To: Craig Bauer

Re: Manuscript ID UCRY-2016-0011

Thank you for the review of our MiniMac manuscript submitted to Cryptologia.

We feel that the review misunderstands an essential feature of our contribution. Also, we feel that manuscripts deserve more than one review for statistical reliability. Therefore would like to request that you seek (an) additional review(s).

As its main criticism, the one review states: *``One of the major claims in the paper is that Mini-MAC does not take any additional bits (e.g., Table 1). But this is false. It appears to add 0 to 32 bits (possibly 64 bits; it's not completely clear if you ever use more than 4-byte authentication tags) to each message. This would definitely have the effect of delaying messages as other ECUs would have to wait for longer messages, contrary to the claim in the paper.’’* The review’s assertion is incorrect. MiniMAC produces a variable-length tag that fits entirely within available space. As such MiniMAC never adds bits to messages, and it does not delay messages.

The review makes several additional points, some valid, some invalid, but none of these points addresses our main contribution. Instead, these points deal with optional choices or engineering issues (however important) surrounding the central contribution. (1) For example, the review states, *``You're also assuming that every ECU has writable, nonvolatile storage in which it can store counter values or previous messages. This seems unrealistic.’’* Many ECUs, however, have sufficient non-volatile storage for MiniMAC—for example, the TI Hercules device has ample storage.  (2) As another example, the review states, *``You say that ECUs should generate their own keys. There's no way that is going to work.’’* Our manuscript, however, does not require ECUs to have such capability. We offer this keying method as a preferred option, and we describe an alternative when the capability does not exist in the ECU. (3) As a third example, the review states, *``In Section 9.2, you suggest repurposing the CRC field in the CAN frame to add additional bytes to the MAC. You say that this could be implemented by modifying low-level code in the CAN stack. This doesn't work. The CRC is added by the transceiver hardware and is not under the control of software unless CAN is implemented via  
bit-banging--which it isn't in real ECUs.’’* As our manuscript points out, our work does not depend on repurposing the CRC field, and CRC fields are typically added by hardware. We list this repurposing strategy solely as an option, when it is available.

Please feel free to share this correspondence with the reviewer(s).