## Sécurité et activation

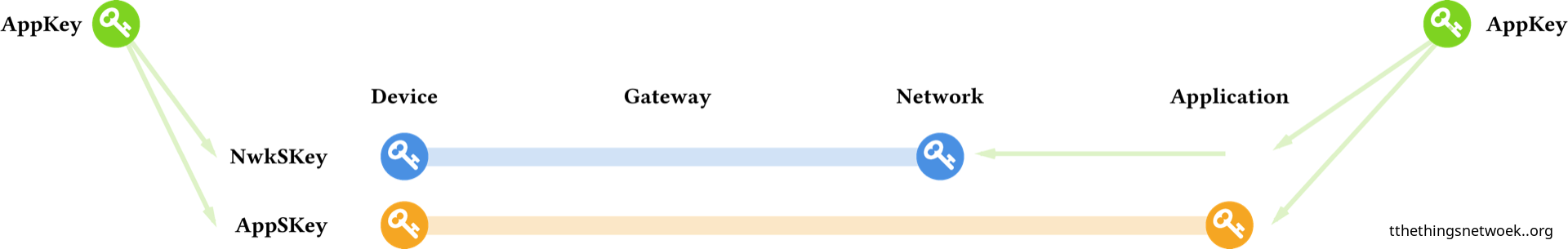
### Sécurité dans un réseau LoRaWAN

En matière de sécurité, la norme LoRaWAN spécifie trois clefs AES-128 :

**NwkSKey**, clef de session réseau, qui est utilisée lors des échanges entre le end-device et le coeur de réseau. Elle assure l’authenticité des devices en calculant et vérifiant un *Message Integrity Code*, MIC, à partir du *Header* et du *Payload* chiffré.

**AppSKey**, clef de session applicative, spécifique à un end-device est utilisée pour chiffrer et déchiffrer le payload.

**AppKey**, clef applicative connue seulement par l’application et le end-device et qui permet de déduire les deux clefs précédentes.



La clé de session applicative est connue uniquement par le fournisseur d’application. Il est donc impossible pour un tiers, y compris l’opérateur, de consulter les données.

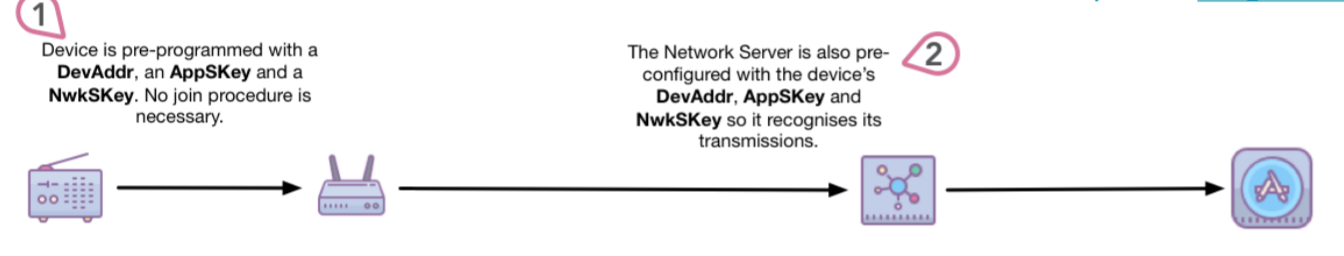
On utilise également un *frame counter* pour se protéger contre les *attaques par rejeu*, qui consistent à répéter une transmission malicieusement interceptée. Le compteur est incrémenté à chaque transmission. La passerelle et les end-devices rejettent les transmissions dont la valeur de compteur est inférieure à celle attendue.

* Pour faire partie d’un réseau LoRaWAN, chaque device doit obtenir les deux clefs de session. Cette étape d’activation peut s’effectuer de deux manières : *Over-The-Air Activation* (OTAA) ou *Activation By Personalization* (APB).

Activation By Personalization

En ABP, les clefs de session *NwkSKey*, *AppSKey* et l’adresse du device *DevAddr* sont directement stockées sur le end-device. Il est possible alors de passer outre la procédure de join request, join accept.

*NwkSKey* et *AppSKey* doivent être uniques !



## Over-The-Air Activation

L'activation par liaison radio consiste en une « demande de jointure » et une « acceptation de jointure » entre un terminal et un serveur.

## 

### JOIN REQUEST

Lorsque le processus d'activation démarre, une clé AppKey doit être affectée aux deux périphériques

et le serveur de réseau. L’appareil final devrait connaître AppEUI etDevEUI, et devrait être capable de générer son DevNonce.

AppKey est une clé racine AES-128 spécifiée pour un périphérique final.

AppEUI est un identifiant d'une application, tandis que DevEUI est un unique identifiant d'un terminal. DevNonce est une séquence de nombres aléatoires et elle est générée

en émettant une séquence de mesures d'indicateur d'intensité du signal reçu (RSSI)

et il est supposé être idéalement aléatoire .

Lorsque la procédure de jointure commence, elle envoie d'abord une demande de jointure par liaison radio

Le message "Demande de jointure" n'est pas chiffré, mais il utilise AppKey pour générer le message

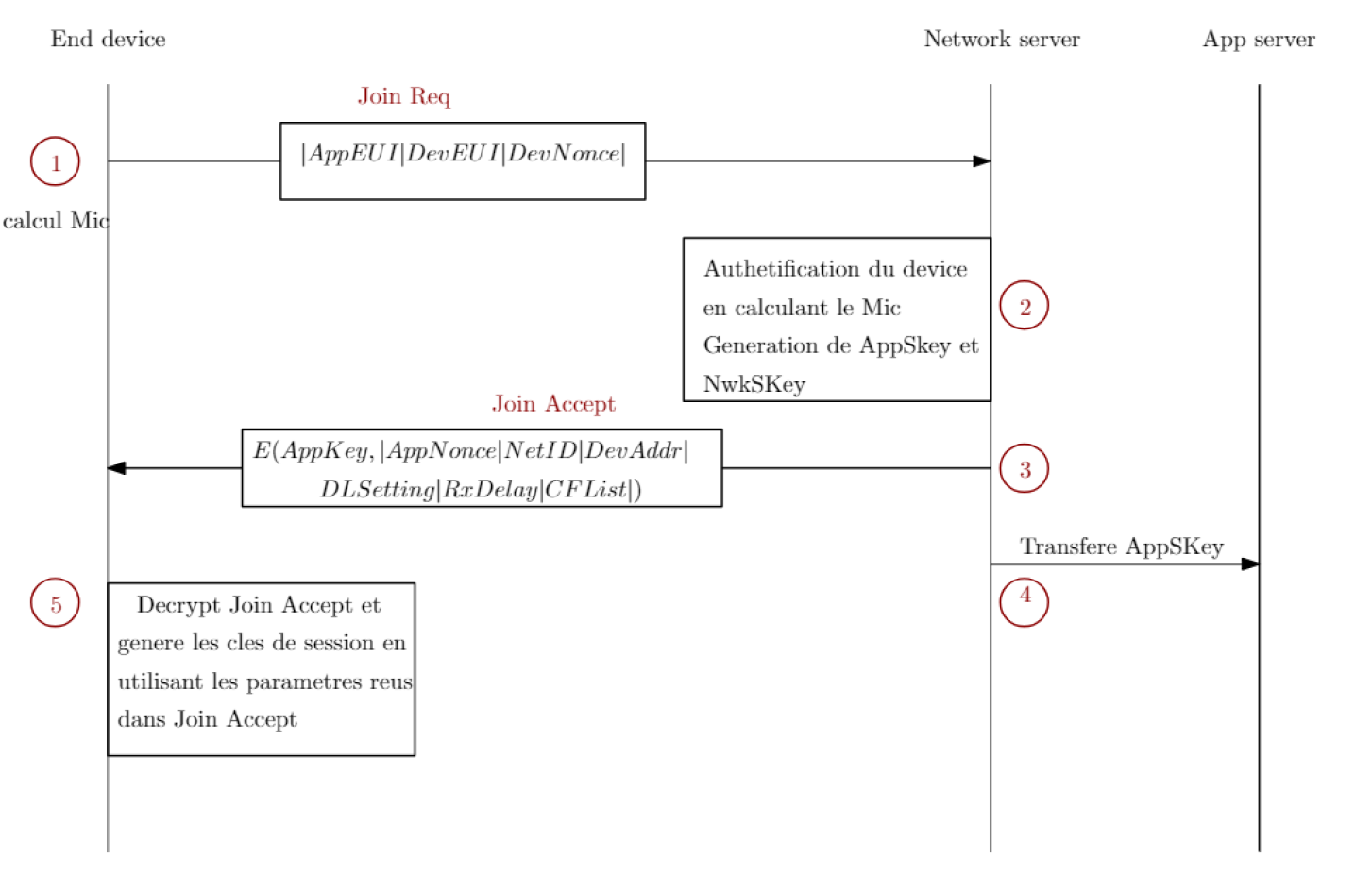
Code d'intégrité (MIC) pour assurer l'intégrité du message.

### JOIN ACCEPT

Le réseau du serveur envoie un message de acceptation de joint à chaque périphérique contenant AppNonce (valeur aléatoire générée par le serveur), NetID (identifiant de réseau), DevAddr (identifiant du périphérique), DLSettings (configuration du périphérique), RXDelay ( Le délai entre la transmission et la réception) et CFList qui est facultatif (pour les fréquences de canal). Le message Acceptation de la jointure est chiffré à l'aide d'AppKey au lieu du message de demande de jointure, ce qui peut entraîner de graves problèmes de sécurité. Tout le monde peut connaître l'ID des périphériques et l'ID de l'application à laquelle le périphérique est dédié. Vous pouvez également connaître le DevNonce inclus dans la génération des clés.

Puis le serveur transfère l'AppSkey sur le serveur d'applications pour l'utiliser dans le déchiffrement des messages.

Après avoir reçu le message de confirmation d'acceptation, le dispositif le déchiffre et génère les clés de session en utilisant les paramètres inclus dans le message d'acceptation de jointure.



### CARACTÉRISTIQUES DE OTAA

OTAA fournit des mécanismes de sécurité.

Tout d'abord, il utilise des paramètres uniques. Dans OTAA, AppKey, DevEUI, AppEUI, AppNonce et

DevNonce devrait tous être unique entre les terminaux. Dans ce cas, compromettre un

périphérique final ne signifie pas compromettre l'ensemble du réseau.

Deuxièmement, il existe un tampon pour DevNonce pour empêcher les attaques par rejeu. Chaque fois une nouvelle

demande de jointure est reçue, le serveur devrait vérifier le tampon pour voir si le nonce a été

utilisé avant S'il a été utilisé, l'appareil final n'est pas autorisé à rejoindre le réseau.

Dans ce cas, la copie d'une demande de jointure et sa relecture ne sont pas possibles.

