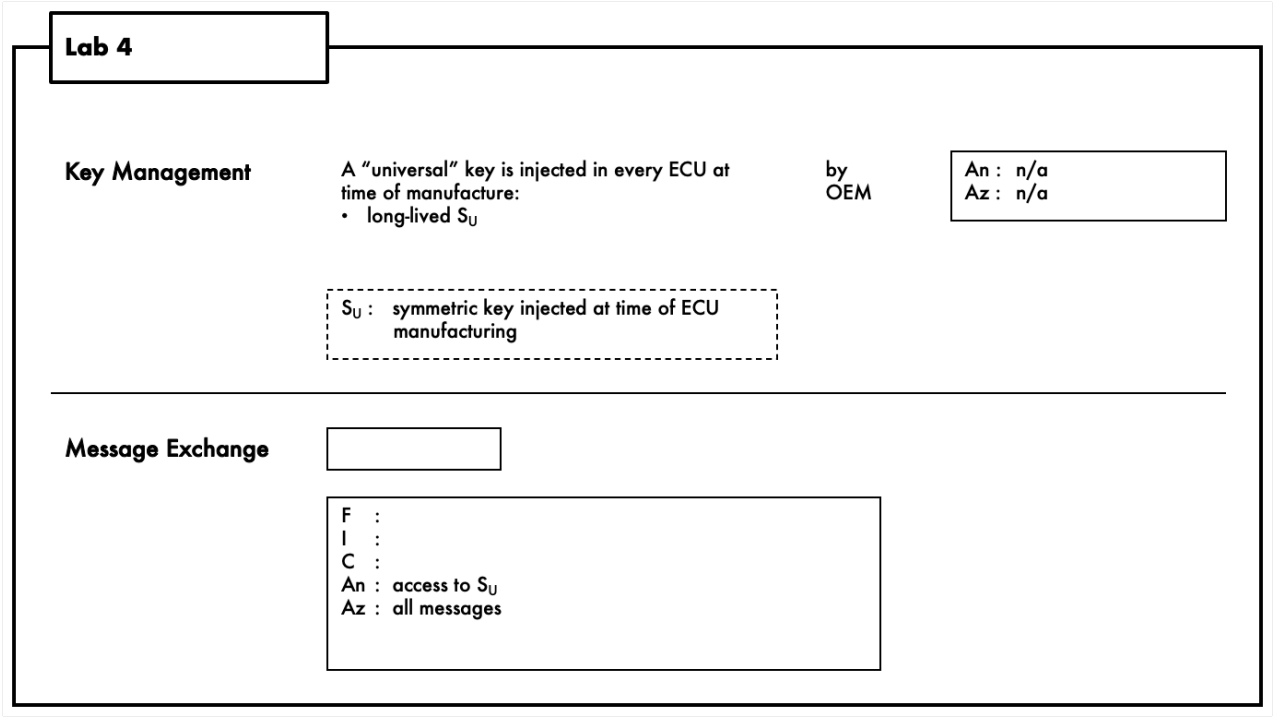

Scenario : Now the product engineers, test engineers and product support engineers are unhappy because when they log the network traffic to debug hard problems they see only random values. They insist (rightly) that for the most part command-and-control messages don't need confidentiality. Rather, they need integrity.
Like last time, the consensus is that it is "affordable" to lose one byte to security overhead, as long as the rest of the data is in plaintext.

Exercise 1 : Propose a method using *only* AES CMAC to provide integrity in each message. There is no need to create a PoC, rather, just populate the framework above to describe your solution.

Exercise 2 : Explain why lacking a freshness value makes it trivial to attack the proposal. (again, no need for a PoC).

Exercise 3 : [DEFEND] This is an example where the policy is really important. The product engineering manager tells you that you must detect invalid messages. If you detect 3 invalid messages you stop communicating with the ECU.
Assume you can only use byte 8 for security overhead, propose a message exchange scheme that has *both* freshness and integrity. Document your concept using the framework. Is this "good enough" security? (hint: the answer is "it depends" -- what does it depend on? When is it good enough? When is it not good enough?)

Hints/Notes:

- Your freshness value can have implicit and explicit components.
- Your explicit freshness and tag do not have to be equal length.
- Since your are limited to one key S_U, how can you make replays over the long-term use of the network much harder? Think about what should be done when your vehicle first starts up.

<continues on next page>

Exercise 4 : Create a PoC for your proposal in exercise 3. Turn in examples of a) a run where there is no attack and b) a run where there is an attack. (you should write an error message to the console, include why the message is invalid; correlate the console log to the candump log.)

Exercise 5 -- Week 2 : Question and answer session with your classmates. You will explain your design and they will ask questions and/or suggest improvements. Use two pages/slides.

- Page 1 -- Key Management and FICAA.
- Page 2 -- Message Exchange and FICAA. Also discuss your replay prevention.