LoRaWAN
в разрезе безопасности

Олег ПЛОТНИКОВ,
директор центра Промышленного Интернета, компания «Интерсвязь»

Технология беспроводной передачи данных – LoRa – используется на российском рынке уже давно и перестала быть диковинкой. Свои сети развернули несколько десятков операторов, заявляются даже два федеральных игрока. Однако у заказчиков LoRa до сих пор остаются сомнения в ее безопасности. В этой статье мы не будем касаться темы надежности передачи, а попробуем непредвзято рассмотреть вопрос информационной безопасности, причем не просто радиоканала LoRa как модуляции, а именно стандарта канального уровня LoRaWAN.

Активация LoRaWAN

Технология LoRa (Long Range), разработанная французской компанией Semtech, ориентирована на передачу небольших объемов данных огромным количеством устройств, т. е. LoRaWAN не предназначена для передачи больших объемов данных от одного устройства. Более того, стандарт это прямо запрещает через специальный параметр – duty cycle, который определяет максимальное время нахождения оконечного устройства в эфире, оно равно 1%. Речь в дальнейшем пойдет о спецификации версии 1.02-1.03, хотя самая свежая спецификация на сегодняшний день – 1.1, но она почти не внедрена на местах.

Для начала рассмотрим первоначальную регистрацию устройства в сети. В LoRaWAN этот процесс называется активацией. Ниже приведен список необходимых нам терминов. Если немного запутаетесь – можете возвращаться сюда и сверяться. Возвращаться наверняка придется, так как аббревиатуры многих терминов очень похожи. Кроме того, в описании приведены аналогии, чтобы вы понимали, с чем можно сравнить тот или иной термин. Точные аналогии подобрать не всегда возможно, потому не судите эту колонку слишком строго (рис. 1).

Активация в LoRaWAN может производиться по воздуху (Over-the-Air-Activation – OTAA) или по преднастройкам оператора (Activation by Personalization – ABP).

По воздуху

В случае активации по воздуху на радиомодуле должны быть заданы три параметра: его уникальный идентификатор (DevEUI), идентификатор сервера (AppEUI) и ключ сервера (AppKey). Со стороны сервера также должны быть получены следующие параметры: идентификатор радиомодуля,

Параметр	Значение	Аналогия/Физический смысл
DevEUI (Device Extended Unique Identifier)	Уникальный идентификатор устройства. Не должен повторяться вообще	Что-то вроде MAC-адреса
AppEUI (Application Extended Unique Identifier)	Уникальный идентификатор сервера	Наш логин для входа на сервер при первом соединении
AppKey (Application Key)	Ключ сервера	Это не пароль от логина выше. Он шифрует ответы сервера при активации. Аналогию придумать не удалось
DevAddr (Device Adress)	Уникальный адрес устройства в пределах сети. В разных сетях может повторяться	Что-то вроде «серого» IP-адреса
NwkSKey (Network Session Key)	Сессионный ключ сетевого сервера	Шифрует обмен пакетами между радиомодулем и сетевым сервером после активации
AppSKey (Application Session Key)	Сессионный ключ сервера приложений	Шифрует обмен пакетами между радиомодулем и сервером приложений после активации
MIC (Message Integrity Code)	Код целостности сообщения	Контрольная сумма сообщения

Рис. 1. *Расшифровка терминологии, используемой в LoRaWAN*

Size (bytes)	8	8	2
Join Request	AppEUI	DevEUI	DevNonce

Рис. 2. *Формат запроса к серверу join_request*

идентификатор сервера и ключ. То есть сервер должен изначально знать то устройство, которое попытается к нему присоединиться. Если мы знаем идентификаторы и ключи сервера, но наш DevEUI не внесен в его базу данных, то соединение не состоится. Тем не менее сервер будет «слушать» все устройства и принимать их пакеты, однако не сможет расшифровать и не будет отвечать таким устройствам.

При первоначальном включении радиомодуль отправляет в эфир пакет join_request с запросом на подключение на одной из трех заранее оговоренных частот присоединения. Этим пакетом он запрашивает, есть ли поблизости сеть, которая его «узнает». На рис. 2 приведен состав пакета join_request. Как видим, он содержит те самые DevEUI и AppEUI, а также DevNonce.

DevNonce – величина случайная. Сервер хранит ее в памяти и, если придет join_request с таким же DevNonce, как один из предыдущих, данный запрос проигнорирует. Это сделано для защиты от атаки повторения, когда злоумышленник может записать запрос на активацию, а потом повторить его со своего устройства. Кстати, защитой от подобной атаки могут похвастаться далеко не все стандарты IoT.

AppKey в данном сообщении напрямую не используется, но через него считывается контрольная сумма MIC в конце кадра. Этот ключ понадобится нам чуть дальше, в ответном сообщении сервера – join_accept. Причем запрос join_request передается в незашифрованном виде.

Положительный ответ сервера (join_accept) поступит в том случае, если серверу известны AppEUI и DevEUI, а также нет совпадения по полю DevNonce и проблем с контрольной суммой MIC. Иначе ответа не последует. Если все проверки пройдены, то сервер генерирует ответное сообщение join_accept (рис. 3).

Для взаимодействия с устройством сервер передает клиентскому оборудованию два сессионных ключа – сетевого взаимодействия с сервером (NwkSKey) и взаимодействия с приложением (AppSKey). Вместе с другой информацией они шифруются ключом шифрования сервера AppKey и отправляются радиомодулю. Далее все сообщения шифруются этими двумя сессионными ключами, что приводится в качестве аргумента защищенности LoRaWAN.

На самом деле ситуация с шифрованием сложнее. Если в пакете имеются данные для приложения и MAC-команды для сервера, то шифрование действительно выполняется двумя ключами. Пакет только с MAC-командами, которые не требуют от сервера обращения к приложению, ключом приложения уже не шифруется. Пакет с данными NwkSKey не принимает непосредственного участия в шифровании, но участвует в подсчете контрольной суммы и влияет на ее результат. Следует отметить, что NwkSkey и AppSKey уникальны для каждого отдельного клиента сети LoRaWAN.

Два ключа используются для дополнительного уровня защиты, чтобы сервер мог расшифровать только те послания, что адресованы ему, т. е. MAC-команды. Приложение в таком случае увидит лишь полезную составляющую пакетов – сами передаваемые данные. Нужно это для того, чтобы разделить уровни сети и приложения, поскольку сервер чаще всего будет стоять у провайдера, а вот приложение вполне может быть размещено у клиента. Двойное шифрование затрудняет провайдеру анализ содержания пакета, предназначенного для приложения.

Помимо двух ключей в join_accept по OTAA есть еще одна важная информация – расширенный список частот (CFList). Напомним, что изначально радиомодуль знает только три частоты, на которых он может работать. После активации ему передаются дополнительные частоты для выхода на связь, что очень удобно, если точно неизвестно, в какой сети будет работать устройство. Можно договориться, что во всех сетях три частоты (+RX2) будут всегда совпадать, остальные пять – на усмотрение провайдера. С увеличением количества нелицензируемых частот в поддиапазоне 868 это вдвойне актуально.

Преднастроенные устройства

Чтобы не заниматься активацией по воздуху, в LoRaWAN используется упрощенная процедура, когда сессионные ключи сразу зашиваются в радиомодуль и изначально прописаны со стороны провайдера. Радиомодуль сразу готов к работе. К сожалению, это не всегда удобно для большой сети, а также если устройства и сеть принадлежат различным компаниям. Кроме того, в данном случае нельзя менять частоты передачи динамически. Обычно такая процедура применяется в тестовых сетях, когда заранее известно, что с сервером могут происходить метаморфозы.

Size (bytes)	3	3	4	1	1	(16) Optional
Join Accept	AppNonce	NetID	DevAddr	DLSettings	RxDelay	CFList

Рис. 3. *Формат положительного ответа сервера join_accept*

Следует отметить: режим ABP устойчивее к атаке клонирования, но плата за такую устойчивость слишком высока.

Криптоанализ

Итак, основная нагрузка по шифрованию ложится на сессионные ключи сетевого сервера и сервера приложений. Рассмотрим их более подробно.

Главная претензия критиков стандарта LoRaWAN заключается в том, что при активации устройства в сети появляются два ключа, которые могут месяцами и даже годами не меняться, пока не произойдет реактивация устройства. В нормальных условиях радиомодуль активировали и забыли, так что срок жизни ключа в три-четыре года – вполне реальная перспектива. Собственно говоря, от установки до исчерпания заряда батарейки.

Насколько надежны наши ключи? Спецификация сообщает, что они соответствуют загадочному RFC4493. Что это такое? Это алгоритм шифрования, более известный как AES-CMAC. Мы не будем погружаться в дебри криптографии и ограничимся общим пониманием картины (рис. 4).

Принцип AES-CMAC примерно такой: шифруемое сообщение разбивают на 128-битные блоки (M1, M2, … Mn). Каждый блок шифруется отдельно AES-ключом. Причем при шифровании второго блока помимо ключа используется результат шифрования первого. А при шифровании третьего – результат второго и (косвенно) первого. Такая цепочка усложнений.

Насколько надежен этот принцип? Весьма надежен. Алгоритм вышел больше десяти лет назад. С тех пор на него провели множество различных атак и в конце концов в теории доказали, что его таки можно взломать. Проблема в том, что для взлома понадобится широкая выборка пакетов – несколько тысяч. Тогда есть шанс понять, что же внутри зашифрованных блоков.

Сможет ли злоумышленник с нужными знаниями получить данную выборку, если мы говорим про перехват пакетов LoRaWAN? Давайте прикинем. Пусть пакеты передаются раз в час. За месяц от радиомодуля уйдет 720 пакетов – маловато для взлома. Для реальной угрозы понадобится очень терпеливый злоумышленник, который будет месяцами писать пакеты. И то не факт, что он сможет взломать алгоритм, чтобы получить заветные ключи. Не забудем, что такое терпение надо будет проявлять в отношении каждого клиентского устройства отдельно. Кроме того, напомним, что передача пакетов раз в час – это ОЧЕНЬ часто. На практике промежутки намного больше – часов шесть, а то и раз в сутки.

Но даже эта призрачная возможность сейчас закрыта после выхода спецификации 1.1, где реализованы команды реактивации и запроса join_server. Понятие «дырявости» – это вообще удел открытых стандартов. Когда есть огромное сообщество, которое под микроскопом изучает все уязвимости, то они находятся быстро. И в этом преимущество открытого подхода. При очередной модернизации стандарта разработчики точно знают, на что смотреть в первую очередь. У проприетарных стандартов такой возможности нет, и их внутренние проблемы либо остаются известны только в среде разработчиков, либо вообще находятся уже в процессе эксплуатации в самый неподходящий момент.

В итоге получаем, что угроза безопасности скорее иллюзорна. Где-то в глубокой теории взлом произвести можно, но на практике вряд ли. Теперь помножим на ценность полученной информации. Станет ли наш злоумышленник месяцами писать пакеты, чтобы узнать показания счетчика? Маловероятно.

```
+------+          +------+          +------+
| M_1  |          | M_2  |          | M_n  |
+------+          +------+          +------+
   |                 |                 |     +--+
   |        +----> (+)        +----->(+)<-|K1|
   |        |        |        |        |     +--+
+------+    |     +------+    |     +------+
|AES_K|     |     |AES_K|     |     |AES_K|
+------+    |     +------+    |     +------+
   |        |        |        |
   +-------+         +-------+
```

Рис. 4. *Упрощенная схема шифрования по алгоритму AES-CMAC*

Возможен ли взлом?

Хорошо, расшифровать пакеты злоумышленнику будет проблематично. А как насчет возможности удаленно ввести в заблуждение сервер или организации атаки на отказ в обслуживании – DoS? Рассмотрим несколько наиболее популярных техник атаки на радиосети.

Атака повторения

В случае атаки повторения злоумышленник записывает сообщение от устройства (без расшифровки) и позже передает его в эфире, «прикидываясь» устройством. Такая атака не будет успешной, поскольку в каждом пакете есть контрольная сумма, привязанная к счетчику пакетов, т. е. следующий пакет, даже с тем же содержимым, должен иметь другую контрольную сумму.

DoS преамбулами

Преамбула – это часть сообщения, которая «привлекает»

внимание сервера. Что-то вроде «слушай, сейчас будет сообщение». Атака заключается в забивании эфира преамбулами, в результате чего сервер будет все время ждать какого-то сообщения и не получать его. Реальный пакет в такой ситуации может просто не пройти. Это классический SYN-flood, с которого начинались первые DoS-атаки на IP-сети. И вот тут, увы, LoRaWAN сдает позиции. Сервер прослушивает все пакеты в эфире, поэтому он весьма чувствительно реагирует на подобные атаки. Справедливости ради надо отметить, что такой болезнью страдают не только LoRaWAN, но практически все радиопротоколы.

Глушение

Передача в эфир мощной помехи на рабочих частотах – более простой способ DoS-атаки. Заглушить можно любую радиосвязь: вопрос в соотношении мощностей передатчиков полезного сигнала и помехи. У LoRaWAN очень хорошие показатели работы в шумном эфире, технология может работать даже ниже уровня шума. Но физику никто не отменял, и ощутимая помеха, особенно рядом с базовой станцией или устройством, вызовет потерю пакетов.

Человек посередине

Еще более сложная атака, когда между сервером и устройством вклинивается злоумышленник. Он перехватывает пакеты от устройства, а на сервер отправляет уже измененную информацию. И наоборот. Для проведения такой атаки в рабочей сети злоумышленник должен знать сессионные ключи и идентификаторы устройства. Их еще надо раздобыть (возвращаемся к разговору выше). Теоретически при наличии только идентификаторов можно попробовать вклиниться в момент активации, т. е. ввести в заблуждение и сервер, и клиентское устройство.

Но тут две проблемы. Первая – нужно точно знать момент активации, умудриться перехватить join-пакеты с обеих сторон и не дать им пройти по маршруту. Непростая задача в условиях, когда устройства активируются в зоне действия легальной базовой станции. Да и персонал провайдера может заподозрить что-то неладное по косвенным признакам. Например, по уровню сигнала от устройства или явному несоответствию приходящих данных. Второй вариант – добраться до устройства и произвести его переактивацию. Это уже ближе к реальности, и все равно внеочередные запросы на реактивацию могут насторожить инженеров провайдера. В общем, чисто технически это осуществимо, но требует высокой квалификации и хорошего оборудования, а также точного понимания логики работы не только технологии, но и инженеров провайдера.

Клонирование

Создание клона устройства – частный случай атаки человека посередине. В предыдущем случае мы вводим в заблуждение как сервер, так и оконечное устройство. В случае клонирования обманывать будем лишь сервер. Нужно устройство со всеми теми же идентификаторами, что и легальный терминал. Кроме того, режим активации должен быть выбран OTAA. Проводим активацию клона, сервер считает, что у нас переактивировалось оригинальное устройство и обновляет сессионные ключи. Готово. Клон теперь воспринимается сервером и приложением как легальный источник данных, а настоящее устройство сервер перестает слышать, поскольку стирает его сессионные ключи. Старый терминал продолжает слать пакеты, но сервер их уже не воспринимает из-за устаревших сессионных ключей.

Тут есть две тонкости. Во-первых, нужно откуда-то взять набор идентификаторов легального устройства для клона. Во-вторых, некоторые производители добавляют в оконечные устройства оригинальную защиту, не предусмотренную спецификацией. Например, если терминал слишком долго не слышит ответов от сервера, то он запрашивает реактивацию. И вот тут сервер будет игнорировать уже сообщения клона.

Плюс – внимательность инженеров на сервере, которые должны заметить слишком частые запросы на реактивацию устройства. Клон может повести себя не так, как оригинальное устройство, что тоже должно вызвать подозрения.

К примеру, у нас есть специальная проверка на большой выборке данных: водосчетчик шлет данные, а мы записываем их в базу. В какой-то момент вклинился злоумышленник и начал нам отправлять заниженные показания. Если в момент вклинивания очередной пакет с показаниями от водосчетчика будет меньше, чем предыдущий, то сработает триггер: «Счетчик начал мотать назад». Генерируется аварийный акт и назначается разбирательство «руками инженера», вплоть до выезда на место.

Не знаю, как у других, но подозреваю, что не одни мы такие умные, и большинство серверов крупных провайдеров также обложены проверками.

Заключение

Итог. Любой стандарт, любой канал связи можно взломать. Особенно беспроводной. Это всегда вопрос ресурсов и времени. Если вам говорят, что технология совершенна и никто не может ее взломать, – не верьте. Либо кривят душой, либо сами не знают про какую-нибудь уязвимость. Другой вопрос, когда изначально ясно, какие ресурсы потребуются для взлома. Тогда можно оценить, стоит ли овчинка выделки. Обычное соотношение цена/качество является основой для реальной информационной безопасности. Очевидно, что через LoRaWAN не будут передавать данные, которые могут заинтересовать ЦРУ или МОССАД, – спецслужбы с большими финансовыми возможностями. А месяцами слушать пакеты, чтобы узнать показания водосчетчика, причем без гарантии успеха – удел весьма странного и терпеливого злоумышленника. Такого еще поискать в наших широтах. ■