Hardware Cybersecurity on the FPGA Platform during Network Interaction of Distributed Industrial Equipment

Valery A. Kokovin
Department of Automation of Technological Processes
State University "Dubna", branch "Protvino"
Protvino, Moscow reg., Russia
kokovin@uni-protvino.ru

Alexander N. Sytin
Department of Automation of Technological Processes
State University "Dubna", branch "Protvino"
Protvino, Moscow reg., Russia
alexander.sytin@gmail.com

Victor V. Skvortsov
Department of Experimental Physics Institute for High Energy Physics of the National Research Centre 'Kurchatov Institute'
Protvino, Moscow reg., Russia
Victor.Skvortsov@ihep.ru

***Abstract*—Abstract В данной статье рассматриваются методы уменьшения киберугроз при сетевом взаимодействии промышленного оборудования под управлением DCS. Предполагается, что в составе управляющих и сетевых контуров оборудования используются решения на платформе FPGA. Предложены методы защиты сетевых обменов сообщениями путем замены отдельных, специально выделенных сегментов сети, имеющих неконтролируемый доступ, на аппаратно protected network segment (PNS). В составе PNS два блока с FPGA Cyclone 10 LP 10CL025YU256C8G фирмы INTEL. Кроме этого, в каждом блоке есть Transceiver RS-485 для подключения к сегменту сети RS-485 и два DS-link на основе LVDS Transceivers. Предложен алгоритм шифрования в PNS каждого бита передаваемого байта данных RS-485, в отличие от широко распространенных алгоритмов блочного шифрования. При передаче данных внутри PNS использовалось DS – кодирование. Представленное исследование сосредоточено на разработке и анализе эффективной реализации алгоритма шифрования данных в реальном времени. Рассмотрен пример формирования кодовых последовательностей при шифровании данных.**

Keywords—hardware cybersecurity, FPGA Cyclone 10, RS-485, encryption algorithm

I. INTRODUCTION

При разработке распределенных управляющих систем технологическими процессами промышленных или научных комплексов выполняется построение модели, основанной на формализации входных параметров и параметров текущего состояния этих процессов. Кроме того, модель должна учитывать коммуникационное взаимодействие технологических подсистем. Как правило, в понятие распределенной управляющей системой (distributed control system - DCS) вкладывается не только географическое распределение отдельных подсистем и промышленного оборудования, но и распределение алгоритма управления, направленного на решение общей технологической задачи. При разработке многих DCS необходимо учитывать управление быстрыми технологическими процессами, которые требуют реакции исполнительных устройств в реальном времени. Эти требования задают жесткие ограничения на время реакции программно-аппаратных приложений DCS.

Возросший уровень интеллектуализации управляющих и исполнительных устройств позволяет расширить функциональность технологических систем, повысить их производительность и эффективность прогнозирования сбоев оборудования. Еще больший выигрыш достигается при использовании интеллектуального оборудования в распределенных управляющих системах с применением сетевых технологий. Появление сетевых экосистем, таких как Industrial IoT (IIoT), Internet of Robotic Things (IoRT) [1] дало мощный импульс для использования открытых информационных и коммуникационных технологий. Но это же послужило причиной возросших информационных атак на IoT/IIoT, IoRT - приборы, инфраструктурные решения которых, основаны на технологиях и устройствах сетевых экосистем [2]. Оборудование, в состав которого входят IoRT, может непосредственно принимать решения и воздействовать на окружающую физическую среду. В этом случае опасность кибератак и последствия их с точки зрения критических уязвимостей технологических процессов гораздо выше.

Исторически, сетевые взаимодействия в Industrial Control Systems (ICS) были организованы с использованием последовательных сетей полевого уровня RS-485 and RS-422. Протоколы на основе этих интерфейсов, такие как Modbus, Profibus и другие, были разработаны без учета безопасности, поскольку использовались для развертывания в средах с ограниченным доступом.

При организации киберзащиты систем контроля физического доступа (Physical Access Control System, PACS) к технологическим системам основное внимание уделяется проблеме предоставления или блокировке доступа. В настоящее время большинство PACS имеют в

качестве коммуникационного интерфейса RS-485. Это связано в первую очередь с небольшими затратами на реализацию PACS. При этом устройства системы доступа могут подключаться последовательно и на большом расстоянии от основного сервера PACS. Это дает возможность без больших затрат добавлять дополнительные точки доступа к оборудованию. Во-вторых, переход к современным PACS на основе протокола TCP/IP требует значительных вложений и квалифицированный обслуживающий персонал.

Под аппаратной безопасностью передаваемых сообщений будем понимать набор средств, которые защищают конфиденциальность этих сообщений, know-how разработчиков и физическую безопасность оборудования. Кибератаки на разрушение промышленного оборудования чаще всего применяются на вычислительных устройствах и на коммуникационных сетях при организации межмашинных взаимодействий или Интернета вещей (IoT)

В данной статье рассматриваются методы уменьшения киберугроз при сетевом взаимодействии промышленного оборудования под управлением DCS. Предполагается, что в составе управляющих и сетевых контуров оборудования используются решения на платформе FPGA. Предложены методы защиты сетевых обменов сообщениями в устройствах, имеющих в своем составе интерфейсы TIA-485 / EIA-485.

Статья организована следующим образом: в Секции II проанализированы работы, связанные с проблемами повышения кибербезопасности промышленного оборудования на платформе FPGA. Алгоритмы шифрования данных и способы их эффективной реализации в сетевых решениях в RS-485, на основе FPGA, рассмотрены в Секции III. Предложенный алгоритм шифрования в PNS каждого бита передаваемого байта данных RS-485, в отличие от широко распространенных алгоритмов блочного шифрования. представлены в Секции IV. В Секции V обсуждаются результаты исследования и даны выводы.

II. Related Works

Проблеме повышения кибербезопасности промышленного оборудования уделяется повышенное внимание в научных статьях и в разрабатываемых международных стандартах.

Важным показателем организации безопасной работы промышленных систем является правильная организация контроля доступа к распределенным системам и оборудованию этих систем. Стандарт IEC 60839-11-5, принятый Международной электротехнической комиссией в 2020 году, specifies the Open Supervised Device Protocol (OSDP) for electronic access control systems [3]. Стандарт направлен в первую очередь на улучшения взаимодействия между устройствами контроля доступа и безопасности. Он использует the AES-128 algorithm [4], поддерживает многоточечную установку и контролирует соединения, сообщая о проблемах считывания. Кроме того, исследование аппаратной реализации AES-128 на платформе FPGA показало более быстрое и конфигурируемое решение [5].

Широкое использование FPGA для реализации вычислительных и коммуникационных решений позволяет решать многие задачи, когда требуется большое быстродействие и параллельная многопоточная обработка данных. Это дает возможность реализовать на FPGA не только контуры управления быстрыми процессами, но и создавать аппаратные системы защиты от внешних киберугроз. В статье [6] авторами предлагается метод для предотвращения атак по побочным каналам, основанных на синхронизации. Как правило, атаки по побочным каналам направлены на взлом криптосистемы с целью завладения криптографическими ключами.

Гораздо сложнее организовать защиту от кибератак для устройств IIoT. Такая защита имеет свои особенности из-за небольших вычислительных программных и аппаратных ресурсов. Необходимые требования по безопасности функционирования IoT/IIoT, которые включают: однозначную идентификацию устройства (например, RFID), авторизацию доступа к ресурсам и конфиденциальность доступа к определенной информации (например, к файлам конфигурации при использовании FPGA), регулярное обновление прошивки безопасности сформулированы в работе [7]. Учитывая ограниченность ресурсов, авторы предлагают использовать технологию роя (swarm technology), когда простые устройства объединяются в кластеры, увеличивая свои интеллектуальные возможности для отражения киберугроз.

В работе [8] предложено метод аутентификации на основе измерении с большой точностью времени передачи сообщения со стороны отправителя и времени приема на стороне получателя. Для измерения временных интервалов в драйверы приема и передачи добавлены подсистемы измерения времени на основе time-to-digital converter (TDC) с разрешением не хуже 55 ps. Измеренное время (идентификатор) позволяет сравнивать время передачи на одном участнике обмена сообщениями и время приема (на другом), что дает возможность идентифицировать передачу как достоверную. Это дает гарантию целостности, но не обеспечивает правильную идентификацию отправителя. Для усиления аутентификации в передаваемое сообщение дабавлен Crypto Identifier (CI) передающего модуля. Он основан на предположение о не идеальности повторяемости микросхем FPGA при их производстве, даже с одинаковой заводской маркировкой. Когда мы пропускаем сигнал по одинаковому пути разных микросхем, то наблюдаем девиацию временного интервала, что и является основой идентификатора CI, который является уникальным для данной FPGA.

Проблема аутентификации отправителя очень остро стоит при обмене сообщениями между распределенным по большой территории оборудованием. В этом случае появляются участки коммуникационной сети, физически доступные для злоумышленников. Возникает киберугроза, названная man-in-the-middle (MiTM) attack. Атака

направлена на нарушение конфиденциальности и целостности данных отправителя в сети, вплоть до замены на вредоносные данные. Это может привести к выходу оборудования из строя и спровоцировать аварию.

III. Formulation of the Problem

Проблемы безопасности и возможные решения защиты сетей полевого уровня проанализированы в работе [9]. Предлагается анализировать последовательные пакеты полевого уровня с помощью пассивного анализатора, который устанавливается между полевыми устройствами и контроллерами и обеспечивает захват данных в режиме реального времени (Fig.1). Определены требования к анализатору, которые включают полную изоляцию своих сигналов от сети управления. Анализатор также защищает

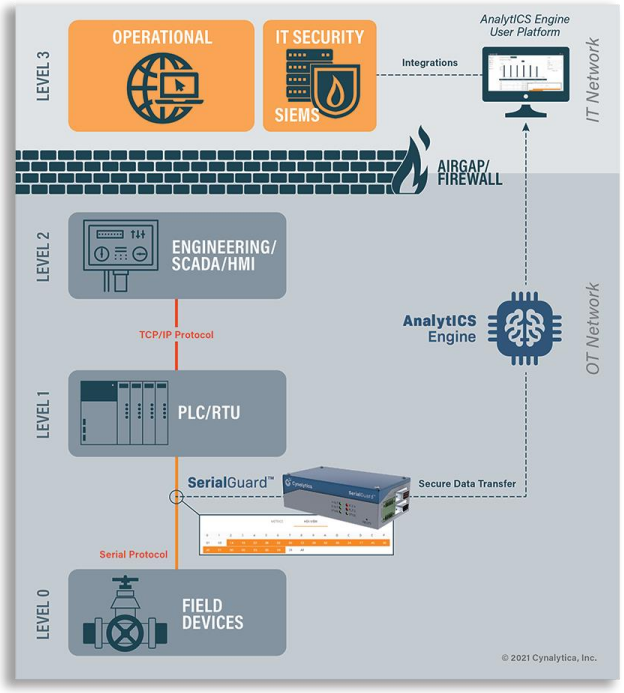

Fig. 1. Мониторинг последовательных пакетов сети полевого уровня анализатором [9].

данные, инкапсулируя перехваченные сообщения в зашифрованные пакеты данных TCP перед их пересылкой в систему проверки поведения последовательных устройств. Такая конфигурация системы контроля безопасности может служить защитой от распространенных сценариев атак, таких как атаки MiTM, несанкционированные команды и внутренние угрозы.

Защита последовательных сетей полевого уровня в DCS является более сложной задачей, поскольку нет единого центра управления, и решения принимаются в каждой распределенной подсистеме. В работе [10] представлена система мониторинга и регистрации timing messages (TM), с помощью которых выполняется синхронизация работы систем ускорительно-накопительного комплекса. TM передаются по последовательной сети между Timing Message Generators (TMG), которые выполняют преобразование событий от физических датчиков в TM. Для лучшего контроля и проверки достоверности передаваемых сообщений TMG объединены кольцевой магистралью. Входными и выходными потоками в TMG управляет Timing Network Controller (TNC), реализованный на платформе FPGA. Выбранная платформа позволяет выполнять диагностику и мониторинг TM. Все сообщения проверяются встроенным в FPGA электронным сканером в реальном времени на соответствие принятого кода сообщения правильной очередности его поступления [11]. При обнаружении несоответствия формируется сигнал alarm для оператора. Такая проверка позволяет увеличить безопасность функционирования устройств TMG, но не дает достаточной защиты от злоумышленников.

В работе [12] предлагается шифровать пакеты данных, передаваемые по RS-485 с помощью широко используемого протокола ModBUS RTU. Для шифрования использован алгоритм BlowFish. Алгоритм Blowfish обеспечивает высокий уровень безопасности и конфиденциальности данных. Он использует симметричное шифрование, то есть один и тот же ключ для шифрования и расшифровки данных. Кроме этого, данный алгоритм достаточно просто реализуется на платформе FPGA [13]. Для усиления шифрования предлагается использовать генератор шума. Основным недостатком алгоритма BlowFish является отсутствие встроенной аутентификации.

IV. Hardware-based Cybersecurity on the FPGA Platform for Dedicated Serial Network Segments

Как упоминалось ранее, протоколы на основе интерфейсов RS-485 and RS-422 были разработаны без учета защиты от внешних киберугроз. Проанализированные статьи по данной тематике показывают необходимость внешней программной или аппаратной надстройки для обеспечения кибербезопасности передаваемых данных. Отсутствие MAK – адреса в устройствах, взаимодействующих по RS-485, является большим недостатком с точки зрения контроля, фильтрации и идентификации участников обмена. Современные системы безопасности, как правило, базируются на протоколах TCP/IP, но в промышленной автоматизации на полевом уровне продолжают использоваться протоколы обмена данными на RS-485 and RS-422.

Представленная в данной статье идея защиты данных, передаваемых по RS-485, основана на замене отдельных, специально выделенных сегментов сети, имеющих неконтролируемый доступ (открытый для злоумышленника), на сегменты, которые защищены аппаратными средствами. Рассмотрим структурную организацию защищенного сегмента сети (protected network segment (PNS) и аппаратные ресурсы, необходимые для решения поставленной задачи.

A. *Стандарты передачи данных используемые для реализации PNS*

Основным стандартом, определяющим идеологию PNS, является IEEE Standard 1355 [14]. Стандарт представляет собой коммуникационный последовательный интерфейс, предназначенный для соединения точка-точка, использующий среду передачи данных витую пару или оптику. Выбор кабеля типа витая пара или оптики определяется необходимой скоростью передачи информации. Способ передачи электрических сигналов в PNS определяется стандартом ANSI/TIA/EIA-644-A 2001 [15]. Этот стандарт определяет энергоэффективный и высокоскоростной способ передачи сигналов с помощью low-voltage differential signaling (LVDS). На сигнальном уровне используется DS-кодирование, которое требует две сигнальных линии: D – линия для передачи данных и S – линия для передачи строба. DS-кодирование обладает свойством самосинхронизации: в комбинации DS-сигналов закодирован синхросигнал. Этот сигнал может восстанавливаться на приемной стороне логической операцией XOR над сигналами D и S. Самосинхронизация позволяет произвольно менять скорость передачи, даже в пределах одного сообщения. Это дает хорошее преимущество в безопасности. В интерфейсах RS-485 and RS-422 скорость фиксированная и выбирается из предложенного ряда.

B. *Функциональная схема для PNS и аппаратные ресурсы для реализации сегмента*

Использование в составе PNS элементов на платформе FPGA дает широкие функциональные возможности: использование сторонних IP-ядер с адаптацией к решаемой задаче, изменение алгоритма "на лету", меняя конфигурацию FPGA, и подстраиваясь к изменяющимся условиям работы системы. При проверке работоспособности PNS использовалась FPGA семейства Cyclone 10 LP 10CL025YU256C8G фирмы INTEL [16].

На Fig.2 представлена функциональная схема PNS, состоящая из двух идентичных блоков. В каждом блоке содержатся следующие элементы:

- Вычислительное ядро, реализованное на FPGA с каждой стороны сегмента в составе модуля TEI0003-03 [17]. Powerful FPGA module integrating an Intel Cyclone 10 LP FPGA, 8 MByte SDRAM, 8 MByte Flash, USB.
- Transceiver RS-485 для подключения к сегменту сети RS-485.
- DS-link - входные и выходные порты последовательного асинхронного интерфейса локальной сети [14]. DS-link на каждом блоке позволяет реализовать обмен в режиме Full – Duplex. Драйверы входов и выходов DS-link объединяются стандартным кабелем UTP на 4 пары.
- Питание Transceivers и модулей TEI0003-03 выполняется либо через USB порт от планшетного компьютера, либо от внешнего источника питания.
- Планшетный компьютер может использоваться для загрузки новой конфигурации по Wi-Fi со сторонних ресурсов.

Два DS-link используются для возможности замены сегмента сети на основе интерфейса RS-422. В этом случае необходимо добавить с каждой стороны PNS соответствующие драйверы RS-422. Загрузка конфигурации FPGA может обеспечиваться через USB порт от планшетного компьютера.

Предложенная структура PNS может быть создана в различных конфигурациях:

- Если нет необходимости менять конфигурацию FPGA "на лету", то от планшетного компьютера

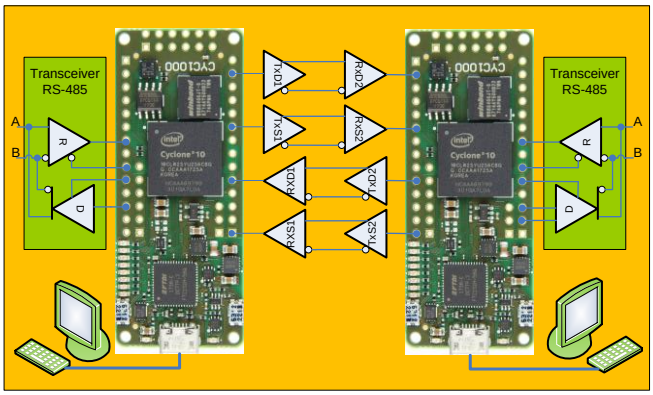

Fig. 2. Функциональная схема и ресурсы для реализации PNS.

- можно отказаться и загружать от внутренней конфигурационной памяти модуля TEI0003-03
- Можно заменить планшетный компьютер на миниатюрную плату типа Raspberry Pi 3 Model A+ [18] в промышленном исполнении. Наличие на плате беспроводной сети и встроенного интерфейса видеокамеры позволяет контролировать не только сетевые сообщения, но и фиксировать с помощью камеры возможный несанкционированный доступ к PNS в реальном времени.

V. PNS Operating Algorithm

Для предложенной архитектуры PNS и используемых в нем ресурсов можно использовать разнообразные алгоритмы шифрования. Например, широко используемый алгоритм AES-128 симметричного шифрования с минимальным размером блока входных данных 128 бит. В работе [19] представлена аппаратная реализация этого алгоритма шифрования на FPGA. Авторы представили высокопроизводительное и эффективное по используемым ресурсам FPGA решение. В работе [12] показано использование алгоритма BlowFish, который просто реализуется на платформе FPGA.

В основе алгоритма работы PNS, предложенного в данной статье, лежат следующие идеи:

- Рассмотренные алгоритмы AES-128, BlowFish и похожие на них являются блочными алгоритмами шифрования (128 и 64 бита соответственно). Предложенный алгоритм шифрует каждый бит передаваемого байта данных и бит управления. Это возможно только в том случае, если скорость передачи по PNS, во много раз превышает скорость передаваемых данных по RS-485. Стандарт TIA-485/EIA-485 определяет максимальную скорость передачи данных 10 Mbit/s на дистанции не более 12 м. На максимальной дистанции, определенной стандартом, скорость не должна превышать 100 kbit/s. Аппаратные ресурсы PNS (FPGA, DS-link на технологии LVDS) более чем в 10 раз превышают по Switching Characteristics указанные параметры RS-485.

- Заполнение временного интервала битовых переменных передаваемого байта данных большим количеством импульсов. В этом случае истинное логическое значение этой переменной (“0” or “1”) будет закодировано.

- Перед началом шифрования в каждом блоке PNS создается (или загружается) таблица Table_of_Key с 256 строчками, в которых размещаются 8 - разрядные указатели KeyByte и секретные ключи CipherKey. Условие: значение CipherKey $\neq 0$!

- В пределах временного интервала управляющего бита START, передаваемого байта данных, размещается 8 - разрядный ключ, названный KeyByte. KeyByte для каждого байта выбирается случайным образом из согласованной таблицы для приемника и передатчика сегмента PNS.

- При использовании KeyByte для шифрования каждого бита формируется последовательность кодовых комбинаций (Code1, Code2, ... CodeM, где M = (1, 2, … , 9) для кодирования управляющих (P) и данных (D0 – D7) бит. Максимальное количество импульсов (разрядов кодовой комбинации) во временном интервале битовой переменной составляет либо 8, либо больше и определяется ограничениями на скорость передачи RS-485 и длиной сегмента PNS.

При описании алгоритма предполагается, что PNS работает в следующей конфигурации: Half Duplex режим, 8 бит данных и один STOP. Алгоритм кодирования передаваемого байта состоит из нескольких параллельных процессов. Первый процесс обеспечивает подготовку набора из переменных KeyByte, Code1 - Code9 для кодирования текущего байта данных и биты P (без STOP). В этом процессе Code1 - Code9 вычисляются по определенным правилам. KeyByte выбирается из Table_of_Key (TABLE1).

TABLE I. TABLE_OF_KEY

Line addres, Hex	Data coding			
	KeyByte, Hex	***CipherKey, Hex***	***Code (M), Hex***	***N, Hex***
20	4F	3A		
21		28	Code1 = 62	N = 3
22		57	Code2 = 91	N = 5
23		7C	Code3 = B6	N = 5
24		6F	Code4 = A9	N = 6
25		93	Code5 = CD	N = 4
26		A3	Code6 = DD	N = 4
27		45	Code7 = 7F	N = 3
28		A1	Code8 = DB	N = 3
29		33	Code9 = 6D	N = 4

Рассмотрим подробнее правила вычисления переменных в первом процессе (для передающего блока PNS).

- На основе Table_of_Key, по выбранному KeyByte определяется первый CipherKey.

- Выполняется арифметическое суммирование двух переменных: CipherKey и переменной из следующей строчки (из столбца CipherKey). Это и есть переменная Code1.

- Далее операция суммирования повторяется (для выбора Code2), но суммируются уже CipherKey и переменная на две строчки ниже. Процесс вычисления переменных Code повторяется еще 7 раз. Таким образом, вычисляются все Code1 – Code9. В интервале биты STOP записывается побитная инверсия переменной KeyByte.

Второй процесс начинает работу, когда на вход первого блока PNS начинают поступать данные из Transceiver RS-485. В исходном состоянии все Transceivers RS-485 находятся в режиме приема. По мере поступления бит данных из Transceiver выполняется загрузка в сдвиговый регистр закодированных данных, полученных в первом процессе. Далее выполняется сдвиг всех 88 бит в DS-link для передачи во второй блок PNS.

Принимающий второй блок PNS, после приема по DS-link переменной KeyByte, переводит Transceivers RS-485 в режим передачи данных и формирует сигнал START. Далее начинается Decoding переменных Code1 – Code9.

- При поступлении переменной Code1, выполняется вычитание из Code1 переменной CipherKey. В полученном результате подсчитывается число “1” . Получаем переменную N (N = 1, 2, …, 8). Сдвигаем вправо результат вычитания на N тактов. На выходе сдвигового регистра получаем значение (“0” or “1”) для записи в бит D0.

- Повторяем операцию еще 8 раз. В бит STOP всегда записываем значение “1”.

В таблице TABLE1 представлены данные для кодировки байта данных *DATA* = [*0 1 1 1 0 1 0 0*] и бит четности *P = 0*. В первом блоке PNS определены *KeyByte* = [*4F*] и *CipherKey* = [*3A*]. Вычислены кодовые комбинации *Code1* (*D0*) = [*62*], *Code2* (*D1*) = [*91*], *Code3* (*D2*) = [*B6*], *Code4* (*D3*) = [*A9*], *Code5* (*D4*) = [*CD*], *Code6* (*D5*) = [*DD*], *Code7* (*D6*) = [*7F*], *Code8* (*D7*) = [*DB*], *Code9* (*P*) = [*6D*].

После приема кодовых последовательностей, во втором блоке были вычислены недостающие CipherKey для каждого бита, кроме START и STOP: CipherKey = [28] (D0 = 0), CipherKey = [57] (D1 = 1), CipherKey = [7C] (D2 = 1), CipherKey = [6F] (D3 = 1), CipherKey = [93] (D4 = 1), CipherKey = [A3] (D5 = 1), CipherKey = [45] (D6 = 1), CipherKey (D7)= [A1], CipherKey = [33] (P = 0). Далее подсчитывается число “1” N для каждого значения CipherKey и сдвигается на определенное число N, получая восстановленное значение бита.

VI. Discussion and Conclusions

Представленное исследование сфокусировано на разработке универсального аппаратно защищенного сегмента PNS сети в реальном времени. Универсальность сегмента заключается в возможности встраивать PNS в сеть на основе интерфейсов RS-485 или RS-422. Кроме того работа алгоритма и всего сегмента не зависит от протокола передачи данных. Предложенный алгоритм шифрования в PNS каждого бита передаваемого байта данных RS-485 позволяет получить большую защиту, чем при блочном шифровании. Это обеспечивается возможностью динамического изменениями таблиц с CipherKey и фактическим шифрованием каждого бита своим секретным ключем. Возможность использования в FPGA параллельных процессов шифрования, передачи и приема данных на больших частотах позволяет работать PNS в реальном времени.

В развитие предложенного исследования планируется добавить в блоки PNS подсистемы измерения времени с большим разрешением (десятки ps). Для этих целей можно добавить time-to-digital converter (TDC), который будет измерять время передачи (на передающем блоке) и время приема данных (на приемном блоке). В качестве TDC можно использовать TDC7200, разработанный Texas Instruments [20]. Сравнение двух временных интервалов позволит идентифицировать достоверность передаваемых данных. Разработка проектов для FPGA выполнялась в пакете Quartus Prime 18.0 Lite Edition [21]

References

[1] Vermesan O, Bahr R, Ottella M, Serrano M, Karlsen T, Wahlstrøm T, Sand HE, Ashwathnarayan M and Gamba MT (2020), “Internet of Robotic Things Intelligent Connectivity and Platforms”, *Front. Robot. AI* 7:104. doi: 10.3389/frobt.2020.00104

[2] U. Tariq, I. Ahmed, A. K. Bashir and K. Shaukat, "A critical cybersecurity analysis and future research directions for the Internet of Things: A comprehensive review", *Sensors*, vol. 23, no. 8, pp. 4117, Apr. 2023.

[3] IEC 60839-11-5:2020 Alarm and electronic security systems - Part 11-5: Electronic access control systems - Open supervised device protocol (OSDP). (Online) Available https://standards.iteh.ai/catalog/standards/sist/f781536c-0460-401f-8c0d-777ca648e37b/iec-60839-11-5-2020

[4] Daemen J. and Rijmen V. “The Design of Rijndael. AES — the Advanced Encryption Standard”. Springer, 2002.

[5] A. M. Borkar, R. V. Kshirsagar and M. V. Vyawahare, "FPGA implementation of AES algorithm," *2011 3rd International Conference on Electronics Computer Technology*, Kanyakumari, India, 2011, pp. 401-405, doi: 10.1109/ICECTECH.2011.5941780.

[6] S. Mukherjee, S. k. Saikia, S. Anand, R. Chouhan and H. Das, "A Counter Measure to Prevent Timing-based Side-Channel Attack on FPGA," *2021 6th International Conference on Communication and Electronics Systems (ICCES)*, Coimbatre, India, 2021, pp. 983-988, doi: 10.1109/ICCES51350.2021.9489054.

[7] O. Vermesan et al "The Next Generation Internet of Things – Hyperconnectivity and Embedded Intelligence at the Edge," in *Next Generation Internet of Things Distributed Intelligence at the Edge and Human Machine-to-Machine Cooperation* , River Publishers, 2018, pp.19-102.

[8] V. A. Kokovin, A. N. Sytin and V. V. Skvortsov, "Methods for Increasing the Cybersecurity of FNC Devices on the FPGA-Based Platform in Network Communications," *2022 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)*, Sochi, Russian Federation, 2022, pp. 825-830, doi: 10.1109/ICIEAM54945.2022.9787121.

[9] The Security Gap that ICS Cybersecurity Companies Refuse to Talk About. (Online) Available https://cynalytica.com/the-security-gap-that-ics-cybersecurity-companies-refuse-to-talk-about/

[10] V.Komarov et al. “Modernization of U-70 general timing system”, Proceedings of ICALEPCS-2005, Geneva, Switzerland, October 10-14, 2005

[11] V. Kokovin, S. Uvaysov, “Diagnostic port for scanning the selected objects in the electronic means on FPGA”, Kontrol'. Diagnostika, Izdat. dom "Spektr". 2015. № 12. pp. 54 – 59 DOI: 10.14489/td.2015.12.pp.054-059

[12] V.V. Kochetkov and B.B. Zobnin, “Problems and prospects of secure data transmission in the automated process control system using blowfish and ‘noise,” International Journal of Humanities and Natural Sciences, vol. 12-1 (51), 2020, pp. 141-147

[13] K. N. Prasetyo, Y. Purwanto and D. Darlis, "An implementation of data encryption for Internet of Things using blowfish algorithm on FPGA," *2014 2nd International Conference on Information and Communication Technology (ICoICT)*, Bandung, Indonesia, 2014, pp. 75-79, doi: 10.1109/ICoICT.2014.6914043.

[14] IEEE Computer Society, “IEEE Standard for Heterogeneous Interconnect (HIC) (Low-Cost, Low-Latency Scalable Serial Interconnect for Parallel System Construction)”, IEEE Standard 1355 - 1995, IEEE, June 1996

[15] ANSI/TIA/EIA-644-1995. Electrical Characteristics of Low Voltage Differential Signaling (LVDS) Interface Circuits, 2001. (Online) Available https://standards.globalspec.com/std/1618348/TIA-644

[16] Intel® Cyclone® 10 10CL025 FPGA, Datasheet. [Online]. Available: https://ark.intel.com/content/www/us/en/ark/products/210436/intel-cyclone-10-10cl025-fpga.html.

[17] TEI0003-03-QFCR4A Trenz Electronic FPGA module with Intel Cyclone 10 LP FPGA. (Online) Available https://wiki.trenz-electronic.de/display/PD/TEI0003+Resources.

[18] Raspberry Pi 3 Model A+, 1.4GHz 64-bit quad-core processor, dual-band wireless LAN. (Online) Available https://www.raspberrypi.com/products/raspberry-pi-3-model-a-plus/

[19] A. Barrera, C. -W. Cheng and S. Kumar, "Improved Mix Column Computation of Cryptographic AES," *2019 2nd International Conference on Data Intelligence and Security (ICDIS)*, South Padre Island, TX, USA, 2019, pp. 229-232, doi: 10.1109/ICDIS.2019.00042.

[20] Texas Instruments: SNAS647D: TDC7200 time-to-digital converter for time-of-flight applications in lidar, magnetostrictive and flow meter. [Online]. Available: https://www.ti.com/lit/ds/symlink/tdc7200.pdf

[21] Quartus Prime 18.0 Lite Edition

International