Hycrypt: More Performant Full Disk Encryption with Chacha and Poly1305

Bernard Dickens
University of Chicago bd3@cs.uchicago.edu

Ariel Feldman
University of Chicago arielfeldman@cs.uchicago.edu

Henry Hoffmann
University of Chicago hankhoffmann@cs.uchicago.edu

Abstract

Have Your Cake and eat it too
Summarize problem, introduction/contributions, implementation, conclusion in 250? interesting words or less.

1 Introduction

Somewhat concise introduction like in the Chacha paper. Is this worth it? Briefly address project meaningfulness. Status quo disk encryption has a non-trivial cost. Enumerate any potential tradeoffs. Describe extra benefits of a Chacha-LFS construction (i.e. integrity checking, simpler design) over XTS.

Caveats, major limitations, and how they're handled.

- Here we summarize the main contributions (as with the brief introduction) with a focus on the justification for the project's existence. Tease apart and enumerate any other contributions.
- Argue that this is a meaningful project by showing that encryption has a non-trivial cost [use initial FDE vs NFDE experiments]
- Chart FDE vs NFDE for reads/writes of random data (perhaps dd tests, perhaps random data file tests) assists
- Limitation: hardware accelerated AES makes the disparity between NFDE and FDE disappear; countered by HAAES not being available very widly on many mobile devices, especially embedded
- Show Chacha as a stream cipher to be faster than AES in a stream cipher-ey mode (CTR or GSM to compare with integrity checking). Need to show that these modes of AES are always? faster than the very slow double-keyed XTS construction. This would establish that there is some slack to be played with between AES-XTS and other modes and something like a chacha.
- Argue (perhaps in the Salsa20 section below) that making a stream cipher work for FDE in this instance is non-trivial and comes with costs. Characterize. "We save X at the cost of Y". DE is the problem. The solution is Z. It costs Y. If willing to pay.
- Need to tell what these costs are, if and how they can be minimized, and what trade-offs they constitute.

2 Related Works

(broken off from the introduction)

2.1 AES-XTS

2.2 Salsa20/Chacha20

3 Hycrypt Full Disk Encryption

3.1 Threat Model

(this will be very similar to the threat model established by the XTS project and writeups)

3.2 Design and Implementation

Describe LFS construction in detail, explain design decisions, preservation of security guarantees, extra benefits (i.e. integrity checking, simpler design) over XTS

4 Experimental Setup

Describe evaluation of Chacha-LFS versus AES-XTS

5 Experimental Evaluation

- Charts showing actual benchmarks of i/o bound and/or i/o heavy applications (i.e. git, several others) under NFDE (control), AES-XTS, and ChaCha-LFS
- Benchmarking suites? TBD

6 Conclusion

7 Acknowledgments

A polite author always includes acknowledgments. Thank everyone, especially those who funded the work.

8 Availability

It's great when this section says that MyWonderfulApp is free software, available via anonymous FTP from

```
ftp.site.dom/pub/myname/Wonderful
```

Also, it's even greater when you can write that information is also available on the Wonderful homepage at

```
http://www.site.dom/~myname/SWIG
```

References