Implementing and Benchmarking a LWE-based Fully Homomorphic Encryption Scheme

Meghan L. Clark
University of Michigan

Alex L. James
University of Michigan

Travis B. Martin
University of Michigan

Abstract

Fully homomorphic encryption (FHE) provides a way for third parties to compute arbitrary functions on encrypted data. This has the potential to revolutionize cloud computing services. Unfortunately, since their emergence in 2009, FHE schemes have become notorious for incurring enormous costs in time and space. Over the last four years optimizations have been proposed with impressive rapidity. However, the degree of progress is unknown, as these improvements are usually only asymptotically beneficial. The newness of the field and relative opacity of the literature has resulted in few implementations and evaluations of actual performance. To fill this gap, we implement a recent FHE scheme based on the Learning with Errors (LWE) hardness problem. We compare the performance of our system with an implementation of an earlier FHE scheme based on the Approximate GCD (AGC) hardness problem. We find that our system does [BETTER/WORSE] than the AGCD scheme. We also release our system to the public to promote additional experimentation and to increase the accessibility of this new cryptographic construct.

1 Introduction

Why you should care about FHE, why there are no implementations, why implementations would be awesome, why our contribution is therefore awesome.

2 Background

A brief history and explanation of FHE. Enough so that the reader knows what the eff we're talking about.

Now we're going to cite somebody. Watch for the cite tag. Here it comes [1]. The tilde character (~) in the source means a non-breaking space. This way, your reference will always be attached to the word that preceded it, instead of going to the next line.

3 Related Work

Specifically, the implementation papers - the AES paper, Implementing Gentry (maybe? They actually wrote code, right?), the "Practical" paper, the CNT/Scarab projects, whichever LWE was implemented but not released.

This section is for highlighting what makes our work different from previous work.

4 Methodology

This is about the experimental design. Start off with a super high-level description of our approach. A.k.a, given an existing Python SAGE implementation for AGCD, write our own LWE implementation also in Python SAGE, and then benchmark each on a variety of parameters described here.

IMPORTANT: This is where we make the case that our benchmarking comparisons are fair. This whole section is not just describing but also *justifying* our experimental design choices. Namely, the optimizations we chose, the parameters we chose, etc., and why those choices make it fair.

This is the most important section of the paper. This is where the reader can tell whether we did good science or bad science.

5 Implementation

The nitty-gritty about the implementation. This can just be a straight-up description about how our code works.

If you want to include code snippets, here's some example LaTeXfor that:

```
int main() {
    int result = 1
    if (result != 2) {
        printf("You get here every time, man.\n");
```

Figure 1: Description of what information is being displayed. Results/trends of note. What those results/trends mean (impact).

```
    }
}
```

6 Results

Here's a typical figure reference. The figure is centered at the top of the column. It's scaled. It's explicitly placed. You'll have to tweak the numbers to get what you want.

This text came after the figure, so we'll casually refer to Figure 1 as we go on our merry way.

7 Discussion

Can be short. Talk about the main results, particularly the impact. Were our results reasonable or surprising? Do our results mean that we should advocate for LWE or AGCD? Are they too close to call? Or are our results too specific to the implementations and optimizations used? When might we get different results?

8 Future Work

Can be short. Stuff not only that you'd want to do on this project in the future, but what any FHE researcher might try next after having read this.

9 Conclusion

A summary - basically the abstract again, but more detailed about the results. State problem, state why problem is bad, state the solution we tried, state whether or not it worked and to what degree. Close with a line about the future of FHE.

10 Availability

We feel very strongly that releasing our implementation to the public for examination and experimentation is one of the major contributions of this work. Our code and directions for running it can be found at:

`https://github.com/tbmbob/bootstraplessfhe`

If you have questions, we will do our best to answer them.

References

[1] GENTRY, C. *A Fully Homomorphic Encryption Scheme*. PhD thesis, Stanford University, 2009.