Poking the S in SD cards

Nicolas Oberli
Introduction

• SD stands for Secure Digital
  – What is the Secure for?
• No physical attacks on the cards
What is an SD card?

- Basically a microcontroller interfacing the SD interface with flash memory
- See bunnie and xobs talk @ 30C3 for details

https://en.wikipedia.org/wiki/SD_card
Communication

- SD cards support 3 communication protocols
  - SPI Bus protocol
    - Classic SPI
  - SD / UHS-I Bus protocol
    - CLK, CMD, Up to 4 data lines
  - UHS-II Bus protocol
    - RCLK, 2 differential data lines

https://en.wikipedia.org/wiki/SD_card
Time to dig into the specs

• Specs are freely available in a simplified format on the SD association website
  – 262-pages document (general specs – part 1)
  – Presents the general description of the SD System
Initialization sequence

SD specs part 1, figure 7-2
Protocol

- Query/reply-based
- Each command has a number and is usually referenced with it
  - eg. CMD0 - GO_IDLE_STATE
Protocol – cont.

- 7 different response formats
  - Depends on the command
- Protocol implements a block transfer feature
  - Used to transfer more than 4 bytes
  - Block starts with 0xFE
  - Length is defined by CMD16 (512 bytes by default)
Security features

• SDMI – Secure Digital Music Initiative
  – Detailed under specs part 3
    • Available only to SD members / NDA
  – Not covered here

• Can be read- and/or write-protected
  – Available through CMD27 and CMD42
  – These commands are mandatory to get SD label
Interfacing with SD card

• First need to communicate correctly with the card

• Implemented a Python script to drive the SPI interface and access the SD card using raw commands

• CRC is optional in SPI mode, easier to play with
Setup
CMD42 – LOCK_UNLOCK

• Used to control the password protection mechanism
  - Up to 16 bytes
  - Not limited to printable characters
  - Keyspace: $2^{128}$ – Same as an AES key
    • Brute force is unachievable
Locking the SD card

- The **CMD42** command controls the password locking functions
  - Takes no parameter, but card expects a following data block
CMD42 data block

- Contains the command options, length and the actual password
Unlocking SD card

• Send `CMD42`
• Send a data block, unseting the `LOCK` bit, setting the password length and the password
• Card will assert the MISO line, then send an ACK once the command has been processed
• Lock status is available in the status bits (CMD13)
Unlocking SD card

- Send \texttt{CMD42}
- Send a data block, unsetting the \texttt{LOCK} bit, setting the password length and the password
- \textbf{Card will assert the MISO line, then} send an ACK once the command has been processed
- Lock status is available in the status bits (CMD13)
Guess what happens?

CMD42
Password block
Response

00000

000000
What’s happening?

- SD controller checks for the length of the password, then compares each byte to the correct password.
- Returns an error as soon as there’s a discrepancy.
- Possible to determine a correct byte by measuring processing time.
Measuring time using SPI

- During processing time, read dummy bytes as fast as possible
- As long as we read zeroes, the password check is still ongoing
- Once we read a 1, count the number of zeroes
In practice:

```
sd > test_len
  00 : 122
  01 : 124
  02 : 124
  03 : 124
  04 : 124
  05 : 124
  06 : 130
  07 : 124
  08 : 124
  09 : 124
  10 : 124
  11 : 124
  12 : 124
  13 : 124
  14 : 124
  15 : 124
  16 : 124
Length: 6
Sd >
```
So?

- Bought a bunch of SD cards (~20)
  - Different vendors
  - Different sizes
- Also asked colleagues / friends for SD cards
  - The only card I permanently locked was not mine (‘-’*)
- Locked them with “123456” as password
Special cases – Sony SD

• Card refuses to check for the password after three failed attempts

• Need to remove and insert the card again to get 3 more attempts
  – In fact, doing a reset sequence (CMD0) is enough to get 3 more tries
  – Slightly makes the bruteforce slower
Special cases – Sony uSD

• Card seems to have a really fast checking time
  – Can get no or maybe one zero bit

• Our sampling rate might be too slow
  – SPI interface is ~42MHz
  – Using logic analyzer (100MSPS) still does not show any usable results
Faster !!

• Used lab oscilloscope
  – Up to 40GSPS, more than enough
• Had to setup a trigger for correct measurement
And...
And...
Special cases - Kingston

• It is possible to count the password length, but not the password chars

• Took a lot of measurements until I found this:
100000
120000
123450
Still vulnerable

- Password checking works on groups of 4 bytes
- If remaining bytes to check is $\geq 4$, test each byte individually
- Attack takes more time, but works anyways

¯\_(ツ)_/¯
## Results

<table>
<thead>
<tr>
<th>Card</th>
<th>Manufacturer</th>
<th>Prod. date*</th>
<th>Vulnerable ?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transcend uSD 4GB</td>
<td>Transcend (0x74)</td>
<td>09/2011</td>
<td>Yes</td>
</tr>
<tr>
<td>Transcend uSD 16GB</td>
<td>Transcend (0x74)</td>
<td>10/2012</td>
<td>Yes</td>
</tr>
<tr>
<td>Hama 8GB</td>
<td>Phison (0x27)</td>
<td>06/2010</td>
<td>Yes</td>
</tr>
<tr>
<td>Maxell 32GB</td>
<td>Phison (0x27)</td>
<td>10/2011</td>
<td>Yes</td>
</tr>
<tr>
<td>Sony uSD 32GB</td>
<td>Sony (0x9c)</td>
<td>07/2012</td>
<td>Yes</td>
</tr>
<tr>
<td>Sony 32GB</td>
<td>Sony (0x9c)</td>
<td>12/2011</td>
<td>Yes</td>
</tr>
<tr>
<td>Kingston uSD 32GB</td>
<td>Unknown (0x9f)</td>
<td>10/2012</td>
<td>Yes</td>
</tr>
<tr>
<td>Sandisk Extreme 128GB</td>
<td>Sandisk (0x03)</td>
<td>03/2012</td>
<td>No</td>
</tr>
<tr>
<td>Sandisk mobile ultra 16GB</td>
<td>Sandisk (0x03)</td>
<td>12/2009</td>
<td>No</td>
</tr>
<tr>
<td>Samsung Evo+ uSD 32GB</td>
<td>Samsung (0x1b)</td>
<td>10/2012</td>
<td>Unsupported</td>
</tr>
</tbody>
</table>

* Production date format is not consistent
Ouch

- Sandisk only controller I tested not vulnerable to this attack
- Remember: SD vendor != Controller manufacturer
- Samsung cards respond with invalid command using CMD42
Conclusions

• Useless vulnerability?
  - Feature not supported by any OS
• Affects a lot of manufacturers
• Reading specs is fun
Future work

- COP protection
  - Added in specs v5.00
  - Adds a password to protect the clear password feature
  - Couldn’t find a card that supports it
Takeaways

• SD cards have a password protection builtin
  – Not supported by major operating systems
  – Many implementations exist

• This protection is vulnerable to a timing attack on some manufacturer cards
Thank you!

Nicolas Oberli
@baldanos