LoRaWAN Security

Olivier Seller

Technical Fellow, Semtech, and LoRa Alliance® Technical Committee Vice-Chair, Fremont, CA, United States
E-mail: oseller@semtech.com

Received 19 December 2019; Accepted 10 March 2020;
Publication 30 April 2021

Abstract

The LoRaWAN security design adheres to state-of-the-art principles: use of standard, well-vetted algorithms, and end-to-end security. The fundamental properties supported in LoRaWAN security are mutual end-point authentication, data origin authentication, integrity and replay protection, and confidentiality. The use of symmetric cryptography and prior secret key sharing between a device and a server enables an extremely power efficient and network efficient activation procedure.

Keywords: LoRaWAN, security, authentication, encryption, Join Server, activation, personalization, provisioning.

1 Introduction

The LoRaWAN protocol is optimized for low power consumption wide area networks. It supports low-cost, mobile, and secure bi-directional communication for Internet of Things applications. LoRaWAN networks can handle millions of devices. Security is a fundamental need in all IoT applications, it is therefore an important aspect of the LoRaWAN specification. LoRaWAN security fits the general LoRaWAN design criteria: low power consumption, low implementation complexity, low cost and high scalability. As devices are deployed in the field for long periods of time (years), security must

Journal of ICT, Vol. 9_1, 47–60. *River Publishers*
doi: 10.13052/jicts2245-800X.915

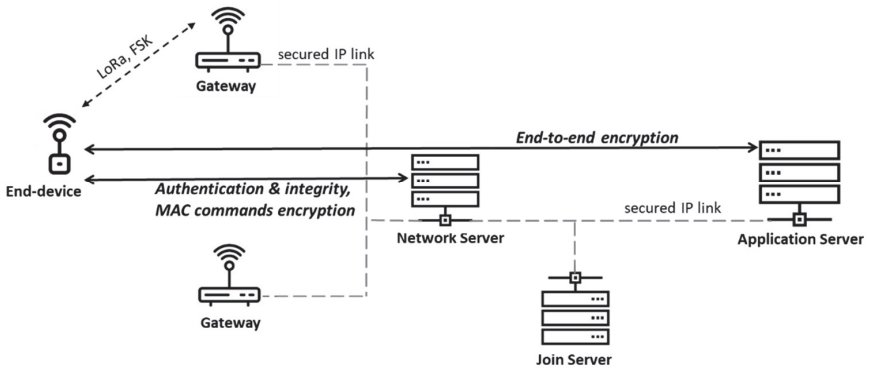

Figure 1 End-to-end security.

be future-proof. The LoRaWAN security design adheres to state-of-the-art principles: use of standard, well-vetted algorithms, and end-to-end security. Like all symmetric cryptography telecommunication systems, state-of-the-art implementation of root keys management is mandatory to ensure security at system level.

This paper focuses on version 1.0.3 of LoRaWAN link Layer specification [4], which is currently deployed with tens of millions of devices. This paper starts detailing security primitives and end to end security, then authentication and encryption functions once a device is operating on a network. After this, we describe how a device is activated when it joins a network, and last how provisioning is performed on end-device and network side.

2 Security Primitives

The fundamental properties supported in LoRaWAN security are mutual authentication, integrity protection, and confidentiality.

The security mechanisms of LoRaWAN rely on the well-tested and standardized AES (Advanced Encryption Standard) cryptographic algorithms. These algorithms have been analyzed by the cryptographic community for many years, are NIST approved, and widely adopted as a best security practice for constrained nodes and networks. LoRaWAN security in particular uses the AES cryptographic primitive combined with several modes of operation: CMAC (Cipher-based Message Authentication Code) [1, 2] for integrity protection, and CTR [1, 3] (Counter Mode Encryption) for encryption.

Each LoRaWAN device is personalized with a unique 128 bit AES key called AppKey, a globally unique identifier (EUI-64-based DevEUI), and a Join Server identifier (EUI-64-based JoinEUI), all of which are used during the device activation procedure. The root key AppKey is an AES-128 key specific to the end device. It should be generated or derived randomly with high entropy so that keys cannot be guessed by observing other devices of the same batch. DevEUI identifies a device roaming across LoRaWAN networks. JoinEUI identifies which Join Server shares the secret AppKey with the device. Allocation of EUI-64 identifiers require the assignor to have an Organizationally Unique Identifier (OUI) from the IEEE Registration Authority. Similarly, LoRaWAN networks are identified by a 24-bit globally unique identifier assigned by the LoRa Alliance®, called NetID.

The use of symmetric cryptography and prior secret key sharing between a device and a server enables an extremely power efficient and network efficient activation procedure. Symmetric cryptography is common in wireless cellular systems, where SIM cards store and use symmetric keys. In some other networks, usage of asymmetric cryptography avoids the need for prior key sharing, but this method pushes some complexity back to the end-device. Secure session negotiation using Asymmetric cryptography requires more data exchanges. This could potentially limit a device battery duration or its communication capabilities, since most LoRaWAN networks use unlicensed spectrum which have duty cycle or time on air limitations. This is why LoRaWAN security primitives rely on AES-128 symmetric cryptography.

3 End to End Security

LoRaWAN security implements end-to-end encryption for application payloads exchanged between the devices and Application Servers. In fact, LoRaWAN is one of the very few IoT networks that implements end-to-end encryption. In some traditional cellular networks, traffic is encrypted over the air interface, but it is transported as plain text in the operator's core network. Consequently, end users are burdened by having to select, deploy and manage an additional security layer, which is generally implemented by some type of VPN or application layer encryption security such as TLS. This approach is not well- suited in LPWANs, where over-the-top security layers add considerable power consumption, complexity, and cost.

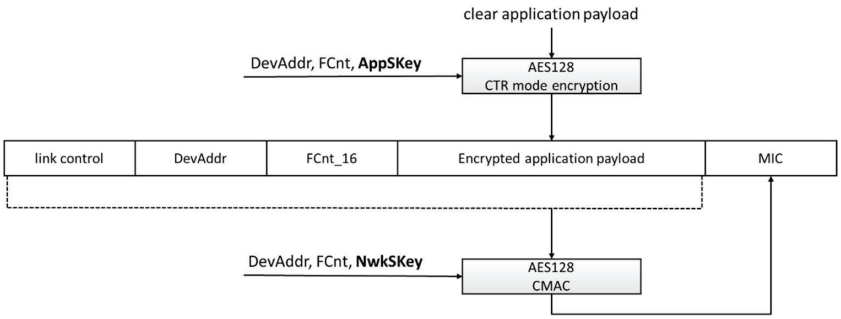

Figure 2 Frame authentication and encryption.

The gateways receive radio traffic from end-devices, and relay it to the network server through standard secured IP connections. The gateways are transparent for the end-devices; all messages are intended to a Network Server, without a specific route.

The Network Server authenticates and verifies messages integrity, then transmits the application payload through a standard secured IP connection. The Application Server who owns the device can decrypt the payload. Messages to the end device follow the opposite route. Join server is responsible for storing the AppKey for a given device identified by its DevEUI. During the activation procedure, the end-device and the Join Server use the AppKey as a shared secret key for mutual authentication, and generate session keys at the end of the successful activation procedure. Session keys are provided by the Join Server to the Network Server and to the Application Server for subsequent use.

4 Authentication and Encryption

All LoRaWAN traffic is protected using two session keys. Each payload is encrypted by AES-CTR, using the 128 bits application session key AppSKey. Each frame carries a frame counter to avoid packet replay, and a 32 bits Message Integrity Code (MIC) to avoid packet tampering. MIC is computed with AES-CMAC, using the 128 bits network session key NwkSKey.

The frame counter Fcnt is 32 bits long, but only the 16 least significant bits are transmitted as part of the frame, in order to reduce overhead. All 32 bits are used to compute the MIC. Payload encryption also takes the frame counter as part of its input. Therefore, the same clear payload results in a different encrypted payload for every frame.

There are two frame counters per device, one that counts the uplink frames from the device to the Network Server, and one that counts the downlink frames originated by the Network Server. When an uplink frame is sent, the uplink frame counter is used, and when a downlink frame counter is sent, the downlink frame counter is used. Both ends keep track of these counters to avoid replay attacks. The AES-CTR encryption scheme will generate different encryption streams for uplink and downlink, because communication direction is part of the input in addition to DevAddr, FCnt and key.

It is important that frame counter increments at each new frame, a given frame counter value shall not be re-used with the same NwkSKey and AppSKey. Some devices do not have the capability to renew these keys (so-called ABP devices). However, this is not a true limitation because 2^{32} messages, even at the fastest data rate, correspond to 1000 times what a typical battery can transmit.

To optimize the link performance, the LoRaWAN protocol has the option to force the end-device to transmit each frame several times. In that case, the repeated transmissions are strictly identical, and the Network Server knows how many repetitions it shall receive.

Device address DevAddr is the 32 bits network address of the device. It is assigned by the Network Server during device activation, and uniquely identifies a device on a given LoRaWAN network.

Link control field is composed of several fields such as frame type, protocol version, piggybacked acknowledgements, operation mode of the device, piggybacked MAC commands, adaptive data rate signaling. These fields are not encrypted, they only have a meaning between the end-device and the Network Server, for link layer related tasks. These fields are integrity protected, replay protected, and authenticated. MAC commands can be either present in this Link control field, or as application data. In that case, the payload of the frame is only composed of MAC commands, and NwkSKey replaces AppsKey for AES-CTR encryption of the payload, and encryption/decryption is carried out by the end-device and the Network Server. This possibility is useful to transmit many MAC commands in a single message, or to encrypt MAC commands for instance to keep secret network access optimization.

Verifying the MIC uses the same AES128-CMAC operation than its computation. Decrypting AES-CTR mode is also the same operation as encryption. This limits complexity on the device side, which only implements one operation for authentication, and one for encryption/decryption.

5 Secured Device Activation

Device activation consists of equipping the device with DevAddr, NwkSKey and AppSKey. Once activation is performed, NwkSKey is stored on the device and on the Network Server, AppSKey is stored on the device and on the Application Server. DevAddr identifies the device on a Network Server, and it includes a prefix AddrPrefix which identifies the network. This enables roaming, in a passive roaming mode, where visited gateways or Network Servers redirect traffic to the home Network Server.

There are two options to perform device activation. The first is called Activation By Personalization (ABP), and the second is Over The Air Activation (OTAA).

5.1 Activation by Personalization

Activation by personalization ties an end-device directly to a specific network. DevEUI, DevAddr, NwkSKey and AppSKey are stored directly in the end-device. The end-device is equipped with the required information for participating in a specific LoRaWAN network as soon as it starts. DevEUI, DevAddr, NwkSKey and AppSKey should also be stored in the Network Server and Application Server before the device starts operating. Compromising the keys of one end-device shall not compromise the security of the communications of other end-devices. The process to build the keys is left to the implementer; this process shall not be a derivation from publicly available information such as DevEUI or DevAddr.

5.2 Over the Air Activation

OTAA is the recommended practice for scalability. This procedure allows a device to change dynamically the network it operates on, for roaming or for portability. It also allows manufacturing devices independently from provisioning them on the network they will use.

OTAA consists of three messages: a join request message sent by the end-device, a join accept, and a confirmation of this cryptographic handshake, which consists of transmitting a normal uplink frame that is protected using the newly-generated session keys. From the join accept message, the end-device derives session keys from its AppKey. The join procedure has been designed for low power, low-throughput efficiency and high security.

Header signals on 8 bits that the frame is a join request, along with protocol major version. JoinEUI is 64 bits, it is the unique identifier of

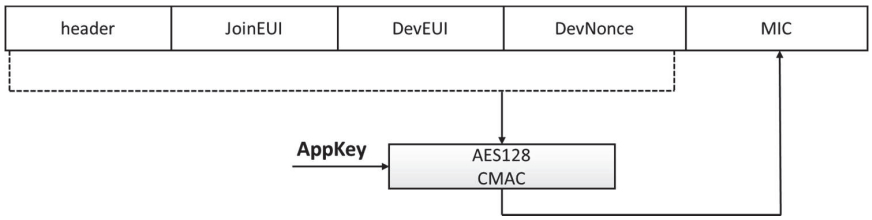

Figure 3 Join request frame.

the Join Server where the device is provisioned. This Join Server securely stores the (DevEUI, AppKey) pair of the end-device. This Join Server can authenticate the join request frame using its MIC.

DevNonce is a 16 bits counter or non-repeating value. It never repeats even over power cycles, and is updated or incremented at each new join request. A join request can only be transmitted once. DevNonce protects against replay attacks. The join request frame is not encrypted.

The network answers with a set of radio parameters, the DevAddr on 32 bits to identify the device on this network, the network identifier NetID on 24 bits, and a 24 bits JoinNonce. A join accept message is sent only if the join request is valid. The radio parameters have variable length depending on the region. They carry the downlink settings, the time delay between downlink and uplink in class A, and optionally a list of additional channels to use. DevAddr both identifies the network with a NwkID on 6 to 17 bits, and the device within this network with NwkAddr on 7 to 25 bits. LoRaWAN Backend interface specification [5] defines 7 classes of NetID, corresponding to different addressing spaces of end-devices. Depending on that class, NwkID contains all or part of NetID. NetIDs are allocated by the LoRa Alliance, they are used for network identification during roaming.

The Join accept is encrypted using AES-128 ECB mode. ECB is Electronic Codebook, it is a block cipher mode of operation. The decrypt function is used on the network side, which allows the end-node to decode the message using the AES-128 encrypt function. This way, the end-device only needs to implement the encrypt function, whether it is encrypting or decrypting a message.

JoinNonce is a non-repeating, 24 bits value provided by the Join Server. The Join Server and the end-device both derive the two session keys NwkSKey and AppSKey, using an AES 128 encryption by AppKey of (1,JoinNonce,NetID,DevNonce) and (2, JoinNonce,NetID,DevNonce)

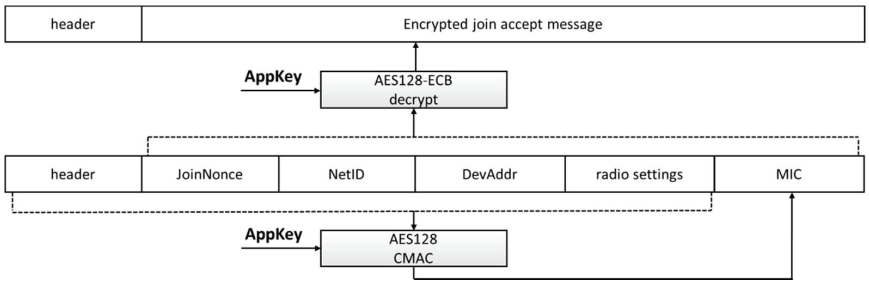

Figure 4 Join accept frame.

6 Device Personalization

LoRaWAN end-devices security credentials are provisioned during device personalization. Frequently, this is part of manufacturing and does not require end customer actions like inserting a SIM card into the device. This allows low-cost and efficient device deployments as any manual actions are very costly in IoT. Personalization is done by configuring three key parameters: DevEUI, JoinEUI, and AppKey. The Join Server identified by JoinEUI stores the device root key.

The final step of personalization is the selection of a Join Server that has activation agreements with the target Network Server operator(s). Each Join Server can expose one or multiple JoinEUI that uniquely identify it across all LoRaWAN networks. JoinEUI is injected in the device during the personalization phase so that the Join Server can be identified during activation.

The Network Servers will forward the join request to the Join Server corresponding to JoinEUI. Roaming platforms may be used for this. Alternatively, DNS resolution can get the IP address of a Join Server from its JoinEUI. In addition, JoinEUI and DevEUI can be combined before DNS resolution to direct different manufacturing batches of devices to different Join Servers.

The LoRaWAN root keys are the root of security of the entire system. One critical element of the device manufacturing is the key injection step. In most secure manufacturing lines, keys are generated in Hardware Secure Modules (HSMs), which are tamper-proof physical appliances. HSMs protect secrets from physical and remote accesses, and they connect securely to the appliance injecting the root key into the device. The AppKey is not exposed, even during manufacturing. When keys are not injected during

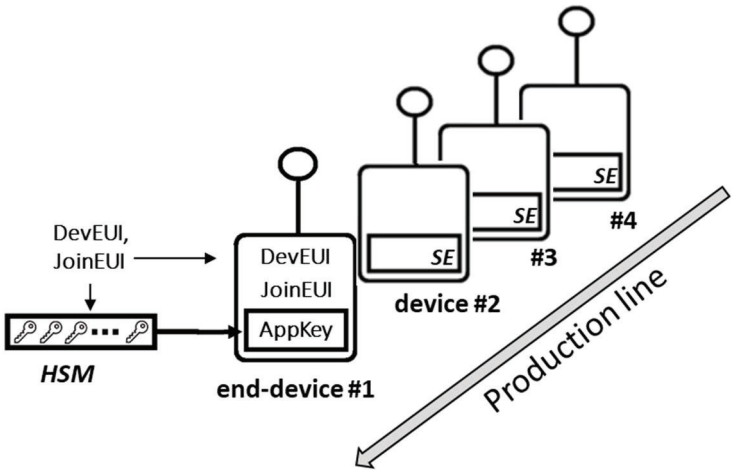

Figure 5 Example of device personalization using HSM and Secure Element.

manufacturing, it is possible to design secure remote protocols to generate and inject cryptographic secrets into the device via other connectivity layers than LoRaWAN.

For applications requiring a high level of security comparable to SIM cards, devices can be equipped with secure elements. These are tamper-proof subsystems, which are part of the device. They hold root keys injected by secure element manufacturers under high security and deriving session keys and MIC separately from any application code running on the main processor. They ensure no spying or hacking is possible when a device is physically accessible to attackers. A system on chip can integrate a secure element, this means the chip itself comes equipped with keys. DevEUI, JoinEUI and AppKey can then be provisioned at the same time during production, as shown on Figure 5.

7 Device Provisioning and Key Management

Before activation, devices must be provisioned out-of-band into the three LoRaWAN network elements: Join Server, Network Server, and Application Server. Join server is where the root key for security is stored. Network server and Application Server know the device but do not share cryptographic secrets with it at this stage, though they are ready to receive session keys securely from the Join Server.

Once the Join Server supplier is identified, keys are shared with it directly or indirectly through Key Management Systems (KMS). This step must be secured at the transport level and at the storage level on both sides. The Join Server and KMS also often make use of HSMs to secure the storage and even the transport of these secrets. In some cases, Join Server is determined during the personalisation of the device: keys are injected with HSMs during production into a batch of devices, and in a Join Server corresponding to this batch.

Join servers are also provisioned with Key Encryption Keys (KEKs) that are used to transport the session keys derived during device activation to the Network Server and the Application Server. These keys are AES-128 keys and are shared between the Join Server and network/Application Server securely. Sharing also uses KMSs.

The Join Server needs to know the home Network Server for the home discovery procedure, so it is provisioned with this NetID for roaming use cases.

Devices must be provisioned on a home Network Server. This server holds the primary connectivity for the device and can interact with visited Network Servers in case of roaming. The home network requires all necessary information to manage connectivity: device profile with radio channels configuration and supported MAC commands, connectivity plan, routing profiles to Application Server(s).

Finally, devices are provisioned in an Application Server. This server runs the end-user application(s). It should be provisioned with transport- and application-level information: KEK is used to securely transport the

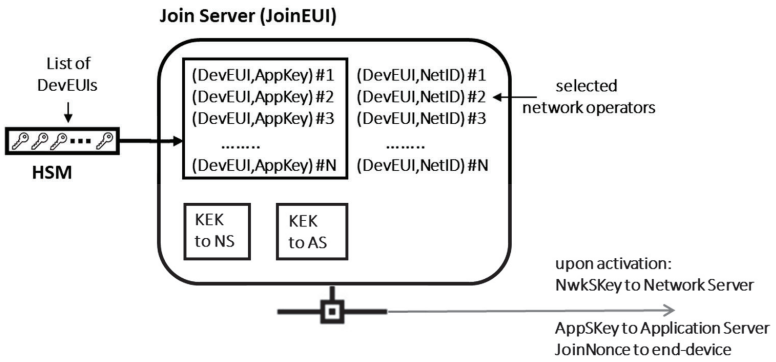

Figure 6 Example of device provisioning using HSM.

AppSKey, device data decoder, and any other business logic. The LoRaWAN standard specifies how binary data and the AppSKey are delivered to the Application Server. However, the rest of the security is out of the specification scope.

Using HSMs, root key generation occur securely and independently in the devices on the production line (see Figure 5), and in the Join Server as in Figure 6 above.

8 Future Evolutions

From a security features standpoint, the first protocol versions 1.0.1, 1.0.2 and 1.0.3 are identical. The LoRaWAN link layer specification version 1.0.4, published in April 2020, adds security improvements by mandating what were previously recommended practices: all frame counters are made 32 bits, which allows end-devices using ABP to never reset frame counters, and DevNonce shall be counters.

The next version of the protocol specification, LoRaWAN link layer specification version 1.1.1, will be published by the end of 2021. It further improves security by applying recommended practices for asymmetric cryptography. Two root keys are used, to separate the network from the application. Frame counters are also separated between network specific and application specific frames. In addition, more session keys are derived, to have one key per security usage (MIC computation, encryption, join procedure). Piggy-backed MAC commands to and from the network become systematically encrypted, which was not the case previously as these were only authenticated and integrity protected (frames carrying only MAC commands are encrypted in all LoRaWAN versions). Last, rekeying end-devices is made easier and more systematic with the addition of periodic re-join mechanisms.

9 Conclusion

The LoRaWAN protocol uses standard, well-vetted algorithms in an efficient way, fitting the complexity and power consumption limitations of end-devices and the constraints of the Sub-Ghz unlicensed ISM bands. The result is to allow simple, low-cost, mobile, and secure bi-directional communication for Internet of Things applications.

LoRaWAN covers end-to-end security, from key derivation at two endpoints to key distribution over multiple roaming networks, securing integrity, authenticity, and end-to-end confidentiality of the device data up to the Application Server.

The LoRaWAN protocol evolves to implement good practices in security. Furthermore, Firmware Update Over The Air (FUOTA) has been specified to provide the ability to patch any software bugs in the field once devices are deployed, some of which for 10+ years. This mechanism is an insurance against future software problems including security related ones.

References

[1] NIST, FIPS 197, "Advanced Encryption Standard (AES)", November 2001.
[2] Song, J.H., Poovendran, R., Lee, J., and T. Iwata, "The AES-CMAC Algorithm", RFC 4493, RFC Editor, June 2006.
[3] Dworkin, M., "Recommendation for Block Cipher Modes of Operation: Methods and Techniques", NIST Special Publication 800-38A, December 2001.
[4] LoRaWAN Link Layer Specification version 1.0.3, LoRa Alliance Technical Committee, July 2018.
[5] LoRaWAN Backend Interfaces Specification version 1.0, LoRa Alliance Technical Committee, October 2017.

Biography

Olivier Seller is technical fellow at Semtech, and vice-chair of LoRa Alliance Technical Committee. He is the editor of the LoRaWAN link layer specification since 2019. He is co-inventor of the LoRa technology, and co-founder

of Cycleo that developed LoRa initially, his focus was the LoRa modulation and implementation of receivers. Upon acquisition of Cycleo by Semtech in 2012, Olivier architected the LoRa localisation solution, designed the "find my stuff" radio capability of LoRa chips, and co-designed the LR-FHSS modulation.

Olivier studied in Ecole Polytechnique and Telecom Paris, then worked in the telecom and semiconductor industries with experiences in an 802.16 start-up, in Orange Labs, and in Cambridge Silicon Radio.