***Phase 2-2, Efficiency of the method and drawbacks***

In order to achieve confidentiality and prevent data tampering, encrypt-then-MAC is a highly secure solution as the data is first encrypted and then the MAC of the cipher-text is computed. This easily enables the receiving party to reject inauthentic packages in the case where the MAC does not match without having to go through the decryption process. Moreover, Since the data is encrypted before applying a MAC, if an accidental or intentional information leakage is to happen, only the cipher-text would be disclosed. However, this highly secure approach comes with some minor downsides. Most designs that follow the Encrypt-then-MAC approach are exposed to nonce misuse (Dutta, 2019). This can lead to a universal forgery attack due to a single repetition of the nonce value as the hash key is disclosed, increasing the vulnerability of the model. Furthermore, since it is an encrypt-then-authenticate scheme, it requires two different algorithms for encryption and authentication that needs to be implemented separately (Kavun2016). Moreover, additional implementation effort for each algorithm and additional code space would be required which would affect its overall performance.

When it comes to efficiency, faster encryption and message authentications are limited here as both the encryption and the message authentication are performed in a traditional two passes instead of one. This is due to the use of a block cipher as a building block which limits the authenticated encryption performance to a maximum of one message block per block cipher evaluation (Goce Jakimoski and Samant Khajuria, 2011). Furthermore, in the encrypt-then-MAC model, there are 2 mathematically connected keys (one for encryption and another one for decryption) required in the implementation (a public-key and a private- key) compared to other 2 approaches where a single public key is used. This would result in increased complexity and execution times.

Despite all the drawbacks, Encrypt-then-MAC can still be regarded as a secure method to compose a secret key and a MAC into the authenticated encryption. The other two approaches, MAC-and-Encrypt and the Encrypt-then-MAC models do not meet the cipher-text integrity requirement, which can result in attacks like chosen cipher-text attacks. In both approaches, MAC is generated against plain text which can result in a possibility that the MAC can cause a privacy leak of the plain text. The Encrypt-then-MAC approach ensures more protection against both cipher-text attacks and privacy leaks. In addition, encrypt-then-MAC meets the definition of authenticated-encryption and is a very reliable approach to secure the Minecraft API.

***References:***

KAVUN, MIHAJLOSKA, YALÇIN. A Survey on Authenticated Encryption—ASIC Designer’s perspective, 2017. [online] Available at: https://dl-acm-org.ezproxy.lib.rmit.edu.au/doi/pdf/10.1145%2F3131276 [Accessed 6 November 2021].

Goce Jakimoski, Samant Khajuria,. An authenticated encryption Stream Cipher., 2011 [online] Available at: https://link-springer-com.ezproxy.lib.rmit.edu.au/content/pdf/10.1007%2F978-3-642-28496-0.pdf  [Accessed 6 November 2021].

Dutta,Nandi,Talnikar. Advances in Cryptology – EUROCRYPT 2019., [online] Available at: https://link-springer-com.ezproxy.lib.rmit.edu.au/content/pdf/10.1007%2F978-3-030-17653-2.pdf [Accessed 6 November 2021].