AES256GCM10G25G IP Demo Instruction on KR260

Table of Contents

AES256GCM10G25G IP Demo Instruction on KR260

Rev1.00 12-Jul-2024

This document describes the instruction to demonstrate the operation of AES256GCM10G25GIP on Kria KR260 Robotics Starter Kit. In the demonstration, AES256GCM10G25GIP is used to encrypt/decrypt data between two memories in FPGA and provide authentication tag. User can fill memory with Additional Authenticated Data (AAD), DataIn patterns, set encryption/decryption key, Initialization Vector (IV), and control test operation via serial console.

1 Environment Setup

To operate AES256GCM10G25GIP demo, please prepare following test environment.

1) FPGA development boards (KR260 board).
2) Host PC.
3) Micro USB cable for UART connection connecting between KR260 board and Host PC.
4) Serial console software such as TeraTerm installed on PC. The setting on the console is Baud rate=115200, Data=8-bit, Non-parity and Stop=1.
5) Test application provided by Design Gateway for running on KR260
   - Application folder named “AES256GCM10G25GIP”.
   - Demo software named “AES256GCM10G25GDEMO”.

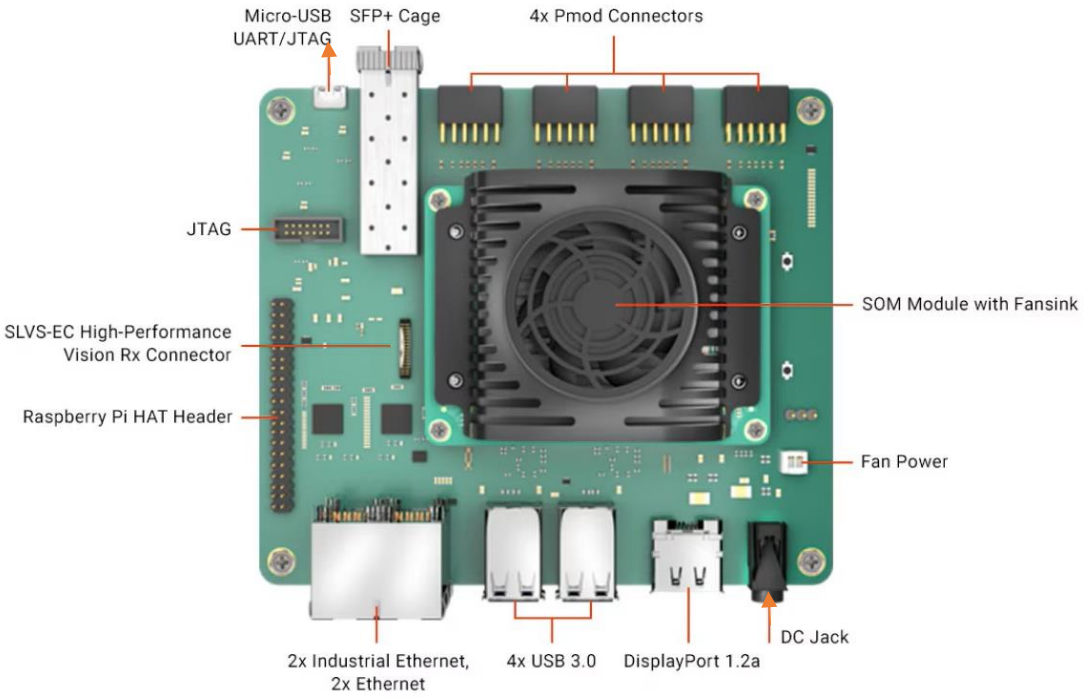

Figure 1-1 AES256GCM10G25GIP demo environment on KR260 board

Linux OS setup (Ubuntu for KR260)

1) Follow the instructions on the AMD setup guide to download and prepare the SD card image. (https://www.amd.com/en/products/system-on-modules/kria/k26/kr260-robotics-starter-kit/getting-started/setting-up-the-sd-card-image.html).
2) After flashing image to SD card, user can see Linux file system as Figure 1-2.

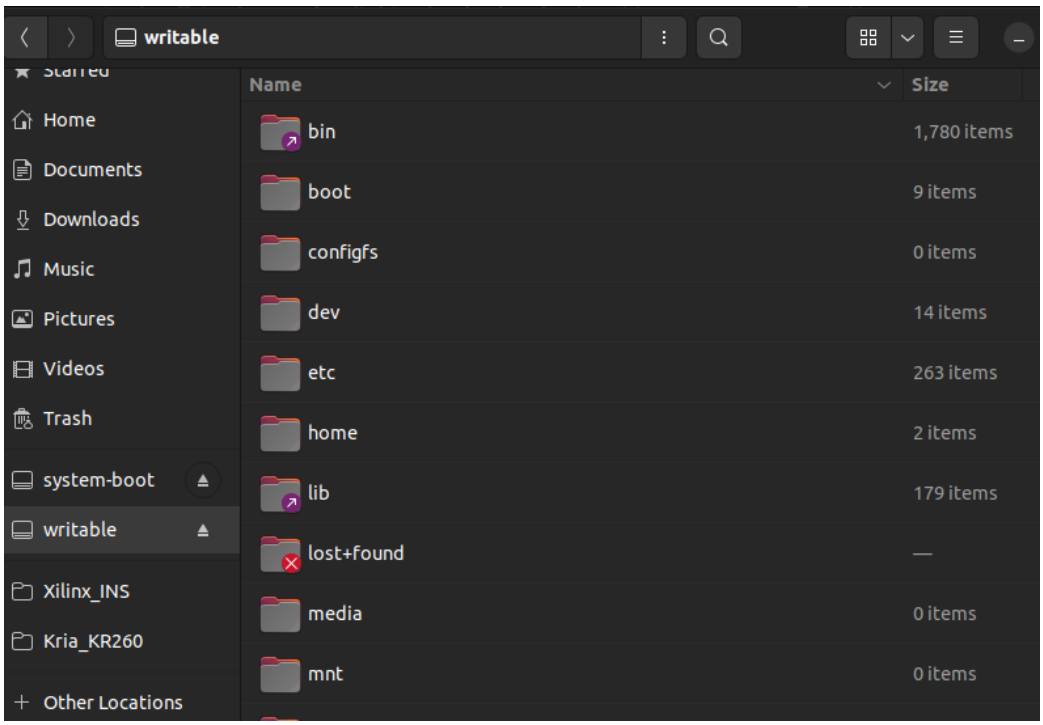

Figure 1-2 Example of Linux file system on SD card

3) Copy folder “AES256GCM10G25GIP” from our demo package to “/lib/firmware/xilinx” on SD card.

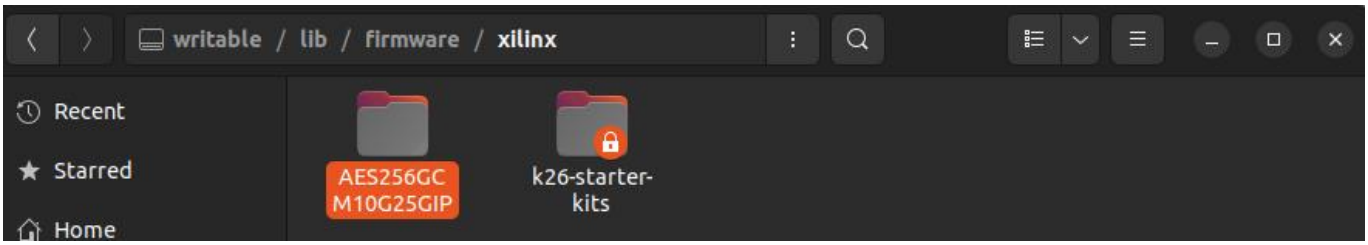

Figure 1-3 "/lib/firmware/xilinx" directory on SD card

4) Copy software “AES256GCM10G25GDEMO” from our demo package to “/home” on SD card.

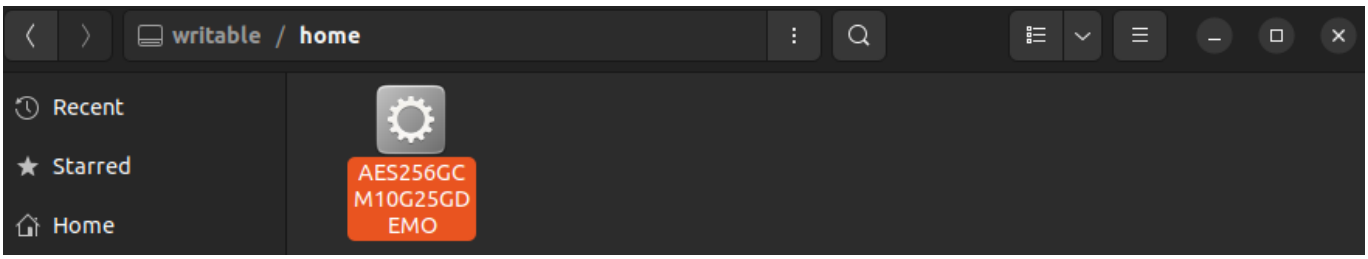

Figure 1-4 "/home" directory on SD card

5) Remove SD card from PC, then insert SD card into the slot located under the KR260 board.
6) Connect the power supply to the FPGA development board. The board will automatically boot into Ubuntu as shown in Figure 1-5.

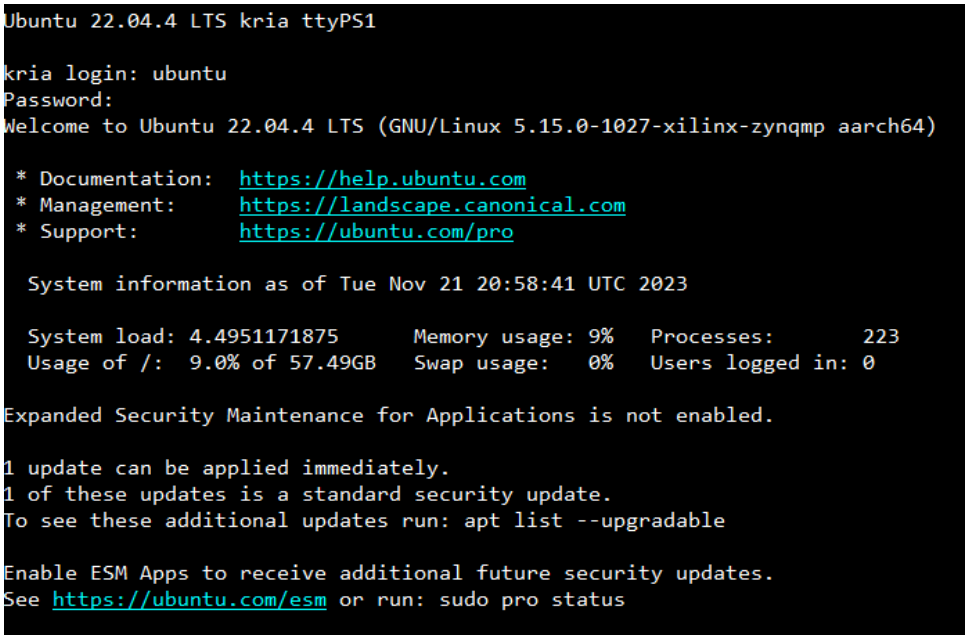

Figure 1-5 KR260 board booting into Ubuntu

KR260 loadapp

To activate AES256GCM10G25G accelerator, if there is already another accelerator/firmware being activated, user must unload it first and then switch to AES256GCM10G25G accelerator as follows.

1) Unload the default hardware application using command below.

   sudo xmutil unloadapp

2) Load new hardware application by using command below. The Figure 1-6 shows example result for loading application on KR260 board.

   sudo xmutil loadapp AES256GCM10G25GIP

```
ubuntu@kria:~$ sudo xmutil unloadapp
remove from slot 0 returns: 0 (Ok)
ubuntu@kria:~$ sudo xmutil loadapp AES256GCM10G25GIP
[  418.937542] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /fpga-full/firmware-name
[  418.947691] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /fpga-full/resets
[  418.957393] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/overlay0
[  418.967248] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/overlay1
[  418.977091] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/afi0
[  418.986584] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/clocking0
[  418.996515] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/overlay2
[  419.006349] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/AES256GCM10G25GIP
[  419.016969] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/axi_intc_0
[  419.035020] zocl-drm axi:zyxclmm_drm: IRQ index 32 not found
AES256GCM10G25GIP: loaded to slot 0
ubuntu@kria:~$
```

Figure 1-6 Example result for loading application on KR260 board

2 AES256GCM10G25G Demo

To run the AES256GCM10G25GDEMO, use the command “sudo ./AES256GCM10G25GDEMO”. This will display the AES256GCM10G25G demo command menu as shown in Figure 2-1. Through this menu, users can fill RAMs with additional authenticated data, plain or cipher data patterns, set encryption/decryption keys, initialize vectors (IV), and control test operations via the serial console. Detailed information on each menu is described in topic 3.

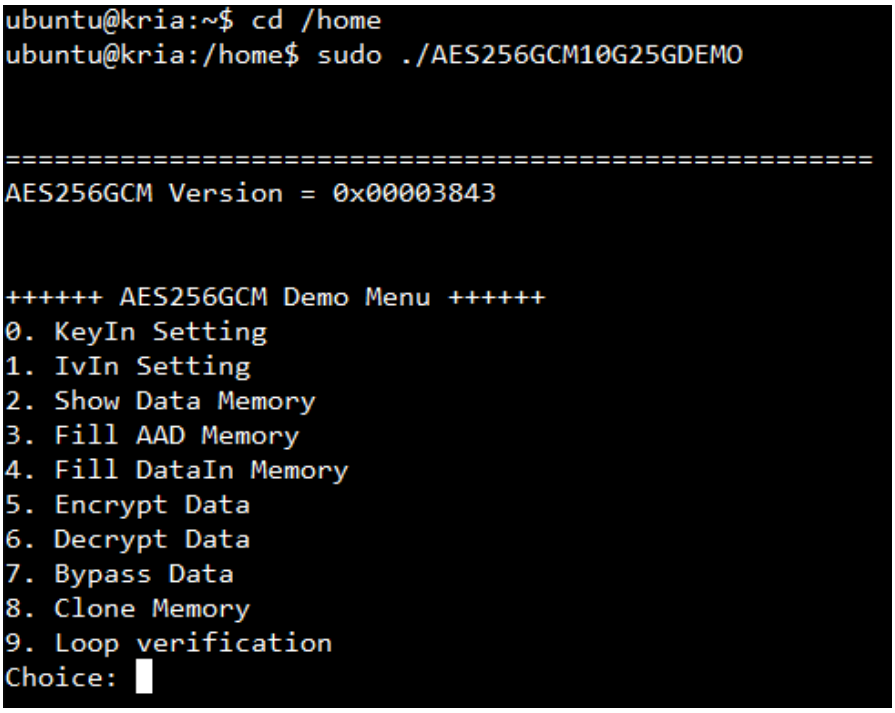

Figure 2-1 Serial console

3 Command detail and testing result

3.1 KeyIn Setting

Step to set key as follows

a) Select “KeyIn Setting”.

b) Current key will be displayed on serial console as shown in Figure 3-1.

c) Set new key: User is allowed to input new key in hex format or press “enter” to skip setting new key. Then the current key is printed again.

```
++++++ AES256GCM Demo Menu ++++++
0. KeyIn Setting
1. IvIn Setting
2. Show Data Memory
3. Fill AAD Memory
4. Fill DataIn Memory
5. Encrypt Data
6. Decrypt Data
7. Bypass Data
8. Clone Memory
9. Loop verification
Choice: 0

+++ KeyIn Setting +++
                KeyIn= 0x0000000000000000000000000000000000000000000000000000000000000000
(enter to use KeyIn)= 0x00112233445566778899aabbccddeeff00112233445566778899aabbccddeeff
          new KeyIn= 0x00112233445566778899AABBCCDDEEFF00112233445566778899AABBCCDDEEFF
```

Figure 3-1 KeyIn setting example

3.2 IvIn Setting

Step to set IV as follows

a) Select “IvIn Setting”.

b) Current IV will be displayed on serial console as shown in Figure 3-2.

c) Set new IV: User is allowed to input new IV in hex format or press “enter” to skip setting new IV. Then the current IV is printed again.

```
++++++ AES256GCM Demo Menu ++++++
0. KeyIn Setting
1. IvIn Setting
2. Show Data Memory
3. Fill AAD Memory
4. Fill DataIn Memory
5. Encrypt Data
6. Decrypt Data
7. Bypass Data
8. Clone Memory
9. Loop verification
Choice: 1

+++ IvIn Setting +++
               IvIn= 0x000000000000000000000000
(enter to use IvIn)= 0x1001200f0011000f20003400
          new IvIn= 0x1001200F0011000F20003400
```

Figure 3-2 IvIn setting example

3.3 Show Data Memory

To show data in memory, user can select “Show Data Memory”. User can input the desired length of data in byte to show. The data length will be aligned to 128 bits. DataIn and DataOut will be displayed in table-form as shown in Figure 3-3. User can press “enter” to use 80 bytes as default value.

```
++++++ AES256GCM Demo Menu ++++++
0. KeyIn Setting
1. IvIn Setting
2. Show Data Memory
3. Fill AAD Memory
4. Fill DataIn Memory
5. Encrypt Data
6. Decrypt Data
7. Bypass Data
8. Clone Memory
9. Loop verification
Choice: 2

+++ Show Data Memory +++
Number of Data in byte (enter = 80):

                DataIn Memory                          DataOut Memory
Addr#  .0.....3 .4.....7 .8.....B .C.....F  .0.....3 .4.....7 .8.....B .C.....F
0000:  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
0001:  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
0002:  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
0003:  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
0004:  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
```

Figure 3-3 Displayed data when input the desired length of data

3.4 Fill AAD Memory

Step to set AAD as follows

a) Select “Fill AAD Memory”.

b) Input the desired length of AAD in byte. In case of zero-length AAD operation, user can input “0” or press “enter” then end process of this menu. In case of non-zero-length AAD, user can select AAD pattern as shown in Figure 3-4.

c) There are four pattern to fill AAD memory.

   a. zero pattern

   b. 8-bit counter

   c. 16-bit counter

   d. 32-bit counter

d) AAD memory will be filled with selected pattern by the number of AAD and zero-padding to become 128-bit padded data.

```
++++++ AES256GCM Demo Menu ++++++
0. KeyIn Setting
1. IvIn Setting
2. Show Data Memory
3. Fill AAD Memory
4. Fill DataIn Memory
5. Encrypt Data
6. Decrypt Data
7. Bypass Data
8. Clone Memory
9. Loop verification
Choice: 3

+++ Fill AAD Memory +++
Length of AAD in byte (enter = 0): 123
Choose AAD pattern
a. zero pattern
b. 8-bit counter
c. 16-bit counter
d. 32-bit counter
Choice: b

                DataIn Memory                       DataOut Memory
Addr#   .0.....3 .4.....7 .8.....B .C.....F  .0.....3 .4.....7 .8.....B .C.....F
0000:   00010203 04050607 08090A0B 0C0D0E0F  00000000 00000000 00000000 00000000
0001:   10111213 14151617 18191A1B 1C1D1E1F  00000000 00000000 00000000 00000000
0002:   20212223 24252627 28292A2B 2C2D2E2F  00000000 00000000 00000000 00000000
0003:   30313233 34353637 38393A3B 3C3D3E3F  00000000 00000000 00000000 00000000
0004:   40414243 44454647 48494A4B 4C4D4E4F  00000000 00000000 00000000 00000000
0005:   50515253 54555657 58595A5B 5C5D5E5F  00000000 00000000 00000000 00000000
0006:   60616263 64656667 68696A6B 6C6D6E6F  00000000 00000000 00000000 00000000
0007:   70717273 74757677 78797A00 00000000  00000000 00000000 00000000 00000000
```

Figure 3-4 Displayed data when set AAD pattern

3.5 Fill DataIn Memory

Step to fill DataIn in memory as follows

a) Select “Fill DataIn Memory”.

b) Input the desired length of data in byte. In case of zero-length DataIn operation, user can input “0” or press “enter” on keyboard then end process of this menu. In case of non-zero-length DataIn, user can select data pattern.

c) There are four pattern to fill memory.

   a. zero pattern

   b. 8-bit counter

   c. 16-bit counter

   d. 32-bit counter

d) Whole DataIn memory is filled with selected pattern after AAD according to the number of input data length as displayed in Figure 3-5.

```
++++++ AES256GCM Demo Menu ++++++
0. KeyIn Setting
1. IvIn Setting
2. Show Data Memory
3. Fill AAD Memory
4. Fill DataIn Memory
5. Encrypt Data
6. Decrypt Data
7. Bypass Data
8. Clone Memory
9. Loop verification
Choice: 4

+++ Fill DataIn Memory +++
Length of DataIn in byte (enter = 0): 112
Choose DataIn pattern
a. zero pattern
b. 8-bit counter
c. 16-bit counter
d. 32-bit counter
Choice: c

                DataIn Memory                          DataOut Memory
Addr#  .0.....3 .4.....7 .8.....B .C.....F  .0.....3 .4.....7 .8.....B .C.....F
0000:  00010203 04050607 08090A0B 0C0D0E0F  00000000 00000000 00000000 00000000
0001:  10111213 14151617 18191A1B 1C1D1E1F  00000000 00000000 00000000 00000000
0002:  20212223 24252627 28292A2B 2C2D2E2F  00000000 00000000 00000000 00000000
0003:  30313233 34353637 38393A3B 3C3D3E3F  00000000 00000000 00000000 00000000
0004:  40414243 44454647 48494A4B 4C4D4E4F  00000000 00000000 00000000 00000000
0005:  50515253 54555657 58595A5B 5C5D5E5F  00000000 00000000 00000000 00000000
0006:  60616263 64656667 68696A6B 6C6D6E6F  00000000 00000000 00000000 00000000
0007:  70717273 74757677 78797A00 00000000  00000000 00000000 00000000 00000000
0008:  00000001 00020003 00040005 00060007  00000000 00000000 00000000 00000000
0009:  00080009 000A000B 000C000D 000E000F  00000000 00000000 00000000 00000000
000A:  00100011 00120013 00140015 00160017  00000000 00000000 00000000 00000000
000B:  00180019 001A001B 001C001D 001E001F  00000000 00000000 00000000 00000000
000C:  00200021 00220023 00240025 00260027  00000000 00000000 00000000 00000000
000D:  00280029 002A002B 002C002D 002E002F  00000000 00000000 00000000 00000000
000E:  00300031 00320033 00340035 00360037  00000000 00000000 00000000 00000000
```

Figure 3-5 Displayed data when set DataIn length and data pattern

3.6 Encrypt Data

Select “Encrypt Data” to encrypt DataIn in memory. Current length of AAD and length of DataIn are printed on serial console. When the encryption process is finished, both DataIn and DataOut will be displayed in table-form and 128-bit encryption tag will be printed as shown in Figure 3-6.

```
++++++ AES256GCM Demo Menu ++++++
0. KeyIn Setting
1. IvIn Setting
2. Show Data Memory
3. Fill AAD Memory
4. Fill DataIn Memory
5. Encrypt Data
6. Decrypt Data
7. Bypass Data
8. Clone Memory
9. Loop verification
Choice: 5

+++ Encrypt Data +++
Length of encrypt-AAD : 123 byte
Length of encrypt-Data : 112 byte

                DataIn Memory                       DataOut Memory
Addr#  .0.....3 .4.....7 .8.....B .C.....F  .0.....3 .4.....7 .8.....B .C.....F
0000:  00010203 04050607 08090A0B 0C0D0E0F  00010203 04050607 08090A0B 0C0D0E0F
0001:  10111213 14151617 18191A1B 1C1D1E1F  10111213 14151617 18191A1B 1C1D1E1F
0002:  20212223 24252627 28292A2B 2C2D2E2F  20212223 24252627 28292A2B 2C2D2E2F
0003:  30313233 34353637 38393A3B 3C3D3E3F  30313233 34353637 38393A3B 3C3D3E3F
0004:  40414243 44454647 48494A4B 4C4D4E4F  40414243 44454647 48494A4B 4C4D4E4F
0005:  50515253 54555657 58595A5B 5C5D5E5F  50515253 54555657 58595A5B 5C5D5E5F
0006:  60616263 64656667 68696A6B 6C6D6E6F  60616263 64656667 68696A6B 6C6D6E6F
0007:  70717273 74757677 78797A00 00000000  70717273 74757677 78797A00 00000000
0008:  00000001 00020003 00040005 00060007  DDF0CA11 4C764E96 86BE4884 96BDCDBF
0009:  00080009 000A000B 000C000D 000E000F  7042B8F5 E7992D9D 7E05B475 BCFAE8A0
000A:  00100011 00120013 00140015 00160017  404C4651 0009B5EC FC8DE8D5 4A474C9C
000B:  00180019 001A001B 001C001D 001E001F  A8C9D384 D9D9AF2E BCDAC47C 56D4D92E
000C:  00200021 00220023 00240025 00260027  61B102ED 06055796 7FB29D51 B7D7B39E
000D:  00280029 002A002B 002C002D 002E002F  A6BF1270 D6CD8386 87C0E35B EB06EB91
000E:  00300031 00320033 00340035 00360037  8BDDCDD5 AD42B614 7FA7BFBB 3EAD73F9


Tag : 32E2954A01B49F9D94C8FE237A510D36
```

Figure 3-6 Serial console after finished encryption process

3.7 Decrypt Data

Select “Decrypt Data” to decrypt DataIn in memory. Current length of AAD and length of DataIn are printed on serial console. When the decryption process is finished, both DataIn and DataOut will be displayed in table-form and 128-bit decryption tag will be printed as shown in Figure 3-7.

```
++++++ AES256GCM Demo Menu ++++++
0. KeyIn Setting
1. IvIn Setting
2. Show Data Memory
3. Fill AAD Memory
4. Fill DataIn Memory
5. Encrypt Data
6. Decrypt Data
7. Bypass Data
8. Clone Memory
9. Loop verification
Choice: 6

+++ Decrypt Data +++
Length of decrypt-AAD : 123 byte
Length of decrypt-Data : 112 byte

                DataIn Memory                       DataOut Memory
Addr#  .0.....3 .4.....7 .8.....B .C.....F  .0.....3 .4.....7 .8.....B .C.....F
0000:  00010203 04050607 08090A0B 0C0D0E0F  00010203 04050607 08090A0B 0C0D0E0F
0001:  10111213 14151617 18191A1B 1C1D1E1F  10111213 14151617 18191A1B 1C1D1E1F
0002:  20212223 24252627 28292A2B 2C2D2E2F  20212223 24252627 28292A2B 2C2D2E2F
0003:  30313233 34353637 38393A3B 3C3D3E3F  30313233 34353637 38393A3B 3C3D3E3F
0004:  40414243 44454647 48494A4B 4C4D4E4F  40414243 44454647 48494A4B 4C4D4E4F
0005:  50515253 54555657 58595A5B 5C5D5E5F  50515253 54555657 58595A5B 5C5D5E5F
0006:  60616263 64656667 68696A6B 6C6D6E6F  60616263 64656667 68696A6B 6C6D6E6F
0007:  70717273 74757677 78797A00 00000000  70717273 74757677 78797A00 00000000
0008:  00000001 00020003 00040005 00060007  DDF0CA11 4C764E96 86BE4884 96BDCDBF
0009:  00080009 000A000B 000C000D 000E000F  7042B8F5 E7992D9D 7E05B475 BCFAE8A0
000A:  00100011 00120013 00140015 00160017  404C4651 0009B5EC FC8DE8D5 4A474C9C
000B:  00180019 001A001B 001C001D 001E001F  A8C9D384 D9D9AF2E BCDAC47C 56D4D92E
000C:  00200021 00220023 00240025 00260027  61B102ED 06055796 7FB29D51 B7D7B39E
000D:  00280029 002A002B 002C002D 002E002F  A6BF1270 D6CD8386 87C0E35B EB06EB91
000E:  00300031 00320033 00340035 00360037  8BDDCDD5 AD42B614 7FA7BFBB 3EAD73F9


Tag : 38D40A66DE0401DA37C47D215A4FF9C4
```

Figure 3-7 Serial console after finished decryption process

3.8 Bypass Data

Select “Bypass Data” to Bypass DataIn in memory. Current length of AAD and length of DataIn are printed on serial console. When the Bypass process is finished, both DataIn and DataOut will be displayed in table-form as shown in Figure 3-8.

```
++++++ AES256GCM Demo Menu ++++++
0. KeyIn Setting
1. IvIn Setting
2. Show Data Memory
3. Fill AAD Memory
4. Fill DataIn Memory
5. Encrypt Data
6. Decrypt Data
7. Bypass Data
8. Clone Memory
9. Loop verification
Choice: 7

+++ Bypass Data +++
Length of decrypt-AAD : 123 byte
Length of decrypt-Data : 112 byte

                DataIn Memory                       DataOut Memory
Addr#  .0.....3 .4.....7 .8.....B .C.....F  .0.....3 .4.....7 .8.....B .C.....F
0000:  00010203 04050607 08090A0B 0C0D0E0F  00010203 04050607 08090A0B 0C0D0E0F
0001:  10111213 14151617 18191A1B 1C1D1E1F  10111213 14151617 18191A1B 1C1D1E1F
0002:  20212223 24252627 28292A2B 2C2D2E2F  20212223 24252627 28292A2B 2C2D2E2F
0003:  30313233 34353637 38393A3B 3C3D3E3F  30313233 34353637 38393A3B 3C3D3E3F
0004:  40414243 44454647 48494A4B 4C4D4E4F  40414243 44454647 48494A4B 4C4D4E4F
0005:  50515253 54555657 58595A5B 5C5D5E5F  50515253 54555657 58595A5B 5C5D5E5F
0006:  60616263 64656667 68696A6B 6C6D6E6F  60616263 64656667 68696A6B 6C6D6E6F
0007:  70717273 74757677 78797A00 00000000  70717273 74757677 78797A00 00000000
0008:  00000001 00020003 00040005 00060007  00000001 00020003 00040005 00060007
0009:  00080009 000A000B 000C000D 000E000F  00080009 000A000B 000C000D 000E000F
000A:  00100011 00120013 00140015 00160017  00100011 00120013 00140015 00160017
000B:  00180019 001A001B 001C001D 001E001F  00180019 001A001B 001C001D 001E001F
000C:  00200021 00220023 00240025 00260027  00200021 00220023 00240025 00260027
000D:  00280029 002A002B 002C002D 002E002F  00280029 002A002B 002C002D 002E002F
000E:  00300031 00320033 00340035 00360037  00300031 00320033 00340035 00360037
```

Figure 3-8 Serial console after finished Bypass process

3.9 Clone Memory

Select “Clone Memory” for copy DataOut memory to DataIn memory. When the process is finished, both DataIn and DataOut will be displayed in table-form as shown in Figure 3-9.

```
++++++ AES256GCM Demo Menu ++++++
0. KeyIn Setting
1. IvIn Setting
2. Show Data Memory
3. Fill AAD Memory
4. Fill DataIn Memory
5. Encrypt Data
6. Decrypt Data
7. Bypass Data
8. Clone Memory
9. Loop verification
Choice: 8

+++ Clone Memory +++

                DataIn Memory                       DataOut Memory
Addr#   .0.....3 .4.....7 .8.....B .C.....F   .0.....3 .4.....7 .8.....B .C.....F
0000:   00010203 04050607 08090A0B 0C0D0E0F   00010203 04050607 08090A0B 0C0D0E0F
0001:   10111213 14151617 18191A1B 1C1D1E1F   10111213 14151617 18191A1B 1C1D1E1F
0002:   20212223 24252627 28292A2B 2C2D2E2F   20212223 24252627 28292A2B 2C2D2E2F
0003:   30313233 34353637 38393A3B 3C3D3E3F   30313233 34353637 38393A3B 3C3D3E3F
0004:   40414243 44454647 48494A4B 4C4D4E4F   40414243 44454647 48494A4B 4C4D4E4F
0005:   50515253 54555657 58595A5B 5C5D5E5F   50515253 54555657 58595A5B 5C5D5E5F
0006:   60616263 64656667 68696A6B 6C6D6E6F   60616263 64656667 68696A6B 6C6D6E6F
0007:   70717273 74757677 78797A00 00000000   70717273 74757677 78797A00 00000000
0008:   00000001 00020003 00040005 00060007   00000001 00020003 00040005 00060007
0009:   00080009 000A000B 000C000D 000E000F   00080009 000A000B 000C000D 000E000F
000A:   00100011 00120013 00140015 00160017   00100011 00120013 00140015 00160017
000B:   00180019 001A001B 001C001D 001E001F   00180019 001A001B 001C001D 001E001F
000C:   00200021 00220023 00240025 00260027   00200021 00220023 00240025 00260027
000D:   00280029 002A002B 002C002D 002E002F   00280029 002A002B 002C002D 002E002F
000E:   00300031 00320033 00340035 00360037   00300031 00320033 00340035 00360037
```

Figure 3-9 Serial console after finished Clone Memory process

3.10 Loop verification

Select “Loop verification”, to check both encryption and decryption. In this menu, DataIn in memory will be encrypted/decrypted with all current parameters (key, IV, AAD and data in DataIn memory).

The function begins by read and store data from the DataIn memory as an original data and clear the DataOut memory before encryption, then start encryption process. After the encryption is completed, the data from the DataOut memory is cloned to the DataIn memory and decryption process is performed. Once the decryption is completed, the decrypted data is compared with the original data, and the encryption tag is compared with the decryption tag.

If the decrypted data and decryption tag match with original data and encryption tag, respectively, “Loop verification succeeded.” is printed as shown in Figure 3-10.

```
++++++ AES256GCM Demo Menu ++++++
0. KeyIn Setting
1. IvIn Setting
2. Show Data Memory
3. Fill AAD Memory
4. Fill DataIn Memory
5. Encrypt Data
6. Decrypt Data
7. Bypass Data
8. Clone Memory
9. Loop verification
Choice: 9

+++ Loop verification +++

KeyIn= 0x00112233445566778899AABBCCDDEEFF00112233445566778899AABBCCDDEEFF
 IvIn= 0x1001200F0011000F20003400
Length of encrypt-AAD  : 62 byte
Length of encrypt-Data : 56 byte

                Original Data                          Encrypted Data
Addr#   .0.....3 .4.....7 .8.....B .C.....F   .0.....3 .4.....7 .8.....B .C.....F
0000:   00010203 04050607 08090A0B 0C0D0E0F   00010203 04050607 08090A0B 0C0D0E0F
0001:   10111213 14151617 18191A1B 1C1D1E1F   10111213 14151617 18191A1B 1C1D1E1F
0002:   20212223 24252627 28292A2B 2C2D2E2F   20212223 24252627 28292A2B 2C2D2E2F
0003:   30313233 34353637 38393A3B 3C3D0000   30313233 34353637 38393A3B 3C3D0000
0004:   00000001 00020003 00040005 00060007   DDF0CA11 4C764E96 86BE4884 96BDCDBF
0005:   00080009 000A000B 000C000D 000E000F   7042B8F5 E7992D9D 7E05B475 BCFAE8A0
0006:   00100011 00120013 00140015 00160017   404C4651 0009B5EC FC8DE8D5 4A474C9C
0007:   00180019 001A001B 00000000 00000000   A8C9D384 D9D9AF2E 00000000 00000000

Encrypted Tag : 404544F835F7E98DF1376D210D48FF2A

               Encrypted Data                          Decrypted Data
Addr#   .0.....3 .4.....7 .8.....B .C.....F   .0.....3 .4.....7 .8.....B .C.....F
0000:   00010203 04050607 08090A0B 0C0D0E0F   00010203 04050607 08090A0B 0C0D0E0F
0001:   10111213 14151617 18191A1B 1C1D1E1F   10111213 14151617 18191A1B 1C1D1E1F
0002:   20212223 24252627 28292A2B 2C2D2E2F   20212223 24252627 28292A2B 2C2D2E2F
0003:   30313233 34353637 38393A3B 3C3D0000   30313233 34353637 38393A3B 3C3D0000
0004:   DDF0CA11 4C764E96 86BE4884 96BDCDBF   00000001 00020003 00040005 00060007
0005:   7042B8F5 E7992D9D 7E05B475 BCFAE8A0   00080009 000A000B 000C000D 000E000F
0006:   404C4651 0009B5EC FC8DE8D5 4A474C9C   00100011 00120013 00140015 00160017
0007:   A8C9D384 D9D9AF2E 00000000 00000000   00180019 001A001B 00000000 00000000

Decrypted Tag : 404544F835F7E98DF1376D210D48FF2A
Loop verification succeeded.
```

Figure 3-10 Serial console after loop verification is succeeded

4 Revision History

Revision	Date	Description
1.00	12-Jul-24	Initial version release