Securing a Stream of Messages

Leaving individual message unchanged.

Lab5

Pretty Good Security(?): Protect a Stream with Low Overhead

Remember

There is no 100% security

Security, like all engineering, involves tradeoffs

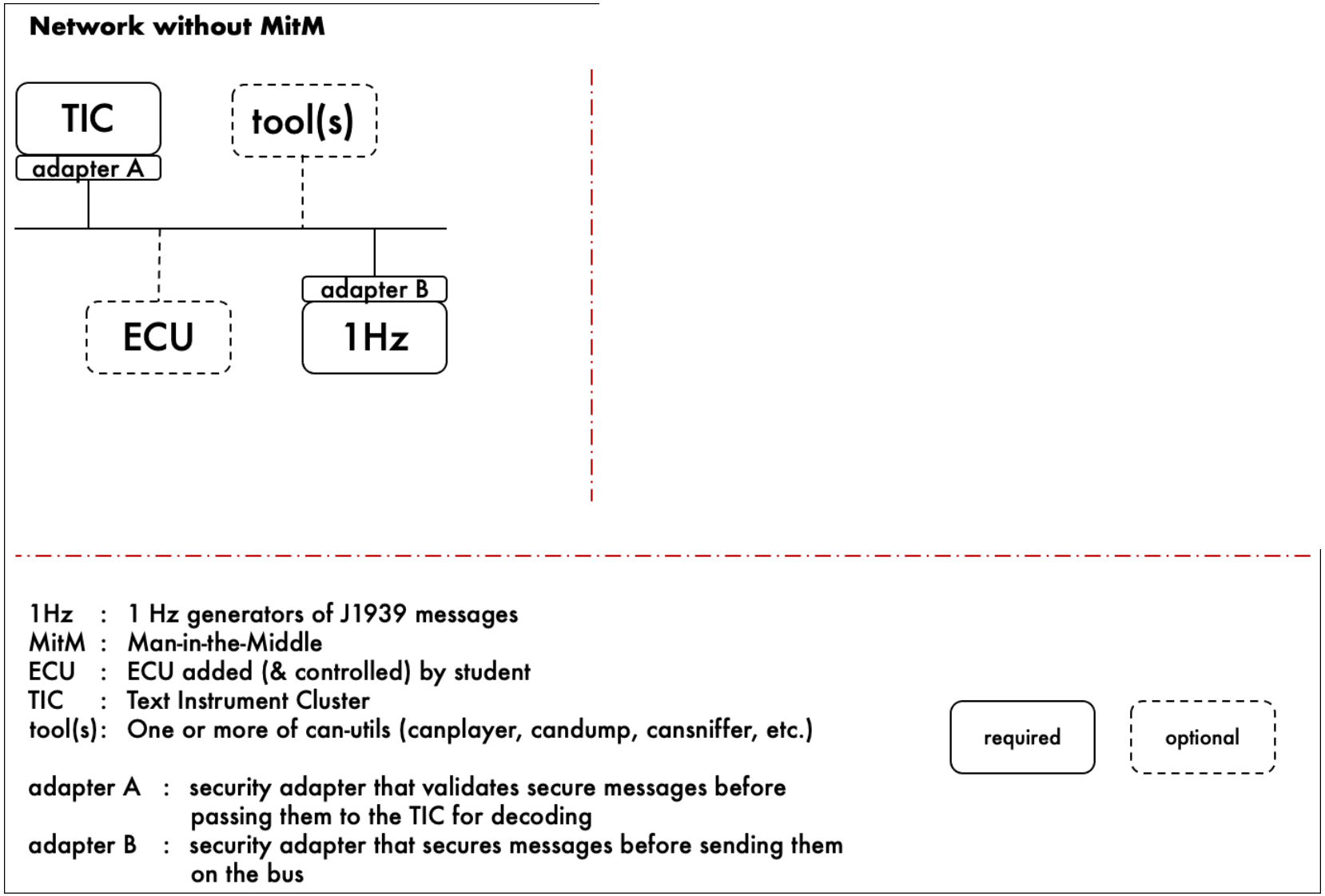
Know what you are trying to secure

The adversary…

State Sponsored

Network Configuration

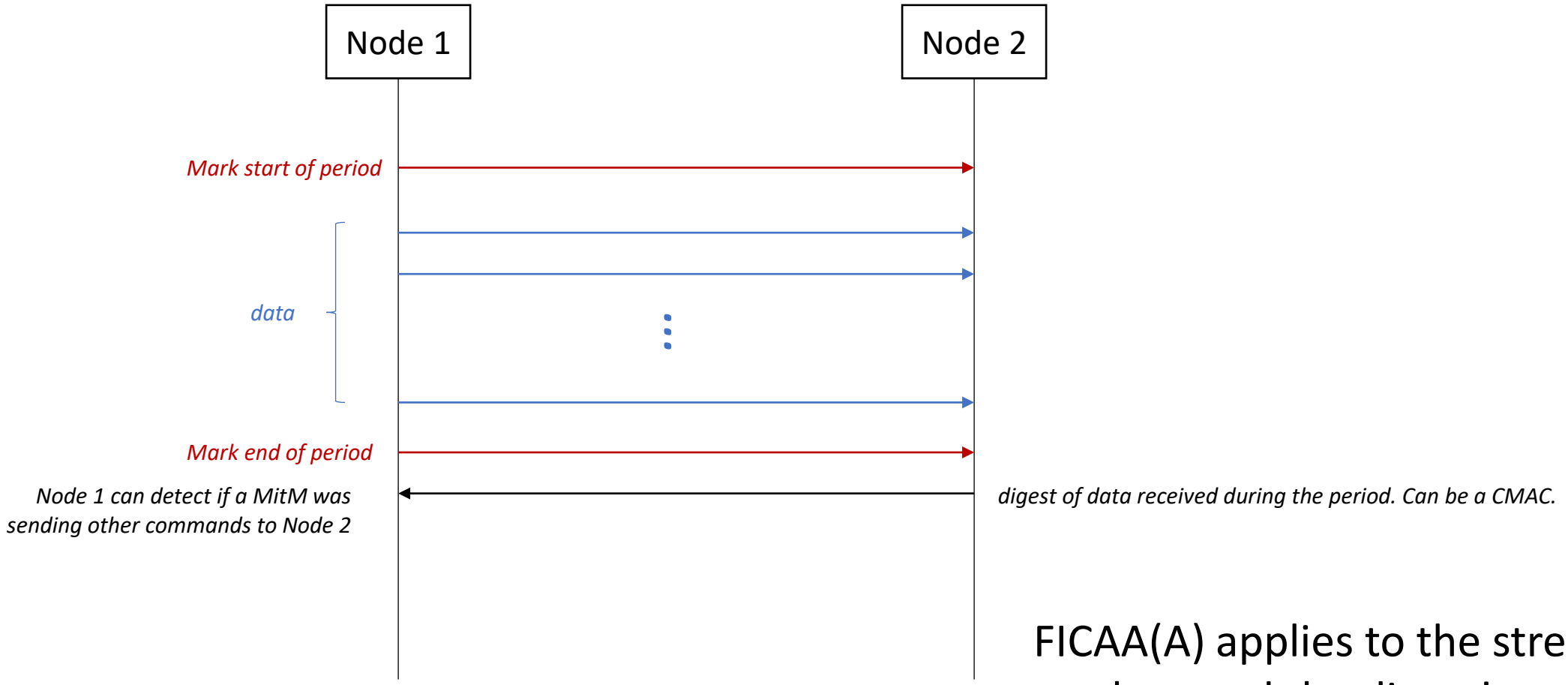
Simple Network for this Lab



Historical Reference

- Personal experience & academic papers
  - Wrap a stream of data with a CMAC that is validated periodically
  - Often make the simplifying assumption that messages won't be dropped or get out of order
  - Securing CAN Traffic on J1939 Networks https://www.researchgate.net/profile/David_Nnaji2/publication/353920347_Securing_CAN_Traffic_on_J1939_Networks/links/6132430cc69a4e487979bef1/Securing-CAN-Traffic-on-J1939-Networks.pdf
    - Intrusion Detection System (IDS)
- Another "free" (as in no overhead) security idea
  - Optimizing CAN Bus Security with In-Place Cryptography 2019-01-0098
  - doi:10.4271/2019-01-0098
  - $35 to download from SAE

Variant 1 : Point-to-Point

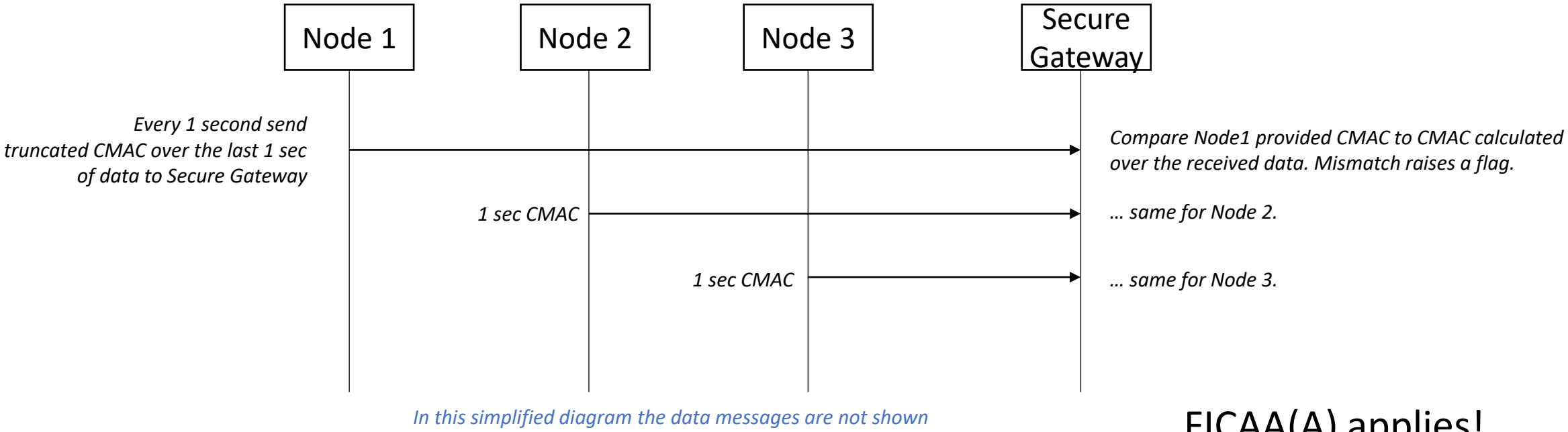


FICAA(A) applies to the stream markers and the digest!

Variant 1 : Point-to-Point

- FICAAA?
  - This is an example where AVAILABILITY is relevant to the protocol.
  - We need to be careful and intentional about the kind of digest used.
  - Will the protocol fail often?
  - Ex1: Due to out-of-order (could be common, if heavy traffic and transmit queue uses priority instead of FIFO to pick next message to send)
    - In these cases, the application has no way to know what order the messages were actually placed on the bus. Which makes it hard for the application to calculate the digest.
  - Ex2: Due to lost messages (could be common, if bursts of heavy traffic overfill the receive queue of Node2)
    - In these cases, the application at Node2 will calculate a digest that doesn't match Node1.

Variant 2 : Multi-Node



FICAA(A) applies!

From Daily, et. al.

Variant 2 : Multi-Node

- Daily paper, Av (availability) as “false-positives”. (emphasis added)

The false positive rate is determined by a ratio of the reported CMAC mismatches to the number of CMACs calculated with no intrusions. Due to the way CMACs are calculated, the timing of the sentinel message containing the CMAC from the CAN Conditioner must provide sufficient gaps such that the same messages are used to calculate the CMAC on the Secure Gateway. If the order of a message is changed, then the CMAC comparison would fail, leading to a false positive. After some initial tuning of the timing, a 20 millisecond gap on either side of the sentinel message lead to a low false positive rate. In one bench test that went for 24 hours, there were 508,762 CMACs that matched and 12 that did not. This is a false positive rate of 0.00236%, which is arguably too much. This would lead to false maintenance actions and operators would tend to ignore the warnings. However, this is a promising preliminary result and similar numbers were found when testing on vehicle. Additional logic regarding the number of messages and better tuning of the timing parameters should drop the false positive rate even lower.

Lab

- Use Sx = 00000000 11111111 22222222 33333333