南京郵電大學
Nanjing University of Posts and Telecommunications

无线物联网安全-LoRa

汇报人：张宏
汇报日期：2025.5.13

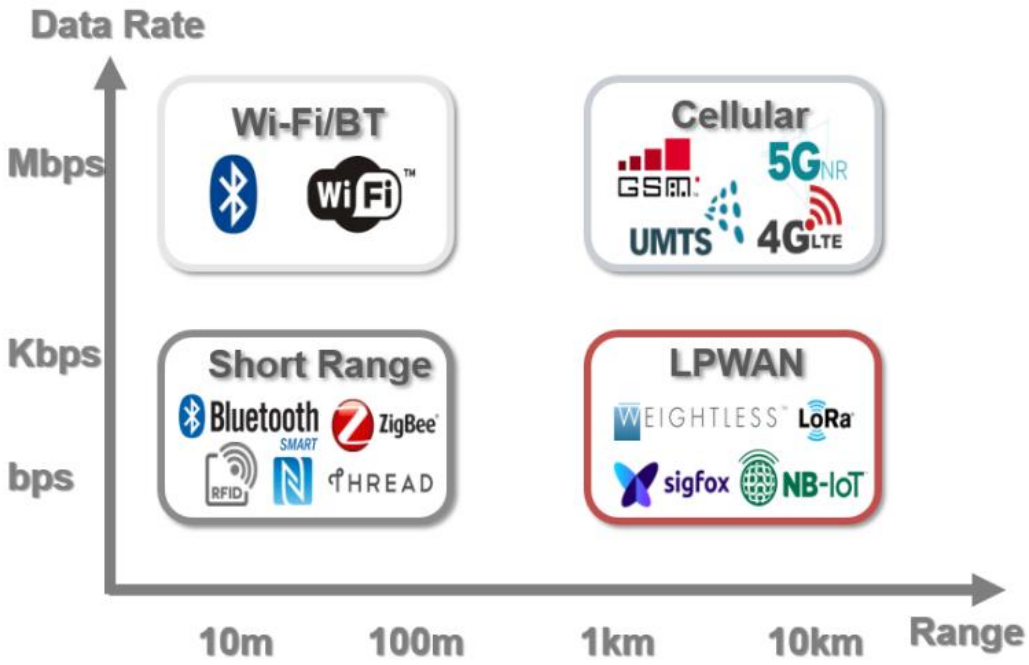
LoRa介绍

低功耗广域网LPWAN(Low Power Wide Area Network)

发展：为满足物联网连接需求，一批远距离、低功耗、低带宽的协议大量涌现。

分类：

- 基于蜂窝的组网技术：以NB-IoT为代表。（由电信运营商和设备商主导，基于既有的3G、4G长距离通信系统）
- 私有化组网技术：以LoRa为代表。依靠高灵敏物理层调制技术和低占空比通信模式。工作在免费频段ISM，允许用户通过自行部署网关构建私有化网络系统，成本较低。



LoRa介绍

LoRa: Long Range Communication

发展状况： 2013年，Semtech公司发布了SX127x系列芯片

2015年 3月,Semtech公司牵头成立了一个开放的非盈利组织——LoRa联盟(LoRa Alliance)，负责协议推广和应用等工作

2015年 6月,LoRa联盟提出了基于远距离调制技术(LoRa)的低功耗通信协议 LoRaWAN

到目前为止,已有数千家国际知名企业加入了 LoRa联盟,如华为、阿里巴巴、腾讯、IBM、Intel、Amazon等

2021年 12月, LoRaWAN正式被ITU(国际电信联盟)批 准 为 LPWAN国际标准

LoRa介绍

1. 工作频段

工作在ISM（Industrial Scientific Medical Band） 免费频段，不同地区频段不同，
这是一种各国开放给工业、科学及医学机构使用的频段。它们无须许可证及费用，只需要遵守一定的发射功率（一般低于1W），不要对其他频段造成干扰即可。

欧洲：868 MHz
北美：915 MHz
亚洲：433 MHz / 923 MHz

2. 长距离，通信距离能长，城市2-5km，郊区可达15km+

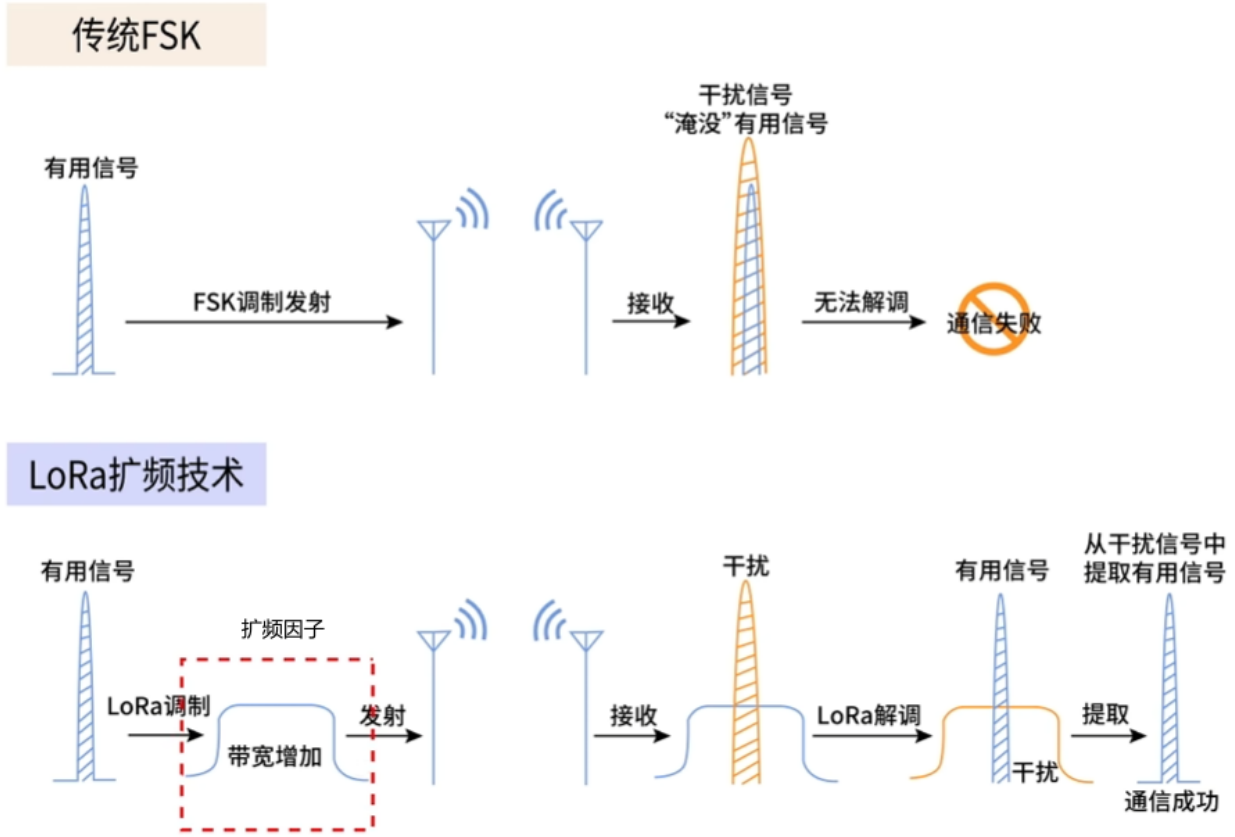
3. 低带宽，0.3-50kbps

4. 低功耗，电池寿命可达5-10年（如智能水表）

5. 调制方式，采用线性调频扩频技术CSS（Chirp Spread Spectrum）
有强抗干扰能力，比传统FSK技术覆盖更远，适合电池供电的传感器设备

LoRa介绍

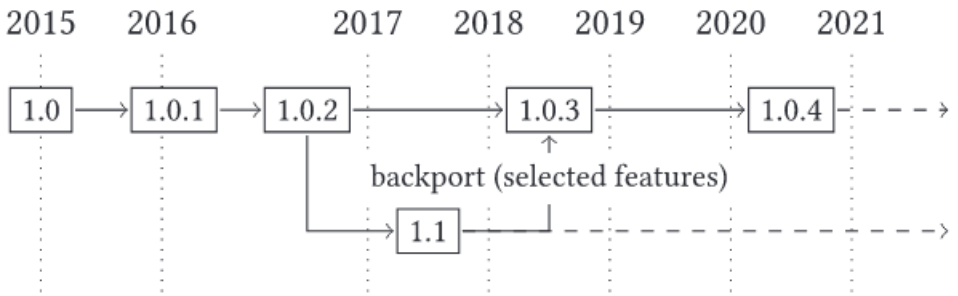
5. 调制方式，采用线性调频扩频技术CSS（Chirp Spread Spectrum）有强抗干扰能力，比传统FSK技术覆盖更远，适合电池供电的传感器设备。



LoRaWAN介绍

LoRaWAN

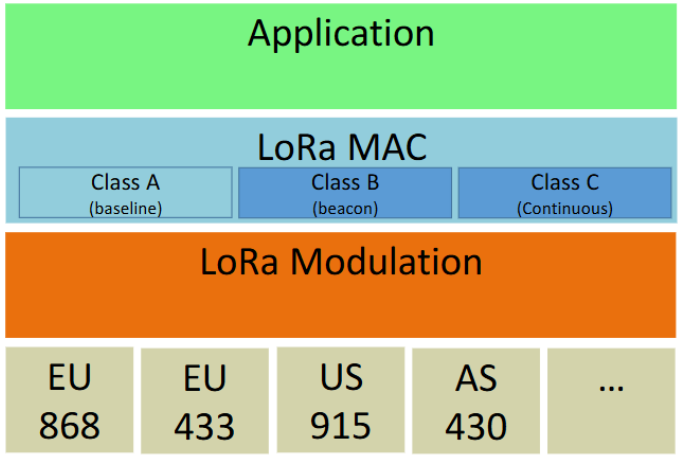
LoRaWAN是由LoRa联盟在LoRa物理层编码技术的基础上提出的MAC层协议，由LoRa联盟负责维护。LoRaWAN规范1.0版本于2015年6月发布。LoRaWAN协议主要规定了节点与网关、网关与服务器之间的连接规范，确定了LoRa网络的星型拓扑结构。受LoRa节点成本和能耗的限制，现有的LoRaWAN协议基本采用纯ALOHA机制，即节点在发送数据前不进行载波侦听，也就是没有使用CSMA/CA，而是随机选择时间进行发送。



LoRaWAN介绍

LoRa和LoRaWAN

	LoRa（物理层）	LoRaWAN（网络协议）
定位	无线调制技术（PHY层）	基于LoRa的通信协议（MAC层 + 部分网络层）
开发者	Semtech公司研发（芯片技术）	LoRa Alliance制定（开放标准）
功能	定义信号如何远距离传输传输（CSS调制）	定义设备如何组网、加密、管理数据流
依赖关系	可独立使用	必须依赖LoRa调制技术



单一的LoRa节点向LoRa网关发送数据主要采用CSS调制方法，很多LoRa节点向同一网关发送数据，就需要MAC协议来协调不同节点间的数据传输。

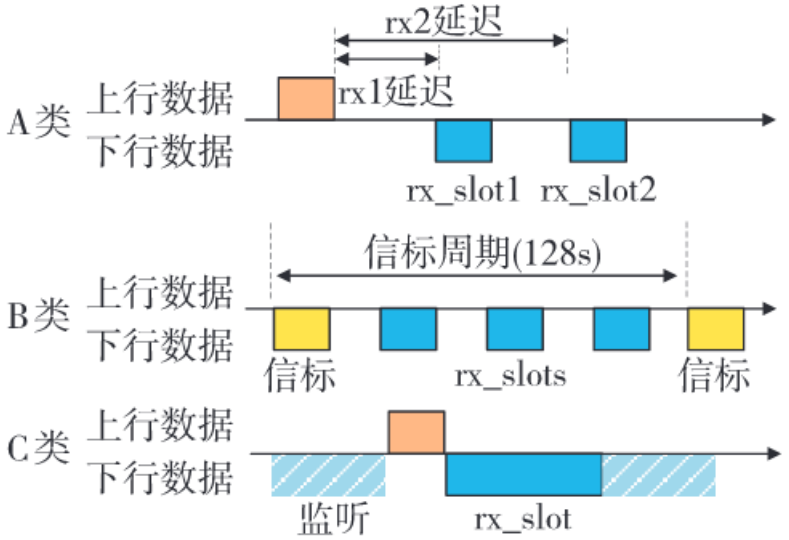
LoRaWAN介绍

工作模式

- Class A模式主要提供低功耗上行连接。此类 LoRa终端可以在任意时间发送上行链路消息,然后间隔一定时间打开 2个下行链路接收窗口,服务器可以在两个接收窗口打开期间进行响应,其余时间LoRa终端将进入睡眠状态,该类功耗是最低的。

- Class B模式提供节点与网关的周期性连接。在这个类中,LoRa终端双向通信,具有额外预定的接收槽。除了在A类模式下打开的两个接收窗口外,B类使用来自网关的同步信标激活一系列接收时隙;可以达到较大的下行通信速率。

- Class C模式提供节点与网关的持续性连接，该模式下终端设备始终处于唤醒状态,除了发送数据期间,其余时隙可以连续接收数据,所以适用于低延迟的通信。与其他类相比, 需要消耗较大的能量。



LoRa网络架构

1、End Nodes（终端节点）

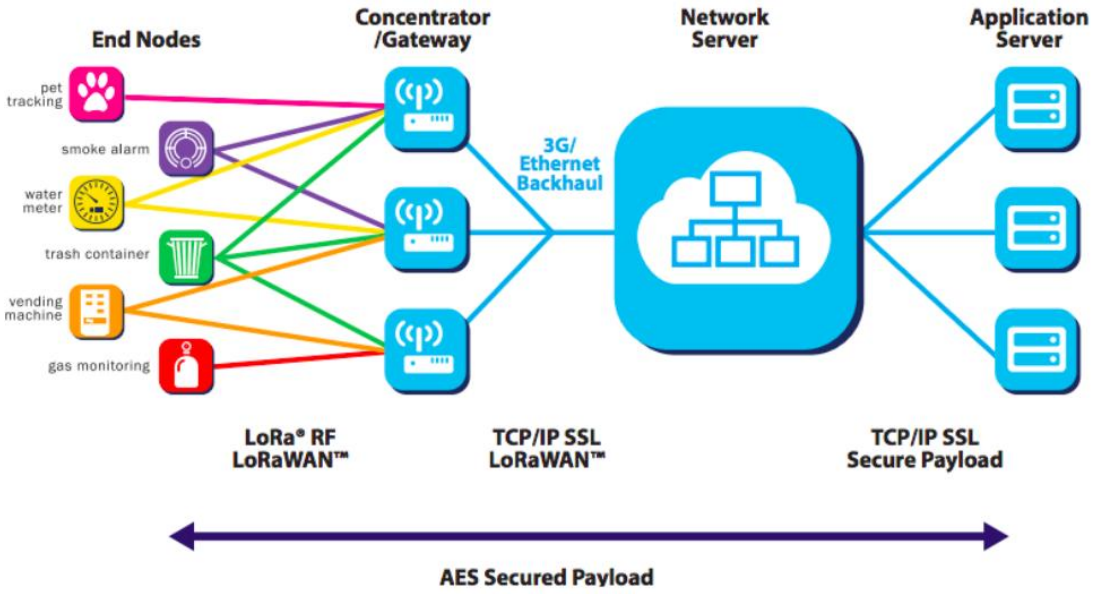
终端节点通常搭配传感器使用，从环境中采集各种信息，如烟雾、天气等。终端设备在每次发送数据包都需要随机切换信道，以便降低同频干扰和无线信号衰减。

2、Gateway（网关）

网关用于转发“终端节点”与“网络服务器”的之间的数据。网关与终端节点之间没有进行绑定，同一个节点的数据能被多个接收到，LoRa终端和 LoRa网关之间的连接采用LoRaWAN协议的无线通信。

3、Network Server（网络服务器）

网络服务器用于把终端节点产生的数据转发给对应的应用服务器，并提供对终端节点认证和授权。网关与网络服务器之间使用 TCP/IP 协议栈，采用透明传输。常见的协议有Packet Forwarder（现在被归类为 Legacy）、MQTT、CoAP、Protobuf。



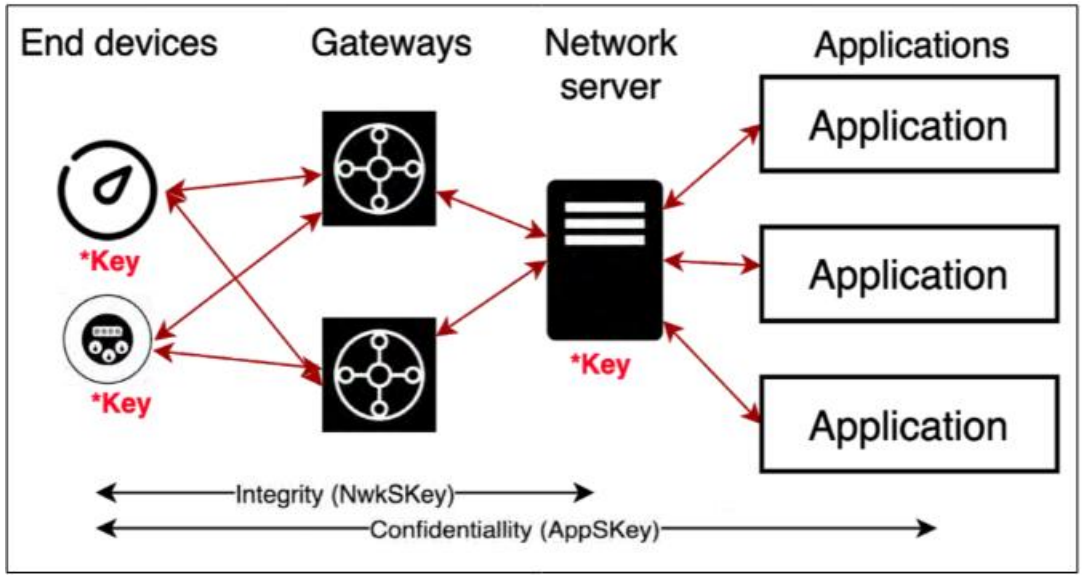
4、Applicaton Server（应用服务器）

应用服务器根据用户需要而设计，通常包括终端节点数据的展示（数据统计、异常数据告警）以及对节点的远程控制等。

LoRa协议栈与安全机制

LoRaWAN定义了两层安全机制：网络层安全与应用层安全。
网络层安全确保终端设备在网络中的真实性，为设备与网络服务器之间提供完整性保护。
应用层安全则通过端到端加密实现设备与应用服务器之间的数据保密性，防止第三方获取传输中的应用程序数据。

每个安全层都使用一个128位密钥：
网络会话密钥（NwkSKey）
应用会话密钥（AppSKey）
需要特别说明的是，网络服务器与应用服务器之间的数据完整性由服务提供商负责维护，而非协议本身定义。



LoRa协议栈与安全机制

一、入网方式

当一个 LoRa终端需要连接LoRa网络时,有两种方式可以激活入网连接到 LoRa网关:
空中激活 (over the air activation,OTAA)
个性化激活(activation by personalization,ABP)

ABP激活方式要求LoRa终端在入网前提前存储由NS提供的DevAddr和NwkSKey、AppSKey,设备就可以直接和网络服务器通信传输消息。

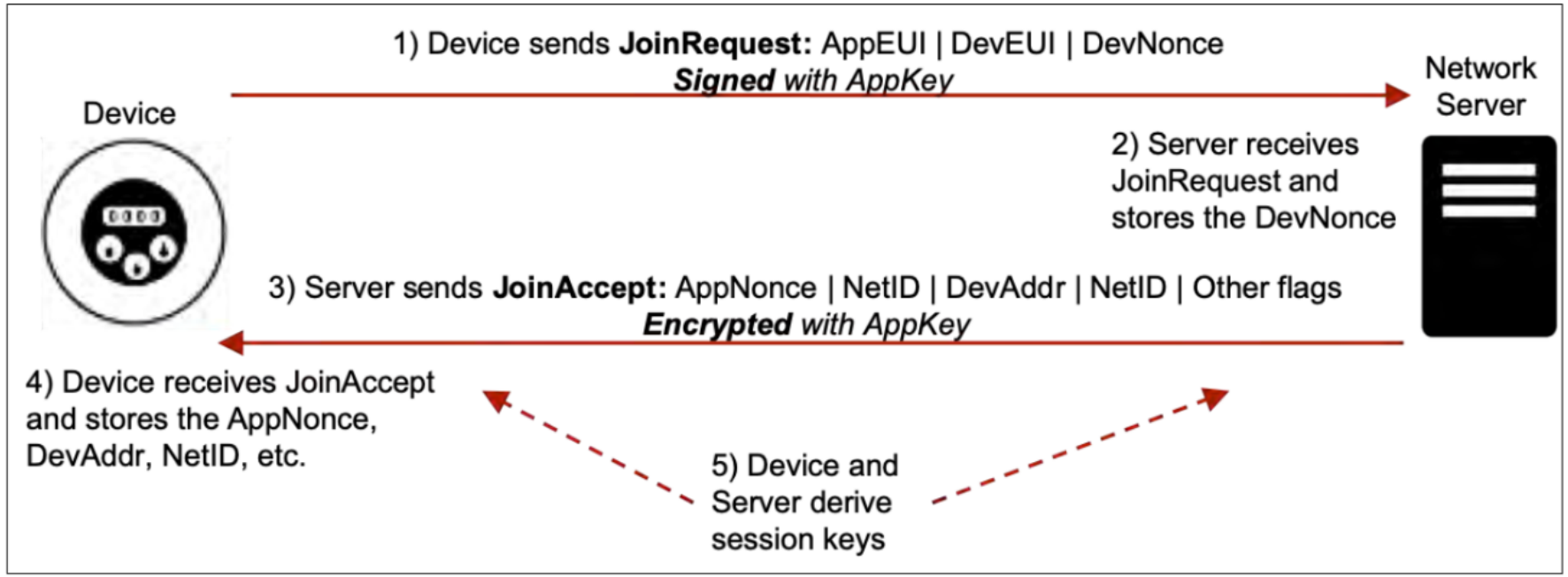
具体步骤如下:
a)首先配置 LoRa终端的三元组信息,将 DevAddr(32位)、NwkSKey、AppSKey烧录到设备中,并对必要的通信参数进行配置, 如数据速率、信道设置等。
b)NS需要配置与 LoRa终端对应的网络会话信息,包括设备地址和会话密钥。
c)设备加入到网络时,在本地连接后即可正常使用。LoRa终端使用预配置的 DevAddr、NwkSKey 和 AppSKey信息加入网络,而无须交互验证过程,通过在每个上行数据报文中包含固定的设备地址和会话密钥来进行通信。

LoRa协议栈与安全机制

OTAA

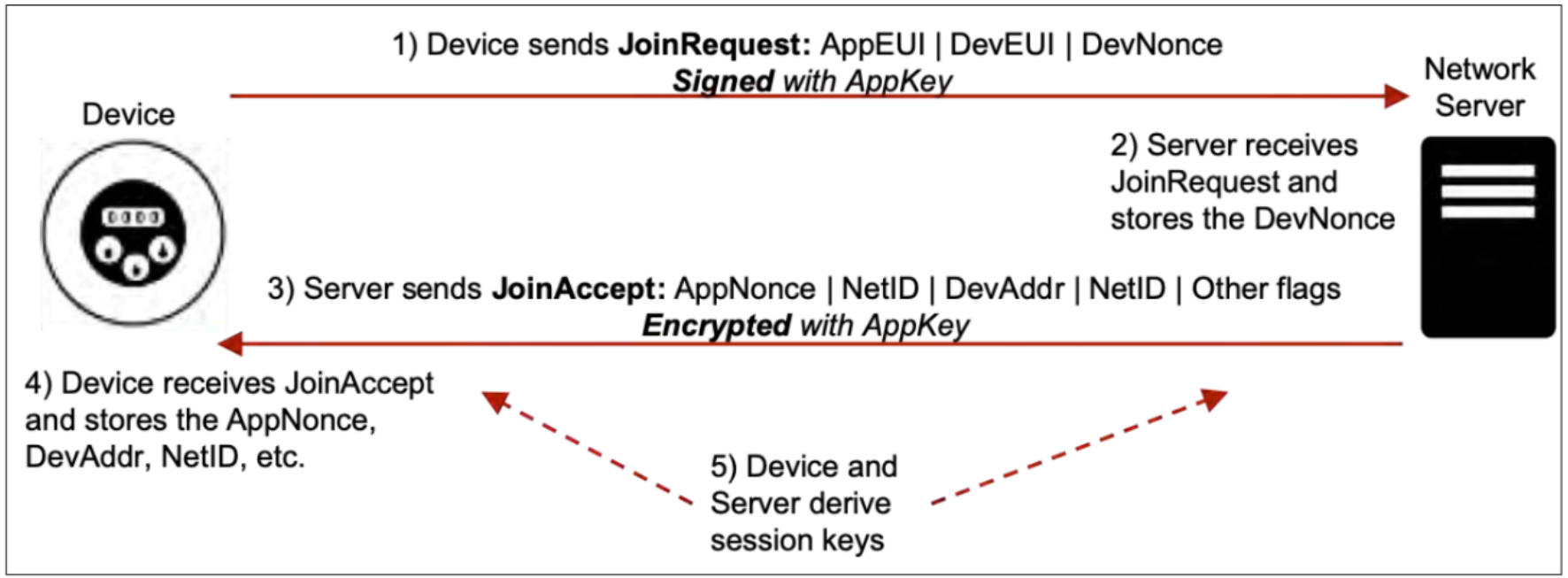
OTAA激活方式采用请求确认的模式入网

AppKey：AES-128 根密钥，终端和NS具备



LoRa协议栈与安全机制

OTAA



应用标识符(AppEUI)：这是IEEE EUI64地址空间中的全局应用ID，用于唯一标识能够处理入网请求的实体。

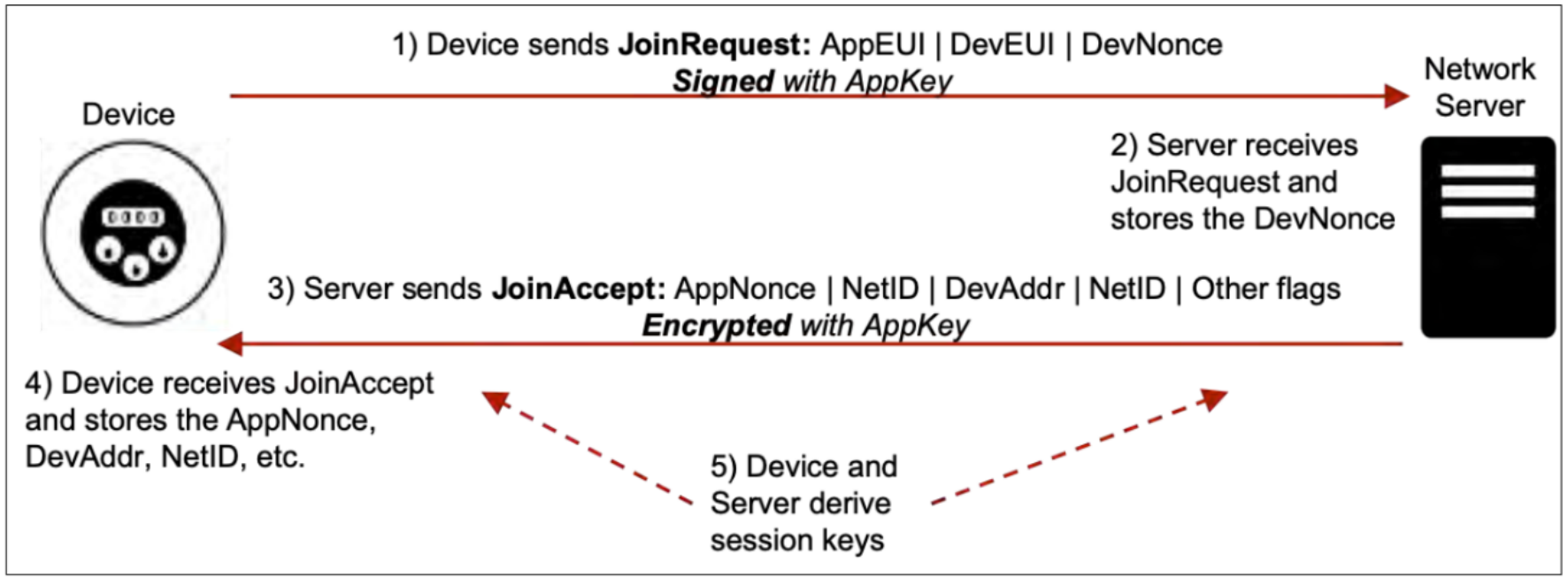
终端设备标识符(DevEUI)：IEEE EUI64地址空间中的全局终端设备ID，用于唯一标识终端设备。然而实际部署中通常不遵循这种唯一性要求，该值极易被伪造。

设备随机数(DevNonce)：2字节，防止重放攻击

MIC：签名，AES-CMAC计算

LoRa协议栈与安全机制

OTAA



终端设备地址(DevAddr)：这是当前网络中终端设备的32位标识符。DevAddr类似于设备的会话ID，通常会随着每次会话而改变，具体取决于所使用的网络服务器的实现方式。在ABP设备中，该字段在设备的整个生命周期内保持不变。

网络标识符(NetID)：这是同一LoRaWAN网络中所有设备共享的值。

应用随机数(AppNonce)：3字节，防止重放攻击

LoRa协议栈与安全机制

OTAA

当设备与网络服务器完成这些消息交换后，双方即可利用交换的数值生成会话密钥。
其计算过程如下：

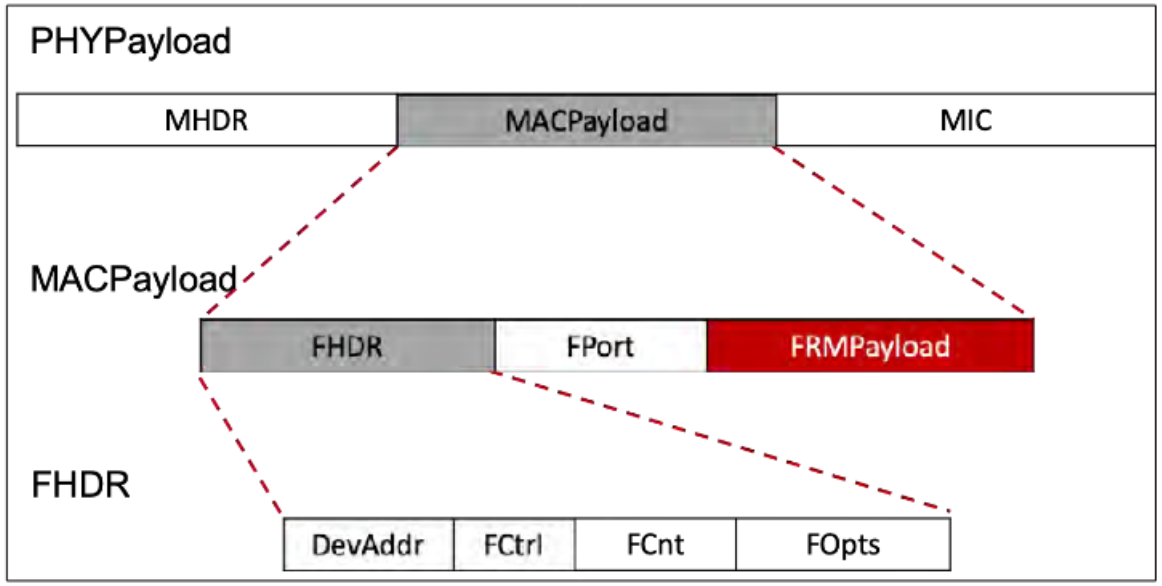
$$\text{NwkSKey} = \text{aes}_{128}\text{encrypt}(\text{AppKey}, \texttt{0x01}|\text{AppNonce}|\text{NetID}|\text{DevNonce}|\text{pad}_{16}),$$

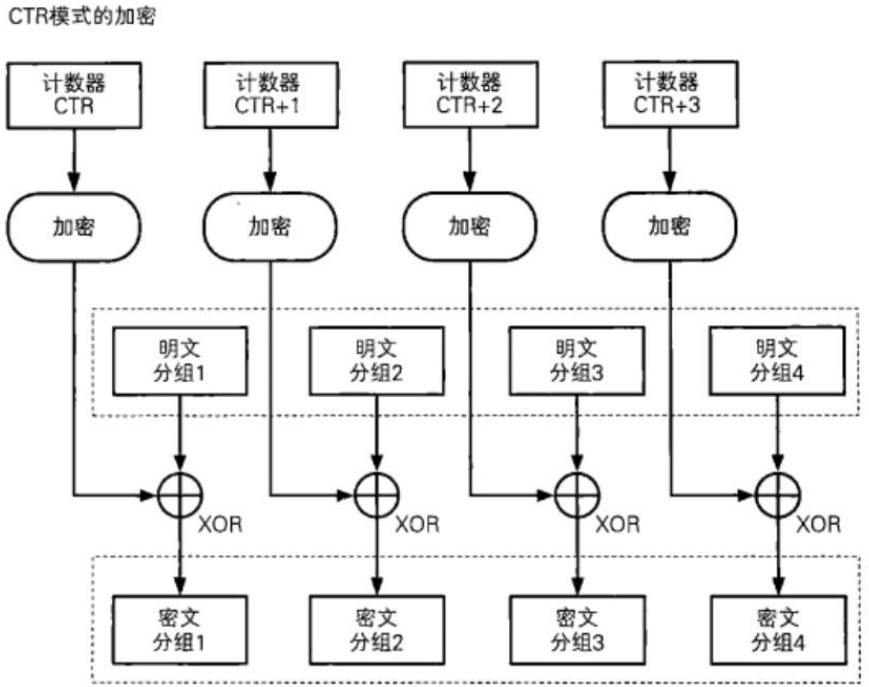
$$\text{AppSKey} = \text{aes}_{128}\text{encrypt}(\text{AppKey}, \texttt{0x02}|\text{AppNonce}|\text{NetID}|\text{DevNonce}|\text{pad}_{16}).$$

LoRa协议栈与安全机制

二、 LoRaWAN 中的机密性

在LoRaWAN协议中，消息的机密性是通过仅加密数据负载（FRMPayload，即设备与应用服务器之间交换的数据）来实现的。MAC头（MHDR）、帧头（FHDR）、可选端口字段(Fport)及其数据字段，以及消息完整性码（MIC），均以明文形式传输。用于加密的密钥是AppSKey，使用AES-CTR模式。





LoRa协议栈与安全机制

三、 LoRaWAN 中的完整性

用于保障消息完整性的密钥是NwkSKey。为保护LoRaWAN消息的完整性，协议采用消息完整性码（MIC）机制。该MIC基于整个LoRaWAN消息生成，并以4字节长度附加在每条消息末尾。MIC由CMAC运算结果的前4个字节构成。

CMAC运算需作用于完整的LoRaWAN消息（MHDR | MACPayload），且必须在加密操作完成后执行。

B0= 0x49 | 4 * 0x00 | Dir | DevAddr| FCntUp or FCntDown | 0x00 | len(MHDR | MACPayload)

cmac = aes128_cmac(NwkSKey, B0 | MHDR | MACPayload)

MIC = cmac[0..3]

B0 是一个由固定字节池和可变字节组成的字节数组，其中包含：

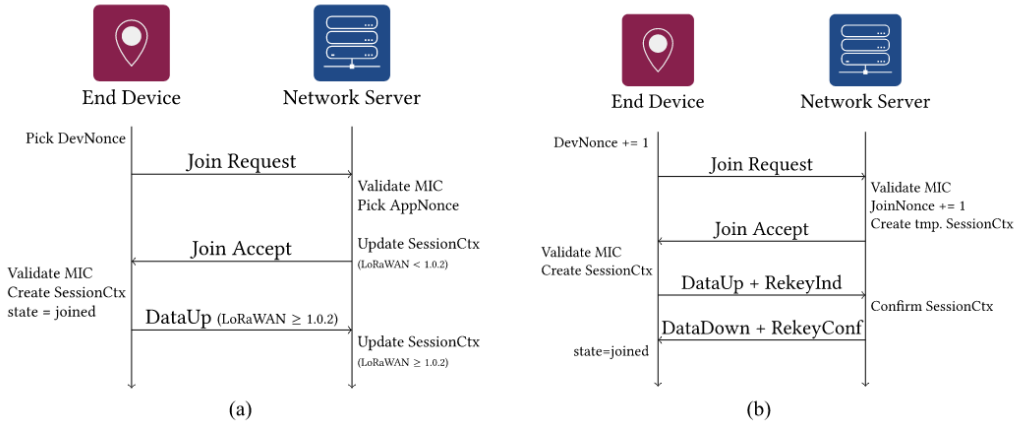
- Dir：表示LoRaWAN消息方向的标志位（0代表上行帧，1代表下行帧）
- DevAddr：设备地址
- FCnt：帧计数器值(16位的上行链路和下行链路计数器，防重放)

LoRaWAN v1.1 版本的安全增强措施

- 新增Join Server：在LoRaWAN架构中增设了Join Server（入网服务器），由其负责派生会话密钥，是一个受信任的第三方。这一设计确保网络服务器始终不接触AppSKey。
- 双根密钥机制：从单一根密钥改为使用两个根密钥（AppKey和NwkKey）。传统LoRaWAN 1.0仅使用单一AppKey，存在以下安全隐患：网络层与应用层安全耦合（如NS可推导AppSKey），无法支持多用户场景的密钥隔离，漫游场景下密钥管理混乱。
- 独立计数器与位宽升级：为网络层和应用层实现独立的32位计数器（旧版为16位）。

LoRaWAN v1.1 版本的安全增强措施

V1.1下的OTAA



1. 随机数改为计数器：

设备随机数（DevNonce）和入网随机数（JoinNonce，原称AppNonce）改用计数器生成，无需记录历史值（旧版需跟踪所有已用随机数，对小型传感器节点尤其不现实）。

因随机数仅用于保证时效性（freshness），而非引入随机性。

2. 会话上下文切换双向确认：

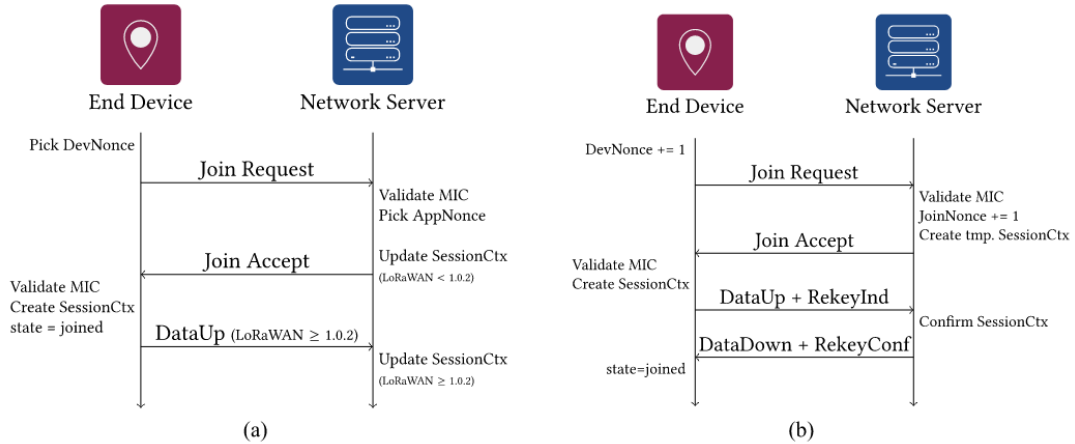
通过重配密钥指示（RekeyInd）和重配密钥确认（RekeyConf）这一对MAC命令实现。

终端设备需在上行消息中用RekeyInd命令确认收到Join Accept，以便网络服务器验证激活成功。

NS通过下行消息的RekeyConf命令回复确认，确保双方均认可新会话上下文生效。若ED未收到RekeyConf，则重新发起Join Request。

LoRaWAN v1.1 版本的安全增强措施

V1.1下的OTAA



3. 对于OTA（空中激活）设备，从NwkKey根密钥派生出两个特定生命周期的密钥：

JsIntKey：用于计算Rejoin-Request类型消息和Join-Accept应答的消息完整性码（MIC）

JSEncKey：用于加密由Rejoin-Request触发的Join-Accept消息

$$JSIntKey = aes128\_encrypt(NwkKey, 0x06 \mid DevEUI \mid pad_{16})$$

$$JSEncKey = aes128\_encrypt(NwkKey, 0x05 \mid DevEUI \mid pad_{16})$$

LoRaWAN v1.1 版本的安全增强措施

派生密钥

$$\text{FNwkSIntKey} = \text{aes}_{128}\text{encrypt}(\text{NwkKey}, \texttt{0x01}|\text{JoinNonce}|\text{JoinEUI}|\text{DevNonce}|\text{pad}_{16}),$$

$$\text{SNwkSIntKey} = \text{aes}_{128}\text{encrypt}(\text{NwkKey}, \texttt{0x03}|\text{JoinNonce}|\text{JoinEUI}|\text{DevNonce}|\text{pad}_{16}),$$

$$\text{NwkSEncKey} = \text{aes}_{128}\text{encrypt}(\text{NwkKey}, \texttt{0x04}|\text{JoinNonce}|\text{JoinEUI}|\text{DevNonce}|\text{pad}_{16}),$$

$$\text{AppSKey} = \text{aes}_{128}\text{encrypt}(\text{AppKey}, \texttt{0x02}|\text{JoinNonce}|\text{JoinEUI}|\text{DevNonce}|\text{pad}_{16}).$$

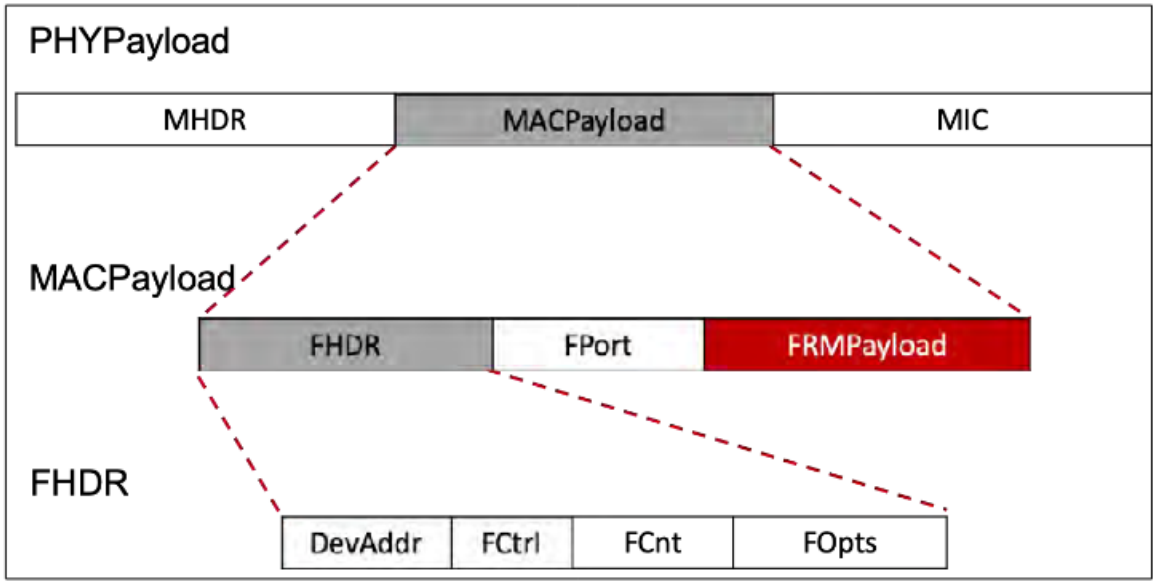
为兼容不支持双根密钥的LoRaWAN 1.0网络服务器：
（该兼容条件通过Join-accept消息的DLsetting字段第7位“OptNeg”置零来通知终端设备）

- 使用NwkKey派生AppSKey和FNwkSIntKey会话密钥
- 将SNwkSIntKey和NwkSEncKey设置为与FNwkSIntKey相同

LoRaWAN v1.1 版本的安全增强措施

NwkSEncKey与AppSKey用于加密

帧头（FHDR）包含终端设备的短地址（DevAddr，4字节）、帧控制字节（FCtrl，1字节）、2字节的帧计数器（FCnt），以及用于传输MAC命令的帧选项字段（FOpts，最多15字节）。



LoRaWAN v1.1 版本的安全增强措施

NwkSEncKey与AppSKey用于加密

帧头（FHDR）包含终端设备的短地址（DevAddr，4字节）、帧控制字节（FCtrl，1字节）、2字节的帧计数器（FCnt），以及用于传输MAC命令的帧选项字段（FOpts，最多15字节）。

使用的 key K 取决于数据消息的 FPort：

0	网络层管理消息	MAC命令	NwkSEncKey
1~25	应用层业务消息	传感器数据、设备控制指令	AppSKey

LoRaWAN v1.1 版本的安全增强措施

NwkSEncKey与AppSKey用于加密

Size (bytes)	1	4	1	4	4	1	1
A_i	0x01	4 x 0x00	Dir	DevAddr	FCntUp or NFCntDwn or AFCntDnw	0x00	*i*

Blocks Ai for i = 1..k with k =748ceil(len(pld) / 16)

1.Si = aes128_encrypt(K, Ai) for i = 1..k

S = S1 | S2 | .. | Sk

3. PRMPload={(pld | pad16) xor S}[:len(pld)]

- 数据保密性：基于DevAddr（设备地址）和FCnt（帧计数器）和i构造唯一加密块Ai，通过AES-ECB生成密钥流，再与明文异或实现流加密，确保即使重复明文也输出不同密文。
- 防重放攻击：FCnt的严格递增性（上行FCntUp/下行FCntDwn）和块序号i的引入，使每条消息甚至同一消息内的分块均产生唯一密钥流，有效阻断数据重放。
- 密钥隔离：根据FPort值选择NwkSEncKey（网络层）或AppSKey（应用层），实现网络控制与应用数据的权限分离，避免单点泄露导致全局沦陷。

LoRaWAN v1.1 版本的安全增强措施

SNwkSIntKe与FNwkSIntKey用于完整性保护

the block B_0 is defined as follows:

Size (bytes)	1	4	1	4	4	1	1
B_0	0x49	0x0000	Dir = 0x00	DevAddr	FCntUp	0x00	len(*msg*)

the block B_1 is defined as follows:

Size (bytes)	1	2	1	1	1	4	4	1	1
B_1	0x49	ConfFCnt	TxDr	TxCh	Dir = 0x00	DevAddr	FCntUp	0x00	len(*msg*)

TxDr：上行帧传输使用的数据速率
TxCh：上行帧传输使用的信道索引
ConfFCnt：当上行帧的ACK位置1时，ConfFCnt值为被确认下行帧的帧计数器值对2^16取模。其他情况下ConfFCnt=0x0000

$$cmacS = aes128\_cmac(\mathbf{SNwkSIntKey}, B_1 \mid msg)$$
$$cmacF = aes128\_cmac(\mathbf{FNwkSIntKey}, B_0 \mid msg)$$

- LoRaWAN1.0 网络服务器：
  MIC = cmacF[0..3]
- LoRaWAN1.1 网络服务器：
  MIC = cmacS[0..1] | cmacF[0..1]
  SNwkSIntKey生成部分：供服务网络服务器（SNS）验证。
  FNwkSIntKey生成部分：供转发网络（FNS）部分验证

参考资料

[1]Cerrudo, C., Martinez Fayo, E., & Sequeira, M. (2020). LoRaWAN networks susceptible to hacking: Common cyber security problems, how to detect and prevent them [White paper]. IOActive.
[2]Hessel, F., Almon, L., & Hollick, M. (2023). LoRaWAN security: An evolvable survey on vulnerabilities, attacks and their systematic mitigation. ACM Transactions on Sensor Networks, 18(4), Article 70. https://doi.org/10.1145/3561973
[3]LoRa Alliance, Inc. (2015). LoRaWAN specification [Technical standard]. LoRa Alliance.
[4]LoRa Alliance. (2017). LoRaWAN™ 1.1 specification [Technical report]. LoRa Alliance, Inc. https://www.lora-alliance.org
[5]刘亚荣, 吴雪涛, 谢晓兰. (2024). 对LoRa网络的攻击与防御技术综述. 计算机应用研究, 41(11), 1-8
[6]Gemalto, Actility, & Semtech. (2017). LoRaWAN security: Full end-to-end encryption for IoT application providers [White paper]. LoRa Alliance.

谢谢！

计算机学院 软件学院
网络空间安全学院
School of Computer Science