**Министерство науки и высшего образования Российской Федерации**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Факультет безопасности информационных технологий**

**Выпускная Квалификационная Работа**

**На тему:**

«Разработка интерактивной методики оценки эффективности систем цифровой подписи на основе библиотеки PyCryptodome»

**Выполнил:**

Гутник Дмитрий Вячеславович,

студент группы N34491

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(подпись)

**Проверил:**

Таранов Сергей Владимирович,

доцент факультета БИТ

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(отметка о выполнении)

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(подпись)

Санкт-Петербург

2023г.

Содержание

[Введение 3](#_Toc132750864)

[1 Анализ и разделение на виды актуальных алгоритмов цифровой подписи 5](#_Toc132750865)

[1.1 PKCS#1 v1.5: 5](#_Toc132750866)

[1.2 PKCS#1 PSS 11](#_Toc132750867)

[2 Описание критериев к методике оценки цифровых подписей 17](#_Toc132750868)

[3 Описание методики оценки эффективности пороговых подписей 18](#_Toc132750869)

[4 Описание программной реализации методики 19](#_Toc132750870)

[5 Рекомендации к практическому использования методики 20](#_Toc132750871)

[Заключение 21](#_Toc132750872)

[Список использованных источников 22](#_Toc132750873)

Введение

Современная информационная эпоха требует надежных и безопасных методов передачи данных. Одним из таких методов является использование систем цифровой подписи, которые позволяют обеспечить аутентичность и целостность информации. Однако, эффективность таких систем может быть ограничена различными факторами, такими как выбор алгоритма подписи, длина ключа и многими другими. В данном дипломном проекте рассматривается разработка интерактивной методики оценки эффективности систем цифровой подписи на основе библиотеки PyCryptodome. Эта методика позволит улучшить качество и безопасность передачи данных в современном информационном мире.

**Актуальность** дипломной работы заключается в том, что согласно TIOBE [[1]](#TIOBE) на апрель 2023 года Python является самым популярным языком программирования. Рейтинг самых популярных языков программирования расположен на рисунке 1

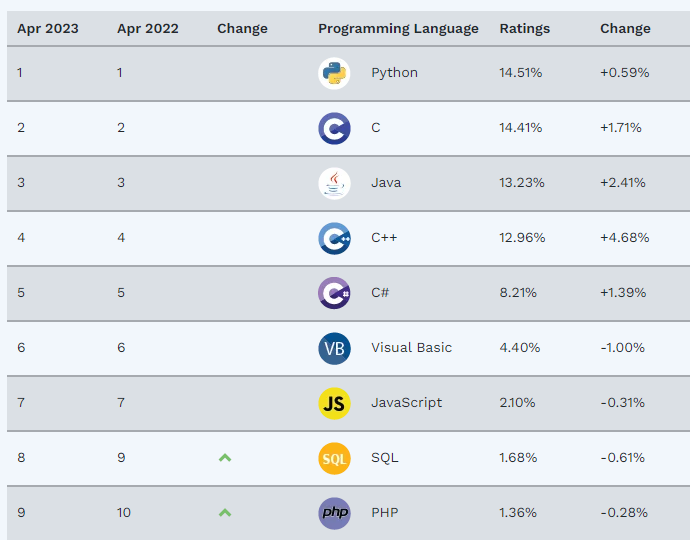


Рисунок 1 – Самые популярные языки программирования на апрель 2023 года [[1]](#TIOBE)

А PyCryptodome, в свою очередь, является одной из самых популярных библиотек [[2]](#tleapps) для имплементации криптографических протоколов и алгоритмов, эти факторы в купе с поддерживаемыми алгоритмами электронной подписи обусловливают выбор данной библиотеки в качестве базиса создание методики, а её пакета Crypto.Signature в качестве объекта исследования данной выпускной квалификационной работы .

**Цель работы** – Повышение эффективности применения алгоритмов цифровой подписи путём рекомендации подходящего вида ЦП в зависимости от условий применения

Для достижения поставленной цели необходимо выполнить следующие задачи:

1. Анализ и разделение на виды актуальных алгоритмов цифровой подписи

2. Разработка критериев к методике оценки цифровых подписей

3. Создание методики оценки эффективности пороговых подписей

4. Разработка программной реализации методики

5. Разработка рекомендаций к практическому использования методики

# Анализ и разделение на виды актуальных алгоритмов цифровой подписи

Согласно официальной документации [[3]](#documentation) Библиотека PyCryptodome поддерживает 3 алгоритма электронной подписи, каждый из которых имеет две вариации.

Алгоритмы:

1. RSA со следующими вариациями:

* PKCS#1 v1.5
* PKCS#1 PSS

1. EdDSA со следующими вариациями:

* EdDSA
* Pure EdDSA

1. DSA со следующими вариациями:

* DSA
* ECDSA

Ниже рассмотрены вышеперечисленные алгоритмы и их вариации согласно с RFC 8017 [[4]](#Rfc8017).

## PKCS#1 v1.5:

PKCS#1 v1.5, далее RSASSA-PKCS1-v15 объединяет примитивы RSASP1 и RSAVP1 с методом кодирования EMSA-PKCS1-v15. Он совместим со схемой IFSSA, определенной в IEEE [[5]](#ieee1363) 13639, где примитивы подписи и проверки являются IFSP-RSA1 и IFVP-RSA1, а метод кодирования сообщения - EMSA-PKCS1-v15

Длина сообщений, на которых может работать RSASSA-PKCS1-v15, либо не ограничена, либо ограничена очень большим числом, в зависимости от хэш-функции, лежащей в основе метода кодирования EMSA-PKCS1-v15.

Предполагая, что вычисление e-ых корней по модулю n является невыполнимым и хэш-функция в EMSA-PKCS1-v15 обладает соответствующими свойствами, предполагается, что RSASSA-PKCS1-v15 обеспечивает безопасную подпись. Более точно, подделка подписей без знания закрытого ключа RSA предполагается вычислительно невозможной. Кроме того, в методе кодирования EMSA-PKCS1-v15 идентификатор хэш-функции встроен в кодирование. Из-за этой особенности злоумышленник, пытающийся найти сообщение с той же подписью, что и ранее подписанное сообщение, должен найти коллизии конкретной используемой хэш-функции; атака на другую хэш-функцию, отличную от выбранной подписантом, не является полезной для злоумышленника.

Примечание: как отмечено в PKCS #1 v1.5, метод кодирования EMSA-PKCS1-v15 имеет свойство того, что закодированное сообщение, преобразованное в целочисленное представление сообщения, гарантированно большое и по крайней мере отчасти "случайное". Это предотвращает атаки, предложенные Десмедтом и Одлизко, где мультипликативные отношения между представителями сообщений разрабатываются путем факторизации представителей сообщений на набор малых значений (например, набор малых простых чисел). Корон, Накаш и Стерн показали, что более сильная форма этого типа атаки может быть довольно эффективной против некоторых случаев схемы подписи ISO/IEC 9796-2. Они также проанализировали сложность этого типа атаки на метод кодирования EMSA-PKCS1-v15 и пришли к выводу, что атака будет непрактичной, требуя больше операций, чем поиск коллизий в основной хэш-функции (т.е. более 2^80 операций). Coppersmith, Halevi и Jutla впоследствии расширили атаку Корон и др., Чтобы сломать схему подписи ISO/IEC 9796-1 с восстановлением сообщения. Различные атаки иллюстрируют важность тщательного построения входных данных для примитива RSA-подписи, особенно в схеме подписи с восстановлением сообщения. Следовательно, метод кодирования EMSA-PKCS-v15 явно включает хэш-операцию и не предназначен для схем подписи с восстановлением сообщения. Кроме того, хотя атака на метод кодирования EMSA-PKCS-v15 неизвестна, рекомендуется постепенный переход к EMSA-PSS в качестве предосторожности против будущих разработок.

Операция генерации подписи

(1)

Вход:

K - закрытый ключ RSA подписанта

M - сообщение для подписи, строка октетов

Выход:

S подпись, строка октетов длиной k, где k - длина в октетах модуля RSA n

Ошибки: "сообщение слишком длинное"; "модуль RSA слишком короткий"

Шаги:

1. Кодирование EMSA-PKCS1-v15: примените операцию кодирования EMSA-PKCS1-v15 к сообщению M, чтобы получить закодированное сообщение EM длиной k октетов:

Если операция кодирования выводит "сообщение слишком длинное", выведите "сообщение слишком длинное" и остановитесь. Если операция кодирования выводит "намеренная длина закодированного сообщения слишком короткая", выведите "модуль RSA слишком короткий" и остановитесь.

2. RSA-подпись:

a. Преобразование закодированного сообщения EM в целочисленное представление сообщения m:

(3)

b. Применение примитива подписи RSASP1 к закрытому ключу RSA K и целочисленному представлению сообщения m для получения целочисленного представителя подписи s:

(4)

c. Преобразование целочисленного представителя подписи s в подпись S длиной k октетов:

(5)

3. Вывод подписи S.

Операция проверки подписи

(6)

Вход:

(n, e) - открытый ключ RSA подписанта

M - сообщение, подпись которого требуется проверить, октетовая строка

S - подпись, которую необходимо проверить, октетовая строка длиной k,

где k - длина в октетах модуля RSA n

Выход: "верная подпись" или "неверная подпись"

Ошибки: "слишком длинное сообщение"; "слишком короткий модуль RSA".

Шаги:

1. Проверка длины: если длина подписи S не равна k октетам, вывести "неверная подпись" и остановиться.

2. Проверка RSA:

a. Преобразование подписи S в целочисленный представитель подписи s:

(7)

b. Применение примитива проверки RSAVP1 к открытому ключу RSA (n, e) и целочисленному представителю подписи s для получения целочисленного представителя сообщения m:

(8)

Если RSAVP1 выдает "целочисленный представитель подписи вне диапазона", вывести "неверная подпись" и остановиться.

c. Преобразование целочисленного представителя сообщения m в закодированное сообщение EM длиной k октетов:

(9)

Если I2OSP выдает "слишком большое целое число", вывести "неверная подпись" и остановиться.

3. Кодирование EMSA-PKCS1-v1\_5: Применение операции кодирования EMSA-PKCS1-v1\_5 к сообщению M для получения второго закодированного сообщения EM' длиной k октетов:

(10)

Если операция кодирования выдает "слишком длинное сообщение", вывести "слишком длинное сообщение" и остановиться. Если операция кодирования выдает "недостаточная длина закодированного сообщения", вывести "слишком короткий модуль RSA" и остановиться.

4. Сравнение закодированных сообщений EM и EM'. Если они совпадают, вывести "верная подпись"; в противном случае, вывести "неверная подпись".

Примечание: Другой способ реализации операции проверки подписи - применение "декодирования" (не указанного в данном документе) к закодированному сообщению для восстановления базового хеш-значения, а затем сравнение его с вновь вычисленным хеш-значением. Это имеет преимущество в том, что требуется меньше промежуточного хранения (два хеш-значения вместо двух закодированных сообщений), но недостаток в том, что требуется дополнительный код.

## PKCS#1 PSS

PKCS#1 PSS, далее RSASSA-PSS объединяет примитивы RSASP1 и RSAVP1 с методом кодирования EMSA-PSS. Он совместим с схемой подписи с приложением Integer Factorization Signature Scheme with Appendix (IFSSA), как это определено в IEEE [[5]](#ieee1363) 1363, где примитивы подписи и проверки - IFSP-RSA1 и IFVP-RSA1, а метод кодирования сообщения - EMSA4. EMSA4 немного более общий, чем EMSA-PSS, так как он действует на битовые строки, а не на октетные строки. EMSA-PSS эквивалентен EMSA4, ограниченному случаем, когда операнды, а также значения хеша и соли являются октетными строками.

Длина сообщений, на которых может работать RSASSA-PSS, либо неограничена, либо ограничена очень большим числом, в зависимости от хеш-функции, лежащей в основе метода кодирования EMSA-PSS.

Предполагая, что вычисление e-й корней по модулю n невозможно и функции генерации хеша и маски в EMSA-PSS имеют соответствующие свойства, RSASSA-PSS обеспечивает безопасные подписи. Это заверение может быть доказано в том смысле, что сложность подделки подписей может быть непосредственно связана со сложностью инвертирования функции RSA, при условии, что функции генерации хеша и маски рассматриваются как черные ящики или случайные оракулы. Границы в доказательстве безопасности являются в основном "жесткими", что означает, что вероятность успеха и время выполнения для лучшего фальсификатора против RSASSA-PSS очень близки к соответствующим параметрам для лучшего алгоритма инвертирования RSA.

В отличие от схемы подписи RSASSA-PKCS1-v1\_5, идентификатор хеш-функции не встроен в закодированное сообщение EMSA-PSS, поэтому в теории возможно, чтобы злоумышленник заменил выбранную подписантом хеш-функцию на другую (и потенциально более слабую). Поэтому рекомендуется, чтобы функция генерации маски EMSA-PSS была основана на той же хеш-функции. Таким образом, вся закодированная сообщение будет зависеть от хеш-функции, и для злоумышленника будет трудно заменить другую хеш-функцию, чем ту, которую задумал подписант. Это соответствие хеш-функций необходимо только для предотвращения замены хеш-функций и не является необходимым, если замена хеш-функций решается другими средствами (например, проверяющий принимает только определенную хеш-функцию). См. для дальнейшего обсуждения этих вопросов. Доказуемая безопасность RSASSA-PSS не зависит от того, является ли функция генерации маски хеш-функцией, применяемой к сообщению.

RSASSA-PSS отличается от других схем подписи на основе RSA тем, что он вероятностный, а не детерминированный, включая случайно генерируемое значение соли. Значение соли повышает безопасность схемы, предоставляя "более жесткое" доказательство безопасности, чем детерминированные альтернативы, такие как Full Domain Hashing (FDH). Однако случайность не является критической для безопасности. В ситуациях, когда случайная генерация невозможна, можно использовать фиксированное значение или номер последовательности, и результативная доказуемая безопасность будет аналогична FDH [FDH].

Операция генерации подписи

(11)

Входные данные:

K - закрытый ключ RSA подписчика

M - сообщение, которое должно быть подписано, октетовая строка

Выходные данные:

S - подпись, октетовая строка длиной k, где k - длина в октетах модуля RSA n

Ошибки: "сообщение слишком длинное"; "ошибка кодирования"

Шаги:

1. Кодирование EMSA-PSS: Применить операцию кодирования EMSA-PSS к сообщению M, чтобы получить закодированное сообщение EM длиной ceil ((modBits - 1)/8) октетов так, чтобы длина битового представления целого числа OS2IP (EM) была не больше modBits - 1, где modBits - длина в битах модуля RSA n:

(12)

Обратите внимание, что длина октетового представления EM будет на один меньше k, если modBits - 1 кратно 8, и равна k в противном случае.

Если операция кодирования выдает "сообщение слишком длинное", выдать "сообщение слишком длинное" и остановиться.

Если операция кодирования выдает "ошибка кодирования", выдать "ошибка кодирования" и остановиться.

2. RSA подпись:

a. Преобразование закодированного сообщения EM в представление целого числа m

(13)

b. Применение примитива подписи RSASP1 к закрытому ключу RSA K

и представлению целого числа сообщения m, чтобы получить представление целого числа подписи s:

(14)

c. Преобразование представления целого числа подписи s в подпись S длиной k октетов

(15)

3. Вывод подписи S.

Операция проверки подписи

(16)

Входные данные:

(n, e) - открытый ключ RSA подписчика

M - сообщение, подпись которого должна быть проверена, октетовая строка

S - подпись, которую нужно проверить, октетовая строка длиной k, где k - длина в октетах модуля RSA n

Выходные данные: "верная подпись" или "неверная подпись"

Шаги:

1. Проверка длины: если длина подписи S не равна k октетам, выдать "неверная подпись" и остановиться.

2. Проверка RSA подписи:

a. Преобразование подписи S в представление целого числа подписи s:

(17)

b. Применение примитива верификации RSAVP1 к открытому ключу RSA (n, e) и представлению целого числа подписи s, чтобы получить представление целого числа сообщения m:

(18)

Если RSAVP1 выдает "signature representative out of range", выдать "invalid signature"и остановиться.

c. Преобразование представления целого числа сообщения m в закодированное сообщение EM длиной emLen = \ceil ((modBits - 1)/8) октетов, где modBits - длина в битах модуля RSA n:

(19)

Обратите внимание, что если modBits - 1 кратно 8, то emLen будет на один меньше k,а в противном случае emLen равен k. Если I2OSP выдает "integer too large", выдать "invalid signature" и остановиться.

3. Проверка EMSA-PSS: Применить операцию проверки EMSA-PSS к сообщению M и закодированному сообщению EM, чтобы определить, соответствуют ли они друг другу:

(20)

4. Если Result = "consistent", выдать "valid signature". В противном случае выдать "invalid signature".

# Описание критериев к методике оценки цифровых подписей

# Описание методики оценки эффективности пороговых подписей

# Описание программной реализации методики

# Рекомендации к практическому использования методики

Заключение

Было выполнено то-се пятое-десятое…

Это позволили получить / закрепить навыки и т.п…

Мы все сделали, мы молодцы, поставьте нам **отлично автоматом**.

Список использованных источников

1. TIOBE Index for April 2023 – URL:

<https://www.tiobe.com/tiobe-index/>

1. 3 Best Python Encryption Libraries in 2023 – Url:

<https://tleapps.com/best-python-encryption-libraries/>

1. PyCryptodome’s documentation – URL:

<https://www.pycryptodome.org>

1. RFC 8017 – URL:

<https://datatracker.ietf.org/doc/html/rfc8017>

1. 1363-2000 - IEEE Standard Specifications for Public-Key Cryptography

<https://ieeexplore.ieee.org/document/891000>

1. Васильева И. Н. Криптографические методы защиты информации: учебник и практикум для академического бакалавриата / Москва: Юрайт, 2016. – 349 с.
2. Крамаров С. О., Криптографическая защита информации: Учебное пособие / Крамаров С. О., Митясова О.Ю., Соколов С. В., Тищенко Е.Н., Шевчук П.С., Под ред. Крамарова С.О - Минск: РИОР, 2018. – 324 с.
3. A. Menezes, P. van Oorschot and S. Vanstone. Handbook of Applied Cryptography. CRC Press 1996. – 816с.
4. R. Gennaro, S. Goldfeder. Fast Multiparty Threshold ECDSA with Fast Trustless Setup. Conference on Computer and Communications Security 2018. – 31c.