Recovering data from USB Flash memory sticks that have been damaged or electronically erased

(Invited Paper)

B. J. Phillips
CHiPTec
The University of Adelaide
Adelaide, Australia

C. D. Schmidt
CHiPTec
The University of Adelaide
Adelaide, Australia

D. R. Kelly
CHiPTec
The University of Adelaide
Adelaide, Australia

ABSTRACT

In this paper we consider recovering data from USB Flash memory sticks after they have been damaged or electronically erased. We describe the physical structure and theory of operation of Flash memories; review the literature of Flash memory data recovery; and report results of new experiments in which we damage USB Flash memory sticks and attempt to recover their contents. The experiments include smashing and shooting memory sticks, incinerating them in petrol and cooking them in a microwave oven.

Categories and Subject Descriptors

B.3 [**Hardware**]: Memory Structures

General Terms

Experimentation

Keywords

Data recovery, semiconductor data remanence, Flash memory

1. INTRODUCTION

Under what circumstances is it possible to recover data from a USB Flash memory stick? It is well known that operating systems do not immediately remove deleted files from Flash memories but delay electronically erasing the memory until additional free space is required [2]; hence software to recover files that have been deleted but not electronically erased are standard computer forensic tools [8]. In this paper we consider the viability of recovering data that is beyond the reach of these software tools, either because the USB Flash memory stick has been damaged so that it no longer responds to software, or because the memory has been electronically erased so that it reads as all ones or all zeroes.

e-Forensics 2008 Adelaide, Australia

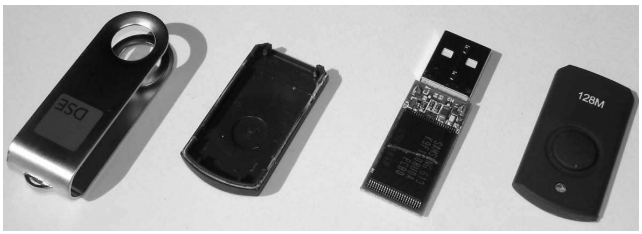

Figure 1: The components of a USB Flash memory stick. The PCB is shown with the Flash memory chip on the upper side.

The paper begins with a brief overview of the structure and theory of operation of Flash memories. In Section 2.2 we review the literature of data recovery from Flash memories. Section 3 presents the results of experiments in which we deliberately damage USB Flash memory sticks and then attempt to recover their original data.

2. BACKGROUND

Figure 1 shows a typical USB Flash memory stick, one of the inexpensive DSE 128 MB USB devices[1] used for the experiments described in Section 3. The stick contains a printed circuit board (PCB) with components soldered onto both sides. The size of the board, and of the stick itself, is dictated by the size of the USB connector and the Flash memory chip. The latter, in this case a Hynix device[2], is clearly visible in Figure 1 and occupies most of one side of the PCB.

Memory sticks also contain a controller which communicates with a host device via the USB connection. Sticks that conform to the *USB mass storage device class* [21] present a simple interface which allows the host to treat the stick as if it were a hard drive. The operating system on the host can read and write *sectors* of data and maintain the files on the stick according to any file system.

2.1 Flash Memory Theory of Operation

Flash is a *non-volatile* memory technology: data is not lost when power is removed. A particular variety, NAND Flash, is typically used for USB memory sticks due to its high density in bits per area, and its low cost per bit.

[1]Dick Smith Electronics catalog number XH8250
[2]HY27UF081G2M in a 48-pin TSOP1 package

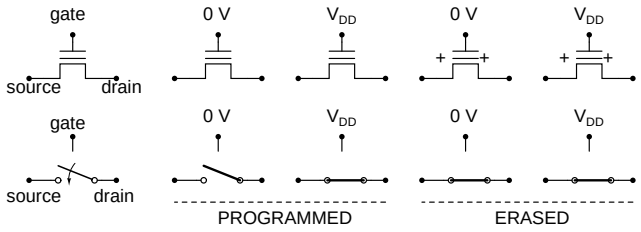

Figure 2: Operation of a floating gate field effect transistor for NAND Flash.

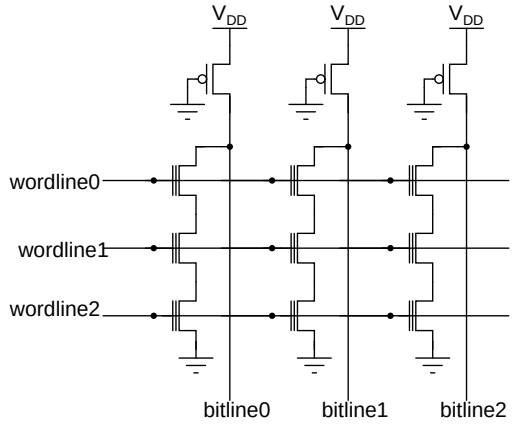

Figure 3: A simplified example of a 3-by-3 NAND Flash array.

A Flash memory cell to store a bit of data contains a single *floating gate field effect transistor* [16]. The bit is stored according to charge trapped on an isolated terminal of the transistor called the *floating gate*. This is formed by suspending a layer of conductive polysilicon in the silicon dioxide insulator between the gate and channel of a conventional field effect transistor. The floating gate can be charged or discharged by applying high voltages to the other terminals of the transistor. Electrons jump the thin insulating oxide via processes known as *channel hot electron* injection or *Fowler-Nordheim tunneling*. The latter is typically employed in NAND Flash.

Figure 2 illustrates the operation of a floating gate transistor as used for NAND Flash. When the floating gate is discharged the transistor is said to be in the *programmed* state. If a positive voltage V_{DD} is applied to the *gate* terminal, the transistor will turn on such that the *drain* terminal is connected to the *source* terminal; but if 0 V is applied to the gate, the transistor will turn off, and disconnect the drain from the source. When the transistor is in an *erased* state, a positive charge is stored on the floating gate, and the transistor will be turned on irrespective of whether 0 V or V_{DD} is applied to the gate.

Figure 3 shows a simplified NAND Flash array. Consider reading the cell connected to wordline0 and bitline0. To do this we set wordline0 to 0 V and set the other wordlines to V_{DD}. Thus the other transistors connected to bitline0 are turned on irrespective of whether they are programmed or erased. The transistor being read will pull bitline0 to 0 V if it is programmed; it will leave the bitline0 at V_{DD} if it is erased.

Now that we have discussed the structure of a *typical* Flash memory it is important to note that a huge variety of different physical structures and circuit arrangements are used by the different Flash memory manufacturers as they aggressively complete to improve density, yield and power dissipation.

2.2 Recovering Data

2.2.1 Recovering Erased Data

Once a Flash memory chip has been electronically erased it will return all zeroes or all ones when it is read. There are, however, *semiconductor remanence* mechanisms that may permit recovery of the original data. In widely cited papers, [10] and [11], Gutmann identified a large number processes that change the behaviour of a memory cell over time and that depend on the data stored in the cell. These processes include physical changes in the wires of a memory (*electromigration*), changes in transistor behaviour due to accumulation of trapped charge (*hot carriers*), and changes in distribution of contaminant ions (*ionic contamination*).

Semiconductor remanence appears in academic literature most often as a security concern for systems such as automatic teller machines or smart-cards that store confidential data over long periods. Papers which treat remanence in this way include [15] and [7]. Despite being thus widely acknowledged as a valid concern, there are very few published instances in which erased data is recovered using remanence.

The one published example of data recovered from a Flash memory using remanence is due to Skorobogatov [19, 20]. By glitching the supply voltage during a memory read operation, and also applying an erase operation that was prematurely terminated, he was able to discern the small difference in threshold voltage between memory cells that had been programmed and erased and those that had not been programmed before being erased. Even after 500 consecutive erase cycles he was still able to recover remanent data. However it is important to note that Skotobogatov's method was specific to NOR Flash memories (not NAND Flash). Moreover the techniques he used to observe changes in memory cell behaviour can not be directly applied to modern NAND Flash chips. In these devices the data is only accessed indirectly via an on-chip controller that latches whole pages of data at one time; program and erase voltages are generated internally; and program and erase sequences are handled by the on-chip controller.

Another way to approach the question of recovering data from Flash memories is from the perspective of a security agency intent on maintaining the secrecy of confidential data. Standard procedures exist for secure deletion of data from electronic media including Flash memories. For example, in the USA publications [9, 6] and [18] all contain instructions for the sanitisation of electronic media. In Australia, government agencies are required to follow the media sanitisation procedures set out in the *Australian Government Information and Communications Technology Security Manual* ACSI 33 [5]. The unclassified version of this manual gives the following procedures for sanitising a Flash memory containing information that is classified *in-confidence*, *restricted* or *protected*:

> Erase as per the manufacturer's specification or using a third party tool.

> Agencies *should* verify the effectiveness of the erasure process before approving it for use as a sanitisation method. If no effective process is available, then the media *should* be destroyed.

Just as there are different classifications of confidential information, one can consider different degrees of sanitisation. The NIST *Guidelines for Media Sanitization* [18] give instructions for *clearing*, *purging* and *destroying* electronic media. Clearing information is sufficient to protect from 'robust keyboard attacks' in which data is recovered using software techniques alone. Purging must be sufficient to protect information against recovery by specially trained personnel in a laboratory environment. To clear a Flash memory the NIST guidelines recommend overwriting using agency-approved and validated tools. The guidelines recommend no method for purging a Flash memory other than physical destruction.

2.2.2 Damaged Flash Memories

The sanitisation procedures discussed above present an alternative to erasing a Flash memory: a memory still containing data can be destroyed. How much destruction is sufficient to render the data unrecoverable?

Physically smashing a Flash memory chip into parts will not remove the charge stored on the floating gate transistors. If a transistor is still intact it may be possible to read its data bit, but the process will not be straightforward. At a feature size of 50 nm, a floating gate will have an area in the order of tens of square micrometers and the wires connecting to the transistor terminals will only be hundreds of nanometers wide.

The difficulty of recovering data from a damaged memory chip is well illustrated by the case of an EEPROM recovered from the crash of Swissair Flight 111 in 1998 as recounted in [12]. The chip's package was damaged and bond wires had been pulled loose. The semiconductor die remained in a single piece but had been exposed to salt water which had corroded aluminum bond pads and connectors on the chip. The data was ultimately recovered by repairing the chip so that its usual read mechanism worked. To do this the circuit topology of the EEPROM was first reverse-engineered by grinding other samples of the same model to reveal progressive layers of metal and polysilicon. To ensure the data on the damaged EEPROM was not disturbed, all operations on it were performed at temperatures below 100°C, even though EEPROMs typically retain data for hours at 200°C. Hence bond pads were repaired using blobs of silver epoxy which could be cured at low temperatures. Also, a scanning electron microscope was not used because beam radiation is known to alter transistor thresholds if it penetrates down to the gate oxide. A focused ion beam workstation [4, 2] was used to modify and repair sections of the chip, and parts of the circuit not required to read data were disabled.

In addition to the techniques used above, the fields of semiconductor failure analysis and reverse engineering have provided tools that could be used to extract data from shards of a broken Flash memory. These include laser and electron beam probing techniques developed at Sandia Labs [1] and semi-invasive probing with a lithium niobate crystal [22]. An accessible overview of low-cost reverse-engineering and probing techniques for integrated circuits is provided in [14].

Finally, a quote from [2] gives an insight into the perceived data recovery capabilities of well-funded laboratories:

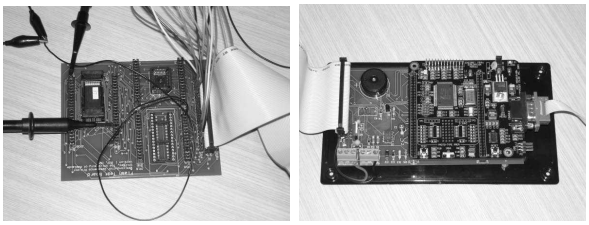

Figure 4: The test rig used to read Flash memory chips directly.

> after tests showed that 1 mm chip fragments survived the protective detonation of a control device carried aboard airborne command posts, the software was rewritten so that all key material was stored as two separate components, which were kept at addresses more than 1 mm apart on the chip surface.

3. EXPERIMENTS

This section describes a series of experiments in which we deliberately damaged USB Flash memory sticks with the aim of rendering their data irretrievable. We chose simple, low-cost ways of damaging the sticks that could be applied in a hurry. After damaging a stick we then sought to recover its original data.

We used DSE 128 MB USB Flash memory sticks as shown in Figure 1. These were pre-formatted with a FAT-16 file system. To each stick we copied a large text file, and sufficient compressed audio files to almost fill the available memory. Prior to sanitisation experiments, an image of the USB drive was taken using *DiskExplorer for Fat*[3].

Whenever a USB stick stopped functioning we attempted to read the Flash memory chip directly. To do this we built the test rig shown in Figure 4. A zero-insertion-force test socket was used to hold the Flash memory while its contents were read by a microcontroller.

3.1 Over-voltage from a Car Battery

The USB 2.0 electrical interface consists of 2 signal lines (D+ and D-), power (V_{BUS}) and ground [3]. The supply voltage V_{BUS} is rated at a maximum of 5.25 V. The signal lines have a rated maximum of 3.6 V. The Hynix Flash memory used in the DSE memory sticks has an absolute maximum rating for V_{CC} and the I/O pins of -0.6 to 4.6 V [13]. The circuit architecture of the sticks is such that none of the USB interface pins other than ground are connected directly to the Flash memory.

The following experiments where conducted using a 12 V lead acid battery from a large 6 cylinder car:

1. The ground pin of the memory stick was connected to the negative battery terminal. The power pin of the stick was connected to the positive battery terminal. The stick smoked and the hookup wire connecting it to the car battery burned out like a fuse.

2. When the stick was connected to a computer, the computer shut down immediately.

[3]Runtime's DiskExplorer for Fat, Version 3.03, available from http://www.runtime.org

3. The stick was opened and the circuit traced to reveal that a protection diode, connected in reverse bias between power and ground, had gone short circuit. When it was removed, the stick functioned correctly and it was possible to read the original data.

4. A new memory stick was connected to the battery. The battery was connected to the D+ and D- data lines. The polarity of the battery was then reversed. The stick was plugged into a computer which did not recognise that a USB device had been connected.

5. The stick was opened to revel discoloration around some of the pins of the controller chip. The Flash memory was not visibly damaged.

6. The Flash memory chip was de-soldered from the PCB and read using the microcontroller test rig. The original data was recovered.

3.2 Soaking in Water

Immersing a solid state electronic circuit in liquid can damage the circuit due to sort circuits or corrosion. The former will only occur if the circuit is still powered. Corrosion is an issue for powered and un-powered circuits but can be much worse in powered circuits due to electrolysis.

We soaked one of the DSE memory sticks in a glass of clean drinking water for 24 hours. Water was shaken out from the case and it was left on a sunny windowsill for a day to dry. Some corrosion was visible around the USB connector but the stick operated correctly and the original data was recovered.

3.3 Incinerating in Petrol

The Hynix Flash memory used the DSE memory sticks has an absolute maximum rating for storage temperature of 150°C [13]. Although the data sheet notes that stresses above the absolute maximum ratings may cause permanent damage to the device, temperatures above 150°C do not typically cause a Flash memory to suddenly lose data. For example, tests on the non-volatile memories in Microchip microcontrollers show that data is retained for long over 700 hours at 250°C[17].

The following experiments were conducted:

1. A memory stick was placed in a wide shallow can containing 100 ml of unleaded petrol and set alight (Figure 5). The petrol burned for around 90 seconds. The plastic casing of the stick continued to burn for a further 2 minutes. In this case the Flash memory was on the upper side and can be seen in Figure 5. The Flash memory chip was removed, intact from the remains of the PCB.

2. The previous experiment was repeated, this time with a memory stick arranged such that the Flash memory was on the lower side. Once again it was possible to remove the Flash memory chip intact from the remains of the stick.

3. A third memory stick was suspended by wires above a small can containing 100 ml of unleaded petrol. This arrangement produced a long slow burn of over 4 minutes duration (Figure 6). For most of this time the stick was held within the flames. Following this procedure the Flash memory chip was removed intact from the remains of the PCB. The chip's package has blistered.

4. Only the first of the three memory chips above was tested using the microcontroller rig. Its original data was recovered.

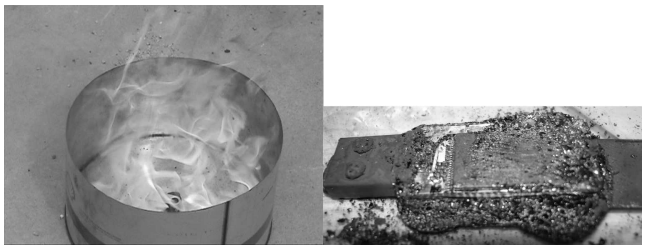

Figure 5: Left: a memory stick is burned in petrol in a wide, shallow can. The stick is lying with the Flash memory up. Right: the remains of the memory stick.

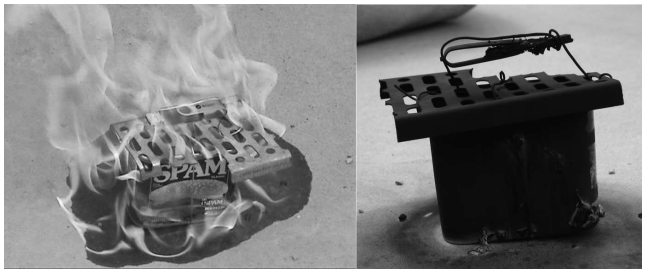

Figure 6: Left: a memory stick is incinerated over a can of burning petrol. Right: the remains of the memory stick and can.

3.4 Stomping

The following experiments were conducted by a fit individual wearing heavy shoes with a solid rubber heel:

1. The memory stick was placed on an asphalt surface and stomped on 7 times over a 5 second interval (Figure 7). The memory stick's package was intact and after bending the USB connector back into shape, it was possible to read the memory using a computer.

2. This stick was stomped upon for a further 10 seconds, and once again it was possible to bend the USB connector into shape and read the memory using a computer.

3. A further 5 seconds stomping broke the package away and dislodged a surface mount capacitor and resistor and the crystal. The computer was unable to recognise the stick and reported the error, 'One of the USB devices attached to this computer has malfunctioned, and Windows does not recognise it'.

4. The capacitor, resistor and a new crystal were soldered back onto the PCB. The computer recognised the stick and read its contents correctly.

5. The PCB without the stick packaging was stomped upon and ground underfoot for a further 5 seconds on each side. The PCB did not shatter but many of

Figure 7: Left: A memory stick is stomped upon. Right: The remains of a memory stick after it has been struck 36 times with a hammer.

Figure 8: Left: a memory stick is shot with a pistol. Right: the remains of the memory stick. The 3 mm slice of chip is all that could be found of the Flash memory.

the smaller components were dislodged. The Flash memory chip suffered some scratching on the surface of the package but was otherwise left intact.

6. The Flash memory chip was de-soldered from the PCB and read using the microcontroller test rig. The original data was recovered.

3.5 Striking with a Hammer

A memory stick was placed on a concrete surface and firmly struck 36 times over a 20 second interval with a metal carpenter's hammer. Although the plastic packaging did not come away from the PCB, it was crushed such that the Flash memory chip broke into pieces. The largest fragment remained soldered to the PCB and was a 6 mm wide slice of one side of the chip (Figure 7).

3.6 Shooting with a Pistol

Two memory sticks were shot at 5 m range with a Glock 17, 9 mm semi-automatic pistol loaded with 115 gm full-metal-jacket bullets.

Figure 8 shows the parts of the first stick that were recovered. The large fragment of PCB, containing a part of the Flash memory, was found at the back of the range against the bullet stop.

For the second experiment the stick was taped to a telephone directory such that after it was shot, many of the fragments were found embedded inside the directory. Figure 9 shows the results.

3.7 Cooking in a Microwave Oven

When metal is placed inside a microwave oven, it acts as an antenna. Current will flow in the metal and cause it to heat up. Also, large voltage differences can occur between adjacent conductors which may lead to arcing between the

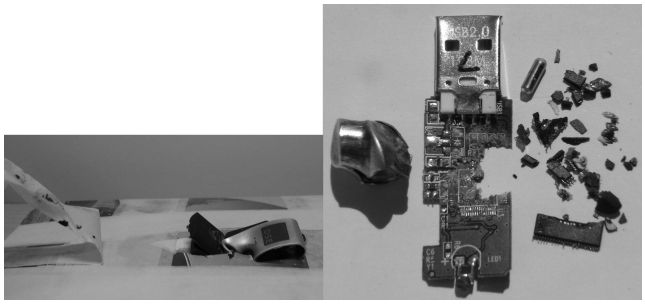

Figure 9: Left: a memory stick is embedded in the telephone directory that supported it when it was shot with a pistol. Right: the remains of the memory stick and projectile.

conductors. Intense heat and arcing within an integrated circuit are likely to cause permanent failures by damaging its internal conductors and thin insulating layers.

The following experiments were conducted with a domestic microwave oven set on *high*:

1. A memory stick was set on the microwave platter and cooked for 5 seconds alongside a mug of water. When it was removed from the oven, the stick worked correctly and it was possible to recover the original data.

2. The stick was set on an overturned bowl and cooked for 5 seconds alongside a mug of water. When the stick was plugged into a computer, the computer failed to recognise that a USB device had been connected.

3. The stick was set on an upturned mug and cooked for 1. The stick sparked, smoked and eventually the plastic package caught alight (Figure 10). When the plastic was chipped from the Flash memory chip, some of the chip's packaging also come away to reveal the silicon die beneath (Figure 10).

4. A new memory stick was set on the microwave platter and cooked for 15 seconds without the mug of water. The stick sparked and sizzled. When plugged into a computer the operating system reported the following error, 'Power surge on hub port. A USB device has exceeded the power limits of its hub port'. Two small blisters were observed on the package of the Flash memory chip.

5. The Flash memory chip was de-soldered from the PCB and read using the microcontroller test rig. The chip did not respond to commands so that it was not possible to read the original data.

4. CONCLUSIONS

Although semiconductor remanence is a valid concern for security system designers, to the best of the authors' knowledge, there has not been a published account in which data is recovered from a NAND Flash memory once it has been erased according to the manufacturer's specification.

We were able to recover data from USB Flash memory sticks which had been stomped upon, damaged with an overvoltage from a car battery, soaked in water and incinerated

Figure 10: Left: a memory stick catches alight in a microwave oven. Right: the remains of the memory stick. The Flash memory die is visible through its damaged packaging.

in petrol. In cases where the USB stick itself ceased to function, we were able to read the data directly from the Flash memory chip using a microcontroller.

Shooting a memory stick with a pistol does appear to be an effective way of smashing the Flash memory chip. In the DSE memory sticks, the Flash memory occupied almost the entire area of one side of the PCB. In the experiments, bullets that struck the stick towards the centre of its package smashed the Flash memory into shards and dust. The largest shard of memory found was the 3 mm long slice from one edge of the chip shown in Figure 8. Smashing the stick by striking it with a hammer on a hard surface was as effective as a well aimed shot with a pistol.

In our experiments, cooking a memory stick in a microwave oven quickly rendered the stick unresponsive and damaged the Flash memory chip so that it could not be read directly using a microcontroller.

5. ACKNOWLEDGMENTS

The authors are grateful for the assistance of Adam Langman (who conducted the petrol incineration experiments), the staff of the Marksman Indoor Firing Range Adelaide, Andrew Allison and Ian Matthews (who donated microwave ovens), and the technical staff of the School of Electrical & Electronic Engineering at the University of Adelaide.

6. REFERENCES

[1] C. Ajluni. Two new imaging techniques promise to improve IC defect identification. *Electronic Design*, 43(14):37–38, July 1995.

[2] R. Anderson. *Security engineering: a guide to building dependable distributed systems*. Wiley, New York, 2001.

[3] Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, NEC, and Philips. Universal serial bus specification. Technical report, Apr. 2000. Revision 2.0.

[4] J. H. Daniel, D. F. Moore, and J. F. Walker. Focused ion beams for microfabrication. *Engineering Science and Education Journal*, pages 53–56, Apr. 1998.

[5] Defence Signals Directorate. Australian government information and communications technology security manual. Technical Report ACSI 33, Defence Signals Directorate, Department of Defence, Sept. 2005.

[6] Department of Defense. National industrial security program operating manual NISPOM January 1995. Technical Report DoD 5220.22-M, Department of Defense, Department of Energy, Nuclear Regulatory Commission, Central Intelligence Agency, July 1997.

[7] G. Di Crescenzo. Security of erasable memories against adaptive adversaries. In *Proc. ACM Workshop on Storage Security and Survivability (StorageSS'05)*, Nov. 2005.

[8] D. Farmer and W. Venema. *Forensic Discovery*. Addison Wesley Professional, dec 2004.

[9] J. K. Goldston. A guide to understanding data remanence in automated information systems. Technical Report NCSC-TG-025, National Computer Security Center, Sept. 1991.

[10] P. Gutmann. Secure deletion of data from magnetic and solid-state memory. In *Proc. Sixth USENIX Security Symposium*, pages 77–90, July 1996.

[11] P. Gutmann. Data remanence in semiconductor devices. In *Proc. Tenth USENIX Security Symposium*, Aug. 2001.

[12] R. Haythornthwaite, A. Earle, A. Rahal, and D. James. Case history: novel FA techniques used to recover EEPROM data fromthe Swissair 111 crash. In *Proc. 39th Annual IEEE International Reliability Physics Symposium*, pages 283–288, Apr. 2001.

[13] Hynix Semiconductor. 1Gbit (128Mx8bit / 64Mx16bit) NAND Flash Memory. Technical datasheet, Hynix Semiconductor, Nov. 2005. Rev 1.1.

[14] O. Kömmerling and M. Kuhn. Design principles for tamper-resistant smartcard processors. In *Proc. USENIX Workshop on Smartcard Technology*, pages 9–20, May 1999.

[15] M. Neve, E. Peeters, D. Samyde, and J.-J. Quisquater. Memories: a survey of their secure uses in smart cards. In *Proc. Second IEEE International Security in Storage Workshop (SISW'03)*, pages 62–72, Oct. 2003.

[16] J. M. Rabaey, A. Chandrakasan, and B. Nikolic. *Digital Integrated Circuits*. Prentice Hall, 2nd edition, Dec. 2002.

[17] R. Richey. Flash memory technology: Considerations for application design. Application Note TB072, Microchip Technology Inc., May 2003.

[18] M. Scholl, R. Kissel, S. Skolochenko, and X. Li. Guidelines for media sanitization. NIST Special Publication 800-88, National Institute of Standards and Technology, Gaithersburg, MD 20899-8930, Feb. 2006.

[19] S. Skorobogatov. Data remanence in flash memory devices. In *Cryptographic Hardware and Embedded Systems - CHES 2005*, volume 3659 of *Lecture Notes in Computer Science*, pages 339–353. Springer, Aug. 2005.

[20] S. Skorobogatov. Semi-invasive attacks - a new approach to hardware security analysis. Technical Report UCAM-CL-TR-630, Computer Laboratory, University of Cambridge, Apr. 2005.

[21] USB Implementers Forum. Usb device class definition for mass storage devices. Spec. Rev. 1.2, June 2003.

[22] J. M. Wiesenfeld. Electro-optic sampling of high-speed devices and integrated circuits. *IBM Journal of Research and Development*, 34(2/3):141–161, Mar. 1990.