Zero Knowledge Compilers

John T. McCall
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA 56267
mcca0798@morris.umn.edu

ABSTRACT

<Insert Abstract Here>

General Terms

Need to figure this out yet

Keywords

Zero Knowledge Protocols, Compilers

1. INTRODUCTION

I will focus on Zero-Knowledge Compilers which are compilers that automatically generate Zero-Knowledge proofs. This is how I plan to use the following sources:

- I expect [1, 2, 4] to be my core sources, depending on how relevent [4] turns out to be I'll replace it with a better source.
- I will use [3, 5, 6] for background information and examples of Zero-Knowledge Protocols.
- I will need to find some papers for background information on compilers.

As stated above I need to find some sources about compilers. I probably will need to find more papers dealing with ZK-Compilers as well.

1.1 Key Points

What main problems(s) or questions(s) does the research address?

The main problem that the research address is how to create reliable zero knowledge protocols. They can be difficult to define and even harder to verify. Zero knowledge compilers help because they can efficiently generate zero knowledge protocols, and because of how they are constructed the user can trust that they will work.

What are the key contributions of each of your main sources?

Source [1] provides a great deal of information about their zero knowledge compiler, ZKCrypt. They go into detail about zero knowledge protocols, how their compiler produces them, and they give a proof verifying that their protocols are valid. They also talk about a few applications of their compiler.

Source [4] talks in depth about ZKPDL, which is a language they created for writing zero knowledge protocols. They also created an interpreter for this language, which performs optimizations to lower computational and space overhead. This paper also provides an example dealing with electronic cash.

Source [2] uses Σ-Protocols in a compiler to automatically generate sound and efficient zero knowledge proofs of knowledge. The compiler automatically generates the implementation of the protocol in Java, or it can output a description of the protocol in LaTeX.

How are the main sources related to each other?

The main sources all use compilers to generate zero knowledge protocols, but the ways they are implemented are all different so there is some room for comparison. All the compilers are also based off Σ-Protocols, or variations of Σ- Protocols.

What is the state of the research?

The current state is that the compilers have been implemented and tested. They all provided enough data to back up their research. Most of the work they are doing now will extend the applications of their compilers to support other proof types.

What background material will you need to present in order for your audience to understand the research?

I will need to provide background information on zero knowledge protocols and compilers. It's probably more important that I focus on zero knowledge protocols and only give basic compiler background.

2. ZERO KNOWLEDGE PROTOCOLS

The section will cover Zero Knowledge Protocols and will provide background and examples. There will probably be an easy to grasp example, such as the cave example, and a more advanced example.

2.1 Background

2.2 Examples

3. COMPILERS

UMM CSci Senior Seminar Conference, December 2013 Morris, MN.

This section will provide some basic background information about compilers.

3.1 Background

4. ZERO KNOWLEDGE COMPILERS

This section will be the main section. Here I will talk about my core sources and how they are using their compilers.

4.1 Sigma-Protocols

Here I'll talk about Sigma-protocols and how they are used in the following compilers.

Note to myself: Look into the Fiat-Shamir heuristic

4.2 ZKCrypt

Here I'll talk about ZKCrypt, it's implementation and verification steps.

4.3 ZKPDL

Similar to ZKCrypt section.

5. APPLICATIONS

Here I'll talk about how those compilers are used in the real world.

5.1 Electronic Cash

5.2 Another Application

6. CONCLUSION

This is where I'll neatly wrap everything up.

7. REFERENCES

[1] J. B. Almeida, M. Barbosa, E. Bangerter, G. Barthe, S. Krenn, and S. Z. Beguelin. Full proof cryptography: verifiable compilation of efficient zero-knowledge protocols. In *Proceedings of the 2012 ACM conference on Computer and communications security*, CCS '12, pages 488–500, New York, NY, USA, 2012. ACM. *This paper is one of the Core papers. It deals heavily with ZK-Protocols and on ZK-Compilers.*

[2] E. Bangerter, T. Briner, W. Hencecka, S. Krenn, S. Ahmad-Reza, and T. Schneider. Automatic generation of sigma-protocols. In *Proceedings of the 6th European conference on Public key infrastructures, services and applications*, EuroPKI'09, pages 67–82, Berlin, Germany, 2009. Springer-Verlag. *This paper is a core paper and focues on compilers that automatically generate sound and efficient Zero knowledge proofs of knowledge based on sigma-protocols.*

[3] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game. In *Advances in Cryptology Proceedings*, CRYPTO' 89, pages 218–229, New York, NY, USA, 1987. ACM. *This paper is referenced by most papers that deal with zero knowledge protocols, but it's old. As such it will be used for backround information and maybe an example.*

[4] S. Meiklejohn, C. C. Erway, A. Kupcu, T. Hinkle, and A. Lysyanskaya. Zkpdl: a language-based system for efficient zero-knowledge proofs and electronic cash. In *USENIX Security'10 Proceedings of the 19th USENIX conference on Security*, Security '10, Berkeley, CA, USA, 2010. USENIX Association. *This is a core paper, it delves into efficient ZK-Proofs and an application dealing with electronic cash.*

[5] A. Mohr. A survey of zero-knowledge proofs with applications to cryptography. *This article is great backround information and has several great examples I can use.*

[6] J.-J. Quisquater, L. Guillou, M. Annick, and T. Berson. How to explain zero-knowledge protocols to your children. In *Proceedings on Advances in cryptology*, CRYPTO '89, pages 628–631, New York, NY, USA, 1989. Springer-Verlag New York, Inc.