

Gmini3860 USB-SE 测试使用说明

V1.0

修订日期	版本	修订内容
2025 年 3 月	V1.0	初版

1. 概述

本文主要介绍 USB-SE 模块的测试工具使用，包括算法功能测试、密钥管理测试、性能测试等。

2. 工具

2.1. 目录结构

测试工具在 tool 目录下，测试工具里包含 USB-SE 模块的驱动库文件、软算法库文件和测试工具文件。

crypto_sw	#软算法库
libs	#USB-SE 驱动库（基于 libusb）
kmt.c	#密钥管理工具代码
main.c	#算法测试用例代码
ktest.c	#密钥管理测试工具代码

2.2. 编译部署

安装依赖 libusb-1.0

CentOS:

```
yum install libusb-devel
```

Ubuntu:

```
apt install libusb-1.0-0-dev
```

编译驱动库:

```
cd libs;
```

```
make
```

```
make install
```

编译密钥管理工具(kmt)/测试工具(ktest)/性能测试工具(se)

make

编译完生成 kmt,ktest,se 3个可执行程序

3. 测试

3.1. 密钥管理测试

密钥管理测试主要使用 kmt 和 ktest 工具进行测试。

设备插入后可识别到一个 usb 设备，可通过 lsusb 命令查看：

```
[root@localhost usb_u]# lsusb
Bus 002 Device 006: ID 16c3:1234
Bus 002 Device 002: ID 2109:0815 VIA Labs, Inc.
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 003: ID 0403:6010 Future Technology Devices International, Ltd FT2232C/D/H Dual UART/FIFO IC
Bus 001 Device 002: ID 2109:2815 VIA Labs, Inc.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

设备访问前要先获取访问权限，通过命令【./kmt 0】获取权限。

```
[root@localhost usb_u]# ./kmt 0
random[32]
D3 71 B5 EC BD 2F CB 46 A5 0C 27 65 1D 54 4B 73
CA 58 21 E5 C7 C7 FE 2D 8D 65 9C 31 02 71 DB AD

MCF_GetDevicePermission Success!
```

※ **!!! 如果设备是第一次使用需要先导入 SN、设备密钥和加密密钥。**

- 【./kmt 22】命令导入 SN，kmt 工具里会导入一个固定的 SN，只能导入一次，第二次导入会返回失败
- 【./kmt 23】命令导入设备密钥，kmt 工具里会导入一个固定的设备密钥，只能导入一次，第二次导入会返回失败
- 【./kmt 24】命令导入加密密钥，kmt 工具里会导入一个固定的加密密钥，可以多次导入

上述准备完成后可进行密钥管理测试。

密钥的访问和管理都需要权限，首先要获取设备使用权限，获取权限后可以一直使用设备，知道设备断电或者主动释放权限。

```
[root@localhost usb_u]# ./kmt 0
random[32]
90 E8 DB 06 54 21 A1 5B 3A B8 69 10 BF CB 2C 79
84 B7 3D 6B 7C 1A 14 B5 58 F9 05 BD F9 35 89 C3

MCF_GetDevicePermission Success!
[root@localhost usb_u]#
```

获取密钥状态：默认所有密钥均为空

```
[root@localhost usb_u]# ./kmt 26  
MCF_GetKeyInfo Success!  
cipher(16) sm2(16) rsa1024(16) rsa2048(16)
```

```
SM2: 16  
0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0
```

```
RSA1024: 16  
0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0
```

```
RSA2048: 16  
0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0
```

```
CIPHER: 16  
0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0
```

3.1.1. 密钥生成测试

依次生成一下密钥进行测试：

生成 1、2 号对称密钥

生成 1、2 号 SM2 密钥

生成 1、2 号 RSA1024 密钥

生成 1、2 号 RSA2048 密钥

```

[root@localhost usb_u]# ./kmt 5 1
MCF_GenerateCipherKey Success!
[root@localhost usb_u]# ./kmt 5 2
MCF_GenerateCipherKey Success!
[root@localhost usb_u]# ./kmt 2 1
MCF_GenerateECCKey Success!
[root@localhost usb_u]# ./kmt 2 2
MCF_GenerateECCKey Success!
[root@localhost usb_u]# ./kmt 3 1
^[[AMCF_GenerateRSAKey Success!
[root@localhost usb_u]# ./kmt 3 2
MCF_GenerateRSAKey Success!
[root@localhost usb_u]# ./kmt 4 1
MCF_GenerateRSAKey Success!
[root@localhost usb_u]# ./kmt 4 2
MCF_GenerateRSAKey Success!
[root@localhost usb_u]# ./kmt 26
MCF_GetKeyInfo Success!
cipher(16) sm2(16) rsa1024(16) rsa2048(16)

SM2: 16
0 1 1 0 0 0 0 0
0 0 0 0 0 0 0 0

RSA1024: 16
0 1 1 0 0 0 0 0
0 0 0 0 0 0 0 0

RSA2048: 16
0 1 1 0 0 0 0 0
0 0 0 0 0 0 0 0

CIPHER: 16
0 1 1 0 0 0 0 0
0 0 0 0 0 0 0 0
[root@localhost usb_u]#

```

3.1.2. 密钥导入测试

密钥导入只能向空位置导入密钥，如果导入的位置已经存在密钥，则返回失败。删除密钥后可重新导入。

- 使用方法：

a.out op EKID IV KID

op：操作码，密钥导入、导出

EKID：导入时使用的保护密钥，==16 时表示使用加密密钥，0~15 时表示使用用户密钥，（用户密钥必须存在，否则直接返回失败）

IV：导入时使用的，向量值，==0 表示使用随机数 IV，==1 时表示使用固定 IV 值。

KID：导入密钥的位置，范围为 0~15

● 测试

使用加密密钥，使用随机数 IV 保护导入，依次导入一下密钥进行测试。

导入 3 号对称密钥

导入 3 号 SM2 密钥

导入 3 号 RSA1024 密钥

导入 3 号 RSA2048 密钥

```
[root@localhost usb_u]# ./kmt 6 16 0 3
usage: a.out op EKID IV KID
ECCKey[96]
09 4C C2 FD B6 66 3C 76 F1 7C A7 5E A9 04 7D EB
6B 79 E0 CD 31 E9 97 9A 3A 8D 4A 5B 06 9A D1 0F
D8 6F 33 2E 49 6D 85 98 56 E5 63 C3 A4 E5 56 68
13 AF 0A 8C 77 48 72 64 AB E7 6F A2 36 4D 78 7B
C7 F9 08 1B 66 33 71 95 B4 75 44 63 92 B9 8E E8
28 38 E9 A6 AD 54 4F 0C 52 07 DF FE 4D 83 95 DA
EK[16]
C0 C0 C0 C0 C0 C0 C0 C0 C0 C0 C0 C0 C0 C0 C0
SE_UIV[16]
D1 4A 30 85 09 44 F1 41 DD 51 19 10 4C CD A1 4B
ECCKeyCipher[96]
C0 1F 28 A0 A4 A6 9E F3 7D AC F6 F2 79 B9 B2 98
52 BC 7C FF 6E 8F 61 09 52 B9 A8 31 EB 33 7E 3C
F5 03 F8 92 85 0E DF CF DB C9 A1 73 4F BC 35 1A
9C E6 E2 30 AD F8 96 81 FB 17 0C BA BE F8 C9 2E
A2 82 6F D0 E3 60 C2 18 4B 6F A0 38 48 47 49 4A
B4 F6 15 D5 2C CA 48 BF C9 ED 1A 9C 01 03 59 39
MCF ImportECCKey Success!
```

```
[root@localhost usb_u]# ./kmt 7 16 0 3
usage: a.out op EKID IV KID
RSAKey[704]
9D 4D A6 2B A4 54 81 E4 33 FA DD BE 63 BB 25 01
80 A8 7F 17 0C AF 34 48 F7 C4 BF DA D5 A7 24 08
E6 E3 8A AF 8A F9 00 94 D3 E0 DC D2 81 98 6F 19
82 3F BE A6 EF A6 4C 2B F3 25 A1 56 53 47 CA F9
F3 FA 27 80 EC C0 5C 26 13 CE 6A 6C 17 EB B8 9F
B4 D5 F3 65 3D D9 27 3E B7 1D 54 3F D0 98 26 98
39 D0 41 29 DB 2D 4E 26 7F 7F 98 9F AE 7A 9F 77
```

```
[root@localhost usb_u]# ./kmt 8 16 0 3
usage: a.out op EKID IV KID
RSAKey[1408]
7F CA F2 D0 0B A5 38 0B 40 A8 04 5B BC 64 C4 E6
67 E5 56 00 5B 38 6B 86 F6 CB 06 D9 C0 6B 76 8C
E4 93 DD 62 47 26 27 C0 D5 AC E2 C1 68 46 E9 26
34 D1 48 5B F8 03 22 C7 41 4E 4E C5 C4 9E 58 24
FD F2 FB 7C B0 34 9D 29 F9 AF 69 96 43 E0 29 CC
00 1A 40 6E 1A 42 9A 22 13 03 BE 11 5A 35 74 43
6F 28 76 3C 22 81 EE B7 4D D9 00 84 8D 4E CF 01
7F 03 10 C3 F9 EE B7 5A 1E 97 72 E5 D3 61 C5 32
39 A4 5C 2F 1A 86 29 97 83 16 19 99 EA 1E FF 33
```

```

[root@localhost usb_u]# ./kmt 9 16 0 3
usage: a.out op EKID IV KID
CipherKey[16]
13 13 13 13 13 13 13 13 13 13 13 13 13 13 13
EK[16]
C0 C0 C0 C0 C0 C0 C0 C0 C0 C0 C0 C0 C0 C0
SE_UIV[16]
7E F5 9C 60 71 84 F4 36 44 D3 37 5D D9 76 96 E5
CipherKeyCipher[16]
FB 33 5F 0D 02 94 35 3D FE 6E 9A F8 5E 1C D4 B8
MCF_ImportCipherKey Success!

[root@localhost usb_u]# ./kmt 26
MCF_GetKeyInfo Success!
cipher(16) sm2(16) rsa1024(16) rsa2048(16)

SM2: 16
0 1 1 1 0 0 0 0
0 0 0 0 0 0 0 0

RSA1024: 16
0 1 1 1 0 0 0 0
0 0 0 0 0 0 0 0

RSA2048: 16
0 1 1 1 0 0 0 0
0 0 0 0 0 0 0 0

CIPHER: 16
0 1 1 1 0 0 0 0
0 0 0 0 0 0 0 0
[root@localhost usb_u]#

```

3.1.3. 密钥导出测试

密钥导出只能导出已存在的密钥，如果密钥不存在或者超出范围（0~15）则返回失败。

● 使用方法：

a.out op EKID IV KID

op：操作码，密钥导入、导出

EKID：导出时使用的保护密钥，==16时表示使用加密密钥，0~15时表示使用用户密钥，（用户密钥必须存在，否则直接返回失败）

IV：导出时使用的，向量值，==0表示使用随机数 IV，==1时表示使用固定 IV 值。

KID：导出密钥的位置，范围为 0~15

● 测试

导出密钥，使用加密密钥进行保护，使用随机数 IV，依次导出以下密钥进行测试。

导出 1 号对称密钥

导出 1 号 SM2 密钥

导出 1 号 RSA1024 密钥

导出 1 号 RSA2048 密钥

```
[root@localhost usb_u]# ./kmt 10 16 0 1
usage: a.out op EKID IV KID
MCF_ExportECCKey[96]
  6D E0 0B 1E FF BE 90 C5 1F B8 73 9E 92 7B 98 51
  E4 DD B9 B1 4A EC CA 64 80 E5 A3 0C 42 5D 1E A7
  D4 1B DB 89 0F 02 52 80 4D 3D F5 A3 DA 7A 31 2F
  67 25 92 A0 69 87 39 92 9E 1A CB 8C 70 74 65 21
  1C A7 73 AC 3D 56 9C 9A 7F F1 7C 7B 9C D2 09 86
  00 F2 6A 01 7C 5D 1F 63 DD 6E 27 CB A4 7C 4B 1C
[root@localhost usb_u]# ./kmt 11 16 0 1
usage: a.out op EKID IV KID
MCF_ExportRSAKey[704]
  AA 1F 30 D1 CC 5F 81 6C 53 93 18 66 21 75 E4 67
  6F AB C4 E1 88 71 47 B4 7D 54 4A 84 36 E0 89 1E
  CA E8 33 FA D3 BB 1B E4 9D 13 FA BE 04 94 FA 17
  D0 CC DA F4 0F AC A4 0C 6A 92 A4 C0 73 DC A0 96
  D6 E2 FE 46 22 B9 B7 29 71 71 7C 6E 0D 5E DC 8F
  47 CD 2D 65 3D 46 8A 28 24 06 18 60 DE C9 F0 1B
  B4 81 00 71 62 2E E4 8D 55 8E 7E 79 23 10 D4 49
  74 E4 87 13 64 88 1B 8F 53 15 8B 83 7B 85 55 88
```

```
[root@localhost usb_u]# ./kmt 12 16 0 1
usage: a.out op EKID IV KID
MCF_ExportRSAKey[1408]
  93 4D A1 2A 81 3A 2C 68 70 3A 4E CC 73 DC 33 35
  43 5B D7 E7 3A AE E4 53 C0 EC 17 47 54 69 FC 9D
  A0 56 6E 2F 6E DE 98 8E D2 F2 E8 3F 13 CA 3B 11
  BD 2E 84 09 8D 61 E1 6A AA C4 43 32 DC 85 F9 FE
  8F C2 6A 8C 30 11 94 A7 96 E2 F4 AE 95 77 D6 2A
  63 9B 19 77 F6 23 E9 34 47 02 45 EC D5 DF 5C D0
  F1 92 CD 48 4D 75 5A CF C5 29 86 E2 FF C5 76 F8
  13 9A 71 ED B8 B2 DC D7 2B FC 51 FA 0E 28 D0 83
  6D 4E B0 E6 33 54 2A 65 98 00 DF 53 0E B7 56 32
  43 AC 46 02 C7 8D B2 E0 9A 41 BD 0B 7C 2A 4E 92
```

```
[root@localhost usb_u]# ./kmt 13 16 0 1
usage: a.out op EKID IV KID
CipherKeyCipher[16]
  0B 35 11 3F 65 67 27 FD D4 8A 6C 2D DF 54 1F DD
```

3.1.4. 密钥销毁测试

依次销毁以下密钥进行测试。

销毁 1、3 号对称密钥

销毁 1、3 号 SM2 密钥

销毁 1、3 号 RSA1024 密钥

销毁 1、3 号 RSA2048 密钥

```

[root@localhost usb_u]# ./kmt 14 1
MCF_DestroyECCKey Success!
[root@localhost usb_u]# ./kmt 14 3
MCF_DestroyECCKey Success!
[root@localhost usb_u]# ./kmt 15 1
MCF_DestroyRSAKey Success!
[root@localhost usb_u]# ./kmt 15 3
MCF_DestroyRSAKey Success!
[root@localhost usb_u]# ./kmt 16 1
MCF_DestroyRSAKey Success!
[root@localhost usb_u]# ./kmt 16 3
MCF_DestroyRSAKey Success!
[root@localhost usb_u]# ./kmt 17 1
MCF_DestroyCipherKey Success!
[root@localhost usb_u]# ./kmt 17 3
MCF_DestroyCipherKey Success!
[root@localhost usb_u]# ./kmt 26
MCF_GetKeyInfo Success!
cipher(16) sm2(16) rsa1024(16) rsa2048(16)

SM2: 16
0 0 1 0 0 0 0 0
0 0 0 0 0 0 0 0

RSA1024: 16
0 0 1 0 0 0 0 0
0 0 0 0 0 0 0 0

RSA2048: 16
0 0 1 0 0 0 0 0
0 0 0 0 0 0 0 0

CIPHER: 16
0 0 1 0 0 0 0 0
0 0 0 0 0 0 0 0
[root@localhost usb_u]# █

```

3.1.5. 用户管理测试

1. 枚举用户
2. 创建用户
3. 登录用户
4. 登出用户

```

[root@localhost usb_u]# ./kmt 29

枚举用户信息：
    用户类型:NULL, 索引:1, 注册状态:注销, 登录状态:登出
    用户类型:NULL, 索引:2, 注册状态:注销, 登录状态:登出
    用户类型:NULL, 索引:3, 注册状态:注销, 登录状态:登出
    用户类型:NULL, 索引:1, 注册状态:注销, 登录状态:登出
[root@localhost usb_u]# ./kmt 30

请插入 Ukey, 输入待增加用户类型, 1 - 管理员, 2 - 操作员: 1
请输入待添加用户索引, 范围 [1 ~ 3]: 2
请输入 16 字节 UKEY **默认保护口令**: 1234567812345678

SDFE_AddUser Success!
[root@localhost usb_u]# ./kmt 31

请插入 Ukey 并输入用户类型, 1 - 管理员, 2 - 操作员: 1
请输入用户索引, 范围 [1 ~ 3]: 2
请输入 16 字节 Ukey 保护口令: 1234567812345678

SDFE_LoginUser Success!
[root@localhost usb_u]# ./kmt 32

请插入 Ukey 并输入用户类型, 1 - 管理员, 2 - 操作员: 1
请输入用户索引, 范围 [1 ~ 3]: 2

SDFE_LogoutUser Success!

```

3.2.算法测试

- 使用方法

usage : algo A/Sync Start End size loops check iKey IKID IVID

algo: 算法

A/Sync: 同步、异步模式, 目前支持同步, 固定为 0

Start: 固定为 0

End: 测试线程数

size: 单包长度

loops: 测试 loop 次数

check: 是否检查结果, ==1 时加过会和软件计算结果对比, ==0 时不检查结果

iKey:是否使用内部密钥, ==1 时使用内部密钥, ==0 时使用外部密钥 (程序中的固定值)

IKID: iKey==1 时有效, 表示使用的内部密钥索引值, 范围 0~15, 选择的密钥必须存在, 如果不密钥不存在则直接返回失败

IVID: 未使用, 测试程序中使用了固定的 IV 值。

3.2.1. 外部密钥测试

SM4-CBC 算法测试:

```
[root@localhost usb_u]# ./se 1 0 0 1 1024 1 1 0 0 0
th_p[0].algo=1 SM4_CBC size = 1024
th_p→algo =1

SM4_CBC : cost: 0 ms data: 0.000000 Mb perf: 19.728537 Mbps 2525.252686 Ops

[root@localhost usb_u]# ./se 1 0 0 3 1024 1 1 0 0 0
th_p[0].algo=1 SM4_CBC size = 1024
th_p[1].algo=1 SM4_CBC size = 1024
th_p[2].algo=1 SM4_CBC size = 1024
th_p→algo =1
th_p→algo =1
th_p→algo =1

SM4_CBC : cost: 0 ms data: 0.000000 Mb perf: 48.225307 Mbps 6172.839355 Ops

[root@localhost usb_u]#
```

3.2.2. 内部密钥测试

SM4-CBC 算法测试, 使用2 号密钥测试

```
0 0 0 0 0 0 0 0
[root@localhost usb_u]# ./se 1 0 0 1 1024 1 1 1 2 0
th_p[0].algo=1 SM4_CBC size = 1024
th_p→algo =1

SM4_CBC : cost: 0 ms data: 0.000000 Mb perf: 15.562750 Mbps 1992.031982 Ops
```

3.3.性能测试

单线程和多线程性能测试

```
[root@localhost usb_u]# ./se 1 0 0 1 1024 10000 1 0 2 0
th_p[0].algo=1 SM4_CBC      size = 1024
th_p→algo =1

SM4_CBC        : cost: 800 ms  data: 72.000000 Mb perf: 97.598549 Mbps 12492.614258 Ops

[root@localhost usb_u]# ./se 1 0 0 3 1024 10000 1 0 2 0
th_p[0].algo=1 SM4_CBC      size = 1024
th_p[1].algo=1 SM4_CBC      size = 1024
th_p[2].algo=1 SM4_CBC      size = 1024
th_p→algo =1
th_p→algo =1
th_p→algo =1

SM4_CBC        : cost: 1208 ms  data: 216.000000 Mb perf: 193.896423 Mbps 24818.742188 Ops

[root@localhost usb_u]# █
```

4. 参考性能

对称	USB2.0 Mbps	USB3.0 Mbps	摘要算法	USB2.0 Mbps	USB3.0 Mbps
SM4_ECB	128	207	HASH_SM3	83	91
SM4_CBC	129	192	HASH_SHA1	83	90
SM4_CFB	127	192	HASH_SHA256	86	91
SM4_OFB	129	190	HASH_SHA384	86	91
SM4_XTS	124	189	HASH_SHA512	86	92
SM4_CTR	127	192	ONETIME_HASH_SM3	153	174
SM1_ECB	122	207	ONETIME_HASH_SHA1	149	170
SM1_CBC	125	205	ONETIME_HASH_SHA256	152	173
SM1_CFB	119	206	ONETIME_HASH_SHA384	143	175
SM1_OFB	128	209	ONETIME_HASH_SHA512	147	176
SM1_CTR	127	203	HMAC_SM3	145	173
SM7_ECB	123	183	HMAC_SHA1	144	171
AES128_ECB	126	208	HMAC_SHA256	148	173
AES128_CBC	128	208	HMAC_SHA384	144	176
AES128_CFB	127	204	HMAC_SHA512	150	175
AES128_OFB	123	205			
AES128_XTS	128	204	链式	USB2.0 Mbps	USB3.0 Mbps
AES128_CTR	122	145	SM4_SM3	109	118
AES192_ECB	125	205	AES128_SHA1	104	123
AES192_CBC	122	204	SM3_SM4	90	102
AES192_CFB	123	205	SHA1_AES128	91	103
AES192_OFB	123	200			
AES192_CTR	123	204	公钥算法	USB2.0 OPS	USB3.0 OPS
AES256_ECB	123	204	SM2_SIGN	9351	9384
AES256_CBC	123	201	SM2_VERIFY	4990	4995
AES256_CFB	122	202	SM2_ENC	2549	2560
AES256_OFB	121	198	SM2_DEC	3197	3190
AES256_XTS	122	190	RSA1024_SIGN	739	738
AES256_CTR	122	202	RSA1024_VERIFY	2841	2841
DES-ECB	125	196	RSA2048_SIGN	162	162
DES-CBC	123	196	RSA2048_VERIFY	1491	1491
DES-CTR	124	195			
TDES-ECB	122	152			
TDES-CBC	117	152			
3DES-ECB	121	151			
3DES-CBC	121	150			