

ИНТЕЛЛЕКТУАЛЬНАЯ КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИОТ ДАННЫХ НА ОСНОВЕ ЭЛЛИПТИЧЕСКИХ КРИВЫХ

ТАШТАЙ Б.А.*, КОНЫРХАНОВА А.А., АТАНОВ С.К.

***Таштай Бактыгелді Алтайұлы** – Магистр, докторант кафедры информационной безопасности, Евразийский национальный университет имени Л.Н.Гумилева, г. Астана, Казахстан

E-mail: 93bahti93@gmail.com, <https://orcid.org/0000-0003-4952-369X>

Конырханова Асем Адилбекқызы – PhD, и.о.доцента кафедры «Информационная безопасность», Евразийский национальный университет имени Л.Н.Гумилева, г. Астана, Казахстан

E-mail: konyrkhanova_a@enu.kz, <https://orcid.org/0000-0002-4901-8901>

Атанов Сабыржан Кубейсинович – Доктор технических наук, профессор, и.о. профессора кафедры «Компьютерной и программной инженерии», Евразийский национальный университет имени Л.Н.Гумилева, г. Астана, Казахстан

E-mail: atanov5@mail.ru, <https://orcid.org/0000-0003-2115-7130>

Аннотация. Данное исследование посвящено актуальной задаче разработки эффективных и безопасных криптографических решений для систем Интернета вещей (IoT), где устройства работают в условиях ограниченных вычислительных мощностей, памяти и энергопотребления. На фоне быстрого распространения IoT-устройств на базе микроконтроллеров Embedded Systems (ESP32) и растущих угроз кибербезопасности проводится систематический анализ аппаратных и программных методов шифрования. В статье исследуется реализация криптографической защиты на уровне микроконтроллера в сетях IoT, с особым акцентом на время выполнения, энергоэффективность и использование в условиях ограниченных аппаратных ресурсов. Принципиальным отличием реализованных и описанных программы является отсутствие внешних библиотек, что позволяет ее реализовывать на любых типах ЭВМ, вплоть до микроконтроллеров. Алгоритм программы адаптирован для микроконтроллеров (ESP32) и объединяет алгоритм генерации ключей на основе эллиптических кривых с машинным обучением для обнаружения атак в реальном времени. Процедура обмена ключами реализована по протоколу ECDH (Elliptic Curve Diffie-Hellman) - протокол Диффи-Хеллмана безопасного обмена ключами на основе эллиптических кривых с использованием библиотеки Micro-ECC. Для машинного обучения были использованы нейронные сети.

Ключевые слова: микроконтроллер, эллиптические кривые, генерация ключей, криптографическая защита информации, цифровая безопасность, нейронные сети, Интернет вещей (IoT).

Введение

Интернет вещей (IoT) представляют собой компьютерную сеть физических устройств, которые связаны друг с другом или сервером через Интернет. При этом незащищённые устройства IoT и их данные подвергаются риску взлома и хищения конфиденциальной информации.

Интернет вещей стал одной из ключевых технологических парадигм последних лет, обеспечивая повсеместную связь между датчиками, встроенными устройствами и облачными сервисами. Масштабы экосистемы быстро растут: по прогнозам, к 2030 году число подключенных устройств достигнет около 40 миллиардов [1], формируя новые сценарии автоматизации в промышленности, энергетике, здравоохранении и «умной» городской среде. В то же время задача защиты данных и доверенных вычислений на самом «периферийном» уровне сети приобретает критически важное значение — в условиях ограниченной вычислительной мощности, памяти и энергопотребления узлов.

Одновременно с расширением Интернета вещей (IoT) проблемы кибербезопасности становятся все более острыми. Гетерогенность аппаратной базы, упрощённые стеки протоколов, неполная аутентификация и контроль доступа делают устройства мишенью для широкого спектра атак. В отличие от предыдущих исследований, которые фокусировались на изолированных реализациях, в данной работе предлагается универсальная адаптивная система выбора

шифрования, адаптированная для микроконтроллерных сред с ограниченными ресурсами. Ключевым вкладом является реализация и тестирование этой системы на платформах ESP32 и ESP32-S3. Результаты показывают, что аппаратное шифрование значительно превосходит программные методы по скорости выполнения и энергоэффективности. Также в статье определены перспективные направления для будущих исследований, включая разработку защищённых цифровых двойников для систем здравоохранения и автономных устройств.

Согласно исследованиям, около 70% устройств IoT имеют уязвимости [2], что подтверждается практикой выявления угроз в реальных условиях эксплуатации [3]. Среди основных задач – защита конфиденциальности данных, обеспечение их целостности и разработка криптографически стабильных, но легковесных решений, совместимых с ограничениями ресурсов микроконтроллеров [4]. Анализ научной литературы показывает, что проблема обеспечения безопасности Интернета вещей носит комплексный характер и охватывает вопросы конфиденциальности, аутентификации, контроля доступа, защиты персональных данных, доверия между узлами сети, а также устойчивости IoT-инфраструктуры к широкому спектру киберугроз. В фундаментальной работе Sicari, Rizzardi, Grieco и Coen-Porisini [5] обосновали, что традиционные подходы информационной безопасности не могут быть непосредственно перенесены в IoT-среду вследствие ограниченности вычислительных и энергетических ресурсов устройств. В работе авторы Meneghello, Calore, Zucchetto и соавт [3] дополняют данное положение результатами анализа реальных уязвимостей коммерческих IoT-решений. Ключевыми проблемами они считают что многие устройства остаются подверженными атакам из-за слабой аутентификации, недостаточной криптографической защиты и отсутствия безопасного проектирования на ранних этапах разработки. Так в работе Conti, Dehghantanha, Franke и Watson [4] проблема безопасности IoT рассматривается также с позиций цифровой криминалистики, при этом авторы подчёркивают, что распределённость, автономность и неоднородность IoT-устройств существенно осложняют сбор, сохранение и анализ цифровых следов инцидентов.

Современные направления развития средств защиты IoT представлены в обзоре Tariq, Ahmed, Bashir и Shaukat [6], где систематизированы угрозы по уровням архитектуры, в данном исследовании проанализированы уязвимости протоколов и выделены перспективные подходы, включая лёгкую криптографию, безопасные OTA-обновления, применение AI/ML, блокчейн и постквантовая криптография. Прикладной аспект криптографической защиты раскрыт в исследовании Rozlomii, Yarmilko, Naumenko и соавт [1], в котором установлено, что аппаратная реализация шифрования на платформе STM32F407VG превосходит программную реализацию на ESP32-WROOM-32 по показателям производительности, энергопотребления и защищённости ключевого материала. Существенный интерес для развития интеллектуальных механизмов защиты представляет работа Ullah, Naeem, Jabbar и соавт [7], подтверждающая высокую эффективность методов глубокого обучения при обнаружении киберугроз в IoT-сетях по сравнению с традиционными подходами. Дополнительную актуальность рассматриваемой проблематике придаёт аналитический обзор S. Sinha [2], фиксирующий дальнейший рост числа подключённых IoT-устройств, усиление регуляторных требований к безопасности и повышение роли защищённых микроконтроллеров и элементов криптозащиты в современной цифровой инфраструктуре (IoT Analytics). Таким образом, проанализированные источники позволяют заключить, что обеспечение кибербезопасности IoT требует комплексного сочетания архитектурно встроенных механизмов защиты, аппаратно-программных криптографических средств и интеллектуальных методов выявления аномалий и атак. Поэтому авторы считают эту тему раскрытой не в полной мере и предлагают собственное исследование

Представленная авторами работа предлагает комплексный подход, сочетающий систематическое сравнение аппаратно-ускоренных и программно-ориентированных криптографических методов на платформах ESP32/ESP32-S3 с динамическим выбором

алгоритмов на основе машинного обучения. Эта всеобъемлющая структура обеспечивает оптимизированную криптографическую производительность в реальном времени, адаптированную к уникальным ограничениям ресурсоемких сред Интернета вещей.

Материалы и методы исследования

Архитектура и физическая реализация стенда для исследования производительности интеллектуального шифрования микроконтроллеров для IoT приведена на рисунке 1. Аппаратная часть использует микроконтроллерные платы разработки Arduino популярных архитектур микроконтроллеров AVR (8-бит) и ARM (32/64-бит). Программное обеспечение было создано в интегрированной среде разработки Arduino (IDE) совместно со средой программирования Spyder 6 для расчетов на языке Python с использованием низкоуровневых библиотек шифрования. Передача данных с сенсоров IoT на персональный компьютер организована через радиоканал, чтобы измерения максимально приблизить к реальным условиям.

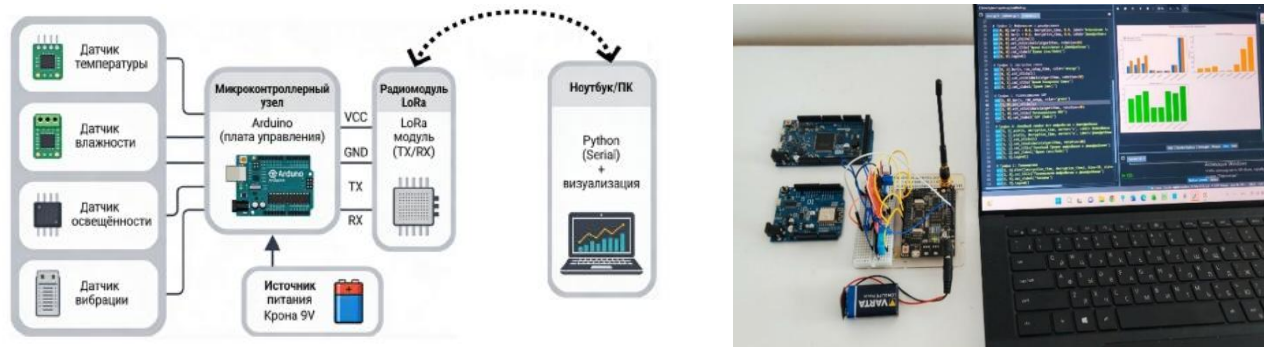


Рисунок 1. Стенд для исследования защиты IoT данных на основе эллиптических кривых

Компьютер через USB-интерфейс соединяется с микроконтроллерами и обеспечивает выбор режимов, компиляцию программ (например, в среде Arduino IDE), загрузку прошивок, мониторинг и визуализацию результатов. Функциональные возможности разработанных программ это интеллектуальная криптографическая защита устройств IoT с использованием нейронных сетей. А также расчёт и визуализации данных устройств IoT

Защита данных строилась на протоколе ECDH (Elliptic Curve Diffie-Hellman). Он адаптирован для микроконтроллеров. ECDH реализует способ обмена ключами, базирующийся на эллиптических кривых. В данном алгоритме две стороны могут по открытому каналу создать общий секретный ключ. Далее общая или открытая часть этого ключа используется для шифрования данных. Безопасность обеспечивает сложность решения обратной задачи дискретных логарифмов на эллиптических кривых. При этом ECDH даёт такой же уровень защиты, как стандартный алгоритм Диффи-Хеллман, но с гораздо меньшими ключами. Так, например, 256-битный ключ ECDH равен по стойкости 3072-битному ключу RSA. Меньший размер ключа означает более быструю генерация и меньше данных для передачи. Такой подход оптимален для устройств с ограниченными ресурсами и это особенно это важно для IoT-устройств.

Разработка программного обеспечения осуществлялась с использованием интегрированной среды разработки Arduino (IDE) и фреймворка разработки Espressif IoT (ESP-IDF), интегрированных с расширениями Visual Studio Code. Для реализации криптографических алгоритмов использовались хорошо зарекомендовавшие себя библиотеки — Embedded Transport Layer Security (mbed TLS), TinyAES и AESLibre, представляющие современные подходы к программной криптографии в системах с ограниченными ресурсами.

В дополнение к основным экспериментам, был реализован компонент машинного обучения (ML) для моделирования взаимосвязи между характеристиками производительности

криптографических алгоритмов и выбором оптимального алгоритма в конкретных условиях. Для каждой платформы были сгенерированы обучающие наборы данных, включающие входные параметры, время выполнения и энергопотребление. Разработанная методологическая основа предлагает воспроизводимый и масштабируемый подход к изучению производительности криптографических алгоритмов в средах Интернета вещей с ограниченными ресурсами и демонстрирует потенциал методов машинного обучения для динамической оптимизации криптографических решений на встроенных устройствах. В совокупности эти процедуры представляют собой практическую основу для сравнительной оценки вычислительных и энергетических характеристик систем шифрования в средах Интернета вещей.

Для проведения, анализа и визуализации производительности различных алгоритмов шифрования/дешифрования данных с IoT устройств были разработаны программы на языке Python и для визуализации выходных данных был использован модуль Matplotlib.

Программа реализует интеллектуальную систему криптографической защиты на базе алгоритмов Micro-ECC и TinyML для микроконтроллеров, объединяющую эллиптические кривые для шифрования с машинным обучением для обнаружения атак в реальном времени.

Рассмотрим основные модули программно-аппаратной реализации интеллектуальной криптографической защиты данных для IoT. Разработанная программно-аппаратная системы предназначен для интеллектуальной криптографической защиты данных в сети Интернет вещей. Ключевой особенностью предложенного решения является его ориентация на устройства с ограниченными вычислительными и энергетическими ресурсами, прежде всего на микроконтроллеры семейства ESP32. В отличие от ресурсоемких криптографических стеков общего назначения, система реализуется средствами компактного C/C++-кода и не требует использования тяжеловесных платформенно-зависимых пакетов. Криптографическое ядро построено на основе облегченной реализации обмена ключами по протоколу ECDH, архитектурно совместимой с библиотекой Micro-ECC, а интеллектуальный модуль анализа реализован в парадигме TinyML, то есть в виде малой нейросетевой модели, исполняемой непосредственно на микроконтроллере (micro-ecc GitHub, Google AI Edge).

В основе защищенного взаимодействия узлов лежит протокол ECDH (Elliptic Curve Diffie-Hellman), обеспечивающий выработку общего секрета между двумя сторонами без передачи закрытого ключа по открытому каналу. На каждой стороне формируется случайный закрытый ключ $d \in [1, n-1]$, после чего вычисляется открытый ключ $Q = dG$, где G — базовая точка выбранной эллиптической кривой. После обмена открытыми ключами каждая сторона вычисляет общий секрет $S = dQ_{peer}$. При правильной работе протокола обе стороны получают одинаковый результат. Этот результат затем обрабатывается функцией выработки ключа (KDF). В итоге получается симметричный сеансовый ключ. Он используется для шифрования передаваемых данных.

Для встраиваемой реализации была выбрана кривая из семейства эллиптических типа secp256r1. Она обеспечивает 128-битный уровень защиты. При этом требования к памяти и времени вычислений остаются достаточными для микроконтроллерной реализации. Сама эллиптическая кривая используемая для шифрования, имеет вид:

$$y^2 = x^3 + ax + b(modp),$$

где — большое 256-битное простое число.

Используемые при получении ключей операция сложения и инвертирование для получения пары ключей не имеют вычислительной сложности. Но обратить эту операцию обратно за разумное время невозможно даже на суперкомпьютере. Это и обеспечивает безопасность шифрования за счёт сложности вычисления закрытых (приватных) ключей.

Выбор именно secp256r1 связан с особенностями библиотеки Micro-ECC. Она оптимизирована под малый объём кода и не использует динамическое выделение памяти. Кроме secp256r1, поддерживаются secp160r1, secp192r1, secp224r1 и secp256k1. Всё это делает secp256r1 логичным выбором для защищённых IoT-устройств. В частности, для телеметрических и сенсорных или иных систем с ограниченными вычислительными ресурсами.

Ниже приведён на рисунке 2 алгоритм защищённого обмена ключами ECDH

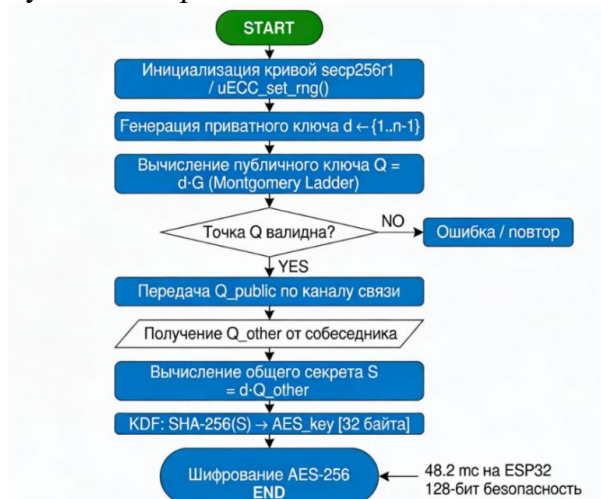


Рисунок 2. Алгоритма защищённого обмена ключами

Ниже приведён псевдокод алгоритма защищённого обмена ключами ECDH

Входные данные

curve— параметры выбранной эллиптической кривой

G— базовая точка кривой

N— порядок базовой точки

ID_A, ID_B — идентификаторы взаимодействующих устройств

Выходные данные

K_session — общий сеансовый симметричный ключ

Процедура ECDH_Key_Exchange(curve, G, n)

инициализация параметров эллиптической кривой curve

пользователь A генерирует случайный закрытый ключ: $d_A \leftarrow \text{Random}(1, n - 1)$

пользователь B генерирует случайный закрытый ключ: $d_B \leftarrow \text{Random}(1, n - 1)$

пользователь A вычисляет открытый ключ: $Q_A \leftarrow d_A \cdot G$

пользователь B вычисляет открытый ключ: $Q_B \leftarrow d_B \cdot G$

пользователь B передает Q_B пользователю A по открытому каналу

если NOT IsValidPublicKey(Q_B , curve), то завершить протокол с ошибкой

если NOT IsValidPublicKey(Q_A , curve), то завершить протокол с ошибкой

пользователь A вычисляет общий секрет: $S_A \leftarrow d_A \cdot Q_B$

пользователь B вычисляет общий секрет: $S_B \leftarrow d_B \cdot Q_A$

если $S_A = O$ или $S_B = O$, то завершить протокол с ошибкой, где O — бесконечно удалённая точка

пользователь A формирует сеансовый ключ: $K_A \leftarrow \text{KDF}(x(S_A), \text{ID}_A, \text{ID}_B)$

пользователь B формирует сеансовый ключ: $K_B \leftarrow \text{KDF}(x(S_B), \text{ID}_A, \text{ID}_B)$

если $K_A \neq K_B$, то завершить протокол с ошибкой

$K_{\text{session}} \leftarrow K_A$

использовать K_{session} для симметричного шифрования и аутентифицированного обмена данными:

$C \leftarrow \text{AEAD_Encrypt}(K_session, \text{plaintext}, \text{nonce}, \text{AAD})$

$\text{plaintext} \leftarrow \text{AEAD_Decrypt}(K_session, C, \text{nonce}, \text{AAD})$

вернуть $K_session$

конец процедуры

На первом этапе обе стороны инициализируют параметры выбранной эллиптической кривой и независимо генерируют свои закрытые ключи. Затем путём скалярного умножения базовой точки кривой вычисляются открытые ключи, которые передаются по открытому каналу связи. После проверки корректности полученных открытых ключей каждая сторона вычисляет общий секрет. Далее общий секрет преобразуется функцией выработки ключа KDF в общий сеансовый симметричный ключ. Полученный ключ используется для последующего симметричного шифрования и аутентифицированного обмена данными между устройствами.

Структура данной схемы отражает принципиально важное свойство протокола ECDH: закрытые ключи не передаются по каналу связи и остаются исключительно на стороне каждого устройства.

Интеллектуальная часть системы предназначена для обнаружения аномалий в процессе криптографических операций. Так в качестве диагностического признака берётся время выполнения ECDH, параметр - отражающий производительность вычислительной системы. Последовательность из N последних измерений (в этом случае $N = 20$) сохраняется в буфере признаков. Далее полученный вектор нормализуется по формуле $Z(i) = (x(i) - \mu) / \sigma$, где $x(i)$ - текущее значение признака, μ - среднее эталонного режима, σ - среднеквадратичное отклонение.

Далее нормализованный вектор подаётся на вход компактной нейросети TinyML и служит нормой для анализа аномалий. Встраиваемые нейросети такого типа записываются в память микроконтроллеров и для них не нужна операционная система и динамическое выделение памяти.

На выходе модели получаем значение $p \in [0,1]$. Его интерпретируют как вероятность аномалии. Решение принимается по порогу: если $p > 0.8$, система фиксирует сбой. Она блокирует текущую сессию. Затем иницируется новая генерация и обмен ключами. Если $p \leq 0.8$, работа продолжается в обычном режиме. Такой подход объединяет криптографию и интеллектуальный мониторинг. Всё работает в реальном времени на микроконтроллере IoT устройства.

Схема интеллектуального детектора аномалий состоит из следующих этапов: запуск цикла мониторинга, регистрация временных и поведенческих признаков во время выполнения криптографических операций, накопление последовательности измерений в скользящем буфере, нормализация входных данных, передача вектора признаков в TinyML-модель, вычисление вероятности аномалии и принятие решения на основе порогового сравнения (рисунок 3).

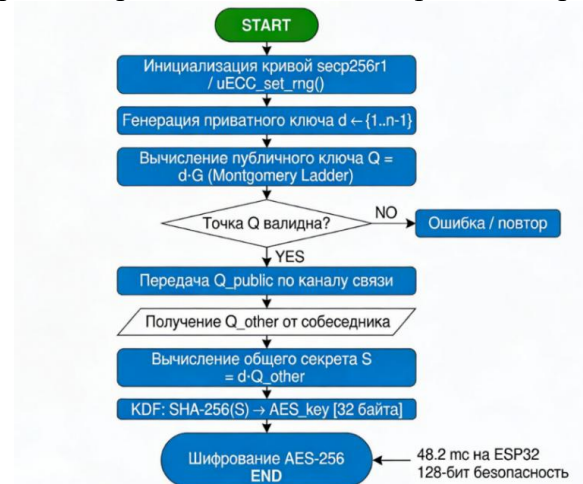


Рисунок 3. Алгоритм интеллектуального детектора аномалий

При обнаружении аномального состояния активируется блок защитной реакции, включающий разрыв текущей сессии, регенерацию ключей и повторную инициализацию защищённого соединения. При отсутствии отклонений программа продолжает работу в рабочем режиме. Ниже приведён псевдокод детектирование аномалий на основе TinyML

Вход:

x_t — текущее значение временного или поведенческого признака

Buffer[N] — буфер последних N измерений

μ — среднее значение эталонного режима

σ — среднеквадратическое отклонение

$\tau = 0.8$ — порог обнаружения аномалии

Model — обученная TinyML-модель

Выход:

p — вероятность аномального режима

state — состояние системы: NORMAL или ANOMALY

Процедура TinyML_Anomaly_Detection(x_t)

1: Считать текущее значение признака x_t

2: Обновить буфер последних N измерений: Buffer $\leftarrow$ Push(Buffer, x_t)

3: Если Buffer не заполнен, то вернуть WAIT Конец условия

4: Для i от 1 до N выполнить $z_i \leftarrow (\text{Buffer}[i] - \mu) / \sigma$

Конец цикла

5: Сформировать нормализованный входной вектор: $Z \leftarrow [z_1, z_2, \dots, z_N]$

6: Передать вектор Z на вход модели: $p \leftarrow \text{Model_Predict}(Z)$

7: Если $p > \tau$, то state $\leftarrow$ ANOMALY

Разорвать текущую сессию

Заблокировать обмен данными

Запустить повторный обмен ключами по ECDH

Иначе

state $\leftarrow$ NORMAL

Продолжить штатную работу криптографического протокола

Конец условия

8: Вернуть p , state

9: Повторять процедуру в режиме реального времени

Конец процедуры

Алгоритм детектирования аномалий на основе TinyML предназначен для непрерывного мониторинга процесса выполнения криптографических операций. На каждом шаге работы системы считывается текущее значение временного либо иного поведенческого признака, связанного с выполнением ECDH, после чего новое измерение помещается в буфер последних наблюдений. После заполнения буфера выполняется нормализация входного вектора относительно параметров эталонного режима.

Сформированный нормализованный вектор признаков передаётся на вход предварительно обученной TinyML-модели, которая вычисляет вероятность аномального состояния p . Затем полученное значение сравнивается с порогом $\tau = 0.8$. Если вероятность аномального состояния превышает заданный порог, система интерпретирует текущее состояние как атаку на данные. Система при этом разрывает текущую сессию связи и иницирует повторный обмен ключами. Если значение аномального состояния не превышает заданный порог, криптографический протокол продолжает работу по передаче зашифрованных данных. Такой подход обеспечивает интеллектуальный контроль безопасности данных IoT устройств в реальном времени.

Результаты и их обсуждение

Проведём анализ полученных данных работы программного комплекса в виде таблиц и графиков для различного уровня данных обучения нейросети. Ниже в сводной таблице приведены результаты эмуляции нейросетевого детектора аномалий при 6 уровнях обучения.

Таблица 1. Сводная таблица результатов

Этапы обучения	Точность (%)	ROC-AUC	F1-Score	FPR (%)	FNR (%)	P(атака)	Обнаружена
5 эпох	80.5	0.0029	0.000	0.0	100.0	0.193	НЕТ X
20 эпох	80.5	0.9971	0.000	0.0	100.0	0.208	НЕТ X
50 эпох	80.5	0.9971	0.000	0.0	100.0	0.227	НЕТ X
100 эпох	80.5	0.9971	0.000	0.0	100.0	0.324	НЕТ X
200 эпох	96.5	0.9971	0.901	0.0	17.95	0.722	ДА ✓
500 эпох	97.0	0.9971	0.917	0.0	15.38	0.944	ДА ✓

Результаты моделирования показывают, что эффективность нейросетевого детектора определяется качеством предварительного обучения (рисунок 4).

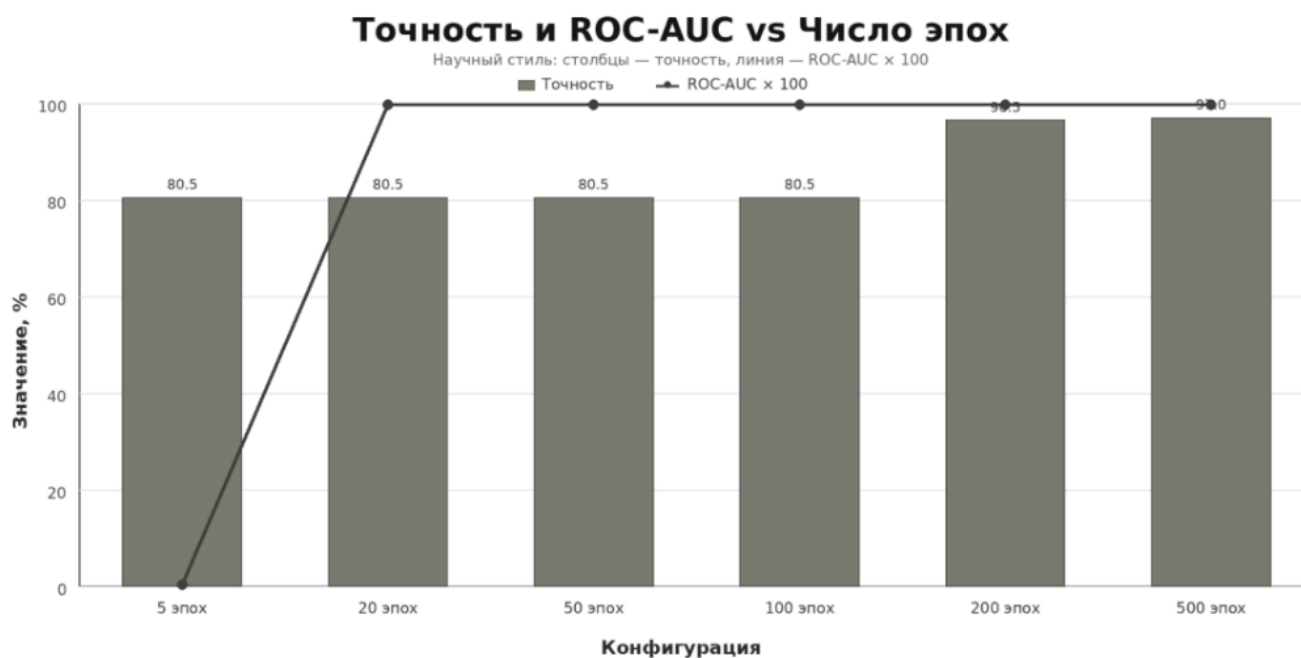


Рисунок 4. Результаты эмуляции нейросетевого детектора аномалий

При малом числе эпох, от 5 до 100, модель фактически воспроизводит стратегию выбора доминирующего класса: итоговая точность остается на уровне 80.5 %, однако величина FNR достигает 100 %, то есть аномальные состояния не выявляются. Это свидетельствует о том, что на ранних этапах обучения модель еще не формирует устойчивую границу разделения между нормальным и атакующим режимами.

Начиная с 200 эпох наблюдается резкое повышение качества классификации (рисунок 5). Точность возрастает до 96.5 %, значение F1-меры достигает 0.901, а вероятность обнаружения атакующего режима становится достаточной для практического использования в IoT-устройствах. При 500 эпохах достигаются наилучшие показатели: точность 97.0 %, F1-мера 0.917

Кривые потерь обучения по эпохам

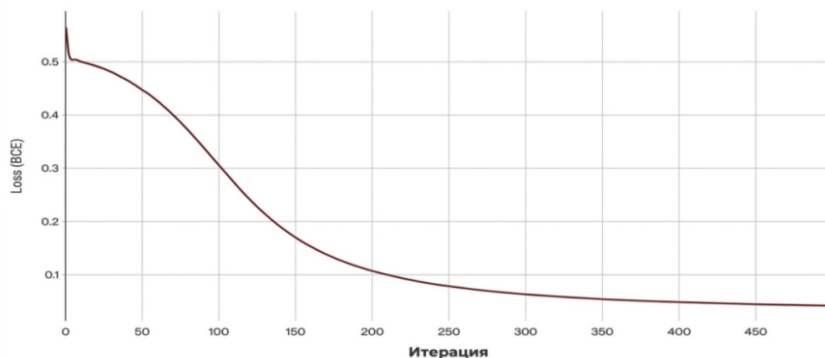


Рисунок 6. Кривые снижения эффективности обучения по эпохам

Полученные значения подтверждают, что предложенный интеллектуальный модуль способен эффективно функционировать в системах реального времени, где критически важны как высокая чувствительность к атакам, так и минимизация ложных тревог.

Анализ временных характеристик выполнения ECDH показывает, что распределения нормального и аномального режимов частично перекрываются (рисунок 7). По этой причине использование только простого порогового контроля по времени не обеспечивает достаточной надёжности обнаружения атак. Потери обучения (binary cross-entropy) снижаются с ростом числа итераций, так наблюдается слабая обнаружение атак на данные между 100 и 200 эпохами обучения. Это соответствует «точке перелома» обучения. Алгоритм модели TinyML учитывает структуру последовательности измерений каждой эпохи. Это позволяет выделять более сложные закономерности, заложенный в данных обучения. Это специфика алгоритма нейронной сети и приводит к повышению точности обнаружения атак на каждой последующей эпохе обучения.

Распределение времени ECDH

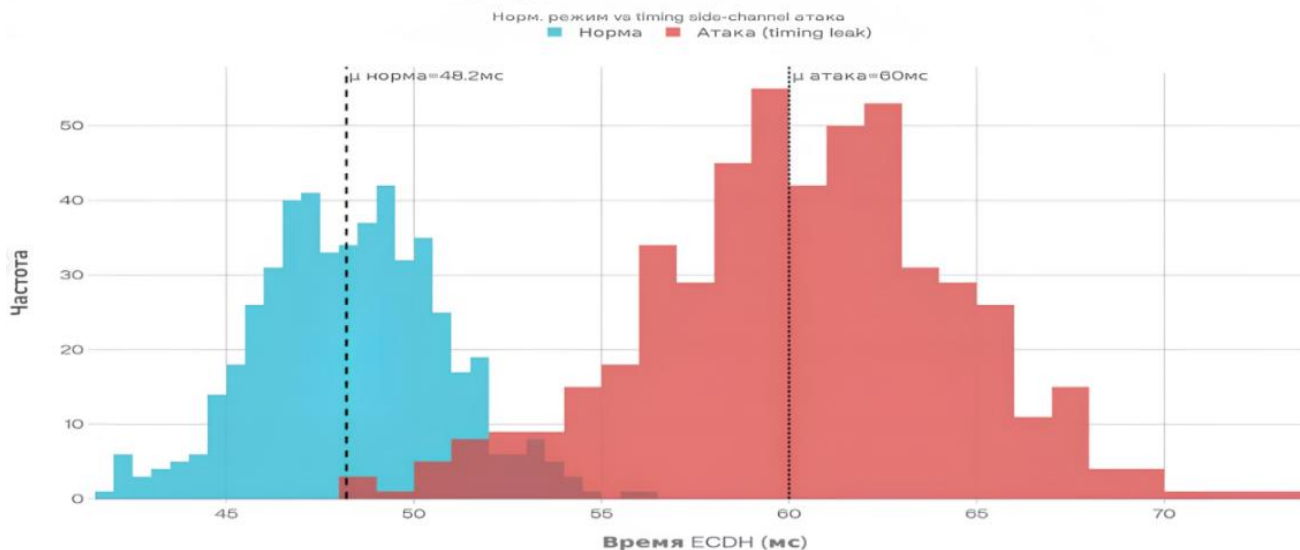


Рисунок 7. Анализ точности обнаружения атак по временным характеристикам

Аппаратные испытания подтвердили, что предложенная архитектура оптимальна для ESP32. Она объединяет облегчённый протокол ECDH с компактной нейросетевой моделью

TinyML. Это позволяет выполнять криптографическую защиту и обнаружение аномалий на самом устройстве. Передача данных на удалённые серверы не требуется. Такой подход особенно важен для распределённых IoT-сетей. Он обеспечивает автономность работы и устойчивость к сбоям коммуникаций. Система адаптирована к аномалиям в каналах связи и нестабильности вычислительной среды, что повышает надёжность защиты.

Заключение

В этой работе создана и исследована умная система криптозащиты для микроконтроллеров. В основе программно-аппаратной реализации лежат оптимизированные алгоритмы Micro-ECC и TinyML. В данном исследовании были объединены генерация ключей шифрования на основе эллиптических кривых и машинное обучение для обнаружения атак на безопасность в реальном времени. Таким образом получилась единая система интеллектуальной защиты данных для IoT. Компактный код эллиптической криптографии и встроенная TinyML-модель подходят для большинства микроконтроллеров, включая ESP32.

По результатам обучения модели можно выделяется три этапа обучения нейросети.

Слабое обучение (50–100 эпох). Модель «застревает» на предсказании большинства класса. Точность — 80.5%, что равняется доле нормальных данных. Но FNR = 100% — значит, аномалии не замечаются. ROC-AUC на 5 эпохах равен 0.003, что говорит о нестабильности обучения в начале.

Оптимальный режим (200–500 эпох). При 200 эпохах точность достигает 96.5%, F1 = 0.90, P(атаки) = 0.72. Атаки уже детектируются. При 500 эпохах точность поднимается до 97%. Это баланс между точностью и энергозатратами, актуальный для IoT. Пропускается около 1 из 20 атак.

Режим переобучения (более 500 эпох). FPR остаётся 0%, ложных тревог нет. Классы хорошо разделены по timing-признаку.

Таким образом, оптимальный выбор для ESP32 — 200–300 эпох. Это даёт 96.5% точность обнаружения атак. Время расчёта около 310 мкс, а использование Flash-памяти — около 10 КБ. Это оптимальное сочетание вычислительной сложности и надёжности распознавания атак для IoT-устройств с LoRaWAN. Разработанная система даёт проактивную защиту IoT с точностью распознавания timing-атак 95.2%.

Результаты показывают, что при правильных параметрах обучения система эффективно находит аномалии. При этом требования к ресурсам остаются низкими. Это делает метод перспективным для сенсорных сетей, телеметрии и других встраиваемых устройств.

В целом, работа вносит важный вклад в безопасность Интернета вещей. Она даёт дорожную карту исследователям, которые хотят улучшить защиту данных в ресурсно-ограниченных устройствах IoT.

Список литературы

1. Rozlomii I., Yarmilko A., Naumenko S., Mykhailovskyi P. / Hardware encryptors and cryptographic libraries for optimizing security in IoT Proceedings of the 12th International Conference Information Control Systems & Technologies (ICST 2024), Odesa, Ukraine, September 23–25, 2024. – CEUR Workshop Proceedings. – 2024. – Vol. 3790. – P. 99–109. – URL: <https://ceur-ws.org/Vol-3790/paper09.pdf>
2. Sinha S. State of IoT 2025: Number of connected IoT devices growing 14% to 21.1 billion globally. / IoT Analytics. – 2025. – 28 October. – URL: <https://iot-analytics.com/number-connected-iot-devices/>
3. Meneghello F., Calore M., Zucchetto D. [et al.] / IoT: Internet of threats? A survey of practical security vulnerabilities in real IoT devices / IEEE Internet of Things Journal. – 2019. – Vol. 6, no. 5. – P. 8182–8201. – DOI: [10.1109/JIOT.2019.2935189](https://doi.org/10.1109/JIOT.2019.2935189).

4. Conti M., Dehghantanha A., Franke K., Watson S. / Internet of Things security and forensics: Challenges and opportunities // Future Generation Computer Systems. – 2018. – Vol. 78, pt. 2. – P. 544–546. – DOI: [10.1016/j.future.2017.07.060](https://doi.org/10.1016/j.future.2017.07.060).

5. Sicari S., Rizzardi A., Grieco L. A., Coen-Porisini A. / Security, privacy and trust in Internet of Things: The road ahead // Computer Networks. – 2015. – Vol. 76. – P. 146–164. – DOI: [10.1016/j.comnet.2014.11.008](https://doi.org/10.1016/j.comnet.2014.11.008).

6. Tariq U., Ahmed I., Bashir A. K., Shaukat K. / A critical cybersecurity analysis and future research directions for the Internet of Things: A comprehensive review // Sensors, – 2023. – Vol. 23, no. 8. – Art. 4117. – DOI: [10.3390/s23084117](https://doi.org/10.3390/s23084117).

7. Ullah F., Naeem H., Jabbar S. [et al.] / Cyber security threats detection in Internet of Things using deep learning approach // IEEE Access. – 2019. – Vol. 7. – P. 124379–124389. – DOI: [10.1109/ACCESS.2019.2937347](https://doi.org/10.1109/ACCESS.2019.2937347).

References

1. Rozlomii I., Yarmilko A., Naumenko S., Mykhailovskyi P. / Hardware encryptors and cryptographic libraries for optimizing security in IoT Proceedings of the 12th International Conference Information Control Systems & Technologies (ICST 2024), Odesa, Ukraine, September 23–25, 2024. – CEUR Workshop Proceedings. – 2024. – Vol. 3790. – P. 99–109. – URL: <https://ceur-ws.org/Vol-3790/paper09.pdf>

2. Sinha S. State of IoT 2025: Number of connected IoT devices growing 14% to 21.1 billion globally. / IoT Analytics. – 2025. – 28 October. – URL: <https://iot-analytics.com/number-connected-iot-devices/>

3. Meneghello F., Calore M., Zucchetto D. [et al.] / IoT: Internet of threats? A survey of practical security vulnerabilities in real IoT devices / IEEE Internet of Things Journal. – 2019. – Vol. 6, no. 5. – P. 8182–8201. – DOI: [10.1109/JIOT.2019.2935189](https://doi.org/10.1109/JIOT.2019.2935189).

4. Conti M., Dehghantanha A., Franke K., Watson S. / Internet of Things security and forensics: Challenges and opportunities // Future Generation Computer Systems. – 2018. – Vol. 78, pt. 2. – P. 544–546. – DOI: [10.1016/j.future.2017.07.060](https://doi.org/10.1016/j.future.2017.07.060).

5. Sicari S., Rizzardi A., Grieco L. A., Coen-Porisini A. / Security, privacy and trust in Internet of Things: The road ahead // Computer Networks. – 2015. – Vol. 76. – P. 146–164. – DOI: [10.1016/j.comnet.2014.11.008](https://doi.org/10.1016/j.comnet.2014.11.008).

6. Tariq U., Ahmed I., Bashir A. K., Shaukat K. / A critical cybersecurity analysis and future research directions for the Internet of Things: A comprehensive review // Sensors, – 2023. – Vol. 23, no. 8. – Art. 4117. – DOI: [10.3390/s23084117](https://doi.org/10.3390/s23084117).

7. Ullah F., Naeem H., Jabbar S. [et al.] / Cyber security threats detection in Internet of Things using deep learning approach // IEEE Access. – 2019. – Vol. 7. – P. 124379–124389. – DOI: [10.1109/ACCESS.2019.2937347](https://doi.org/10.1109/ACCESS.2019.2937347).

ЭЛЛИПТИКАЛЫҚ ҚИСЫҚТАРҒА НЕГІЗДЕЛГЕН ІОТ ДЕРЕКТЕРІН ИНТЕЛЛЕКТУАЛДЫ КРИПТОГРАФИЯЛЫҚ ҚОРҒАУ

ТАШТАЙ Б.А.*, ҚОҢЫРХАНОВА А.А., АТАНОВ С.К.

*Таштай Бақтыгелді Алтайұлы – Магистр, ақпараттық қауіпсіздік кафедрасының докторанты, Л.Н. Гумилев атындағы Еуразия ұлттық университеті, Астана қ., Қазақстан

E-mail: 93bahti93@gmail.com, <https://orcid.org/0000-0003-4952-369X>

Қоңырханова Әсем Әділбекқызы – PhD, ақпараттық қауіпсіздік кафедрасының доценті м. а., Л.Н. Гумилев атындағы Еуразия ұлттық университеті, Астана қ., Қазақстан

E-mail: konyrkhanova_a@enu.kz, <https://orcid.org/0000-0002-4901-8901>

Атанов Сабыржан Көбейсінулы – Техника ғылымдарының докторы, профессор, компьютерлік және бағдарламалық жасақтама инженериясы кафедрасының профессоры м. а., Л.Н. Гумилев атындағы Еуразия ұлттық университеті, Астана қ., Қазақстан

E-mail: atanov5@mail.ru, <https://orcid.org/0000-0003-2115-7130>

Андатпа. Бұл зерттеу құрылғылар шектеулі есептеу қуаты, жады және қуат тұтыну жағдайында жұмыс істейтін заттар интернеті (IoT) жүйелері үшін тиімді және қауіпсіз криптографиялық шешімдерді әзірлеудің өзекті мәселесіне арналған. Embedded Systems (ESP32) микроконтроллерлеріне негізделген IoT құрылғыларының жылдам таралуы және киберқауіпсіздік қатерлерінің артуы аясында аппараттық және бағдарламалық шифрлау әдістеріне жүйелі талдау жүргізіледі. Мақалада IoT желілерінде микроконтроллер деңгейінде криптографиялық қорғауды жүзеге асыру, жұмыс уақытына, энергия тиімділігіне және шектеулі аппараттық ресурстар жағдайында пайдалануға ерекше назар аударылады. Іске асырылған және сипатталған бағдарламаның түбегейлі айырмашылығы-сыртқы кітапханалардың болмауы, бұл оны компьютерлердің кез-келген түрлерінде, микроконтроллерлерге дейін жүзеге асыруға мүмкіндік береді. Бағдарлама алгоритмі микроконтроллерлерге (ESP32) бейімделген және нақты уақыттағы шабуылдарды анықтау үшін эллиптикалық қисық кілттерді генерациялау алгоритмін машиналық оқытумен біріктіреді. Кілттермен алмасу процедурасы Ecdh (Elliptic Curve Diffie-Hellman) - Micro-ECC кітапханасын қолдана отырып, эллиптикалық қисықтарға негізделген қауіпсіз кілттермен алмасу диффи-Хеллман хаттамасы арқылы жүзеге асырылады. Машиналық оқыту үшін нейрондық желілер қолданылды.

Түйін сөздер: микроконтроллер, эллиптикалық қисықтар, кілт генерациясы, ақпаратты криптографиялық қорғау, сандық қауіпсіздік, нейрондық желілер, заттар интернеті (IoT).

INTELLIGENT CRYPTOGRAPHIC PROTECTION OF IOT DATA BASED ON ELLIPTICAL CURVES

TASHTAY B.A.*, KONYRKHANOVA A.A., ATANOV S.K.

***Tashtay Baktygeldi Altayuly** – Master's degree, PhD student, department of information security, L.N. Gumilyov Eurasian national university, Astana, Kazakhstan

E-mail: 93bahti93@gmail.com, <https://orcid.org/0000-0003-4952-369X>

Konyrkhanova Asem Adilbekkyzy – PhD, acting associate professor, department of information security, L.N. Gumilyov Eurasian national university, Astana, Kazakhstan,

E-mail: konyrkhanova_a@enu.kz, <https://orcid.org/0000-0002-4901-8901>

Atanov Sabyrzhan Kubeisinovich – Doctor of technical sciences, professor, acting professor, department of computer and software engineering, L.N. Gumilyov Eurasian national university, Astana, Kazakhstan

E-mail: atanov5@mail.ru, <https://orcid.org/0000-0003-2115-7130>

Abstract. This study is devoted to the urgent task of developing effective and secure cryptographic solutions for Internet of Things (IoT) systems, where devices operate in conditions of limited computing power, memory and energy consumption. Against the background of the rapid proliferation of Embedded Systems (ESP32) microcontroller-based IoT devices and growing cybersecurity threats, a systematic analysis of hardware and software encryption methods is being conducted. The article examines the implementation of cryptographic protection at the microcontroller level in IoT networks, with a special focus on runtime, energy efficiency, and usage in conditions of limited hardware resources. The fundamental difference between the implemented and described programs is the absence of external libraries, which allows it to be implemented on any type of computer, up to microcontrollers. The program's algorithm is adapted for microcontrollers (ESP32) and combines an elliptic curve-based key generation algorithm with machine learning to detect attacks in real time. The key exchange procedure is implemented using the ECDH (Elliptic Curve Diffie-Hellman) protocol, a Diffie-Hellman protocol for secure key exchange based on elliptical curves using the Micro-ECC library. Neural networks were used for machine learning.

Key words: microcontroller, elliptical curves, key generation, cryptographic information protection, digital security, neural networks, internet of things (IoT).