УДК 621.391.81
DOI 10.33286/2075-8693-2021-49-21-30.

© *Е. А. Степанова*[1], *П. А. Ашаева*[2]
[1]Омский научный центр СО РАН (Институт радиофизики и физической электроники), Омск, Российская Федерация
[2]Омский научно-исследовательский институт приборостроения, Омск, Российская Федерация

ИСПОЛЬЗОВАНИЕ СРЕДСТВ ПРОГРАММНОГО И ПРОГРАММНО-АППАРАТНОГО МОДЕЛИРОВАНИЯ ДЛЯ ИССЛЕДОВАНИЯ УЯЗВИМОСТЕЙ LORAWAN-СЕТЕЙ

Часть 1

Представлен обзор существующих программных и программно-аппаратных инструментов для моделирования сети LoRaWAN. Рассмотрены их особенности и возможность применения для исследования уязвимостей оконечных устройств, базовых станций и протокола.

Ключевые слова: LoRaWAN, LoRa PHY, LoRa MAC, Интернет вещей, моделирование сети.
Для цитирования: *Степанова Е. А., Ашаева П. А.* Использование средств программного и программно-аппаратного моделирования для исследования уязвимостей LoRaWAN-сетей (часть 1) // Техника радиосвязи. 2021. Выпуск 2 (49). С. 21–30. DOI: 10.33286/2075-8693-2021-49-21-30.

© *E. A. Stepanova*[1], *P. A. Ashaeva*[2]
[1]Omsk Scientific Center SB RAS (Institute of Radiophysics and Physical Electronics), Omsk, Russian Federation
[2]Omsk Scientific-Research Institute of Instrument Engineering, Omsk, Russian Federation

USING SOFTWARE AND HARDWARE SIMULATION SYSTEMS FOR LORAWAN VULNERABILITY RECEARSH

Part 1

This paper presents an overview of existing software and hardware simulation tools for the LoRaWAN network, considers their features and the possibility of using them to analyze the vulnerabilities of end devices, gateways and protocol.

Keywords: LoRaWAN, LoRa PHY, LoRa MAC, Internet of things, network simulation.
For citation: *Stepanova E. A., Ashaeva P. A.* Using software and hardware simulation systems for LoRaWAN vulnerability recearsh (part 1) // Radio communication technology. 2021. Issue 2 (49), pp. 21–30. DOI: 10.33286/2075-8693-2021-49-21-30.

Введение

В связи с активным внедрением технологии Интернета вещей (Internet of Things, IoT) в промышленность, сельское и жилищно-коммунальное хозяйство, размещением сетевой инфраструктуры на объектах критической информационной инфраструктуры, использованием сенсорных сетей на беспилотных летательных аппаратах и другим все острее встает вопрос о комплексном подходе к безопасности таких систем. К сожалению, при существующем разнообразии протоколов и их версий, а также большого количества различных аппаратных реализаций не все аспекты, связанные с информационной безопасностью, детально проработаны. Оценить правильность принятых технических решений, разработать сценарии реагирования на различные инциденты и внештатные ситуации помогают системы программного или программно-аппаратного моделирования.

Целью настоящей работы является оценка возможностей программного моделирования для проведения пентеста беспроводных сетей, построенных на базе технологии Интернета вещей на примере технологии LoRa.

Протокол LoRaWAN (long range wide area network), предназначенный для использования в нелицензируемом диапазоне частот ISM868, является одним из наиболее популярных в России решений для организации энергоэффективных сетей дальнего радиуса действия. Рассмотрим существующие возможности для моделирования сети LoRa.

Основными угрозами безопасности LoRa на сегодняшний день все еще остаются:

– атаки воспроизведения (replay attacks);

– компрометация ключевой информации путем физического доступа к устройству;

– подавление радиосигнала;

– атака посредника (man in the middle);

– отказ в обслуживании (distributed denial of service);

– фальсификация сообщений подтверждений (ACK-спуфинг) [1, 2].

Для воспроизведения атак на оконечные устройства, базовые станции и другие компоненты IoT-инфраструктуры не всегда требуется физически строить сеть, ведь покупка оборудования влечет за собой значительные затраты. Наиболее доступный способ анализа работы сети – это использование бесплатного программного обеспечения с открытым исходным кодом, позволяющего при необходимости самостоятельно разрабатывать и добавлять нужные программные блоки, необходимые при реализации различных сценариев атак. Рассмотрим несколько вариантов построения (моделирования) сети без использования покупных устройств.

Программные симуляторы

Существующие программные симуляторы для сетей, работающих по протоколу LoRaWAN, предназначены прежде всего для решения задач, связанных с построением сети, таких как исследование масштабируемости, оптимизация энергопотребления, повышение качества обслуживания, исследование влияния помех, оценка производительности и др.

Одним из популярных инструментов для программного моделирования сетей LoRa является симулятор LoRaSim [3], который написан на языке Python с

использованием библиотек SimPy, Matplotlib и NumPy. Он позволяет реализовать сеть, состоящую из *N* оконечных узлов и *M* шлюзов с максимальным количеством поддерживаемых базовых станций – 24. В качестве метрик оценки в LoRaSim предусмотрен расчет скорости извлечения данных (DER) и энергопотребления в сети (NEC). Данные метрики используются для оценки всей сети в целом, а не каждого узла в отдельности. Во второй дополненной версии симулятора под названием FREE [4] добавлена поддержка двунаправленной связи, т. е. восходящего и нисходящего каналов, улучшены метрики оценки энергопотребления, введены ограничение рабочего цикла и возможность исследования влияния затухания на работу сети.

Решение LoRaWANSim является расширением первой версии LoRaSim, к функционалу которого была добавлена реализация MAC-уровня, появилась возможность организовывать двунаправленную связь. Данная программа не является решением с открытым исходным кодом, поэтому далее ее рассматривать не будем.

Симулятор, написанный на языке программирования Java [5], предназначен для американского диапазона частот US915 (902–928 МГц) [6]. Помимо полосы частот, американский стандарт отличается от российского битовой скоростью передачи данных, мощностью передатчика и размером PHY полезной нагрузки, поэтому этот симулятор не может быть использован для моделирования российской версии LoRaWAN.

Для сетевого симулятора сетей NS-3 написаны модули LoRaWAN. С помощью решения [7, 8] можно смоделировать сеть уже из десятков тысяч оконечных устройств, оно содержит в себе следующие базовые компоненты:

– модель измерения мощности сигнала на приемном устройстве (Link Measurement Model);

– модель для измерения затухания сигнала при прохождении через стены (Building Penetration Loss Model);

– модель затухания сигнала с различными переменными (Correlated Shadowing model);

– модель оценки производительности канала при изменении параметра расширения спектра SF.

Модуль [9] можно использовать для изучения сетей с несколькими шлюзами и организации двунаправленного канала связи (Uplink, Downlink). Оценка энергопотребления в нем не поддерживается. Модель ошибок для симулятора построена на базе канала основной полосы частот приемопередатчика при воздействии аддитивного белого гауссовского шума (AWGN). Модуль [10] поддерживает нисходящий трафик.

LoRaEnergySim [11] позволяет построить программную модель канала между оконечным устройством и базовой станцией. Данный симулятор поддерживает адаптивную скорость передачи данных и нисходящий трафик. Недостатком можно назвать отсутствие моделирования поведения сервера сети LoRaWAN. Решение FLoRa [12], основанное на сетевом симуляторе OMNET++, представляет собой платформу, способную имитировать сквозную связь в сетях LoRa. Благодаря применению технологии адаптивной скорости передачи данных, FLoRa может динамически управлять параметрами моделирования, а также выполнять сбор данных об энергопотреблении каждого оконечного устройства.

В табл. 1 представлена сравнительная характеристика различных программных симуляторов с открытым исходным кодом.

Таблица 1

Функционал симуляторов с открытым исходным кодом

Название (язык)	*Помехи по соседнему каналу*	*Прием более сильного сигнала*	*Трафик нисходящего канала*	*Ограничение рабочего цикла*	*Расчет потребления энергии*
LoRaSim (python)	–	+	–	–	+
FREE (python)	+	+	+	+	+
Java, 915 МГц	–	+	–	–	–
NS-3 модуль PHY (C++)	–	+	–	+	+
NS-3 модуль MAC (C++)	+	+	+	+	–
LoRaEnergySim (python)	–	+	+	+	+
FLoRa (C++)	–	+	+	+	+

Как видно из табл. 1, существующие программные симуляторы предназначены для решения определенных специфических задач, не связанных с информационной безопасностью, например, оценка нагрузки на базовые станции, определение предельного расстояния распространения сигнала в зданиях и т. п. Следует отметить, что готовых симуляторов, позволяющих воспроизвести угрозы безопасности LoRaWAN-сетей, в настоящее время не существует. Тем не менее их можно создать на основе уже готовых решений.

Так, например, в симуляторы LoRaSim, FREE, Java 915 МГц и LoRaEnergySim необходимо внести функции сетевого уровня. Для этого в каждом симуляторе нужно добавить сетевой сервер, а также внести в него соответствующие функции (например, анализ поля контрольной суммы пакета). Помимо этого, в функции сервера входят шифрование и дешифрование пакетов, в соответствии со стандартом LoRaWAN. Стоит отметить, что реализация этих функций в симуляторах является нетривиальной задачей, поскольку для этого необходимо подробно изучить существующие стандарты, а также провести анализ работы сети, созданной на основе официальных программных и аппаратных компонентов. Также в коде к модулю Java 915 МГц необходимо изменить частотный диапазон, скорость передачи данных, мощность передатчика и размер PHY полезной нагрузки.

В симуляторах NS-3 и FLoRa такой компонент, как сетевой сервер, уже присутствует. Однако чтобы исследовать уязвимости сети, в симуляторы необходимо добавить новый компонент – атакующее устройство. Функционал атакующего устройства зависит от типа атаки. Так, например, для реализации DDoS-атаки устройство должно отправлять большое количество сообщений в короткий промежуток времени. Для создания такого атакующего устройства в качестве основы можно использовать код для обычного оконечного устройства, изменив в нем период отправки сообщений в меньшую сторону.

Стоит отметить, что в симуляторе FLoRa по сравнению с NS-3 у передаваемых сообщений отсутствует возможность изменения параметров сообщения (например, его длины, полезной нагрузки), поэтому FLoRa нельзя использовать для атаки ACK-спуфинг.

Программно-аппаратные симуляторы

Для использования LoRa-технологии в соответствующих устройствах должен присутствовать проприетарный радиочип [13]. Принцип работы физического уровня был запатентован, и в настоящее время не разглашается разработчиками, однако в открытом доступе можно найти информацию об исследовании этого уровня методом обратной инженерии. Благодаря этому, функционал приемопередатчика может быть реализован с помощью программно-определяемого радио (SDR). Для обработки полезной нагрузки, которую необходимо передать через SDR, на ПК используют программы, написанные на языке Python, C++ или других. Входные данные в виде квадратурных выборок с вычислителя (стационарного компьютера или ноутбука) поступают на передатчик. При приеме сигнала выполняются аналогичные действия в обратном порядке. Такие программы-обработчики можно писать самостоятельно или использовать готовый открытый инструментарий, например GNU Radio [14].

С помощью GNU Radio можно создать большое количество схем, таких как радиоприемник или радиопередатчик, используя программные блоки. Каждый блок выполняет определенную функцию: модуляцию, декодирование, фильтрацию и т. д. Для LoRa существует несколько готовых модулей под SDR.

Одним из таких решений является модуль gr-lora, созданный Питером Робинсом [15]. Данный модуль можно применять для версий GNU Radio 3.7 и 3.8, он позволяет обрабатывать как сигнал, транслируемый устройством в реальном времени, так и ранее записанный в файл. Здесь имеется модель приемника, который может интерпретировать сигналы, поступающие с оконечных устройств. Недостатком этой реализации является отсутствие модели передатчика, т. е. для передачи сигналов нет возможности использовать SDR. На основе этого решения были реализованы многие последующие модули.

Модуль gr-lora_sdr [16] содержит в себе примеры одноканального приемника и передатчика. В имеющихся моделях есть функция проверки CRC заголовка и полезной нагрузки кадра. К модулю имеется исходный код, поэтому в него можно легко внести изменения. В gr-lora_sdr находится большое количество блоков, каждый из которых выполняет свою функцию: детектирование передачи кадра, извлечение его заголовка, декодирование и т. д. Для его использования необходимо программно задать такие параметры, как: полоса частот; несущая частота; коэффициент расширения спектра; скорость кодирования; режим передачи заголовка (явный или неявный); наличие или отсутствие поля CRC.

Блок-схема приема и передачи LoRa-сигналов представлена на рис. 1.

Для передачи сообщений с помощью gr-lora_sdr необходимо выполнить последовательно: сложение по модулю 2; добавление заголовка; вычисление контрольной суммы (CRC) полезной нагрузки; кодирование Хэмминга; перемежение; кодирование Грея; модуляцию.

Сложение по модулю 2 применяют к полезной нагрузке, представленной в виде битов. Вторым слагаемым является псевдослучайная последовательность, заданная в виде массива длиной 255 элементов, как представлено на рис. 2. Данное действие позволяет устранить постоянное амплитудное смещение – отклонение волновой формы относительно оси нулевого уровня, возникающее при модуляции сигнала, и тем самым снизить вероятность битовой ошибки.

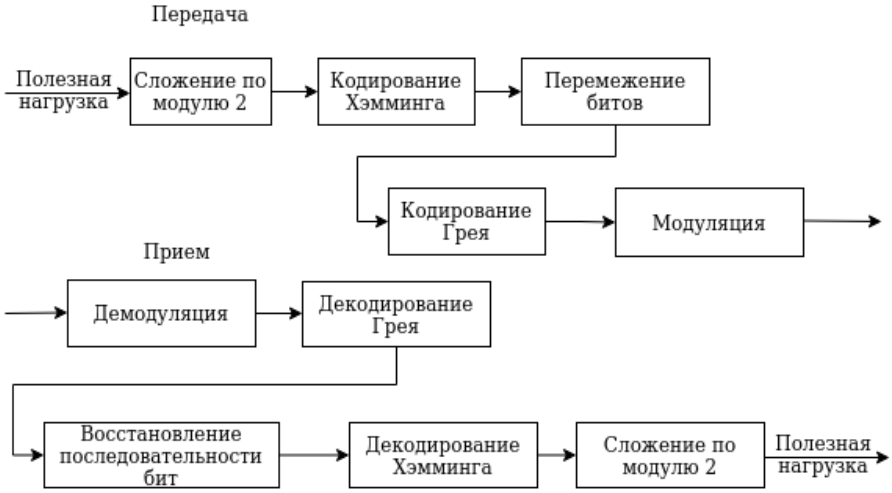

Рис. 1. Упрощенная схема приема и передача сигнала в модуле gr-lora_sdr

```
const uint8_t whitening_seq[] = {
    0xFF, 0xFE, 0xFC, 0xF8, 0xF0, 0xE1, 0xC2, 0x85, 0x0B, 0x17, 0x2F, 0x5E, 0xBC, 0x78, 0xF1, 0xE3,
    0xC6, 0x8D, 0x1A, 0x34, 0x68, 0xD0, 0xA0, 0x40, 0x80, 0x01, 0x02, 0x04, 0x08, 0x11, 0x23, 0x47,
    0x8E, 0x1C, 0x38, 0x71, 0xE2, 0xC4, 0x89, 0x12, 0x25, 0x4B, 0x97, 0x2E, 0x5C, 0xB8, 0x70, 0xE0,
    0xC0, 0x81, 0x03, 0x06, 0x0C, 0x19, 0x32, 0x64, 0xC9, 0x92, 0x24, 0x49, 0x93, 0x26, 0x4D, 0x9B,
    0x37, 0x6E, 0xDC, 0xB9, 0x72, 0xE4, 0xC8, 0x90, 0x20, 0x41, 0x82, 0x05, 0x0A, 0x15, 0x2B, 0x56,
    0xAD, 0x5B, 0xB6, 0x6D, 0xDA, 0xB5, 0x6B, 0xD6, 0xAC, 0x59, 0xB2, 0x65, 0xCB, 0x96, 0x2C, 0x58,
    0xB0, 0x61, 0xC3, 0x87, 0x0F, 0x1F, 0x3E, 0x7D, 0xFB, 0xF6, 0xED, 0xDB, 0xB7, 0x6F, 0xDE, 0xBD,
    0x7A, 0xF5, 0xEB, 0xD7, 0xAE, 0x5D, 0xBA, 0x74, 0xE8, 0xD1, 0xA2, 0x44, 0x88, 0x10, 0x21, 0x43,
    0x86, 0x0D, 0x1B, 0x36, 0x6C, 0xD8, 0xB1, 0x63, 0xC7, 0x8F, 0x1E, 0x3C, 0x79, 0xF3, 0xE7, 0xCE,
    0x9C, 0x39, 0x73, 0xE6, 0xCC, 0x98, 0x31, 0x62, 0xC5, 0x8B, 0x16, 0x2D, 0x5A, 0xB4, 0x69, 0xD2,
    0xA4, 0x48, 0x91, 0x22, 0x45, 0x8A, 0x14, 0x29, 0x52, 0xA5, 0x4A, 0x95, 0x2A, 0x54, 0xA9, 0x53,
    0xA7, 0x4E, 0x9D, 0x3B, 0x77, 0xEE, 0xDD, 0xBB, 0x76, 0xEC, 0xD9, 0xB3, 0x67, 0xCF, 0x9E, 0x3D,
    0x7B, 0xF7, 0xEF, 0xDF, 0xBF, 0x7E, 0xFD, 0xFA, 0xF4, 0xE9, 0xD3, 0xA6, 0x4C, 0x99, 0x33, 0x66,
    0xCD, 0x9A, 0x35, 0x6A, 0xD4, 0xA8, 0x51, 0xA3, 0x46, 0x8C, 0x18, 0x30, 0x60, 0xC1, 0x83, 0x07,
    0x0E, 0x1D, 0x3A, 0x75, 0xEA, 0xD5, 0xAA, 0x55, 0xAB, 0x57, 0xAF, 0x5F, 0xBE, 0x7C, 0xF9, 0xF2,
    0xE5, 0xCA, 0x94, 0x28, 0x50, 0xA1, 0x42, 0x84, 0x09, 0x13, 0x27, 0x4F, 0x9F, 0x3F, 0x7F
};
```

Рис. 2. Матрица с псевдослучайными значениями

Заголовок кадра содержит информацию о длине полезной нагрузки, скорости кодирования, наличии поля CRC для полезной нагрузки и CRC самого заголовка. Циклический избыточный код (CRC – Cyclic redundancy check) используется для контроля безошибочной передачи заголовка и полезной нагрузки. Если заголовок кадра присутствует, то кадр передается в явном режиме (explicit mode), иначе будет выбран неявный режим (implicit mode). Информация заголовка должна быть указана в настройках приемника.

Кодирование Хэмминга также используют для обнаружения и коррекции ошибок при передаче. Для этого в сообщение добавляют избыточные биты, вычисленные на основе изначально передаваемых данных. Количество избыточных битов определяет скорость кодирования (от 4/5 до 4/8 включительно).

Операция перемежения необходима для минимизации количества ошибок от помех, накладываемых на подряд идущие биты. Кодирование Грея позволяет избежать большого числа неверно интерпретированных бит на приеме, так как соседние символы отличаются всего на один бит.

Блоки приемника, соответственно, выполняют обратные операции: синхронизацию кадра; демодуляцию; декодирование Грея; восстановление последовательности бит; декодирование Хэмминга; декодирование заголовка; сложение по модулю 2; проверку контрольной суммы (CRC).

Синхронизация кадра в эфире происходит путем выявления его преамбулы, которая предшествует началу сеанса связи. Прием сигнала осуществляется с учетом возможных расхождений между тактовой и несущей частотой приемни-

ка и передатчика, что особо актуально для дешевых малопотребляющих устройств, использующих в качестве источника опорной частоты кварцевые резонаторы или малопотребляющие термокомпенсированные генераторы.

В качестве примера в симуляторе NS-3 было передано сообщение: «LoRa payload». Как видно из рис. 3, на стороне приемника, помимо самого сообщения, также выводтся информация о заголовке и результате проверки CRC.

```
--------Header--------
Payload length: 12
CRC presence: 1
Coding rate: 4
Header checksum valid!

message: LoRa payload

CRC valid!
```

Рис. 3. Вывод принятого сообщения на стороне приемника

Описанный выше модуль позволяет реализовать только физический уровень технологии (PHY LoRa), информация о котором не разглашается разработчиками технологии. Для создания полноценной сети необходимы также канальный и сетевой уровни. Поэтому для создания сети таким методом необходимо потратить больше времени на изучение технологии. Так, можно программно внести функционал вышестоящих уровней, добавить поддержку передачи сообщений на нескольких каналах, а также добавить новые опции, например дополнительное шифрование.

Модуль gr-lora2 [17] является доработкой предыдущей версии и позволяет создать программы для SDR-приемопередатчика. Отличием от предыдущего варианта является разбиение на большее количество блоков. Благодаря этому каждый блок выполняет более узкую задачу, что позволяет подробнее изучить процесс формирования и обработки сигнала. В табл. 2 представлено сравнение функциональных возможностей описанных модулей GNU Radio для SDR.

Таблица 2

Сравнение программно-аппаратных симуляторов

Функция	*gr-lora*	*gr-lora2*	*gr-lora_sdr*
Модель передатчика	–	+	+
Поддержка GNU Radio 3.8	–	+	+
Прием сигнала с реальных устройств	+	–	+
Поддержка изменения параметров кадра	–	–	+
Декодирование сигнала из файла	+	+	–

Из данных табл. 2 можно увидеть, что у каждого из модулей есть свои ограничения, однако, дополнив их функционал, можно реализовать такие угрозы безопасности, как атака посредника или человек посередине.

Для дальнейших исследований был выбран инструментарий gr-lora_sdr, так как в нем реализована поддержка приема сигнала с реальных устройств.

Возможные векторы атак

На текущий момент известно довольно много уязвимостей сетей LoRa. Стоит отметить, что, так как стандартом предусмотрено два способа активации оконечных устройств: активация «по воздуху» (OTAA) и персонализация устройства (ABP), уязвимости, основанные на перехвате статических сессион-

ных ключей, будут актуальны лишь для варианта с ABP. Рассмотрим наиболее актуальные, на наш взгляд, векторы атак на сеть LoRa. Их можно разделить на группы в зависимости от компонента сети, на который осуществляется атака.

1. Атаки на сервер приложений.

Сервер приложений отвечает за удаленное управление и взаимодействие с пользователем, физически он может быть объединен с сетевым сервером. К атакам на сервер относятся такие способы, как перебор вариантов учетных данных администратора и изменение данных на уровне бит, передаваемых между сетевым сервером и сервером приложений (табл. 3).

2. Атаки на оконечные устройства.

К реализуемым атакам на оконечные устройства Lora можно отнести риски, связанные с возможностью физического доступа. Это актуально в том случае, если оборудование установлено в общедоступном или слабо защищенном от проникновения месте, что встречается довольно часто. В таких ситуациях нарушитель может попытаться выполнить клонирование прошивки через интерфейс SPI или UART, а затем извлечь ключевую информацию для дальнейшего перехвата сообщений или подмены датчика. Также возможно осуществить перезагрузку оборудования для сброса счетчика пакетов и дальнейшего проведения атаки повторного воспроизведения.

Таблица 3

Возможные векторы атак на сервер приложений

Тип	*Описание*	*Цель*
Подбор учетных данных методом перебора	Подбор логина и пароля для доступа к учетной записи администратора сети	Компрометация учетной записи, чтение/изменение данных об используемых оконечных устройствах
Инъекции в базы данных	Внедрение команд в доступные поля ввода web-формы	Получение данных о конфигурации сервера/изменение данных
HTML-инъекции	Внедрение HTML кода в доступные поля ввода web-формы	Получение данных о конфигурации сервера, затруднение клиентского доступа, распространение вредоносного ПО
Атака посредника (MITM)	Внедрение в канал связи, перехват трафика между клиентом и сервером и/или сетевым сервером и сервером приложений	Извлечение/изменение информации, передаваемой от клиента к серверу/от сетевого сервера к серверу приложений
Изменение данных на уровне бит (bit flipping) [1]	Изменение определенных битов зашифрованного сообщения так, чтобы сервер смог его расшифровать и принять измененные данные как верные	Подмена передаваемых сообщений

3. Атаки на базовые станции.

Атаки на базовые станции, представленные в табл. 4, могут быть как пассивными (прослушивание трафика), так и активными, например несанкционированный доступ (НСД), отказ в обслуживании (DoS) или ACK-спуфинг. Отдельно можно выделить случаи, связанные с возможностью физического доступа к оборудованию.

Таблица 4

Возможные векторы атак на базовые станции

Тип	*Описание*	*Цель*
НСД	Нелегитимный доступ к БС через ssh/com порт/BS-Dashboard (web-приложение)	Изменение параметров работы/отключение БС, блокирование ресурсов
DoS/подавление радиопомехами	Отправка большого количества запросов/данных на базовую станцию	Затруднение/отказ доступа к сети легитимным устройствам
Пассивный перехват трафика	Прослушивание трафика между базовой станцией и сетевым сервером	Извлечение передаваемой информации, анализ трафика от БС к сетевому серверу
Активный перехват трафика	Внедрение в канал, перехват трафика между сетевым сервером и БС так, чтобы стороны считали, что общаются друг с другом без посредника	Извлечение/изменение передаваемой информации, анализ трафика
Подмена БС [2]	Подключение к сети нелегитимной БС	Извлечение/изменение передаваемой информации, анализ трафика между БС и сетевым сервером
ACK-спуфинг [1]	Отправка подтверждения о принятии сообщения на оконечное устройство	Выборочная передача сообщений на сетевой сервер

4. Атаки совместимости версий.

Иногда при эксплуатации оборудования возникают ситуации, когда оконечное устройство и сетевой сервер используют различные версии протокола LoRaWAN, тогда у нарушителя появляется возможность обойти механизм защиты. Возможные для такой ситуации атаки представлены в табл. 5.

Таблица 5

Возможные векторы атак при различных версиях протокола

Атакуемый компонент	*Тип атаки*	*Описание*	*Цель*
Совместимость версий (оконечное устройство и сервер)	Отказ в обслуживании (DoS-атака) [18]	Создание разных сессионных ключей на сервере и оконечном устройстве	Отказ доступа к сети легитимным устройствам
Совместимость версий (оконечное устройство и БС)	Атака воспроизведения (replay attack) [18]	Использование более ранних сообщений от оконечного устройства для перенаправления трафика	Обрыв связи между легитимным оконечным устройством и базовой станцией

Заключение

Как видно из табл. 3–5, существует множество известных векторов атак как на физические устройства, так и на протокол. На сегодняшний день специализированных решений по исследованиям в области информационной безопасности сети LoRa нет, однако, воспользовавшись средствами программного и программно-аппаратного моделирования, возможно успешно исследовать неко-

торые виды атак. Во второй части статьи будут представлены результаты экспериментов с программными симуляторами.

Финансирование

Работа выполнена по государственному заданию Омского научного центра СО РАН в соответствии с Программой ФНИ ГАН на 2013–2020 годы (номер регистрации проекта в системе ЕГИСУ НИОКТР АААА-А19-119052890058-2).

ЛИТЕРАТУРА

1. *Xueying Yang, Evgenios Karampatzakis, Christian Doerr, and Fernando Kuipers.* Security Vulnerabilities in LoRaWAN. URL: https://www.cyber-threat-intelligence.com/ publications/IoTDI2018-LoraWAN.pdf.

2. *Lukas S*. Laufenberg Impersonating LoRaWAN gateways using Semtech PacketForwarder. URL: https://arxiv.org/pdf/1904.10728.pdf.

3. LoRaSim. URL: https://www.lancaster.ac.uk/scc/sites/lora/lorasim.html.

4. LoRaFREE simulator. URL: https://www.github.com/kqorany/FREE.

5. LoRaSim simulator. URL: https://www.thingscs.vcalgary.ca/lorasim.zip.

6. Lora Alliance. URL: https://www.lora-alliance.org/wp-content/uploads/2020/11/lorawan_ regional_parameters_v1.0.3reva_0.pdf.

7. *Magrin D., Centenaro M., Vangelista L*. Performance Evaluation of LoRa Net-works in a Smart City Scenario (Source Code on GitHub). URL: https://github.com/ signetlab-dei/lorawan.

8. *Martin D., Centenaro M., Vangelista L*. Performance Evaluation of LoRa Net-works in a Smart City Scenario // IEEE International Conference on Communications (ICC), 2017. P. 1–7.

9. Lora-ns3-module. URL: https://github.com/drakkar-lig/lora-ns3-module.

10. Ns-3-dev-git. URL: https://github.com/imec-idlab/ns-3-dev-git/tree/lorawan.

11. LoRaEnergySim. URL: https://github.com/GillesC/LoRaEnergySim.

12. FLoRa. URL: https://flora.aalto.fi/.

13. *Tapparel J., Afisiadis O., Mayoraz P., Balatsoukas-Stimming A., Burg A.* An open-source LoRa physical layer prototype on GNU Radio. URL: https://arxiv.org/pdf/2002/08.208.pdf.

14. GnuRadio. URL: https://www.gnuradio.org.

15. Gr-lora. URL: https://github.com/BastilleResearch/gr-lora.

16. Gr-lora_sdr. URL: https://github.com/tapparelj/gr-lora_sdr.

17. Gr-lora 2. URL: https://github.com/rpp0/gr-lora.

18. *Tahsin C. M. Dönmez, Ethiopia Nigussie*. Security of LoRaWAN v1.1 in Backward Compatibility Scenarios // Procedia Computer Science. 2018. Vol. 134. P. 51–58.

Сведения об авторах

Степанова Елизавета Андреевна, канд. техн. наук, старший научный сотрудник ИРФЭ ОНЦ СО РАН, e-mail: trs@oniip.ru.

Ашаева Полина Александровна, сотрудник АО «ОНИИП», e-mail: trs@oniip.ru.

Поступила в редакцию: февраль 2021 г.
Рецензирование: февраль 2021 г.
Принята в печать: май 2021 г.