AES256GCM10G25G IP Demo Instruction

Rev1.02 20-Sep-2022

This document describes the instruction to demonstrate the operation of AES256GCM10G25GIP on ZCU106 Evaluation Board. In the demonstration, AES256GCM10G25GIPs are used to encrypt and decrypt data between two memories in FPGA and provide authentication tag. User can fill memory with Additional Authenticated Data (AAD), plain or cipher data 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 (ZCU106 board).
2) Test PC.
3) Micro USB cable for JTAG connection connecting between ZCU106 board and Test PC.
4) Micro USB cable for UART connection connecting between ZCU106 board and Test PC.
5) Vivado tool for programming FPGA installed on Test PC.
6) Serial console software such as TeraTerm installed on PC. The setting on the console is Baudrate=115,200, Data=8-bit, Non-parity and Stop=1.
7) Batch file named "AES256GCM10G25GIPTest_ZCU106.bat". (To download these files, please visit our web site at www.design-gateway.com)

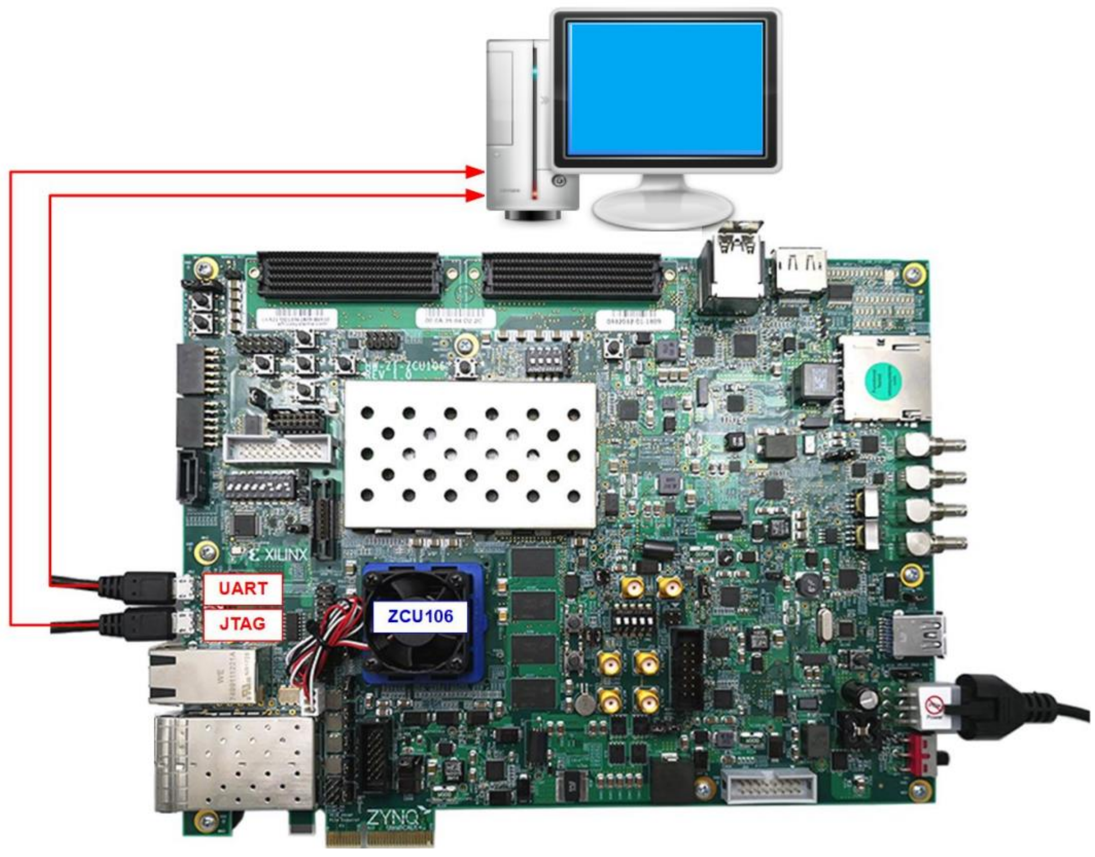
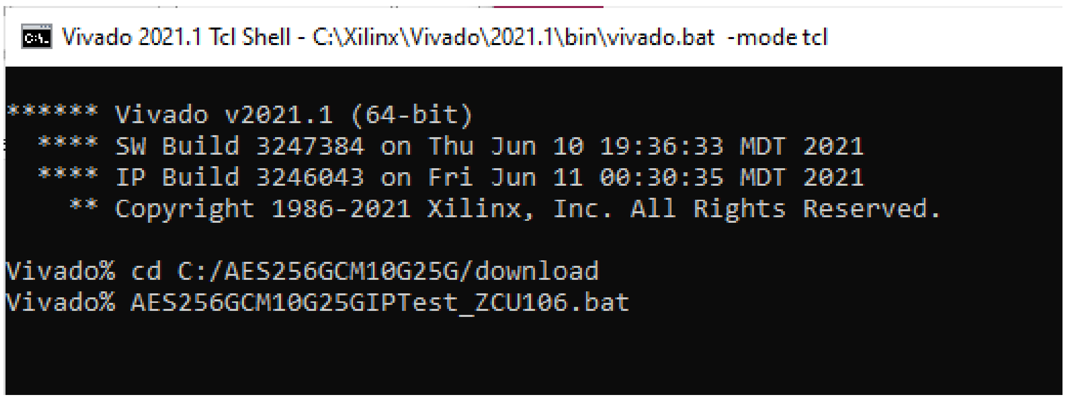


Figure 1-1 AES256GCM10G25GIP demo environment on ZCU106 board

ZCU106 board setup

1) Make sure power switch is off and connect power supply to FPGA development board.
2) Connect two USB cables between FPGA board and PC via micro USB ports.
3) Power on system.
4) Download configuration file and firmware to FPGA board by following step,
   a) Open Vivado TCL shell.
   b) Change current directory to download folder which includes demo configuration file.
   c) Type "AES256GCM10G25GIPTest_ZCU106.bat", as shown in Figure 1-2.

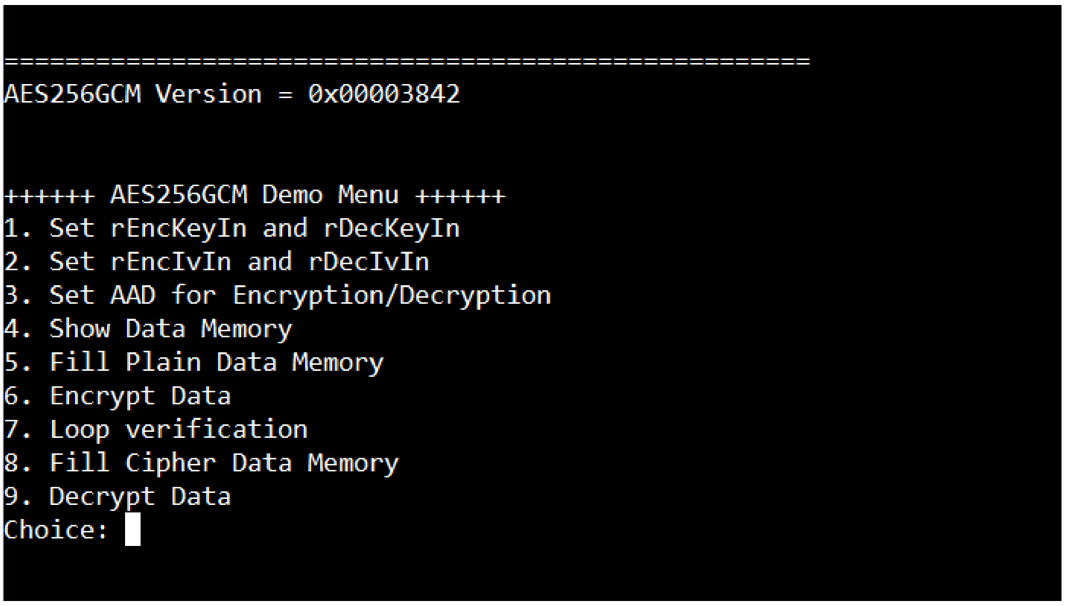


```
Vivado 2021.1 Tcl Shell - C:\Xilinx\Vivado\2021.1\bin\vivado.bat -mode tcl

****** Vivado v2021.1 (64-bit)
  **** SW Build 3247384 on Thu Jun 10 19:36:33 MDT 2021
  **** IP Build 3246043 on Fri Jun 11 00:30:35 MDT 2021
    ** Copyright 1986-2021 Xilinx, Inc. All Rights Reserved.

Vivado% cd C:/AES256GCM10G25G/download
Vivado% AES256GCM10G25GIPTest_ZCU106.bat
```

Figure 1-2 Example command script for download configuration file

2 Serial Console

User can fill RAMs with AAD, plain or cipher data patterns, set encryption/decryption key, IV and control test operation via serial console. When configuration is completed, AES256GCM10G25G demo command menu will be displayed as shown in Figure 2-1. The detailed information of each menu is described in topic 3.



```
=====================================================
AES256GCM Version = 0x00003842


++++++ AES256GCM Demo Menu ++++++
1. Set rEncKeyIn and rDecKeyIn
2. Set rEncIvIn and rDecIvIn
3. Set AAD for Encryption/Decryption
4. Show Data Memory
5. Fill Plain Data Memory
6. Encrypt Data
7. Loop verification
8. Fill Cipher Data Memory
9. Decrypt Data
Choice:
```

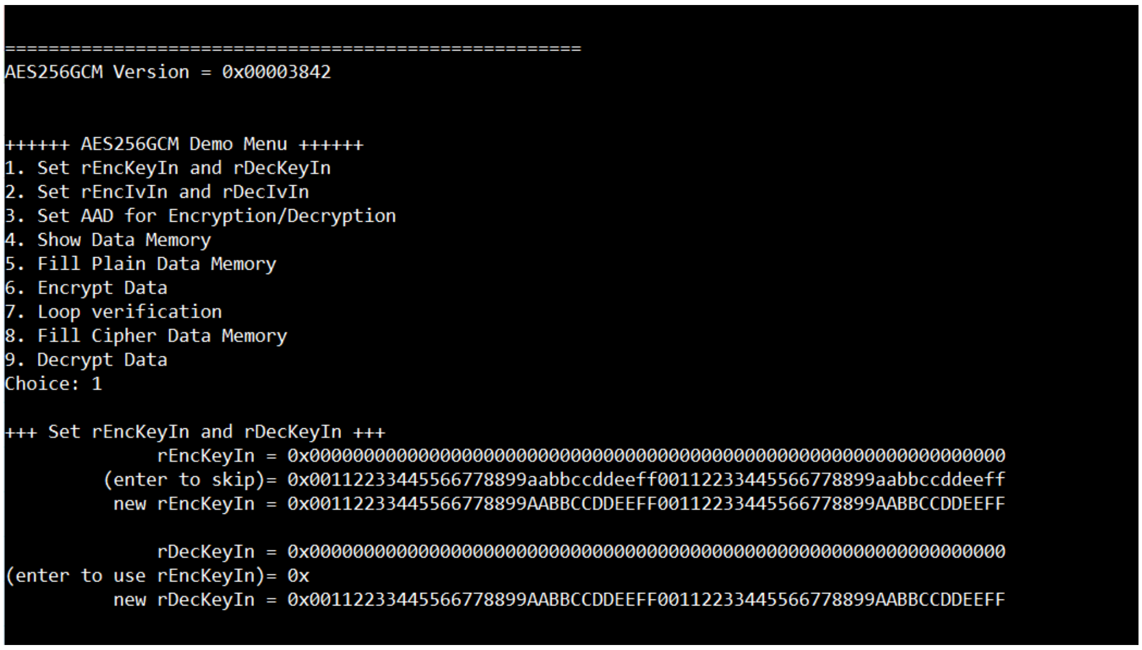
Figure 2-1 Serial console

3 Command detail and testing result

3.1 Set encryption/decryption key

Step to set encryption key and decryption key as follows

a) Select “Set rEncKeyIn and rDecKeyIn”.
b) Current encryption key will be displayed on serial console as shown in Figure 3-1.
c) Set new encryption key: User is allowed to input new key in hex format or press “enter” to skip setting new key. Then the current encryption key is printed again.
d) Current decryption key will be displayed on serial console.
e) Set new decryption key: User is allowed to input new key in hex format or press “enter” to use rEncKeyIn as rDecKeyIn. Then the current decryption key is printed again.



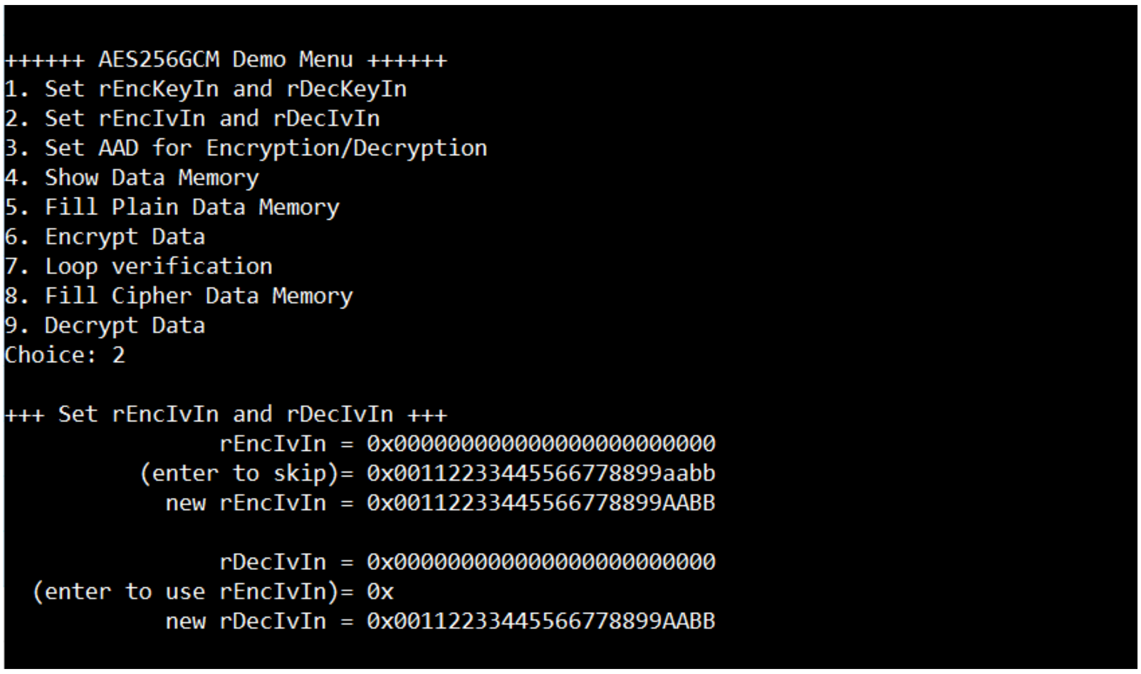
```
=====================================================
AES256GCM Version = 0x00003842


++++++ AES256GCM Demo Menu ++++++
1. Set rEncKeyIn and rDecKeyIn
2. Set rEncIvIn and rDecIvIn
3. Set AAD for Encryption/Decryption
4. Show Data Memory
5. Fill Plain Data Memory
6. Encrypt Data
7. Loop verification
8. Fill Cipher Data Memory
9. Decrypt Data
Choice: 1

+++ Set rEncKeyIn and rDecKeyIn +++
              rEncKeyIn = 0x0000000000000000000000000000000000000000000000000000000000000000
         (enter to skip)= 0x00112233445566778899aabbccddeeff00112233445566778899aabbccddeeff
          new rEncKeyIn = 0x00112233445566778899AABBCCDDEEFF00112233445566778899AABBCCDDEEFF

              rDecKeyIn = 0x0000000000000000000000000000000000000000000000000000000000000000
(enter to use rEncKeyIn)= 0x
          new rDecKeyIn = 0x00112233445566778899AABBCCDDEEFF00112233445566778899AABBCCDDEEFF
```

Figure 3-1 rEncKeyIn and rDecKeyIn setting example

3.2 Set encryption/decryption IV

Step to set encryption IV and decryption IV as follows

a) Select "Set rEncIvIn and rDecIvIn".
b) Current encryption IV will be displayed on serial console as shown in Figure 3-2.
c) Set new encryption IV: User is allowed to input new IV in hex format or press "enter" to skip setting new IV. Then the current encryption IV is printed again.
d) Current decryption IV will be displayed on serial console.
e) Set new decryption IV: User is allowed to input new IV in hex format or press "enter" to use rEncIvIn as rDecIvIn. Then the current decryption IV is printed again.



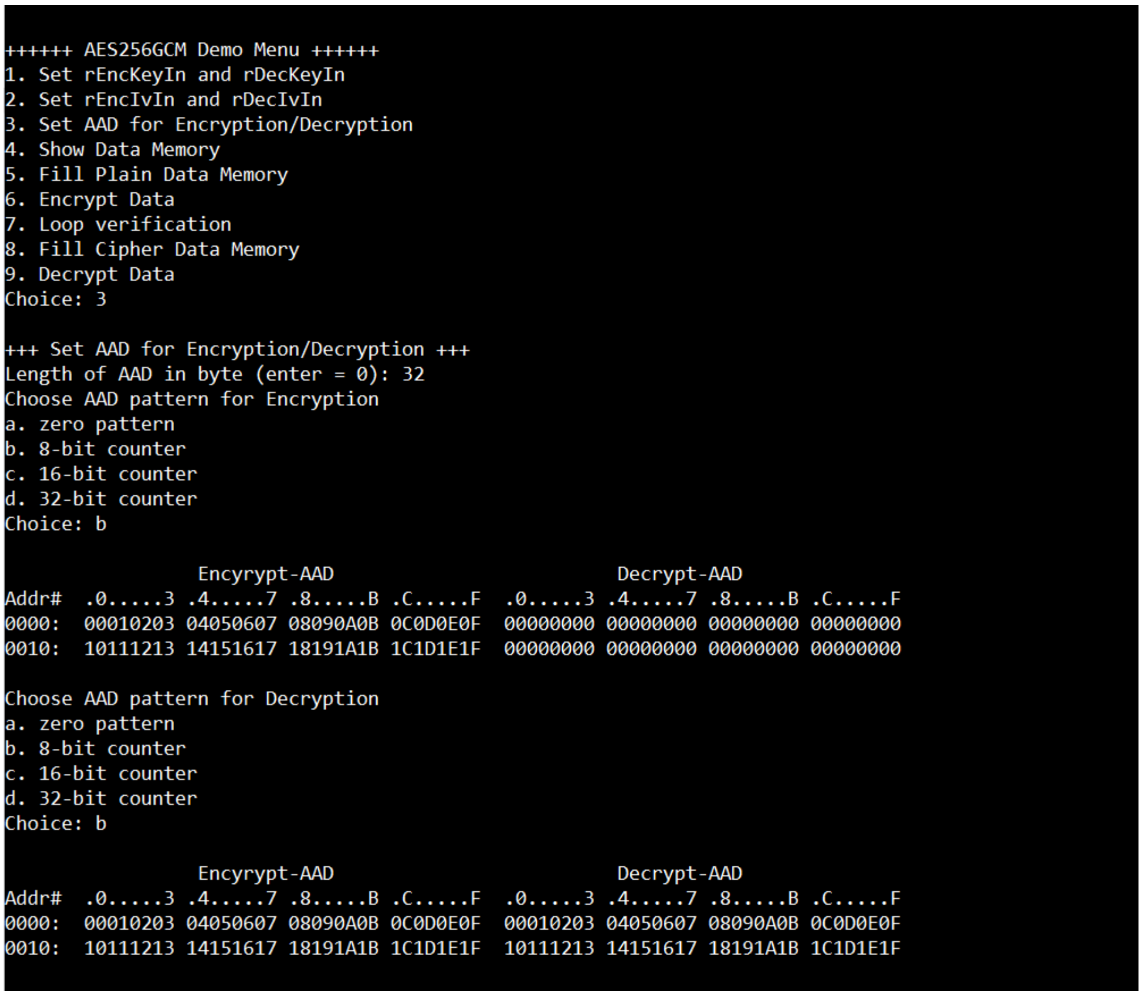
```
++++++ AES256GCM Demo Menu ++++++
1. Set rEncKeyIn and rDecKeyIn
2. Set rEncIvIn and rDecIvIn
3. Set AAD for Encryption/Decryption
4. Show Data Memory
5. Fill Plain Data Memory
6. Encrypt Data
7. Loop verification
8. Fill Cipher Data Memory
9. Decrypt Data
Choice: 2

+++ Set rEncIvIn and rDecIvIn +++
                rEncIvIn = 0x000000000000000000000000
          (enter to skip)= 0x00112233445566778899aabb
            new rEncIvIn = 0x00112233445566778899AABB

                rDecIvIn = 0x000000000000000000000000
  (enter to use rEncIvIn)= 0x
            new rDecIvIn = 0x00112233445566778899AABB
```

Figure 3-2 rEncIvIn and rDecIvIn setting example

3.3 Set AAD for Encryption/Decryption

Step to set AAD for encryption and decryption as follows

a) Select "Set AAD for Encryption/Decryption".
b) Input the desired length of AAD in byte. This length will be used for both encryption and decryption. 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 for encryption and decryption as shown in Figure 3-3.
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.

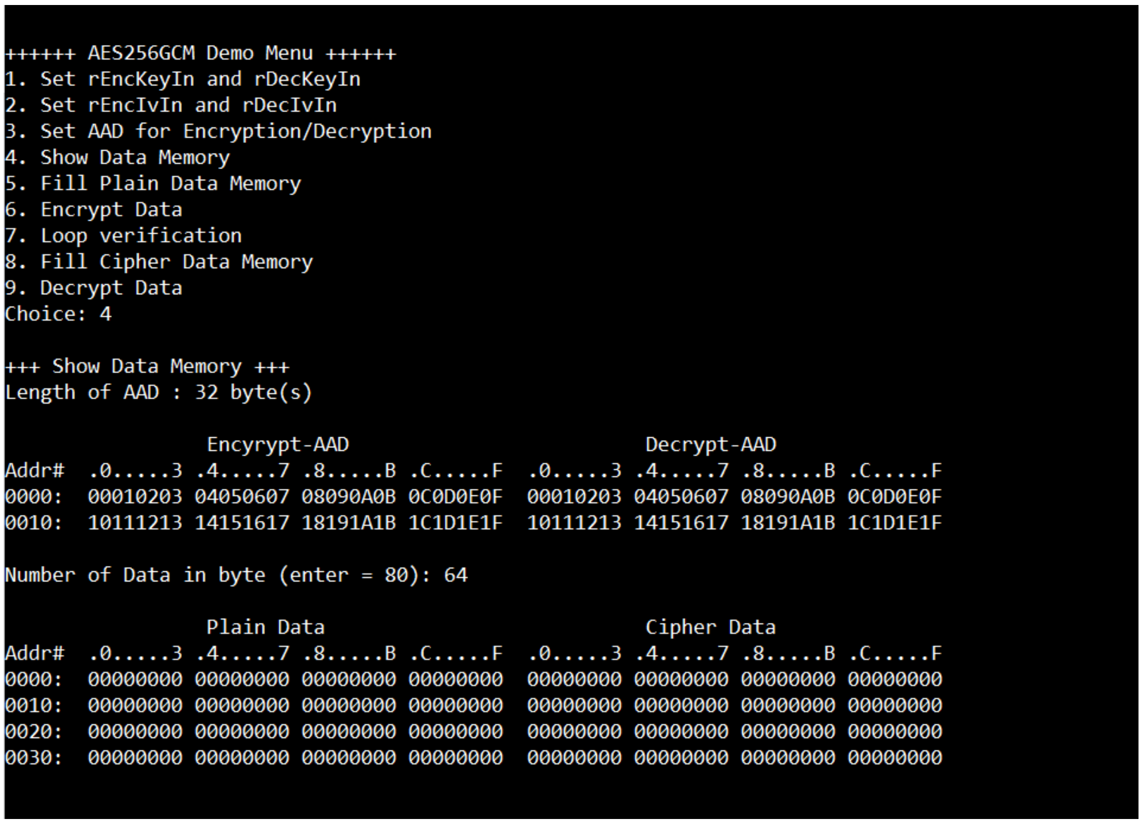


```
++++++ AES256GCM Demo Menu ++++++
1. Set rEncKeyIn and rDecKeyIn
2. Set rEncIvIn and rDecIvIn
3. Set AAD for Encryption/Decryption
4. Show Data Memory
5. Fill Plain Data Memory
6. Encrypt Data
7. Loop verification
8. Fill Cipher Data Memory
9. Decrypt Data
Choice: 3

+++ Set AAD for Encryption/Decryption +++
Length of AAD in byte (enter = 0): 32
Choose AAD pattern for Encryption
a. zero pattern
b. 8-bit counter
c. 16-bit counter
d. 32-bit counter
Choice: b

                 Encyrypt-AAD                          Decrypt-AAD
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
0010:  10111213 14151617 18191A1B 1C1D1E1F  00000000 00000000 00000000 00000000

Choose AAD pattern for Decryption
a. zero pattern
b. 8-bit counter
c. 16-bit counter
d. 32-bit counter
Choice: b

                 Encyrypt-AAD                          Decrypt-AAD
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
0010:  10111213 14151617 18191A1B 1C1D1E1F  10111213 14151617 18191A1B 1C1D1E1F
```

Figure 3-3 Displayed data when set AAD pattern

3.4 Show Data Memory

To show AAD and data in memory, user can select “Show Data Memory”. AAD will be shown by the number of AAD that user already set in menu “Set AAD for Encryption/Decryption”. User can input the desired length of data in byte to show. Both plain data and cipher data will be displayed in table-form as shown in Figure 3-4. User can press “enter” key to skip putting the number of data, then serial console will display 80 bytes of plain data and cipher data at address 0x0000-0x004F in five rows of table.



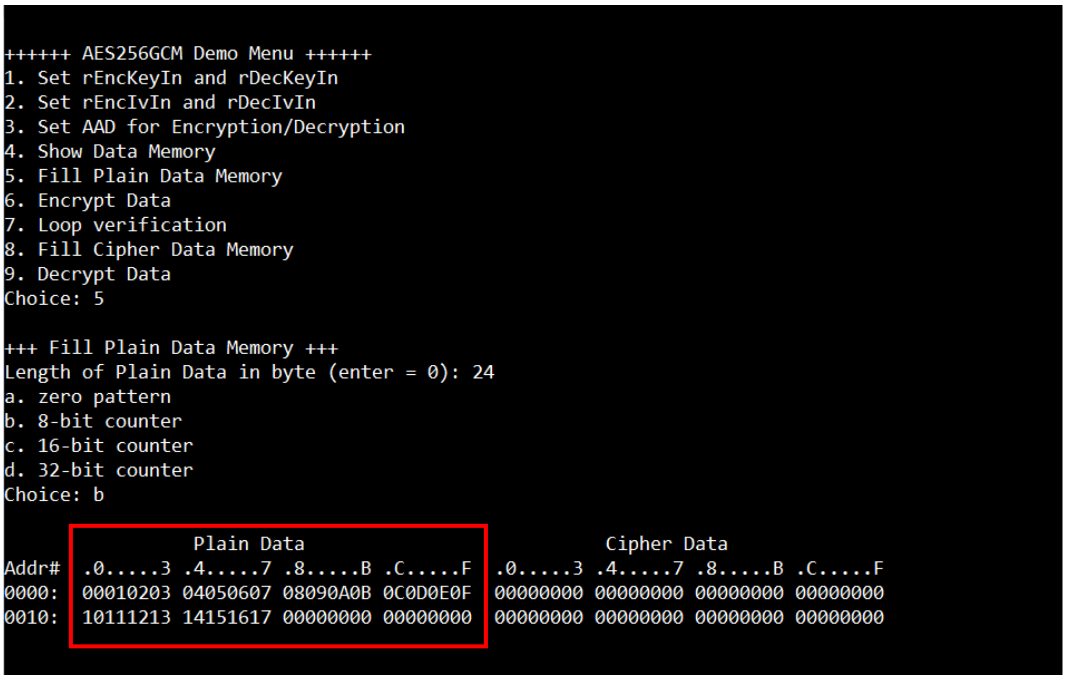
```
++++++ AES256GCM Demo Menu ++++++
1. Set rEncKeyIn and rDecKeyIn
2. Set rEncIvIn and rDecIvIn
3. Set AAD for Encryption/Decryption
4. Show Data Memory
5. Fill Plain Data Memory
6. Encrypt Data
7. Loop verification
8. Fill Cipher Data Memory
9. Decrypt Data
Choice: 4

+++ Show Data Memory +++
Length of AAD : 32 byte(s)

                 Encyrypt-AAD                           Decrypt-AAD
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
0010:  10111213 14151617 18191A1B 1C1D1E1F  10111213 14151617 18191A1B 1C1D1E1F

Number of Data in byte (enter = 80): 64

                 Plain Data                             Cipher Data
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
0010:  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
0020:  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
0030:  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
```

Figure 3-4 Displayed data when input the desired length of data

3.5 Fill Plain Data Memory

Step to fill plain data in memory as follows

a) Select “Fill Plain Data Memory”.
b) Input the desired length of data in byte. In case of zero-length plain data operation, user can input “0” or press “enter” on keyboard then end process of this menu. In case of non-zero-length plain data, 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
e) Whole plain-data memory is filled with selected pattern by the number of input data length and zero-padding to become 128-bit padded data as displayed in Figure 3-5.

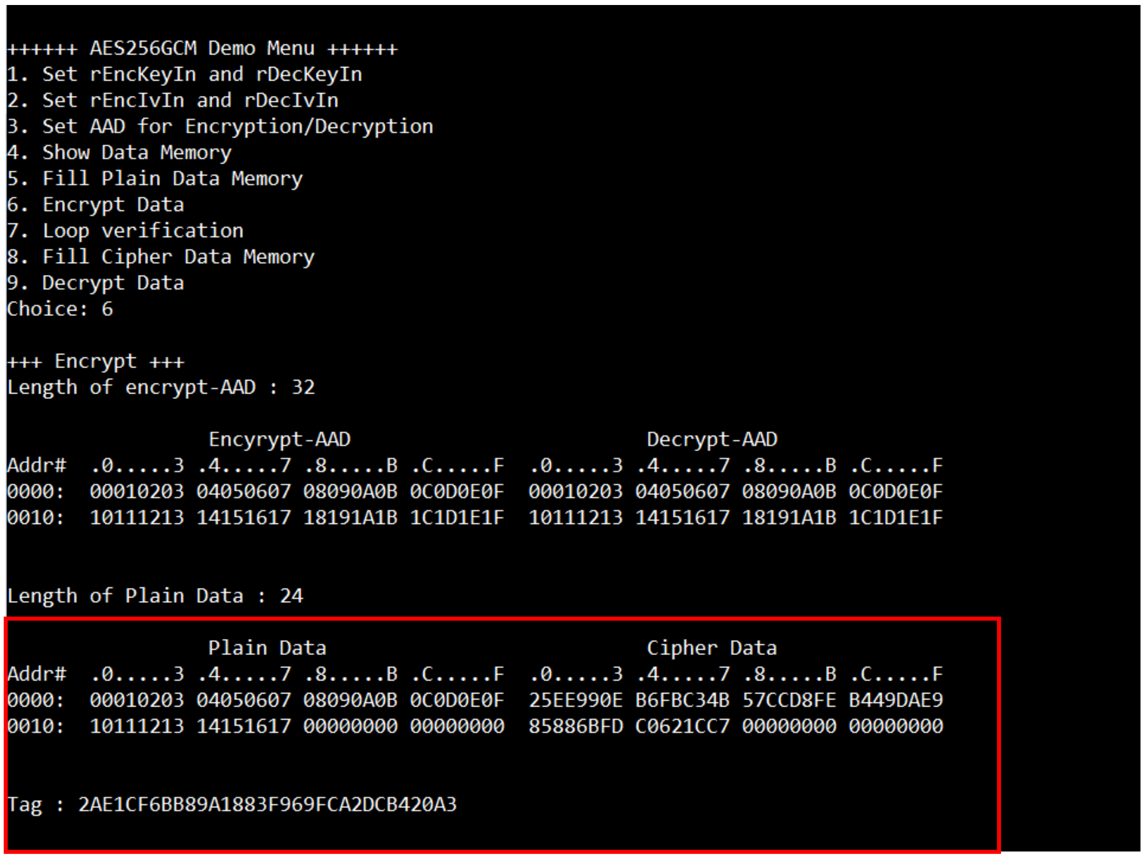


```
++++++ AES256GCM Demo Menu ++++++
1. Set rEncKeyIn and rDecKeyIn
2. Set rEncIvIn and rDecIvIn
3. Set AAD for Encryption/Decryption
4. Show Data Memory
5. Fill Plain Data Memory
6. Encrypt Data
7. Loop verification
8. Fill Cipher Data Memory
9. Decrypt Data
Choice: 5

+++ Fill Plain Data Memory +++
Length of Plain Data in byte (enter = 0): 24
a. zero pattern
b. 8-bit counter
c. 16-bit counter
d. 32-bit counter
Choice: b

                Plain Data                              Cipher Data
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
0010:  10111213 14151617 00000000 00000000  00000000 00000000 00000000 00000000
```

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

3.6 Encrypt

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



```
++++++ AES256GCM Demo Menu ++++++
1. Set rEncKeyIn and rDecKeyIn
2. Set rEncIvIn and rDecIvIn
3. Set AAD for Encryption/Decryption
4. Show Data Memory
5. Fill Plain Data Memory
6. Encrypt Data
7. Loop verification
8. Fill Cipher Data Memory
9. Decrypt Data
Choice: 6

+++ Encrypt +++
Length of encrypt-AAD : 32

                 Encyrypt-AAD                           Decrypt-AAD
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
0010:  10111213 14151617 18191A1B 1C1D1E1F  10111213 14151617 18191A1B 1C1D1E1F


Length of Plain Data : 24

                 Plain Data                             Cipher Data
Addr#  .0.....3 .4.....7 .8.....B .C.....F  .0.....3 .4.....7 .8.....B .C.....F
0000:  00010203 04050607 08090A0B 0C0D0E0F  25EE990E B6FBC34B 57CCD8FE B449DAE9
0010:  10111213 14151617 00000000 00000000  85886BFD C0621CC7 00000000 00000000


Tag : 2AE1CF6BB89A1883F969FCA2DCB420A3
```

Figure 3-6 Serial console after finished encryption process

3.7 Loop verification

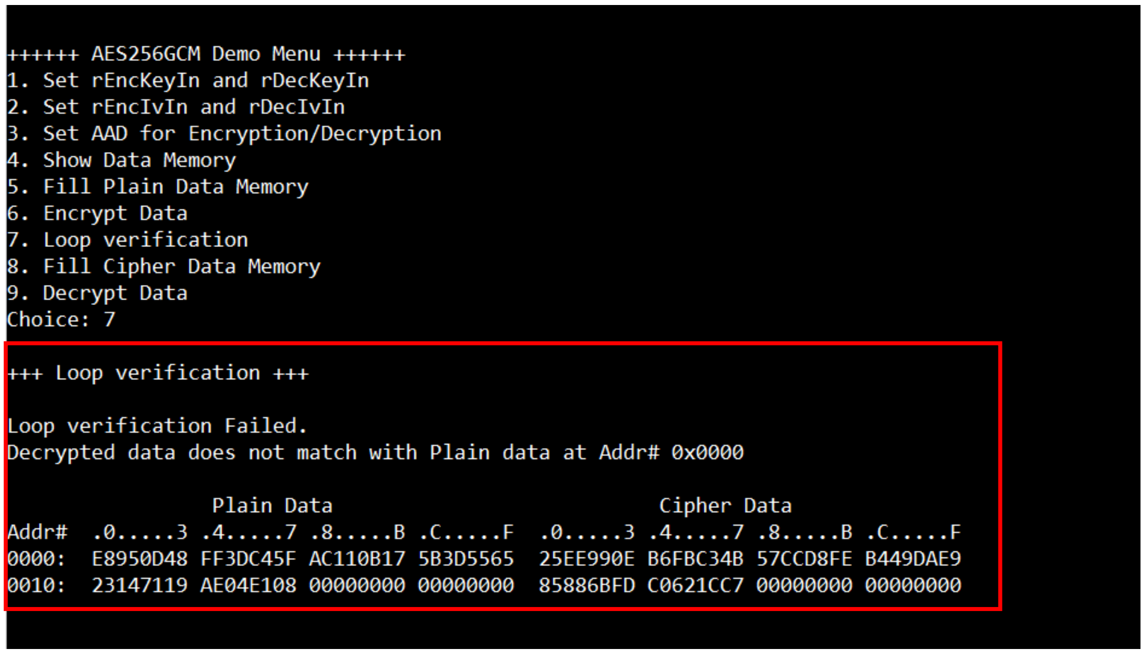
Select “Loop verification”, to check both encryption and decryption. In this menu, plain data in memory will be encrypted with current encryption key, IV and AAD and be stored in cipher data memory which is already cleared.

Then decryption is started, the cipher data will be decrypted with current decryption key, IV and AAD which user could set to be different from encryption parameters. The decrypted data will be stored in plain data memory which is already cleared.

If the decrypted data does not match with plain data input, the first address of the decrypted data which does not match and the data in memory will be printed in table-form on serial console as shown in Figure 3-7.

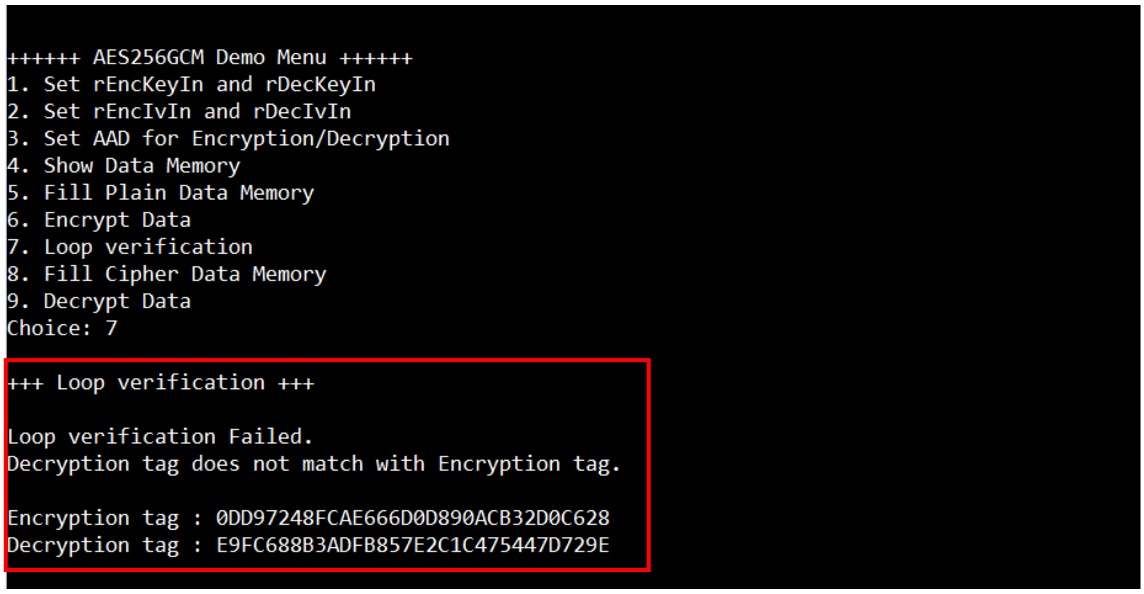
If the decryption tag does not match with encryption tag, both encryption tag and decryption tag will be printed on serial console as shown in Figure 3-8.

If the decrypted data and decryption tag match with plain data input and encryption tag, respectively, “Loop verification succeeded.” is printed on serial console as shown in Figure 3-9.



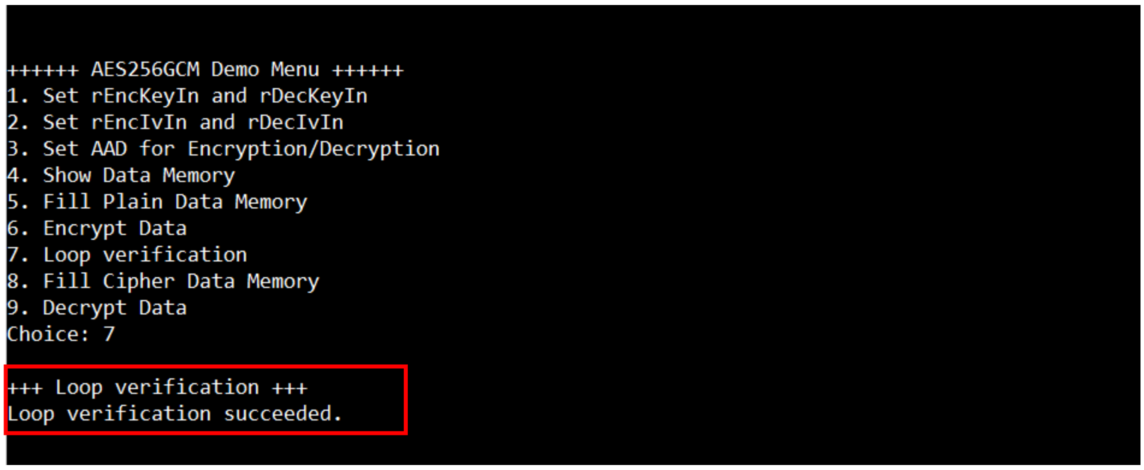
```
++++++ AES256GCM Demo Menu ++++++
1. Set rEncKeyIn and rDecKeyIn
2. Set rEncIvIn and rDecIvIn
3. Set AAD for Encryption/Decryption
4. Show Data Memory
5. Fill Plain Data Memory
6. Encrypt Data
7. Loop verification
8. Fill Cipher Data Memory
9. Decrypt Data
Choice: 7

+++ Loop verification +++

Loop verification Failed.
Decrypted data does not match with Plain data at Addr# 0x0000

                 Plain Data                                   Cipher Data
Addr#  .0.....3 .4.....7 .8.....B .C.....F  .0.....3 .4.....7 .8.....B .C.....F
0000:  E8950D48 FF3DC45F AC110B17 5B3D5565  25EE990E B6FBC34B 57CCD8FE B449DAE9
0010:  23147119 AE04E108 00000000 00000000  85886BFD C0621CC7 00000000 00000000
```

Figure 3-7 Serial console after loop verification is failed with data error



```
++++++ AES256GCM Demo Menu ++++++
1. Set rEncKeyIn and rDecKeyIn
2. Set rEncIvIn and rDecIvIn
3. Set AAD for Encryption/Decryption
4. Show Data Memory
5. Fill Plain Data Memory
6. Encrypt Data
7. Loop verification
8. Fill Cipher Data Memory
9. Decrypt Data
Choice: 7

+++ Loop verification +++

Loop verification Failed.
Decryption tag does not match with Encryption tag.

Encryption tag : 0DD97248FCAE666D0D890ACB32D0C628
Decryption tag : E9FC688B3ADFB857E2C1C475447D729E
```

Figure 3-8 Serial console after loop verification is failed with tag error



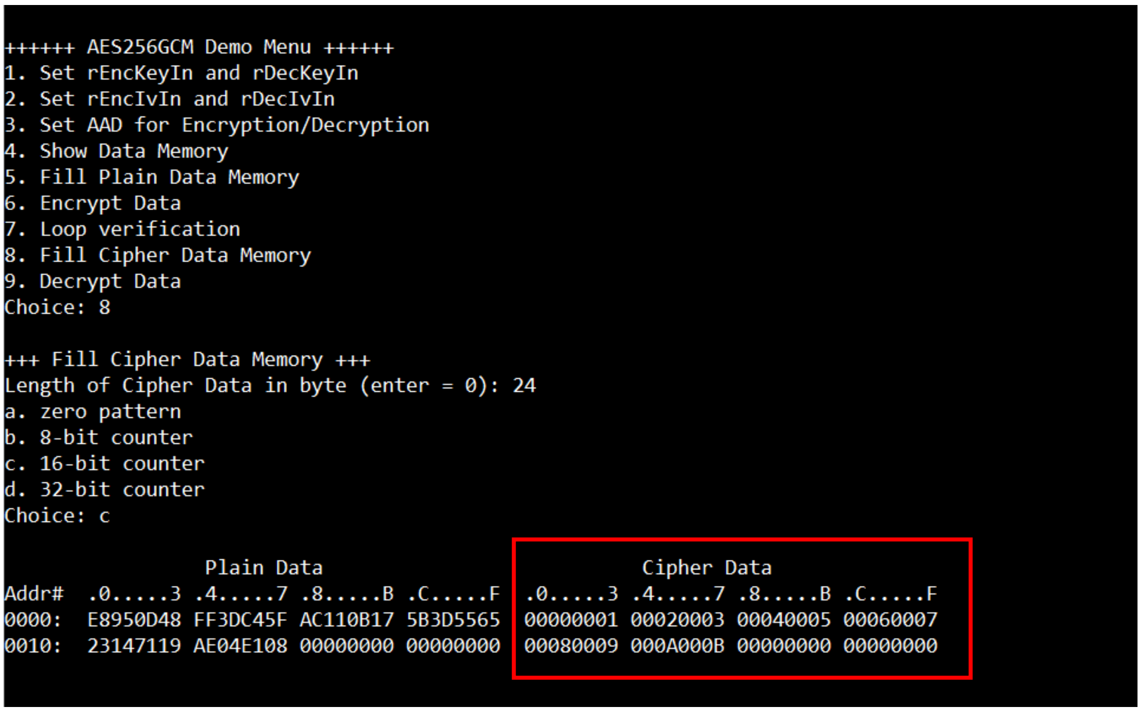
```
++++++ AES256GCM Demo Menu ++++++
1. Set rEncKeyIn and rDecKeyIn
2. Set rEncIvIn and rDecIvIn
3. Set AAD for Encryption/Decryption
4. Show Data Memory
5. Fill Plain Data Memory
6. Encrypt Data
7. Loop verification
8. Fill Cipher Data Memory
9. Decrypt Data
Choice: 7

+++ Loop verification +++
Loop verification succeeded.
```

Figure 3-9 Serial console after loop verification is succeeded

3.8 Fill Cipher Data Memory

Step to fill cipher data in memory as follows

a) Select “Fill Cipher Data Memory”.
b) Input the desired length of data in byte. In case of zero-length cipher data operation, user can input “0” or press “enter” on keyboard then end process of this menu. In case of non-zero-length cipher data, 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
f) Whole cipher-data memory is filled with selected pattern by the number of input data length and zero-padding to become 128-bit padded data as displayed in Figure 3-10.

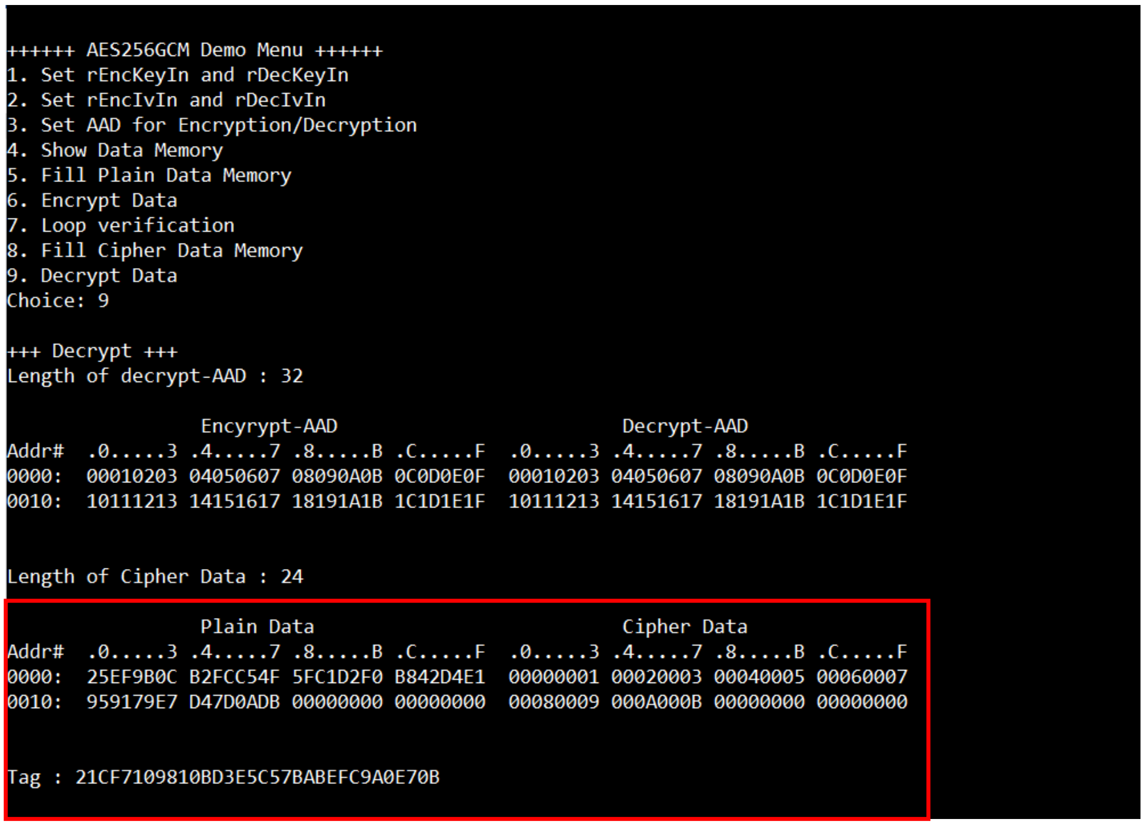


```
++++++ AES256GCM Demo Menu ++++++
1. Set rEncKeyIn and rDecKeyIn
2. Set rEncIvIn and rDecIvIn
3. Set AAD for Encryption/Decryption
4. Show Data Memory
5. Fill Plain Data Memory
6. Encrypt Data
7. Loop verification
8. Fill Cipher Data Memory
9. Decrypt Data
Choice: 8

+++ Fill Cipher Data Memory +++
Length of Cipher Data in byte (enter = 0): 24
a. zero pattern
b. 8-bit counter
c. 16-bit counter
d. 32-bit counter
Choice: c

                 Plain Data                                 Cipher Data
Addr#  .0.....3 .4.....7 .8.....B .C.....F  .0.....3 .4.....7 .8.....B .C.....F
0000:  E8950D48 FF3DC45F AC110B17 5B3D5565  00000001 00020003 00040005 00060007
0010:  23147119 AE04E108 00000000 00000000  00080009 000A000B 00000000 00000000
```

Figure 3-10 Displayed data when set cipher data length and data pattern

3.9 Decrypt

Select “Decrypt” to decrypt cipher data in memory. Current length of AAD, decryption AAD and length of cipher data are printed on serial console. When the decryption process is finished, both plain data and cipher data will be displayed in table-form and 128-bit decryption tag will be printed as shown in Figure 3-11.



```
++++++ AES256GCM Demo Menu ++++++
1. Set rEncKeyIn and rDecKeyIn
2. Set rEncIvIn and rDecIvIn
3. Set AAD for Encryption/Decryption
4. Show Data Memory
5. Fill Plain Data Memory
6. Encrypt Data
7. Loop verification
8. Fill Cipher Data Memory
9. Decrypt Data
Choice: 9

+++ Decrypt +++
Length of decrypt-AAD : 32

                 Encyrypt-AAD                          Decrypt-AAD
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
0010:  10111213 14151617 18191A1B 1C1D1E1F  10111213 14151617 18191A1B 1C1D1E1F


Length of Cipher Data : 24

                 Plain Data                            Cipher Data
Addr#  .0.....3 .4.....7 .8.....B .C.....F  .0.....3 .4.....7 .8.....B .C.....F
0000:  25EF9B0C B2FCC54F 5FC1D2F0 B842D4E1  00000001 00020003 00040005 00060007
0010:  959179E7 D47D0ADB 00000000 00000000  00080009 000A000B 00000000 00000000


Tag : 21CF7109810BD3E5C57BABEFC9A0E70B
```

Figure 3-11 Serial console after finished decryption process

4 Revision History

Revision	Date	Description
1.00	17-Jun-2022	Initial version release
1.02	20-Sep-2022	Update description for new design