CCX TECHNOLOGIES

Cryptographic Signing of ARINC-429 Data

White paper

Charles Eidsness
June 2021

CONTENTS

1. The lack of ARINC-429 authorization

The ARINC-429 standard was introduced in 1978 based on the older ARINC-419 standard, which was first introduced in 1966.

At the time very little thought was given to authenticating the originator of ARINC-429 messages. This has left ARINC-429 interfaces open to data injection and spoofing attacks, similar to those described on CAN Bus interfaces in CISA ISC-ALERT-19-211-01 [1].

> *"An attacker with physical access to the aircraft could attach a device to an avionics CAN bus that could be used to inject false data, resulting in incorrect readings in avionic equipment.*
>
> *The researchers have outlined that engine telemetry readings, compass and attitude data, altitude, airspeeds, and angle of attack could all be manipulated to provide false measurements to the pilot."*
>
> *– CISA*
> *ICS Alert*

2. ARINC-429 injection and spoofing attack vectors

The ARINC-429 databus is high impedance when the transmitter is idle. As a result, it is possible to physically attach a second transmitter that can transmit spoofed data while the real transmitter is idle.

Most ARINC-429 receivers will ignore data with an incorrect parity bit which makes it possible to perform a denial-of-service (DoS) attack by corrupting the parity bit with a second transmitter attached to the bus.

These two attacks can be combined to completely replace all ARINC-429 messages by installing a small device to the wiring harness of an ARINC-429 databus.

Cryptographic signing of ARINC-429 data would provide receivers with a means to authenticate the transmitter which would eliminate this attack vector.

---

1 https://us-cert.cisa.gov/ics/alerts/ics-alert-19-211-01

3. Difficulties signing (or encrypting) short, real-time data samples

Unfortunately, due to the nature of ARINC-429 data, it is difficult to sign (or encrypt) the messages in real time.

The data is transmitted in small 4-byte messages with acute timing requirements which is too short to sign or encrypt with a cryptographically sound algorithm. Most modern cryptographic algorithms are designed to operate on larger frames of at least 16-bytes.

To overcome this obstacle CCX is proposing signing cached historic data. This will delay the time required to identify a nefarious message however it can be implemented in a backwards compatible way on existing ARINC-429 interfaces.

4. A backward compatible ARINC-429 data signing protocol

CCX has implemented a backward compatible data signing protocol for ARINC-429 traffic by collecting a block of ARINC-429 messages, signing them as a single block, and then sending the signature using multiple ARINC-429 messages.

1. On the transmitter, we calculated a running SHA-256 checksum of all ARINC-429 messages in a specific period of time.

   The shorter the time period the quicker an attack can be identified, the longer the time period the less additional ARINC-429 bandwidth is required to transmit the signature.

2. After either a fixed period of time or number of messages we used the EdDSA algorithm to sign the checksum which results in creating a 96-byte long signature. The EdDSA algorithm requires a public/private keypair.

   - For the purposes of our development, we are generating the keypair at boot-time and sending the public key to all receivers using a custom ARINC-429 label (0368 octal).

   - It would also be possible to utilize a standardized PKI system for sharing keys however one does not currently exist for avionics systems (but should, and most likely will one day).

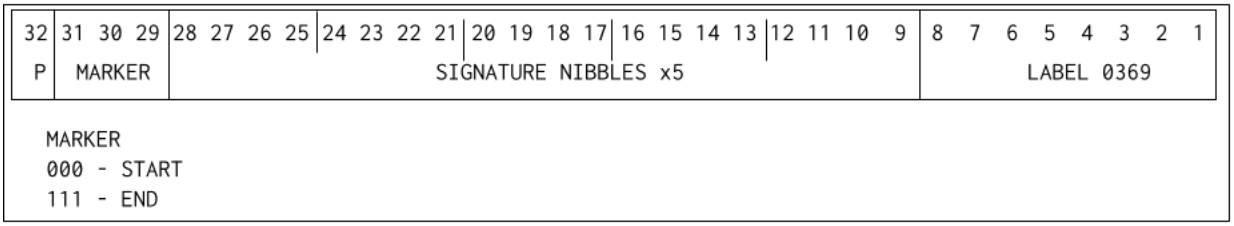
3. The transmitter then transmits the 96-byte signature, 20 bits at a time using standard ARINC-429 messages with label 0369 (octal).

   We used the two remaining bits in the ARINC-429 word to identify a start-of-signature and end-of-signature event.

   - In our demo system we have used a transmit interval of 100ms, which allows for a signature check every 3.9 seconds.

4. The receiver can then verify signatures if it also calculates a running sha256 checksum and collects all of the transmitted signature bytes.

By utilizing an unused ARINC-429 label it is possible to add an optional signature in a backward compatible way. The receiver will ignore the signature if it doesn't support the check. The signature check does not need to be performed by every receiver, but could instead be performed by an intrusion detection system monitoring the ARINC-429 traffic.



ARINC-429 signature word format

5. Other uses

This method of injecting parts of a historical signature into a stream of real-time data can be used to add signatures to other interface types as well, in a backward compatible way, ie. CAN bus, ADSB, AFDX, MIL-1553, etc.

Want to discuss this topic? Contact us:

info@ccxtechnologies.com

Contact us:

+1 (613) 701-6363
info@ccxtechnologies.com
ccxtechnologies.com