# Flaws and Solutions: Disk Encryption

Rob Hornbuckle

East Carolina University

ICTN6823 Summer 2014

# Abstract

Providers of encryption software do not protect against side channel attacks, leaving organizations vulnerable to exploitation. For those companies who have a business justification to employ methods to virtually eliminate the weaknesses within encryption, several options are available. Of note, are methods used by Trevisor, Cryptkeeper, and TPM. Also note that to completely remove the vulnerabilities inherent with Cryptkeeper, it needs to be developed further using concepts from Trevisor. These solutions in their current form are cost prohibitive from an implementation standpoint for most companies.

Cryptologists often recommend, with hardware-based attacks, if an attacker has ‘worked’ with an organization’s hardware, that organization should stop using it for sensitive data, maintaining that it is impossible to prevent attacks once you have been discovered to this level. This often leads to an argument between physical security and encryption. It’s often recommended, to insure safety, to lock away a laptop away at all times. Often wondered, and rightly so, is if the laptop is locked away at all times, why would encryption be needed?

To get a better sense of why organizations care about encryption, consider the compelling reasons to encrypt endpoints. The most notable reason to encrypt is confidentiality. In other words, encrypt endpoints to keep company data out of the hands of those who shouldn’t have access. Of substantial importance as well is controlling the secrecy and privacy of personal employee data. In addition, encryption is essential to make sure the data on the endpoint is un-tampered with by third parties, insuring its integrity. Relatedly, encryption assures the trustworthiness of the endpoint data. Some industries go as far as to require encryption such as organizations within the medical or legal fields. Lastly, encryption allows for maintaining access control in the face of an attacker gaining physical access.

Assume for these reasons encryption of endpoint machines has been created and implemented. Even with a great encryption system in place including adequate random number generation, block cipher modes for operation to increase complexity, and key derivation from passwords, still ignored are side channel attacks. While encryption can be one component to keep a network secure, consider attacks on the user or even attacks on the hardware itself, referred to as a side channel attack. Despite near bulletproof encryption, if a hacker can penetrate the network through a side channel, the best encryption is useless.

Perhaps the more curious among us would wonder how these attacks could be pulled off and why we don't hear about them more often. The first thing worthy of note is that the cryptography companies don't consider side channel attacks to be included in the realm of things they protect against. How convenient, as they can claim 100% effectiveness in their area despite their product being truly 100% secure. Consider the conversation between Joanna Rutkowska, a security researcher, and the publishers of the software True Crypt. True Crypt stated in no uncertain terms that they don't care.

> TC (…) We never consider the feasibility of hardware attacks; we simply have to assume the worst. After an attacker has 'worked' with your hardware, you have to stop using it for sensitive data. It is impossible for TPM to prevent hardware attacks (…).
>
> JR: And how can you determine that the attacker have or have not 'worked' with your hardware? Do you carry your laptop with you all the time?
>
> TC: Given the scope of our product, how the user ensures physical security is not our problem. Anyway to answer your question (…) you could use e.g. a proper safety case with a proper lock or (…) store it in a good strongbox.
>
> JR: If I could arrange for a proper lock or an impenetrable strongbox, then why in the world should I need encryption? Selifonov, D. (2013, August 2)

To further illustrate this point, the documentation provided by TrueCrypt on the product includes the following, 'TrueCrypt does not secure any data on a computer if an attacker has physical access to the computer before or while TrueCrypt is running on it.' This highlights the issue that side channel attacks must be addressed as cryptologists will not address these issues.

In order to protect against side channel attacks, consider all the various known attacks that can be done on an encrypted machine to bypass the encryption with physical access. Known attacks include four types of attacks: non-intrusive, hacking, mechanical, and soldering attacks. Non-intrusive attacks include things such as stealing a key through social engineering, guessing a key, video taping a user logging in, recording the sounds of keystrokes to pull a key, and direct memory access. Hacking attacks include things such as using a virus to pull data out of RAM containing the key, reprogramming bios to bypass the encryption check, a compromised boot loader (evil maid attack), and direct memory access attacks. Mechanical attacks include things such as replacing the TPM chip on a computer, accessing the RAM chips directly, and cold booting. Soldering attacks include things such as dissecting the TPM chip under an electron microscope, and adding a new component to the motherboard to change up the boot process. To better understand, consider how a few of these attacks work.

A compromised boot loader is basically a rewrite of the boot loader to put up a screen pre-boot to ask for the users credentials before any other security measures can come into play. The information that is entered is then stored and retrieved later or sent out when the OS fully boots up. The attack is referred to as an evil maid attack because, for instance, a hotel maid could load the software onto a shut down computer when an employee is away from the hotel and then retrieve it the next night before checkout now having full access to the machine. As an alternative, an attacker could watch the entering of credentials and record them.

A cold boot attack is pulled off by drastically lowering the temperature level of the RAM chips in a computer at shutdown and then transferring them to a reading system to pull off the data. With the way encryption works, an encryption key will be loaded in this memory, in the clear, and can be pulled off this way. This works because the lower the temperature that RAM is, the slower it will lose the data stored on it after a loss of power.

A direct memory access attack is pulled off using a PCI connected device. The device is reconfigured to use the direct RAM access that is given to PCI devices and is used to pull down and store all the information in RAM. This information can then be retrieved or stored on the PCI device itself. In fact, there are actually computer forensic devices designed for just this purpose using these exact principles.

Clearly, side channel attacks must be addressed in order to maintain a secure network. Note that most side channel attacks rely on the encryption key stored in RAM, making a compelling argument for storing the key somewhere else. A grad student, Tilo Muller, wrote a thesis on this subject and has come up with a viable solution. He suggests that the CPU itself would present a viable option, specifically the debug registers of the chip. The chip being centrally located in the architecture will help with performance. With this space utilized for encryption key storage, most of the side channel attacks previously mentioned would be stopped. Muller called his solution TRESOR. Following this theory, TRESOR utilized the debug space on an X86 processor to store encryption keys and performs at much the same levels as a standard RAM storage model. When the solution was put to use on 64 bit processors, the performance was slightly better using TRESOR than with the standard RAM approach.

With a key that can be stored on the CPU, there is still the issue that a key of some kind needs to be stored in RAM to boot properly. To solve this, consider two 128-bit keys that work together to enable encryption. This would provide a much more secure key that never leaves the CPU memory and a second, less secure version that stays on the RAM. The RAM key becomes more secure because it won't function without its counterpart on the CPU, but it can still get through boot up.

However, note that TRESOR by default is vulnerable to direct memory access attacks that can be used to dump out processor memory. To address this vulnerability, consider an IOMMU (input/output memory management unit). Originally developed for server virtualization, IOMMU was used to sandbox PCI devices for purposes such as adding a network card to a virtual machine for operational purposes. Repurposed, this technology can be used to remove PCI devices from having arbitrary access, preventing them from having access to the CPU and being able to pull down the information stored. To shore up the vulnerability within TRESOR, Muller paired it with a hypervisor called Bitvisor, calling it TreVisor. It allows for a single OS to run with full disk encryption and all the protections discussed so far, making for the safest and most secure full disk encryption that is currently available.

While this seems like the perfect solution, RAM is still vulnerable even though the full disk encryption may be safe. Due to the fact that RAM holds active files, SSH/PGP keys, or even just general password data, this can still be a problem. Stepping outside of the stolen hardware model for a moment, the same clear text data storage in RAM issue is the reason why exploits like Heartbleed can function. Encrypting RAM or at least encrypting most of RAM to reduce attack surface is a viable solution for this kind of problem. In fact, Peter Peterson came up with a proof of concept for encrypting RAM and the subsequent product known as Cryptkeeper.

Cryptkeeper functions by taking the RAM space and running it in a split mode. In this mode, about 95% of RAM is encrypted with the rest being in clear mode and used mostly on startup. This drastically reduces attack surface and protects data that is still on RAM when powered up or in sleep mode. The primary security issue with this model though is that the main key for Cryptkeeper to encrypt the RAM is stored in the clear part of the RAM memory. However, this could be combined with the TRESOR processor storage method to keep the key there and remove this vulnerability. However, this capability would need to be developed. The second issue with Cryptkeeper and the reason it was originally scrapped was the performance hit that is taken by encrypting the RAM itself. In benchmark tests, there was approximately a 10% decrease in speed between a non-encrypted chip and it's encrypted counterpart. Back in 2010, when this was first developed, a 10% speed decrease at the core of the system was a killer. However, with today's technology the speed degradation isn't as pronounced. For example, with normal internet usage there is a negligible speed decrease and with intensive PC or graphical usage the speed decrease is in the 5% realm. In order to create a fully secure system, this small speed decrease is within an acceptable range.

Now that a fully secure and detailed out encryption model has been established, safely booting the computer is the major concern. In order to accomplish this, utilize the small, unencrypted section of RAM that is still vulnerable. Minimize the risk of doing so by clearing it just after it's use. In order to begin this process, we need to confirm the integrity of the machine. Insure that the machine hasn't been tampered with prior to our booting it up into the secure state and authenticating to it because that's where it is at the most vulnerable. There are a few options as to how to accomplish this. One option is to initiate some kind of pre-password or pre-key programming onto the machine. For instance, keep a unique picture on a keychain and have the camera of the machine pick up and identify the 'key' before starting the rest of the machines processes to finish boot up. While this is a clunky solution and not that much better then a pre-bios password, it would have the added benefit of preventing people executing separate boot loaders.

Another option is to use the TPM (trusted platform module) chip that is located on most motherboards. The original function of the chip was to secure devices on a machine with cryptographic keys and was defeated pretty soon after being implemented both in functionality and sophistication. However, the chip is still located on most motherboards and even with its less than stellar reputation, it could still help with mitigating vulnerability. The TPM has the ability to measure the boot up sequence. These measurements along with some of the programming options of the TPM help insure the integrity of the machine. The side effect of this is that you can seal data to a particular startup configuration of the machine.

Using the TPM functions, we can perform the following features to help insure the integrity of the computer.  These features limit the unauthorized boot up attempts the computer performs in a row, the number of times a pre-OS password can be attempted, and the number of passwords entered incorrectly.  It can also set a maximum time since the last startup a new startup can be performed, block out a machine startup by time or date, and a distress or self-destruct password can be created.  Lastly, there is basic tamper protection using the original function of the chip. If any of these triggers are set off, the system will wipe all keys and require reauthorization from a master machine in order to start working again.

Even if the methods described are implemented, there are still at least two notable weaknesses in the system. If an attacker can get the master machine or figure out how to spoof it, access can be gained past this first stage of the system. There is also an inherent flaw because the chip is a separate chip on the board and can be individually attacked, with sufficient skill.  To protect against an attacker removing the chip for exploit, consider reducing risk by epoxying the chip to the board.  This insures that the chip is destroyed if it is removed and destroys the keys along with it.  Note that if the TPM chip can be removed, it is vulnerable to replacement or dissection.  Replacement is a highly skilled option and has a lot of risk involved. Dissection would need to be done with an electron microscope but is the safer option. In order to pull it off, access to an electron microscope is required but note that it is cost prohibitive and the process is time consuming, even with a highly skilled operator.  To accomplish dissection, an attacker would use the electron microscope to identify the data and reconstruct it from the images provided, bit by bit.  Also note that the older TPM chips have a software attack problem and have been broken by attacking the LPC bus on the chip directly. The newer Intel based TPM chips do not appear to have the same problem.

Using a combination of methods, an organization can bolster the inherent flaws in encryption by insuring the integrity of the machine, getting out of real mode as fast as possible or even bypassing it all together, and getting into a pre-boot authorization environment as fast as possible to reduce the attack surface. Current products and methods that help to accomplish this are using Trevisor to run the OS and full disk encryption, using Cryptkeeper to encrypt RAM while storing it's key on the CPU, running a custom pre-boot authorization process, and configuring the TPM to help insure the integrity of the machine. While this is a solution to circumvent the inherent flaws in relying on encryption to protect data, it is hardly cost effective. For most small to medium sized companies these techniques would be well above their requirements. Most companies are of the stance that they are secure because they aren't particularly large or well known, that the limited knowledge of their company is protective against attacks. Add further that most of these companies operate under the pretense that their data isn't desirable enough for an attacker to go to all the trouble to get it. These companies focus their efforts on trying to protect against the random acts of theft that may occur and to make sure data isn't a factor in them.

For companies who would like to bolster their security but not to the scale of full implementation of the ideas and products discussed above, a piecemeal implementation could be considered with Trevisor, Cryptkeeper, or even a TPM configuration alone or in combination. However, it would be advised to install them together to avoid the need for collecting the hardware multiple times and to avoid reconfiguration times needed to augment existing implementations when a new component is added.

For those organizations that do care intensely about the security of their data, such as those in government, intellectual property heavy operations, or as time goes on, heavily regulated industries, significant changes to the imaging process of machines must be made in order to implement this model. A controller would need to be setup and administer Trevisor, Cryptkeeper, TPM, and the original full disk encryption software. Also consider that the help desk would need to be trained and incoming calls would subsequently increase because of the new security measures. With the added cost of administration, even under the assumption that the licensing is free, most companies would accept the risk and decide against an implementation of this scale. However, companies are now armed with the information that encryption alone is not foolproof, that it can be circumvented through side channel attacks, and that measures are available to mitigate that risk should there be a business justification to do so.

# References

Bruce, S. (2009, October 23). "Evil Maid" Attacks on Encrypted Hard Drives. *Blog*. Retrieved July 23, 2014, from https://www.schneier.com/blog/archives/2009/10/evil_maid_attac.html

Fisher, D. (2009, October 19). *How to Defeat Full-Disk Encryption in One Minute - The Community's Center for Security*. Retrieved July 23, 2014, from http://threatpost.com/how-defeat-full-disk-encryption-one-minute-101909/72345

* Halderman, A., Schoen, S., Heninger, N., Clarkson, W., Paul, W., Calandrino, J., et al. (2008, February 21). Lest We Remember: Cold Boot Attacks on Encryption Keys. *Princeton CITP News*. Retrieved July 23, 2014, from http://citpsite.s3-website-us-east-1.amazonaws.com/oldsite-htdocs/pub/coldboot.pdf

How Does Xen Work?. (2009, December 1). . Retrieved July 23, 2014, from http://www-archive.xenproject.org/files/Marketing/HowDoesXenWork.pdf

IOMMU. (2014, July 7). *Wikipedia*. Retrieved July 23, 2014, from http://en.wikipedia.org/wiki/IOMMU

* Muller, T., Freiling, F., & Dewald, A. (2010, January 1). TRESOR Runs Encryption Securely Outside RAM. . Retrieved July 23, 2014, from https://www.usenix.org/legacy/event/sec11/tech/full_papers/Muller.pdf

* Muller, T., Taubmann, B., & Freiling, F. (n.d.). TreVisor. . Retrieved July 23, 2014, from http://mirror.robert-marquardt.com/downloads/trevisor.pdf

* Peterson, P. (2010, November 10). Cryptkeeper: Improving security with encrypted RAM. . Retrieved July 23, 2014, from http://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CB0QFjAA&url=http%3A%2F%2Fwww.researchgate.net%2Fpublication%2F224201954_Cryptkeeper_Improving_security_with_encrypted_RAM%2Ffile%2Fd912f50cfdf4b2e20b.pdf&ei=esPPU62QDu7MsQSMxoKwAQ&usg=AFQjCNHQ8VPo9oicRoDMedeirsxGankLfA&sig2=mLKPx_jBom-m08YV3BJvtw&bvm=bv.71667212,d.cWc

Selifonov, D. (2013, August 2). [DEFCON 21] A Password is Not Enough: Why disk encryption is broken and how we might fix it. YouTube. Retrieved July 23, 2014, from https://www.youtube.com/watch?v=EFsoCr589GI

* Shinagawa, T., Eiraku, H., Tanimoto, K., Omote, K., Hasegawa, S., Horie, T., et al. (2009, March 1). BitVisor: A Thin Hypervisor for Enforcing I/O Device Security. . Retrieved July 23, 2014, from http://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=5&sqi=2&ved=0CEgQFjAE&url=http%3A%2F%2Fwww.researchgate.net%2Fpublication%2F221137798_BitVisor_a_t

hin_hypervisor_for_enforcing_io_device_security%2Ffile%2F3deec51fd0c9cece1c.pdf&ei=wRbMU_zNLOr28AHlz4HgDA&usg=AFQjCNHR2z8zYohqP1kvjhbgIr1a2Be-eA&sig2=d2Bcfomjw3Ct540St2KHzg&bvm=bv.71198958,d.b2U

* TrueCrypt User's Guide. (2009, October 21). . Retrieved July 23, 2014, from http://www.mia-net.org/pub/pc/win/crypto/TrueCrypt/TrueCrypt%20User%20Guide.pdf

Trusted Platform Module. (2014, July 13). *Wikipedia*. Retrieved July 23, 2014, from http://en.wikipedia.org/wiki/Trusted_Platform_Module