Introducción a Seguridad para IoT

Tecnologías para la Internet de las Cosas

NEON
Network of Competence on Internet of Things

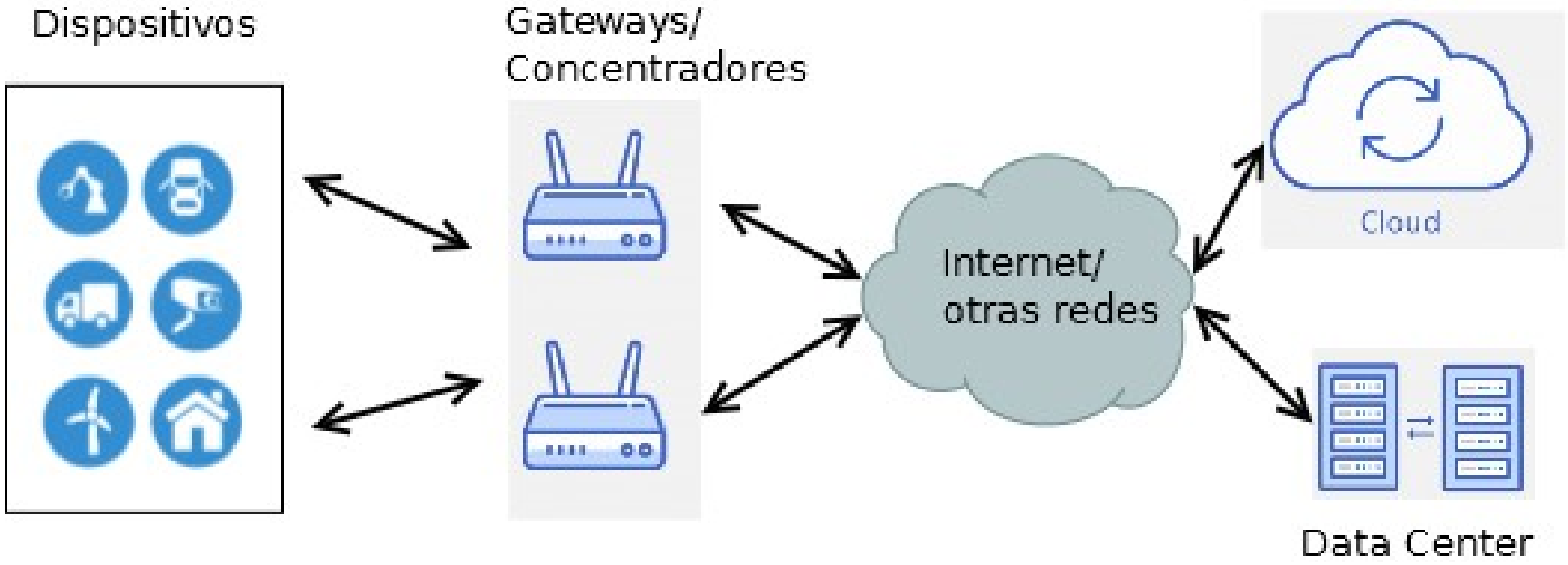
FACULTAD DE INGENIERIA

UNIVERSIDAD DE LA REPUBLICA
URUGUAY

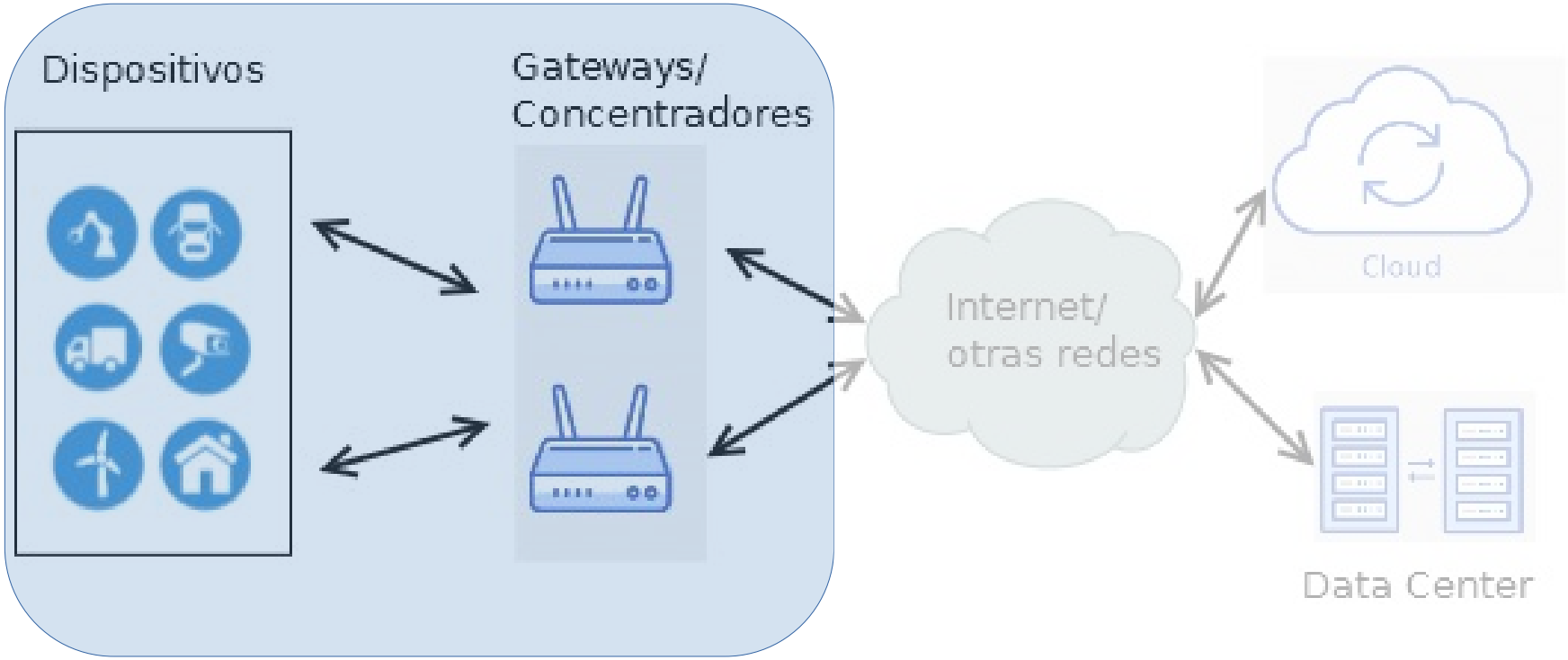
Introducción

- IoT: Conjunto de sistemas informáticos interconectados
  - Seguridad comparte la mayoría de los objetivos con los sistemas tradicionales
  - Nuevas restricciones (poder de cómputo, longevidad, ambiente)
  - Nuevos modos de ataque
  - Algunos objetivos nuevos
- Muchas herramientas en común

Arquitectura de referencia



Arquitectura de referencia



- Nuevos desafíos

Principios básicos

- Confidencialidad
- Integridad
- Freshness (evitar replay)
- Autenticación
- Autorización
- Disponibilidad
- **Eficiencia**
- **Autonomía**

Confidencialidad

- Evitar que un atacante pueda obtener información del tráfico
- Solución tradicional: cifrado
  - Transformación matemática del mensaje para hacerlo indescifrable para quien no posea la clave
  - Importante la elección y seguridad de la clave
  - Importante quienes conocen la clave
- Nuevas restricciones: Capacidad de los dispositivos
  - Cada vez menos restrictivo

Cifrados simétricos

- Operación de cifrado y descifrado operan usando la misma clave
- Cifrados en bloque o stream
  - Según si operan sobre un bloque de largo fijo, o individualmente sobre cada bit/byte
- Ejemplos de algoritmos: Skipjack, RC5, RC6, AES
- Modo de operación (cifrados en bloque): cómo se utiliza el algoritmo para cifrar un mensaje/sucesión de mensajes
  - CBC, CFB, OFB, **CTR**

Cifrados asimétricos
Clave pública

- Claves de cifrado y descifrado distintas, relacionadas
  - No se puede deducir una clave de la otra
- Se hace pública una de las claves
- Utilizado para firma, autenticación, intercambio de claves simétricas
  - Mucho más caros computacionalmente
- Ejemplo RSA, ECC (logaritmos sobre curvas elípticas)
- Algoritmos: Diffie Hellman (DH), DSA (firmas), ECDH, ECDSA, etc.
- Usualmente usados en conjunto con un cifrado simétrico

Autenticación

- Verificación de identidad de la contraparte
  - Al ingreso a la red (capa de red)
  - A nivel de aplicación
  - Algoritmos simétricos o asimétricos
  - Una vez autenticado, autorización (qué le permito hacer)
- Verificación de la autenticidad de los mensajes
  - Message Authentication Codes (MAC)
  - Clave simétrica

Integridad

- Algoritmos para verificar integridad
  - Hashes
  - MAC (combinan integridad con autenticación del mensaje)
- Combinado con nonces o timestamps, evita replay

Protocolos

- Las primitivas y algoritmos criptográficos se utilizan en **protocolos**
  - Tan importantes como las primitivas
- Ejemplos:
  - TLS: Transport Layer Security (“HTTPS”)
  - DTLS: similar a TLS, para datagramas (UDP)
  - IPSec
  - Intercambio de claves de Diffie Hellman
  - Etc
- Esfuerzos por definir protocolos más “livianos” para IoT

Amenazas a IoT. Acceso físico

- Acceso físico
  - Muchos dispositivos en muchos casos fácilmente accesibles en ambientes no controlados
- Robo/modificación/reprogramación?
- Ingeniería inversa
  - Robo de claves. Robo de propiedad intelectual.
- Nodos maliciosos
  - Datos de medidas adulterados afectando decisiones
  - Ataques a las comunicaciones
  - Negación de servicio
  - Ataques a terceros

Protección contra ataque físico

- Difícil en muchas situaciones
- Contra destrucción/desconexión:
  - Aplicación resistente a la pérdida de parte de los dispositivos
  - Detección de dispositivos “faltantes”
- Dispositivos “tamper-proof”
  - Encarecen solución
  - Pueden evitar robo de información sensible
  - ¿en casos de infraestructura crítica?
- Asegurarse que el compromiso de un dispositivo no pone en riesgo toda la red

Amenazas a IoT. Negación de servicio

- Ataques que de alguna manera impiden el correcto funcionamiento del sistema
- Comunes y fáciles de implementar
- En capa física:
  - Jamming: señales en el rango de frecuencias que impiden el normal funcionamiento
    - Codificación adecuada (ej. Spread Spectrum) puede mitigar

Negación de servicio (cont)

- Capa de enlace/red:
  - Introducir errores para forzar retransmisiones, consumo y uso del canal
  - Encaminamiento erróneo forzado.
    - Blackholing
    - Consumo de energía y ancho de banda
  - Sybil Attacks: Nodo malicioso que toma múltiples identidades
    - Información maliciosa, ruteo, etc
- Capa de transporte/aplicación:
  - Consumo de recursos/puertos/etc.
  - Evita el uso legítimo

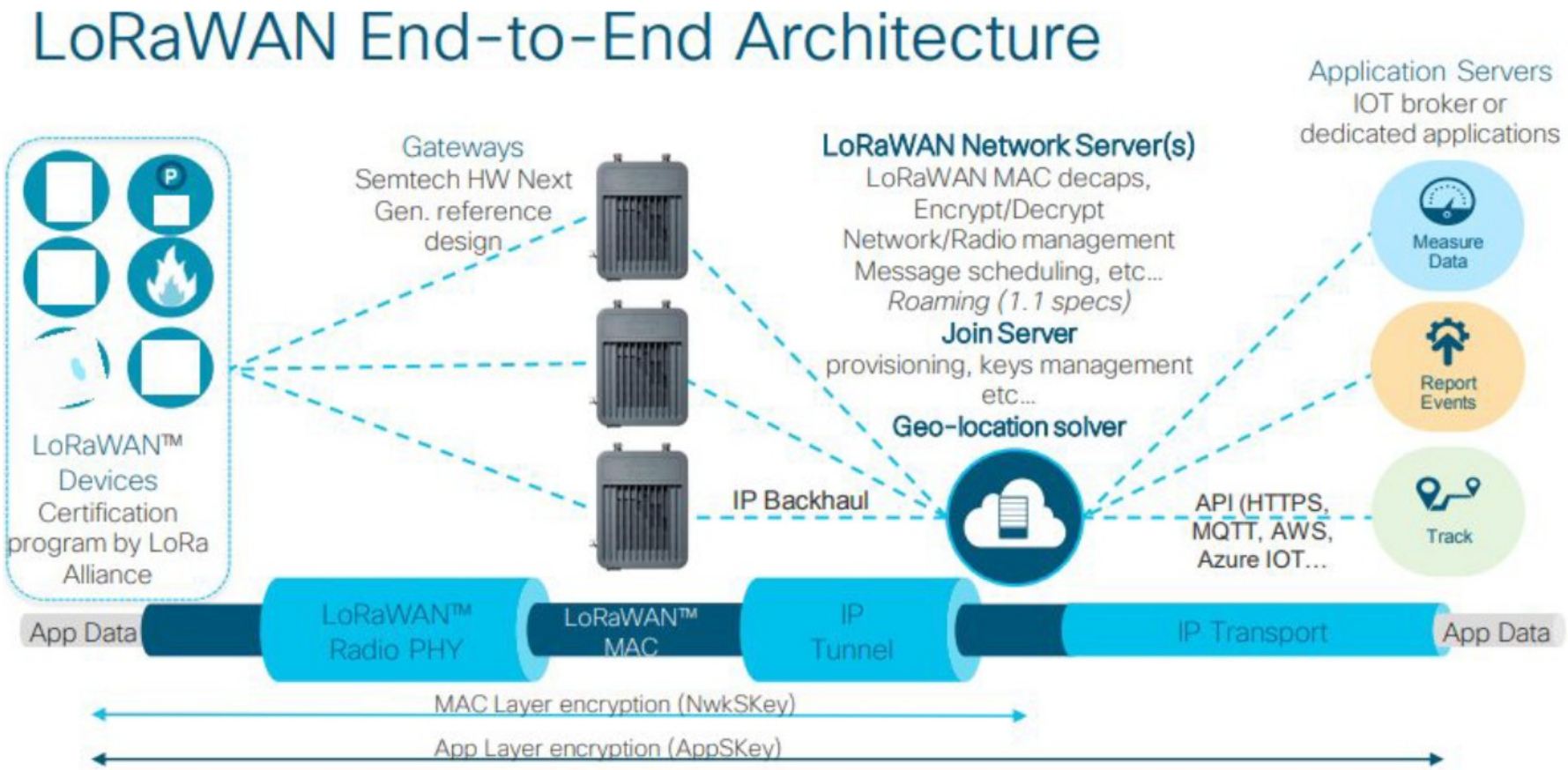
Amenazas a IoT. Espionaje

- Dependiendo de la aplicación, mucha información interesante para atacantes
- Captura directo del medio
- Captura en elemento comprometido
- Mejor defensa: criptografía
  - Posible en varias capas
  - Usual en Aplicación y red

Algunas referencias

- RFC 8576 - Internet of Things (IoT) Security: State of the Art and Challenges. Abril 2019 (Informational)
  - Descripción del ciclo de vida de un dispositivo, amenazas, protocolos de seguridad (IP), desafíos, posibles soluciones, etc.
- “SEGURIDAD EN IOT. PROCESO EN URUGUAY”. ISOC, 2019
  - Documentos similares en otros países

Ejemplo: LoRaWAN



Aprovisionamiento

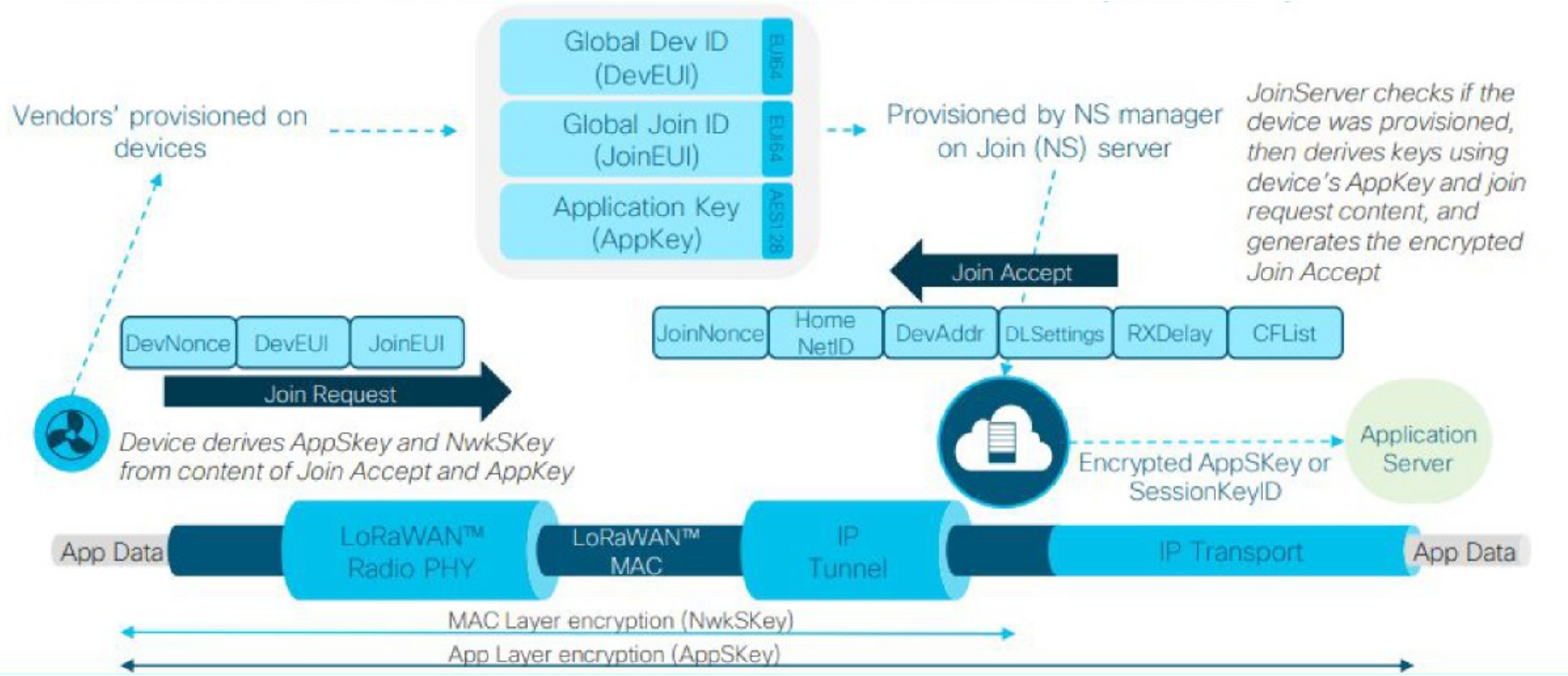
- Datos provisionados “de fábrica” (grabados en dispositivo y configurados en backend)
  - DevEUI: Identificación del nodo. “público”
  - AppEUI/JoinEUI: Identificador del “Join Server”
  - Network session key (NwkKey)
    - Clave AES de 128 bits
  - Application session key (AppKey)
    - Clave AES de 128 bits
- NwkKey, AppKey específicas de cada dispositivo
  - Almacenamiento seguro “left to implementation”
- En LoraWan 1.0 una sola clave, NwkKey

Claves de sesión

- AppSKey – Clave de sesión de aplicación. Derivada de AppKey
- FNwkSIntKey , SnwkSIntKey, NwkSEncKey

  Claves de sesión de red. Derivadas de NwkKey
- Compatibilidad con LoraWan 1.0:
  - AppSKey se deriva de NwkKey
  - FNwkSIntKey = SnwkSIntKey = NwkSEncKey
- JSIntKey y JSEncKey derivadas de NwkKey para proteger procedimiento de ReJoin

Activación Over the Air



Activación Over the Air

- Join-request
  - JoinEUI, DevEUI, DevNonce (16 bits)
- Join-accept
  - JoinNonce, + parámetros
  - JoinNonce se incrementa con cada Join-accept
- En base a las claves persistentes y los nonces, se derivan las claves temporales.
- Ejemplo (lorawan 1.1):

  AppSKey = aes128_encrypt(AppKey, 0x02 | JoinNonce | JoinEUI | DevNonce | pad16)

  FNwkSIntKey = aes128_encrypt(NwkKey, 0x01 | JoinNonce | JoinEUI | DevNonce | pad16)

  ……
- Rejoin-request permite refrescar las claves

Activación por personalización

- Las claves se configuran en el dispositivo
  - AppSKey, FNwkSIntKey , SnwkSIntKey, NwkSEncKey
- El dispositivo queda “atado” a una determinada red
- Se usan las mismas claves durante la vida del dispositivo
  - Compromiso de las claves de sesión afectan catastróficamente la seguridad del dispositivo
- Varios contadores se utilizan para la derivación de claves (nonces) y evitar replay (frame counters)
  - Debe asegurarse que no se repetirán

Uso de AES

- Para el cifrado con AES se utiliza una variante de "Counter Mode"
  - Se cifra un contador, y se hace el or exclusivo de los bits obtenidos con el mensaje a cifrar
  - Para mantener la seguridad, no debe repetirse el contador
    - Se utiliza como parte del contador el número de frame (uplink o downlink). En 1.1 no se permite repetirlo
- Para el cálculo de MIC

Vulnerabilidades LoRaWan 1.0.2

- Se reutilizan los frame counters
  - En reset (ABP) o por overflow (AVP y OTAA)
  - Prohibido en 1.1
  - Mejorado en 1.0.4
- Reuso de nonces (se generan pseudoaleatoriamente, no se verifica no reuso)
  - En 1.1 se generan secuencialmente (idem 1.0.4)
- No hay mecanismo para evitar el replay de mensajes Join-accept
  - Si en 1.1
- El reconocimiento no indica la secuencia del mensaje reconocido → Replay
  - En 1.1 se incluye en cálculo del MIC

Vulnerabilidades (cont)

- En 1.0.2 el cambio de contexto de seguridad (rekey de claves) no se confirma correctamente
  - Corregido en 1.1 con nuevos comandos
- No hay protección de integridad end to end. Si hay protección entre dispositivo y NS, pero no entre dispositivo y AS
  - Se deja abierto a implementación. A priori se asume NS confiable

Escenarios de fall-back

- Back-end versión 1.1, dispositivos 1.0.2
- Back-end versión 1.0.2, dispositivos 1.1
- En ambos casos se presentan vulnerabilidades
- Soluciones propuestas, dependientes de implementación
- Versión 1.0.4: soluciona la mayoría de los problemas

NB-IoT y LTE-M

- Utilizan bandas licenciadas
  - Más fácil buscar interferencias
- Heredan características de las redes LTE y 5G
  - Autenticación mútua y generación de claves mediante secreto compartido (SIM)
- Finalmente, la seguridad depende de la correcta configuración del proveedor, y del servicio ofrecido
  - ¿Es necesario acceder a toda Internet o puedo filtrar permitiendo solamente los servidores autorizados?
  - ¿VPNs?
  - Esto no es exclusivo de las tecnologías 3GPP/GSMA
- ¿Interacciones simplificadas para bajar overhead?

Otras

- Actualizaciones:
  - Tiempo de vida esperado de los dispositivos: varios años
  - Alta probabilidad de requerir actualizaciones en algún momento
    - Mecanismos seguros y escalables para actualizar
    - ¿disponibilidad de actualizaciones del fabricante?
- Seguridad del proveedor en los distintos eslabones
  - ¿apps tercerizadas?
  - ¿proveedor de la conectividad?
  - ¿proveedor de los dispositivos?
  - “Seguridad en la cadena de suministro”