FICAA

Security Characteristics Framework

Remember

There is no 100% security

Security, like all engineering, involves tradeoffs

Know what you are trying to secure

The adversary…

State Sponsored

Secure Network Communications

Various characteristics of “secure”

Mnemonic device **FICAA**

F : Freshness

I : Integrity

C : Confidentiality

A : Authentic (sometimes written “An”)

A : Authorized (sometimes written “Az”)

FICAA

Freshness

Recipient of message has reason to believe the message is related to the current state of the vehicle.

Recipient will accept a given message once and only once.

Recipient will reject messages not sent “recently”.

FICAA

Integrity

Recipient can detect if a message was tampered with.

Recipient rejects messages that have been tampered with.

FICAA

Confidentiality

Contents of a message are not disclosed outside of the secure network.

In command-and-control networks this characteristic is not often needed.

The data looks random to non-members.

FICAA

Authentic, “An”

This can have three meanings.

1) The nodes participating in the secure network are authentic and present cryptographically strong evidence to demonstrate this.
2) The messages sent are authentic (not spoofed by a nonmember) and the message contains evidence to demonstrate this.
3) (less common) The authentic message came from a specific authentic node and the message contains evidence to demonstrate this.

The difference isn’t usually important – we get authentic messages only from authentic nodes.

What is important is understanding the granularity of authentication evidence required:

- Evidence is for a unique controller (e.g., identified by part number and serial number)
- Evidence is for any controller of a given type (e.g., identified by the part number)
- Evidence is any controller made by the OEM
- None authentication – any controller that can “speak” the protocol is allowed to participate in the network

FICA**A**

Authorized, “Az”

Each node participating in the secure network is authorized to send all or a subset of all possible secure messages and present cryptographically strong evidence to demonstrate this.

Network Naming Scheme Examples

SAE J1939-91C (proposed)

Key Management	Onboard generation of long-lived S_N & vehicle S_V S_N : Network key generated and shared by leader S_V : Vehicle (session) key from KDF(S_N, joint entropy)	by nodes	An : (PN, SN) tied to VIN Az : all
Message Exchange	[F32|32 I31]64 *or* [F32|32 I31 C64]0		

Note appearance of FICAA

AEF TIMv2 (proposed)

Key Management	Onboard generation of long-lived S_C & vehicle S_S S_C : Couple key derived using x25519 S_V : Session key from KDF(S_C, joint entropy)	by nodes	An : OEM any (or higher) Az : role
Message Exchange	[F32|4 I4]56		

Note appearance of FIAA and lack of C (since there is no confidentiality in this protocol)

Network Naming Scheme Details

FIC Annotations

```
F x|y     x: total length of freshness, in bits
          y: length of explicit portion, in bits
I x       x: length of explicit integrity tag, in bits
C x       x: length of ciphertext, in bits
An x      x: size of public key msg is authentic to, in bits
```

AA Annotations

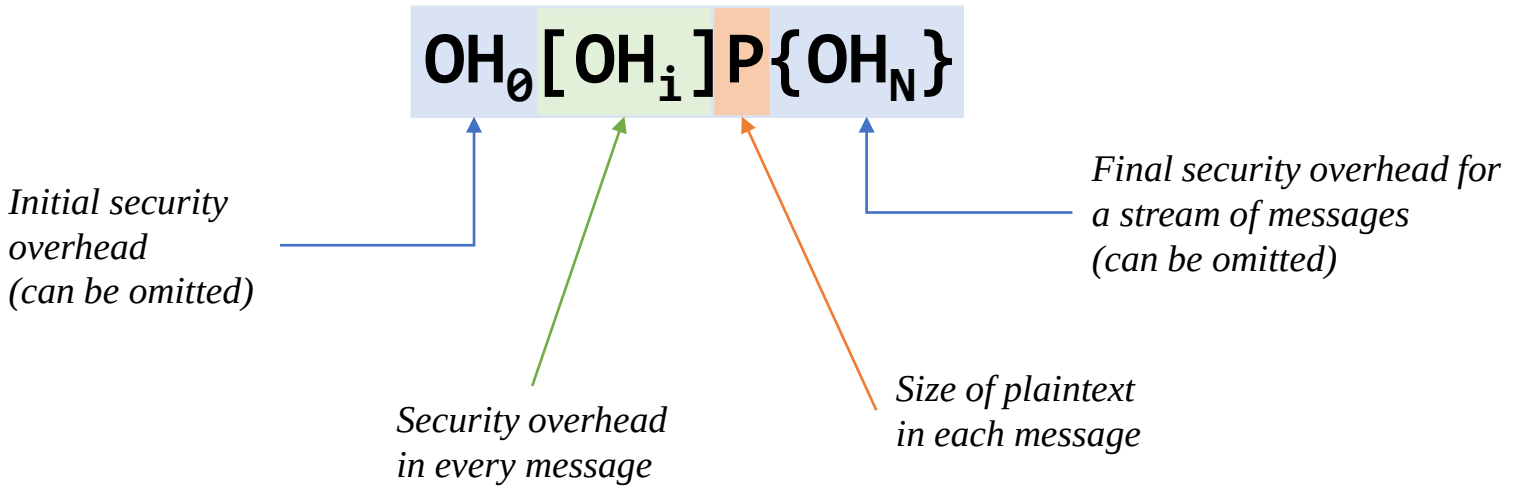
`An x  x: (Pn, Sn), (Pn), OEM any, none`

(Pn, Sn)	authentication to unique ECU, by part number and serial number
(Pn)	authentication to all ECUs of the same part number
OEM any	authentication to any ECU made by the OEM
none	no authentication, any ECU that knows the protocol can participate

`Az x  x: all, role`

all	authorized to have keys for all messages
role	authorized to have keys for a subset of messages (e.g., a channel, or specific PGNs)

Message Exchange Annotations Generic Form



FICAA Annotations Examples

We'll come to these in over the semester…

1.	No message security	**`[]64`**
2.	Block cipher	**`[C128]0`**
3.	Fresh messages with stream cipher	**`[F8|8 C56]0`**
4.	Message with integrity tag	**`[I8]56`**
5.	Fresh messages with integrity tag	**`[F4|4 I8]52`**
6.	NaCl box	**`[F192|192 I128 CN An256]0`**
	or	**`An256[F192|192 I128 CN]0`**
7.	Wrap a stream of N plaintext msgs (preceded by nonce, followed by tag)	**`nonce ([]64)xN {I128}`**