МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №12 НА ТЕМУ:**

**Исследование алгоритмов генерации и верификации электронной цифровой подписи**

Выполнила студентка 3 курса 4 группы

Коржова Валерия Сергеевна

Минск 2023

**Задание:** Разработать приложение, реализующее хеширование сообщений.

**Электронная цифровая подпись** (ЭЦП) – контрольная характеристика сообщения, которая вырабатывается с использованием личного ключа, проверяется с использованием открытого ключа, служит для контроля целостности и подлинности сообщения и обеспечивает невозможность отказа от авторства. ЭЦП представляет из себя бинарную последовательность, позволяющую однозначно идентифицировать отправителя.

Как правило, ЭЦП вычисляется на основе хеша подписываемого сообщения. То есть подпись зависит от содержания того, что ей подписывается.

**ЭЦП на основе RSA**

ЭЦП на основе RSA вычисляется по следующей формуле:

Где *d* и *n* – элементы тайного ключа отправителя,

H – хеш-функция,

*M* – сообщение.

При получении подписанного сообщения пользователь вычисляет H(*M*) и сравнивает его со значением:

Где *S* – подпись,

*e* и *n* – элементы открытого ключа отправителя.

Если H(*M*) и H’(*M*) совпадают, значит авторство отправителя подтверждено.

Класс на Python, реализующий ЭЦП на основе RSA:

class RSA:

    def \_\_init\_\_(self, p, q, e):

        self.p = p

        self.q = q

        self.n = p \* q

        self.e = e

        phi = (p - 1) \* (q - 1)

        self.d = gcd\_extended(e, phi)[1]

        if self.d < 0:

            self.d = phi + self.d

        self.public\_key = (self.n, self.e)

        self.private\_key = (self.n, self.d)

    def get\_hash(self, m: str):

        return int.from\_bytes(sha256(m.encode('utf-8')).digest(), 'big') % self.n

    def get\_signature(self, m):

        return (self.get\_hash(m) \*\* self.d) % self.n

    def check\_signature(self, m, s):

        return self.get\_hash(m) == ((s \*\* self.e) % self.n)

Листинг 1 – класс, реализующий ЭЦП на основе RSA

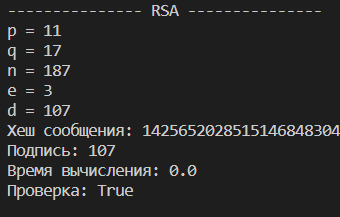


Рисунок 1 – проверка работы класса для ЭЦП на основе RSA

**ЭЦП на основе Эль-Гамаля**

За основу берётся алгоритм асинхронного шифрования Эль-Гамаля. При вычислении используется случайное число *k*, взаимно простое с *p – 1* (*p* – элемент ключевой информации в алгоритме Эль-Гамаля). ЭЦП состоит из двух блоков (*a* и *b*), вычисляемых по формулам:

Где *g, p, x* – элементы тайного ключа отправителя,

H – хеш-функция,

*M* – сообщение,

*a* – первый блок ЭЦП,

*b* – второй блок ЭЦП.

При получении подписанного сообщения пользователь вычисляет 2 параметра:

Где g, p, y – элементы открытого ключа отправителя,

H – хеш-функция,

*M* – сообщение,

*a* – первый блок ЭЦП,

*b* – второй блок ЭЦП.

Если значения *V* и *W*, совпадают, то отправитель верифицирован.

Класс на Python, реализующий ЭЦП на основе Эль-Гамаля:

class Elgamal:

    def \_\_init\_\_(self, p, g, x):

        self.p = p

        self.g = g

        self.x = x

        self.y = (g \*\* x) % p

        self.public\_key = (self.p, self.g, self.y)

        self.private\_key = (self.p, self.g, self.x)

    def get\_hash(self, m: str):

        return int.from\_bytes(sha256(m.encode('utf-8')).digest(), 'big') % self.p

    def get\_signature(self, m):

        h = self.get\_hash(m)

        k = 2

        while gcd\_extended(k, self.p - 1)[0] != 1:

            k += 1

        k\_r = gcd\_extended(k, self.p - 1)[1]

        a = (self.g \*\* k) % self.p

        b = ((h - self.x \* a) \* k\_r) % (self.p - 1)

        return (a, b)

    def check\_signature(self, m, s):

        h = self.get\_hash(m)

        a = s[0]

        b = s[1]

        return ((self.g \*\* h) % self.p) == ((self.y \*\* a) \* (a \*\* b)) % self.p

Листинг 2 – класс, реализующий ЭЦП на основе Эль-Гамаля

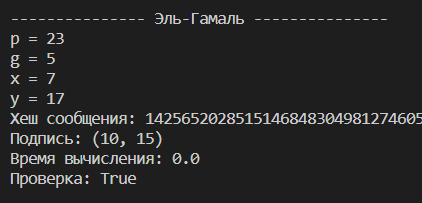


Рисунок 2 – проверка класса ЭЦП на основе Эль-Гамаля

**ЭЦП Шнорра**

ЭЦП Шнорра является вариантом алгоритма Эль-Гамаля. Подпись также состоит из двух блоков (*h* и *b*), но они вычисляются по другим формулам:

Где *p, g, q, x* – элементы тайного ключа отправителя,

*k* – случайное число, взаимно простое с *p – 1*,

H – хеш-функция,

*M* – сообщение.

При получении подписанного сообщения пользователь вычисляет параметр *X*:

Где p, g, y – элементы публичного ключа отправителя.

Если значение параметра *X* совпадает со значением блока *h* ЭЦП, отправитель верифицирован.

Класс на Python, реализующий ЭЦП Шнорра:

class Schnorr:

    def \_\_init\_\_(self, p, q, g, x):

        self.p = p

        self.q = q

        self.g = g

        self.x = x

        self.y = (g \*\* (q - x)) % p

        self.public\_key = (self.p, self.q, self.g, self.y)

        self.private\_key = (self.p, self.q, self.g, self.x)

    def get\_hash(self, m: str):

        return int.from\_bytes(sha256(m.encode('utf-8')).digest(), 'big') % self.p

    def get\_signature(self, m):

        k = 2

        while gcd\_extended(k, self.p - 1)[0] != 1:

            k += 1

        a = (self.g \*\* k) % self.p

        h = self.get\_hash(m + str(a))

        b = (k + self.x \* h) % self.q

        return (h, b)

    def check\_signature(self, m, s):

        h = s[0]

        b = s[1]

        X = ((self.g \*\* b) \* (self.y \*\* h)) % self.p

        return h == self.get\_hash(m + str(X))

Листинг 3 – класс, реализующий ЭЦП Шнорра

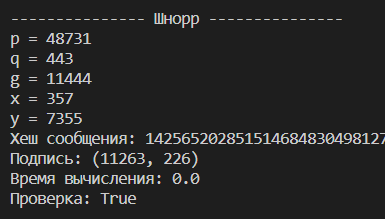


Рисунок 3 – проверка класса, реализующего ЭЦП Шнорра

**Контрольные вопросы:**

**1. Дать определение ЭЦП.**

**Электронная цифровая подпись** – бинарная последовательность, прикрепляемая к отправляемому документу, которая позволяет однозначно идентифицировать отправителя.

**2. Охарактеризовать основные функции ЭЦП.**

1. Верификация автора сообщения;
2. Контроль целостности сообщения;
3. Защита сообщений от подделки;
4. Доказательство авторства лица, подписавшего сообщение, если это лицо отрицает свое авторство.

**3. В чем заключаются сходства и различия между собственноручной и электронной подписью?**

Сходства заключаются в том, что ЭЦП выполняет те же функции, что и подпись от руки:

Отличия ЭЦП от ручной подписи:

• ЭЦП представляет собой бинарную последовательность.

• Бинарная последовательность зависит от содержания подписываемого сообщения.

**4.** **Охарактеризовать основные способы реализации ЭЦП.**

1. На основе симметричных систем.
2. На основе симметричных систем и посредника.
3. На основе асимметричных систем.

**5. Имеется ли различие в использовании ключевой информации при передаче зашифрованных сообщений и при передаче подписанных (ЭЦП) сообщений?**

При вычислении ЭЦП с помощью ассиметричной системы используется тайный ключ отправителя, в отличии от процесса шифрования, при котором используется публичный ключ получателя сообщения. Также стоит учесть, что подписанные сообщения можно передавать и в зашифрованном виде, так что перед отправкой сообщения могут быть одновременно использованы и публичный ключ получателя, и тайный ключ отправителя.

**6. Охарактеризовать криптостойкость ЭЦП на основе RSA, схемы Эль-Гамаля, схемы Шнорра.**

**RSA** является одним из самых широко используемых алгоритмов ЭЦП. Криптостойкость ЭЦП на основе RSA зависит от сложности факторизации больших чисел на простые множители. Для достижения достаточной криптостойкости, длина ключа RSA должна быть не менее 2048 бит. Однако, с учетом быстрого развития вычислительной техники, в будущем рекомендуется использовать более длинные ключи RSA.

**Схема Эль-Гамаля** основана на сложности дискретного логарифмирования. Криптостойкость ЭЦП на основе этой схемы также зависит от длины ключа. Рекомендуется использовать ключи длиной не менее 2048 бит.

**Схема Шнорра** также основана на сложности дискретного логарифмирования. Она обладает преимуществами перед RSA и схемой Эль-Гамаля в том, что она имеет более короткие ключи при той же криптостойкости. Однако, криптостойкость ЭЦП на основе схемы Шнорра также зависит от длины ключа. Рекомендуется использовать ключи длиной не менее 256 бит.

В целом, криптостойкость ЭЦП на основе RSA, схемы Эль-Гамаля, схемы Шнорра является достаточной для большинства практических задач.

**7. Какие элементы составляют ключевую информацию алгоритмов реализации ЭЦП, перечисленных в вопросе 6?**

Ключевая информация включает в себя параметры, используемые для генерации открытого и секретного ключей, а также сами открытый и секретный ключи. Открытый ключ является публичной информацией и может быть распространен широко. Секретный ключ, с другой стороны, должен храниться в безопасном месте и использоваться только владельцем ключа для подписи сообщений. Стоит отметить, что безопасность ЭЦП напрямую зависит от безопасности ключей. Поэтому генерация ключей должна быть проведена с учетом рекомендаций по безопасности и должна выполняться с помощью криптографически надежных методов.

**8. Дать сравнительные характеристики схемам ЭЦП, перечисленным в вопросе 6.**

**RSA:**

* Простой;
* Быстрый;
* Длина ключа может быть очень большой, что обеспечивает высокий уровень безопасности;
* Низкая производительность в режиме ЭЦП

**Схема Эль-Гамаля:**

* Производительнее RSA;
* Стойкость к атакам выше, чем у RSA;
* Длина ключа должна быть больше, чем в RSA, для обеспечения такого же уровня безопасности.

**Схема Шнорра:**

* Высокая производительность;
* Высокий уровень безопасности;
* Малая длина подписи и ключей.

**9. Охарактеризовать особенности государственного стандарта ЭЦП в Республике Беларусь.**

Согласно государственному стандарту РБ СТБ 34.101.45-2013 ЭЦП должна основывать на схеме Шнорра.

**Вывод:** В ходе лабораторной работы было разработано приложение, позволяющее подписывать сообщения с помощью ЭЦП на основе схем RSA, Эль-Гамаля и Шнорра, а также верифицировать полученные подписанные сообщения.