SDK Manual

Swiper

Magnetic Stripe Reader for iPhone 3G, 3GS, 4 and iPod Touch

80097503-001-A
10/12/2010

Swiper User Manual

Revision History

Rev	Date	Description of Changes	By
A	10/12/2010	Initial Release	JW

Table of Contents

1 Introduction

Swiper is a snap-on magnetic stripe reader designed to work with iPhone and iPod Touch. The reader delivers superior reading performance with the ability to encrypt sensitive card data. The data encryption process prevents card holder information from being accessed when the data is stored or in transit, so the data remains secure from end to end. The reader fully supports TDES and AES data encryption using DUKPT key management method.

2 Features and Benefits

- Small form factor for comfort and mobility
- No external power supply required
- Mini USB port enables iPhone/iPod Touch to be charged through an external cable
- Bi-directional card reading
- Reads encoded data that meets ANSI/ISO/AAMVA standards and some custom formats such as ISO track 1 format on track 2 or 3
- Reads up to three tracks of card data
- Provides clear text confirmation data including card holder's name and a portion of the PAN as part of the Masked Track Data
- PCI compliant

3 Specifications

Communication Interface:	UART
Power Consumption:	5 mA during card swipe, 3 mA when idle
Magnetic Stripe Reader:	3 track bi-directional reading capabilities
Operating Life:	300,000 cycle minimum
Operating Environment:	0 °C to 55 °C (32 °F to 131 °F) non-condensing
Storage Environment:	-30 °C to 70 °C (-22 °F to 158 °F) non-condensing
Dimensions:	95 mm (L) x 29.88 mm (H) x 70.78 mm (W)
Cable (optional):	Standard USB A to Mini B Cable

4 iMag Firmware Command

4.1 Setting Command

The setting data command is a collection of many function setting blocks and its format is as follows.

Command
<STX><S><FuncSETBLOCK1>…<FuncBLOCKn><ETX><LRC>

Response
<ACK> for successful settings
or <NAK> for wrong commands such as invalid funcID, length and value

Each function-setting block <FuncSETBLOCK> has following format:

<FuncID><Len><FuncData>

Where:
<FuncID> is the one byte ID identifying the function being set
<Len> is a one byte length count for the function-setting block <FuncData>.
<FuncData> is the current setting for this function. It has the same format as in the sending command for this function.

Example:
Set DUKPT key management
CMD: \02\53\58\01\31\03\3A
OUT: 06

4.2 Get Firmware Version

Sending Get Firmware Version command returns the firmware version back to the application.

Command
<STX><R><FmVerID><ETX><LRC 1>

Response
<ACK> <STX><Version String><ETX><LRC 2>

Version String will be in format of " Swipe Reader x.y.z" x.y.z
is the major and minor version number.

4.3 Get Setting

This command will send current setting to application.

Command
<STX> <R> <ReviewID> <ETX> <LRC 1>

Response
<ACK> <STX> <FuncID> <Len> <FuncData> <ETX> <LRC 2>

<FuncID>, <Len> and <FuncData> definition are same as described above.

Example:
Review all setting
CMD: \02\52\1F\03\4C
OUT: \06\02\7E\01\31\4C\01\31\58\01\31\03\5B

4.4 Function ID Table

The following table shows the available Function IDs with the default setting shown in **bold**.

Function Name	Function ID	Description
EncryptionID	0x4C	Security Algorithm **'0' Clear Text** '1' Triple DES '2' AES
keyManagementID	0x58	**'0' DUKPT** '1' Fixed Key
SecurityLevelID	0x7E	Security Level (Read Only) '0' ~ '3" **Default value '1'**
GetFirmwareVersion	0x22	returns current firmware version

4.4.1 EncryptionID

Set clear text:
CMD: 02 53 4C 01 30 03 2F
OUT: 06
Read EncryptionID:
CMD: 02 52 4C 03 1F
OUT: 06 02 4C 01 30 03 7C

Set Triple DES:
CMD: 02 53 4C 01 31 03 2E
OUT: 06
Read EncryptionID:
CMD: 02 52 4C 03 1F
OUT: 06 02 4C 01 31 03 7D

Set AES
CMD: 02 53 4C 01 32 03 2D

OUT: 06
Read EncryptionID:
CMD: 02 52 4C 03 1F
OUT: 06 02 4C 01 32 03 7E

4.4.2 keyManagementID

set DUKPT:
CMD: 02 53 58 01 31 03 3A
OUT: 06
Read keyManagementID:
CMD: 02 52 58 03 0B
OUT: 06 02 58 01 31 03 69

Set Fixed Key:
CMD: 02 53 58 01 30 03 3B
OUT: 06
Read keyManagementID:
CMD: 02 52 58 03 0B
OUT: 06 02 58 01 30 03 68

4.4.3 Read SecurityLevelID

CMD: 02 52 7E 03 2D
OUT: 06 02 7E 01 33 03 4D

4.4.4 Get Firmware Version

CMD: 02 52 22 03 71
OUT: 06 02 49 44 20 54 45 43 48 20 69 4D 61 67 00 31 31 30 03 04
Firmware Version: ID TECH iMag110

5 Data Output Format

5.1 iMag Unencrypted Data Output Format

Track 1: <Start Sentinel 1><T_1 Data><End Sentinel><Track Separator>
Track 2: <Start Sentinel 2><T_2 Data><End Sentinel><Track Separator>
Track 3: <Start Sentinel 3><T_3 Data><End Sentinel><Terminator>

where: Start Sentinel 1 = %

Start Sentinel 2 = ;
Start Sentinel 3 = ; for ISO, % for AAMVA
End Sentinel all tracks = ?

Start or End Sentinel: Characters in encoding format which come before the first data character (start) and after the last data character (end), indicating the beginning and end, respectively, of data.

Track Separator: A designated character which separates data tracks. The default character is CR (Carriage Return).

Terminator: A designated character which comes at the end of the last track of data, to separate card reads. The default character is CR (Carriage Return).

For example:

```
%B4352378366824999^TFSTEST /THIRTYONE
^05102011000088200882000000?;4352378366824999=051020110000882?
```

5.2 iMag Encrypted Data Output Format

iMag uses ID TECH enhanced data encryption format. In this format, all tracks of the data are encrypted.

Encryption format.

1. Secured output structure setting:

   53 85 01 encryptStructure
   encryptStructure = '0' // Default: encrypt output structure that will match old version
   encryptStructure = '1' // enhanced encrypt output structure will send byte 8 and 9 and CardType will be 1xxxxxxx
2. Encrypt Option Setting: // only effect in enhanced structure
   53 84 01 encrypOpt // default 0x08
   encryptOpt:
   bit0: 1 – tk1 force encrypt *
   bit1: 1 – tk2 force encrypt *
   bit2: 1 – tk3 force encrypt *
   bit3: 1 – tk3 force encrypt when card type is 0

   Note:
   1) When force encrypt is set, this track will always be encrypt, regardless of card type. No clear/mask text will be sent.
   2) If and only if in enhanced encrypt structure, each track encryption is separated,

encrypted data length will round up to 8 or 16 bytes.
3) When force encrypt is not set, it encrypts data just like old structure, that is, only T1 and T2 in type zero will be encrypted.

3. hash Option Setting:
   53 5C 01 hashOpt // default 0x37
   hashOpt: ('0' – '7')
   bit0: 1 – tk1 hash will be sent if data is encrypted
   bit1: 1 – tk2 hash will be sent if data is encrypted
   bit2: 1 – tk3 hash will be sent if data is encrypted

4. Mask Option Setting: // only effected in enhanced structure
   53 86 01 maskOpt // Default: 0x07
   maskOpt:
   bit0: 1 – tk1 mask data allow to send when encrypted
   bit1: 1 – tk2 mask data allow to send when encrypted
   bit2: 1 – tk3 mask data allow to send when encrypted

   Note:
   1) When mask option bit is set – if data is encrypted (but not forced encrypted), the mask data will be sent; If mask option is not set, the mask data will not be sent under the same condition.

Settings for OPOS:

1. Assume reader is under default setting (Encrypt Structure 0)
2. Set to new Encrypt Structure 1:
   53 85 01 31

The OPOS driver/application may also send following command when change (Decode/Raw format)
(Set raw or decode data format)
53 1D 01 30 // RAW data format
53 1D 01 31 // Decoded format

Following is the output structure:

0	STX
1	Data Length low byte
2	Data Length high byte
3	Card Encode Type*
4	Track 1-3 Status
5	T1 data length
6	T2 data length
7	T3 data length

8	Clear/mask data sent status *
9	Encrypted/Hash data sent status *
10	T1 clear/mask data T2 clear/mask data T3 clear/mask data T1 encrypted data T2 encrypted data T3 encrypted data Session ID (8 bytes) (Security level 4 only) T1 hashed (20 bytes each) (if encrypted and hash tk1 allowed) T2 hashed (20 bytes each) (if encrypted and hash tk2 allowed) T3 hashed (20 bytes each) (if encrypted and hash tk3 allowed) KSN (10 bytes) CheckLrc CheckSum ETX

Note:
1) Field 8 (Clear/mask data sent status) and field 9 (Encrypted/Hash data sent status) will only be sent in new encrypt structure.
2) Field 9: Clear/mask data sent status byte:
bit 0: 1--- if TK1 clear/mask data present
bit 1: 1--- if TK2 clear/mask data present
bit 2: 1--- if TK3 clear/mask data present
Bit 3:0—0 reserved future use
Bit 4:0—0 "
Bit 5:0—0 "

3) Field 9: Encrypted data sent status
bit 0: if 1—tk1 encrypted data present
bit 1: if 1—tk2 encrypted data present
bit 2: if 1—tk3 encrypted data present
bit 3: if 1—tk1 hash data present
bit 4: if 1—tk2 hash data present
bit 5: if 1—tk3 hash data present
Bit 6: if 1—session ID present
Bit 7: if 1—KSN present

Card Type:

Value	Encode Type	Description
0 / 80		ISO/ABA format
1 / 81		AAMVA format
3 / 83		Other

4 / 84 Raw; un-decoded format.

Note:

1) Card Type will be 8x in new structure and 0x for old structure

*2) Type 4 or 84: Raw data format; a*ll tracks are encrypted and no mask data is sent. No track indicator '01', '02' or '03' in front of each track. ('01','02' and '03' will still exist for none secured mode raw output when security level < 3)

General concept for each track:

1. If encrypted, no clear data will be sent
2. Clear data always sent if no encrypted data
3. If not encrypted, hash will never be send

5.3 Level 4 Activate Authentication Sequence

The security level changes from 3 to 4 when the device enters authentication mode successfully. Once the security level is changed to level 3 or 4, it cannot go back to a lower level.

Activate Authentication Mode Command

When the reader is in security level 4, it would only transmit the card data when it is in Authenticated Mode.

Authentication Mode Request

When sending the authentication request, the user also needs to specify a time limit for the reader to wait for the activation challenge reply command. The minimum timeout duration required is 120 seconds. If the specified time is less than the minimum, 120 seconds would be used for timeout duration. The maximum time allowed is 3600 seconds (one hour). If the reader times out while waiting for the activation challenge reply, the authentication failed.

Device Response

When authentication mode is requested, the device responds with two challenges: Challenge 1 and challenge 2. The challenges are encrypted using the current DUKPT key exclusive- or'ed with <F0F0 F0F0 F0F0 F0F0 F0F0 F0F0 F0F0 F0F0>.

The decrypted challenge 1 contains 6 bytes of random number followed by the last two bytes of KSN. The two bytes of KSN may be compared with the last two bytes of the clear text KSN sent in the message to authenticate the reader. The user should complete the Activate Authentication sequence using Activation Challenge Reply command.

Command Structure

Host -> Device:

<STX><R><80h><02h><Pre-Authentication Time Limit><ETX><LRC>

Device -> Host:
<ACK><STX><Device Response Data><ETX><LRC> (success)
<NAK> (fail)

Pre-Authentication Time Limit: 2 bytes of time in seconds
Device Response Data: 26 bytes data, consists of <Current Key Serial Number> <Challenge 1> <Challenge 2>

Current Key Serial Number: 10 bytes data with Initial Key Serial Number in the leftmost 59 bits and Encryption Counter in the rightmost 21 bits.
Challenge 1: 8 bytes challenge used to activate authentication. Encrypted using the key derived from the current DUKPT key.
Challenge 2: 8 bytes challenge used to deactivate authentication. Encrypted using the key derived from the current DUKPT key.

Activation Challenge Reply Command

This command serves as the second part of an Activate Authentication sequence. The host sends the first 6 bytes of Challenge 1 from the response of Activate Authenticated Mode command, two bytes of Authenticated mode timeout duration, and eight bytes Session ID encrypted with the result of current DUKPT Key exclusive- or'ed with <3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C>.

The Authenticated mode timeout duration specifies the maximum time in seconds which the reader would remain in Authenticated Mode. A value of zero forces the reader to stay in Authenticated Mode until a card swipe or power down occurs. The minimum timeout duration required is 120 seconds. If the specified time is less than the minimum, 120 seconds would be used for timeout duration. The maximum time allowed is 3600 seconds (one hour).

If Session ID information is included and the command is successful, the Session ID will be changed.

The Activate Authenticated Mode succeeds if the device decrypts Challenge Reply response correctly. If the device cannot decrypt Challenge Reply command, Activate Authenticated Mode fails and DUKPT KSN advances.

Command Structure
Host -> Device:
<STX><S><82h><08h><Activation Data><ETX><LRC>

Device -> Host:

<ACK> (success)
<NAK> (fail)

Activation Data: 8 or 16 bytes, structured as <Challenge 1 Response> <Session ID>

Challenge 1 Response: 6 bytes of Challenge 1 random data with 2 bytes of Authenticated mode timeout duration. It's encrypted using the key derived from the current DUKPT key.
Session ID: Optional 8 bytes Session ID, encrypted using the key derived from the current DUKPT key.

Deactivate Authenticated Mode Command

This command is used to exit Authenticated Mode. Host needs to send the first 7 bytes of Challenge 2 (from the response of Activate Authenticated Mode command) and the Increment Flag (0x00 indicates no increment, 0x01 indicates increment of the KSN) encrypted with current DUKPT Key exclusive- or'ed with <3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C 3C3C>.

If device decrypts Challenge 2 successfully, the device will exit Authenticated Mode. The KSN will increase if the Increment flag is set to 0x01. If device cannot decrypt Challenge 2 successfully, it will stay in Authenticated Mode until timeout occurs or when customer swipes a card.

The KSN is incremented every time the authenticated mode is exited by timeout or card swipe action. When the authenticated mode is exited by Deactivate Authenticated Mode command, the KSN will increment when the increment flag is set to 0x01.

Command Structure
Host -> Device:
<STX><S><83h><08h><Deactivation Data><ETX><LRC>

Device -> Host:
<ACK> (success)
<NAK> (fail)

<Deactivation data>: 8-bytes response to Challenge 2. It contains 7 bytes of Challenge 2 with 1 byte of Increment Flag, encrypted by the specified variant of current DUKPT Key

Get Reader Status Command

Command Structure
Host -> Device:

<STX><R><83h><ETX><LRC>

Device -> Host:
<ACK><STX><83h><02h><Current Reader Status><Pre-conditon><ETX><LRC> (success)
<NAK> (fail)

Current Reader Status: 2-bytes data with one byte of <Reader State> and one byte of <Pre-Condition>

Reader State: indicates the current state of the reader

0x00: The reader is waiting for Activate Authentication Mode Command. The command must be sent before the card can be read.

0x01: The authentication request has been sent, the reader is waiting for the Activation Challenge Reply Command.

0x02: The reader is waiting for a card swipe.

Pre-condition: specifies how the reader goes to its current state as follows

0x00: The reader has no card swipes and has not been authenticated since it was powered up.

0x01: Authentication Mode was activated successfully. The reader processed a valid Activation Challenge Reply command.

0x02: The reader receives a good card swipe.

0x03: The reader receives a bad card swipe or the card is invalid.

0x04: Authentication Activation Failed.

0x05: Authentication Deactivation Failed.

0x06: Authentication Activation Timed Out. The Host fails to send an Activation Challenge Reply command within the time specified in the Activate Authentication Mode command.

0x07: Swipe Timed Out. The user fails to swipe a card within the time specified in the Activation Challenge Reply command

5.4 Decryption Example

Key for all examples is
0123456789ABCDEFFEDCBA9876543210

5.4.1 Security Level 3 Decryption

Example of decryption of a three track ABA card

Enhanced encryption Format (this can be recognized because the high bit of the fourth byte underlined (80) is 1.

029801<u>80</u>3F48236B03BF252A343236362A2A2A2A2A2A2A393939395E42555348
204A522F47454F52474520572E4D525E2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2
A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A3F2A3B343236362A2A2A2A2A2A2A2
A393939393D2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A3F2ADA7F2A52BD3F6DD
8B96C50FC39C7E6AF22F06ED1F033BE0FB23D6BD33DC5A1F808512F7AE18D47
A60CC3F4559B1B093563BE7E07459072ABF8FAAB5338C6CC8815FF87797AE3A7
BEAB3B10A3FBC230FBFB941FAC9E82649981AE79F2632156E775A06AEDAFAF6
F0A184318C5209E55AD44A9CCF6A78AC240F791B63284E15B4019102BA6C50581
4B585816CA3C2D2F42A99B1B9773EF1B116E005B7CD8681860D174E6AD316A0E
CDBC687115FC89360AEE7E430140A7B791589CCAADB6D6872B78433C3A25DA9
DDAE83F12FEFAB530CE405B701131D2FBAAD970248A456000933418AC88F65E1
DB7ED4D10973F99DFC8463FF6DF113B6226C4898A9D355057ECAF11A5598F02C
A31688861C157C1CE2E0F72CE0F3BB598A614EAABB162994901190000000000206E
203

STX, Length(LSB, MSB), card type, track status, length track 1, length track 2, length track 3
02 9801 80 3F 48-23-6B 03BF

The above broken down and interpreted
02—STX character
98—low byte of total length
01—high byte of total length
80—card type byte (interpretation new format ABA card)
3F—3 tracks of data all good
48—length of track 1
23—length of track 2
6B—length of track 3
03—tracks 1 and 2 have masked/clear data
BF—bit 7=1—KSN included
Bit 6=0—no Session ID included so not level 4 encryption
Bit 5=1—track 3 hash data present
Bit 4=1—track 2 hash data present

Bit 3-1—track 1 hash data present
Bit 2=1—track 3 encrypted data present
Bit 1=1—track 2 encrypted data present
Bit 0=1—track 1 encrypted data present

Track 1 data masked (length 0x48)
252A343236362A2A2A2A2A2A2A393939395E42555348204A522F47454F5247452
0572E4D525E2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2
A2A2A2A2A2A2A2A3F2A

Track 1 masked data in ASCII
%*4266********9999^BUSH JR/GEORGE
W.MR^********************************?*

Track 2 data in hex masked (length 0x23)
3B343236362A2A2A2A2A2A2A2A393939393D2A2A2A2A2A2A2A2A2A2A2A2A2A2
A2A2A3F2A

Track2 masked data in ASCII
;4266********9999=****************?*

In this example there is no Track 3 data eith er clear or masked (encrypted and hashed data is below)

Track 1 encrypted length 0x48 rounded up to 8 bytes = 0x48 (72 decimal)
DA7F2A52BD3F6DD8B96C50FC39C7E6AF22F06ED1F033BE0FB23D6BD33DC5A1
F8
08512F7AE18D47A60CC3F4559B1B093563BE7E07459072ABF8FAAB5338C6CC88
15FF87797AE3A7BE

Track 2 encrypted length 0x32 rounded up to 8 bytes =0x38 (56 decimal)
AB3B10A3FBC230FBFB941FAC9E82649981AE79F2632156E775A06AEDAFAF6F0
A
184318C5209E55AD

Track 3 encrypted length 0x6B rounded up to 8 bytes =0x70 (64 decimal)
44A9CCF6A78AC240F791B63284E15B4019102BA6C505814B585816CA3C2D2F42
A99B1B9773EF1B116E005B7CD8681860D174E6AD316A0ECDBC687115FC89360A
EE7E430140A7B791589CCAADB6D6872B78433C3A25DA9DDAE83F12FEFAB530
CE
405B701131D2FBAAD970248A45600093

Track 1 data hashed length 20 bytes
3418AC88F65E1DB7ED4D10973F99DFC8463FF6DF

Track 2 data hashed length 20 bytes
113B6226C4898A9D355057ECAF11A5598F02CA31

Track 3 data hashed length 20 bytes
688861C157C1CE2E0F72CE0F3BB598A614EAABB1

KSN length 10 bytes
62994901190000000002

LCR, check sum and ETX
06E203

Clear/Masked Data in ASCII:
Track 1: %*4266********9999^BUSH JR/GEORGE
W.MR^**********************************?*
Track 2: ;4266********9999=****************?*

Key Value: 1A 99 4C 3E 09 D9 AC EF 3E A9 BD 43 81 EF A3 34
KSN: 62 99 49 01 19 00 00 00 00 02

Decrypted Data:
Track 1 decrypted
%B4266841088889999^BUSH JR/GEORGE
W.MR^080910110000110000000046000000?!
Track 2 decrypted
;4266841088889999=08091011000046?0
Track 3 decrypted
;3333333337676760707077676763333333333767676070707767676333333333376767
607070776767633333333337676760707?2

Track 1 decrypted data in hex including padding zeros (but there are no pad bytes here)
25423432363638343130383838393939395E42555348204A522F47454F52474520572
E4D525E303830393130313130303030313130303030303030303034363030303030303F
21

Track 2 decrypted data in hex including padding zeros
3B34323636383431303838383839393939D303830393130313130303030303034363F300
000000000

Track 3 decrypted data in hex including padding zeros
3B33333333333333333333373637363736303730373037373637363736336333333333333
33333333736373637363037303730373736373637363373633333333333333333333373637363

736303730373037373637363736333333333333333333337363736373630373037 3F32
0000000000

5.4.2 Security Level 4 Decryption

02A001803F48236B03FF252A343236362A2A2A2A2A2A2A2A393939395E42555348
204A522F47454F52474520572E4D525E2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2
A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A3F2A3B343236362A2A2A2A2A2A2A2
A393939393D2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A2A3F2A6D7D5B204D3579694
E148F3FB2565544D35825EA89BA30C966D34363151BF592F995EDA86B94A47EBF
DF6434CB3A075DDD18F616E21F1E2038BC3AD5F96C1387177BD89409DA2E92A
684543E007087F8694AEA8D3DB36BA10BC4D4B2771C622FEC8271A6E021AA564
4ED559EC09CABF19F36B422CA2016B48A7241B2DA9584ED4415B4F30637734CF
5031AF475DAF27C188A1A771264011BAA090E91893BC2A52EDD56F8E6E9554BC
0C5207C04E3C21B6DA2A48F2257DC6946DBFBC87F3189E5C8B954BF7303D01E4
43155911E4137AEAD52441567AA1D50924A7597EC9D758AB4F3A8E82BF81A2E3
418AC88F65E1DB7ED4D10973F99DFC8463FF6DF113B6226C4898A9D355057ECA
F11A5598F02CA31688861C157C1CE2E0F72CE0F3BB598A614EAABB16299490119
0000000003D67C03

Clear/Masked Data:
Track 1: %*4266********9999^BUSH JR/GEORGE
W.MR^********************************?*
Track 2: ;4266********9999=****************?*

Key Value: 89 52 50 33 61 75 51 5C 41 20 CF 45 F4 1A BF 1C
KSN: 62 99 49 01 19 00 00 00 00 03
Session ID: AA AA AA AA AA AA AA AA

Decrypted Data in ASCII:
%B4266841088889999^BUSH JR/GEORGE
W.MR^080910110000110000000046000000?!
;4266841088889999=08091011000046?0
;3333333337676760707077676763333333333767676070707767676333333333376767
6070707767676333333333376767607077
6070707767676333333333376767607077?2

Decrypted Data in Hex:
2542343236363834313038383838393939395E42555348204A522F47454F52474520572
E4D525E303830393130313130303030313130303030303030303034363030303030303F
21
3B343236363834313038383838393939393D303830393130313130303030303034363F300
000000000
3B333333333333333333333736373637363037303730373736373637363733333333333333
333333373637363736303730373037373637363736333333333333333333333373637363

736303730373037373637363736333333333333333333333373637363736303730373F32
0000000000