IEEE BIG DATA 2024
Dec 15-18, 2024, Washington DC, USA

2nd workshop on Big Data Applications for Fight against Crime and Terrorism (BDA4FCT)

A Cloud-Based Key Rolling Technique for Alleviating Join Procedure Replay Attacks in LoRaWAN-based Wireless Sensor Networks

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101070455 (DYNABIC).
Disclaimer: Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Commission. Neither the European Union nor the European Commission can be held responsible for them.

Agenda

- Introduction
- Background Theory
- Related Work
- System Architecture
- Implementation Details
- Evaluation Methodology & Results
- Discussion
- Conclusion

Introduction

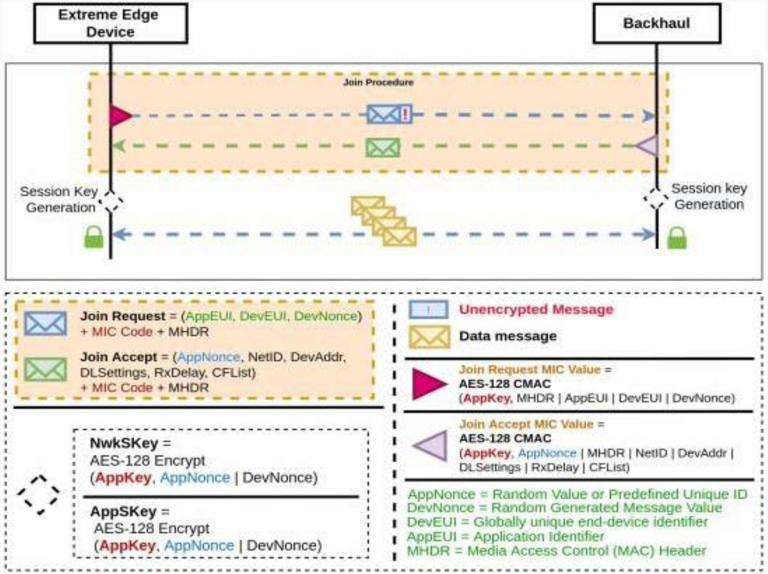
As IoT devices proliferate, Low Power Wide Area Networks like LoRaWAN have become pivotal for connecting devices over long distances with minimal energy consumption

- **Identified Challenge**: Join Procedure of the Over The Air Authentication (OTAA) mode begins with an ***unencrypted Join Request***, which contains a Message Integrity Code (MIC) which is the result of encrypting the contents of the message with the **AppKey**
    - Malicious actors acting as Man-in-the-Middle can reverse engineer the MIC value and derive the AppKey
    - This renders the Join Procedure vulnerable to replay attacks, where attackers can impersonate legitimate devices
- **Proposed Solution**: a dynamic key rolling technique, inspired by automotive security systems
    - Continuous regeneration of the AppKey, necessitating the device to re-join the network and re-authenticate
    - Enhanced authentication process without adding significant overhead

Background Theory (1/2)

LoRaWAN specification version 1.0.2 and 1.1

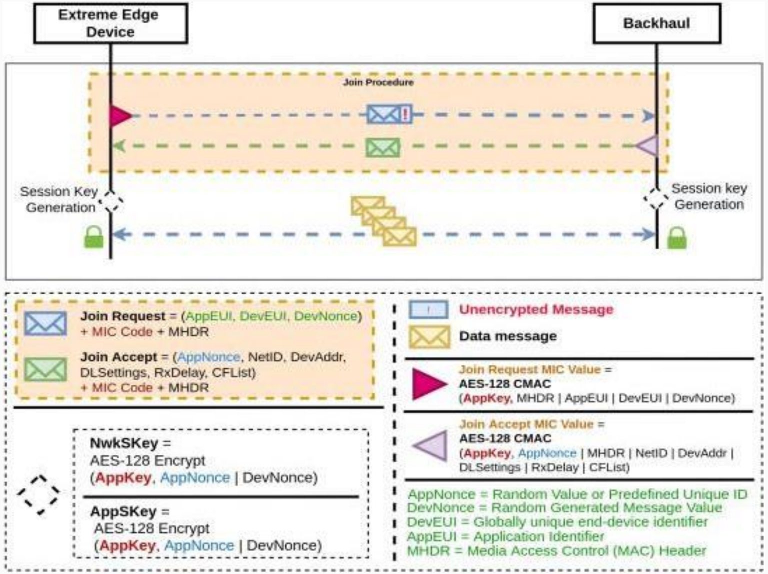
- Device Authentication:
  - **Join Request** includes:
    - AppEUI, DevEUI, DevNonce, MHDR heder
    - MIC value: the result of encrypting AppEUI, DevEUI, DevNonce, MHDR or combination of them by using the AppKey
  - **AppKey**:
    - Randomly generated and saved in the cloud
    - Manually entered in each extreme edge device
    - Used by the cloud to decrypt the MIC value of a Join Request in order to authenticate the extreme edge device
- Extreme edge device – Cloud communication:
  - **AppKey:**
    - used to generate the AppSKey and NwkSKey (session keys) by encrypting the DevNonce or AppNonce (randomly generated in the cloud) value
    - Used by the extreme edge device in conjunction with the DevNonce or AppNonce value to generate the same session keys as the cloud
  - **AppSKey and NewkSKey**:
    - Used by the extreme edge device and the cloud to encrypt and verify the communication between them



Background Theory (2/2)

LoRaWAN specification version 1.0.2 and 1.1

- <u>Security Limitations</u>:
  - The **MIC value** results from the encryption of the contents of the Join Request using AES-128 blocks for encryption (AES-CMAC algorithm)
    - Literature has identified techniques to decrease the processing power required for decryption
  - **AppEUI, DevEUI, DevNonce value, MHDR header** are transmitted unencrypted over the air
  - **DevNonce** is susceptible to interception by using a brute force attack
  - **AppKey:** its static nature of the leads to vulnerabilities:
    - Replay attacks and impersonation risks
    - Challenges in securing unencrypted Join Requests



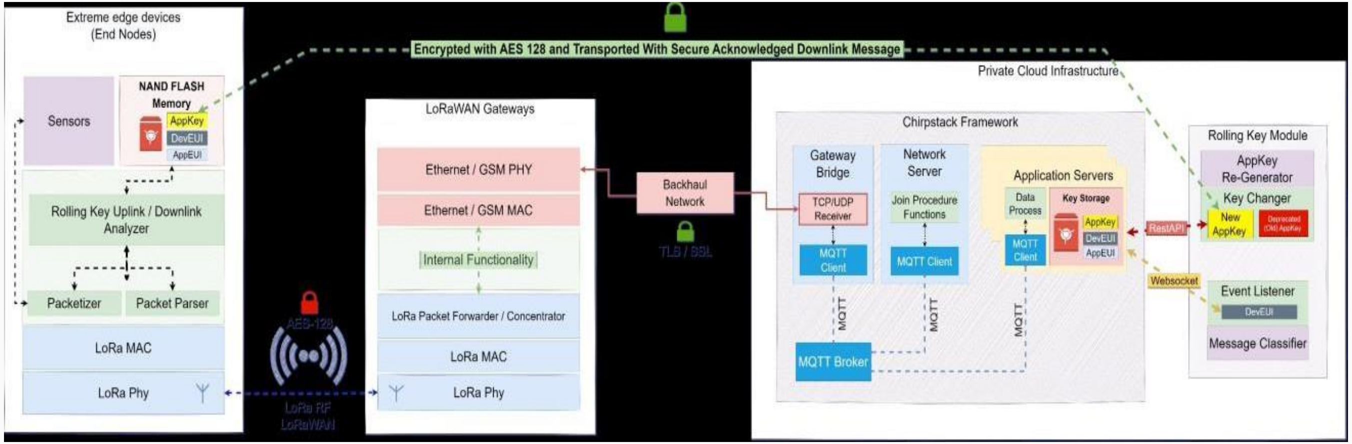
Related Work

Existing Approaches:

- **Masking Techniques**: leverage cryptographic operations, such as AES encryption, to obscure critical identifiers like the DevNonce and MIC.
  - → often require significant modifications to the LoRaWAN protocol, increasing complexity and deployment costs.
- **Timestamp-Based Mechanisms**: by incorporating temporal validation into message exchanges, these methods ensure that replayed packets are identified and rejected
  - → their reliance on synchronized clocks introduces vulnerabilities, as desynchronized devices may inadvertently reject legitimate packets or accept malicious ones.
- **Blockchain Solutions**: ensure integrity and traceability of network events, while smart contracts can enforce security policies
  - → computational overhead and high energy consumption make blockchain less feasible for the resource-constrained environments of IoT networks like LoRaWAN.
- **Adaptive Cryptographic Protocols**: periodically update session keys by using dynamic key management strategies, reducing the attack surface for replay attempts
  - → often involve additional computational and communication overhead, which can impact network efficiency.
- **Hybrid Approaches**: integrating masking with blockchain or timestamping offers promising results
  - → the trade-offs in complexity, compatibility, and scalability remain significant barriers to widespread adoption

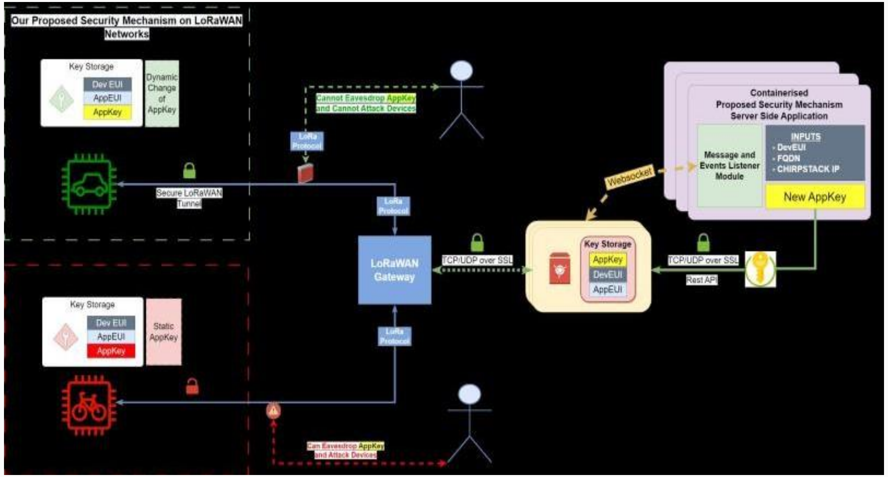
System Architecture

- **Components**:
  - Edge devices
  - LoRaWAN Gateways
  - Private Cloud Infrastructure including the Chirpstack framework and the Key Rolling Module.
- **Key Rolling Module**:
  - Dynamic AppKey updates
  - Transmission of the new AppKey from the cloud to the extreme edge device using secure tunnels.



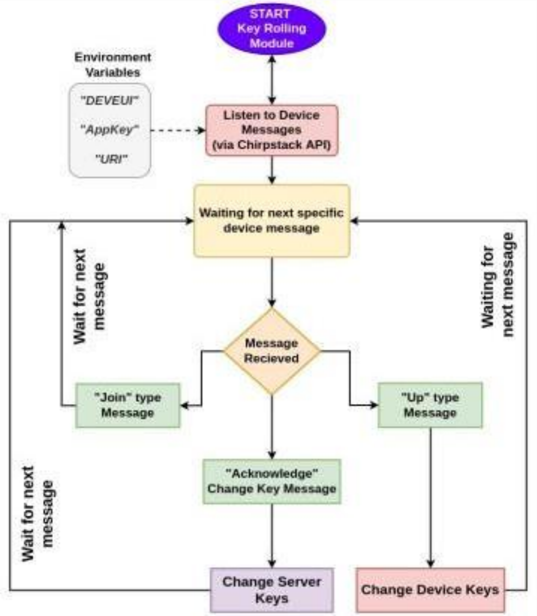
Implementation Details (1/2)

- Assumptions:
  - AppEUI IDs of all extreme edge devices were assigned as "0000000000000000“ → Chirpstack framework does not require an AppEUI in LoRaWAN versions 1.0.2 and/or higher
  - LoRaWAN specification version 1.0.2 was employed due to the specifications of the extreme edge devices and the gateway used → the suggested solution can be modified to support newer versions as well
- Setup:
  - Tools: Chirpstack framework, PyCom FiPy devices, Lorix One devices.
  - Cloud integration with RESTful API and WebSockets.
- Process:
  - Replacing static AppKey with a dynamic key.
  - Secure downlink message transmission and acknowledgment.

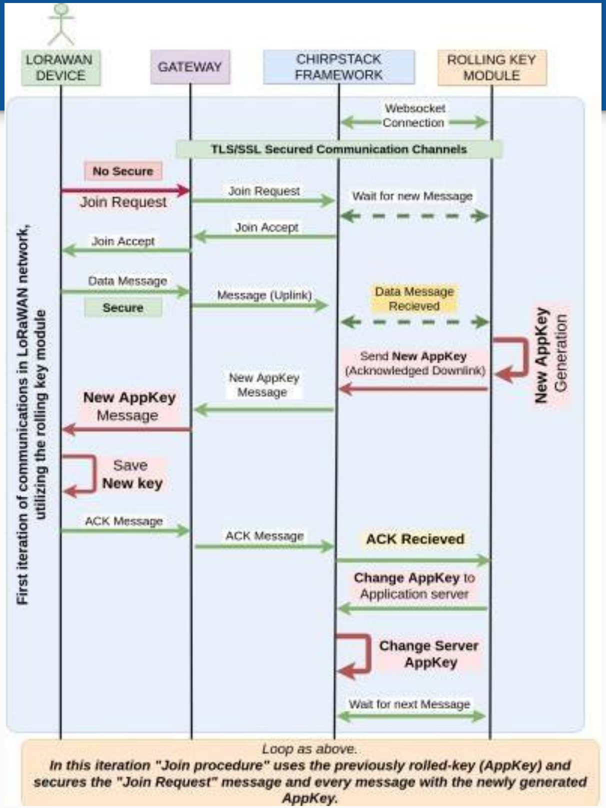


Implementation Details (2/2)

Flow Chart



Sequence Diagram



Evaluation Methodology & Results (1/2)

- Experiments Conducted:
  - **Alpha and Beta** experiments (with and without key rolling mechanism) → evaluating the effectiveness of the proposed approach in mitigating replay attacks
  - **Delay** experiment to measure performance impact → evaluating the key rolling mechanism's performance and the delay induced in sequential procedures
- Variables:
  - **Independent**:
    - Time window: 20 minutes duration for optimal measurements
    - Transmission time: 60 sec of client transmission
    - Packet data:
      - Alpha and Beta experiments:
        - Benign: "Temp: 28 C"
        - Malicious: "Temp: 50.0 C"
      - Delay experiment:
        - Benign: "Temp: 32.0 C"
  - **Dependent**:
    - Alpha and Beta experiments:
      - Packets per minute
    - Delay experiment:
      - Delay in AppKey update.

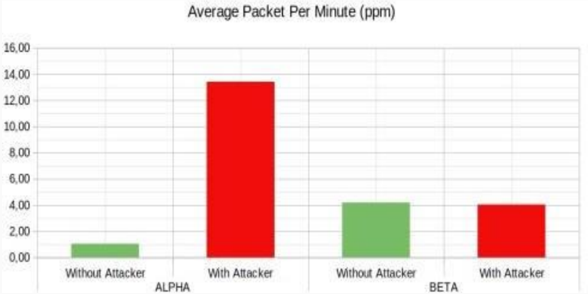
→ **Assumption for all three experiments**: the attacker has already acquired the AppKey from the "victim" device.

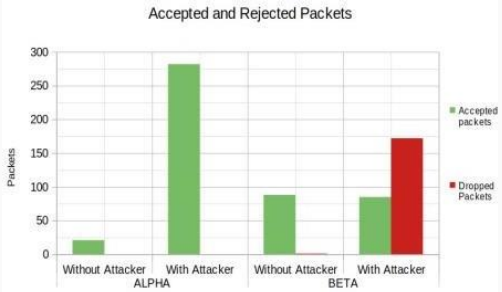
→ 2 phases for the Alpha & Beta Experiments:

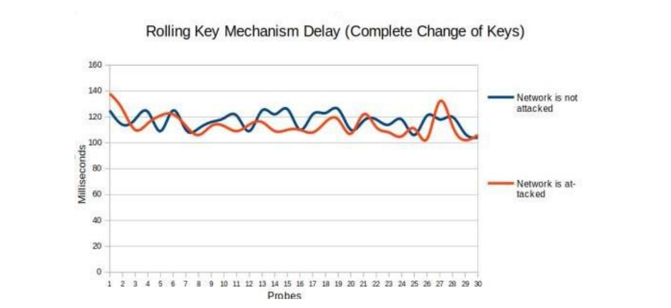
- **"With Attacker":** an attacker executed a replay attack by flooding the LoRaWAN network with packets every 4 seconds
- **"Without Attacker":** no attack performed

Evaluation Methodology & Results (2/2)

- Alpha vs. Beta Experiments:
  - Significant **reduction in malicious packets** with key rolling enabled:
    - Alpha - no key rolling module:
      - With Attacker: 13.42 ppm
      - Without Attacker: 1.05 ppm
    - Beta – key rolling module included:
      - With Attacker: 4.09 ppm
      - Without Attacker: 4.19 ppm
  - **Rejected packets**:
    - Alpha - no key rolling module:
      - With Attacker: 0/282 packets were rejected
    - Beta – key rolling module included:
      - With Attacker: 172/257 packets were rejected
- Delay Experiment:
  - **Minimal delay** induced by the proposed solution for rolling the new AppKey: 113-116ms
  - **Impractical** for attackers **to execute replay attacks** within this window.







Discussion

- Effectiveness:
  - Successfully mitigates replay attacks.
  - Minimal impact on network performance.
- Limitations:
  - Physical attacks and potential future risks with quantum computing propose future steps for research

Conclusion

- Summary:
  - The proposed dynamic key rolling technique enhances security in LoRaWAN networks.
  - Backwards compatible and low complexity.
- Future Work:
  - Explore blockchain integration and real-world scalability.

Thank you for your attention

Q&A

Presented by:
e-mail:

This project has received funding from the European Union’s Horizon Europe research and innovation programme under grant agreement No 101070455 (DYNABIC). Disclaimer: Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Commission. Neither the European Union nor the European Commission can be held responsible for them.