SwitchBox: Security, Energy, and Performance Tradeoffs Leveraging LFS Behavior in Stream Cipher Based Full Drive Encryption

Anonymous Author(s)

Abstract

Full-drive encryption (FDE) is a primary concern among several in modern systems increasingly backed by solid-state storage. Prior work with stream cipher based FDE demonstrates that, by selecting the fastest stream cipher statically at compile time, we can achieve improved I/O throughput over block cipher based FDE while offering stronger security guarantees. This is ideal when the only optimization target is throughput; however, throughput is not the only primary concern—others frequently include security and energy use. For instance: the cipher with the highest throughput may not provide security guarantees at the desired strength; similarly, the cipher with the strongest security guarantees may not be efficient enough given the current energy budget.

In this paper, we show these competing concerns form a tradeoff space between energy, security, and performance. We further characterize this space and present SwitchBox, a software mechanism to navigate such a space dynamically and at runtime via *cipher scheduling*. This is accomplished by taking advantage of the overwrite-averse "append-mostly" behavior of the underlying solid-state storage to trade throughput or total energy use of the file system for desired security guarantees. We implement SwitchBox on an ARM big.LITTLE mobile processor and test its performance under the popular F2FS LFS. We find that SwitchBox is flexible enough to satisfy a wide range of performance and security constraints. **[TODO: Perhaps a sentence-long general explanation of the use cases and the most interesting result(s) from them?]**

1 Introduction

Todo!

ASPLOS'19, ,

2 Motivation

Todo!

3 SwitchBox System Design

Todo!

4 SwitchBox Implementation

Todo!

5 Evaluation

5.1 Experimental Setup

Todo!

5.2 Experimental Methodology

Todo!

6 Related Work

Todo!

7 Conclusion

Todo!

[0]

References

[1] [n. d.]. Android Open Source Project: Full-Disk Encryption. ([n. d.]). https://source.android.com/security/encryption/full-disk

[2] [n. d.]. RedHat: Device-mapper Resource Page. ([n. d.]). https://www.sourceware.org/dm

[3] 2005. Oracle blog: ZFS End-to-End Data Integrity. (2005). https://blogs.oracle.com/bonwick/zfs-end-to-end-data-integrity

[4] 2008. The XTS-AES Tweakable Block Cipher. (2008). IEEE Std 1619-2007.

[5] 2010. Recommendation for Block Cipher Modes of Operation: The XTS-AES Mode for Confidentiality on Storage Devices. (2010). http://nvlpubs.nist.gov/ NIST Special Publication 800-38E.

[6] 2011. Message Authentication Code Standard ISO/IEC 9797-1:2011. (2011). https://www.iso.org/standard/50375.html

[7] 2012. A block device in userspace. (2012). https://github.com/acozzette/BUSE

[8] 2013. Linux kernel device-mapper crypto target. (2013). https://gitlab.com/cryptsetup/cryptsetup

[9] 2014. TLS Symmetric Crypto. (2014). https://www.imperialviolet.org/2014/02/27/tlssymmetriccrypto.html

[10] 2015. EMBEDDED MULTI-MEDIA CARD (eâĂćMMC), ELECTRICAL STANDARD (5.1). (2015). https://www.jedec.org/standards-documents/results/jesd84-b51

[11] Daniel J. Bernstein. 2005. *The Poly1305-AES message-authentication code.* Technical Report. University of Illinois at Chicago.

[12] Daniel J. Bernstein. 2008. *ChaCha, a variant of Salsa20.* Technical Report. University of Illinois at Chicago.

[13] D. Chakraborty, C. Mancillas-LÃşpez, and P. Sarkar. 2015. STES: A Stream Cipher Based Low Cost Scheme for Securing Stored Data. *IEEE Trans. Comput.* 64, 9 (2015), 2691–2707. https://doi.org/10.1109/TC.2014.2366739

[14] Michael Cornwell. 2012. Anatomy of a Solid-state Drive. *Queue* 10, 10, Article 30 (Oct. 2012), 7 pages. https://doi.org/10.1145/2381996.2385276

[15] Andrew Ferraiuolo, Rui Xu, Danfeng Zhang, Andrew C. Myers, and G. Edward Suh. 2017. Verification of a Practical Hardware Security Architecture Through Static Information Flow Analysis. In *Proceedings of the Twenty-Second International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS 2017, Xi'an, China, April 8-12, 2017*. 555–568. https://doi.org/10.1145/3037697.3037739

[16] Trusted Computing Group. 2008. TCG: Trusted platform module summary. (2008).

[17] Shai Halevi and Phillip Rogaway. 2003. *A Tweakable Enciphering Mode.* Springer Berlin Heidelberg, Berlin, Heidelberg, 482–499. https://doi.org/10.1007/978-3-540-45146-4_28

[18] D. Hein, J. Winter, and A. Fitzek. 2015. Secure Block Device – Secure, Flexible, and Efficient Data Storage for ARM TrustZone Systems. In *2015 IEEE Trustcom/BigDataSE/ISPA*, Vol. 1. 222–229. https://doi.org/10.1109/Trustcom.2015.378

[19] Matthew Hicks, Cynthia Sturton, Samuel T. King, and Jonathan M. Smith. 2015. SPECS: A Lightweight Runtime Mechanism for Protecting Software from Security-Critical Processor Bugs. In *Proceedings of the Twentieth International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS '15, Istanbul, Turkey, March 14-18, 2015*. 517–529. https://doi.org/10.1145/2694344.2694366

[20] Connor Imes, Lars Bergstrom, and Henry Hoffmann. 2016. A Portable Interface for Runtime Energy Monitoring. In *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE 2016)*. ACM, 968–974. https://doi.org/10.1145/2950290.2983956

[21] Darko Kirovski, Milenko Drinić, and Miodrag Potkonjak. 2002. Enabling Trusted Software Integrity. In *Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS X)*. ACM, New York, NY, USA, 108–120. https://doi.org/10.1145/605397.605409

[22] Ryusuke Konishi, Yoshiji Amagai, Koji Sato, Hisashi Hifumi, Seiji Kihara, and Satoshi Moriai. 2006. The Linux Implementation of a Log-structured File System. *SIGOPS Oper. Syst. Rev.* 40, 3 (July 2006), 102–107. https://doi.org/10.1145/1151374.1151375

[23] Changman Lee, Dongho Sim, Jooyoung Hwang, and Sangyeun Cho. 2015. F2FS: A New File System for Flash Storage. In *13th USENIX Conference on File and Storage Technologies (FAST 15)*. USENIX Association, Santa Clara, CA, 273–286. https://www.usenix.org/conference/fast15/technical-sessions/presentation/lee

[24] Xun Li, Vineeth Kashyap, Jason K. Oberg, Mohit Tiwari, Vasanth Ram Rajarathinam, Ryan Kastner, Timothy Sherwood, Ben Hardekopf, and Frederic T. Chong. 2014. Sapper: a language for hardware-level security policy enforcement. In *Architectural Support for Programming Languages and Operating Systems, ASPLOS '14, Salt Lake City, UT, USA, March 1-5, 2014*. 97–112. https://doi.org/10.1145/2541940.2541947

[25] ARM Limited. 2009. ARM security technology: Building a secure system using TrustZone technology. (2009). PRD29-GENC-009492C.

[26] Subhamoy Maitra. 2015. *Chosen IV Cryptanalysis on Reduced Round ChaCha and Salsa.* Technical Report. Applied Statistics Unit, Indian Statistical Institute.

[27] NIST. 2008. Public Comments on the XTS-AES Mode. (2008).

[28] Anil Kumar Reddy, P. Paramasivam, and Prakash Babu Vemula. 2015. Mobile secure data protection using eMMC RPMB partition. In *2015 International Conference on Computing and Network Communications (CoCoNet)*. 946–950. https://doi.org/10.1109/CoCoNet.2015.7411305

[29] Phillip Rogaway. 2004. *Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC.* Technical Report. University of California at Davis.

[30] Phillip Rogaway, Mark Wooding, and Haibin Zhang. 2012. *The Security of Ciphertext Stealing.* Springer Berlin Heidelberg, Berlin, Heidelberg, 180–195. https://doi.org/10.1007/978-3-642-34047-5_11

[31] Mendel Rosenblum and John K. Ousterhout. 1992. The Design and Implementation of a Log-structured File System. *ACM Trans. Comput. Syst.* 10, 1 (Feb. 1992), 26–52. https://doi.org/10.1145/146941.146943

[32] Palash Sarkar. 2009. *Tweakable Enciphering Schemes From Stream Ciphers With IV.* Technical Report. Indian Statistical Institute.

[33] Global Platform Device Technology. 2010. TEE client API specification version 1.0. (2010). GPD_SPE_007.

[34] Mohit Tiwari, Jason Oberg, Xun Li, Jonathan Valamehr, Timothy E. Levin, Ben Hardekopf, Ryan Kastner, Frederic T. Chong, and Timothy Sherwood. 2011. Crafting a usable microkernel, processor, and I/O system with strict and provable information flow security. In *38th International Symposium on Computer Architecture (ISCA 2011), June 4-8, 2011, San Jose, CA, USA*. 189–200. https://doi.org/10.1145/2000064.2000087

[35] Jonathan Valamehr, Melissa Chase, Seny Kamara, Andrew Putnam, Daniel Shumow, Vinod Vaikuntanathan, and Timothy Sherwood. 2012. Inspection resistant memory: Architectural support for security from physical examination. In *39th International Symposium on Computer Architecture (ISCA 2012), June 9-13, 2012, Portland, OR, USA*. 130–141. https://doi.org/10.1109/ISCA.2012.6237012

[36] Marten van Dijk, Jonathan Rhodes, Luis F. G. Sarmenta, and Srinivas Devadas. 2007. Offline Untrusted Storage with Immediate Detection of Forking and Replay Attacks. In *Proceedings of the 2007 ACM Workshop on Scalable Trusted Computing (STC '07)*. ACM, New York, NY, USA, 41–48. https://doi.org/10.1145/1314354.1314364

[37] Peng Wang, Dengguo Feng, and Wenling Wu. 2005. *HCTR: A Variable-Input-Length Enciphering Mode.* Springer Berlin Heidelberg, Berlin, Heidelberg, 175–188. https://doi.org/10.1007/11599548_15

[38] Rui Zhang, Natalie Stanley, Christopher Griggs, Andrew Chi, and Cynthia Sturton. 2017. Identifying Security Critical Properties for the Dynamic Verification of a Processor. In *Proceedings of the Twenty-Second*

International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS 2017, Xi'an, China, April 8-12, 2017. 541–554. https://doi.org/10.1145/3037697.3037734