид: [abcd k s i], определяемый как $a = b + ((a + Fun(b, c, d) + X[k] + T[i]) << s)$, где k — номер 32-битного слова из текущего блока, число s задаётся отдельно для каждого раунда, T — таблица констант.

Результат вычислений хранится в переменных a, b, c и d.

2 Конструкторская часть

В этом разделе будут представлены описания модулей программы, а также схема алгоритма шифроваания RSA, SHA1.

2.1 Разработка алгоритмов

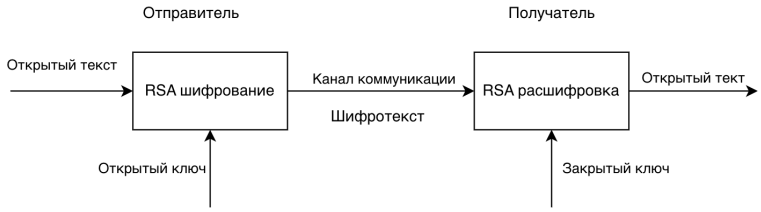

Рисунок 2.1 – Схема шифровального алгоритма RSA

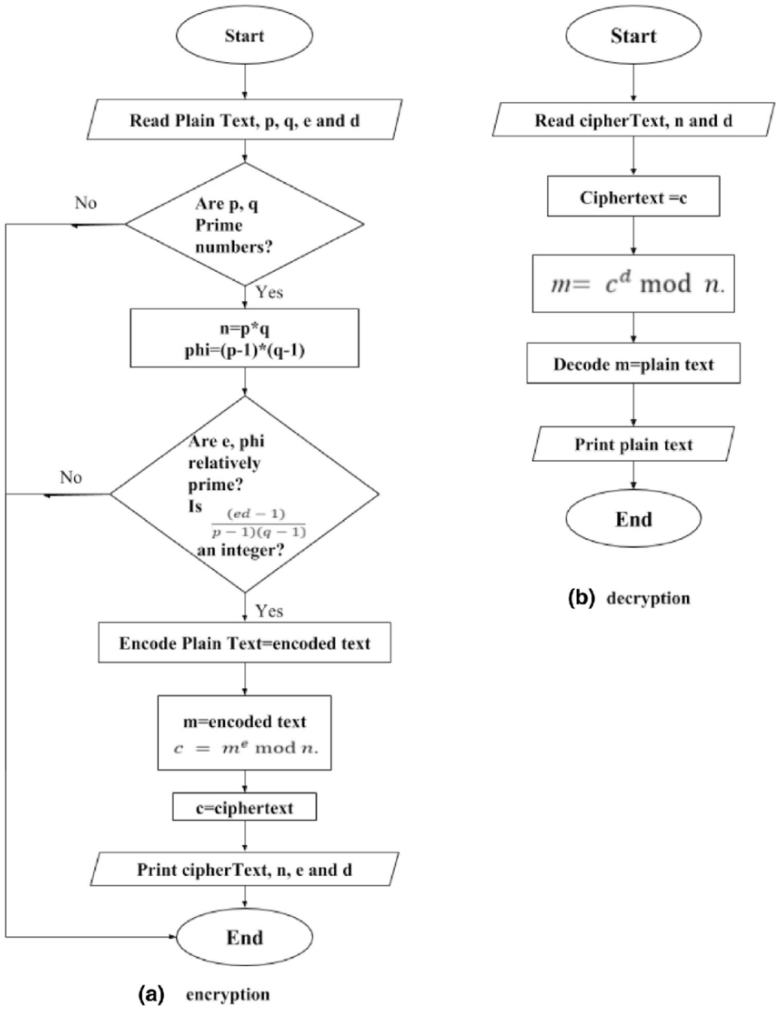

Рисунок 2.2 – Схема шифровального алгоритма работы RSA

Рисунок 2.3 – Схема алгоритма SHA1

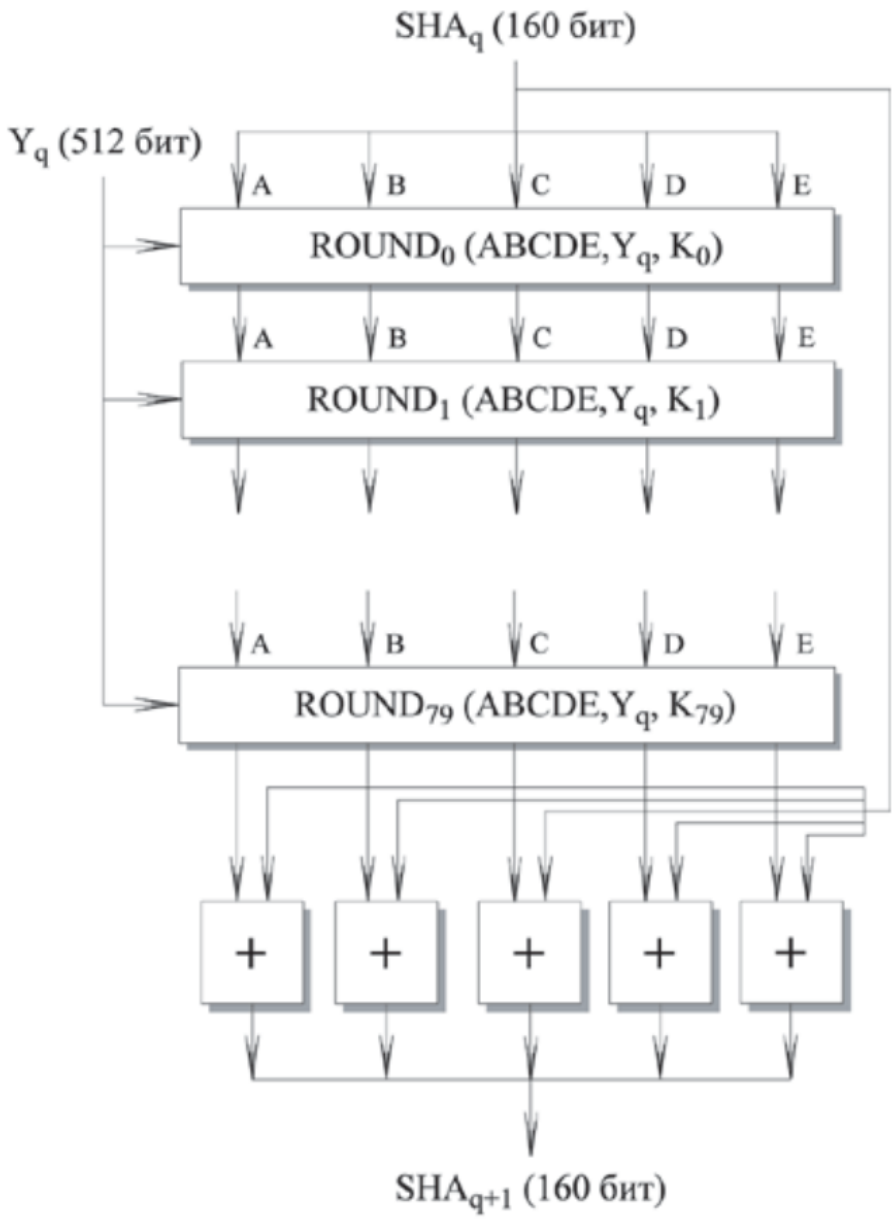

Рисунок 2.4 – Схема алгоритма раунда SHA1

3 Технологическая часть

3.1 Средства реализации

Для программной реализации шифровальной машины был выбран язык C++ [2]. В данном языке есть все требующиеся инструменты для данной лабораторной работы. В качестве среды разработки была выбрана среда CLion [3].

3.2 Реализация алгоритма

Листинг 3.1 – Генерация ключей RSA

```
1  std::pair<PrivateKey, PublicKey>
       RSA::genPublicAndSecretKeys(int64_t min_value, int64_t
       max_value) {
2      auto [p, q] = genPQSimple(min_value, max_value);
3
4      int64_t N = p * q;
5
6      int64_t euler = (p - 1) * (q - 1);
7
8      std::mt19937 generator(std::random_device{}());
9      std::uniform_int_distribution<std::int64_t> distribution(2,
           euler - 1);
10
11     int64_t e = distribution(generator);
12     while (gcd(e, euler) != 1) {
13         e = distribution(generator);
14     }
15
16     int64_t d = std::get<1>(extendet_efclid_alg(e, euler));
17     if (d < 0) {
```

```
18          d += euler;
19      }
20
21      return {
22          PrivateKey{
23              d, N
24          },
25          PublicKey{
26              e, N
27          }
28      };
29  }
```

Заключение

В результате лабораторной работы был реализована программа, позволяющая создать и проверить электронную подпись для документа с использованием алгоритма RSA и алгоритма хеширования MD5/SHA1.

Был и выполнены следующие задачи:

1) изучен криптографический алгоритм RSA и алгоритм хеширования MD5/SH
2) реализован криптографический алгоритм RSA в виде программы, обеспечив возможности создания и проверки подлинности электронной подписи для документа с использованием алгоритма MD5/SHA1;
3) протестирована разработанная программа;
4) описаны и обоснованы полученные результаты в отчёте о выполненной лабораторной работе.

Список использованных источников

1. И.М. Шолин. Алгоритм переносной шифровальной машины энигма. — Кубанский государственный технологический университет.

2. Язык программирования C++. `https://learn.microsoft.com/en-us/cpp/cpp/cpp-language-reference?view=msvc-170`. дата обращения: 15.10.2023.

3. CLion. `jetbrains.com`. дата обращения: 15.10.2023.