NFC access control for the Mifare S50 1K

Written by Amal Graafstra for Dangerous Things

KEYS AND ACCESS BITS

The Mifare S50 1KB tag memory is organized into sectors comprised of 4 blocks each. In each sector, data is stored in the first 3 blocks but the last block is known as the **sector trailer** which holds two authentication keys (key A and B) and a set of access bits. These access bits define read/write access permissions for the first three blocks in the sector, as well as the sector trailer itself. In other words, the access bits can be used to protect themselves, allowing data, keys, and access bits to all become locked and unalterable.

The format of the 16 byte sector trailer block is as follows; Key A is stored in bytes 0-5, access bits are stored in bytes 6-8, byte 9 is a general purpose byte, and key B is stored in bytes 10-15. For official documentation, reference section 6.7.1 of MF1ICS50.

There are 3 access bits per block, defined as C1, C2, and C3. Since there are 4 memory blocks per sector, there are 4 individual C1 bits, 4 individual C2 bits, etc. that are defined as $C1_0$ for block 0, $C1_1$ for block 1, etc. Access rules for block 0 are defined by $C1_0\,C2_0\,C3_0$, block 1 by $C1_1\,C2_1\,C3_1$, block 2 by $C1_2\,C2_2\,C3_2$ and block 3 (the sector trailer) by $C1_3\,C2_3\,C3_3$.

Table 1 – Data block access bit rules (blocks 0-2)

Access bits			Access condition				Application
C1	*C2*	*C3*	*Read*	*Write*	*Increment*	*Dec/Trans/Rest*	
0	0	0	key A|B	key A|B	key A|B	key A|B	transport
0	1	0	key A|B	never	never	never	read/write block
1	0	0	key A|B	key B	never	never	read/write block
1	1	0	key A|B	key B	key B	key A|B	value block
0	0	1	key A|B	never	never	key A|B	value block
0	1	1	key B	key B	never	never	read/write block
1	0	1	key B	never	never	never	read/write block
1	1	1	never	never	never	never	read/write block

Table 2 – Sector trailer access bit rules (block 3)

Access bits			*Access condition for*						*Remarks*
			KEY A		*Access bits*		*Key B*		
C1	*C2*	*C3*	*Read*	*Write*	*Read*	*Write*	*Read*	*Write*	
0	0	0	never	key A	key A	never	key A	key A	Key A is able to read key B
0	1	0	never	never	key A	never	key A	never	Key A is able to read key B
1	0	0	never	key B	key A|B	never	never	key B	
1	1	0	never	never	key A|B	never	never	never	
0	0	1	never	key A	key A	key A	key A	key A	Key A is able to read key B
0	1	1	never	key B	key A|B	key B	never	key B	
1	0	1	never	never	key A|B	key B	never	never	
1	1	1	never	never	key A|B	never	never	never	

* If key B may be read in the corresponding Sector Trailer it cannot serve for authentication (all grey marked lines in previous table). Consequences: If the reader tries to authenticate any block of a sector with key B using grey marked access conditions, the card will refuse any subsequent memory access after authentication.

INVERTED ACCESS BITS

To further complicate things, there are also 3 "***negated***" bits per block that are stored as the opposite value of the "**normal**" bits. So, if **$C1_0$** is **1**, then ***$C1_0$*** must be ***0***. The tag will check to ensure these negated values all check out, and if they don't then the tag will assume the tag's memory is corrupt or there has been some sort of tampering attempt and completely block all access to the entire sector. It is essential that the access bits are properly written to the tag or sectors with invalid sector trailers will be unreadable.

UNFORMATTED S50

Unformatted tags from the manufacturer typically have keys A and B set to a default value of FF FF FF FF FF FF for every sector, and all sector access bits set to FF 07 80 which means blocks 0-2 are able to be read and written using either key A or B, and key A can read or write to the entire* sector trailer.

```
Sector 0 (0x00)
[00]  AD 65 F8 8F BF 88 04 00 |.e......|
 r--  48 85 14 58 45 10 36 10 |H..XE.6.|
[01]  00 00 00 00 00 00 00 00 |........|
 rwi  00 00 00 00 00 00 00 00 |........|
[02]  00 00 00 00 00 00 00 00 |........|
 rwi  00 00 00 00 00 00 00 00 |........|
[03]  FF:FF:FF:FF:FF:FF  Factory default key
 wxx  FF:07:80 69
 (r)  FF:FF:FF:FF:FF:FF  Factory default key
```

```
Sector 1 (0x01)
[04]  00 00 00 00 00 00 00 00 |........|
 rwi  00 00 00 00 00 00 00 00 |........|
[05]  00 00 00 00 00 00 00 00 |........|
 rwi  00 00 00 00 00 00 00 00 |........|
[06]  00 00 00 00 00 00 00 00 |........|
 rwi  00 00 00 00 00 00 00 00 |........|
[07]  FF:FF:FF:FF:FF:FF  Factory default key
 wxx  FF:07:80 69
 (r)  FF:FF:FF:FF:FF:FF  Factory default key
```

FF:07:80	Byte 6	$C2_3$	$C2_2$	$C2_1$	$C2_0$	$C1_3$	$C1_2$	$C1_1$	$C1_0$
		1	1	1	1	1	1	1	1
	Byte 7	$C1_3$	$C1_2$	$C1_1$	$C1_0$	$C3_3$	$C3_2$	$C3_1$	$C3_0$
		0	0	0	0	0	1	1	1
	Byte 8	$C3_3$	$C3_2$	$C3_1$	$C3_0$	$C2_3$	$C2_2$	$C2_1$	$C2_0$
		1	0	0	0	0	0	0	0
	Blk 0	000	Blk 1	000	Blk 2	000	Blk 3	001	

** As illustrated in table 2, key A can never read itself. It is assumed key A is already known if the reader has authenticated with it. This approach saves bit space and makes it possible to define 8 different access configurations that cover the most useful options for 4 different memory blocks – all with only 3 bytes of data.*

NFC FORMATTED S50

Upon formatting the S50 for NFC, a typical NFC application will do the following;

1) Create a MAD in sector 0 that creates NFC AID records to identify all subsequent sectors as storing NFC NDEF data. The sector trailer is also typically updated with a MAD access key value for key A as defined by the NFC spec, a secret value for Key B, and access bits are set to restrict the MAD so it can only be read with Key A and only be updated with key B.

2) For subsequent sectors defined in the MAD by NFC AID records, key A is typically set to a public NDEF key as defined by the NFC spec, while access bits are typically set to allow key A to read/write blocks 0-2 and only read the sector trailer. Access bits are also set to allow key B to update key A, update access bits, update itself, and read/write blocks 0-2.

```
Sector 0 (0x00)
[00]  AD 65 F8 8F BF 88 04 00 |.e......|
 r--  48 85 14 58 45 10 36 10 |H..XE.6.|
[01]  14 01 03 E1 03 E1 03 E1 |........|
 rW-  03 E1 03 E1 03 E1 03 E1 |........|
[02]  03 E1 03 E1 03 E1 03 E1 |........|
 rW-  03 E1 03 E1 03 E1 03 E1 |........|
[03]  A0:A1:A2:A3:A4:A5  MAD access key
 WXW  78:77:88 C1
      XX:XX:XX:XX:XX:XX  (unknown key)
```

```
Sector 1 (0x01)
[04]  00 00 03 0B D1 01 07 54 |.......T|
 rwi  02 65 6E 61 6D 61 6C FE |.enamal.|
[05]  00 00 00 00 00 00 00 00 |........|
 rwi  00 00 00 00 00 00 00 00 |........|
[06]  00 00 00 00 00 00 00 00 |........|
 rwi  00 00 00 00 00 00 00 00 |........|
[07]  D3:F7:D3:F7:D3:F7  Public NDEF key
 WXW  7F:07:88 40
      XX:XX:XX:XX:XX:XX  (unknown key)
```

7F:07:88	Byte 6	$C2_3$	$C2_2$	$C2_1$	$C2_0$	$C1_3$	$C1_2$	$C1_1$	$C1_0$
		0	1	1	1	1	1	1	1
	Byte 7	$C1_3$	$C1_2$	$C1_1$	$C1_0$	$C3_3$	$C3_2$	$C3_1$	$C3_0$
		0	0	0	0	0	1	1	1
	Byte 8	$C3_3$	$C3_2$	$C3_1$	$C3_0$	$C2_3$	$C2_2$	$C2_1$	$C2_0$
		1	0	0	0	1	0	0	0
	Blk 0	000	Blk 1	000	Blk 2	000	Blk 3	011	

LOCKED S50

Upon “locking”, there are multiple levels of severity when setting a locked state. **1)** An application could set access bits for all data blocks as read-only for key A, but not lock the sector trailer block so key A could easily change the read-only access bits on the data blocks back to read/write. **2)** A typical NFC application will set access bits for every sector identified in the MAD by NFC AID records so key A can only read but not write data to each block in the sector, but key B can still write data to every block in the sector. This means if you have key B you can still update both the data block and the access permissions for the access bits themselves and open the tag back up for reading and writing with key A. **3)** A true “lock” would mean setting access bits so both key A and key B could not write to any blocks in the sector, but most NFC applications will not go that far.

This is an example of level 2 locking (as defined above):

```
Sector 0 (0x00)
[000]  AD 65 F8 8F BF 88 04 00 |.e......|
 r--  48 85 14 58 45 10 36 10 |H..XE.6.|
[001]  14 01 03 E1 03 E1 03 E1 |........|
 rW-  03 E1 03 E1 03 E1 03 E1 |........|
[002]  03 E1 03 E1 03 E1 03 E1 |........|
 rW-  03 E1 03 E1 03 E1 03 E1 |........|
[003]  A0:A1:A2:A3:A4:A5  MAD access key
 WXW  78:77:88 C1
      XX:XX:XX:XX:XX:XX  (unknown key)
```

```
Sector 1 (0x01)
[004]  00 00 03 0B D1 01 07 54 |.......T|
 rW-  02 65 6E 61 6D 61 6C FE |.enamal.|
[005]  00 00 00 00 00 00 00 00 |........|
 rW-  00 00 00 00 00 00 00 00 |........|
[006]  00 00 00 00 00 00 00 00 |........|
 rW-  00 00 00 00 00 00 00 00 |........|
[007]  D3:F7:D3:F7:D3:F7  Public NDEF key
 WXW  78:77:88 43
      XX:XX:XX:XX:XX:XX  (unknown key)
```

78:77:88	Byte 6	$C2_3$	$C2_2$	$C2_1$	$C2_0$	$C1_3$	$C1_2$	$C1_1$	$C1_0$
		0	1	1	1	1	0	0	0
	Byte 7	$C1_3$	$C1_2$	$C1_1$	$C1_0$	$C3_3$	$C3_2$	$C3_1$	$C3_0$
		0	1	1	1	0	1	1	1
	Byte 8	$C3_3$	$C3_2$	$C3_1$	$C3_0$	$C2_3$	$C2_2$	$C2_1$	$C2_0$
		1	0	0	0	1	0	0	0
	Blk 0	100	Blk 1	100	Blk 2	100	Blk 3	011	

ABOUT THE AUTHOR

Amal Graafstra is a double RFID implantee and RFID/NFC enthusiast. His endeavors in the field include writing the book RFID Toys and helping fellow hobbyists through the forum at www.rfidtoys.net/forum Questions regarding the S50 and/or how to secure it can be posted to the Q&A forum at www.dangerousthings.com