

Project Zipline Crypto Engine

Micro Architecture Specification

**Authors:**

**Microsoft Corporation**

**Broadcom Corporation**

**Revision History**

|  |  |
| --- | --- |
| Date | Description |
| 04/03/2019 | Version 1.0 |
|  |  |

**License**

Contributions to this Specification are made under the terms and conditions set forth in the Open Web Foundation Contributor License Agreement (“OWF CLA 1.0”) (“Contribution License”) by:

Microsoft Corporation

Broadcom Corporation

Usage of this Specification is governed by the terms and conditions set forth in Open Web Foundation Final Specification Agreement (“OWFa 1.0”) (“Specification License”).

**Note**: The following clarifications, which distinguish technology licensed in the Contribution License and/or Specification License from those technologies merely referenced (but not licensed), were accepted by the Incubation Committee of the OCP:

**None.**

NOTWITHSTANDING THE FOREGOING LICENSES, THIS SPECIFICATION IS PROVIDED BY OCP "AS IS" AND OCP EXPRESSLY DISCLAIMS ANY WARRANTIES (EXPRESS, IMPLIED, OR OTHERWISE), INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR A PARTICULAR PURPOSE, OR TITLE, RELATED TO THE SPECIFICATION. NOTICE IS HEREBY GIVEN, THAT OTHER RIGHTS NOT GRANTED AS SET FORTH ABOVE, INCLUDING WITHOUT LIMITATION, RIGHTS OF THIRD PARTIES WHO DID NOT EXECUTE THE ABOVE LICENSES, MAY BE IMPLICATED BY THE IMPLEMENTATION OF OR COMPLIANCE WITH THIS SPECIFICATION. OCP IS NOT RESPONSIBLE FOR IDENTIFYING RIGHTS FOR WHICH A LICENSE MAY BE REQUIRED IN ORDER TO IMPLEMENT THIS SPECIFICATION. THE ENTIRE RISK AS TO IMPLEMENTING OR OTHERWISE USING THE SPECIFICATION IS ASSUMED BY YOU. IN NO EVENT WILL OCP BE LIABLE TO YOU FOR ANY MONETARY DAMAGES WITH RESPECT TO ANY CLAIMS RELATED TO, OR ARISING OUT OF YOUR USE OF THIS SPECIFICATION, INCLUDING BUT NOT LIMITED TO ANY LIABILITY FOR LOST PROFITS OR ANY CONSEQUENTIAL, INCIDENTAL, INDIRECT, SPECIAL OR PUNITIVE DAMAGES OF ANY CHARACTER FROM ANY CAUSES OF ACTION OF ANY KIND WITH RESPECT TO THIS SPECIFICATION, WHETHER BASED ON BREACH OF CONTRACT, TORT (INCLUDING NEGLIGENCE), OR OTHERWISE, AND EVEN IF OCP HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Table of Contents

[1 Overview 1](#_Toc536792037)

[1.1 Features 1](#_Toc536792038)

[1.2 Interfaces 1](#_Toc536792039)

[1.2.1 Clock, Reset & Miscellaneous 1](#_Toc536792040)

[1.2.2 AXI-4 Crypto Stream Slave Interface 2](#_Toc536792041)

[1.2.3 AXI-4 Key Stream Slave Interface 3](#_Toc536792042)

[1.2.4 AXI-4 Crypto Stream Master Interface 4](#_Toc536792043)

[1.2.5 RBUS Slave Register Interface 5](#_Toc536792044)

[1.2.6 RBUS Master Register Interface 5](#_Toc536792045)

[2 Cryptographic Support Summary 7](#_Toc536792046)

[2.1 Symmetric Cipher Summary 7](#_Toc536792047)

[2.2 Hash Summary 7](#_Toc536792048)

[2.3 Deterministic Random Number Generator (DRNG) 7](#_Toc536792049)

[3 Header Formats 9](#_Toc536792050)

[3.1 CMD TLV 10](#_Toc536792051)

[3.1.1 Word 0 10](#_Toc536792052)

[3.1.2 Word 1 10](#_Toc536792053)

[3.1.3 Word 2 11](#_Toc536792054)

[3.2 Key TLV 12](#_Toc536792055)

[3.2.1 Word 0 12](#_Toc536792056)

[3.2.2 Later Words 13](#_Toc536792057)

[3.3 FRMD TLV 13](#_Toc536792058)

[3.3.1 FRMD\_USER\_VM TLV 14](#_Toc536792059)

[3.3.2 FRMD\_INT\_APP TLV 15](#_Toc536792060)

[3.3.3 FRMD\_INT\_VM\_LONG TLV 16](#_Toc536792061)

[3.3.4 FRMD\_INT\_VM\_SHORT TLV 17](#_Toc536792062)

[3.4 CR\_IV TLV 18](#_Toc536792063)

[3.4.1 Word 0 18](#_Toc536792064)

[3.5 Data TLV 19](#_Toc536792065)

[3.5.1 Word 0 19](#_Toc536792066)

[3.5.2 Later Words 19](#_Toc536792067)

[3.6 Footer TLV 19](#_Toc536792068)

[3.6.1 Word 0 20](#_Toc536792069)

[3.6.2 Word 1-4 21](#_Toc536792070)

[3.6.3 Word 5-6 21](#_Toc536792071)

[3.6.4 Word 7-10 21](#_Toc536792072)

[3.6.5 Word 13-16 22](#_Toc536792073)

[3.6.6 Word 19 22](#_Toc536792074)

[4 Endian Swapping 25](#_Toc536792075)

[5 Block Overview 26](#_Toc536792076)

[5.1 Summary of Blocks 27](#_Toc536792077)

[5.2 Command FIFOs 31](#_Toc536792078)

[5.3 Key FIFOs 31](#_Toc536792079)

[5.4 Other FIFOs 31](#_Toc536792080)

[5.5 IV Processor 32](#_Toc536792081)

[6 Cipher Setup Example 33](#_Toc536792082)

[7 Hash Setup Example 34](#_Toc536792083)

[8 Test Vectors 35](#_Toc536792084)

[8.1 Setup 35](#_Toc536792085)

[8.2 CMD TLV 35](#_Toc536792086)

[8.3 Key TLV 35](#_Toc536792087)

[8.4 FRMD\_USER\_VM TLV 36](#_Toc536792088)

[8.5 Data TLV 36](#_Toc536792089)

[9 Statistics 37](#_Toc536792090)

[10 Interrupts 38](#_Toc536792091)

[11 References 39](#_Toc536792092)

[12 List of Figures 40](#_Toc536792093)

[13 List of Tables 41](#_Toc536792094)

# Overview

Crypto Engine is a single clock domain block that provides encryption, decryption and authentication services through a 64-bit wide AXI4 Streaming Interface. The register space is managed by a RBUS Interface.

## Features

* Supports 64-bit at 800 MHz AXI Stream master and slave interface.
* Supports 32-bit at 800 MHz RBUS slave interface.
* Crypto subsystem maintains packet sequence.
* AES Cipher modes. AES-256 GCM, XEX and XTS with 512b keys.
* HASH modes: SHA-256, HMAC-SHA256, and AES-256-GMAC.
* Authenticated Encryption mode: AES-256-GCM.
* All AES Engines are pipelined to do one round per clock cycle to meet 25 Gbps throughput.
* Two Pipelined SHA-2 Engines does four rounds per clock cycle to meet 25 Gbps throughput.
* Crypto subsystem does not support 0-byte payloads.

## Interfaces

There are six major interfaces on the Cryptography Engine.

### Clock, Reset & Miscellaneous

|  |  |  |
| --- | --- | --- |
| **Signal Name** | **Direction** | **Function** |
| Clk | Input | Main clock at 800 MHz |
| rst\_n | Input | Active low main reset |
| key\_mode | Input | Only allow keys on a dedicated interface |
| crypto\_mode[2:0] | Input | 1 = Encrypt Mode  2 = Decrypt Mode  3 = Integrity Mode (No AES Engine) |
| strap\_crypto\_module\_id[4:0] | Input | Unique Identified for this Crypto instantiation |
| crypto\_int[3:0] (Encrypt/Decrypt modes)  crypto\_int[3:0] (Integrity mode) | Output | Interrupts |
| scan\_en | Input | Scan signals |
| scan\_mode | Input | Scan signals |
| scan\_rst\_n | Input | Scan signals |
| ovstb | Input | Memory support signals |
| Lvm | Input | Memory support signals |
| mlvm | Input | Memory support signals |
| crypto\_stat\_events[63:0] | Output | Statistic Events |
| drng\_idle | Output | Crypto Idle |

Table 1 - Clock, Reset and Miscellaneous Interface

### AXI-4 Crypto Stream Slave Interface

|  |  |  |
| --- | --- | --- |
| **Signal Name** | **Direction** | **Function** |
| crypto\_ib\_in.**tvalid** | Input | Data transfer valid  0 : Not valid  1 : Valid |
| crypto\_ib\_out.**tready** | Output | Slave data acceptance for current cycle:  0 : Not Accepted  1 : Accepted |
| crypto\_ib\_in.**tdata[63:0]** | Input | Input data |
| crypto\_ib\_in.**tstrb[7:0]** | Input | Byte strobes corresponding to:  TSTRB[0] : TDATA[7:0]  TSTRB[1] : TDATA[15:8]  …  TSTRB[7] : TDATA[63:0]    Data Byte Valid:  0 : Not valid  1 : Valid    Only thermometer codes allowed (e.g. 0000\_0001 through 1111\_1111) |
| crypto\_ib\_in.**tlast** | Input | Last data phase of the data block |
| crypto\_ib\_in.**tid[0:0]** | Input | Data block identifier |
| crypto\_ib\_in.**tuser[7:0]** | Input | TUSER[0] : Start of TLV (first beat)  TUSER[1] : End of TLV (last beat)  TUSER[7:2] : All 0 |

Table 2 - AXI-4 Crypto Stream Slave Interface

### AXI-4 Key Stream Slave Interface

|  |  |  |
| --- | --- | --- |
| **Signal Name** | **Direction** | **Function** |
| key\_ib\_in.**tvalid** | Input | Data transfer valid  0 : Not valid  1 : Valid |
| key\_ib\_out.**tready** | Output | Slave data acceptance for current cycle:  0 : Not Accepted  1 : Accepted |
| key\_ib\_in.**tdata[63:0]** | Input | Input data |
| key\_ib\_in.**tstrb[7:0]** | Input | Byte strobes corresponding to:  TSTRB[0] : TDATA[7:0]  TSTRB[1] : TDATA[15:8]  …  TSTRB[7] : TDATA[63:0]    Data Byte Valid:  0 : Not valid  1 : Valid    Only thermometer codes allowed (e.g. 0000\_0001 through 1111\_1111) |
| key\_ib\_in.**tlast** | Input | Last data phase of the data block |
| key\_ib\_in.**tid[0:0]** | Input | Data block identifier |
| key\_ib\_in.**tuser[7:0]** | Input | TUSER[0] : Start of TLV (first beat)  TUSER[1] : End of TLV (last beat)  TUSER[7:2] : All 0 |

Table 3 - AXI-4 Key Stream Slave Interface

### AXI-4 Crypto Stream Master Interface

|  |  |  |
| --- | --- | --- |
| **Signal Name** | **Direction** | **Function** |
| crypto\_ob\_out.**tvalid** | Output | Data transfer valid  0 : Not valid  1 : Valid |
| crypto\_ob\_in.**tready** | Input | Slave data acceptance for current cycle:  0 : Not Accepted  1 : Accepted |
| crypto\_ob\_out.**tdata [63:0]** | Output | Input data |
| crypto\_ob\_out.**tstrb[7:0]** | Output | Byte strobes corresponding to:  TSTRB[0] : TDATA[7:0]  TSTRB[1] : TDATA[15:8]  …  TSTRB[7] : TDATA[63:0]    Data Byte Valid:  0 : Not valid  1 : Valid    Only thermometer codes allowed (e.g. 0000\_0001 through 1111\_1111) |
| crypto\_ob\_out.**tlast** | Output | Last data phase of the data block |
| crypto\_ob\_out.**tid[0:0]** | Output | Data block identifier |
| crypto\_ob\_out.**tuser[7:0]** | Output | TUSER[0] : Start of TLV (first beat)  TUSER[1] : End of TLV (last beat)  TUSER[7:2] : All 0 |

Table 4 - AXI-4 Crypto Stream Master Interface

### RBUS Slave Register Interface

|  |  |  |
| --- | --- | --- |
| **Signal Name** | **Direction** | **Function** |
| rbus\_ring\_i.**addr[31:0]** | Input | Byte Address. Note that address bits [1:0] are not significant, and are always set to “00”. |
| rbus\_ring\_i.**wr\_data[31:0]** | Input | Write Data. |
| rbus\_ring\_o.**rd\_data[31:0]** | Output | Read Data. |
| rbus\_ring\_i.**wr\_strb** | Input | Write strobe indicates the start of a write transaction. |
| rbus\_ring\_i.**rd\_strb** | Input | Read strobe indicates the start of a read transaction. |
| rbus\_ring\_o.**ack** | Output | Acknowledge indicates the end of a transaction (read or write). |
| rbus\_ring\_o.**err\_ack** | Output | Indicates that the slave aborted the transaction due to error. |

Table 5 - RBUS Slave Register Interface

### RBUS Master Register Interface

|  |  |  |
| --- | --- | --- |
| **Signal Name** | **Direction** | **Function** |
| rbus\_ring\_o.**addr[31:0]** | Output | Byte Address. Note that address bits [1:0] are not significant, and are always set to “00”. |
| rbus\_ring\_o.**wr\_data[31:0]** | Output | Write Data. |
| rbus\_ring\_i.**rd\_data[31:0]** | Input | Read Data. |
| rbus\_ring\_o.**wr\_strb** | Output | Write strobe indicates the start of a write transaction. |
| rbus\_ring\_o.**rd\_strb** | Output | Read strobe indicates the start of a read transaction. |
| rbus\_ring\_i.**ack** | Input | Acknowledge indicates the end of a transaction (read or write). |
| rbus\_ring\_i.**err\_ack** | Input | Indicates that the slave aborted the transaction due to error. |

Table 6 - RBUS Master Register Interface

# Cryptographic Support Summary

## Symmetric Cipher Summary

The table below summarizes the cipher algorithm parameters. AES in counter mode for all ciphers.

Table Block Cipher Modes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mode | Key Size | IV/Tweak | AAD | Standard | Notes |
| NULL | NA | NA | NA | -- | No encryption or authentication (bypass) |
| AES-XEX-512 | 2x256 (key1, key2) | 128b | NA | IEEE 1619-2007  [NIST 800-38e](http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38e.pdf) | Padding to 16B using ANSI X.923 |
| AES-XTS-512 | 2x256 (key1, key2) | 128b | NA | IEEE 1619-2007  [NIST 800-38e](http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38e.pdf) | Cipher text stealing used |
| AES-GCM-256 | 256 key | 96b | 0-255B | [NIST SP800-38d](http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38d.pdf) | 128 Authentication tag |
| AES-GMAC-256 | 256 key | 96b | 0-255B | [NIST SP800-38d](http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38d.pdf) | No encryption, auth only. 128b Authentication Tag |

## Hash Summary

The table below summarizes the hash algorithm parameters.

**Table 8 Authentication Algorithm Support**

|  |  |  |  |
| --- | --- | --- | --- |
| Mode | Specification | Key Size | Digest Size |
| NULL | NA | NA | 0 |
| SHA-256 | [FIPS.180-4](https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf) | NA | 256 |
| AES-GMAC | [NIST SP800-38d](http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38d.pdf) | 256 | 128 |
| HMAC-SHA-256 | [FIPS.198-1](https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.198-1.pdf) | 256 | 256 |

## Deterministic Random Number Generator (DRNG)

A DRNG is used to generate a random number from a seed or entropy. The DRNG as specified in [NIST SP800-90A](http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf) revision1 will be implemented. The DRNG is built using AES-256 block cipher per section 10.2.

**Table 9 DRNG NIST Parameters**

|  |  |  |
| --- | --- | --- |
| Parameter | Value | Notes |
| Block Cipher | AES-256 | AES\_ECB (no chaining) |
| Security strength | 256b |  |
| Key size | 256b |  |
| Block length (outlen) | 128b |  |
| Seed Length (seedlen) | 384b | Key size+outlen |
| Df function | None | Not implemented. |
| Prediction resistance | None | Not supported |
| Reseed counter size | 32 bits |  |
| Health Check | Consecutive Values |  |

# Header Formats

Crypto operates on TLV Headers. A TLV Header contains TLV type, TLV Length and Value.



Figure 1 - TLV Format

TLV Type identifies the type of header.

TLV Length is the sum of TLV Type, TLV Length and Value fields.

Start and End of TLV is defined through User Bits (see tuser in Section 1.2.2.)

The layout of the Value field will be discussed in the sections below for the Headers that are relevant to Crypto. All Value fields are Little Endian unless otherwise specified.

All 3 configurations of Crypto (Cipher Encrypt, Cipher Decrypt, and Cipher Integrity) modify the FOOTER TLV, while the Cipher Encrypt and Cipher Decrypt also modify the DATA TLV. All of the other TLVs that Crypto receives are passed through to the Crypto output unmodified (e.g. RQE, CMD, FRMD, CQE, etc). The Cipher Encrypt creates the CR\_IV TLV discussed below. The Cipher Decrypt passes this through to the output, but the Cipher Integrity will delete this TLV since it is only useful to Crypto modules.

## CMD TLV

TLV Type for CMD TLV is 1. Only the fields relevant to Crypto are shown in the diagram and table below.



Figure 2 - CMD TLV Format

### Word 0

|  |  |  |
| --- | --- | --- |
| Field | Width | Description |
| TLVType | 8 | TLV Type. This must be set to 1. |
| TLVEngineID | 4 | Engine ID Number. |
| TLVSeqNum | 8 | Sequence ID Number. |

Table 10 - CMD TLV Word 0

### Word 1

|  |  |  |
| --- | --- | --- |
| Field | Width | Description |
| Debug | 32 | See Top Microarch for the field breakdown. Only Functional Error Insert is supported for Crypto. Byte Mask field is overloaded to indicate the operation:  0x0: Corrupt MSB of IV in Footer TLV  0x1: Corrupt MSB of Tag in Footer TLV |

Table 11 - CMD TLV Word 1

### Word 2

|  |  |  |
| --- | --- | --- |
| Field | Width | Description |
| RawAuthOp | 4 | Raw Authentication Operation  0x0 : NULL  0x1 : SHA2-256  0x2 : HMAC-SHA2-256  Rest: Reserved |
| AuthOp | 4 | Authentication Operation  0x0 : NULL  0x1 : SHA2-256  0x2 : HMAC-SHA2-256  Rest: Reserved |
| CipherOp | 4 | Cipher Operation  0x0: NULL  0x1: AES-256-GCM  0x2: AES-256-XTS/XEX (Controlled via CipherPadOp)  0x3: AES-256-GMAC  Rest: Reserved  *For GCM/GMAC, AuthOp should be NULL.* |
| AADLen | 8 | Length of Additional Authenticated Data (AAD) in bytes from start of frame (AES-GCM Only).  *For GCM, AAD length should be less than payload size.*  *For non-GCM operations, this field should be set to 0.* |
| IVOp | 2 | IV Operation  0x0: Directly from IV Source (FRMD).  0x1: Random  0x2: Increment  0x3: Reserved |
| CipherPadOp | 1 | Cipher Pad Operation  0x0: No Padding (AES-XTS)  0x1: ANSI X.923 Padding to align to 16 byte boundary (AES-XEX) [PRD]  *This field should be set 0x0 for non-XTS/XEX operation.* |

Table 12 - CMD TLV Word 2

## Key TLV

TLV Type for Key TLV is 2. Only relevant fields to Crypto are discussed below.



Figure 3 - Key TLV Format

### Word 0

|  |  |  |
| --- | --- | --- |
| Field | Width | Description |
| TLVType | 8 | TLV Type. This must be set to 2. |
| TLVEngineID | 4 | Engine ID Number. |
| TLVSeqNum | 8 | Sequence ID Number. |

Table 13 - Key TLV Word 0

### Later Words

The size of the header is fixed.

|  |  |
| --- | --- |
| Field | Size |
| GUID0…GUID3 | Always 256-bits long. |
| IncrIV0…IncrIV1 | Can be 0, 96 or 128-bit long.  For a 128-bit IV, IncrIV0 will occupy [63:0] of the IV.  For a 96-bit IV, IncrIV0 will occupy [63:0] of the IV and IncrIV1 [31:0] will occupy [95:64] of the IV and IncrIV1 [63:32] will be unused. |
| CiphKey0…CiphKey7 | Can be 0 or 256-bits or 512-bits long.  For 512-bit keys:  Key1 occupies CiphKey0-3 with CipkKey0 occupying [511:448] of Key1  Key2 (Tweak Key) in XTS/XEX occupies CiphKey4-7 with CipkKey4 occupying [255:193] of Key1  For 256-bit keys:  Key occupies CiphKey4-7 with CipkKey4 occupying [255:192] of the Key  CiphKey0-3 are unused |
| HashKey0…HashKey3 | Can be 0 or 256-bit long.  HashKey0 will occupy [255:192] of the key. |
| KME Errors | Errors Generated during Key Generation at KME. Only LSB 16-bits are valid. |
| CRC-32 | CRC-32 over Words 1-18. Only LSB 32-bits are valid. |

Table 14 - Key TLV Later Words

## FRMD TLV

There are many flavors of FRMD TLVs. Crypto will use the following TLVs to extract the IV required in Cipher operation (CTRL0.**IVOp** =0x0). Only the fields relevant to the IV in the FRMDs are shown in below.

### FRMD\_USER\_VM TLV



Figure 4 - FRMD\_USER\_VM TLV Format

#### Word 0

|  |  |  |
| --- | --- | --- |
| Field | Width | Description |
| TLVType | 8 | TLV Type. This must be set to 14. |

Table 15 - FRMD USER VM TLV Word 0

#### Word 1-2

These words carry a maximum of 128-bit IV.

* Word 1 will carry [63:0] of IV.
* Word 2 will carry [127:64] of IV.

GCM/GMAC IV is 96-bits long.

* Word 1 will carry [63:0] of GCM/GMAC IV.
* Bits [31:0] of Word 1 will carry [95:64] of GCM/GMAC IV.
* Bits [63:32] of Word 2 are unused.

### FRMD\_INT\_APP TLV



Figure 5 - FRM INT APP TLV Format

#### Word 0

|  |  |  |
| --- | --- | --- |
| Field | Width | Description |
| TLVType | 8 | TLV Type. This must be set to 15. |

Figure 6 - FRMD\_INT\_APP TLV Word 0

#### Word 4-5

This word carries a maximum of 96-bit IV.

GCM/GMAC IV is 96-bits long and is represented in little endian format.

* Word 4 will carry [63:0] of GCM/GMAC IV.
* Bits [31:0] of Word 5 will carry [95:64] of GCM/GMAC IV.
* Bits [63:32] of Word 5 are unused.

### FRMD\_INT\_VM\_LONG TLV



Figure 7 - FRMD INT VM\_LONG TLV Format

#### Word 0

|  |  |  |
| --- | --- | --- |
| Field | Width | Description |
| TLVType | 8 | TLV Type. This must be set to 18. |

Table 16 - FRMD\_INT\_VM\_LONG TLV Word 0

#### Word 10-11

These words carry a maximum of 128-bit IV.

* Word 10 will carry [63:0] of IV.
* Word 11 will carry [127:64] of IV.

GCM/GMAC IV is 96-bits long.

* Word 10 will carry [63:0] of GCM/GMAC IV.
* Bits [31:0] of Word 11 will carry [95:64] of GCM/GMAC IV.
* Bits [63:32] of Word 11 are unused.

### FRMD\_INT\_VM\_SHORT TLV



Figure 8 - FRMD INT VM\_SHORT TLV Format

#### Word 0

|  |  |  |
| --- | --- | --- |
| Field | Width | Description |
| TLVType | 8 | TLV Type. This must be set to 22. |

Table 17 - FRMD\_INT\_VM\_SHORT TLV Word 0

#### Word 7-8

These words carry a maximum of 128-bit IV.

* Word 7 will carry [63:0] of IV.
* Word 8 will carry [127:64] of IV.

GCM/GMAC IV is 96-bits long.

* Word 7 will carry [63:0] of GCM/GMAC IV.
* Bits [31:0] of Word 8 will carry [95:64] of GCM/GMAC IV.
* Bits [63:32] of Word 8 are unused.

## CR\_IV TLV

This is a TLV that is used by the Cipher Decrypt and Integrity (if present). It is manufactured by Cipher Encrypt. It carries the IV that was generated by IV Processor. Cipher Integrity uses CR\_IV to grow the Footer TLV. If CR\_IV is not present, FMRD TLV is used to grow Footer TLV.



Figure 9 - CR IV Format

### Word 0

|  |  |  |
| --- | --- | --- |
| Field | Width | Description |
| TLVType | 8 | TLV Type. This must be set to 20. |

Table 18 - CR IV TLV Word 0

#### Word 1-2

These words carry a maximum of 128-bit IV.

* Word 1 will carry [63:0] of IV.
* Word 2 will carry [127:64] of IV.

GCM/GMAC IV is 96-bits long.

* Word 1 will carry [63:0] of GCM/GMAC IV.
* Bits [31:0] of Word 1 will carry [95:64] of GCM/GMAC IV.
* Bits [63:32] of Word 2 are unused.

## Data TLV

This header contains all the data that is to be processed by Crypto. It may have AAD as described by the CMD TLV. AAD is data that is to be authenticated but not encrypted. In Compound Commands, AAD is only valid for the first Data TLV.



Figure 10 - Data TLV Format

### Word 0

|  |  |  |
| --- | --- | --- |
| Field | Width | Description |
| TLVType | 8 | TLV Type. This must be set to 5. |

Table 19 - Data TLV Word 0

### Later Words

The number of words and their content is dependent on the size of the Payload and AAD.

## Footer TLV

This TLV contains status fields that are updated by Crypto.

* Integrity Mode: Copy GUID from Key TLV to Footer TLV.
* Integrity Mode: IV used in Cipher is copied to Footer TLV via CR\_IV or FRMD TLVs.
* Encrypt Mode: Generate an Integrity Check Value (ICV).
* Decrypt Mode: Generate an ICV and compare with the Expected ICV.
* Report errors.

Only the fields that are updated by Crypto are shown in the diagram and table below. Shaded Words are added by Integrity Engine.



Figure 11 - Footer TLV Format

### Word 0

|  |  |  |
| --- | --- | --- |
| Field | Width | Description |
| TLVType | 8 | TLV Type. This must be set to 6. |
| FrameNum | 11 | 0 = Simple Command  0-2047 = Compound Command |
| EncCmpDataMacSize | 2 | Encrypted Compressed Data MAC Size  0x0: 64 bits valid (upper 64 bits of the field)  0x1: 128 bits valid (upper 128 bits of the field)  0x2: 256 bits valid (full size of the field)  0x3: Reserved |
| RawDataMacSize | 2 | Raw Data MAC Size  0x0: 64 bits valid (upper 64 bits of the field)  0x1: 128 bits valid (upper 128 bits of the field)  0x2: 256 bits valid (full size of the field)  0x3: Reserved (This implicitly tells Integrity Engine to generate) |

Table 20 - Footer TLV Word 0

### Word 1-4

|  |  |  |
| --- | --- | --- |
| Field | Width | Description |
| GUID0-3 | 64 | This is the placeholder for GUID. It is copied from Key TLV. |

Table 21 - Footer Word 1 to 4

### Word 5-6

|  |  |  |
| --- | --- | --- |
| Field | Width | Description |
| CiphIv0-1 | 64 | This is the placeholder for Generated IV. It is also used to compare against the IV used in Cipher Operation in Decrypt Mode.  CiphIV0 will hold bits [63:0] of a 128-bit Cipher IV.  CiphIV1 will hold bits [127:84] of a 128-bit Cipher IV.  CiphIV0 will hold bits [63:0] of a 96-bit Cipher IV.  CiphIV1 will hold bits [95:64] of a 96-bit Cipher IV in bits [31:0] and bits [63:32] are unused |

Table 22 - Footer Word 5 to 6

### Word 7-10

|  |  |  |
| --- | --- | --- |
| Field | Width | Description |
| RawDataMac0-3 | 64 | This is the placeholder for computed ICV for Integrity Engine (Integrity Mode). It is also used to compare against computed ICV and flag a status bit on failed comparison. Note, this check is not performed in the CCEIP if aux\_cmd.frame\_ctrl.md\_type[0] is set to 1 and the incoming FRMD is of type USER\_VM.  RawDataMac0 will hold bits [255:192] of SHA Tag. |

Table 23 - Footer Word 7 to 10

### Word 13-16

|  |  |  |
| --- | --- | --- |
| Field | Width | Description |
| EncCmpDataMac0-3 | 64 | This is the placeholder for computed ICV for Crypto Engine. It is also used to compare against computed ICV and flag a status bit on failed comparison.  EncCmpDataMac0 will hold bits [255:192] of SHA Tag.  EncCmpDataMac0 will hold bits [127:64] of GCM/GMAC Tag. |

Table 24 - Footer Word 13 to 16

### Word 19

|  |  |  |
| --- | --- | --- |
| Field | Width | Description |
| ErrCode | 16 | The error codes below are listed in order of priority.  KME Errors have a higher priority than Crypto Errors.  **Items in Red Italic will force Crypto to clear the outgoing Data TLV.**  **[KME]**  *130: KIM entry for DakKeyRef is invalid*  *131: VF/PF Validation failed for DAK*  *132: KIM entry for DekKeyRef is invalid*  *133: VF/PF Validation failed for DEK*  *134: KME DRBG Seed Expired on Random GUID*  *135: Encrypted DEK keyblob decryption failed (ICV Tag miscompare)*  *136: Encrypted DAK keyblob decryption failed (ICV Tag miscompare)*  *137: Multibit ECC Uncorrectable Memory Error*  *138: Unsupported Key Type (Overloaded for multiple errors. See note below)*  *139: DEK marked as KEK can only be used to decrypt keys or KDF or both*  *140: DAK marked as KEK can only be used to decrypt keys or KDF or both*  *141: KME DRNG Health Check Error on Random GUID Generation*  **[ENCRYPTOR]**  *104: Key TLV CRC-32 Mismatch.*  *103: Engine ID mismatch between CMD and Key TLV.*  *102: Sequence number mismatch between CMD and Key TLV.*  *101: IV was not supplied through FRMD.*  *105: Invalid combinations of Crypto CMD TLV Word 2.*  106: Data must be at least 16 bytes for a XTS operation.  107: AAD Length should be strictly less than Data Length.  *100: DRNG Seed Expired. This IV is invalid.*  *108: DRNG Health Check Error*  **[DECRYPTOR/VALIDATOR]**  *115: Key TLV CRC-32 Mismatch.*  *114: Engine ID mismatch between CMD and Key TLV.*  *113: Sequence number mismatch between CMD and Key TLV.*  *112: IV was not supplied through FRMD.*  *116: Invalid combinations of Crypto CMD TLV Word 2.*  117: Data must be at least 16 bytes for a XTS operation.  118: AAD Length should be strictly less than Data Length.  110: Tag Compare Fail.  **[INTEGRITY GEN/CHECK]**  *123: Key TLV CRC-32 Mismatch.*  *122: Engine ID mismatch between CMD and Key TLV.*  *121: Sequence number mismatch between CMD and Key TLV.*  120: Tag Compare Fail. |
| ErrFrameNum | 32 | Indicates Frame Number with error. On error, frame number from Word 0 of Footer TLV is copied into here. |

Table 25 - Footer Word 19

**Notes**

The Unsupported Key Type Error is overloaded with the following errors:

1. Unencrypted Key Types 2-6 and disable\_encrypted\_keys OTP bit set to 1.
2. Key Type is NONE (Key Type 0) and key required (i.e. AUX\_CMD.crypto\_ctrl.cipher\_op != NONE or AUX\_CMD.crypto\_ctrl.auth\_op != NONE or AUX\_CMD.crypto\_ctrl.raw\_auth\_op != NONE).
3. Missing IV error
4. Missing GUID error

# Endian Swapping

All Data, IV, Keys and Data are (by default) Little Endian. Crypto algorithms operate on Big Endian. Therefore, IVs, Keys, Data and Tags need to be converted to Big Endian before Crypto operations.

In IV incrementing mode, IVs from Key TLV are incremented in Little Endian and converted to Big Endian before Crypto Operations.

This is how a 64-bit data is converted from Little Endian to Big Endian. This applies to Data, Keys, Tags and 128-bit IVs. Assembling 128-bit data blocks from 64-bit stream: **{Word 0, Word 1}** where Word 0 comes first in time.



This is how a 96-bit IV is converted from Little Endian to Big Endian.



# Block Overview



Figure 12- Crypto Block Diagram

## Summary of Blocks

We will summarize the functionality of each block in this section.

1. TLV Parser
   1. Common Block that identifies the TLV Type.
   2. Generates TLV start and end strobes.
   3. Extract sideband TLV Signals such as user bits.
2. TLV Inspector
   1. Interprets incoming TLVs and issues command to blocks using Command FIFOs.
      1. Small Size Bit in Hash Command indicates if DATA is less than 512 bits. This becomes helpful in load balancing Dual SHA engine.
   2. Strap bit, strap\_keypath\_en, enables Key TLV decoding from AXI Key Bus only.
   3. A fake Key TLV can be generated internally using a register configuration bit.
   4. Cipher and Hash Keys are extracted from Key TLV and pushed into Key FIFOs.
   5. IVs from FRMD (CiphIV) are pushed into Raw IV FIFO.
   6. IVs from Key TLV (IncrIV) are pushed into Key IV FIFO.
   7. GUID, KME Errors, Engine ID and Sequence Number are extracted from Key TLV and pushed into Key Metadata FIFO. Compare result of Key TLV CRC-32 is also stored in this FIFO.
   8. TLV Word 0 of FRMD TLV, Data TLV and Footer TLV are pushed into TLV Sideband FIFO. Only Data and Footer TLVs are modified by Crypto. Data TLV is used to insert a new CR\_IV TLV just before it.
   9. Sideband signals (ex. User bits) are stripped from Footer TLV and the contents are pushed into Footer FIFO.
   10. CQE TLV Word 0 is pushed into Footer TLV. CQE TLV acts as a delimiter.
   11. TLV Word 0 is stripped from Data TLV and the contents are forwarded to Upsizer.
   12. A delimiter is sent to Upsizer on CQE TLV to indicate end of Simple/Compound Command.
   13. Endian Swap Key, IV and Data if enabled.
   14. Counts the number of bits in DATA for Hash Operations and push into Length FIFO.
       1. In SHA-2, number of bits is over DATA only.
       2. In HMAC, number of bits is over DATA and IPAD (512 bits).
       3. In GMAC, number of bits is over DATA excluding AAD length.
   15. Detects if a FRMD-IV was missing.
   16. Detects if CMD TLV Word 2 is malformed.
   17. Handle Debug field in CMD TLV.
   18. Error detection and forcing of the outgoing Data TLV to 0 for the majority of the errors described in 3.6.6 that result in this behavior.
3. IV Processor
   1. Executes one of the three IV Commands
      1. In-Band IV: Pops IV from Raw IV FIFO and pushes into IV FIFO.
      2. Increment: Pops IV from Key IV FIFO. This is incremented by 1 on each Data TLV and pushed into IV FIFO.
      3. Random: A Counter Mode Deterministic Random Number Generator generates a series of 128-bit random numbers and stores them in an internal FIFO. This FIFO is popped and the requested length of random bits is pushed into IV FIFO.

A reseed interrupt may be sent via Registers if reseed counter expires.

1. Upsizer
   1. Upgrades the incoming 64-bit datapath to 128-bit datapath. Both AES and SHA engines accept data in beats of 128-bits.
   2. The delimiter is combined with last word of Data TLV to indicate end of Command. The delimiter instructs the downstream blocks to grab new commands and keys.
2. Partition
   1. Partitions the data into AAD and non-AAD components.
   2. AAD is data that is authenticated but not encrypted. This is only used in AES-GCM.
   3. Applies appropriate ANSI X.923 16-byte alignment padding for AES-XEX [PRD].
   4. Generate XTS Penultimate Signal.
3. Cipher Block
   1. AAD is skipped and forwarded to Hash Block and FIFO Downsizer.
   2. Pops a Cipher Command and uses the keys from Cipher Key FIFO and IVs from IV FIFO to execute XTS, XEX and GCM.
   3. Removes XEX ANSI X.923 Padding [PRD].
   4. All used IVs are pushed into Used IV FIFO.
   5. This block partially executes AES-GMAC. GMAC only needs two AES operations. These AES operation are carried out in this block and forwarded to Hash Block to complete AES-GMAC.
   6. Encrypted Data is sent to Hash Block and FIFO Downsizer.
   7. End of Data TLV grabs a new IV from IV FIFO.
   8. The delimiter initiates the next Cipher Command and Key.
4. FIFO Downsizer
   1. Merge AAD Data with Encrypted Data.
   2. Downgrades the 128-bit datapath to the original 64-bit datapath for the Sequencer.
   3. Endian Swap Data if enabled.
5. Hash Block
   1. Pops a Hash Command and may use the keys from Hash Key FIFO to execute SHA-256-HASH, SHA-256-HMAC and AES-GCM/GMAC.
   2. AES-GCM/GMAC completes the operations started in Cipher Block. Only Galois Field Multiplications are carried out in this block.
   3. Generated 128-bit or 256-bit Digest/ICV is sent to Footer Stitcher.
   4. The delimiter initiates the next Hash Command and Key.
6. Footer Stitcher
   1. Strap bit, strap\_integrity, indicates the location of TAG to be inserted or compared.

In Integrity Engine, Raw Data MAC is compared and full tag is inserted in Footer.

* 1. Strap bit, strap\_encrypt/decrypt, indicates TAG generation or comparison.

In Encrypt Engine, Compressed Encrypted Data MAC is generated.

In Decrypt Engine, Compressed Encrypted Data MAC is compared.

* 1. Strap bit, strap\_integrity, allows GUID and IV to be added to Footer TLV.
     1. GUID comes from Key Metadata FIFO.
     2. IV comes via CR\_IV or FRMD whichever comes last.
  2. Stitcher Command indicates
     1. Number of bits to be inserted or compared in the TAG region.

For insertions, whole TAG region in Footer is overwritten by TAG from Hash Block with zero padding.

For comparison, only the requested number of bits is compared.

* + 1. Cipher IV needs to be updated or compared. Used IV FIFO is then used to overwrite or compare the IV region of the Footer.
    2. Engine ID and Sequence Number of CMD TLV.
    3. FRMD-IV was missing.
  1. Endian Swap of IV and TAG if enabled.
  2. All error codes in Section 3.6.6 are reported.
  3. Error Frame Number is updated on errors. If the incoming Footer TLV already had errors, these fields are left untouched.

1. Sequencer
   1. In the first stage, CR\_IV TLV is manufactured (if needed). CR\_IV TLV is inserted after FRMD. This TLV holds the IV that was generated by IV Processor in Encryption.
   2. In the second stage, Sequencer pops 64-bit Data from Downsizer FIFO. Data TLV Sideband Signals are appended to each beat of Data until end of TLV.
   3. In the second stage, Sequencer accepts Footer Data from Stitcher. Footer TLV Sideband Signals are appended to each beat of Footer until end of TLV.
2. TLV Stitcher
   1. Common Block that accepts TLVs and orders them appropriately.

## Command FIFOs

There are five Command FIFOs designed to store metadata for two frames. This will allow the pipeline to keep operating efficiently.

|  |  |  |
| --- | --- | --- |
| Command FIFO Name | Metadata Stored | FIFO Depth |
| IV Command FIFO | Load Key IV, IV Operation and IV Length (96 or 128-bit) | 4 |
| Partition Command FIFO | AAD Length and Cipher Pad Operation | 2 |
| Cipher Command FIFO | Cipher Operation and AAD Present bit | 2 |
| Hash Command FIFO | Authentication Operation, Small Size bit and AAD Length | 2 |
| Stitcher Command FIFO | Authentication Operation Performed, IV based Cipher, Frame Number, FRMD IV missing indicator and CMD TLV Engine ID and Sequence number | 4 |

Table 26 - Command FIFO Sizing

## Key FIFOs

There are two Key FIFOs designed to store keys for two frames. This will allow the pipeline to keep operating efficiently.

|  |  |  |
| --- | --- | --- |
| Command FIFO Name | Metadata Stored | FIFO Depth |
| Cipher Key FIFO | Cipher Keys. XTS has the largest key: 512 bits. | 2 |
| Hash Key FIFO | Hash Key is 256-bits | 2 |

Table 27 - Key FIFO Sizing

## Other FIFOs

|  |  |  |
| --- | --- | --- |
| Command FIFO Name | Data Stored | FIFO Depth |
| Raw IV FIFO | IVs from FRMD TLVs | 2 |
| Key IV FIFO | IVs from Key TLV | 2 |
| Length FIFO | Length of Hash Data | 2 |
| Footer FIFO | Footer Data in Footer TLV | 32 |
| TLV Sideband FIFO | Word0 of FRMD, Data and Footer TLV | 4 |
| IV FIFO | IV to be used in Cipher | 2 |
| Used IV FIFO | IV to insert into Footer TLV | 4 |
| Key Metadata FIFO | GUID, Engine ID and Sequence number of Key TLV | 2 |
| Cipher FIFO | Sideband Information while Data is churning in AES | 16 |
| Cipher Result FIFO | Stores processed data while data is downsized | 32 |

Table 28 - Other FIFO Sizing

## IV Processor

IV Processor controls an AES Core through a DRNG FSM. There are three ways to generate an IV.

1. In-Band IV: Pop the IV from Raw IV FIFO and push it into IV FIFO.
2. Increment: Key IV is stored and incremented on each IV Command.
3. Generate: Implements CTR-DRBG Generate function to retrieve a pseudorandom IV [DRBG].

The generated IVs are then pushed into IV FIFO1.

# Cipher Setup Example

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CiphType | CiphMode | CiphIvLen | CiphKeyLen | Metadata setup |
| AES-256 | XEX | 16 | 64 | * Set **crypto\_mode** = 1 for encryption, 0 for decryption. * Set **CTRL0.CipherOp** = 0x2. * Set **CTRL0.CipherPadOp** = 0x1 for Padding. * Set **CTRL0.IvOp** = 0x0/0x1/0x2. For 0x0, FRMD TLV must be sent with IV. * Data sequence number as mentioned in standard is provided to Crypto as IV. * ANSI X.932 Padding is applied for Encryption and removed for Decryption. |
| AES-256 | XTS | 16 | 64 | * Set **crypto\_mode** = 1 for encryption, 0 for decryption. * Set **CTRL0.CipherOp** = 0x2. * Set **CTRL0.CipherPadOp** = 0x0 for No Padding. * Set **CTRL0.IvOp** = 0x0/0x1/0x2. For 0x0, FRMD TLV must be sent with IV. * Data sequence number as mentioned in standard is provided to Crypto as IV. * If the last block is not equal to block size, Ciphertext stealing is applied. |
| AES-256 | GCM | 12 | 32 | * Set **crypto\_mode** = 1 for encryption, 0 for decryption. * Set **CTRL0.CipherOp** = 0x1. * Set **CTRL0.AADLen** appropriately. * Set **CTRL0.IvOp** = 0x0/0x1/0x2. For 0x0, FRMD TLV must be sent with IV. |

Table 29 - Cipher Setup Example

# Hash Setup Example

|  |  |  |  |
| --- | --- | --- | --- |
| AuthType | AuthMode | HashKeyLen | Metadata setup |
| SHA-256 | HASH | 0 | * Set **crypto\_mode** = 1 for TAG generation, 2/3 for TAG check. * Set **CTRL0.AuthOp/RawAuthOp** = 0x1. * A 256-bit tag is appended on TAG Generation. * A 256/128/64-bit tag is checked on TAG check depending on Footer TLV Word 7. |
| SHA-256 | HMAC | 32 | * Set **crypto\_mode** = 1 for TAG generation, 2/3 for TAG check. * Set **CTRL0.AuthOp/RawAuthOp** = 0x2 * A 256-bit tag is appended on TAG Generation. * A 256/128/64-bit tag is checked on TAG check depending on Footer TLV Word 7. |
| AES-256 | GMAC | 32 | * Set **crypto\_mode** = 1 for TAG generation, 2/3 for TAG check. * Set **CTRL0.CipherOp** = 0x3 * Set **CTRL0.IvOp** = 0x0/0x1/0x2. For 0x0, FRMD TLV must be sent with IV. * Set **CTRL0.AADLen** appropriately. * A 128-bit tag is appended on TAG Generation * A 128/64-bit tag is checked on TAG check depending on Footer TLV Word 7. |

Table 30 - Hash Setup Example

# Test Vectors

In GCM Specification [GCM], we will run Test Case 16 in Crypto.

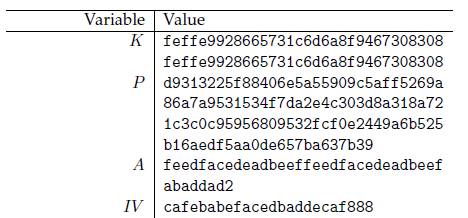


Figure 13 - Excerpt from Test Case 16 in GCM Specification

## Setup

Set crypto\_mode to 1 for Encryption.

## CMD TLV

CipherOp = 0x1

AADLen = 0x14

IVOp = 0x0

## Key TLV

CiphKey4 = 0xfeffe9928665731c

CiphKey5 = 0x6d6a8f9467308308

CiphKey6 = 0xfeffe9928665731c

CiphKey7 = 0x6d6a8f9467308308

## FRMD\_USER\_VM TLV

CiphIV0 = 0xcafebabefacedbad

CiphIV1 = 0xdecaf88800000000

We could have chosen any FRMD in Section 3.3.

## Data TLV

Word 1 = 0xfeedfacedeadbeef

Word 2 = 0xfeedfacedeadbeef

Word 3 = 0xabaddad2d9313225

Word 4 = 0xf88406e5a55909c5

Word 5 = 0xaff5269a86a7a953

Word 6 = 0x1534f7da2e4c303d

Word 7 = 0x8a318a721c3c0c95

Word 8 = 0x956809532fcf0e24

Word 9 = 0x49a6b525b16aedf5

Word 10= 0xaa0de657ba637b39

# Statistics

The following events from Crypto will be sent to Statistics Accumulator.

* Number of frames encrypted by AES-GCM.
* Number of frames encrypted by AES-XTS.
* Number of frames encrypted by AES-XEX.
* Number of frames with no encryption.
* Number of frames authenticated by AES-GMAC.
* Number of frames hashed by SHA-256.
* Number of frames authenticated by HMAC-SHA-256.
* Number of frames with no hashing/authentication.
* Number of frames that failed GCM Tag comparison.
* Number of frames that failed GMAC Tag comparison.
* Number of frames that failed SHA-256 Tag comparison.
* Number of frames that failed HMAC-SHA-256 Tag comparison.
* Number of commands that failed Key TLV-CMD TLV Sequence ID comparison.
* Number of commands that failed Key TLV-CMD TLV Engine ID comparison.

# Interrupts

The following interrupts are supported. Not all interrupts are supported in each Crypto mode as shown in Table 31.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Interrupt Name | Notes | Encrypt  Mode | Decrypt  Mode | Integrity  Mode |
| TLV Parser  Error | TLV parser detected an error | Yes | Yes | Yes |
| ID Mismatch | Sequence ID or Engine ID of CMD-TLV and Key-TLV mismatch | Yes | Yes | Yes |
| ECC Error | Uncorrectable ECC in ISM memory  (Although interrupt status bits exist for this in the CCEIP/CDDIP, the ECC was removed from the ISM and these bits are tied low.) | Yes | Yes | No |
| DRBG Seed Expired | One of the two seeds of DRBG expired. | Yes | No | No |

Table 31 - Interrupt Table

# References

[[2104](http://www.rfc-editor.org/rfc/pdfrfc/rfc2104.txt.pdf)] HMAC: Keyed-Hashing for Message Authentication

[[FIPS-180-2](http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf)] Secure Hash Standard

[[FIPS-197](http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf)] Advanced Encryption Standard (AES)

[[GCM](http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/gcm/gcm-revised-spec.pdf)] The Galois/Counter Mode of Operation (GCM)

[[XTS](http://luca-giuzzi.unibs.it/corsi/Support/papers-cryptography/1619-2007-NIST-Submission.pdf)] The XTS-AES Tweakable Block Cipher - An Extract from IEEE Std 1619-2007

[[ECB](http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf)] NIST Publication 800-38A Recommendation for Block Cipher Modes of Operation

[[CBC](http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf)] NIST Publication 800-38A Recommendation for Block Cipher Modes of Operation

[[CTR](http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf)] NIST Publication 800-38A Recommendation for Block Cipher Modes of Operation

[[DRBG](http://csrc.nist.gov/publications/nistpubs/800-90A/SP800-90A.pdf)] NIST SP800-90A Recommendation for Random Number Generation (Section 10.2)

# List of Figures

[Figure 1 - TLV Format 8](#_Toc536792095)

[Figure 2 - CMD TLV Format 9](#_Toc536792096)

[Figure 3 - Key TLV Format 11](#_Toc536792097)

[Figure 4 - FRMD\_USER\_VM TLV Format 13](#_Toc536792098)

[Figure 5 - FRM INT APP TLV Format 14](#_Toc536792099)

[Figure 6 - FRMD\_INT\_APP TLV Word 0 14](#_Toc536792100)

[Figure 7 - FRMD INT VM\_LONG TLV Format 15](#_Toc536792101)

[Figure 8 - FRMD INT VM\_SHORT TLV Format 16](#_Toc536792102)

[Figure 9 - CR IV Format 17](#_Toc536792103)

[Figure 10 - Data TLV Format 18](#_Toc536792104)

[Figure 11 - Footer TLV Format 19](#_Toc536792105)

[Figure 12- Crypto Block Diagram 24](#_Toc536792106)

[Figure 13 - Excerpt from Test Case 16 in GCM Specification 33](#_Toc536792107)

# List of Tables

[Table 1 - Clock, Reset and Miscellaneous Interface 2](#_Toc536792108)

[Table 2 - AXI-4 Crypto Stream Slave Interface 2](#_Toc536792109)

[Table 3 - AXI-4 Key Stream Slave Interface 3](#_Toc536792110)

[Table 4 - AXI-4 Crypto Stream Master Interface 4](#_Toc536792111)

[Table 5 - RBUS Slave Register Interface 5](#_Toc536792112)

[Table 6 - RBUS Master Register Interface 5](#_Toc536792113)

[Table 7 Block Cipher Modes 6](#_Toc536792114)

[Table 8 Authentication Algorithm Support 6](#_Toc536792115)

[Table 9 DRNG NIST Parameters 6](#_Toc536792116)

[Table 10 - CMD TLV Word 0 9](#_Toc536792117)

[Table 11 - CMD TLV Word 1 9](#_Toc536792118)

[Table 12 - CMD TLV Word 2 10](#_Toc536792119)

[Table 13 - Key TLV Word 0 11](#_Toc536792120)

[Table 14 - Key TLV Later Words 12](#_Toc536792121)

[Table 15 - FRMD USER VM TLV Word 0 13](#_Toc536792122)

[Table 16 - FRMD\_INT\_VM\_LONG TLV Word 0 15](#_Toc536792123)

[Table 17 - FRMD\_INT\_VM\_SHORT TLV Word 0 16](#_Toc536792124)

[Table 18 - CR IV TLV Word 0 17](#_Toc536792125)

[Table 19 - Data TLV Word 0 18](#_Toc536792126)

[Table 20 - Footer TLV Word 0 20](#_Toc536792127)

[Table 21 - Footer Word 1 to 4 20](#_Toc536792128)

[Table 22 - Footer Word 5 to 6 20](#_Toc536792129)

[Table 23 - Footer Word 7 to 10 20](#_Toc536792130)

[Table 24 - Footer Word 13 to 16 21](#_Toc536792131)

[Table 25 - Footer Word 19 22](#_Toc536792132)

[Table 26 - Command FIFO Sizing 29](#_Toc536792133)

[Table 27 - Key FIFO Sizing 29](#_Toc536792134)

[Table 28 - Other FIFO Sizing 30](#_Toc536792135)

[Table 29 - Cipher Setup Example 31](#_Toc536792136)

[Table 30 - Hash Setup Example 32](#_Toc536792137)

[Table 31 - Interrupt Table 36](#_Toc536792138)