УЯЗВИМОСТИ АСИММЕТРИЧНЫХ ШИФРОВ БЛОКЧЕЙН-ПЛАТФОРМ

Ищукова Е. А.[1], Петренко С. А.[2], Леонтьева Ю. П.[3]

DOI: 10.21681/2311-3456-2025-6-35-47

Цель: *выявление потенциально уязвимых мест в реализациях асимметричных алгоритмов криптографии, основанных на использовании эллиптических кривых, применяемых в современных блокчейн-системах.*

Методы исследования: *основываются на использовании теории информации, теории криптографии и криптоанализа, математического аппарата теории вероятностей и математической статистики, теории построения блокчейн-систем, теории информационной безопасности.*

Результаты: *рассмотрены основные приемы, которые используются для построения асимметричных шифров, основанных на использовании эллиптических кривых. Рассмотрены алгоритм сложения двух точек, умножения точки на скаляр, определения второй координаты точки. Отдельно кратко рассмотрены свойства генераторов псевдослучайных последовательностей: принципы их построения и их влияние на стойкость асимметричных шифров, в составе которых они используются. В качестве базового алгоритма рассмотрен алгоритм цифровой подписи ECDSA, который используется в составе таких блокчейн-платформ как Bitcion, Litecoin, Ethereum и многих других.*

Научная новизна *заключается в рассмотрении ряда кейсов, моделирующих возникновение уязвимостей в асимметричной криптографии, используемой в современных блокчейн-системах. Для каждого кейса выполнено описание проблемы, сформулирована постановка задачи, приведено возможное решение и дана оценка его сложности. Показано, что при правильном использовании математического аппарата шифров, соблюдении требований к выбору стартовых параметров, отсутствии ошибок в программных реализациях, обеспечивается достаточная стойкость.*

Ключевые слова: *стойкость, алгоритм шифрования, функция хеширования, криптография, криптоанализ, приватный ключ, публичный ключ.*

Введение

Блокчейн технологии представляют собой разновидность построения систем распределенного реестра. Их отличительной особенностью является формирование единого связанного списка, в котором каждая следующая запись зависит от предыдущей. Это обеспечивает неизменяемость данных: изменение одной записи невозможно без изменения всех, связанных с ней [1, 2]. Формирование единого связанного списка достигается за счет использования механизмов криптографии [3–16]. В большинстве блокчейн систем используются два основных криптографических инструмента: асимметричная криптография на эллиптических кривых и функции хеширования. Функция хеширования используется для контроля целостности данных, сохраняемых в блокчейне. Также функции хеширования могут быть использованы для выстраивания связей между блоками цепочки, как это сделано, например, в механизме консенсуса Proof-of-Work [2]. Алгоритмы асимметричной криптографии используются в блокчейне для взаимодействия абонентов в недоверенной среде. У каждого абонента системы есть пара приватный-публичный ключ. Своим приватным ключом может подтвердить совершаемое в системе действие, управлять переводами криптовалютных средств (подтверждать, что данная транзакция назначена ему и он может ее потратить). Кроме того, через публичный ключ обычно определяется адрес пользователя в сети блокчейна. В разных блокчейнах используются разные алгоритмы выработки адреса, но, как правило, он вырабатывается из публичного ключа путем однократного или многократного хеширования с последующей перекодировкой. Известно, что одной из проблем современного блокчейна является проблема масштабируемости. Популярность криптографических алгоритмов на эллиптических кривых обусловлена высокой степенью безопасности при относительно малом размере ключей, что позволяет делать подписи транзакций сравнительно небольшими (в пределах 64 байт).

1 Ищукова Евгения Александровна, кандидат технических наук, ведущий научный сотрудник, Научный центр информационных технологий и искусственного интеллекта, Научно-технологический университет «Сириус», Адрес: Россия, 354340, Краснодарский край, Федеральная территория «Сириус», ORCID 0000-0002-6818-1608. E-mail: ischukova.ea@talantiuspeh.ru

2 Петренко Сергей Анатольевич, доктор технических наук, профессор, руководитель группы, Научный центр информационных технологий и искусственного интеллекта, Научно-технологический университет «Сириус», Адрес: Россия, 354340, Краснодарский край, Федеральная территория «Сириус», ORCID 0000-0003-0644-1731. E-mail: Petrenko.SA@talantiuspeh.ru

3 Леонтьева Юлия Павловна, студент, Институт компьютерных технологий и информационной безопасности, Южный федеральный университет, Россия. ORCID 0009-0006-4778-131X. E-mail: izavodnova@sfedu.ru

В 2022 году Национальный институт стандартов и технологий США NIST объявил о завершении конкурса постквантовой криптографии. Финалистами конкурса стали пять алгоритмов. Два алгоритма (CRYSTALS-KYBER и NTRU) предназначены для шифрования и обмена ключами. Три алгоритма (CRYSTALS-DILITHIUM, FALCON, SPHINCS+) предназначены для формирования и проверки цифровой подписи. Алгоритмы CRYSTALS-DILITHIUM и FALCON основаны на теории решеток, в то время как алгоритм SPHINCS+ сконструирован на основе функций хеширований. В то же время параметры, рекомендованные для данных алгоритмов шифрования, в настоящий момент имеют критически большие размерности в рамках использования в блокчейн-технологиях [17, 18]. Известно, что одной из проблем блокчейна является проблема постоянного роста хранимой базы данных. И при выборе криптографических примитивов важно использовать такие примитивы, которые при небольших размерах обеспечивают надежную защиту. Эллиптическая криптография в современных блокчейн платформах (например, в Bitcoin, Litecoin and Ethereum) оперирует ключами с размерностью 32 байта и вырабатывает цифровую подпись общим размером 64 байта. В то время как алгоритм CRYSTALS-DILITHIUM имеет публичный ключ размером 1,1 КБ (1024 байт), приватный ключ размером около 2,7 КБ (или 2700 байт) и вырабатывает подпись размером около 2.5 КБ (или 2,440 байт). Алгоритм FALCON имеет публичный ключ размером 1 КБ (800–1000 байт), приватный ключ размером около 1,6 КБ (или 1600 байт) и вырабатывает подпись размером около 887 байт. В алгоритме SPHINCS+ размер публичного ключа может варьироваться от 5 до 20 КБ в зависимости от выбранных параметров, приватный ключ может быть от 20 до 100 КБ в зависимости от конфигурации. Размер подписи для алгоритма SPHINCS+ варьируется в зависимости от выбранной параметризации, но, как правило, составляет около 20 КБ для стандартного уровня безопасности. Как видно из приведенного сравнения, параметры разработанных алгоритмов являются во много раз больше аналогичных параметров ассиметричной криптографии на эллиптических кривых. А если учесть, что в одном блоке блокчейна может находиться несколько тысяч транзакций, каждая из которых содержит подпись, то объемы блокчейна возрастут многократно.

В данной статье предлагается рассмотреть ряд кейсов, моделирующих возникновение потенциальных уязвимостей в асимметричной криптографии, используемой в современных блокчейн-системах.

1. Постановка задачи

На сегодняшний день не существует унифицированного подхода к определению криптографической стойкости. Первые понятия о стойкости шифров заложил К. Шенон. Он же дал определение абсолютно стойкого шифра, криптографические свойства которого не позволяют извлечь статистическую информацию относительно секретных ключей из перехватываемого шифра. Известно, что к абсолютно стойким шифрам на сегодняшний день относится только шифр Вернама. Все остальные криптографические алгоритмы являются условно стойкими. Обычно определяют криптографическую стойкость алгоритма относительно того, сколько времени уйдет на вскрытие сообщения или восстановление ключа, а также какова будет стоимость оборудования, необходимого для проведения анализа. При этом могут учитываться и другие параметры. Например, какой объем памяти необходим для проведения анализа, какое необходимо затратить количество энергии и др. Таким образом, различают условную криптографическую стойкость по времени вычислений (или по количеству совершаемых операций. Исходя из которого можно определить затрачиваемое время) и условную криптографическую стойкость по стоимости вычислений, но также могут быть использованы и дополнительные метрики [19].

Целью настоящей работы является выявление потенциально уязвимых мест в реализациях асимметричных алгоритмов криптографии, основанных на использовании эллиптических кривых, применяемых в современных блокчейн-системах. Для достижения поставленной цели необходимо:

1. Выявить основные алгоритмы, лежащие в основе асимметричных алгоритмов, используемых в современных блокчейн системах.
2. Сформулировать кейсы, моделирующие возникновение уязвимостей в асимметричной криптографии, используемой в современных блокчейн-системах. Для каждого кейса выполнить описание проблемы, сформулировать постановку задачи, привести возможное решение
3. Для каждого сформулированного кейса дать оценку его сложности и по возможности провести численное моделирование.

2. Объект исследования

Объектом исследования являются асимметричные алгоритмы, использующие эллиптическую криптографию. В первую очередь фокус сосредоточен на алгоритме ECDSA, который используется в составе таких блокчейн-платформ как Bitcion, Litecoin, Ethereum и многих других.

Также к объектам исследования стоит отнести генераторы псевдослучайных последовательностей (ПСП). В блокчейн системах генераторы ПСП используются для генерации случайных чисел при выработке ключей для используемого асимметричного

алгоритма шифрования, а также при использовании алгоритмов электронной подписи. От того насколько стойкий генератор ПСП используется при этом, напрямую зависит стойкость самой блокчейн-системы по отношению к взлому (получения доступа к управлению активами пользователя) или подлогу информации.

3. Генераторы ПСП

Под генератором ПСП понимается алгоритм, на выходе которого образуется битовая последовательность на первый в згляд кажущаяся случайной. Псевдослучайность заключается в том, что рано или поздно битовая последовательность начнет повторяться. Длина ПСП до начала ее повторения называется периодом. Стойкие ПСП должны иметь как можно больший период с тем, чтобы аналитику было сложно отличить вырабатываемую последовательность от случайной, а также чтобы не давать аналитику возможности предсказать появление следующих символов вырабатываемой последовательности на основе анализа предыдущих выработанных значений [20]. Для проверки на случайность используют различные статистические тесты [21].

Выделяют два основных способа построения генераторов ПСП – аппаратные и программные. В виду того, что настоящая работа нацелена на исследование в области блокчейн технологий, то в данном случае особый интерес представляют программные реализации генераторов ПСП. Считается, что построение качественных программных генераторов ПСП является задачей более сложной, нежели построение аппаратных генераторов. Обычно в качестве стартовых параметров программные генераторы ПСП могут использовать случайные системные процессы, такие как: системное время; особенности клавиатурного почерка пользователя; данные, вводимые пользователем; параметры операционной системы или содержимое буфера. Здесь важным является применение правила: хороший генератор должен использовать много разных источников случайности, комбинировать их и менять их. Известны случаи, когда, например, использование только одного параметра приводило систему к уязвимости, несмотря на качество самого используемого генератора. Так, в 2019 году была обнаружена уязвимость в менеджере паролей от Лаборатории Касперского (занесена в реестр под номером CVE-2020-27020) в связи с тем, что стартовое значение генератора зависело только от системного времени.

Кейс № 1.

Описание задачи: Оценить качество реализации выбранного генератора ПСП. Определить с какой вероятностью p генератор вырабатывает значения $k1$ и $k2$, отстоящие друг от друга не более чем d позиций.

Постановка задачи: Имеется реализация генератора ПСП, выполняющая преобразование $k = Gen()$. Определить вероятность p, с которой $|k_i - k_j| \le d$, где $i, j = 1, \ldots, n$.

Решение:

Вход: Генератор $k = Gen()$; расстояние d; количество рассматриваемых чисел n.

Выход: Вероятность p ($|k_i - k_j| \le d$, где $i, j = 1, \ldots, n$.

Данный кейс сводится к задаче определения математической статистики распределения формируемых случайным образом чисел по числовой оси. Чем больше будет накопленная статистика, тем точнее будет полученный результат. Для того, чтобы формируемые генератором значения могли рассматриваться как уязвимые, расстояние d между случайно сгенерированными точками должно быть много меньше размерности используемого модуля и доступно для перебора (например, в диапазоне от 232 до 264). При этом вероятность формирования такой разности должна быть не менее 0,5. Будем использовать в алгоритме n точек. Чем больше n, тем точнее определено значение вероятности. При этом количество рассмотренных комбинаций составит $\frac{n(n-1)}{2}$.

Алгоритм 1:

1. Инициализировать переменные $k[0] = Gen(); Sum = 0;$
2. Для всех i от 1 до n:
   2.1. $k[i] = Gen();$
   2.2. Для всех j от 0 до i-1:
      2.2.1. Если ($|k[i] - k[j]| \le d$), то $Sum = Sum + 1;$
3. $p = \frac{2 \times Sum}{n(n-1)}$.

4. Асимметричная криптография на основе эллиптических кривых

Для асимметричных шифров на основе эллиптической криптографии используются эллиптические кривые двух видов: бинарные эллиптические кривые и эллиптические кривые, ограниченные модулем простого числа p. Дадим определение каждому из видов кривой.

Эллиптические кривые в простом поле ограниченны модулем простого числа p, задаются в форме Вейерштрасса, которая имеют следующий вид:

$$y^2 = x^3 + ax + b, \quad (1)$$

где $a, b \in F_p$ и $4a^3 + 27b^2 \neq 0$. При этом все действия с кривой ограничиваются модулем p.

Эллиптические кривые в бинарном поле рассматриваются над конечным полем F_q, где $q = 2m$ и задаются уравнением вида:

$$y + xy = x^3 + ax^2 + b, \quad (2)$$

где $a, b \in F_p$.

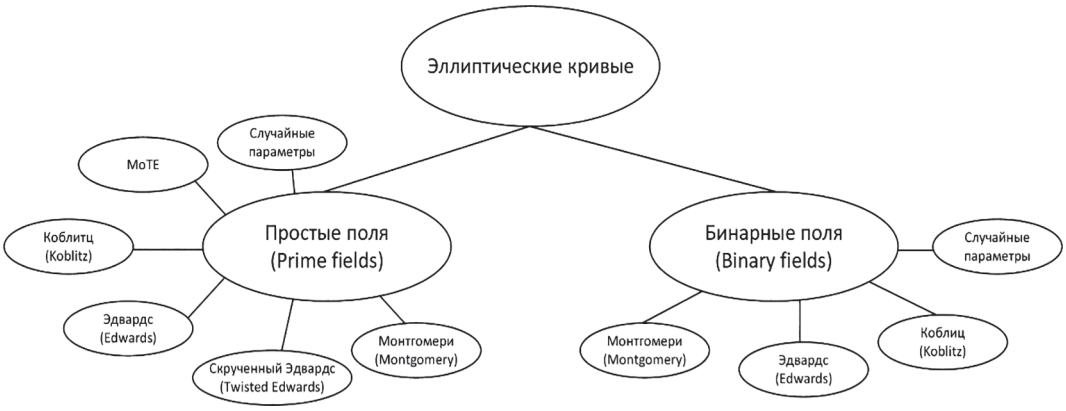

Рис. 1. Виды эллиптических кривых

В зависимости от того, как задаются параметры, различают следующие виды эллиптических кривых: кривая Монтгомери, кривая Коблица, кривая Эдвардса, скрученная кривая Эдвардса, кривая MoTE и кривая со случайными параметрами. Схематично весь спектр используемых эллиптических кривых представлен на рис. 1, а более детальное описание приведено в работах [22–25].

В настоящей работе мы ограничимся рассмотрением эллиптических кривых Коблица вида secp256k1, так как именно они применяются в самых известных блокчейн платформах, таких как Bitcoin и Ethereum. Кривая Коблица задается в виде уравнения (1) в случае использования эллиптических кривых в простом поле и в виде уравнения (2) в случае использования эллиптических кривых в бинарном поле при условии, что параметр $a \in F_2$.

В общем случае эллиптические кривые над полем F_p задаются в виде кортежа:

$$T = (p, a, b, G, n, h),$$

где p – большое целое число, определяющее конечное поле F_p; a и b – параметры уравнений (1) или (2); G – базовая точка эллиптической кривой, которая задается в виде двух координат (x, y); n – простое число, определяющее порядок базовой точки G; h – кофактор подгруппы.

Приведем рекомендованные параметры кортежа T в табл. 1. Для всех рассматриваемых кривых $h = 01$.

Порядок группы определяет количество всех точек на эллиптической кривой в соответствии с заданным модулем p и может быть вычислен в соответствии с теоремой Хассе. Параметр n определяет порядок базовой точки G, то есть показывает количество в циклической подгруппе, порожденной точкой G. Более подробную информацию о параметре n можно найти в стандарте SEC2 . В общем случае моделирование изменения параметра порядка группы для кривой вида $y^2 = x^3 + 7$ при малых размерностях модуля p отражен на графике рис. 2 и показывает, что разные точки эллиптической кривой будут обеспечивать разные уровни криптографичекой стойкости.

Рассмотрим основные математические приемы, которые используются в эллиптической криптографии. В эллиптической криптографии над точками производятся операции сложения и умножение точки на скаляр. При этом умножение точки на скаляр выполняется как многократное сложение разных точек или удвоения одинаковых. В настоящей работе будет рассмотрен классический способ умножения точки на скаляр.

Алгоритм «Сложение двух разных точек эллиптической кривой»

Случай 1. Вход: Эллиптическая кривая в простом поле вида (1). Две точки эллиптической кривой $P(x_1, y_1)$ и $Q(x_2, y_2)$, при этом $P, Q \in F_p$; $P \neq Q$, $x_1 \neq x_2$.

Таблица 1.

Рекомендованные параметры эллиптических кривых

Эллиптич. кривая	Модуль p	Уравнение	G
secp160k1	$2^{160}-2^{32}-2^{14}-2^{12}-2^{9}-2^{8}-2^{7}-2^{3}-2^{2}-1$	$y^2 = x^3 + 7$	02 3B4C382C E37AA192 A4019E76 3036F4F5 DD4D7EBB
secp224k1	$2^{224}-2^{32}-2^{12}-2^{11}-2^{9}-2^{7}-2^{4}-2-1$	$y^2 = x^3 + 5$	03 A1455B33 4DF099DF 30FC28A1 69A467E9 E47075A9 0F7E650E B6B7A45C
secp256k1	$2^{256}-2^{32}-2^{9}-2^{8}-2^{7}-2^{6}-2^{4}-1$	$y^2 = x^3 + 7$	02 79BE667E F9DCBBAC 55A06295 CE870B07 029BCFDB 2DCE28D9 59F2815B 16F81798

Рис. 2. График зависимости отклонения порядка группы n от модуля p

Действия:

1. $\lambda = \frac{y_2 - y_1}{x_2 - x_1}$;
2. $x_3 = \lambda^2 - x_1 - x_2$;
3. $y_3 = \lambda^2(x_1 - x_3) - y_1$.

Выход: Точка $R(x_3, y_3)$, для которой справедливо $R = P + Q$, $R \in F_p$.

Случай 2.

Вход: Эллиптическая кривая в простом поле вида (1). Две одинаковые точки эллиптической кривой $P(x_1, y_1)$ и $P(x_1, y_1)$, при этом $R \in F_p$, $y_1 \neq 0$.

Действия:

1. $\lambda = \frac{3x_1^2 + a}{2y_1}$;
2. $x_3 = \lambda^2 - 2x_1$;
3. $y_3 = \lambda^2(x_1 - x_3) - y_1$.

Выход: Точка $R(x_3, y_3)$, для которой справедливо $R = P + P = 2P$, $R \in F_p$.

Случай 3.

Вход: Эллиптическая кривая в бинарном поле вида (2). Две точки эллиптической кривой $P(x_1, y_1)$ и $Q(x_2, y_2)$, при этом P, $Q \in F_{2^m}$; $P \neq Q$, $x_1 \neq x_2$.

Действия:

1. $\lambda = \frac{y_1 + y_2}{x_1 + x_2}$;
2. $x_3 = \lambda^2 + \lambda + x_1 + x_2 + a$;
3. $y_3 = \lambda(x_1 + x_3) + x_3 + y_1$.

Выход: Точка $R(x_3, y_3)$, для которой справедливо $R = P + Q$; $R \in F_{2^m}$.

Случай 4.

Вход: Эллиптическая кривая в бинарном поле вида (2). Две одинаковые точки эллиптической кривой $P(x_1, y_1)$ и $P(x_1, y_1)$, при этом $P \in F_{2^m}$, $y_1 \neq 0$.

Действия:

1. $x_3 = \frac{x_1^2 + b}{x_1^2}$;
2. $y_3 = x_1^2 + \frac{x_1 + y_1}{x_1}x_3 + x_3$.

Выход: Точка $R(x_3, y_3)$, для которой справедливо $R = P + P = 2P$, $R \in F_{2^m}$.

Случай 5.

Вход: Эллиптическая кривая любого вида (1) или (2). Две точки эллиптической кривой $P(x_1, y_1)$ и $Q(x_2, y_2)$, при этом $P \neq Q$, $x_1 = x_2$.

Выход: Точка $R(x_3, y_3) = P + Q = \infty$ (нейтральный элемент).

Случай 6.

Вход: Эллиптическая кривая любого вида (1) или (2). Две точки эллиптической кривой $P(x_1, y_1)$, при этом $y_1 = 0$.

Выход: Точка $R(x_3, y_3) = 2P = \infty$ (нейтральный элемент).

Случай 7.

Вход: Эллиптическая кривая любого вида (1) или (2). Две точки эллиптической кривой $P(x_1, y_1)$ и $Q = \infty$ (нейтральный элемент).

Выход: Точка $R(x_3, y_3) = P + Q = P + \infty = P$.

Алгоритм «Умножение точки эллиптической кривой на скаляр»

Вход: Эллиптическая кривая в простом поле вида (1). Точка эллиптической кривой $P(x_1, y_1)$, при этом $P \in F_p$; скаляр k – целое число, которое можно представить в виде битовой последовательности $k = (k_{m-1}, k_{m-2}, \ldots, k_1, k_0)$.

Действия:

1. Инициализация данных. Точка $R = (0, 0)$.
   Точка $S = (x_1, y_1)$.
2. Для всех i от 0 до $m - 1$
   2.1. Если $k_i = 0$, то
   2.1.1 Если $R = 0$, то $R = S$ иначе $R = R + S$
   2.2 $S = S + S$

Выход: Точка $R = kP$, $R \in F_p$.

Часто для экономии пространства точка эллиптической кривой сохраняет только одну координату X.

Так как эллиптическая кривая задается квадратным уравнением, то всегда существует две координаты Y для одного заданного X. Для того, чтобы избежать путаницы, перед координатой X помещают дополнительный байт b и вместе они образуют последовательность в 33 байта $(b||X)$. Если $b = 02$, то координата Y четная, если $b = 03$, то координата Y нечетная. Для одного и того же X координаты Y всегда будут образовывать пару из четного и нечетного числа в силу того, что операции в поле ограничены нечетным числом. Также первый байт b может быть равен 04. Обычно это означает, что точка сохранена в развернутом виде с обеими координатами.

Алгоритм «Восстановление координаты y для точки эллиптической кривой»

Вход: Эллиптическая кривая в простом поле вида (1). Координата точки эллиптической кривой $(b||x)$.

Действия:

1. $y = \sqrt{(x^3 + ax + b)} \bmod p$.
2. Если $((b = 2)$ и $((y \,\&\, 1) = 1))$, то $y = p - y$.

Выход: Точка $R = (x, y)$.

Теперь с использованием введённых операций над точками эллиптических кривых рассмотрим сами криптографические примитивы. В данной части предлагается рассмотреть новый авторский вариант использования эллиптических кривых для передачи данных в зашифрованном виде, а также известный классический алгоритм ECDSA для создания и проверки электронной подписи.

Алгоритм электронной подписи ECDSA

Вход: Эллиптическая кривая в простом поле вида (1) с кортежем параметров $T = (p, a, b, G, n, h)$, общим для двух пользователей системы A и B, сообщение M, $h(M)$ – заданная хеш-функция.

Генерация ключа (Выполняет пользователь A):

1. Генерируется случайное число d_A не больше величины n. Число d_A – приватный ключ пользователя A.
2. Вычисляется точка $Q_A = d_A G$. Точка Q_A – публичный ключ пользователя A.
3. Пользователю B сообщается публичный ключ Q_A.

Подпись (Выполняет пользователь A):

1. Генерируется число k_s не больше величины n. Число k_s – сессионный ключ, каждый раз разный для подписи сообщений M.
2. Вычисляется точку $k_s G = (x_1, y_1)$ и $r = x_1 \bmod n$.
3. Если $r = 0$, то необходимо вернуться к шагу 1
4. Вычисляется $t = k_s - 1 \bmod n$ (при помощи расширенного алгоритма Евклида).
5. Вычисляется $z = h(M)$, результат представляется в виде большого целого числа.
6. Вычисляется $s = t(z + rd_A) \bmod n$.
7. Если $s = 0$, то необходимо вернуться к шагу 1
8. Пользователю B пересылается сообщение M и подпись к нему в виде пары значений (r, s).

Выход: Подпись = (r, s).

Проверка подписи (Выполняет пользователь B):

1. Выполняется проверка, что числа r и s лежат в диапазоне от 1 до $n - 1$.
2. Вычисляется $z = h(M)$, результат представляется в виде большого целого числа.
3. Вычисляет $w = s^{-1} \bmod n$ (при помощи расширенного алгоритма Евклида).
4. Вычисляет $u = zw \bmod n$.
5. Вычисляет $v = rw \bmod n$.
6. Вычисляет точку $(x, y) = uG + vQ_A$
7. Если $r \neq x \bmod n$, то подпись некорректна.

Выход: Точка $uG + vQ_A = uG + vd_A G = (u + vd_A)G = (zw + rwd_A)G = (z + rd_A)wG = (z + rd_A)s^{-1}G = (z + rd_A)s^{-1}G = st^{-1}s^{-1}G = t^{-1}G = (k_s^{-1})^{-1}G = k_s G$.

Известно, что две самые известные платформы Bitcoin и Ethereum используют одну и ту же эллиптическую кривую secp256k1 с одинаковыми параметрами (табл. 1). Это означает, что данные платформы будут вырабатывать одинаковые пары ключей. В этом легко можно убедиться с использованием любого онлайн генератора ключей для данных платформ (например, www.rfctools.com). На рис. 3 показан результат генерации ключей для обеих платформ. В верхней части окно для генерации ключей платформы Эфириум, а в нижней части – окно для генерации ключей платформы Биткоин. Видно, что при одном и том же заданном параметре d сгенерированы одинаковые ключи с той лишь разницей, что для платформы Биткоин ключ представлен в сжатом виде и имеет первый добавленный байт 02, означающий, что координата y будет иметь четное значение.

Проблема: с точки зрения криптографической стойкости для блокчейн систем первоочередную роль играют способы извлечения или подбора приватного ключа пользователя системы, а также сценарии подмены подписи в сообщениях.

Описание общей задачи: В общем случае задача дискретного логарифмирования сводится к следующему. Задана эллиптическая кривая с кортежем параметров $T = (p, a, b, G, n, h)$. Для двух точек P и Q необходимо найти такое целочисленное d, для которого $dP = Q$. Подходы, которые применяются к анализу во многом зависят от параметров самой эллиптической кривой. Наиболее эффективным методом анализа на сегодняшний день является ро-метод Полларда, время выполнения которого оценивается как $\frac{1}{r}1/\mathrm{r}\sqrt{\frac{\pi n}{2}}$, где n – порядок точки эллиптической кривой, а r – число параллельных процессоров для проведения вычислений [22]. Задача дискретного логарифмирования для эллиптических кривых с рекомендованными параметрами (табл. 1) является на сегодняшний день сложно вычислимой и не применимой на практике. Тем не менее, рассмотрим несколько ситуаций, когда ошибки в реализациях

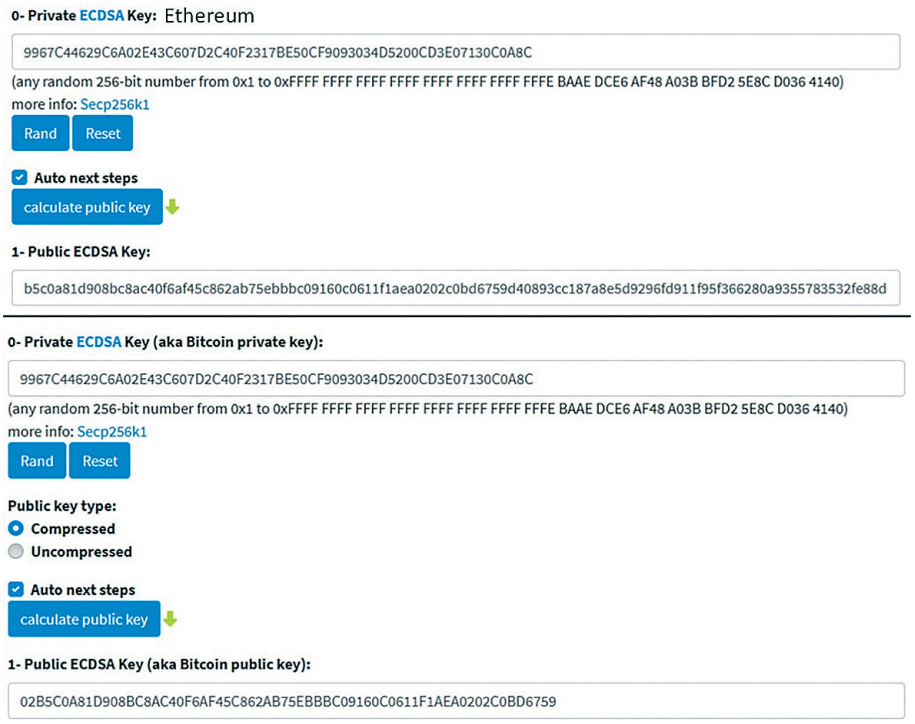

Рис. 3. Пример генерации одинаковых адресов для платформ Биткоин и Эфириум

или неправильное использование параметров системы может привести к уязвимостям.

Кейс № 2.

Описание задачи: Определить приватный ключ пользователя, если известен его публичный ключ, а в реализации алгоритма выработки ключей используется слабая ПСП.

Постановка задачи: генерация пары ключей выполняется в соответствии с алгоритмом цифровой подписи ECDSA с заданным кортежем параметров $T = (p, a, b, G, n, h)$. Известен публичный ключ пользователя $A\, Q_A = (x, y)$. Известно, что реализация генератора ПСП $k = Gen()$ генерирует параметры k, для которых $p\ (|k_i - k_j| \le d) \ge 0{,}5$. Требуется определить приватный ключ пользователя A.

Решение:

Вход: генератор $k = Gen()$; расстояние d; публичный ключ пользователя A

$$Q_A = (x, y).$$

Выход: приватный ключ пользователя $A\, d_A$.

В данном случае можно предположить, что если сформировать новый приватный ключ, то он с вероятностью 0,5 будет отстоять от ключа пользователя A не более чем на d точек. Сложность работы алгоритма будет напрямую зависеть от величины d.

Алгоритм 2:

1. Сгенерировать $k = Gen()$.
2. Вычислить точку $Q1 = k^*G$;
3. Если $Q1 = Q_A$, то $d_A = k$; прервать работу алгоритма.
4. Для всех i от 1 до d:
5. $Q1 = Q1 + G$;
6. Если $Q1 = Q_A$, то $d_A = k + i$; прервать работу алгоритма.
7. Если ключ не найден, то вернуться к шагу 1.

Результаты:

Все эксперименты для каждого набора параметром проводились по 100 раз, после чего бралось усредненное значение затраченного времени. Здесь и далее для других экспериментов программный код написан на языке Python 3.9, испытания проводились на ПК с процессором AMD Ryzen 5 3500U with Rdeon Vega Mobile Gfx 2,10 GHz. Эксперимент выполнялся для эллиптической кривой стандарта secp256k1, параметры для кортежа которой определены в табл. 1. Результаты эксперимента представлены в табл.2.

Кейс № 3.

Описание задачи: определить приватный ключ пользователя, если известно, что в реализации алгоритма подписи ECDSA допущена ошибка и используется постоянное значение сеансового ключа k_s.

Постановка задачи: Генерация пары ключей выполняется в соответствии с алгоритмом цифровой подписи ECDSA с заданным кортежем параметров $T = (p, a, b, G, n, h)$. Известно, что значение k_s является

Таблица 2.

Количественные характеристики результатов определения приватного ключа при слабой ПСП для рекомендованных параметров кривой secp256k1

№ п/п	d	Время подбора, сек	Кол-во попыток	d_A в 16-ричной форме	Q_A в сжатой форме
1	5	0,0069	9	9b27f4269de7d343 103ff00ec251ceff 7c09109df7ef9cc6 ab7203f2b8bb382b	0345ded15ee5f5e5da 33f29c8e9970c9f4 4b036581d66b996f 4d56cf3378882667
2	50	0,0423	54	0f0593537d2c49e6 896e197f9f6cfe29 5e4373df9aaf3021 a55ec26ad04c4e4b	032256596f03a2b3c0 a331e147db42e8e6 7fdc5a1f036e1f8d 89f8df0ff8f822da
3	1000	0,7556	1076	5a9fc8eb89957a96 beea4cfb27856a3d 58a5e7ad94a3f4ea 4dbd3194e7f53b43	038309b865ded35114 23860f20fda7a365 d2e5e6cf0a054d1c a379dee08d4d2a1
4	100000	83,3632	101204	c6cbcbf0736f8193 dbe6c09a09b31955 de385328e4094f40 67d10b6d98d03b7d	0341b90854b5aa5650 90ad60c5e469e18a 803f06d8ad84a86f 6792437116f46cad

одинаковым для всех сообщений M. Известны два сообщения $M1$ и $M2$ и соответствующие им значения подписей (r_1, s_1) и (r_2, s_2) соответственно. Требуется определить приватный ключ пользователя A.

Решение:

Вход: Сообщения $M1$ и $M2$ и их подписи (r_1, s_1) и (r_2, s_2) соответственно.

Выход: приватный ключ пользователя A d_A.

Так как значение сессионного ключа не меняется, то координата r для точек $M1$ и $M2$ будет вычислена одинаково: $r = r_1 = r_2 = k_sG$.

Для сообщения $M1$ параметр s_1 вычисляется как $s_1 = k_s^{-1}(h(M1) + rd_A)$.

Для сообщения $M2$ параметр s_2 вычисляется как $s_2 = k_s^{-1}(h(M2) + rd_A)$.

Умножив оба уравнения на k_s, получим:

$$s_1k_s = h(M1) + rd_A;$$
$$s_2k_s = h(M2) + rd_A.$$

Тогда:

$$s_1k_s - s_2k_s = h(M1) + rd_A - h(M2) + rd_A;$$
$$k_s(s_1 - s_2) = h(M1) - h(M2);$$
$$k_s = (h(M1) - h(M2))(s_1 - s_2)^{-1}.$$

Так как значения s_1, s_2, $M1$, $M2$ известны, то мы легко можем определить параметр k_s. Найдя его, из формулы $s_1k_s = h(M1) + rd_A$ определим значение секретного ключа d_A:

$$d_A = (s_1k_s - h(M1))r^{-1}.$$

Результаты.

Для данного кейса было проведено два эксперимента. В первом эксперименте использовалось уравнение для кривой secp256k1, однако все параметры были взяты в маленьких размерностях, для проверки работоспособности предложенного кейса (табл. 3). Для второго эксперимента был взят кортеж параметров из табл. 1, соответствующий кривой secp256k1 (табл. 4). Генерация постоянного сеансового ключа реализована его единоразовой выработкой для последующего использования при подписании всех сообщений. В качестве сообщений взяты два текста $M1$ = «Message1», $M2$ = «Message2». Строковые значения сообщений $M1$ и $M2$ преобразуются в последовательность байтов соответствующих символов в кодировке UTF-8, после чего вычисляется хэш с помощью алгоритма SHA-256:

UTF-8 «Message1» = ‘0x4d65737361676531’.

$h(M1)$ = 960c9384c0db44a860f1309fa04b2d 2ec34370ee0148838b659f65d75bf1c85c.

UTF-8 «Message2» = ‘0x4d65737361676532’.

$h(M2)$ = 6b5bcdd8bc2e343fe28e6061e5f565 9f85f83cc84350488b71b28298bb0c7b15.

В случае первого эксперимента для небольших значений модуля p в качестве хеш-значения использовался результат $h(M) \bmod p$. На примере кортежа T_3 = (163, 0, 7, (105;150), 138, 1) и сообщений $M1$ = «Message1», $M2$ = «Message2» разберём процесс восстановления приватного ключа.

Случайно сгенерированный сеансовый ключ: $k_s = 105$.

Публичный ключ $Q_A = (70, 113)$.

Координата r для сообщений $M1$ и $M2$: r1 = r2 = $= k_sG = 6$.

Таблица 3.

Параметры и результаты определения приватного ключа при постоянном значении сеансового ключа k_s

№ п/п	$T = (p,a,b,G,n,h)$	Среднее время работы программы
1	T_1 = (7, 0, 7, (2;6), 6, 1)	0,0001185
2	T_2 = (61, 0, 7, (30;44), 60, 1)	0,0001597
3	T_3 = (163, 0, 7, (105;150), 138, 1)	0,0001798
4	T_4 = (349, 0, 7, (97;163), 312, 1)	0,0002110
5	T_5 = (433, 0, 7, (236;293), 396, 1)	0,0002934

Таблица 4.

Результаты моделирования вычисления приватного ключа при постоянном значении сеансового ключа k_s

Сеансовый ключ ks	24cc15aa81a8affdf09ac591c2f8bb2fed22c79b16bf4cdfd6e242ee1f8ddbf1
Параметр r_1 подписи сообщения $M1$	c6a25157b952d0ab79dca9694bda8431eb68bc74509ce4efbf70f21441c0da6
Параметр s_1 подписи сообщения $M1$	e90a808b407fa02eb380f63b98e66827332d591006907c73baae722148c39ad8
Параметр r_2 подписи сообщения $M2$	c6a25157b952d0ab79dca9694bda8431eb68bc74509ce4efbf70f21441c0da6
Параметр s_2 подписи сообщения $M2$	7d5923769e75baf2fe1cc377275bb87b30473cbea534749ca6d04d2dc2a14f59
Восстановление секретного ключа d_A	ee87f2154fe7b2bc24f3befec93676d100db075354e5f92e4aec698ca779e112
Среднее время работы программы	0,1915586

Для сообщения $M1$ параметр

$$s_1 = k_s^{-1}(h(M1) + rd_A) = 134.$$

Для сообщения $M2$ параметр

$$s_2 = k_s^{-1}(h(M2) + rd_A) = 111.$$

Вычисление приватного ключа

$$d_A = (s_1 k_s - h(M1))r^{-1} = 127.$$

Кейс № 4.

Описание задачи. Подменить подпись сообщения $M1$ на подпись сообщения $M2$ в алгоритме ECDSA, если известно, что в реализации алгоритма подписи допущена ошибка и генерация сеансового ключа k_s генерирует параметры $k = Gen()$, для которых $p(|k_i - k_j| \le d) \ge 0{,}5$.

Постановка задачи. Генерация пары ключей выполняется в соответствии с алгоритмом цифровой подписи ECDSA с заданным кортежем параметров $T = (p, a, b, G, n, h)$. Известно, что $p(|k_i - k_j| \le d) \ge 0{,}5$. Известно сообщение $M1$ и его подпись (r_1, s_1). Требуется для сообщения $M2$ сформировать подпись (r_2, s_2) от лица пользователя A без знания его секретного ключа.

Решение:

Вход: Генератор $k = Gen()$; расстояние d; сообщение $M1$ и его подпись (r_1, s_1).

Выход: Подпись (r_2, s_2) для сообщения $M2$.

Первая часть работы алгоритма схожа с Алгоритмом 2 для Кейса № 2. Ожидается, что для новой подписи параметр k_s с вероятностью 0,5 будет отстоять от того, который был использован для подписи сообщения $M1$ не более чем на d точек. Сложность работы алгоритма будет напрямую зависеть от величины d. Тогда с помощью Алгоритма 3 можно определить параметр k_s известной подписи.

Алгоритм 3:

1. Сгенерировать $k = Gen()$.
2. Вычислить точку $Q1 = k^*G = (x_2, y_2)$;
3. Если $r_1 = x_2$, то $k_s = k$; прервать работу алгоритма.
4. Для всех i от 1 до d:
5. $Q1 = Q1 + G = (x_{i+2}, y_{i+2})$;
6. Если $r_1 = x_{i+2}$, то $k_s = k + i$; прервать работу алгоритма.
7. Если k_s не найден, то вернуться к шагу 1.

В случае успешного срабатывания Алгоритма 2, найденный параметр k_s для подписи сообщения $M1$

будет использован для формирования подписи сообщения $M2$. Для сообщения $M1$ подпись сформирована следующим образом:

$$r_1 = k_s G$$
$$s_1 = k_s^{-1}(h(M1) + rd_A)$$

Задача заключается в том, чтобы сформировать подпись для сообщения $M2$ вида:

$$r_2 = r_1 = k_s G$$
$$s_2 = k_s^{-1}(h(M2) + rd_A)$$

Таким образом получается, что надо вычислить только значение s_2. Преобразуем уравнения для s_1 и s_2 следующим образом:

$$s_1 k_s = h(M1) + rd_A;$$
$$s_2 k_s = h(M2) + rd_A.$$

Тогда:

$$s_1 k_s - s_2 k_s = h(M1) + rd_A - h(M2) - rd_A;$$
$$k_s(s_1 - s_2) = h(M1) - h(M2);$$
$$s_1 - s_2 = (h(M1) - h(M2))k_s^{-1};$$
$$s_2 = s_1 - (h(M1) - h(M2))k_s^{-1}.$$

Так как значения s_1, $M1$, $M2$ и k_s известны, то легко можно определить параметр s_2. Подпись (r_2, s_2) сформирована без применения приватного ключа пользователя A и будет корректно проверена при использовании публичного ключа пользователя A.

Результаты.

Для реализации поставленной задачи были взяты четыре различных значения расстояния d = 5, 50, 1000, 100000, входные сообщения $M1$, $M2$, и кортеж параметров эллиптической кривой стандарта

Таблица 5.

Параметры и результаты подделки подписи сообщения при ошибке в алгоритме генерации сеансового ключа k_s

d	Время работы программы, сек	Кол-во попыток перебора k_s	d_A	Q_A	k_s
5	0,0070	8	7bffa3bf1e22cee4 f70c62f7c79e671d c007a3754b29e4b2 32cd7ccb4cc77dfb	0331eb2c9de1cb9779 ed3995fcfa68c9a1 062c6a206f204de9 d35b74883e2dc135	ec1deee6be60e11a 5da9bb0c16f199a7 d1be103ad066a77a bb45be0c2a74bc66
50	0,0750	86	577cd96fe3b2f95b debdf8fe36741986 0be1a8dce235c489 621382e6513f9c73	036c734bd04a7a3d94 1f0bfe187b60d17e a88d32568d98c620 3d7d741d88c49312	9fff0bc70e3a4046 9d29e37ea5424b56 f6620946ecffb778 2722d4ce8497debb
1000	1,0680	1172	cc17968b9e781a78 564013feee370cf1 d0a8c02a8733bc4f 873af4352c1bdc21	03a333bd403738329d b03194e8081ae6fe 237c98e1473d7d42 4584860fec206a8a	46caaef576866ed8 2cdb82cae0a9da3d 1eb2e8f42f076537 e91ed0657f1fbd4e
100000	364,3640	249230	5aa8f20f98389035 722db3a5adf51d6e 6c13c8abc380a551 f050a1f751a55baa	03f31b53b04e172c95 7a981c0a70f22c8b a1a984dcf9c7b7c4 e1d0689fea4261a6	a23fceea41475299 c796b4f96d6887fc 988d47d12d4fd7a0 39538f5f7fa1ae6f

Таблица 6.

Параметры и результаты подделки подписи сообщения при ошибке в алгоритме генерации сеансового ключа k_s

d	r_1	s_1	r_2	s_2
5	6d9cc47faeff7678 d1c53da125b7f4d8 9286b646dbcbb1e6 4852689b54410495	9d641b1f78158495 0bd858fcb04fa939 da6c96e26dcb8044 d0a1633a53fba374	26d2f6ce9154631e 4a5be8abacb15c36 f40e73750a7ab02a 45d495f09960fa55	ae1ed24d3cf7cda2 9572a34242dfe4a8 61539b8f14c97e1d 57c9f4446dcb0e56
50	2f2f93d07da96f86 b1f8cdab9ccf710c 3e1904dc20851d30 8fc675c538f67031	bd9859c8394f1e6a 661804f555c3a3d0 37ada9747e1a0ee7 36ec90496b15b4ec	753ebb733d7e79aa d8ec74be34409da6 d836c33ddbedacc8 744feb12b605506f	1c3016afd047551f 3e5333d2fab22613 94e3f978657567f5 78bf06539f16ba38
1000	51917a1a80feb186 2c18c6516f17d426 bb3211f9f52cc3d5 762e110e9b08c753	9be668e11e48f942 bbf77cbdd2f8a2c7 baa451670515be82 52fa838cd1b8c22c	85e8a767ac51c383 ad29fe8b5cba7b0b c106d9b0133a3e8f 02a499dc35892efd	1d17e28fd8c48558 7425c66c50eea3ef b54465d4aadb2509 1bb07fcf1f3681ee
100000	726f60201ce9a52ec 5b4db8fb89d29b91 122ab7057034cd5 ca75c7b3409924df	8a660b2e4cc856d 2ed15529fd253a57 6071fe3b1086f148d b55a6005ad38423d	12adf440b5dcb6f6 49529b7088e9375e 0a5d5840261e1ad0 cbe722c46dd7d883	1cf760c24bce61f7 6c6f73e0eb94457d 313695e6a9645f7d eaab10173bcd2104

secp256k1 из табл. 1. Результаты экспериментов представлены в табл. 5 и 6.

Генерация сеансового ключа реализована его единоразовой выработкой для последующего использования при подписании всех сообщений, расстояние d, как величина возможного отклонения, в процессе подбора ключа задавалось различное для каждого эксперимента. С увеличением расстояния d уменьшается вероятность подбора корректного сеансового ключа, так как перебор будет увеличивать диапазон допустимых значений.

Выводы

В работе рассмотрены потенциально уязвимые места в использовании асимметричных алгоритмов шифрования в современных блокчейн-системах. Рассмотренные кейсы моделируют ситуации, которые могут возникнуть в результате неправильного использования стартовых параметров криптосистемы или при неверной реализации вспомогательных компонентов. Таких, как например, генераторы псевдослучайных последовательностей. Для каждого рассмотренного кейса приведено математическое решение сформулированной проблемы, определена вычислительная сложность и проведены эксперименты, в том числе с использованием параметров, рекомендованных для эллиптической кривой secp256k1.

Достоверность предлагаемого научного подхода подтверждается применением общенаучных методов исследования, достаточным информационным обеспечением, а также корректным применением методов криптографии, в том числе в построении формульных доказательств и выводов и экспериментальным подтверждением работоспособности выведенных формул.

Результаты получены при финансовой поддержке проекта «Технологии противодействия ранее неизвестным квантовым киберугрозам», реализуемого в рамках государственной программы федеральной территории «Сириус» «Научно-технологическое развитие федеральной территории «Сириус» (Соглашение №23-03 от 27.09.2024 г.).

Литература

1. Kakarlapudi P. V., Mahmoud Q. H. A Systematic Review of Blockchain for Consent Management. Healthcare. 2021; 9(2):137. https://doi.org/10.3390/healthcare9020137.
2. Ищукова Е. А., Панасенко С. П., Романенко К. С., Салманов В. Д. Криптографические основы блокчейн-технологий. – М.: ДМК Пресс, 2022. – 302 с. ISBN: 978-5-97060-865-4.
3. Zhang H., Jiang W., Ding J. A Blockchain Network Admission Control Mechanism Using Anonymous Identity-Based Cryptography. Applied Sciences. 2025; 15(1):130. https://doi.org/10.3390/app15010130.
4. Chin Y.-C., Hsu C.-L., Lin T.-W., Tsai K.-Y. A Hierarchical Blockchain System for Social Economy Services. Electronics. 2024; 13(20):4004. https://doi.org/10.3390/electronics13204004.
5. Joni S. A., Rahat R., Tasnin N., Ghose P., Uddin M. A., Ayoade J. Hybrid-Blockchain-Based Electronic Voting Machine System Embedded with Deepface, Sharding, and Post-Quantum Techniques. Blockchains. 2024; 2(4):366–423. https://doi.org/10.3390/blockchains2040017.
6. Kim H., Kim W., Kang Y., Kim H., Seo H. Post-Quantum Delegated Proof of Luck for Blockchain Consensus Algorithm. Applied Sciences. 2024; 14(18):8394. https://doi.org/10.3390/app14188394.
7. Gu H., Shang J., Wang P., Mi J., Bhattacharjya A. A Secure Protocol Authentication Method Based on the Strand Space Model for Blockchain-Based Industrial Internet of Things. Symmetry. 2024; 16(7):851. https://doi.org/10.3390/sym16070851.
8. Thantharate P., Thantharate A. ZeroTrustBlock: Enhancing Security, Privacy, and Interoperability of Sensitive Data through ZeroTrust Permissioned Blockchain. Big Data and Cognitive Computing. 2023; 7(4):165. https://doi.org/10.3390/bdcc7040165.
9. Thanalakshmi P., Rishikhesh A., Marion Marceline J, Joshi GP, Cho W. A Quantum-Resistant Blockchain System: A Comparative Analysis. Mathematics. 2023; 11(18):3947. https://doi.org/10.3390/math11183947.
10. Wenhua Z., Qamar F., Abdali T.-AN., Hassan R., Jafri S. T. A., Nguyen Q. N. Blockchain Technology: Security Issues, Healthcare Applications, Challenges and Future Trends. Electronics. 2023; 12(3):546. https://doi.org/10.3390/electronics12030546.
11. Di Scala A. J., Gangemi A., Romeo G., Vernetti G. Special Subsets of Addresses for Blockchains Using the secp256k1 Curve. Mathematics. 2022; 10(15):2746. https://doi.org/10.3390/math10152746.
12. Longo R., Mascia C., Meneghetti A., Santilli G., Tognolini G. Adaptable Cryptographic Primitives in Blockchains via Smart Contracts. Cryptography. 2022; 6(3):32. https://doi.org/10.3390/cryptography6030032.
13. Bellés-Muñoz M., Whitehat B., Baylina J., Daza V., Muñoz-Tapia J. L. Twisted Edwards Elliptic Curves for Zero-Knowledge Circuits. Mathematics. 2021; 9(23):3022. https://doi.org/10.3390/math9233022.
14. Martínez V. G., Hernández-Álvarez L., Encinas L. H. Analysis of the Cryptographic Tools for Blockchain and Bitcoin. Mathematics. 2020; 8(1):131. https://doi.org/10.3390/math8010131.
15. Sala M., Sogiorno D., Taufer D. A Small Subgroup Attack on Bitcoin Address Generation. Mathematics. 2020; 8(10):1645. https://doi.org/10.3390/math8101645.
16. Марков А. С. Кибербезопасность и информационная безопасность как бифуркация номенклатуры научных специальностей // Вопросы кибербезопасности. 2022, № 1(47), с. 2–9.

17. Petrenko A. S., Petrenko S. A. Basic Algorithms Quantum Cryptanalysis. The journal «Cybersecurity Issues», 2023, no. 1(53), pp. 100–115. doi: 10.21681/2311-3456-2023-1-100-115.
18. Petrenko A. S. Applied Quantum Cryptanalysis (scientific monograph). River Publishers, 2023, 256 pp. ISBN 9788770227933. doi: 10.1201/9781003392873.
19. Марков А. С. Важная веха в безопасности открытого программного обеспечения // Вопросы кибербезопасности, 2023, № 1(53), С.2–12.
20. Pandey D. K., Nicolosi A. R. Pseudorandom Function from Learning Burnside Problem. Mathematics. 2025; 13(7):1193. https://doi.org/10.3390/math13071193.
21. Ishchukova, E., Borlakov, R. Reliability of Information Conversion When Encrypting Graphic Images. In: Raza, Z., Babenko, M., Sajid, M., Lapina, M., Zolotarev, V. (eds) AISMA-2023: International Workshop on Advanced Information Security Management and Applications. AISMA 2023. Lecture Notes in Networks and Systems, vol. 1207. Springer, Cham, 2024. https://doi.org/10.1007/978-3-031-77229-0_10.
22. Jebrane J., Chhaybi A., Lazaar S., Nitaj A. Elliptic Curve Cryptography with Machine Learning. Cryptography. 2025; 9(1):3. https://doi.org/10.3390/cryptography9010003.
23. Martinez-Diaz I., Ali R., Jamil M. K. On the Search for Supersingular Elliptic Curves and Their Applications. Mathematics. 2025; 13(2):188. https://doi.org/10.3390/math13020188.
24. Aljaedi A., Rashid M., Jamal S. S., Alharbi A. R., Alotaibi M. An Optimized Flexible Accelerator for Elliptic Curve Point Multiplication over NIST Binary Fields. Applied Sciences. 2023; 13(19):10882. https://doi.org/10.3390/app131910882.
25. Lone P. N., Singh D., Stoffová V., Mishra D. C., Mir U. H., Kumar N. Cryptanalysis and Improved Image Encryption Scheme Using Elliptic Curve and Affine Hill Cipher. Mathematics. 2022; 10(20):3878. https://doi.org/10.3390/math10203878.
26. Sattar B., Sadkhan A. Proposed Developments of Pollards Rho Method for Attacking the ECDLP // 2021 7th International Engineering Conference «Research & Innovation amid Global Pandemic» (IEC). DOI: 10.1109/IEC52205.2021.9476119.

VULNERABILITIES OF ASYMMETRIC CIPHERS OF BLOCKCHAIN PLATFORMS

Ishchukova E. A.[4], Petrenko S. A.[5], Leonteva I. P.[6]

Keywords: *mandatory access control principle, «write down», «write down» event flow, degradation of the access control system, «post-maximum attenuation» effect.*

Purpose: *the aim of this work is to identify potential vulnerabilities in the implementations of asymmetric ciphers based on the elliptic curves, applied in modern blockchain systems.*

Method: *the research methods are based on the use of information theory, the theory of cryptography and cryptanalysis, the mathematical apparatus of probability theory and mathematical statistics, the theory of constructing blockchain systems, and the theory of information security.*

Results: *the paper considers the main techniques used to construct asymmetric ciphers based on the use of elliptic curves. The algorithm for adding two points, multiplying a point by a scalar, and determining the second coordinate of a point is considered. The properties of pseudorandom sequence generators are briefly considered separately: the principles of their construction and their impact on the stability of asymmetric ciphers in which they are used. The ECDSA digital signature algorithm, which is used in blockchain platforms such as Bitcion, Litecoin, Ethereum, and many others, is considered as a basic algorithm.*

The scientific novelty *lies in the consideration of a number of cases simulating the emergence of vulnerabilities in asymmetric cryptography used in modern blockchain systems. For each case, a description of the problem is made, a statement of the task is formulated, a possible solution is given and an assessment of its complexity is given. It is shown that with the correct use of the mathematical apparatus of ciphers, compliance with the requirements for the selection of starting parameters, the absence of errors in software implementations, sufficient stability is ensured.*

References

1. Kakarlapudi P.V., Mahmoud Q.H. A Systematic Review of Blockchain for Consent Management. Healthcare. 2021; 9(2):137. https://doi.org/10.3390/healthcare9020137.
2. Ishhukova E. A., Panasenko S. P., Romanenko K. S., Salmanov V. D. Kriptograficheskie osnovy blokchejn-tehnologij. – M.: DMK Press, 2022. – 302 s. ISBN: 978-5-97060-865-4.

4 Evgeniya A. Ishchukova, Ph.D. (in Tech.), Leading researcher, Scientific Center for Information Technologies and Artificial Intelligence, Sirius University of Science and Technology, Address: Olimpiyskiy ave. b.1, Sirius, Sirius Federal Territory, Krasnodar region, Russia, 354340. ORCID 0000-0002-6818-1608. E-mail: ischukova.ea@talantiuspeh.ru
5 Sergei A. Petrenko, Dr.Sc. (of Tech.), Professor), Team Leader, Scientific Center for Information Technologies and Artificial Intelligence, Sirius University of Science and Technology, Address: Olimpiyskiy ave. b.1, Sirius, Sirius Federal Territory, Krasnodar region, Russia, 354340. ORCID 0000-0003-0644-1731. E-mail: Petrenko.SA@talantiuspeh.ru
6 Yuliya P. Leontieva, Student, Institute of Computer Technologies and Information Security, Southern Federal University. ORCID 0009-0006-4778-131X, E-mail: izavodnova@sfedu.ru

3. Zhang H., Jiang W., Ding J. A Blockchain Network Admission Control Mechanism Using Anonymous Identity-Based Cryptography. Applied Sciences. 2025; 15(1):130. https://doi.org/10.3390/app15010130.
4. Chin Y.-C., Hsu C.-L., Lin T.-W., Tsai K.-Y. A Hierarchical Blockchain System for Social Economy Services. Electronics. 2024; 13(20):4004. https://doi.org/10.3390/electronics13204004.
5. Joni S. A., Rahat R., Tasnin N., Ghose P., Uddin M. A., Ayoade J. Hybrid-Blockchain-Based Electronic Voting Machine System Embedded with Deepface, Sharding, and Post-Quantum Techniques. Blockchains. 2024; 2(4):366-423. https://doi.org/10.3390/blockchains2040017.
6. Kim H., Kim W., Kang Y., Kim H., Seo H. Post-Quantum Delegated Proof of Luck for Blockchain Consensus Algorithm. Applied Sciences. 2024; 14(18):8394. https://doi.org/10.3390/app14188394.
7. Gu H., Shang J., Wang P., Mi J., Bhattacharjya A. A Secure Protocol Authentication Method Based on the Strand Space Model for Blockchain-Based Industrial Internet of Things. Symmetry. 2024; 16(7):851. https://doi.org/10.3390/sym16070851.
8. Thantharate P., Thantharate A. ZeroTrustBlock: Enhancing Security, Privacy, and Interoperability of Sensitive Data through ZeroTrust Permissioned Blockchain. Big Data and Cognitive Computing. 2023; 7(4):165. https://doi.org/10.3390/bdcc7040165.
9. Thanalakshmi P., Rishikhesh A., Marion Marceline J, Joshi GP, Cho W. A Quantum-Resistant Blockchain System: A Comparative Analysis. Mathematics. 2023; 11(18):3947. https://doi.org/10.3390/math11183947.
10. Wenhua Z., Qamar F., Abdali T.-AN., Hassan R., Jafri S. T. A., Nguyen Q. N. Blockchain Technology: Security Issues, Healthcare Applications, Challenges and Future Trends. Electronics. 2023; 12(3):546. https://doi.org/10.3390/electronics12030546.
11. Di Scala A. J., Gangemi A., Romeo G., Vernetti G. Special Subsets of Addresses for Blockchains Using the secp256k1 Curve. Mathematics. 2022; 10(15):2746. https://doi.org/10.3390/math10152746.
12. Longo R., Mascia C., Meneghetti A., Santilli G., Tognolini G. Adaptable Cryptographic Primitives in Blockchains via Smart Contracts. Cryptography. 2022; 6(3):32. https://doi.org/10.3390/cryptography6030032.
13. Bellés-Muñoz M., Whitehat B., Baylina J., Daza V., Muñoz-Tapia J. L. Twisted Edwards Elliptic Curves for Zero-Knowledge Circuits. Mathematics. 2021; 9(23):3022. https://doi.org/10.3390/math9233022.
14. Martínez V. G., Hernández-Álvarez L., Encinas L. H. Analysis of the Cryptographic Tools for Blockchain and Bitcoin. Mathematics. 2020; 8(1):131. https://doi.org/10.3390/math8010131.
15. Sala M., Sogiorno D., Taufer D. A Small Subgroup Attack on Bitcoin Address Generation. Mathematics. 2020; 8(10):1645. https://doi.org/10.3390/math8101645.
16. Markov A. S. Kiberbezopasnost' i informacionnaja bezopasnost' kak bifurkacija nomenklatury nauchnyh special'nostej // Voprosy kiberbezopasnosti. 2022, № 1(47), c. 2–9.
17. Petrenko A. S., Petrenko S. A. Basic Algorithms Quantum Cryptanalysis. The journal «Cybersecurity Issues», 2023, no. 1(53), pp. 100–115. doi: 10.21681/2311-3456-2023-1-100-115.
18. Petrenko A. S. Applied Quantum Cryptanalysis (scientific monograph). River Publishers, 2023, 256 pp. ISBN 9788770227933. doi: 10.1201/9781003392873.
19. Markov A. S. Vazhnaja veha v bezopasnosti otkrytogo programmnogo obespechenija // Voprosy kiberbezopasnosti, 2023, № 1(53),. S. 2–12.
20. Pandey D. K., Nicolosi A. R. Pseudorandom Function from Learning Burnside Problem. Mathematics. 2025; 13(7):1193. https://doi.org/10.3390/math13071193.
21. Ishchukova, E., Borlakov, R. Reliability of Information Conversion When Encrypting Graphic Images. In: Raza, Z., Babenko, M., Sajid, M., Lapina, M., Zolotarev, V. (eds) AISMA-2023: International Workshop on Advanced Information Security Management and Applications. AISMA 2023. Lecture Notes in Networks and Systems, vol. 1207. Springer, Cham, 2024. https://doi.org/10.1007/978-3-031-77229-0_10.
22. Jebrane J., Chhaybi A., Lazaar S., Nitaj A. Elliptic Curve Cryptography with Machine Learning. Cryptography. 2025; 9(1):3. https://doi.org/10.3390/cryptography9010003.
23. Martinez-Diaz I, Ali R, Jamil M. K. On the Search for Supersingular Elliptic Curves and Their Applications. Mathematics. 2025; 13(2):188. https://doi.org/10.3390/math13020188.
24. Aljaedi A., Rashid M., Jamal S. S., Alharbi A. R., Alotaibi M. An Optimized Flexible Accelerator for Elliptic Curve Point Multiplication over NIST Binary Fields. Applied Sciences. 2023; 13(19):10882. https://doi.org/10.3390/app131910882.
25. Lone P. N., Singh D., Stoffová V., Mishra D. C., Mir U. H., Kumar N. Cryptanalysis and Improved Image Encryption Scheme Using Elliptic Curve and Affine Hill Cipher. Mathematics. 2022; 10(20):3878. https://doi.org/10.3390/math10203878.
26. Sattar B. Sadkhan A Proposed Developments of Pollards Rho Method for Attacking the ECDLP // 2021 7th International Engineering Conference «Research & Innovation amid Global Pandemic» (IEC). DOI: 10.1109/IEC52205.2021.9476119.