Individual Assignment

ARINC 429 BUS Spoofing Attack and Defense

Course: CYSE 465 Fall 2025
Student:EB
Protocol: ARINC 429

1. Report: Attack and mitigation

Attack Details: ARINC 429 Bus data injection/spoofing

ARINC 429 is the primary data bus standard applied in commercial aviation for inter-subsystem communication. The standard is defined by its bit structure; it sends 32-bit words, of which 24 bits carry the actual data and 8 bits are used for designating the data in between. But this simplicity results in a vulnerability in security. The protocol's requirements do not include any built-in methods for verifying message authenticity, maintaining integrity, or ensuring confidentiality.

A spoofing attack/injection directly exploits the lack of authentication. In this attack, a malicious actor with physical or logical access to the ARINC 429 bus can impersonate a legitimate transmitter.

- Attack vector: The attacker connects a malicious device, like a rogue transmitter or a compromised legitimate unit, to the bus.

- Attack method: The malicious device listens to the bus to understand the data format and label. It then begins transmitting messages with the same label as a legitimate source but containing malicious data. For example, it could spoof the air data computer and transmit false air speed data.

- Impact: Since receiving systems have no way to verify the source or integrity of a message, they will process the spoofed data as if it were genuine. This can lead to catastrophic consequences, such as the autopilot executing dangerous maneuvers based on false information.

This attack is likely because ARINC 429 is a broadcast protocol on a shared medium with no source identification beyond the data label itself.

Proposed defense authentication

To mitigate the spoofing attack, I propose augmenting the ARINC 429 protocol with a cryptographic authentication mechanism.

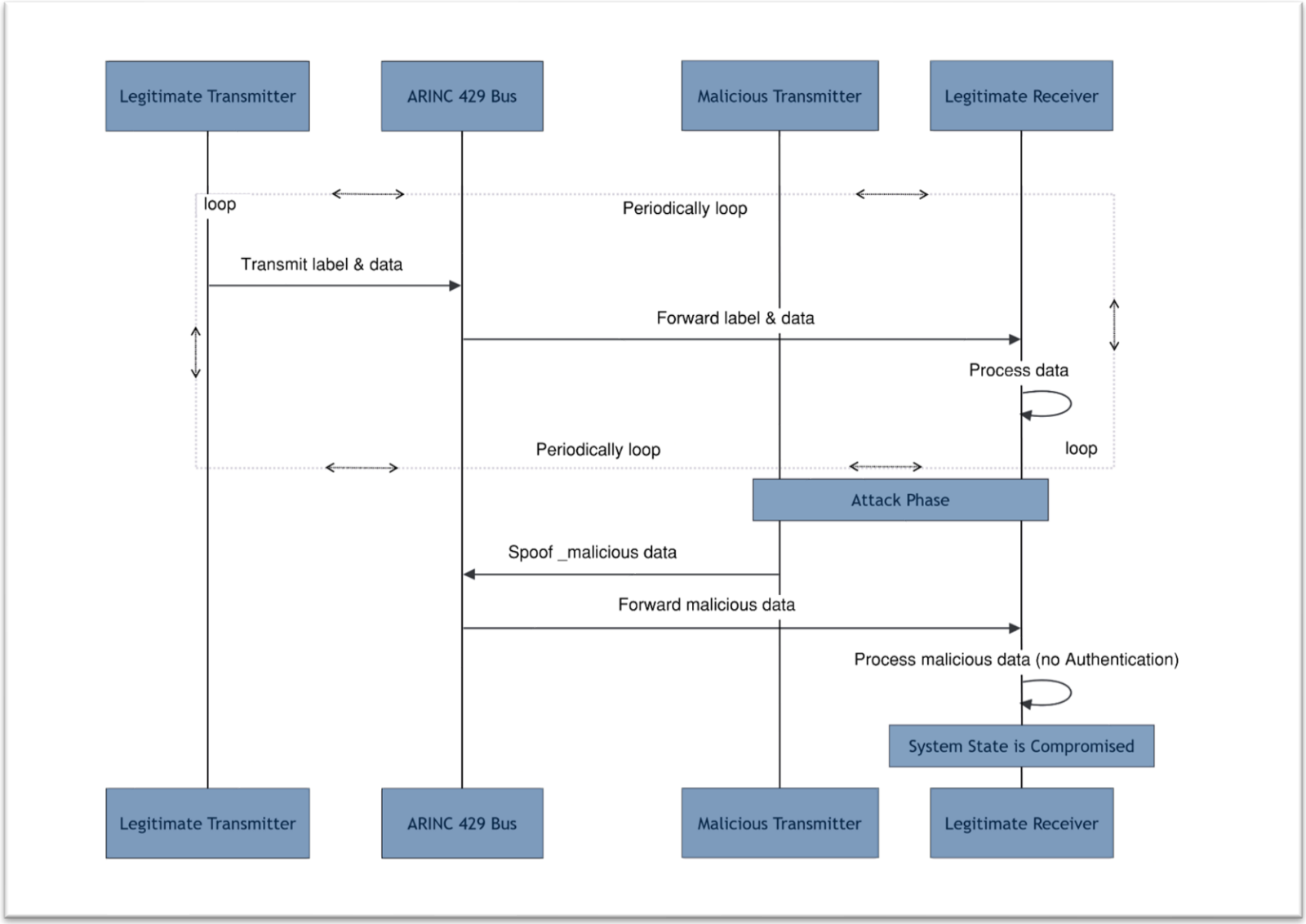
- Mechanism: For each critical data word, the transmitting system calculates a MAC using a secret symmetric key shared only with the intended receiver. This MAC is a crypted and the data word's contents. The MAC is then appended to the standard 32-bit ARINC word as a new field. Given the 32-bit limit of a single ARINC word, this would typically require using a second, dedicated ARINC word to transmit the MAC.

- Process: The receiver, possessing the same shared key, recalculates the MAC upon message receipt. If the assessed MAC matches the received MAC, the message is authenticated, and its integrity is verified. A mismatch indicates a spoofed or corrupted message, which should be rejected by the receiver.

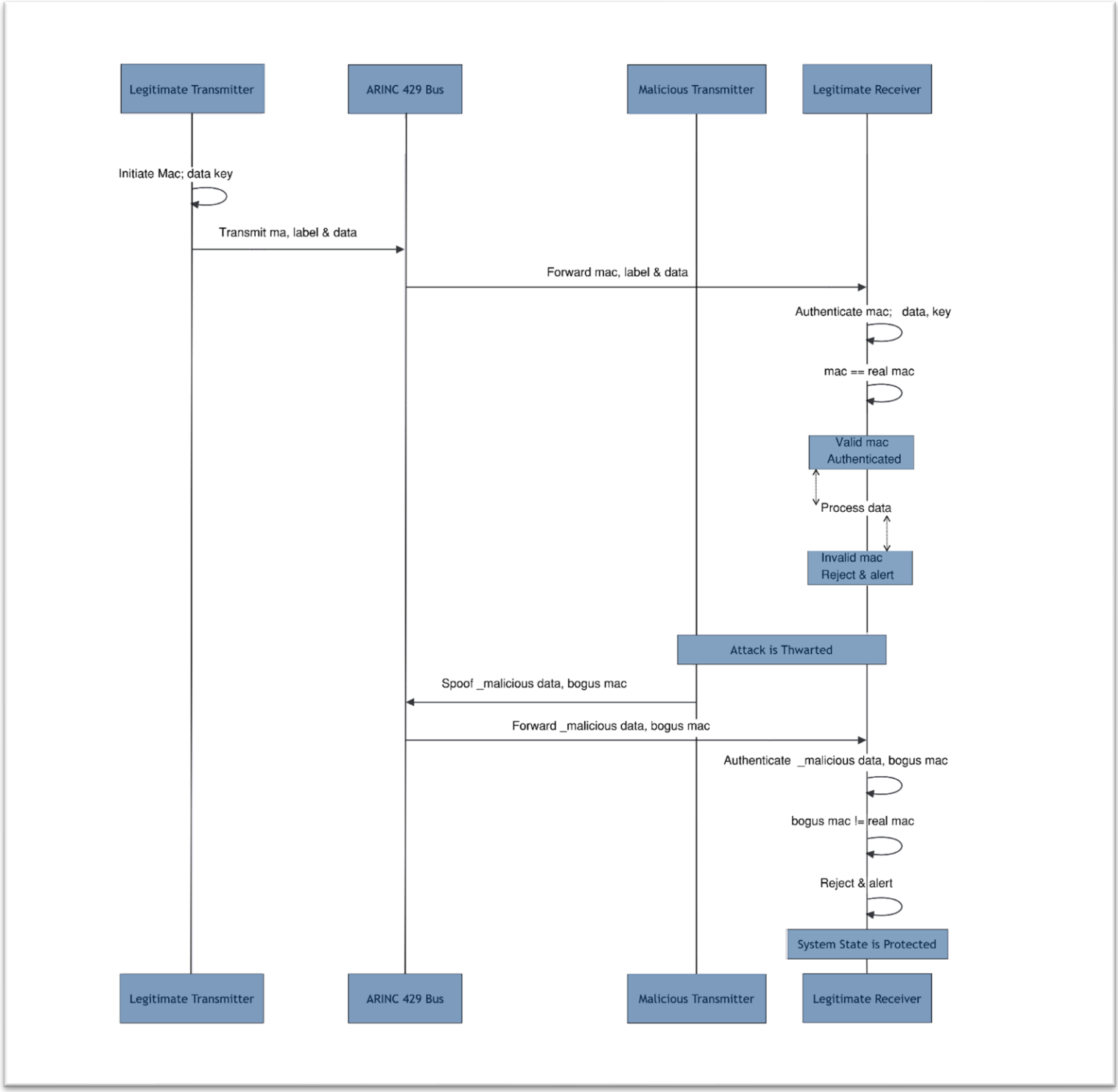
- Implementation Consideration: This defense is not part of the ARINC 429 standard and would require modifications to both transmitting and receiving part. It introduces latency and requires secure key management.

2.**SYSML**

Attack



Defense


Legitimate Transmitter
ARINC 429 Bus
Malicious Transmitter
Legitimate Receiver
Initiate Mac; data key
Transmit ma, label & data
Forward mac, label & data
Authenticate mac; data, key
mac == real mac
Valid mac
Authenticated
Process data
Invalid mac
Reject & alert
Attack is Thwarted
Spoof _malicious data, bogus mac
Forward _malicious data, bogus mac
Authenticate _malicious data, bogus mac
bogus mac != real mac
Reject & alert
System State is Protected
Legitimate Transmitter
ARINC 429 Bus
Malicious Transmitter
Legitimate Receiver

3. Assurance case for the Authentication defense

Claim:

- The system is resilient against ARINC 429 spoofing attacks.
- Critical ARINC 429 messages are authenticated.
- A secure MAC algorithm is used.
- MACs are correctly generated by transmitters.
- MACs are correctly validated by receivers.
- Spoofed messages coming.
- Crypted keys are managed.
- Keys are distributed securely.
- Keys are stored securely in hardware
- Keys are rotated periodically.
- The authentication solution is performant.
- MAC calculation is bounded.
- Increased bus load from MAC is acceptable.

Evidence:

- Argue that all spoofed messages are detected and rejected.
- Deploy authentication on critical data paths.
- NIST-approved standards
- Software and hardware verification.
- Integration tests see rejection
- Ensure keys remain secret
- Use of secure key
- Use of Hardware Security Modules
- Established key lifecycle method
- Cheak bus loads are within acceptable limit.
- Performance profiling on bus
- Bus traffic analysis

4.Final verdict

This assurance case shows that while the mitigation is technically healthy, it relies on several essential supporting processes. Testing and standard processes reinforce the core cryptography. However, effective key management, which involves organizational and routine controls, along with performance validation, an engineering constraint, are also vital for ensuring overall security