draft of 5-11-2018

Self-encrypting deception: weaknesses in the encryption of solid state drives (SSDs)

Carlo Meijer
Radboud University, the Netherlands
C.Meijer@cs.ru.nl

Bernard van Gastel
Radboud University, the Netherlands
Open University of the Netherlands
Bernard.vanGastel@{ru.nl,ou.nl}

***Abstract*—We have analyzed the hardware full-disk encryption of several SSDs by reverse engineering their firmware. In theory, the security guarantees offered by hardware encryption are similar to or better than software implementations. In reality, we found that many hardware implementations have critical security weaknesses, for many models allowing for complete recovery of the data without knowledge of any secret.**

BitLocker, the encryption software built into Microsoft Windows will rely exclusively on hardware full-disk encryption if the SSD advertises supported for it. Thus, for these drives, data protected by BitLocker is also compromised.

This challenges the view that hardware encryption is preferable over software encryption. We conclude that one should not rely solely on hardware encryption offered by SSDs.

I. Introduction

In recent years, protection of sensitive data has received increased attention. Protection of digital data has become a necessity, certainly in the light of new European Data Protection Regulation. Technically, encryption is the go to protection mechanism; it may be implemented in software or hardware (or both). It can be applied on the level of individual files, or the entire drive, which is called *full-disk encryption*. Full-disk encryption is often the solution of choice as it takes away concerns of sensitive data leakage through, for example, temporary files, page files and caches. Several software solutions for full-disk encryption exist, and modern operating systems typically integrate it as a feature. However, purely software-based encryption has inherent weaknesses, such as the encryption key being present in RAM at all times and performance drawbacks.

In an attempt to address these weaknesses, hardware full-disk encryption is often proposed; the encryption is performed within the drive itself, thereby confining the encryption key exclusively to the drive. Typically, the encryption itself is performed by a dedicated AES co-processor, whereas the software on the drive (firmware) takes care of the key management. It is often regarded as the successor of software full-disk encryption. Full-disk encryption software, especially those integrated in modern operating systems, may autonomously decide to rely solely on hardware encryption in case it is supported by the storage device (via the TCG Opal standard). In case the decision is made to rely on hardware encryption, software encryption is disabled. In fact, BitLocker, the full-disk encryption software built into Microsoft Windows, switches off software encryption and completely relies on hardware encryption by default if the drive advertises support.

Contribution. This paper evaluates both internal and external storage devices, from multiple vendors, adhering to standards for secure storage. The vendors combined produce close to half of the SSDs currently sold. An overview is given of possible flaws that apply in particular to hardware-based full-disk encryption (Section V), and a methodology is provided for the analysis (Section IV). We have analyzed firmwares from different SSD models offering hardware encryption, focusing on these flaws (see Section VI and Table I). The analysis uncovers a pattern of critical issues across vendors. For multiple models, it is possible to bypass the encryption entirely, allowing for a complete recovery of the data without any knowledge of passwords or keys. The situation is worsened by the delegation of encryption to the drive by BitLocker. Due to the default policy, many BitLocker users are unintentionally using hardware encryption, exposing them to the same threats. As such, we should reconsider whether hardware encryption is a true successor to its software counterpart, and whether the established standards actually promote sound implementations.

Related work. At OHM in 2013, Domburg demonstrated the possibility of debugging a hard drive through JTAG and created possibly the first publicly demonstrated hard drive firmware rootkit [4]. Domburg's work has inspired more research around anti-forensics such as [20], [7]. Leaked documents indicate that even the NSA is using these techniques [11]. Besides, proprietary cryptographic systems have often shown to be much weaker in practice than standardized publicly available alternatives once implementation details are uncovered [18]. Within the scope of storage devices with integrated hardware encryption, serious vulnerabilities have also previously been identified in external drives using proprietary protection schemes. An example is the external Secustick, which unlocks by simply sending a command (not containing a password) [5]. Another example is the Western Digital MyPassport family of external drives, which suffers from RAM leakage, weak key attacks, or even hardcoded keys [2]. However these findings are isolated incidents limited to proprietary solutions, and neither consider implementations of established standards for secure storage nor consider these