



CIPHER

API 参考

文档版本 00B01

发布日期 2012-08-15

版权所有 © 深圳市海思半导体有限公司 2012。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HISILICON、海思和其他海思商标均为深圳市海思半导体有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受海思公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，海思公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

深圳市海思半导体有限公司

地址： 深圳市龙岗区坂田华为基地华为电气生产中心 邮编：518129

网址： <http://www.hisilicon.com>

客户服务电话： +86-755-28788858

客户服务传真： +86-755-28357515

客户服务邮箱： support@hisilicon.com



前 言

概述

CIPHER 是海思数字媒体处理平台提供的数据加解密模块，它提供了 DES、3DES 和 AES 三种加解密算法，主要用于对音视频码流进行加解密。

产品版本

与本文档相对应的产品版本如下。

产品名称	产品版本
Hi3518 芯片	V100



读者对象

本文档（本指南）主要适用于以下工程师：




- 技术支持工程师
- 软件开发工程师

符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

符号	说明
 危险	表示有高度潜在危险，如果不能避免，会导致人员死亡或严重伤害。
 警告	表示有中度或低度潜在危险，如果不能避免，可能导致人员轻微或中等伤害。



符号	说明
 注意	表示有潜在风险，如果忽视这些文本，可能导致设备损坏、数据丢失、设备性能降低或不可预知的结果。
 窍门	表示能帮助您解决某个问题或节省您的时间。
 说明	表示是正文的附加信息，是对正文的强调和补充。

修订记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

文档版本 00B01 (2012-08-15)

第 1 次发布。



目 录

前 言.....	i
1 概述.....	1
1.1 概述.....	1
1.2 使用流程.....	1
2 API 参考	2
3 数据类型.....	8
4 错误码.....	13



表格目录

表 4-1 CIPHER 模块的错误码	13
---------------------------	----



1 概述

1.1 概述

CIPHER 是海思数字媒体处理平台提供的数据加解密模块，它提供了 DES、3DES 和 AES 三种加解密算法，主要用于对音视频码流进行加解密。

1.2 使用流程

对数据进行加解密的过程如下：

1. 打开 CIPHER 设备。调用接口 `HI_UNF_CIPHER_Open` 完成。
2. 创建一路 CIPHER，并获取 CIPHER 句柄。调用接口 `HI_UNF_CIPHER_CreateHandle` 完成。
3. 配置 CIPHER 控制信息，包含密钥、初始向量、加密算法、工作模式等信息。调用接口 `HI_UNF_CIPHER_ConfigHandle` 完成。
4. 对数据进行加解密。调用接口 `HI_UNF_CIPHER_Encrypt` 或 `HI_UNF_CIPHER_Decrypt` 分别完成加密或解密。
5. 销毁 CIPHER 句柄。调用接口 `HI_UNF_CIPHER_DestroyHandle` 完成。
6. 关闭 CIPHER 设备。调用接口 `HI_UNF_CIPHER_Close` 完成。

----结束

1.3 注意事项

- 一次加解密操作的数据长度必须小于 1MB。如果数据长度大于或等于 1M，请拆分成两部分进行处理。
- CIPHER 使用用户配置的 `HI_UNF_CIPHER_BIT_WIDTH_E` 值为块进行分块加密。
- 创建一路 CIPHER，配置属性之后（假设配置的工作模式需要使用 IV 向量），后面多次调用加解密接口时，IV 向量会依次轮流使用。



例如：用户需依次加密数据 0，数据 1。向量为 a,b,c,d。用户加密完数据 0 之后，数据 0 的最后一个分块数据使用了 IV 向量中的 b 进行加密处理；此时，用户再加密数据 1 时，数据 1 的第一个分块数据将会使用 IV 向量 c 进行加密，然后依次为 d,a,b,c,d...

因此在加解密时，必须要保证两次向量使用的一致性。重新配置 CIPHER 控制信息将设置 IV 向量从第一个开始使用。

- 建议每次加解密之前都进行配置 CIPHER 控制信息操作，以使每次加解密操作都将从 IV 向量起始位置开始执行。



2 API 参考

CIPHER 提供以下 API:

- [HI_UNF_CIPHER_Open](#): 打开 CIPHER 设备。
- [HI_UNF_CIPHER_Close](#): 关闭 CIPHER 设备。
- [HI_UNF_CIPHER_CreateHandle](#): 创建一路 CIPHER, 并获取 CIPHER 句柄。
- [HI_UNF_CIPHER_DestroyHandle](#): 销毁某路 CIPHER。
- [HI_UNF_CIPHER_ConfigHandle](#): 配置 CIPHER 控制信息。
- [HI_UNF_CIPHER_Encrypt](#): 对数据进行加密。
- [HI_UNF_CIPHER_Decrypt](#): 对数据进行解密。

HI_UNF_CIPHER_Open

【描述】

打开 CIPHER 设备。

【语法】

```
HI_S32 HI_UNF_CIPHER_Open(HI_VOID);
```

【参数】

无。

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件: hi_error_ecs.h、hi_type.h、hi_unf_ecs.h、priv_cipher.h
- 库文件: libhi_cipher.a



【注意】

无。

【举例】

无。

HI_UNF_CIPHER_Close

【描述】

关闭 CIPHER 设备。

【语法】

```
HI_S32 HI_UNF_CIPHER_Close(HI_VOID);
```

【参数】

无。

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_error_ecs.h、hi_type.h、hi_unf_ecs.h、priv_cipher.h
- 库文件：libhi_cipher.a

【注意】

无。

【举例】

无。

HI_UNF_CIPHER_CreateHandle

【描述】

创建一路 CIPHER，并获取 CIPHER 句柄。

【语法】

```
HI_S32 HI_UNF_CIPHER_CreateHandle(HI_HANDLE* phCipher);
```

【参数】



参数名称	描述	输入/输出
phCipher	CIPHER 句柄指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_error_ecs.h、hi_type.h、hi_unf_ecs.h、priv_cipher.h
- 库文件：libhi_cipher.a

【注意】

- phCipher 不能为空。
- 该句柄将用于数据加解密时的输入。
- 最大支持 7 路 cipher。
- 不支持单分组加解密

【举例】

无。

HI_UNF_CIPHER_DestroyHandle

【描述】

销毁一路 CIPHER。

【语法】

```
HI_S32 HI_UNF_CIPHER_DestroyHandle(HI_HANDLE hCipher);
```

【参数】

参数名称	描述	输入/输出
hCipher	CIPHER 句柄。	输入

【返回值】

返回值	描述
0	成功。



返回值	描述
非 0	参见错误码。

【需求】

- 头文件：hi_error_ecs.h、hi_type.h、hi_unf_ecs.h、priv_cipher.h
- 库文件：libhi_cipher.a

【注意】

支持重复销毁。

【举例】

无。

HI_UNF_CIPHER_ConfigHandle

【描述】

配置 CIPHER 控制信息。详细配置请参见结构体 HI_UNF_CIPHER_CTRL_S。

【语法】

```
HI_S32 HI_UNF_CIPHER_ConfigHandle(HI_HANDLE hCipher,  
HI_UNF_CIPHER_CTRL_S* pstCtrl);
```

【参数】

参数名称	描述	输入/输出
hCipher	CIPHER 句柄。	输入
pstCtrl	控制信息指针。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_error_ecs.h、hi_type.h、hi_unf_ecs.h、priv_cipher.h
- 库文件：libhi_cipher.a

【注意】



控制信息指针不能为空。

【举例】

无。

HI_UNF_CIPHER_Encrypt

【描述】

对数据进行加密。

【语法】

```
HI_S32 HI_UNF_CIPHER_Encrypt(HI_HANDLE hCipher, HI_U32 u32SrcPhyAddr,  
HI_U32 u32DestPhyAddr, HI_U32 u32ByteLength);
```

【参数】

参数名称	描述	输入/输出
hCipher	CIPHER 句柄。	输入
u32SrcPhyAddr	源数据（待加密的数据）的物理地址。	输入
u32DestPhyAddr	存放加密结果的物理地址。	输出
u32ByteLength	数据的长度（单位：字节）。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_error_ecs.h、hi_type.h、hi_unf_ecs.h、priv_cipher.h
- 库文件：libhi_cipher.a

【注意】

- CIPHER 句柄必须已创建。
- 可多次调用。
- 数据的长度至少为 16 字节，且不能大于或等于 1024*1024 字节。

【举例】

无。



HI_UNF_CIPHER_Decrypt

【描述】

对数据进行解密。

【语法】

```
HI_S32 HI_UNF_CIPHER_Decrypt(HI_HANDLE hCipher, HI_U32 u32SrcPhyAddr,  
HI_U32 u32DestPhyAddr, HI_U32 u32ByteLength);
```

【参数】

参数名称	描述	输入/输出
hCipher	CIPHER 句柄。	输入
u32SrcPhyAddr	源数据（待解密的数据）的物理地址。	输入
u32DestPhyAddr	存放解密结果的物理地址。	输出
u32ByteLength	数据的长度（单位：字节）。	输入

【返回值】

返回值	描述
0	成功。
非 0	参见错误码。

【需求】

- 头文件：hi_error_ecs.h、hi_type.h、hi_unf_ecs.h、priv_cipher.h
- 库文件：libhi_cipher.a

【注意】

- CIPHER 句柄必须已创建。
- 可多次调用。
- 数据的长度至少为 16 字节，且不能大于或等于 1024x1024 字节。

【举例】

无。



3 数据类型

相关数据类型、数据结构定义如下：

- [HI_HANDLE](#)：定义 CIPHER 的句柄类型。
- [HI_UNF_CIPHER_WORK_MODE_E](#)：定义 CIPHER 工作模式。
- [HI_UNF_CIPHER_ALG_E](#)：定义 CIPHER 加密算法。
- [HI_UNF_CIPHER_KEY_LENGTH_E](#)：定义 CIPHER 密钥长度。
- [HI_UNF_CIPHER_BIT_WIDTH_E](#)：定义 CIPHER 加密位宽。
- [HI_UNF_CIPHER_CTRL_S](#)：定义 CIPHER 控制信息结构体。

HI_HANDLE

【说明】

定义 CIPHER 的句柄类型。

【定义】

```
typedef HI_U32 HI_HANDLE;
```

【成员】

无。

【注意事项】

无。

【相关数据类型及接口】

无。

HI_UNF_CIPHER_WORK_MODE_E

【说明】

定义 CIPHER 工作模式。

【定义】



```
typedef enum hiHI_UNF_CIPHER_WORK_MODE_E
{
    HI_UNF_CIPHER_WORK_MODE_ECB = 0x0,
    HI_UNF_CIPHER_WORK_MODE_CBC = 0x1,
    HI_UNF_CIPHER_WORK_MODE_CFB = 0x2,
    HI_UNF_CIPHER_WORK_MODE_OFB = 0x3,
    HI_UNF_CIPHER_WORK_MODE_CTR = 0x4,
    HI_UNF_CIPHER_WORK_MODE_BUTT = 0x5
};
```

【成员】

成员名称	描述
HI_UNF_CIPHER_WORK_MODE_ECB	ECB（Electronic CodeBook）模式
HI_UNF_CIPHER_WORK_MODE_CBC	CBC（Cipher Block Chaining）模式
HI_UNF_CIPHER_WORK_MODE_CFB	CFB（Cipher FeedBack）模式
HI_UNF_CIPHER_WORK_MODE_OFB	OFB（Output FeedBack）模式
HI_UNF_CIPHER_WORK_MODE_CTR	CTR（Counter）模式

【注意事项】

无。

【相关数据类型及接口】

无。

HI_UNF_CIPHER_ALG_E

【说明】

定义 CIPHER 加密算法。

【定义】

```
typedef enum hiHI_UNF_CIPHER_ALG_E
{
    HI_UNF_CIPHER_ALG_DES = 0x0,
    HI_UNF_CIPHER_ALG_3DES = 0x1,
    HI_UNF_CIPHER_ALG_AES = 0x2,
    HI_UNF_CIPHER_ALG_BUTT = 0x3
}HI_UNF_CIPHER_ALG_E;
```

【成员】



成员名称	描述
HI_UNF_CIPHER_ALG_DES	DES 算法
HI_UNF_CIPHER_ALG_3DES	3DES 算法
HI_UNF_CIPHER_ALG_AES	AES 算法

【注意事项】

无。

【相关数据类型及接口】

无。

HI_UNF_CIPHER_KEY_LENGTH_E

【说明】

定义 CIPHER 密钥长度。

【定义】

```
typedef enum hiHI_UNF_CIPHER_KEY_LENGTH_E
{
    HI_UNF_CIPHER_KEY_AES_128BIT = 0x0,
    HI_UNF_CIPHER_KEY_AES_192BIT = 0x1,
    HI_UNF_CIPHER_KEY_AES_256BIT = 0x2,
    HI_UNF_CIPHER_KEY_DES_3KEY = 0x2,
    HI_UNF_CIPHER_KEY_DES_2KEY = 0x3,
}HI_UNF_CIPHER_KEY_LENGTH_E;
```

【成员】

成员名称	描述
HI_UNF_CIPHER_KEY_AES_128BIT	AES 运算方式下采用 128bit 密钥长度
HI_UNF_CIPHER_KEY_AES_192BIT	AES 运算方式下采用 192bit 密钥长度
HI_UNF_CIPHER_KEY_AES_256BIT	AES 运算方式下采用 256bit 密钥长度
HI_UNF_CIPHER_KEY_DES_3KEY	3DES 运算方式下采用 3 个 key
HI_UNF_CIPHER_KEY_DES_2KEY	3DES 运算方式下采用 2 个 key

【注意事项】

- AES 的密钥长度可以为 128bit，192bit 或 256bit。



- 3DES 算法的密钥长度可以为 2 个或 3 个 key，一个 key 指 DES 加密所用的密钥，它的长度为 64bit。
- DES 算法该项无效。

【相关数据类型及接口】

无。

HI_UNF_CIPHER_BIT_WIDTH_E

【说明】

定义 CIPHER 加密位宽。

【定义】

```
typedef enum hiHI_UNF_CIPHER_BIT_WIDTH_E
{
    HI_UNF_CIPHER_BIT_WIDTH_64BIT = 0x0,
    HI_UNF_CIPHER_BIT_WIDTH_8BIT = 0x1,
    HI_UNF_CIPHER_BIT_WIDTH_1BIT = 0x2,
    HI_UNF_CIPHER_BIT_WIDTH_128BIT = 0x3,
}HI_UNF_CIPHER_BIT_WIDTH_E;
```

【成员】

成员名称	描述
HI_UNF_CIPHER_BIT_WIDTH_64BIT	64bit 位宽
HI_UNF_CIPHER_BIT_WIDTH_8BIT	8bit 位宽
HI_UNF_CIPHER_BIT_WIDTH_1BIT	1bit 位宽
HI_UNF_CIPHER_BIT_WIDTH_128BIT	128bit 位宽

【注意事项】

无。

【相关数据类型及接口】

无。

HI_UNF_CIPHER_CTRL_S

【说明】

定义 CIPHER 控制信息结构体。

【定义】

```
typedef struct hiHI_UNF_CIPHER_CTRL_S
```



```
{  
    HI_U32                u32Key[8];  
    HI_U32                u32IV[4];  
    HI_BOOL               bKeyByCA;  
    HI_UNF_CIPHER_ALG_E   enAlg;  
    HI_UNF_CIPHER_BIT_WIDTH_E enBitWidth;  
    HI_UNF_CIPHER_WORK_MODE_E enWorkMode;  
    HI_UNF_CIPHER_KEY_LENGTH_E enKeyLen;  
} HI_UNF_CIPHER_CTRL_S;
```

【成员】

成员名称	描述
u32Key[8]	密钥
u32IV[4]	初始向量
bKeyByCA	是否使用高安全 CA 加密或解密 KEY
enAlg	加密算法
enBitWidth	加密或解密的位宽
enWorkMode	工作模式
enKeyLen	密钥长度

【注意事项】

- ECB 模式、CTR 模式下不需要初始向量。
- Hi3518 芯片只支持 bKeyByCA 为 HI_FALSE，需要 CA 模块（芯片无改模块）支持。

【相关数据类型及接口】

无。



4 错误码

CIPHER 提供的错误码如下。

表4-1 CIPHER 模块的错误码

错误代码	宏定义	描述
0x804D0001	HI_ERR_CIPHER_NOT_INIT	设备未初始化
0x804D0002	HI_ERR_CIPHER_INVALID_HANDLE	Handle 号无效
0x804D0003	HI_ERR_CIPHER_INVALID_POINT	参数中有空指针
0x804D0004	HI_ERR_CIPHER_INVALID_PARA	无效参数
0x804D0005	HI_ERR_CIPHER_FAILED_INIT	初始化失败
0x804D0006	HI_ERR_CIPHER_FAILED_GETHANDLE	获取 handle 失败
-1	HI_FAILURE	操作失败