Self-Encrypting Disks pose Self-Decrypting Risks

How to break Hardware-based Full Disk Encryption

Tilo Müller, Tobias Latzo, and Felix C. Freiling
Friedrich-Alexander Universität
Erlangen-Nürnberg, Germany
{tilo.mueller,tobias.latzo,felix.freiling}@cs.fau.de

ABSTRACT

Hardware-based full disk encryption (FDE) drives, such as Intel's SSD 320 and 520 series, are widely believed to be a fast and secure alternative to software-based solutions like TrueCrypt and BitLocker. Since encryption keys are stored inside a crypto chip of the disk drive itself, rather than in RAM or inside the CPU, traditional attacks like cold boot appear to be futile. We show, however, that depending on the configuration of a system, hardware-based FDE is generally *as insecure* as software-based FDE. The reason for this is a new class of surprisingly simple attacks that exploit the fact that a self-encrypting drive does not notice whether the SATA cable is replugged to a different computer, effectively turning a self-encrypting device into a self-decrypting device. We also adapt known attacks from software-based FDE and evaluate the practicability of all attacks with twelve different computer systems, including desktops and laptops, that were configured in their "most secure" way. We were able to break hardware-based FDE on eleven of those systems provided that they were running or in standby mode.

Categories and Subject Descriptors

E.3 [**Data**]: Data Encryption; B.m [**Hardware**]: Misc.

General Terms

Security

Keywords

Self-Encrypting Drives, Cold Boot, Hot Plug, Evil Maid

1. INTRODUCTION

Today, sensitive data of an organization is spread over mobile clients and servers throughout the world and is often compromised by lost or stolen laptops [22]. Also, server systems can be subject to confiscation by law enforcement agencies [19]. Setting passwords on OS and BIOS level is inadequate for protecting data against unauthorized access in such scenarios, and encryption becomes necessary. A SECUDE survey [26] on U. S. enterprises published in 2012 revealed that 75 % of all organizations use encryption, from which hard disk encryption (58 %) is most popular. In 2010, a survey by Ponemon [23] came to a similar result, namely that 59 % of all U. S. enterprises deploy disk encryption.

Technical report for the German talk "(Un)Sicherheit Hardware-basierter Festplattenverschlüsselung" given at the 29th Chaos Communication Congress (29c3): Not my department, Chaos Computer Club (CCC), Hamburg, Germany, December 2012.

1.1 Full Disk Encryption

Full disk encryption (FDE) means to apply encryption transparently to the entire hard disk in order to enforce data security in scenarios where a disk, or a whole machine, is *physically* lost or stolen. For this purpose, FDE needs either support from the operating system software or from special hardware. Hence, full disk encryption can basically be classified into *software-based* and *hardware-based* solutions. Whereas software-based solutions like BitLocker and TrueCrypt are available for end-users since more than a decade, hardware-based solutions have had their breakthrough only recently with the rise of solid-state drives (SSDs). Intel's SSD 320 and 520 series are prominent examples for hardware-based FDE. These disks have a built-in encryption logic inside the disk drive controller, so that encryption keys are never present in the computer's main memory or CPU. Therefore, such systems are often called *self-encrypting disks* or *self-encrypting drives* (SEDs).

Besides undisputed advantages of this technique, such as maintainability, OS transparency, and a significant gain in performance [4], SEDs are commonly believed to be more secure than software-based solutions. A Ponemon study [24] on the use of SEDs from 2011 reports that 70% of the respondents believe that SEDs "would have had an enormous and positive impact on the protection of sensitive and confidential data." In addition, according to this study 65% believe that SEDs "will become the standard of excellence in desktop and laptop drive security" [24]. Overall, it is argued that "while self-encrypting drives are a new technology, the IT staff we interviewed believe they are more secure than software-based encryption" [24].

The common technological argument for this belief is twofold. First, by design SEDs always provide *full* disk encryption, including the master boot record (MBR). With software-based solutions, the MBR must necessarily be present unencrypted for bootstrapping reasons, leading to the threat of MBR manipulation attacks like *evil maid* [16]. Second, with SEDs the encryption key never enters RAM, removing this entity as potential attack vector. The attack vector "RAM" is well-known to harm software-based FDE in the form of *cold boot* [12] and *DMA attacks* [2, 5].

1.2 Physical Access Threats

We assume that an attacker wishes access to the data on a hard disk through physical access. This can be achieved in different ways, for example, by attacking the key management. But, as we will see, retrieving the encryption key is only one possibility to gain access to the data. Other attacks target protection mechanisms of the operating system, or trick a user into revealing his or her password. The basic requirement is that the attacker has physical access. We do not consider remote attackers that infiltrate systems with malware over a network connection. Disk encryption is a protection mechanism to provide data confidentiality after a drive is lost, stolen, or seized.

As we will see, most physical access attacks require the target system to be running or in standby mode. If the machine is running, we additionally assume it is locked because otherwise accessing the data becomes trivial. The most common scenario for physical access attacks are laptops, that are in standby mode, and servers or desktops, that are running but locked. Usually, laptops are considered as the premier targets for physical access attacks since they get frequently lost and stolen at public places like airports [22]. However, desktops and servers can be subject to such attacks, too. For instance, desktops and servers may be accessed by attackers from inside the enterprise or confiscated by law enforcement.

To the best of our knowledge, it has not been investigated methodically yet if, and to which extent, SEDs are affected by physical access attacks. On the contrary, the simplicity of the SED design and the protection through dedicated hardware as well as the marketing strategy from many vendors have led the community to believe that SEDs are immune to all or at least most of these attacks. However, as shown in this paper, this is incorrect.

1.3 Contributions

In this paper, we systematically evaluate the security of hardware-based FDE and compare it with disk encryption based on software for the first time. For this purpose, we take the natural threat model of disk encryption as a basis, i.e., we focus on physical access attacks. More detailed, our contributions and insights to the field of SED security are:

- *Hot Plug Attacks:* We show that, depending on the hardware configuration of a system, there exists a class of attacks that is specific to self-encrypting disks. Roughly speaking, the idea of these attacks is to move an SED from one machine to another without cutting power, e.g., by replugging only the SATA cable. Consequently, we call these attacks "hot plug attacks". In many cases, these attacks even succeed against switched-off SEDs if a computer is in standby mode.
- *Adaption of Known Attacks:* Motivated by the discovery of hot plug attacks, we studied the applicability of attacks known from software-based FDE to hardware-based systems. For every setting in which a known attack against software-based FDE exists, we found a successful attack against hardware-based FDE. These scenarios include DMA-based attacks, cold boot attacks, and evil maid attacks. In this sense, hardware-based FDE is *as insecure* as software-based FDE.
- *Evaluation Results:* Our study is based on experiments with twelve up-to-date computer systems, including eight laptops and four desktop machines. Not all systems are equally vulnerable because the security of an SED-based system depends on its hardware configuration. Rather surprisingly, it is not the specific model of the SED that determines security, but the specific motherboard model and BIOS version of the computer. Overall, only a few hardware-based FDE systems withstood more attacks than software-based systems. The majority of machines is equally vulnerable in both scenarios.
- *Attack Guidelines:* We provide an attack guide describing best practices in attacking SEDs for certain systems and system states, based on our evaluation results. We suggest a priority order for all identified attacks, and we apply our guidelines exemplarily to our test set. Video material showing how we performed our attacks is provided at `http://www1.cs.fau.de/sed`.

2. BACKGROUND AND RELATED WORK

We now give an overview of physical access attacks against software-based FDE (Sect. 2.1) and background information on hardware-based FDE (Sect. 2.2).

2.1 Physical Access Attacks

We identified three types of physical access attacks against software-based FDE, that we present in chronological order.

2.1.1 DMA-based Attacks

A major vulnerability of software-based systems arises from the fact that encryption keys are stored inside main memory, basically exposed to be read out with direct memory access (DMA). But DMA-based attacks can even *write* into memory. In 2005, Dornseif, Becher, and Klein showed how to compromise an Apple Macintosh through the DMA capabilities of a FireWire device [10]. Similar attacks were later used to defeat BitLocker under Windows 7 [2]. In principle, all other DMA interfaces exhibit the same vulnerability, including PCI and ExpressCard [13], and, as recently shown in 2012, Thunderbolt [5]. Possible countermeasures, e.g., in the form of the IOMMU, exist in current hardware but are not supported by software to date.

2.1.2 Cold Boot Attacks

Another way to access main memory exploits the *remanence effect* of RAM, first described by Gutmann [11], which says that memory contents fade away gradually over time. This effect enables attackers to restore keys from RAM after rebooting the machine with a mini OS from USB thumb drive. Such attacks have become known in 2008 as the *cold boot attack* by Halderman et al. [12]. Recent studies [8] confirm the practicability of cold boot attacks on various computer systems. Unlike DMA attacks, which require DMA capable interfaces, cold boot attacks are more generic and countermeasures are difficult. Until today, all widespread software-based FDE solutions are vulnerable to cold boot attacks, including BitLocker, FileVault, and TrueCrypt.

An academic solution to the cold boot problem proposed in the literature is to move encryption keys from main memory into CPU caches or into CPU registers [18, 20]. The Linux patch TRESOR [18] stores keys inside debug registers, but as proven by Blass and Robertson [1], TRESOR does not defeat DMA-based attacks.

2.1.3 Evil Maid Attacks

Another vulnerability we identified for software-based FDE came up in 2009. It arises from unencrypted master boot records that cannot be eliminated due to bootstrapping reasons. Since MBRs are unencrypted, they can always be manipulated, for example, through the infiltration of software keyloggers. Such attacks are called *evil maid attacks* (Rutkowska [16]), or just referred to as *bootkits* (Kleissner [21]). They typically require access to the target machine twice: before and after the victim enters a password. With the first physical access, attackers install a keylogger to the MBR, and with a second physical access they collect the logged password. As shown by *tamper and revert* attacks from Türpe et al. [29], securing the boot process by means of the trusted platform module (TPM) does only partially mitigate this threat. More promising proposals like *Anti Evil Maid* from Rutkowska [17] exist, but they are not present in proprietary systems.

2.2 Hardware-based FDE Background

Solid-state drives (SSDs) have become the storage of choice in the recent past, predominantly in the laptop market but also for desktop systems. In a survey from SNIA [9], the word is that "it is likely that by about 2017 all HDDs will shift to SSD capable units." SSDs are mostly favored because of their mechanical robustness and their performance impact that stems from built-in NAND flash memory. However, many SSD vendors brought another feature to the consumer market, namely drive-level encryption.

Early disk encryption techniques were already available in hardware before the rise of SSDs, but we focus on recently popular models like the Intel SSD 320 and 520. In particular, we focus on SSD models that provide built-in encryption based on *ATA security authentication*. Another authentication method for SEDs is provided by the *TCG SSC Opal specification* (cf. Sect. 2.2.3).

2.2.1 ATA Security

Advanced technology attachment (ATA) [27] is the standard interface for connecting mass storage devices since the 1990s. The ATA security feature (a.k.a. DriveLock, HDD Password, or Security Lock) was standardized in ATA-3 (1997). Today, serial ATA (SATA), standardized in ATA-7 (2003), is the prevalent bus for connecting SSDs. ATA drives have the ability to get locked and remain "inaccessible" until a correct password is entered, but a drive lock does not necessarily involve encryption.

ATA security defines two kinds of passwords, *user* and *master*, as well as two security levels, *high* and *maximum*. On high security level, the user and master password can be used interchangeable for unlocking the drive. On maximum level, the master password can only be used to securely erase the drive and to reset the user password, but not to read data. In other words, the master password enables a company or vendor to reset disks in the case of password loss. Both user and master passwords are defined to be 32-byte long.

Plain ATA passwords provide only little more security than BIOS and operating system passwords, because an ATA password does *not* imply that the drive is encrypted. That is, plain SATA drive locks are not to be confused with SEDs. To remove ATA passwords, attackers can bypass drive controller chips and access storage memory directly. Usually this requires specialized hardware, but full access to the disk can always be gained [25]. To complete the physical security layer, ATA passwords must be paired with drive-level encryption.

2.2.2 Self-Encrypting Disks

Many SSD-vendors started to ship their products with built-in encryption facilities, so called self-encrypting disks or drives (SEDs). These models use the AES standard to encrypt user data. For example, Intel's SSD 320 series uses AES-128 [14], and Intel's SSD 520 series was promoted to use AES-256 [15].

Each SED has a unique encryption key which is generated from entropy sources inside the drive, such that it is not known to the manufacturer. This key is called the *media encryption key* (MEK), also known as *data encryption key* (DEK). The MEK is used to encrypt the actual user data and is encrypted by means of a *key encryption key* (KEK). KEKs are derived from user passwords and disks are powered up locked until the correct password is entered. Re-encryption is avoided because only the MEK must be encrypted newly with the KEK when a password is changed. Setting a new password, or unsetting a password completely, does not involve lengthy encryption or decryption procedures either.

Encryption works out-of-the-box for SEDs, but to activate any degree of protection, the drive password must be set. This password can often be set with ATA commands, as described above, from any compatible BIOS or *extensible firmware interface* (EFI). Most laptops have built-in support for ATA security commands for years, and so it seems natural that many SED vendors build on this infrastructure for authentication.

2.2.3 TCG Storage Security: Opal

Another infrastructure that SEDs can be build on is the standardized *Opal Security Subsystem Class* (Opal SSC) from the Trusted Computing Group (TCG) [28]. Opal SSC is a set of specifications that SED vendors can comply with to integrate their hardware into a trusted platform host. Some SED manufacturers provide OPAL compliance for their devices, others not. According to the TCG at least Hitachi, Samsung, Seagate, and Toshiba offer Opal-compliant disks, but many other vendors like Intel and Kingston do not. For example, Intel states about its SSD 320 series that "current Intel SSDs do not support this specification [Opal]; however, this may be added to future Intel SSDs" [14]. Also the follow-up Intel SSD 520 [15] does not support Opal and we could not find any Intel SSD that is compliant to this standard at the time of this writing.

Contrary to the SATA based approach for SED authentication, Opal SSC suggests another procedure: "When the BIOS requests the Master Boot Record from the drive, the drive instead returns the pre-boot record to the user. [..] The pre-boot image requests the Authentication Credentials from the user."

3. HOT PLUG ATTACKS

We now describe a novel attack that is specific to hardware-based FDE; software solutions are not affected.

3.1 Attack Concept

The attack is inspired by one variant of the cold boot attack, described by Halderman et al. [12]. In this variant of the cold boot attack, RAM chips are removed from running

PCs and then replugged into another PC in order to extract their contents. But as it is pointless to replug RAM chips in the context of hardware-based FDE, we considered to replug the disk itself instead. This idea turned out to be an effective though simple attack against all SEDs.

The implementation flaw of SEDs that we exploit is the fact that SEDs do not detect whether SATA cables are unplugged as long as they stay connected to power. An SED gets locked only if its power connection is cut. If its data connection is cut, however, it stays unlocked. This leads us to the following scenario: With physical access to a running PC, the SATA connector of an SED can be unplugged, and then be replugged into another PC. During this procedure, the original PC acts as energy supplier and keeps the disk on power. The second PC, which is under full control of the attacker, acts as data collector. The original PC crashes a few seconds after disk removal, e.g., with a blue screen under Windows, but the SED stays unlocked. If the attacker's PC supports SATA hot-plugging, SEDs can directly be read out. Otherwise, the attacker must boot or reboot the machine first. But SATA hot-plugging is not a requirement for the attack and in both cases full access to the data is possible without the password.

We call this attack *hot plug attack* because it requires the disk to be running and unlocked before seizure. PC systems that are switched off completely, such that the disk is off and locked, cannot be attacked by hot plug attacks. However, physical access attacks against software solutions, like cold boot attacks, require the system to be running, too. As a consequence, if a system is running, SEDs are generally more insecure than software-based FDE due to the ease and effectiveness of hot plug attacks. In this view, a switched-on SED behaves more like a "self-decrypting disk" than a self-encrypting disk.

3.2 Desktop and Server Systems

On running desktop and server systems, the hot plug attack is advantageous over cold boot and DMA attacks, because it works irrespectively of BIOS settings and additional DMA interfaces. Settings like "password on reboot" and the "boot device order" do not affect hot plug attacks. Furthermore, the presence of DMA ports is not required. Since server systems are running 24/7, they constitute a perfect target for hot plug attacks. This can, for example, be relevant for law enforcement executing a search warrant against server clusters.

Another advantage with desktop and server systems is their construction type. SEDs are connected to the motherboard with long and flexible SATA cables that can be replugged into a nearby machine easily. As we will see, the case for laptops is more sophisticated due to their construction type. Servers and desktops, however, are well suited for hot plug attacks.

3.3 Laptops and Suspend-to-RAM

The case of laptops is often considered as more critical regarding the security of full disk encryption, because laptops are constantly at risk to get lost or stolen. Unlike server clusters, which are usually protected by guards and other forms of physical security, laptops can fall into the wrong hands quickly. In laptops, however, the power and data connection of an SED cannot be handled unfettered as in desktops. To the contrary, the power and SATA interfaces inside laptops are usually connected directly to the board, without cables. This makes hot plug attacks against laptops more sophisticated.

Another complicating point with laptops is that they are not running 24/7 like servers. Laptops are mostly carried along in standby mode and get only switched on immediately before usage. In other words, it is more likely that a laptop is lost or stolen when it is in standby mode, than when it is running. The standby mode of choice for most people is ACPI S3, i.e., *suspend-to-RAM*. In this mode, an SED gets switched off and is locked. The same holds for ACPI S4, i.e., *suspend-to-disk*.

So the question we were faced with was: How can we deploy hot plug attacks against laptops that are suspended to RAM? Rather surprisingly, this is often possible although the disk is locked. Here we profit from another implementation flaw available on virtually all laptops today. Laptops require an ATA password on boot, but they unlock the disk *automatically* on wakeup from S3. This behavior leads us to the following attack: When a disk is switch-off and locked, it can be removed out of the laptop chassis. So we remove the disk during S3, and afterwards we install SATA and power extension cables between the drive and the board. Then we wake up the laptop, and regardless of the extension cables we have installed, the SED gets automatically unlocked. For the remaining attack, we can proceed analogously to the desktop case, i.e., we replug the SATA cable into a second machine to access data. As a result, laptops in suspend-to-RAM mode can often be attacked with hot plug attacks just like desktop systems.

And since attackers can put running laptops into sleep mode, e.g., by closing the lid, running laptops are equally affected. Both lockscreens from Windows and Linux allowed us to put a laptop into sleep without privileges. As a consequence, our attack is no longer restricted to the case that the drive is cycled on. Although an SED taken by itself is secure in this state, the overall system is not.

3.4 Power Supply and Temperature

New laptops from Lenovo detect whether an SED power connector is released during S3. As the standby-based solution requires us to unplug both, the SATA and the power connector, our attack fails against laptops with this property (cf. Sect. 5.3). A way to overcome the detection property would be given if we can unplug the power connector of an SED so briefly that it is not getting locked. If that would be possible, we could pull SEDs out of running laptops and replug them to an external power supply. We thought this might be possible, because from cold boot attacks we now that RAM contents are preserved for several seconds without power. So by analogy to the cold boot attack, we conjectured that SEDs may stay unlocked if we cut power briefly. We unplugged the power cable and replugged it within less than a second several times. Additionally, we cooled down SEDs with cooling sprays and put them into a freezer, but we did not succeed. This indicates that SEDs get locked immediately after power is cut, independently of their operating temperature.

3.5 Retrieving ATA passwords from RAM

The interesting point with machines that are suspended to RAM is the following: All drives are disconnected from power and, consequently, SEDs are locked during S3. Upon wakeup,

however, they get unlocked *automatically*. Self-encrypting disks can only get unlocked through ATA commands with the correct password. This implies that the ATA password must be present in RAM or NVRAM during S3, because other components are not energized. Furthermore, this implies that the password must be present in RAM or NVRAM during the entire uptime, because the ATA password cannot be regained from the disk. Hence, the password is probably retrievable from the computer's memory, similar to encryption keys from software-based FDE which are retrievable through conventional cold boot attacks.

We spent considerable effort to search for the password in RAM and NVRAM images, but we did not succeed. Either the ATA passwords are not stored in clear text but rather in a scrambled or obfuscated form, or – more likely – they are stored inside a protected NVRAM region that is not accessible by software. We could access the first 256 bytes of NVRAM on our test systems through BIOS interrupts, but we conjecture that the NVRAM size is 512 bytes where the upper 256 bytes contain sensitive information such as ATA passwords that are not accessible.

4. ADAPTING KNOWN ATTACKS

Given the effectiveness of hot plug attacks, we now ask whether there are more attacks against SEDs. As a starting point we chose well-known attacks from software-based FDE. Overall, we systematically exposed hardware-based FDE to the same threats in which software-based solutions are vulnerable. We argue that for physical access attacks on software-based FDE, either the very same attack works on hardware-based FDE or there exists an adaption of the attack that succeeds under similar conditions.

4.1 Evil Maid Attacks on SEDs

As compared to other physical access attacks, evil maid attacks do not require the target system to be running or in standby mode but work against switched-off systems. The term "evil maid" is rooted in the following scenario: Let the victim be a traveling salesman who leaves his encrypted laptop in a hotel room and goes out for dinner. An evil maid can gain physical access to her target system unsuspiciously, and so she can replace the MBR with a modified version that additionally performs keystroke logging. Later on, the unaware salesman boots up his machine and enters the password as usual. On the next event, the evil maid reads out the logged password.

In software-based FDE, the master boot record (MBR) of an encrypted hard disk can be manipulated because it must be present unencrypted for bootstrapping. We now argue that this attack can easily be adapted to SEDs even though their MBRs are encrypted. Again, let the victim be a traveling salesman who leaves his hotel room to go out for dinner. An evil maid breaks into his room, but this time she removes the target SED, steals it, and *replaces* it with her own drive before she leaves the room. Later on, the unaware salesman boots up his machine, and a one-to-one copy of his familiar password prompt is displayed. He enters the password, since the forgery can visually not be noticed, and it is immediately sent to the evil maid over a network connection.

Of course, the salesman becomes suspicious when his OS and user data cannot be decrypted, but then it is too late and the evil maid already owns the SED as well as the password. To leave no traces about the connection, or the attack in general, the replaced drive additionally wipes itself. If an identical drive model is used, the salesman may not even recognize that his original drive is gone and probably believe in a hardware failure. Compared to traditional evil maid attacks, this variant requires only *one* physical access – given the fact that a network is in range. But with the increasing availability of wireless networks this is not a limitation (or alternatively, evil maids could set up her own hotspot nearby).

We implemented this attack in practice against Fujitsu PCs that display a text-based password prompt. We used a modified version of the GRUB-2 bootloader, the Linux kernel, and a Debian distribution on top of it. We patched GRUB-2 to be silent while booting Linux, and we patched the Linux kernel to display a password prompt that is similar to the original one. We let the kernel display the prompt to save boot time, because long delays arising from userland prompts are suspicious. After the password is logged in kernel mode, we forward it to the userland and from there, we send it to a predefined IP address via a network connection. After the password has been transmitted, the screen stays blank, and the user probably reboots the machine before he realizes his data loss.

The attack demonstrates that – although MBRs of SEDs are encrypted – the boot process can be manipulated in a variety of ways. Apart from replacing an SSD, a tiny USB thumb drive could be plugged in, the BIOS (or EFI) could be flashed, or the entire target machine could be replaced. Although we did not perform these variants in practice, they pose realistic threats against SEDs. In the case of BIOS and EFI manipulations, the password can be logged without the need to replace the disk, similar to traditional MBR manipulations.

4.2 DMA Attacks on SEDs

DMA-based attacks were pioneered by Dornseif, Becher, and Klein [10] in 2005. Their work was the beginning of a series of attacks that exploit DMA interfaces like PCIe [7], ExpressCard, FireWire [3], and Thunderbolt [5]. In the original attack, an Apple Macintosh was compromised via direct memory access from a malicious iPod. However, in the original attack, disk encryption was not considered explicitly. One possibility to deploy DMA attacks against disk encryption is to target the key of software-based FDE that is kept in main memory. Using DMA, main memory can be scanned for possible keys and these keys can later be used to decrypt the disk. Another possibility to deploy DMA attacks stems from the fact that attackers can *write* into memory and manipulate the system space. This can, for example, be exploited to unlock an OS lockscreen, as proven by attacks against Windows 7 with activated BitLocker [2].

When attacking SEDs, the first variant of DMA attacks (accessing the key in RAM) fails because the encryption key is not present in RAM. But the second variant (unlocking the screen) works the same way. The basic assumption about the state of the system is that the target must be running or in standby mode. We successfully reproduced this attack over FireWire against self-encrypting disks in practice. Our target systems were fully patched Windows 7 machines with password protected SEDs. On the attacking side, we used the Linux software *Inception* [6] to unlock our targets and to read out data. Inception features options to break into

Windows, Mac OS X, and some Linux distributions.

At first glance, these attacks are restricted to targets with built-in FireWire port, which is often not the case these days. But any ExpressCard (or PCMCIA) slot suffices as well, which is the case for many laptops. The reason is that, although the screen is locked, Windows installs FireWire drivers in the background as soon as an ExpressCard-to-FireWire adapter is plugged in. In the official PCIe specification, it is stated that PCIe supports *hot-plugging* for desktop machines, too. But we found this feature to be either not supported by the OS or motherboards, and consequently we could not demonstrate the attack against desktop machines without FireWire port. In future, more systems will be shipped with Thunderbolt, and Thunderbolt is known to enable DMA transfers, too [5].

4.3 Cold Boot Attacks on SEDs

Halderman et al. [12] successfully attacked software-based FDE in 2008 with a technique known as "cold boot attacks". The basic idea of these attacks is to access the key in main memory after a reboot with a mini OS from USB drive or boot CD. In contrast to common belief, the contents of main memory fade away gradually over time (*remanence effect*), and it can take as long as 30 seconds for memory contents to completely disappear after the computer has been powered off. This interval can be extended to minutes by cooling down the memory chips. Rebooting the system must ensure that main memory is not altered unnecessarily.

Rebooting the target system in order to dump RAM contents is pointless against SEDs because the key is not in RAM. But most of our test machines do not ask for the disk password on reboot, and on such machines "cold boot" attacks become trivial. Only a few laptops provide a BIOS setting called "ATA password on reboot" that can be activated. Many laptops, and all desktop machines of our test set, do not provide this setting. Hence, the vulnerability here is not only a configuration problem, but necessary settings are often just missing. We were able to reboot such systems with external drives to start a Linux live system. From the Linux system, we were able to mount partitions of the (still unlocked) SED, and so we read out data without the password. This attack is simple, even simpler than the hot plug attack, but it effectively breaks the encryption feature of all SEDs. Again, an SED behaves more like a "self-decrypting disk" here.

Intel states about its SEDs that "the drive password is required each time the drive is powered on" [15], meaning that it gets locked each time it is powered off. Hence, the drive does not get locked on reboot because it fails to realize that the system reboots as long as it stays connected to power. We verified that hardware resets and software reboots are equally effective, because hardware resets might cut power briefly, thus locking the SED. This is, however, not the case. Since physical reset buttons are mostly not available today, we induced hardware resets by connecting two pins on the motherboard. If BIOS settings disallowed us to boot from external devices, we could insert a second hard disk where that was possible (i.e., in desktops, not in laptops). Nevertheless, this attack depends on – for an attacker often unpredictable – hardware and BIOS settings like BIOS boot passwords.

On the few of our test systems where the drive gets locked during reboot, we conjecture that it gets explicitly locked in software. Software-locking sounds like a good feature for all future PCs to prevent reboot attacks, but we were able to successfully circumvent this mechanism: After inducing a reboot, we unplugged the data cable briefly before shutdown, and replugged it immediately after the machine came up again. This attack requires precise timing, but it prevents the disk from getting locked and enables us to access it without the password. The reason is that an SED does not get locked when it is not present at the time that the ATA lock command is sent. We believe that this is a general flaw, because we were able to attack different models from different vendors with the same approach.

5. SED (IN)SECURITY SURVEY

We now present our results from applying the attacks that we described above in practice. We first discuss general considerations and then report on individual systems sorted by manufacturers.

5.1 General Considerations and Test Set

An early insight from our experiments was that system security depends on the motherboard and BIOS version, and not on the particular SED model. We did not find different behaviors among different SEDs, and so we did extensive testing with two drives: (1) the Intel SSD 320 with an Intel-specific controller, and (2) the new Intel SSD 520 with SandForce controller SF-2281. SEDs from other vendors, e.g., the Kingston KC100, are based on the SF-2281 controller, too, and are therefore likely to exhibit the exactly same behavior.

On the system side, we investigated a set of 12 computer systems from three different vendors: four workstations from Fujitsu, four laptops from Lenovo, and four laptops from Dell. All test systems are from 2010 or later, with the exception of a ThinkPad R60e from 2008. The choice of systems was determined by the type of systems available in our department, but represents typical models that are generally available on the market, too. We probed more laptops than desktops, because desktops with ATA security are still hard to find. For example, we purchased recent desktop boards from ASUS (P8P67 LE) and ASRock (Z77 Pro4) and updated their EFI to the latest available version, but ATA security for setting the passwords was not supported.

5.2 Fujitsu Workstations

System	Motherboard	BIOS
Esprimo P9900	D2912-A1	Phoenix V6.0R1.20.2912
Esprimo P900	D3062-A1	AMI V4.6.4.0R1.5.0
Esprimo P900	D3062-A1	AMI V4.6.4.0R1.14.0
Celsius R570-2	D2628-C1	Phoenix V6.0R1.21.2628

Fujitsu workstations were the targets for our evil maid attacks. They are ideally suited for this kind of attack, because they display a consistent, text-based password prompt. On an Esprimo P900, for example, the legal password prompt is displayed 11 seconds after power-up, while our fake prompt is displayed after 16 seconds. We find this delay acceptable and not too suspicious for ordinary users.

More interesting than evil maid attacks, however, are attacks against running machines. Cold boot attacks, for example, are not defeated by Fujitsu boards because a BIOS setting for "ATA password on reboot" is missing. Hot plug attacks are not defeated on Fujitsu PCs, too, because the PC

chassis can be opened at run time, such that SATA cables can be replugged into an analysis machine. These attacks can be deployed against Fujitsu PCs in S3, too, because BIOS settings for "ATA password on S3 wakeup" are missing.

- Fujitsu Esprimo P9900: The P9900 even *saves* the ATA password in NVRAM after it has been set, and never prompts for it again, neither on boot nor after we completely removed the SED. The SED gets locked when it is cut from power, but it always gets automatically unlocked. The P9900 renders evil maid attacks superfluous.

- Fujitsu Esprimo P900/R1.5.: The P900 is more secure than its predecessor P9900, because a password must be entered on boot and on S4 wakeup. Apart from that, the P900 is equally vulnerable, i.e., it is vulnerable to hot plug and cold boot attacks.

- Fujitsu Esprimo P900/R1.14.: This PC is almost identical to the previous one. The reason that we list it, is that Fujitsu added a BIOS setting to store the ATA password inside NVRAM, thus eliminating the need to enter it on boot, effectively behaving like the P9900. However, even though we do *not* enable this option, the password is stored in NVRAM. As a consequence, we were able to circumvent the password as follows: Once the prompt appears, we hit `Enter` and `F2` repeatedly, and the BIOS opens without ATA password. Inside BIOS we can disable "ATA password on boot" and then reboot. From there on, the disk gets automatically unlocked – without requiring the attacker to know the password.

- Fujitsu Celsius R570.: This PC behaves similar to the Esprimo P900/R.1.5, meaning that the password is required on boot. The ACPI states S3 and S4 did not work for us, but we suppose that the password is needed for S4 wakeup, but not for S3 wakeup.

5.3 Lenovo Laptops

System	Motherboard	BIOS
ThinkPad R60e	0657CTO	7EETC6WW Rev. 2.22
ThinkPad T520	4242PT2	8AET52WW Rev. 1.32
ThinkPad x201	3323DBG	6QET52WW Rev. 1.34
ThinkPad x220i	4290G53	8DET54WW Rev. 1.24

Contrary to Fujitsu PCs, DMA attacks can be deployed against all Lenovo ThinkPads since each ThinkPad has a DMA port which is suitable for hot-plugging (FireWire, ExpressCard, or PCMCIA). This compensates the fact that hot plug attacks are not applicable, because Lenovo detects the disconnection of SED power connectors. Attacks are equally effective against running machines and machines in S3, because options for "ATA password on S3 wakeup" are missing.

- Lenovo ThinkPad R60e: The R60e has a PCMCIA slot and is therefore vulnerable to FireWire attacks. Furthermore, it is vulnerable to cold boot attacks, and it is the only Lenovo ThinkPad that is vulnerable to hot plug attacks. Since SED removal detections during S3 are missing, an attacker can put the R60e to sleep and install SATA extension cables.

- Lenovo ThinkPad T520: The T520 prompts for ATA passwords on reboot and detects if SEDs are unplugged during S3, defeating both cold boot and hot plug attacks. However, this laptop comes with two DMA ports and does not require ATA passwords on S3 wakeup, thus enabling DMA attacks.

- Lenovo ThinkPad x201: The x201 behaves similar to the T520, with the difference that it does not shut down on SED removal detection, but that it displays a prompt to re-enter the password. The x201 has a DMA port, namely ExpressCard, for FireWire attacks.

- Lenovo ThinkPad x220i: The x220i behaves similar to the x201 and x220i. DMA attacks are possible due to an ExpressCard interface, cold boot attacks are not possible as well as hot plug attacks.

5.4 Dell Laptops

System	Motherboard	BIOS
Vostro 3300	030DMJ-A10	Dell Inc. A10Rev.1.2
Precision M4600	O8V9YG-A00	Dell Inc. A05Rev.4.6
Latitude 2120	0YY3FH	Dell Inc. A01Rev.1.0
Latitude XT3	067RKH-A00	Dell Inc. A00Rev.4.6

Whereas Lenovo ThinkPads have a more or less consistent configuration, there is a greater variety among Dell laptops. Two of the weakest laptops in our test set (Precision M4600, and Latitude XT3) as well as the most secure laptop (Latitude 2120) are among them.

- Dell Vostro 3300: The Vostro 3300 prompts for ATA passwords on boot, reboot, and S4 wakeup, but not on S3 wakeup, and does not detect SED removals during S3. It is not vulnerable to cold boot, but to hot plug attacks. Furthermore, DMA attacks are possible due to an ExpressCard slot.

- Dell Precision M4600: The Precision M4600 has two DMA ports: FireWire and ExpressCard. ATA passwords are only required on boot, not on reboot, and SED removals are not detected. Hence, it is one of the weakest laptops, allowing for all attacks: FireWire attacks, cold boot attacks, and hot plug attacks.

- Dell Latitude 2120: The Latitude 2120 netbook is the strongest machine in our test set, because it (1) has no DMA port, (2) requires ATA passwords on boot, reboot, and S4 wakeup, and (3) requires ATA passwords even on S3 wakeup. Neglecting the risk of evil maid attacks, we were not able to attack this machine.

- Dell Latitude XT3: The Latitude XT3 tablet has a built-in FireWire port as well as an ExpressCard slot for DMA attacks. Furthermore, ATA passwords are only required on boot and S4 wakeup, and SED removals are not detected, thus enabling both cold boot and hot plug attacks.

5.5 Summary of Results

Our results are summarized in Fig. 1, giving an overview about system-specific behaviors per target. Note that during our tests, we enabled the strongest BIOS configuration *which is possible*. For example, if settings for "ATA password on reboot" or "ATA password on wakeup" exist, we enabled those (but default configurations might be different).

		Fujitsu				Lenovo				Dell			
		P9900	P900	P900	R570	R60e	T520	x201	x220i	3300	M4600	2120	XT3
ATA-PW	on boot		X		X	X	X	X	X	X	X	X	X
	soft reboot						X		X	X		X	
	hard reset						X		X	X		X	
	S3 wakeup											X	
	S4 wakeup		X		X	X	X	X	X	X	X	X	X
DMA	FireWire						X				X		X
	ExpressCard						X	X	X	X	X		X
	PCMCIA					X							
S3 removal detection							X	X	X				

Figure 1: Summary about relevant behaviors and properties per test system. An *X* indicates that a certain behavior or property is available on the system. ATA passwords and S3 removal detections are desirable properties from a security point of view, i.e., an *X* makes the system more secure. DMA interfaces, on the other hand, make systems more insecure and consequently, the absence of an *X* increases security.

The first block in Fig. 1 indicates when ATA passwords are required to access a system. With the exception of two Fujitsu PCs, which store the passwords in NVRAM permanently, this is always the case on boot and on S4 wakeup. Only four out of twelve systems prompt for ATA passwords on reboot or reset. Even worse, only one out of twelve systems prompts for ATA passwords on S3 wakeup. The second block in Fig. 1 indicates whether hot-pluggable DMA ports are present. Seven out of eight laptops have such an interface, but none of the desktop systems. With the exception of the R60e, which has an older PCMCIA port, laptops usually have ExpressCard slots or additionally a native FireWire port. The last row of Fig. 1 indicates whether a laptop detects the removal of SED power connectors during S3. This feature seems to be Lenovo-specific. Three out of four ThinkPads have this feature, but none of the Dell or Fujitsu systems.

6. PRACTICAL ATTACK GUIDELINES

Assume that an adversary has gained physical access to a system. We now provide general guidelines that point to the most promising attack for each system, and we applied these guidelines to our own test set. Fig. 2 illustrates a decision graph leading to the most suitable attack per system (note that the graph does not consider *all* possible attacks per system). If a machine is switched off, evil maid attacks are often the only option, but if a machine is running or in standby mode, several alternatives are possible. We therefore suggest the following priority order for attacks.

- Hot plug attack: This is the most generic attack as it is never defeated by the construction type of desktop PCs. The situation is different for laptops, but we suggest replug attacks as the first option for laptops, too, whenever this is possible (meaning whenever SATA extension cables can be installed).

- DMA-based attack: This is the second most generic attack, because it can be deployed against all systems with hot-pluggable DMA interfaces, i.e., against most laptops. However, for many desktop PCs hot-pluggable DMA ports are not available. Broadly speaking, DMA attacks are most promising against laptops, and hot plug attacks are most promising against desktops.

- Cold boot attack: This attack depends on additional BIOS settings like the "supervisor password" or "boot device order" which are not foreseeable by attackers.

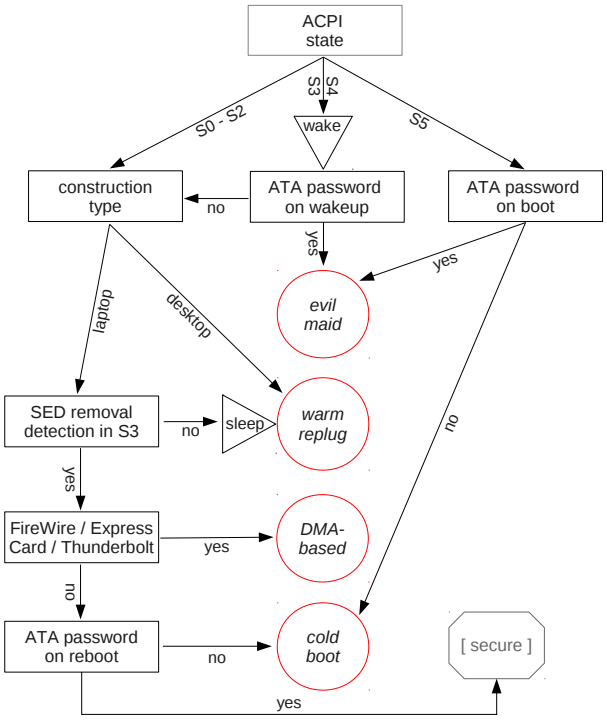

Figure 2: Decision graph for attacking SEDs.

So rebooting a PC with a live system from USB drive should only be tried if previous attacks fail.

- Evil maid attack: Evil maid attacks are always possible, but they should be considered as final option because they require the user to take action. We consider evil maid attacks only if the PC was switched off or in standby, and if the user expects an ATA password prompt.

The last sink of the decision graph in Fig. 2 is labeled with "secure", which means that we were not able to attack such systems. We applied the guidelines to all of our test systems and got an ordered list of attacks per system and per ACPI state, as listed in Fig. 3. If several attacks are possible, the best alternative is listed first. For example, the Dell

FUJITSU	Esprimo P9900	Esprimo P900	Esprimo P900-2	Celsius R570-2
ACPI S0 - S3	replug/reboot	replug/reboot	replug/reboot	replug/reboot
ACPI S4 - S5	boot	evil maid	boot*/evil maid	evil maid
LENOVO	ThinkPad R60e	ThinkPad T520	ThinkPad x201	ThinkPad x220i
ACPI S0 - S3	replug/DMA/reboot	DMA	DMA/reboot	DMA
ACPI S4 - S5	evil maid	evil maid	evil maid	evil maid
DELL	Vostro 3300	Precision M4600	Latitude 2120	Latitude XT3
ACPI S0 - S3	replug/DMA	replug/DMA/reboot	–**	replug/DMA/reboot
ACPI S4 - S5	evil maid	evil maid	evil maid	evil maid
*) with F2 plus Enter combo		**) secure; evil maid attacks can be tried		

Figure 3: Possible attacks against hardware-based FDE per machine/ACPI-state; most suitable attacks are listed first. "Replug" denotes the threat of hot plug attacks; "reboot" denotes a variant of the cold boot attack; "DMA" stands for DMA-based attacks; and "evil maid" denotes our variant of the evil maid attack.

Precision M4600 is vulnerable to SATA replug attacks, to DMA attacks, and to system reboots from thumb drive (cold boot), if running (S0) or suspended to RAM (S3). If suspended to disk (S4) or switched off (S5), evil maid attacks are the only option.

As you can see in Fig. 3, the Dell Latitude 2120 is the most secure system of our test set. This is not only because it has no DMA interface, but also because it is the only system that requires ATA passwords to be entered manually on S3 wakeup. We consider this the best solution for S3 because it renders Lenovo's measure to detect SED removals redundant while being more secure.

7. CONCLUSIONS

Today's computer security requires several layers of protection, including drive protection against physical loss, theft, and seizure. In the event that a computer is lost without drive encryption, attackers can gain full access to the data easily. SEDs are an increasingly popular method to protect against these threats but, as we have shown, they do not provide complete security either.

7.1 Summary

Self-encrypting disks are believed to be more efficient and more user-friendly than software-based FDE, but also to be more secure. While the performance and usability of SEDs are beyond argument, we investigated the security aspect thoroughly – an aspect that was widely misunderstood in the literature before. It was widely believed that SEDs improve security against physical access attacks, but this is generally wrong. We have practically demonstrated that SEDs are vulnerable to DMA and evil maid attacks, to variants of the cold boot attack, and, most notably, to a new class of attacks that we call hot plug attack.

As we have also shown, software-based FDE can be more secure than hardware-based FDE. For example, running desktop PCs with SEDs can be subverted by replugging the SATA cable – an attack which is simpler than comparable cold boot attacks against software-based FDE. Only three of our test systems were more secure with hardware-based FDE than with software-based FDE. The other systems were either equally vulnerable in both setups or even more insecure with hardware-based FDE. The latter is particularly the case when hot plug attacks are possible, which was the case for eight out of twelve systems.

7.2 Countermeasures

The most effective countermeasure is to leave PCs only after power-off and to cease the use of PCs that have been compromised. However, these countermeasures are inconvenient, if not impossible, and directly lead to the question of "physical protection". But physical protection is outside the scope of this writing and mostly supersedes drive encryption. Instead, we identify four layers that may allow to increase SED security in future: users, operating systems, BIOS/motherboards, and SEDs themselves.

- *User layer:* Practical countermeasures that can be taken by users are strong BIOS settings, including a restrictive boot order as well as a boot password which is different to the disk password. Users may have to enter two different passwords during power-up, but cold boot attacks would be defeated.
- *OS layer:* To defeat FireWire attacks, PCs without DMA ports can be bought, but this measure might be unreasonable for many users. Instead, DMA attacks must be defeated on OS layer. With the virtualization technology for directed I/O, it is well-known that DMA transfers can be filtered by means of the IOMMU.
- *BIOS/motherboard layer:* BIOS vendors can take action to secure hardware-based FDE since SATA passwords must never be stored in RAM or NVRAM. When an SED loses power, it must never get unlocked automatically and the user must *always* be prompted to re-enter his or her password manually. It is implemented like this only for one of our test systems, namely the Dell Latitude 2120.
- *SED layer:* To defeat hot plug attacks, a novel type of hardware-sided locks must be introduced that are sensitive to the SATA cable connectivity. Today, SEDs get locked only when power is cut, and S3 removal detections are based on the power connector. Contrary to that, new locks must be based on the *data* connector. Another countermeasure in this direction is to connect SEDs in a way that power and data is transmitted over the same carrier, as it is the case for RAM and PCIe devices. There are already SSDs that get connected via PCIe for performance, like Intel's SSD 910 series. Such SSDs, if they were self-encrypting, would be secure against hot plug attacks.

7.3 Future Work

To date, we have shown that hot plug attacks work against non-Opal SEDs, in particular against the Intel SSD 320 and 520. But the TCG states that Opal is "built to protect the confidentiality of stored user data against unauthorized access once it leaves the owner's control (involving a power cycle and subsequent deauthentication)" [28]. In the brackets it reads that the TCG explicitly excludes a kind of hot plug attacks, which lets us believe that these attacks may succeed against Opal-compliant SEDs as well.

8. REFERENCES

[1] E.-O. Blass and W. Robertson. TRESOR-HUNT: Attacking CPU-Bound Encryption. In *2012 Annual Computer Applications Conference*, Orlando, Florida, Dec. 2012. Northeastern University, College of Computer and Information Science, ACSAC 28.

[2] B. Böck. *Firewire-based Physical Security Attacks on Windows 7, EFS and BitLocker*. Secure Business Austria Research Lab, Aug. 2009.

[3] A. Boileau. Hit by a Bus: Physical Access Attacks with Firewire. In *Proceedings of Ruxcon '06*, Sydney, Australia, Sept. 2006.

[4] B. Bosen. *FDE Performance Comparison: Hardware Versus Software Full Drive Encryption*. Trusted Strategies LLC, Jan. 2010.

[5] Break & Enter: Improving security by breaking it. Adventures with Daisy in Thunderbolt-DMA-land: Hacking Macs through the Thunderbolt interface. http://www.breaknenter.org/2012/02/adventures-with-daisy-in-thunderbolt-dma-land-hacking-macs-through-the-thunderbolt-interface/.

[6] Break & Enter: Improving security by breaking it. Inception. www.breaknenter.org/projects/inception/, Oct. 2011.

[7] Brian D. Carrier and Eugene H. Spafford. Getting Physical with the Digital Investigation Process. *IJDE*, 2(2), 2003.

[8] Carbone and Bean and Salois. An in-depth analysis of the cold boot attack. Technical report, DRDC Valcartier, Defence Research and Development, Canada, Jan. 2011. Technical Memorandum.

[9] T. Coughlin. *Solid Security: The Rise of Self-Encrypting Solid State Drives*. SNIA (Sold State Storage Initiative), 2011.

[10] M. Dornseif, M. Becher, and C. N. Klein. FireWire - All your memory are belong to us. In *Proceedings of the Annual CanSecWest Applied Security Conference*, Vancouver, British Columbia, Canada, 2005. Laboratory for Dependable Distributed Systems, RWTH Aachen University.

[11] P. Gutmann. Data Remanence in Semiconductor Devices. In *Proceedings of the 10th USENIX Security Symposium*, Washington, D.C., Aug. 2001. USENIX Association.

[12] J. A. Halderman, S. D. Schoen, N. Heninger, W. Clarkson, W. Paul, J. A. Calandrino, A. J. Feldman, J. Appelbaum, and E. W. Felten. Lest We Remember: Cold Boot Attacks on Encryptions Keys. In *Proceedings of the 17th USENIX Security Symposium*, pages 45–60, San Jose, CA, Aug. 2008. Princeton University, USENIX Association.

[13] D. Hulton. Cardbus Bus-Mastering: 0wning the Laptop. In *Proceedings of ShmooCon '06*, Washington DC, USA, Jan. 2006.

[14] Intel Corporation. *Data Security Features in Intel Solid-State Drive 320 Series*, 2011. Application Note.

[15] Intel Corporation. *Data Security Features in the Intel Solid-State Drive 520 Series*, 2012. Technology Brief Non-Volatile Memory Storage Solutions from Intel.

[16] Joanna Rutkowska. Evil Maid goes after TrueCrypt. theinvisiblethings. blogspot.com/2009/10/evil-maid-goes-after-truecrypt.html, Oct. 2009. The Invisible Things Lab.

[17] Joanna Rutkowska. AntiEvilMaid. theinvisiblethings. blogspot.de/2011/09/anti-evil-maid.html, Sept. 2011. The Invisible Things Lab.

[18] T. Müller, F. Freiling, and A. Dewald. TRESOR Runs Encryption Securely Outside RAM. In *20th USENIX Security Symposium*, San Francisco, California, Aug. 2011. University of Erlangen-Nuremberg, USENIX Association.

[19] S. Nerz, B. Schlömer, M. Weisband, R. Brosig, W. Schumacher, M. Schrade, and G. Thürmer. Police confiscate German Pirate Party servers. http://vorstand.piratenpartei.de/2011/05/20/polizei-beschlagnahmt-server-der-piratenpartei-deutschland/.

[20] Patrick Simmons. Security Through Amnesia: A Software-Based Solution to the Cold Boot Attack on Disk Encryption. In *Annual Computer Security Applications Conference (ACSAC)*, Orlando, Florida USA, Dec. 2011. University of Illinois at Urbana-Champaign, ACM.

[21] Peter Kleissner. Stoned Bootkit. www.blackhat.com/presentations/bh-usa-09/kleissner/BHUSA09-Kleissner-Stoned Bootkit-Slides.pdf, July 2009. Black Hat, USA.

[22] Ponemon Institute. *Airport Insecurity: The Case of Lost & Missing Laptops*. Executive Summary, U.S. Research, June 2008.

[23] Ponemon Institute. *2010 Annual Study: U.S. Enterprise Encryption Trends*. Nov. 2010.

[24] Ponemon Institute Research Report. *Perceptions about Self-Encrypting Drives: A Study of IT Practitioners*. May 2011.

[25] Seagate Technology LLC. *Can Your Computer Keep a Secret?: Why All Laptop Data Protection Methods Are NOT Created Equal*. Technology Paper, 2007.

[26] SECUDE. *US Full Disk Encryption 2011 Survey*. Research SECUDE AG, Switzerland, 2012.

[27] C. Stevens. *AT Attachment 8 - ATA/ATAPI Command Set (ATA8-ACS)*. Working Draft Project Americaion National Standard, Revision 6 edition, June 2008.

[28] Trusted Computing Group, Incorporated. *TCG Storage Security Subsystem Class (SSC): Opal*, Revision 1.0 edition, Jan. 2009.

[29] S. Türpe, A. Poller, J. Steffan, J.-P. Stotz, and J. Trukenmüller. Attacking the BitLocker Boot Process. In L. Chen, C. J. Mitchell, and A. Martin, editors, *Trusted Computing Second International Conference TRUST*, volume 5471, pages 183–196, Oxford, UK, Apr. 2009. Fraunhofer Institute for Secure Information Technology (SIT), Springer.