**AES-256 is an encryption protocol**.  
Encryption is used for privacy. When a payload (or message) is encrypted, observers who do not have the decryption key, cannot read the message.

**HMAC is not encryption.** [HMAC is "keyed hash"](https://en.wikipedia.org/wiki/HMAC).  It is used for integrity checking and message authentication.   
The HMAC is a specific kind of MAC, aka [Message Authentication Code.](https://en.wikipedia.org/wiki/Message_authentication_code) Basically for each message, there is a unique SHORTER message, that can be derived from the message, and a given shared secret key. So if I send "the quick brown fox jumps over the lazy dog", and the shared secret key is "Secret123", the HMAC (or keyed hash) will be "b995706eefb98d23388ce579fe6cf0f30bf8147f53394850af4ddc7105f6d3c4" (hex encoded). ([check it](https://dinochiesa.github.io/hmachash/index.html)) And a receiver that receives the message, and also possesses the same key, would be able to compute the HMAC and check that the message had not been modified in transit (integrity) and that the sender holds the secret key (authentication).

As you can see, HMAC does not encrypt the message.  With HMAC, you send the message in cleartext along with the HMAC.  The message can be read by any observer.  It's not the same as encryption.

Those are the basics.

You are asking for instructions for how to get the HMAC policy in Apigee to perform encryption.  It won't.  It doesn't.  the HMAC policy computes and checks HMACs. Encryption is different.

If you want encryption, you need to use a different policy, like the builtin [GenerateJWT policy which can encrypt data](https://cloud.google.com/apigee/docs/api-platform/reference/policies/generate-jwt-policy" \l "generate-an-encrypted-jwt" \t "_blank).  Or, the [external custom policy for AES crypto](https://github.com/DinoChiesa/Apigee-CustomPolicy-AesCrypto).

[1 Like](https://www.googlecloudcommunity.com/gc/kudos/messagepage/board-id/cloud-apigee/message-id/80238/tab/all-users)