Оптимизированный метод цифровой подписи на эллиптических кривых для устройств с ограниченными ресурсами

Наурас Х. Сабри[1]

[1]Санкт-Петербругский государственный электротехнический университет «ЛЭТИ», Россия, 197022, г. Санкт-Петербург

Аннотация. Настоящее исследование посвящено решению проблемы обеспечения безопасности в средах с ограниченными ресурсами, таких как микроконтроллеры, используемых в беспилотных транспортных системах (UVS), связанной с вычислительными затратами криптографических алгоритмов. Для решения данной проблемы был предложен оптимизированный метод цифровой подписи на основе эллиптической кривой (ECC), являющийся пригодным для обозначенных условий. Метод содержит два ключевых улучшения: оптимизацию скалярного умножения точки с использованием свойства циклической группы точек эллиптической кривой и свойства противоположного числа в теории групп, а также использование детерминированного подхода к генерации закрытого однократно используемого числа, при этом исключая публичное однократно используемое число (R) из расчёта задачи для повышения эффективности подписания.

Экспериментальная реализация метода была выполнена на микроконтроллере Atmega2560 с использованием низкоуровневого программирования. Результаты демонстрируют значительные улучшения как вычислительной эффективности, так и использования памяти на всех этапах работы с цифровой подписью – генерации ключей, подписания и верификации, при сохранении необходимого уровня безопасности. Данная работа подчёркивает практическую возможность реализации эффективных и безопасных алгоритмов цифровой подписи на базе ECC даже в условиях ограниченных ресурсов.

Ключевые слова: цифровая подпись, ECC, микроконтроллер, устройства с ограниченными ресурсами, ATmega2560.

Abstract. This study focuses on solving the problem of ensuring security in resource-constrained environments, such as microcontrollers used in unmanned vehicle systems (UVS), related to the computational costs of cryptographic algorithms. To address this problem, an optimized elliptic curve cryptography (ECC) electronic signature method suitable for the specified conditions was proposed. The method contains two key improvements: optimization of scalar multiplication of points using the cyclic group property of elliptic curve points and the opposite number property in group theory, as well as the use of a deterministic approach to generating a private one-time number, while excluding the public one-time number (R) from the task calculation to improve signing efficiency.

The experimental implementation of the method was performed on an Atmega2560 microcontroller using low-level programming. The results demonstrate significant improvements in both computational efficiency and memory usage at all stages of working with digital signatures – key generation, signing, and verification – while maintaining the required level of security. This

work highlights the practical possibility of implementing efficient and secure ECC-based digital signature algorithms even under resource-constrained conditions.

Keywords: digital signature, ECC, microcontroller, resource-constrained devices, ATmega2560.

Введение

В условиях современной цифровой эпохи взаимосвязанных устройств обеспечение надёжной безопасности при сохранении вычислительной эффективности становится критически важной задачей. Особую сложность представляет реализация криптографических алгоритмов в ресурсоограниченных средах, таких как микроконтроллеры, где ограничения по вычислительной мощности, объёму памяти и энергопотреблению делают выполнение сложных криптографических операций неэффективным и непрактичным.

Криптография на эллиптических кривых (ECC) зарекомендовала себя как перспективное решение для подобных условий, предлагая высокий уровень безопасности при относительно малых размерах ключей. Данный подход особенно актуален для беспилотных транспортных систем (UVS) и других IoT-приложений. ECC, как форма криптографии с открытым ключом, основана на математических свойствах эллиптических кривых над конечными полями для обеспечения безопасного и эффективного шифрования, электронных подписей и механизмов обмена ключами. Эллиптическая кривая (далее ЭК) определяется уравнением вида ($y^2 = x^3 + ax + b$), где a и b – коэффициенты, удовлетворяющие определённым условиям, обеспечивающим отсутствие сингулярностей на кривой. Множество решений приведённого уравнения вместе с обозначенной точкой на бесконечности образует абелеву группу с определённой операцией сложения.

Фундаментальным свойством данной групповой структуры является её циклический характер, что означает, что базовая точка G может быть использована для генерации всех других точек в конечной подгруппе посредством повторного сложения. В частности, для подгруппы ЭК простого порядка каждая точка P в группе может выражаться как кратное точки G, то есть ($P = k \cdot G$) для некоторого целого k, где операции выполняются по модулю порядка группы. Этот циклический характер, лежащий в основе безопасности схем обмена ключами и электронных подписей на основе ECC, имеет критическое значение для криптографических приложений. Кроме того, ECC обладает свойством, где для каждой точки P(x, y) существует противоположная точка $-P(x, -y)$, причём их сумма даёт точку на бесконечности $\mathcal{O}$ $(P + (-P) = \mathcal{O})$.

Безопасность ECC основывается на вычислительной сложности задачи дискретного логарифмирования (Elliptic Curv□Discrete Logarithm Problem, ECDLP), что делает определение скаляра k из уравнения ($k \cdot P = Q$) вычислительно невозможным. Данное свойство позволяет ECC обеспечивать уровень безопасности, сопоставимый с RSA, но при значительно меньших размерах ключей – 256-битный ключ ECC эквивалентен по стойкости 3072-битному ключу RSA.

Несмотря на преимущества, алгоритмы цифровой подписи (далее ЦП) на основе ECC (ECDSA [1] и EdDSA [2]) сталкиваются с проблемой высокой вычислительной сложности операций, особенно скалярного умножения точек, что ограничивает их применение в ресурсоограниченных устройствах. В данном исследовании предложен двухэтапный метод оптимизации, специально адаптированный для таких условий.

Первый этап оптимизации улучшает базовую операцию ECC – скалярное умножение точек, интегрируя свойство циклической группы и противоположного числа в оконный метод умножения точек ЭК на кривой Edwards25519, расширяя ранее опубликованное исследование [3]. Использование свойства противоположного числа снижает вычислительную сложность операции удвоения точки в рамках расширенной скрученной системы координат Эдвардса ($\mathcal{E}^e$) [4].

Второй этап продолжает исследование [5] и направлен на повышение эффективности алгоритмов цифровой подписи ECC за счёт применения детерминированного подхода к генерации закрытых ключей, аналогичного используемому в алгоритме EdDSA. Ключевым нововведением является метод формирования закрытого одноразового числа (nonce) путем комбинирования хеш-значения закрытого ключа (x) с подписываемым сообщением. Такой подход исключает необходимость использования публичного одноразового числа (R) в вычислительных операциях, что существенно повышает эффективность процесса подписания.

Экспериментальная реализация предложенного метода была выполнена на микроконтроллере Arduino Atmega 2560 R3, выбранном в качестве репрезентативной платформы для ресурсоограниченных устройств. Использование низкоуровневого программирования на языках C и Ассемблера позволило достичь значительного улучшения ключевых показателей производительности: сокращение количества тактов на 40% при подписании, 54% при генерации ключей и 29,6% при верификации подписей. Дополнительным преимуществом стало снижение потребления памяти – до 73% для оперативной памяти (SRAM) и 50% для flash-памяти по сравнению с существующими реализациями.

Основное отличие данного исследования от предыдущих работ заключается в комплексном подходе, сочетающем оптимизацию наиболее ресурсоемкой операции ECC – умножения точек эллиптической кривой за счёт использования свойств циклической группы и противоположных чисел, и детерминированную генерацию закрытых одноразовых чисел с исключением публичного nonce (R) из вычислительных процедур.

Полученные результаты демонстрируют не только возможность эффективной реализации криптографических алгоритмов на платформах с ограниченными ресурсами, но и создают основу для разработки безопасных IoT-решений нового поколения. Предложенная методика особенно актуальна для современных встраиваемых систем, где требования к энергоэффективности и быстродействию сочетаются с необходимостью обеспечения надёжной криптографической защиты.

1. Анализ методов умножения точек ЭК и алгоритмов ЦП

В области умножения точек эллиптической кривой (ECPM) значительный вклад внесли исследования Уллаха и Захилы (2021) [6], которые реализовали ECPM на микроконтроллере AVR ATmega2560 с использованием кривой Curve25519. Их работа сосредоточилась на оптимизации операций сложения и удвоения точек, что позволило достичь высокой производительности: скалярное умножение выполнялось за $2{,}62 - 2{,}9$ секунды ($41{,}92 - 46{,}4$ миллиона тактов при частоте 16 МГц). При этом использовалось всего 7,8% flash-памяти и 17% оперативной памяти, что демонстрирует эффективный баланс между вычислительной нагрузкой и ресурсами.

Дальнейшее развитие в данной области представлено в исследовании Фотиадиса и др. (2024) [7], где рассматривались ECPM для трёх уровней безопасности (159-бит, 207-бит и 255-бит) на 16-битном микроконтроллере MSP430F1611. Кривая Curve2065 (207 бит) показала себя как оптимальное решение, обеспечивая производительность в 1,69 раза выше, чем 255-битные реализации, и оставаясь более безопасной по сравнению с 159-битными вариантами. Например, для Curve2065 скалярное умножение с переменной базой требовало 7 216 173 тактов, а с фиксированной – 2 961 135 тактов.

В контексте цифровой подписи на ЭК выделяется работа Наумы и Явуза (2024) [8], в которой предложены облегчённые алгоритмы, такие как LRSHA и FLRSHA, предназначенные для устройств с ограниченными ресурсами. Традиционные схемы, такие как ECDSA и Ed25519, демонстрируют высокую вычислительную нагрузку: ECDSA требует 79 185 664 тактов для подписи на AVR ATmega2560, а Ed25519 – 55 553 тактов. Новые схемы LRSHA (облегчённые и устойчивые подписи с аппаратной поддержкой) и FLRSHA (форвард-секьюрный вариант LRSHA) позволяют снизить данные затраты без ущерба для безопасности, обеспечивая оптимизированную производительность, адаптированную к потребностям систем UVS, решая узкие места, присущие традиционным подходам.

Аналогичные проблемы исследуются в работе Явуза и Озмена (2019) [9], где представлена схема SEMECS, оптимизированная для маломощных устройств. ECDSA и Ed25519 в их реализации потребляли 4 818 892 и 23 211 611 тактов соответственно, что подчёркивает необходимость разработки более эффективных решений. SEMECS демонстрирует улучшенные показатели вычислительной эффективности и энергопотребления, что делает её перспективной для систем с ограниченными ресурсами.

Рассмотренные исследования проводились в средах, максимально приближённых к условиям предлагаемого исследования, с использованием эллиптических кривых с сопоставимыми уровнями безопасности и размерами простых полей.

2. Оптимизированный метод ЦП на ЭК

Предлагаемая методика основана на реализации усовершенствованных алгоритмов вычислений на скрученной кривой Эдвардса Edwards25519 с использованием проективных координат ($\mathcal{E}^e$) в сочетании с оконным методом умножения точек. Данный подход позволяет значительно повысить эффективность генерации электронных цифровых подписей на устройствах с ограниченными вычислительными ресурсами.

2.1. Оптимизация ECPM для ресурсоограниченных систем

Свойства циклической группы и противоположного числа могут быть интегрированы с проверенными методами умножения точек [3]. В настоящем исследовании эти свойства интегрируются с оконным методом ECPM при размерах окна $\omega = 4$ и $\omega = 6$ над кривой Edwards25519 с использованием ($\mathcal{E}^e$). Такой подход обеспечивает оптимальный баланс между вычислительной сложностью и потреблением памяти, одновременно используя преимущества циклических свойств группы в сочетании с проверенным методом умножения точек.

2.1.1. Свойство циклической группы точек эллиптической кривой

Ключевым преимуществом эллиптических кривых является их циклическая групповая структура относительно операции сложения точек. Благодаря этому любая точка

Q может быть представлена как кратное базовой точки P: $Q = k \cdot P$, где k – скаляр. Порядок группы $(\#E)$ определяет периодичность: $\#E \cdot P = O$ (точка на бесконечности), $(\#E + 1) \cdot P = P$ и $(\#E + 2) \cdot P = 2P$, продолжая циклически [10]. Это фундаментальное свойство обеспечивает как вычислительную эффективность, так и криптографическую стойкость, основанную на сложности решения задачи дискретного логарифмирования (ECDLP).

Предлагаемый метод оптимизирует вычисления за счет приведения скаляра $\bar{s}$ к диапазону $[0, \#E - 1]$ по формуле:

$$\bar{s} = \begin{cases} s - \#E & \text{если } s \geq \#E; \\ s & \text{если } s < \#E. \end{cases}$$

Такое приведение выполняется однократно в начале вычислений, что минимизирует количество операций умножения точек. Например, вместо вычисления $(\#E + 2)P$ достаточно найти 2P, используя циклические свойства группы.

Особую эффективность метод демонстрирует на скрученных кривых Эдвардса, где симметричные свойства позволяют дополнительно оптимизировать вычисления. При этом достигается не только снижение вычислительной сложности, но и защита от атак по времени выполнения, что критически важно для ресурсоограниченных устройств. Простота операций вычитания делает данный подход особенно выгодным для систем с ограниченными возможностями обработки данных.

2.1.2. Свойство противоположного числа в группах точек ЭК

В теории групп свойство обратного элемента означает, что для каждой точки P существует противоположная точка $-P$ такая, что $P + (-P) = \mathcal{O}$, где $\mathcal{O}$ – единичный элемент (точка на бесконечности). Данное свойство позволяет дополнительно оптимизировать вычисления при работе со скалярами, превышающими половину порядка группы ($\frac{\#E}{2}$). Метод преобразует такие приведённые скаляры $\bar{s}$ по правилу:

$$\bar{s}' = \begin{cases} \bar{s} - \dfrac{\#E}{2}, & \text{если } \bar{s} > 0; \\ \bar{s}, & \text{если } \bar{s} \leq 0. \end{cases}$$

Поскольку точка P', соответствующая $\bar{s}'$, является противоположной, целевая точка Q определяется как:

$$Q_x = \mathcal{P} - P'_x, Q_y = \mathcal{P} - P'_y,$$

где $\mathcal{P}$ – модульное простое число, определяющее конечное поле;

(P'_x, P'_y) – координаты противоположной точки P'.

Такой подход сокращает диапазон используемых скаляров вдвое, что существенно уменьшает количество операций умножения точек, тем самым значительно снижая вычислительную сложность.

2.1.3. Устранение параметра в $\mathcal{E}^e$

Как было описано ранее [4], $\mathcal{E}^e$ упрощает операции сложения за счёт устранения параметров d и a при условии $a = -1$. Однако в операции удвоении точек параметр a

сохраняется в выражении $D \leftarrow a \cdot A$, где $a = -1\ (mod\ \mathcal{P})$, а $\mathcal{P} = 2^{255} - 19$, где $\mathcal{P}$ – простое число, определяющее конечное поле.

В данном контексте значение a представлено в виде 32-байтного массива в шестнадцатеричном формате с порядком байтов little-endian, как это используется при реализации умножения точек ЭК (ECPM) в средах с 8-битными микроконтроллерами:

```
a = {0xEC, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
     0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
     0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
     0xFF, 0x7F}
```

Традиционный подход требует вычисления $D = a \cdot A$ через операцию умножения. В оптимизированном варианте используется свойство поля: значение $D = -A$ может быть получено простым вычитанием A из модуля $\mathcal{P}$, то есть

$$D = \mathcal{P} - A.$$

Данный подход заменяет ресурсоёмкое умножение на более простое вычитание. Применение рассмотренной оптимизации в оконных методах умножения точек позволяет существенно снизить общие вычислительные затраты при выполнении ECPM, уменьшая количество умножений и минимизируя число тактов, что особенно важно для устройств с ограниченными ресурсами. Метод сохраняет криптографическую стойкость операций при значительном повышении их эффективности.

2.1.4. Интеграция с устоявшимися технологиями

Интеграция предлагаемого метода с оконным методом ECMP осуществляется следующим образом:

Шаг 1: Приведение скаляра s к диапазону $[0, \#E - 1]$ через вычитание порядка кривой в зависимости от свойства циклической группы ЭК:

$$s \geq \#E : \{\text{уменьшенный скаляр } \bar{s} = s - \#E\}.$$

Шаг 2: Оптимизированная функция умножения точек ЭК оконным методом:

Вход: $\bar{s}$ или оптимизированный скаляр $\bar{s}'$, базовая точка $G, \#E, \mathcal{P}$, размер окна ω (определяет длину окна), предварительно вычисленные точки $(P, 2P, \dots, P_{2^{\omega}-1})$;

Выход: Аффинные координаты (x, y) результирующей точки $(\mathrm{Q} = \mathrm{s} \cdot \mathrm{G})$.

Процедура:

- Если $(\bar{s} \geq \frac{\#E}{2})$, то используется свойство противоположного числа в группах ЭК:
  - Оптимизированный скаляр $\bar{s} = \bar{s}' - \frac{\#E}{2}$;
  - Повторный вызов функции оптимизированного умножения точек ЭК оконным методом;
  - $Q_x = \mathcal{P}$ – противоположная точка (x – координата в аффинной системе);
  - $Q_y = \mathcal{P}$ – противоположная точка (y – координата в аффинной системе);
  - Возврат.
- $Q = Precomputed\_point$[наиболее значимое окно];

- Главный цикл (интеграция по окнам):
  - Внутренний цикл (итерация ω раз): $Q = 2Q$ (удвоение точки);
  - $Q = Q + Precomputed\_points$[текущее значение окна].

Традиционная реализация оконного метода содержит уязвимость – условное пропускание операции сложения при нулевом значении окна [11], что приводит к различиям во времени выполнения и открывает возможность атак по сторонним каналам [12][13]. Для устранения данной проблемы для каждого размера окна ω вводится предварительно вычисленная точка $0G$, соответствующая единичному элементу в расширенной проективной системе координат скрученных кривых Эдвардса и представленная как $0G = (0:1:0:1)$. При таком подходе операция сложения выполняется всегда, даже для нулевого окна, что гарантирует постоянное время выполнения и защиту от временных атак.

Для размеров окон $\omega = 4$ и $\omega = 6$ это требует расширения наборов предвычисленных точек до 17 и 65 элементов соответственно. Важно отметить, что данная оптимизация, сокращая количество операций умножения за счёт свойств порядка кривой, не влияет на уровень безопасности, который остается равным половине битовой длины порядка группы благодаря сложности решения ECDLP алгоритмом Полларда за $O(\sqrt{\#E})$ операций [14].

2.2. Оптимизированный алгоритм ЭЦП на основе ECC для устройств с ограниченными ресурсами

Оптимизация цифровой подписи включает:

- Использование подхода, аналогичного генерации закрытых ключей в алгоритме цифровой подписи (EdDSA), для генерации детерминированных закрытых однократно используемых чисел ($nonce$); в данном исследовании предлагается генерировать $nonce$ путём объединения сообщения с хешированным значением закрытого ключа (x);
- Исключение публичного однократно используемого числа (R) из вычисления значения задачи $c = H_{sig}(X, R, m)$.

Для подписания сообщения (m) отправитель выполняет следующие шаги:

1) Выбор закрытого ключа (x) случайным образом, такого, что $0 < x < \mathcal{P}$, hash = SHA-512(x), x = hash[0:32], prefix = hash[32:64];
2) Вычисление открытого ключа (X), такой что $X = G^x$;
3) Вычисление $h = H(prefix)$;
4) Вычисление закрытого однократно используемого числа $r = H(h||m)$;
5) Вычисление открытого однократно используемого числа $R = G^r$;
6) Вычисление $c = H(X, m)$;
7) Вычисление $s \equiv (r + cx)$.

Подписью является пара (R, s). Таким образом, отправитель передаёт сообщение (m), открытый ключ (X), (R) и (s) получателю, который считает подпись действительной только если $G^s = RX^c$.

Доказательство корректности алгоритма:

$G^s = RX^c$, так как $R = G^r$ и $X = G^x$, тогда $G^s = G^r(G^x)^c = G^{r+xc} = G^s$, где $s = r + xc$.

На рисунке 1 представлена схема выполнения операций подписи и проверки.

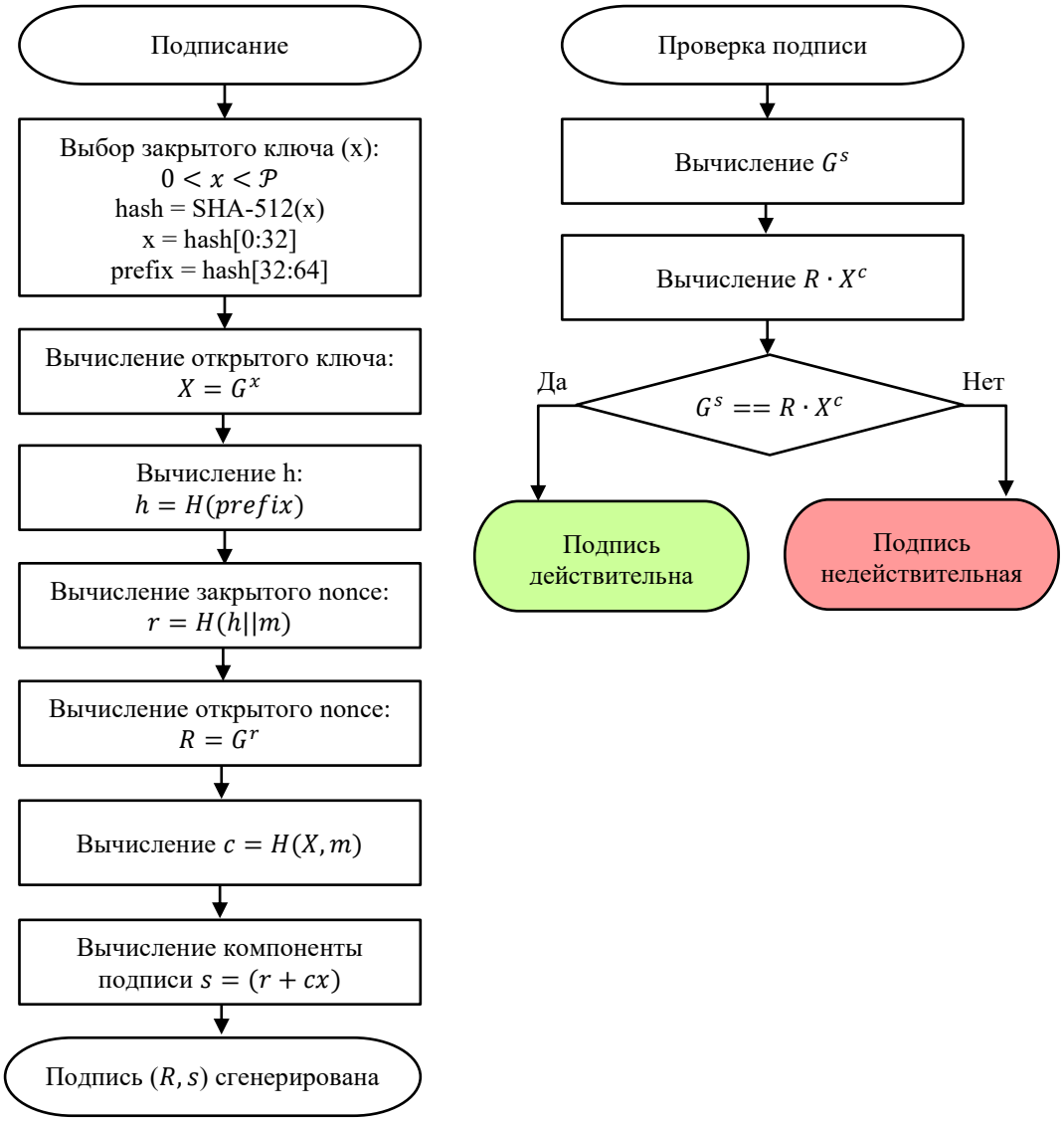

Рисунок 1 – Схема процесса подписания и верификации

3. Реализация метода на микроконтроллере Atmega2560

Алгоритмы цифровой подписи обычно включают три основных этапа: генерация ключей, подписание и проверка [2]. Процесс генерации ключей состоит из трёх шагов, обеспечивающих безопасную генерацию закрытого и открытого ключей.

1) Генерация случайных чисел (RNG): предлагаемый метод использует аппаратную энтропию с аналогового входа на пине A0, который захватывает шум окружающей среды, в качестве начального значения. Потоковый шифр ChaCha20, известный своей криптографической стойкостью, преобразует данную энтропию в равномерно распределённый 32-байтный случайный вывод [15]. Криптографические шифры и хэш-функции служат высококачественными генераторами псевдослучайных чисел [15], обеспечивая надёжную генерацию для формирования префикса и закрытого ключа. Данный этап требует 117 618 тактов.

2) Хеширование: 512-битное хеширование RNG-значения даёт 64-байтный результат, разделяемый на закрытый ключ (нижние 32 байта) и префикс (верхние 32 байта) для генерации nonce [2]. Процесс занимает 277 575 тактов.

3) Вычисление открытого ключа: закрытый ключ умножается на базовую точку для получения соответствующего открытого ключа с использованием предлагаемого метода [3] при $\omega = 4$ и $\omega = 6$. Данный этап требует 14 597 191 такт при $\omega = 4$ и 13 451 712 тактов при $\omega = 6$.

Производительность системы существенно зависит от выбранного размера окна ω. При ω =4 процесс занимает 15 105 854 тактов, используя 30 518 байт (12%) из доступных 253 952 байт программной памяти (flash) и 429 байт (5%) динамической памяти (SRAM). Оптимизация с $\omega = 6$ сокращает время до 13 863 416 тактов при увеличении использования flash-до 37 584 байта (14%) с аналогичным потреблением SRAM.

Для корректной оценки эффективности критически важно сравнивать результаты только с исследованиями, использовавшими аналогичную аппаратную платформу (ATmega2560) и криптографическую кривую Edwards25519 [16]. Сравнения с системами, имеющими существенно отличающиеся характеристики, не позволяют объективно оценить преимущества предложенного метода, что подтверждается данными в таблице 1.

Таблица 1 – Сравнение генерации ключей в сопоставимых условиях

Реализация	Количество тактов	SRAM
crypto_sign_keypair(Low-Area) (ω = 2)	32937940	1282 байт
crypto_sign_keypair(Low-Area) (ω = 2)	21924771	1566 байт
Предлагаемый метод (ω = 4)	15105854	429 байт
Предлагаемый метод (ω = 6)	13863416	433 байт

На рисунке 2 также проиллюстрировано рассматриваемое сравнение, где красным цветом отображены предыдущие исследование, а зелёным – предлагаемый метод.

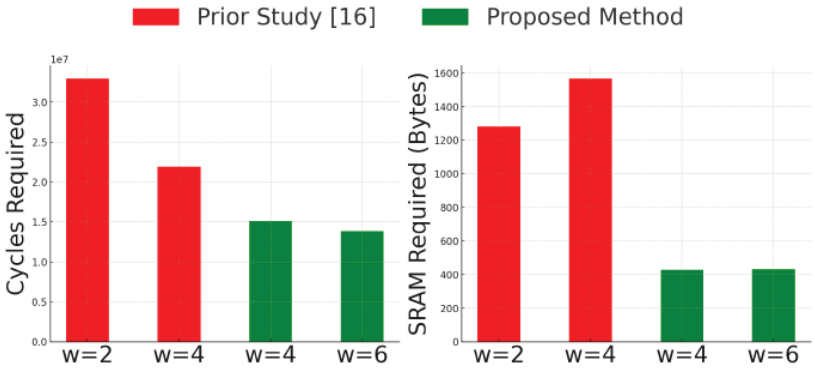

Рисунок 2 – Сравнение генерации ключей в сопоставимых условиях

Предложенный метод демонстрирует значительное улучшение производительности по сравнению с предыдущими реализациями. При размере окна $\omega = 4$ наблюдается сокращение количества тактов на 31% и 54%, а при $\omega = 6$ – на 37% и 58%. Экономия динамической памяти (SRAM) составляет 66% и 73% для обоих вариантов, что особенно важно для ресурсоограниченных устройств. В процессе подписания метод показывает следующие характеристики:

- При ω=4: 15 435 539 тактов, 25 652 байта flash-памяти (10%), 639 байт SRAM (7%);
- При ω=6: 14 286 257 тактов, 32 284 байта flash-памяти (12%), при сохранении того же объёма SRAM.

Сравнение производительности проводилось только с исследованиями, использовавшими аналогичные аппаратные платформы и криптографические параметры, что обеспечивает достоверность оценки эффективности предложенного метода. Результаты сравнения наглядно представлены в таблице, подтверждая преимущества данного подхода для систем с ограниченными ресурсами.

Таблица 2 – Сравнение количества таков в сопоставимых условиях

Метод	Реализация	Количество тактов
crypto_sign [16]	Высокоскоростная, $\omega = 4$ Малозатратная, $\omega = 2$	23211611 34342230
Ed25519 [8]	–	22688583
Предлагаемый метод	$\omega = 4$ $\omega = 6$	15435539 14286257

На рисунке 3 приведено сравнение в аналогичных условиях ограниченных ресурсов.

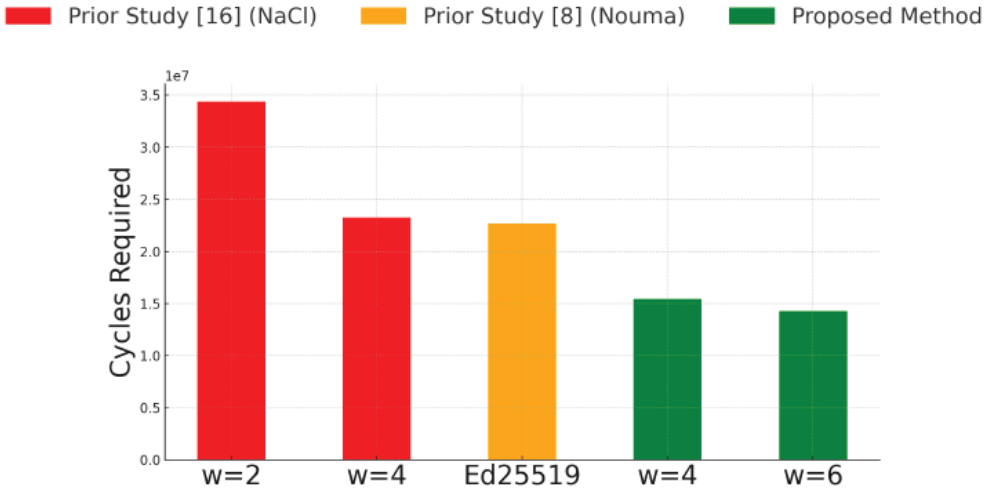

Рисунок 3 – Сравнение количества тактов в сопоставимых условиях

Метод демонстрирует значительное повышение эффективности: при $\omega = 4$ наблюдается улучшение на 32% и 34% (для малозатратной и высокоскоростной реализации соответственно), а при $\omega = 6$ – на 37% и 40% по сравнению с предыдущими решениями. Особенно заметно сокращение использования SRAM – на 50% и 59% (639 байт против 1282-1566 байт в других реализациях). На этапе проверки подписи:

- При $\omega = 4$: 30 521 674 циклов, 28 624 байта flash (11%), 1 187 байт SRAM (14%);
- При $\omega = 6$: 28 230 736 циклов, 45 711 байт flash (18%), при сохранении потребления SRAM.

Сравнение с аналогичными исследованиями [18], представленное в таблице 3 и на рисунке 4, подтверждает преимущества метода в условиях ограниченных ресурсов, устанавливая новый ориентир эффективности для подобных систем.

Таблица 3 – Сравнение результатов этапа проверки в сопоставимых условиях

Реализация	Размер окна	Количество циклов	SRAM
Low-Area [16]	2	40093186	1349 байт
High-Speed [16]	4	32619197	1317 байт
Предлагаемый метод	4	30521674	1187 байт
Предлагаемый метод	6	28230736	1187 байт

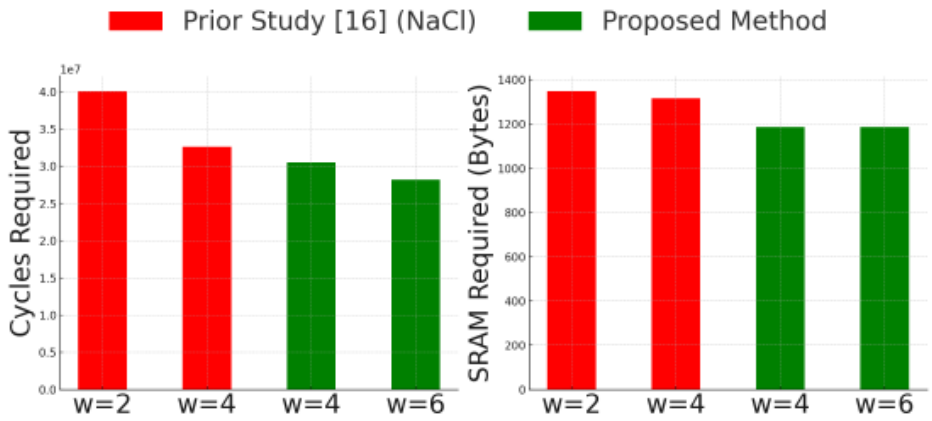

Рисунок 4 – Сравнение результатов этапа проверки в сопоставимых условиях

На этапе верификации предлагаемый метод демонстрирует умеренные, но стабильные улучшения. При $\omega = 4$ достигается сокращение циклов на 6,44% по сравнению с высокоскоростной реализацией и на 23,87% по сравнению с малогабаритной реализацией с экономией SRAM 9,87% и 12% соответственно. Оптимизация с $\omega = 6$ показывает лучшие результаты – снижение на 13% и 30% при сохранении аналогичной экономии памяти и размера окна. Однако прогресс менее выражен, чем на этапах генерации ключей и подписания, что объясняется фундаментальной вычислительной сложностью операций верификации, включающих двойное умножение точек ЭК, хеширование и другие ресурсоемкие операции. Данные ограничения определяют естественный предел для возможных оптимизаций на данном этапе.

Заключение

Предложенный оптимизированный метод ЭЦП демонстрирует значительное повышение эффективности ECC-подписей на микроконтроллере ATmega2560, являющимся платформой с ограниченными ресурсами.

На этапе генерации ключей достигнуто сокращение процессорных циклов до 54% и экономия SRAM до 73% по сравнению с существующими решениями [16]. Этап подписания показывает сопоставимые результаты – уменьшение вычислительной нагрузки на 40% при сокращении использования памяти на 59%, что стало возможным благодаря интеграции детерминированной генерации nonce и оптимизированного скалярного умножения.

Верификация подписей, в силу своей алгоритмической сложности (двойное умножение точек, хеширование), демонстрирует более скромные улучшения – до 29,6% сокращения циклов (и 6,5% при использовании того же размера окна 4) со снижением использования SRAM до 12%. Данные результаты, хотя и менее впечатляющие из-за ресурсоёмкой природы процесса проверки, всё же представляют значительный прогресс для данного этапа.

Перспективы развития предложенного метода включают его адаптацию для более широкого круга архитектур, например, для платформ на базе ARM, более новые модели Arduino и серия STM32N6 от STMicroelectronics. Кроме того, данный метод предполагает дальнейшую математическую оптимизацию операций и интеграцию в блокчейн-системы для повышения эффективности транзакций, что является важным направлением развития.

Полученные результаты подтверждают жизнеспособность предложенного подхода как эффективного решения для реализации ECC-подписей в условиях ограниченных ресурсов, сочетающего криптографическую стойкость с высокой производительностью.

Благодарности

Работа выполнена в рамках государственного задания Министерства науки и высшего образования Российской Федерации № 075-00003-24-02 от 08.02.2024 (проект FSEE-2024-0003).

Список литературы

1. The elliptic curve digital signature algorithm (ecdsa) / Don Johnson, Alfred Menezes, and Scott Vanstone // International Journal of Information Security, 1:36–63, 2001. URL: https://doi.org/10.1007/s102070100002;
2. Edwards-Curve Digital Signature Algorithm (EdDSA) / Simon Josefsson and Ilari Liusvaara // RFC 8032, January 2017;
3. An optimized point multiplication strategy in elliptic curve cryptography for resource-constrained devices / Nawras H Sabbry and Alla B Levina // Mathematics, 12(6):881, 2024. URL: https://www.mdpi.com/2227-7390/12/6/881;
4. Twisted edwards curves revisited / Huseyin Hisil, Kenneth Koon-Ho Wong, Gary Carter, and Ed Dawson // In International Conference on the Theory and Application of Cryptology and Information Security, pages 326–343. Springer, 2008. URL: https://link.springer.com/chapter/10.1007/978-3-540-89255-7_20;
5. Nonce generation techniques in schnorr multi-signatures: Exploring eddsa-inspired approaches / Nawras H Sabbry and Alla Levina // AIMS Mathematics, 9(8):2030420325, 2024. URL: https://www.aimspress.com/article/doi/10.3934/math.2024988;
6. Curve25519 based lightweight end-to-end encryption in resource constrained autonomous 8-bit IoT devices / Shafi Ullah and Raja Zahilah // Cybersecurity, 4:1–13, 2021;
7. X2065: Lightweight key exchange for the internet of things / Georgios Fotiadis, Johann Großschädl, and Peter YA Ryan // In Proceedings of the 10th ACM Cyber-Physical System Security Workshop, pages 43–52, 2024;
8. Lightweight and resilient signatures for cloud-assisted embedded iot systems / Saif E Nouma and Attila A Yavuz // arXiv preprint arXiv:2409.13937, 2024;
9. Ultra lightweight multiple-time digital signature for the internet of things devices / Attila Altay Yavuz and Muslum Ozgur Ozmen // IEEE Transactions on Services Computing, 15(1):215–227, 2019. URL: https://ieeexplore.ieee.org/document/8762164;
10. Understanding Cryptography: From Established Symmetric and Asymmetric Ciphers to Post-Quantum Algorithms / Christof Paar, Jan Pelzl, and Tim Güneysu // Springer, 2024. URL: https://doi.org/10.1007/978-3-662-69007-9;
11. Guide to elliptic curve cryptography / Darrel Hankerson, Scott A. Vanstone, and Alfred Menezes // Springer Professional Computing, 2004;
12. Improved elliptic curve multiplication methods resistant against side channel attacks / Tetsuya Izu, Bodo Möller, and Tsuyoshi Takagi // In Progress in Cryptology—INDOCRYPT 2002: Third International Conference on Cryptology in India Hyderabad, India, December 16–18, 2002 Proceedings 3, pages 296–313. Springer, 2002;

13. New regular sliding window algorithms for elliptic curve scalar point multiplication / NN Shenets and AS Petushkov // Automatic Control and Computer Sciences, 55(8):1029–1038, 2021;
14. Speeding up the pollard's rho algorithm / Silje Christensen and Simen Johnsrud, 2015;
15. Enhancement the chacha20 encryption algorithm based on chaotic maps / Hussain H Alyas and Alharith A Abdullah // In Next Generation of Internet of Things: Proceedings of ICNGIoT 2021, pages 91–107. Springer, 2021;
16. Nacl on 8-bit avr microcontrollers / Michael Hutter and Peter Schwabe // In Progress in Cryptology - AFRICACRYPT 2013: 6th International Conference on Cryptology in Africa, Cairo, Egypt, June 22-24, 2013. Proceedings 6, pages 156–172. Springer, 2013.