



Key Size for 25G-EPON and 50G-EPON

Glen Kramer, Broadcom

- ❑ **Should the 1904.4 support only 128-bit keys, only 256-bit keys, or both 128-bit and 256-bit keys?**

- ❑ **How to answer:**

1. Complexity impact of supporting one key size versus two key sizes
2. Cryptographic strength
- ~~3. Performance impact~~
4. Key sizes currently required by other relevant standards
5. Any external constraints operators may have within their COs
6. Design impact:

1. Key memory

Number of keys per OLT ASIC =

#Ports/chip × (#ONUs/port + #multicast_LLIDs/port) × 2 keys/sec. association

2. Latency
3. Power

1. Complexity impact of supporting two key sizes

Complexity impact is minimal

1. Key distribution:

- OLT decides what key size to use
- The same TLV is used by do distribute key of either size

2. Number of key expansion rounds changes from 10 to 14

- In AES-256, the pipeline will have to get deeper to maintain encryption at line rate. This is not a significant cost additive.

Key Value TLV (128-bit key)

Field size	Field value	Field description
1	0xDB	Extended Attributes branch
2	0x04-03	TLV Leaf value
1	17	Length
1	0 or 1	Key Index (only bit 0 is used)
16	Various	Key Value

Key Value TLV (256-bit key)

Field size	Field value	Field description
1	0xDB	Extended Attributes branch
2	0x04-03	TLV Leaf value
1	33	Length
1	0 or 1	Key Index (only bit 0 is used)
32	Various	Key Value

2. Cryptographic strength



- ❑ FIPS 197 provides three variants of AES, each with a different key size: 128, 192, and 256 bits. The 192-bit and 256-bit variants not only have larger key sizes but also require additional iterations of the AES round function — 12 and 14 rounds, respectively — compared to 10 rounds for the 128-bit variant. It is reasonable to expect that the 192-bit and 256-bit variants will always require significantly more computations to attack than the 128-bit variant and may, therefore, provide alternatives in case of significant advances in computing power or cryptanalysis. Currently, however, all three keys sizes are considered secure, and it seems that this will continue to be the case for the foreseeable future.

From **Review of the Advanced Encryption Standard, NISTIR 8319, July 2021**

2. FIPS 197 update



Rationale for Updating FIPS 197

- The technical content is the specification of a family of three block ciphers: AES-128, AES-192, and AES-256, where the numerical suffix indicates the bit length of the key. Since AES is adopted widely, the main question for the review is whether the specified block cipher family is sufficiently secure. The following is a summary of the security assessment:
 - **Classical security** [Sections 3.1 and 3.2 of NISTIR 8319]: The key sizes remain adequate against classical exhaustive search. For the classical analytic attacks on instances of the AES algorithm that are listed in Table 2 of NISTIR 8319, either 1) the attack only applies to an unapproved, weakened variant, in which the number of rounds is reduced by at least three; or 2) the computational complexity of the attack is prohibitive. In the second category, the largest theoretical reduction in computational complexity over generic, exhaustive search occurs under the restrictive assumption of related-keys, and only for AES-192 and AES-256.
 - **Key Size and Post-Quantum Security** [Section 3.3 of NISTIR 8319]: If large-scale quantum computers are developed, Grover's algorithm would facilitate a brute force search for the key with computational work that is roughly the square root of the classical computational work. NIST expects to issue appropriate guidance on parameter choices and post-quantum security as part of the Post-Quantum Cryptography project.

3. Performance Impact

A Comparative Study on AES 128 bit and AES 256 bit. International Journal Of Computer Sciences And Engineering. Vol 6. 30-33, 2018

- ❑ Larger key requires more key expansion rounds
- ❑ But the performance hit shown in this paper seems excessive
 - **This may not be relevant to silicon implementations. Encryption will always run at line rate. May require deeper pipeline**

	Key Length (<i>Nk words</i>)	Block Size (<i>Nb words</i>)	Number of Rounds (<i>Nr</i>)
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

Figure 4. Key-Block-Round Combinations.

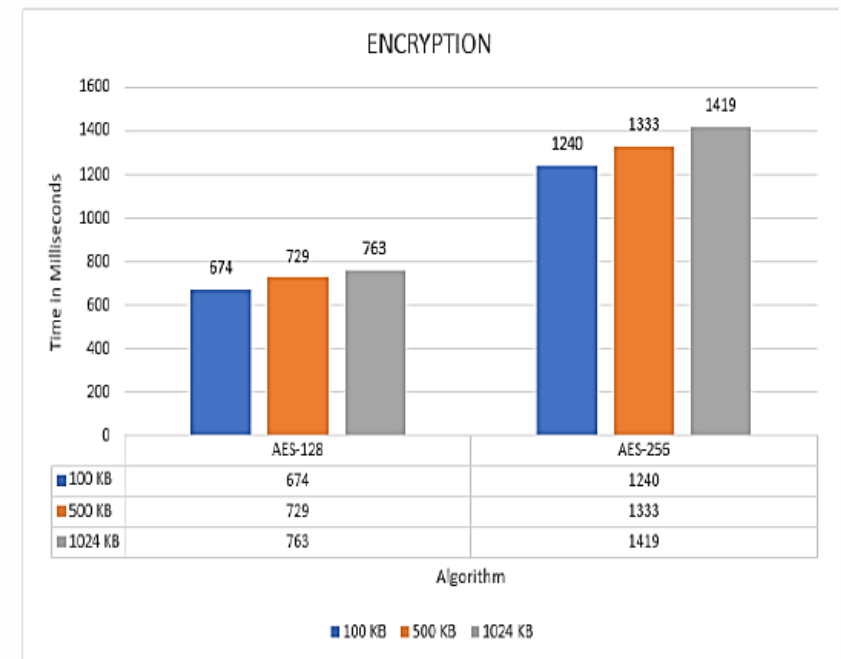


Figure 3. Comparative encryption time of AES-128 and AES-256

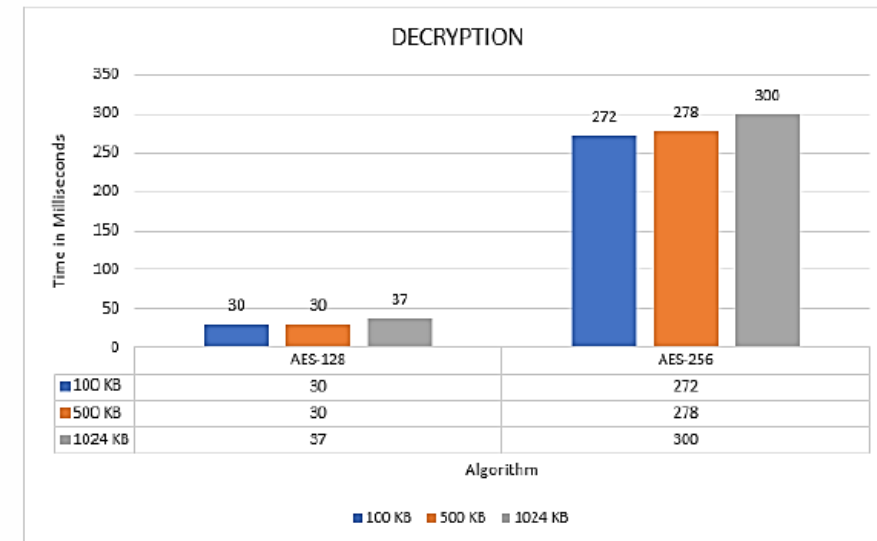


Figure 4. Comparative decryption time of AES-128 and AES-256

4. Key sizes required by other standards

□ IEEE 802.1X

- This standard uses a KDF that is compatible with the counter mode KDF described in the NIST Special Publication 800-108. The KDF uses a pseudorandom function (PRF) which shall be AES-CMAC-128 when the derivation key is 128 bits, and AES-CMAC-256 when the derivation key is 256 bits.

□ ITU-T G.9804.2

- The default algorithm used for XGEM payload encryption is the AES-128 [NIST FIPS-197] cipher, used in Counter mode (AES-CTR), as described in [NIST SP800-38A]. Cipher algorithms AES-128 (mandatory), AES-256 (mandatory), Camellia-128, Camellia-256, and SM4(-128) [ISO/IEC 18033-3:2010] can be configured via the OMCI, as defined in clause 9.1.2 [ITU-T G.988]. The configured cipher algorithm is used for both upstream and downstream, unicast and multicast.
- The default cipher is the advanced encryption standard (AES) encryption algorithm [NIST FIPS-197] with 128 bit key length. The use of additional ciphers (both algorithm and key length) can be configured via the OMCI. Before OMCI configuration occurs, the default cipher will be used.

6. Design Considerations



- Key memory

***Number of keys per OLT ASIC =
#Ports/chip × (#ONUs/port + #multicast_LLIDs/port) × 2 keys/sec. assoc.***

- **Consider an example OLT chip**

- 16 PON ports
- 64 ONUs per port (max)
- 1024 multicast LLIDs per port (max)

- In 25G-EPON, OLT needs to store ~35K keys: **$16 \times (64 + 1024) \times 2 = 34816$**

- Total memory for 256-bit keys = **1.11MB**

- **Compare with 10G-EPON**

- Same OLT parameters, but now, 8 keys per ONU, since each LLID uses a separate key

- In 10G-EPON, OLT needs to store ~50K keys: **$16 \times (64 \times 8 + 1024) \times 2 = 49152$**

- Total memory for 128-bit keys = **0.79MB**

- ❑ Most PON silicon supports multiple generations of PON and multiple flavors (IEEE, ITU-T).
- ❑ AES-128 is the default in ITU-T G.9804.2 and in DPoE 2.0 (and SIEPON Package A, by reference). The new chips are likely to support AES-128 if they support previous generation of PON
- ❑ ITU-T G.9804.2 specifies both AES-128 and AES-256 as mandatory. In a chip that supports both ITU-T (G.hsp) PON and IEEE (Nx25G) PON, not allowing both methods in IEEE PON would simply mean that one of the methods has to be artificially disabled.

There are two questions to consider

1. Can we not support 256-bit key?

- Most standards that specify encryption are adding AES-256 option (MACsec, ITU-t G.9804.2). It is mandatory to support in ITU-T PON.
- Similarly to the arguments voiced for the encryption method, not supporting AES-256 will create a perception that the IEEE 1904.4 is lagging behind the competition.

2. Can we not support 128-bit key?

- The AES-128 remains the default encryption method in ITU-T G.9804.2
- It is the only method in earlier generations of ITU-T and IEEE PON.
- It would not be prudent to abruptly stop supporting the method that is used in most deployed networks today.
- It is conceivable that some operators will opt to maintain the same encryption method on various platforms within their network for some time.
- Most chips will internally support both key sizes because they also support previous generations. We should not have stranded functionality in the chips (implemented, but cannot be enabled)

I prefer the following encryption key sizes for IEEE 1904.4
(vote for one only)

1. Only 128-bit key shall be specified _____

2. Only 256-bit key shall be specified _____

3. 128-bit and 256-bit keys are
mandatory to support;
operator selects the key size to use 5_____

- The IEEE 1904.4 shall support both AES-128 and AES-256
 - Moved: Glen Kramer
 - Second: Marek Hajduczenia
 - (Technical, required $\geq 2/3$)
 - Passed by voice vote without opposition



Thank you