**Notations:**

Enc – encryption

Dec – decryption

AD – associated data

PT – plaintext

CT – ciphertext

Below calculations have been performed based on attached SAEAES and AES (NIST.FIPS.197) specification documents.

For primary variant of SAEAES, AES parameters are: Nk = 4, Nb = 4 and Nr = 10.

Simulation clock period Tclk = 20 ns. (Refer 1\_assumptions\Assumptions)

**Case 1: Base case of empty AD and PT**

Hash:

Diagram, schematic

Description automatically generated

Encryption:

Diagram

Description automatically generated

Decryption:

Diagram

Description automatically generated

Decryption clock cycles calculation is similar to encryption clock cycles calculation for base case λ.

From above figures, Ek algorithm can be replaced with AES Enc.

Overall assumptions:

1. One-time overhead of key expansion
2. 1 clock cycle is consumed for each program internal memory access
3. All intermediate XOR operations occur within the same clock cycle as feeding input or getting output from Ek

Hence. clock cycles for base case encryption/decryption can be calculated as below:

SAEAES Enc/Dec = 3 \* AES Enc

AES Key Expansion

Text

Description automatically generated

Assumptions:

1. Each AES key input is obtained from bdi using SIPO in 4 cycles.
2. A word from key’s bytes elements is obtained in 1 cycle.
3. RotWord and Rcon obtains result in 1 cycle.
4. SubWord takes 1 cycle to obtain result for each word input.
5. Each w[.] element gets updated in 1 cycle inside a while-loop iteration.

Hence, Key Setup Cycles = 4 + (Nk \* 1) + (Nb \* (Nr + 1) – 1 – Nk)

= 4 + 4 + (4 \* 11 – 1 – 4)

= 47

Key Setup Time = 47 \* TClk = 47 \* 20 ns = 940 ns.

AES Enc

Text

Description automatically generated

Before the for-loop, the **state** variable is initialized in 1 clock cycle followed by AddRoundKey function call where each word of **w** for Nb blocks are added to **state** in 1 cycle.

Each iteration of for-loop takes 4 cycles due to update of **state** variable in each instruction. Thus entire for-loop takes 4 \* (Nr – 1) cycles.

After the for-loop, additional 3 cycles are taken to update **state** variable.

Thus AES Encryption Cycles = 1 + 1 + 4 \* (Nr – 1) + 3 = 5 + 4 \* 9 = 41

AES Encryption Time = 41 \* TClk = 41 \* 20 ns = 820 ns.

Finally SAEAES Encryption Time = 3 \* 820 = 2,460 ns

**Total duration to generate CT = Key Setup Time + Encryption Time = 3,400 ns**

**Case 1:**

Simulation results:

Refer Case1\_Vivado\_Console and Case1\_Vivado\_Waveform

Expected Encryption Time = 940 + 3 \* 820 ns. = 3,400 ns

Actual Encryption Time = 2,540 ns

Deviation (savings) = 860 ns = 43 clock cycles

Decryption not applicable due to empty PT.

**Case 2: Non-empty AD and empty PT**

**Case 3: Empty AD and non-empty PT**

**Case 4: Non-empty AD and non-empty PT**

Hash:

A picture containing text, clock

Description automatically generated

Encryption:

Diagram

Description automatically generated

Decryption:

Diagram

Description automatically generated

Due to non-empty AD blocks,

Key Setup Cycles = 4 + (Nk \* 1) + (Nb \* (Nr + 1) – 1 – Nk)

= 4 + 4 + (4 \* 11 – 1 – 4)

= 47

Hence Key Setup Time = 940 ns.

Assumptions of Case 1 apply to current case as well.

Encryption and decryption durations change due to non-empty AD, PT and CT inputs.

Let number of AD blocks = a, PT/CT blocks = p.

Number of times Ek is called for Hash = a

Number of times Ek is called for Encryption/Decryption = p + 1

Similar to Case 1, assuming all intermediate input and output processing steps for an Ek block to take place in the first or last cycle of that Ek block,

SAEAES Encryption/Decryption Cycles

= Key Setup Cycles + (a + p + 1) \* AES Encryption Cycles

= 47 + (a + p + 1) \* 41

**SAEAES Encryption/Decryption Time = 940 + (a + p + 1) \* 820 ns.**

SAEAES Throughput:

T – clock period in µs, f – clock frequency in MHz, throughput in Mbits/s

a = 1, p = 1, TClk = T

Encryption/Decryption Cycles = 47 + 3 \* 41 = 170

Number of CT bits = 64

Number of Tag bits = 128

Total output bits = 64 + 128 = 192

**Throughput = 192 / 170 \* T = 1.13 \* f**

**Case 2:**

Simulation results:

Refer Case2\_Vivado\_Console and Case2\_Vivado\_Waveform

Expected Encryption Time = 940 + (2 + 0 + 1) \* 820 ns. = 3,400 ns

Actual Encryption Time = 3,500 ns

Deviation (delay) = 100 ns = 5 clock cycles

Decryption not applicable due to empty PT.

**Case 3:**

Simulation results:

Refer Case3\_Vivado\_Console and Case3\_Vivado\_Waveform

Expected Encryption/Decryption Time = 940 + (0 + 2 + 1) \* 820 ns. = 3,400 ns

Actual Encryption Time = 3400 ns

Deviation = 0 ns = 0 clock cycles

Actual Decryption Time = 4240 – 3825 + 2980 ns = 3,395 ns

Deviation (savings) = 5 ns = 0.25 clock cycles

**Case 4:**

Simulation results:

Refer Case4\_Vivado\_Console and Case4\_Vivado\_Waveform

Expected Encryption/Decryption Time = 940 + (2 + 2 + 1) \* 820 ns. = 5,040 ns

Actual Encryption Time (till Tag calculation) = 5240 ns

Deviation (delay) = 200 ns = 10 clock cycles

Actual Decryption Time (till Tag verification) = 6100 – 5245 + 4380 ns = 5,235 ns

Deviation (delay) = 195 ns = 9.75 clock cycles