Designing Efficient Zero-Knowledge Proofs in the Ideal Linear Commitment Model

Jonathan Bootle

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of

University College London.

Computer Science Department

University College London

September 23, 2018

I, Jonathan Bootle, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

Zero-knowledge proofs are cryptographic protocols where a prover convinces a verifier that a statement is true, without revealing why it is true or leaking any of the prover's secret information. Since the introduction of zero-knowledge proofs, researchers have found numerous applications to other cryptosystems, such as electronic voting, group signatures, and verifiable computation. Zero-knowledge proofs have also become an integral part of blockchain-based cryptocurrencies.

Thus, designing efficient zero-knowledge proofs is an important goal. Recently, the design space has become extremely large. To simplify protocol design, designers have begun to separate the process into modular steps. Information theoretic protocols are designed in idealised communication models and compiled into real protocols with cryptographic assumptions.

In this thesis, we investigate the Ideal Linear Commitment model, which aims to characterise interactive zero-knowledge protocols where the prover and verifier use homomorphic commitment schemes. We demonstrate the model's power by exhibiting efficient protocols for useful tasks including NP-Complete problems and other more specialised problems. We demonstrate the model's versatility by compiling the idealised protocols into real protocols under two completely different cryptographic assumptions; the discrete logarithm assumption, and the existence of collision-resistant hash-functions.

We show that the Ideal Linear Commitment model is a useful and effective abstraction for producing zero-knowledge protocols. Furthermore, by identifying the limitations of the model and finding protocols outside these constraints, we display special techniques which result in more efficient protocols than ever.

The results are novel and highly efficient protocols. For example, results include the first ever discrete-logarithm argument for general statements with logarithmic communication cost, the first ever three-move discrete-logarithm argument for arithmetic circuit satisfiability with sub-linear communication costs, and an argument for list membership with sub-logarithmic communication, less than the number of bits required to specify a list index. Every single one of our protocols improves the theoretical state-of-the-art.

Impact Statement

This thesis demonstrates how real cryptographic protocols for a variety of tasks can be designed using a special communication model. It shows that one can separate the algebraic machinery used to design the protocols from the cryptographic assumptions and commitment schemes used to prove that the protocols are secure, and still exhibit highly efficient protocols which improve asymptotically on the prior state-of-the-art. This is likely to benefit the discipline considerably as it lowers the barriers to understanding and producing secure protocols of this type. Cryptographic assumptions can be quite specialised, and it can be difficult to understand the mathematics behind them. Having a framework within which one can prove idealised protocols secure and knowing that the result can be made into a real protocol drastically simplifies the task of protocol designers. Several subtle changes to the Ideal Linear Commitment model were also introduced to make it more realistic and effective.

Furthermore, by showing that certain proofs all fit into a framework, it becomes easier to understand their limitations. Linear algebra is an important part of security proofs in the Ideal Linear Commitment model. Therefore, in future, zero-knowledge protocols can be analysed through the lens of linear algebra. Linear algebra is very well studied, and powerful techniques from this discipline may lead to strong results about zero-knowledge protocols, such as lower bounds on the communication complexity of certain types of interactive zero-knowledge protocol.

The protocols presented in this work led to the creation of the zero-knowledge argument Bulletproofs [1]. Bulletproofs has been implemented by cryptocurrencies including Monero and PIVX. PIVX plans to bring Bulletproofs implementations into common use later in 2018. Following a first successful code audit, Bulletproofs is also set to enter widespread use on Monero's blockchain later in 2018, subject to further successful audits. As a result, the author's work will soon have a sizeable impact on the efficiency of payment systems in the real world, which at the time of writing, amount to a market capitalisation value of roughly two billion dollars and a daily trade volume of roughly thirty four million dollars [1]. The addition of Bulletproofs will lead to much smaller amounts of proof data being stored on

the blockchains for these cryptocurrencies, which means better performance and functionality. This may have a measurable impact on cryptocurrency adoption and pricing.

Zero-knowledge proofs are becoming better and better known, not only among research scientists, but increasingly among companies and technology enthusiasts. There are also ongoing standardisation efforts. The implementation of cryptographic protocols is a notoriously difficult task even for experts, and errors can have disastrous consequences. Making protocols easier to design and understand will be of great benefit to interested, non-expert parties who might try to use protocols in the future, promote their usage among wider user communities, or implement them in software or hardware.

[1] Data taken from `coinmarketcap.com`, on the 17th of September 2018.

Acknowledgements

During the four years of study which led to this thesis, I have been incredibly lucky to have the chance to interact with not only my colleagues from the information security group at UCL, but researchers from NTT Secure Platform Laboratory, UniversitÃ© Rennes 1, Microsoft Research Redmond, and IBM Research Zurich. Our discussions always served to open my mind and help me consider things in new ways.

Thanks to my coauthors, collaborators and friends Pyrros Chaidos, Christophe Petit, Essam Ghadafi, Mohammed Hajiabadi, Sune Jakobsen, Mary Maller, Mehdi Tibouchi, Keita Xagawa, Benedikt Bunz, Dan Boneh, Andrew Poelstra, Pieter Wuille, Greg Maxwell, Carsten Baum, Vadim Lyubashevsky, Rafael del Pino, Claire Delaplace, Thomas Espitau and Pierre-Alain Fouque.

Thanks to my supervisor Jens Groth who was a great mentor and provided me with a better model for how to be an academic researcher than I could ever have hoped for. Thanks to Andrea Cerulli who was always willing to help and discuss things, probably at the expense of his own work. Thanks to Vasilios Mavroudis and David Kohan Marzgao for providing wonderful distractions in the form of exciting mathematical puzzles. Thanks to my mother, whose motivation when I was a child made all of this possible.

Contents

List of Figures

List of Tables

Chapter 1

Introductory Material

1.1 Introduction

A zero-knowledge proof [4] is a protocol between two parties: a prover and a verifier. The prover may want to convince the verifier that an instance u belongs to a specific language $\mathscr{L}$ in NP. She wants to convince the verifier that the statement $u \in \mathscr{L}$ is true without revealing any confidential information. Such secret information, unknown to the verifier, which could make the statement easier to prove, should not be leaked as part of the protocol.

Zero-knowledge proofs are widely used in cryptography since it is often useful to verify that a party is following a protocol without requiring her to divulge secret keys or other private information. Applications range from digital signatures and public-key encryption to secure multi-party computation schemes with strong security guarantees, anonymous credentials, and verifiable cloud computing.

More formally, a zero-knowledge proof consists of a triple of algorithms $(\mathscr{G}, \mathscr{P}, \mathscr{V})$. These are the prover $\mathscr{P}$ and the verifier $\mathscr{V}$, and they engage in the protocol. The common-reference-string generator $\mathscr{G}$ produces the necessary setup information for $\mathscr{P}$ and $\mathscr{V}$ to run the protocol. In this thesis, the prover and verifier are interactive algorithms. Based on the setup information produced by the generator, the prover and the verifier exchange messages. At the end of the protocol, the verifier chooses whether the proof was convincing or not, and accepts or rejects the proof.

Zero-knowledge proofs satisfy three basic requirements.

Completeness When the statement is true, the prover always succeeds in convincing the verifier. In other words, when $u \in \mathscr{L}$, and the prover has a valid witness w, the verifier will always accept at the end of the protocol. Completeness can be viewed as more of a functionalilty requirement than a security requirement, and this property guarantees that the protocol works properly when the prover and verifier are honest.

Soundness When the statement is false, or the prover does not know a valid witness, the prover can never convince the verifier, and the verifier will always reject the proof. Viewed another way, if the verifier accepted the proof, then $u \in \mathscr{L}$, and the prover has a valid witness w. Soudness can be seen as a converse to completeness. It is a security requirement which protects the verifier from being fooled into accepting by a malicious prover. As such, it can be seen as a property of the verification algorithm.

Zero-Knowledge Despite taking part in the protocol with the prover, the verifier can never learn anything from the interaction except that the statement is true. This is modelled by showing that the entire interaction between the prover and the verifier can be simulated without any knowledge of the witness. If convincing-looking proofs can be simulated without any secret information, then it follows that nothing secret can be learned from real proofs. Zero-knowledge is a security requirement which protects the prover against a malicious verifier who is trying to learn the prover's secrets. As such, it can be seen as a property of the proving algorithm.

Many efficient zero-knowledge proofs are based on the discrete logarithm assumption [5, 6, 7, 8, 9, 1, 2, 10, 11, 12]. Although efficiency has improved over time, it looks as though all of these protocols draw on the same small collection of techniques, some used for optimising efficiency, and some for proving security.

For instance, all of the protocols use commitment schemes. These allow either party to 'commit' to a message, so that the message stays hidden but is fixed relative to the commitment, and can be revealed later. A party can commit to a message m by applying a commitment algorithm to m, and obtain a commitment c to m. The commitment c can then be sent to other parties. Later, the party can 'open' c by revealing m. The other parties may then check that c corresponds to m.

Commitment schemes satisfy two basic requirements.

Hiding Like a sealed envelope, a commitment should hide the message inside. The commitment c should not leak any information about m.

Binding Once an envelope is sealed, it is not possible to change the message inside without breaking open the envelope. Given a commitment c to a message m, it should be impossible to open c to a different message.

The protocols all use homomorphic commitments, where one can add two commitments together to obtain a commitment to the sum of the messages.

These protocols also follow a similar pattern of interaction between the prover and the verifier. The prover sends an initial message a_1 to the verifier. The verifier responds with a random challenge value e_1. The prover sends another message a_2 to the verifier, who responds with another random challenge e_2. This process continues for n rounds, until messages a_n and e_n have been exchanged. The prover sends a final message a_{n+1} to the verifier, and then the verifier decides whether to accept or reject.

In fact, there are even more similarities to be observed. For many protocols, the prover's messages a_1, a_2 up to a_n are actually single or possibly multiple commitments to secret values obtained by the prover, computed using the witness or the random challenge values that the prover has seen up to that point. The prover's final message a_{n+1} is often a specially chosen sum or linear combination of the secret values that the prover committed to earlier. Then, the verification algorithm involves checking that a_{n+1} really is the correct sum or linear combination of the prover's secret values, using the commitments to achieve this.

In their security proofs, the protocols all seem to use the same techniques, based on polynomial algebra, polynomial identity testing and linear algebra. In fact, on the whole, it does not seem important that the protocols are based on the discrete logarithm assumption, just that they use homomorphic commitment schemes to commit to the elements of finite fields.

One can ask whether it is possible to find the techniques common to all interactive zero-knowledge protocols based on the discrete logarithm assumption, and place them into a single framework. This would drastically simplify the tasks of

designining protocols, and proving mathematically that they are secure. Having fully understood or implemented one protocol of this type, it would be much easier to understand or implement any other.

Can we find such a framework? The first goal of this thesis is to convince the reader that the answer is yes.

Still, however easy it is to understand protocols from a simple framework, the framework is useless unless it can be used to create efficient protocols for a wide range of tasks. Can we use the framework to design interesting zero-knowledge protocols? The second goal of this thesis is to convince the reader that the answer is yes.

1.2 Contributions

As a building block in our arguments, we present an adaptation of the polynomial commitment sub-protocol appearing in [13], which allows the prover to commit to a polynomial so that the verifier can learn an evaluation of the polynomial in a secure manner. The sub-protocol has a square-root communication complexity in the degree of the polynomial.

Zero-Knowledge Proofs for Low-Depth Circuits While very efficient, arguments for general statements, like arithmetic circuit satisfiability, often make use of generic reductions and complex machinery, and fail to be as efficient as arguments specialised for a particular language. We give a zero-knowledge proof for low-depth circuits. In doing so, we bridge the gap between general and simple languages in three ways.

Firstly, we provide a framework to describe the types of languages commonly encountered. Protocols such as the 1-out-of-N membership argument of [14], and the polynomial evaluation argument of [11] prove membership in languages where the witnesses are zeroes of low-degree polynomial relations. In other words, the statement is an arithmetic circuit of low degree, and part of the witness is a satisfying assignment for the circuit. We give a general relation which allows us to recover specific protocols by instantiating with concrete polynomial relations. By separating the task of developing more efficient ways to perform the zero knowledge proof, and

the task of designing better relations to describe a given language, we can explain the logic behind past optimisations of membership proofs in [14, 10], and produce new optimisations for membership proofs and polynomial evaluation proofs.

Secondly, we unify the approaches used in [14, 11, 10] to construct zero-knowledge proofs for membership and polynomial evaluation, which can all be viewed as employing the same construction method. The constructions of zero-knowledge arguments for low degree polynomial relations in these works proceed by masking an input variable u as $f_u = ux + u_b$, using a random challenge x and a random blinder u_b. During the proof, the polynomial or circuit from the statement is computed with f_u in place of u, so that the original relation appears in the leading x coefficient. The communication and computational complexity of the resulting arguments is determined by the degree of the polynomial relation and the number of inputs. By contrast, the complexity of general arithmetic circuit protocols is determined by the number of gates. In the case of [13], the authors embed a polynomial evaluation argument for a polynomial of degree N into a low degree polynomial with $\log N$ inputs and degree $\log N$, obtaining a protocol with $O(\log N)$ communication using 3 moves, and requiring $O(\log N)$ operations to form cryptographic commitments. On the other hand, a polynomial of degree N requires N multiplication gates to evaluate in general, so the best arithmetic circuit protocol [13] can only achieve $O(\log N)$ communication in $O(\log N)$ moves, and uses $O(N)$ operations to form cryptographic commitments. In particular, in some settings, like the discrete logarithm setting, forming cryptographic commitments is based on computing exponentiations in a group of prime order which is much higher than that of computing finite-field multiplications. Computing $O(\log N)$ group exponentiations rather than $O(N)$ leads to a significant performance advantage when considering implementation on constrained devices.

Bayer [15] gives two efficient batch proofs for multiplication and polynomial evaluation, which achieve a square-root communication overhead in the number of proofs to be batched. The key to achieving square-root overhead in [15] is to use Lagrange interpolation to embed many instances of the same relation into a

single field element. This technique can be applied more generally to produce efficient batch proofs for the low-degree relations described above. Furthermore, by combining this with the polynomial commitment subprotocol in section **??**, we improve the communication cost of the batched proof from $\sqrt{t}c$ to $\sqrt{tc}$, where c is the communication cost of the original non-batched proof, and t is a large number representing the number of proofs to be batched together.

Thirdly, we exhibit a general protocol in our framework, and give an efficient batch protocol for proving and verifying t instances of the same relation simultaneously. We then show how to recover protocols of previous works with some optimisation. More specifically, we give new 1-out-of-N membership arguments and polynomial evaluation arguments. Our new instantiations simultaneously decrease communication costs and reduce prover and verifier computation, while retaining the conceptual clarity and simple 3-move structure of the originals. As an example, we obtain the most communication efficient Σ-protocols for membership or non-membership of a committed value in a public list, in the discrete logarithm setting. We also include an argument for range proofs, which captures the folklore method for performing range proofs and demonstrates the expressivity of our general relation.

One notable place where we improve communication efficiency over previous proofs is in our membership and polynomial evaluation proofs instantiated in the discrete logarithm setting, which use a constant number of group elements, but have better communication efficiency regardless of whether the proofs are instantiated in elliptic curve groups or multiplicative subgroups of finite fields. Another is the polynomial evaluation argument with $O(\frac{\log N}{\log\log N})$ communication costs, which is an asymptotic improvement over the previous state-of-the-art, $O(\log N)$. Finally, our batch polynomial evaluation argument improves on [15] by putting the $\log N$ cost inside a square root.

See Tables 1.4 and 1.5 for our results in the discrete logarithm setting. N is the instance-size, t is the number of batched instances, $\mathbb{G}$ means the number of group elements transmitted, $\mathbb{Z}_p$ means the number of field elements transmitted, $(\mathbb{G}, exp)$ means the number of group exponentiations and $(\mathbb{Z}_p, \times)$ means the number of field

multiplications. In the membership proofs, N is the number of items in the list for which we prove membership. In the polynomial evaluation proofs, N is the degree of the polynomial. In the range proofs, N is the width of the range that we consider.

Zero-Knowledge Proofs for Arithmetic Circuits One goal is to build an efficient argument system for the satisfiability of an arithmetic circuit, i.e., a circuit that consists of addition and multiplication gates over a finite field $\mathbb{F}$. The statement is the arithmetic circuit and some specified values for the circuit outputs. The prover's witness is a collection of input values for the arithmetic circuit which give the correct output. Arithmetic circuits are an attractive target for protocol design for several reasons.

- Given an arithmetic circuit and outputs, the problem of deciding whether there exist input wire values satisfying the circuit is NP-Complete. Therefore, if we can design zero-knowledge proofs for arithmetic circuit satisfiability, then this implies that we can give zero-knowledge proofs for all NP languages.

- Many cryptographic systems can be expressed in terms of arithmetic over finite fields of prime order. Given a zero-knowledge proof system for arithmetic circuit satisfiability, we can give zero-knowledge proofs which reason about other cryptosystems, often in order to provide stronger security guarantees.

- There exist compilers which take computer programs written in C (avoiding certain commands) and convert them into arithmetic circuits. Then, a zero-knowledge proof for arithmetic circuit satisfiability can become a zero-knowledge proof that the C program was executed correctly.

We provide two honest verifier zero-knowledge arguments for arithmetic circuit satisfiability. In general, the arguments has a square-root communication complexity. The arguments work by reducing the problem of verifying arithmetic circuit satisfiability to the problem of checking that the prover knows that for three commitments,

[1] We compare against the efficiency when [14] is instantiated using Pedersen commitments, and the prover and verifier know the openings of the list of commitments.

[2] We compare against the efficiency when [10] is instantiated using Pedersen commitments rather than Elgamal ciphertexts.

Protocol	Reference	Communication		Prover Computation		Verifier Computation	
		$\mathbb{G}$	$\mathbb{Z}_p$	$(\mathbb{G}, exp)$	$(\mathbb{Z}_p, \times)$	$(\mathbb{G}, exp)$	$(\mathbb{Z}_p, \times)$
Membership Proof	[13]	$4\log N + 8$	$2\log N + 7$	$12N$	$O(N)$	$4N$	$O(N)$
Membership Proof [1]	[14]	$4\log N$	$3\log N + 1$	$O(\log N)$	$O(N \log N)$	$O(\log N)$	$O(N)$
Membership Proof [2]	[10]	$\log N + 12$	$\frac{3}{2}\log N + 6$	$O(\log N)$	$O(N \log N)$	$O(\log N)$	$O(N)$
Membership Proof	This Work, **??**	7	$4\log N + 4$	$O(\frac{\log N}{\log\log N})$	$O(N \log N)$	$O(\frac{\log N}{\log\log N})$	$O(N)$
Membership Proof	This Work, **??**	$2.7\sqrt{\log N}$ $+5$	$1.9\log N +$ $2.7\sqrt{\log N} + 4$	$O(\frac{\log N}{\log\log N})$	$O(N \log N)$	$O(\frac{\log N}{\log\log N})$	$O(N)$
Batch Membership Proof	This Work, **??**	$4.1\sqrt{t \log N}$	$4.1\sqrt{t \log N}$	$O(t \log tN)$	$O(tN \log tN)$	$O(\sqrt{t} \log tN)$	$O(tN)$

Table 1.1: Efficiency Comparisons when our low-depth circuit protocol is instantiated in the discrete logarithm setting.

Protocol	Reference	Communication		Prover Computation		Verifier Computation	
		$\mathbb{G}$	$\mathbb{Z}_p$	$(\mathbb{G}, exp)$	$(\mathbb{Z}_p, \times)$	$(\mathbb{G}, exp)$	$(\mathbb{Z}_p, \times)$
Polynomial Evaluation	[13]	$4\log N + 8$	$2\log N + 7$	$12N$	$O(N)$	$4N$	$O(N)$
Polynomial Evaluation	[11]	$4\log N + 2$	$3\log N + 3$	$O(\log N)$	$O(N \log N)$	$O(\log N)$	$O(N)$
Polynomial Evaluation	This Work, **??**	7	$3\log N + 4$	$O(\frac{\log N}{\log\log N})$	$O(N \log N)$	$O(\frac{\log N}{\log\log N})$	$O(N)$
Polynomial Evaluation	This Work, **??**	$O(\frac{\log N}{\log\log N})$	$O(\frac{\log N}{\log\log N})$	$O(\frac{\log N}{\log\log N})$	$O(N \log N)$	$O(\frac{\log N}{\log\log N})$	$O(N)$
Batch Polynomial Evaluation	[15]	$O(\sqrt{t}\log N)$	$O(\sqrt{t}\log N)$	$O(t \log N)$	$O(tN \log N)$	$O(\sqrt{t}\log N)$	$O(tN)$
Batch Polynomial Evaluation	This Work, **??**	$2.8\sqrt{t \log N}$	$2.8\sqrt{t \log N}$	$O(t \log tN)$	$O(tN \log tN)$	$O(\sqrt{t}\log tN)$	$O(tN)$
Range Proof	This Work, **??**	7	$3\log N + 4$	$O(\log N)$	$O(\log N)$	$O(\log N)$	$O(\log N)$
Range Proof	This Work, **??**	$O(\frac{\log N}{\log\log N})$	$O(\frac{\log N}{\log\log N})$	$O(\log N)$	$O(\log N)$	$O(\log N)$	$O(\log N)$
Batch Range Proof	This Work, **??**	$2.8\sqrt{t \log N}$	$2.8\sqrt{t \log N}$	$O(t \log N)$	$O(t \log N)$	$O(t \log N)$	$O(t \log N)$

Table 1.2: Efficiency Comparisons when our low-depth circuit protocol is instantiated in the discrete logarithm setting.

two correspond to committed vectors of values, and the third contains the scalar product of the two vectors. The first argument has fewer rounds of interaction than previously published arguments, and highlights some interesting subtleties in our communication model. The second argument has better practical efficiency, and when instantiated using a commitment scheme with a few special properties, a special protocol for scalar products leads to an argument that only requires a *logarithmic* communication complexity.

3-Move Protocol for Arithmetic Circuit Satisfiability We give a 3-move protocol for arithmetic circuit satisfiability. When instantiated using Pedersen commitments, this gives the first arithmetic circuit satisfiability protocol with a square-root communication complexity in only three moves. Unfortunately, the argument has a large (superlinear) computational cost for both the prover and the verifier, so is presented mostly for theoretical interest.

We start from the circuit satisfiability argument of Groth [7], which requires 7 moves and has square root communication complexity in the *total* number of gates. In this argument the prover commits to all the wires using homomorphic multicommitments, verifies addition gates using the homomorphic properties, and uses a product argument to show that the multiplication gates are satisfied.

We first improve Groth's argument into a 5-move argument with square root communication complexity in the number of *multiplication gates* only. We achieve fewer moves compared to [7] by avoiding generic reductions to linear algebra statements. We remove the communication cost of the addition gates in the argument by providing a technique that can directly handle a set of Hadamard products and linear relations together.

Logarithmic Complexity Argument. In spite of all these improvements, the above argument still requires a square root communication complexity with respect to multiplication gates. In the first move the prover commits to all circuit wires using $3m$ commitments to n elements each, where $mn = N$ is a bound on the number

[1] We compare against the efficiency when [14] is instantiated using Pedersen commitments, and the prover and verifier know the openings of the list of commitments.

[2] We compare against the efficiency when [10] is instantiated using Pedersen commitments rather than Elgamal ciphertexts.

Reference	Moves	Communication		Prover Complexity		Verifier Complexity	
		$\mathbb{G}$	$\mathbb{Z}_p$	exp.	mult.	exp.	mult.
[6]	3	$6N$	$5N+2$	$6N$	$6N$	$6N$	0
[7]	7	$9\sqrt{N}+4$	$7\sqrt{N}+6$	$\frac{6N}{\log N}$	$O(N\log N)$	$\frac{39\sqrt{N}}{\log N}$	$O(N)$
[7]	$2\log N+5$	$2\sqrt{N}$	$7\sqrt{N}$	$\frac{6N}{\log N}$	$O(N)$	$\frac{18\sqrt{N}}{\log N}$	$O(N)$
[8]	5	$30\sqrt{N}$	$7\sqrt{N}$	$\frac{6N}{\log N}$	$O(N\log N)$	$\frac{77\sqrt{N}}{\log N}$	$O(N)$
This work	5	$2\sqrt{N}$	$2\sqrt{N}$	$\frac{6N}{\log N}$	$3N\log N$	$\frac{8\sqrt{3N}}{\log N}$	$O(N)$

Table 1.3: Efficiency comparison between our 5-move argument in the discrete logarithm setting and the most efficient constant-move interactive zero-knowledge arguments relying on discrete logarithms and hash-based . Since We express communication in number of group elements $\mathbb{G}$ and field elements $\mathbb{Z}_p$ and computation costs in number of exponentiations over $\mathbb{G}$ and multiplications over $\mathbb{Z}_p$. The efficiency displayed is for a circuit with N multiplication gates.

Protocol	Reference	Communication		Prover Computation		Verifier Computation	
		$\mathbb{G}$	$\mathbb{Z}_p$	$(\mathbb{G}, exp)$	$(\mathbb{Z}_p, \times)$	$(\mathbb{G}, exp)$	$(\mathbb{Z}_p, \times)$
Membership Proof	[13]	$4\log N + 8$	$2\log N + 7$	$12N$	$O(N)$	$4N$	$O(N)$
Membership Proof [1]	[14]	$4\log N$	$3\log N + 1$	$O(\log N)$	$O(N \log N)$	$O(\log N)$	$O(N)$
Membership Proof [2]	[10]	$\log N + 12$	$\frac{3}{2}\log N + 6$	$O(\log N)$	$O(N \log N)$	$O(\log N)$	$O(N)$
Membership Proof	This Work, **??**	7	$4\log N + 4$	$O(\frac{\log N}{\log\log N})$	$O(N \log N)$	$O(\frac{\log N}{\log\log N})$	$O(N)$
Membership Proof	This Work, **??**	$2.7\sqrt{\log N}$ $+5$	$1.9\log N +$ $2.7\sqrt{\log N} + 4$	$O(\frac{\log N}{\log\log N})$	$O(N \log N)$	$O(\frac{\log N}{\log\log N})$	$O(N)$
Batch Membership Proof	This Work, **??**	$4.1\sqrt{t \log N}$	$4.1\sqrt{t \log N}$	$O(t \log tN)$	$O(tN \log tN)$	$O(\sqrt{t} \log tN)$	$O(tN)$

Table 1.4: Efficiency Comparisons when our low-depth circuit protocol is instantiated in the discrete logarithm setting.

Protocol	Reference	Communication		Prover Computation		Verifier Computation	
		$\mathbb{G}$	$\mathbb{Z}_p$	$(\mathbb{G}, exp)$	$(\mathbb{Z}_p, \times)$	$(\mathbb{G}, exp)$	$(\mathbb{Z}_p, \times)$
Polynomial Evaluation	[13]	$4\log N+8$	$2\log N+7$	$12N$	$O(N)$	$4N$	$O(N)$
Polynomial Evaluation	[11]	$4\log N+2$	$3\log N+3$	$O(\log N)$	$O(N\log N)$	$O(\log N)$	$O(N)$
Polynomial Evaluation	This Work, **??**	7	$3\log N+4$	$O(\frac{\log N}{\log\log N})$	$O(N\log N)$	$O(\frac{\log N}{\log\log N})$	$O(N)$
Polynomial Evaluation	This Work, **??**	$O(\frac{\log N}{\log\log N})$	$O(\frac{\log N}{\log\log N})$	$O(\frac{\log N}{\log\log N})$	$O(N\log N)$	$O(\frac{\log N}{\log\log N})$	$O(N)$
Batch Polynomial Evaluation	[15]	$O(\sqrt{t}\log N)$	$O(\sqrt{t}\log N)$	$O(t\log N)$	$O(tN\log N)$	$O(\sqrt{t}\log N)$	$O(tN)$
Batch Polynomial Evaluation	This Work, **??**	$2.8\sqrt{t\log N}$	$2.8\sqrt{t\log N}$	$O(t\log tN)$	$O(tN\log tN)$	$O(\sqrt{t}\log tN)$	$O(tN)$
Range Proof	This Work, **??**	7	$3\log N+4$	$O(\log N)$	$O(\log N)$	$O(\log N)$	$O(\log N)$
Range Proof	This Work, **??**	$O(\frac{\log N}{\log\log N})$	$O(\frac{\log N}{\log\log N})$	$O(\log N)$	$O(\log N)$	$O(\log N)$	$O(\log N)$
Batch Range Proof	This Work, **??**	$2.8\sqrt{t\log N}$	$2.8\sqrt{t\log N}$	$O(t\log N)$	$O(t\log N)$	$O(t\log N)$	$O(t\log N)$

Table 1.5: Efficiency Comparisons when our low-depth circuit protocol is instantiated in the discrete logarithm setting.

of multiplication gates, and in the last move after receiving a challenge he opens commitments that can be constructed from the previous ones and the challenge. By setting $m \approx n$ we get a minimal communication complexity of $O(\sqrt{N})$.

Our key idea to break this square root communication complexity barrier is to replace the last opening step in this protocol with a special protocol for scalar products. In Section **??** we provide an argument system for this problem, which only requires a logarithmic communication with respect to the vector sizes. The argument is built in a recursive way, reducing the size and complexity of the statement further in each recursion step. This uses a special property of commitments, namely homomorphic properties with respect to the keys. Pedersen commitments, based on the discrete logarithm assumption, satisfy this property. As a result, using this inner product argument as a subroutine in our main argument, and instantiating with Pedersen commitments, we obtain an arithmetic circuit satisfiability argument with logarithmic communication complexity based on the discrete lgoarithm assumption. This argument was the first of its kind. The constants in the complexities of the protocol have since been improved in [16].

When using a logarithmic number of moves and applying a reduction similar to [17], our scheme dramatically improves the communication costs with respect to all previous work without incurring any significant overhead. We note that [17] uses the reduction to reduce computation whereas we use it to reduce communication.

1.3 Published Work

In this section, we discuss the author's previously published works.

Group and Ring Signatures The paper [10] appeared at ESORICS 2015. The work proposed a new security model for a variant of ring-signatures called accountable ring signatures, and provided a construction of the cryptosystem. Ring signatures allow a single user to create a signature on behalf of a group of users, formed in an ad-hoc fashion. However, the original security model for ring signatures has no mechanism for revoking anonymity and tracing the origin of a signature in case a user misbehaves. Accountable ring signatures include a tracing mechanism. Explain

in more detail in terms of different parties. The construction of accountable ring signatures given in the paper relies on a zero-knowledge proof that a committed value is a member of a list of values, which are provided in encrypted form. As a new student, my task when working on this paper was to check that the construction, notation, security proof and efficiency calculations for the zero-knowledge proof were correct.

The paper [18] appeared at ACNS 2016. The work proposed a new model for the security and functionality of group signatures, in the case where the group of users can be updated over time by adding new users and removing some users from the system. These were referred to as 'Fully Dynamic Group Signatures', where 'dynamic' refers to the group of users. The paper shows that given any construction of a fully dynamic group signature, one can easily obtain constructions of group signatures in older models, namely the partially dynamic group signatures of [19] and [20], and points to some subtle attacks that can arise in other models since they are not prohibited by the security definitions. In this paper, my task was to show that our model for fully dynamic group models was all-encompassing and that a construction allowed one to easily build a construction of group signatures for the other security models, and prove that the constructions satisfied the appropriate definitions.

Lattice Cryptanalysis The work [21] appeared at CT-RSA 2018. Compact-LWE [22] was a novel, lattice-based encryption scheme presented in [22]. It was proposed as a secure scheme for the post-quantum setting, as part of the recent NIST call for efficient post-quantum key-encapsulation mechanisms and signature schemes. The scheme was also based on a new assumption called Compact LWE, whose hardness was justified with a reduction to the more standard LWE problem, showing that solving the Compact-LWE problem was at least as hard as solving the LWE problem. In our work, we showed that the encryption scheme was easily broken for the concrete parameters given in the original paper, since the secret key could always be recovered efficiently. Furthermore, we showed that the hardness reduction to the LWE problem was flawed, and gave an algorithm for solving the Compact-LWE

problem which essentially showed that solving Compact-LWE was *no harder* than solving LWE. These arguments presented a strong case against the use of Compact-LWE. My own contribution to this paper was quite small, trying to find the best way to explain the details of the various attacks presented.

The work cite! will appear at ASIACRYPT 2018. BLISS [23] is an efficient, lattice-based signature scheme. However, previous work [24] shows that certain variable-time implementations of the signature's rejection sampling algorithm lead to side-channel attacks. Two quantities derived from the signature's secret key are leaked, related to the norm of the secret key and a noisy scalar product of the secret key with another public value. Previous work [24] demonstrates that for a small subset of weak secret keys, one can use the norm leakage to recover the secret at a high computational cost, and dismisses the scalar product leakage, as one would have to solve a problem akin to LWE in order to recover the secret. However, our work observes that the new LWE-like problem does not feature modular reduction, and so can be efficiently solved using linear regression. We measure the number of signatures and the time required to recover the secret key for different BLISS parameter settings. We also formalise the problem of LWE without modular reduction and give theoretical upper and lower bounds for the number of signatures required to solve the new problem, relating these to the BLISS parameter choices. In this paper, my contribution was to spot an idea from another source which used regression algorithms to solve a similar problem, implement the side-channel attack, and investigate the attack for different parameter settings.

Surveys The work [25] was published in the proceedings of FOSAD 2015. The work was a tutorial on zero-knowledge proof systems for the International School on Foundations of Security Analysis and Design (FOSAD). It did not present any new techniques, but was split into three parts. The first was an explanation of the properties of zero-knowledge proofs. The second gave details on the design of some simple interactive zero-knowledge proofs, and the third section did the same for some basic non-interactive zero-knowledge protocols. I was responsible for writing the third section of the tutorial, where I explained how simplified examples

of techniques from the hidden-bits model featured in [26] and a proof [27] based on the Boneh-Goh-Nissim public-key encryption scheme [28], as well as giving some information on pairing-based SNARKs [29, 30, 31, 32, 33, 34, 35, 36, 2].

Prover-Efficient ZK and Hash-based arguments The work [37] appeared at ASIACRYPT 2017. In this work gave the first zero-knowledge proofs for arithmetic circuit satisfiability with sub-linear communication complexity, linear computational cost, or constant overhead, for the prover, and a slightly sub-linear verification cost. The argument works by introducing a new commitment scheme based on hash-functions and error-correcting codes, both of which are computable in linear time and make use of expander graphs. Then, a collection of techniques used in other works such as [38] are abstracted into a new idealised communication model called the Ideal Linear Commitment model. The paper presents a proof of arithmetic circuit satisfiability in the Ideal Linear Commitment model, and a compilation converting zero-knowledge proofs in the idealised model into real zero-knowledge proof with perfect zero-knowledge and soundness based on the existence of suitable collision-resistant hash-functions and error-correcting codes. My personal contribution to this paper was to design all of the ILCprotocols for arithmetic circuit satisfiability, provide security proofs for them, and calculate their efficiency.

The work cite! will appear at ASIACRYPT 2018. The work considers a RAM machine specification called TinyRAM, and the problem of verifying, in zero-knowledge, that a given TinyRAM program was executed correctly. The authors solve the problem using the techniques from [37] and obtain zero-knowledge proofs with sub-linear communication complexity and close to constant computational overhead. The extra overhead arises due to computational costs associated with verifying the RAM machine model of computation that were not present with arithmetic circuits. The methodology is very similar to that of [37]; providing ILC protocols to verify the correctness of a RAM computation, and then using a compiler to produce real zero-knowledge proofs based on hash-functions and error-correcting codes. Again, my personal contribution to the paper was to design all of the extra ILC protocols required to verify correct program execution, as extra arguments, such

as a verifiable shuffle, were required. I was also responsible for their security proofs and efficiency calculations.

Discrete-Logarithm-based arguments The work [9] appeared at EUROCRYPT 2016. In this work, the authors propose two new arguments for arithmetic circuit satisfiability, and sub-protocols for particular tasks, based on the discrete logarithm assumption. The first protocol is a 5-move interactive argument with a square-root communication complexity in the size of the arithmetic circuit, using a sub-protocol which commits to polynomials and then reveals the evaluation of the polynomial at a given point in a verifiable manner. Using a recursive sub-protocol which verifies that two values committed using Pedersen commitments have a given scalar product, which has a logarithmic communication complexity and requires a logarithmic number of moves, the 5-move argument can be converted into a new arithmetic circuit argument with similar computation costs. My contribution to this paper was the formalisation of new security definitions required for the polynomial commitment sub-protocol and optimising that protocol, a description of how to pre-process an arithmetic circuit to convert it into the format required by the main zero-knowledge arguments, and notation and part of the proof of a generalised forking lemma used to prove the knowledge soundness of the logarithmic move arguments in the paper.

The work [1], known as Bulletproofs, appeared at S&P 2018. In this work, the authors optimise the logarithmic-communication argument of [9] to reduce communication costs by a factor of three. They also present a simplified argument for the special task of range proofs, which demonstrate that a committed value lies in a particular interval, and provide an implementation and concrete performance measurements for the new argument. They also give a secure multi-party computation protocol allowing various parties to compute their own zero-knowledge proofs in parallel and them aggregate them securely later on. Having discovered the techniques to cut communication costs by a factor of three in parallel with the rest of the other authors and joined the paper write-up at a later stage when almost complete, I was responsible for choosing the correct definitions of zero-knowledge proofs for the paper and helping to choose good notation for the arguments in the paper.

The work [12] appeared at PKC 2018. In this work, the authors identify the techniques used to give zero-knowledge proofs in previous works such as [39, 2, 10], which are all statements encoded into low-depth circuits, or low-degree polynomials. They specify a relation-framework which encompasses all of the statements proved in those zero-knowledge proofs. They then give a zero-knowledge protocol for single instances of the relation, and a batched protocol which builds on techniques from [15]. They show that for particular choices of relation, one can obtain zero-knowledge membership proofs and polynomial evaluation arguments with better concrete and asymptotic efficiency than previously known, and capture folklore range-proofs based on the discrete logarithm assumption. My contribution to this paper was the security definitions for the polynomial commitment argument and optimisations to the argument itself, which are similar to my contributions in [9]. I identified the method of generalising from arguments for single statement to batched arguments, and I discovered choices of relation within the framework which led to arguments with improved asymptotic properties.

Lattice-based arguments The work [40], appeared at CRYPTO 2018. In this work, the authors give zero-knowledge arguments for arithmetic circuit satisfiability based on cryptographic assumptions in lattices. The arguments have a constant number of moves, a quasilinear computational complexity, and a sub-linear communication complexity. In many respects, this argument is closely related to the square-root communication argument of [9], with modifications to reflect the change from discrete-logarithm groups to lattices. As a crucial step in the argument, the authors provide a zero-knowledge proof of knowledge of values committed using commitments based on the hardness of the Short-Integer-Solution problem. This proof-of-knowledge was a big improvement over prior work, as it proves that the prover knows openings to stated commitments, rather than some multiple of those commitments, which is a weaker security guarantee. Furthermore, this is the first lattice-based zero-knowledge argument for large and general statements which has a sub-linear communication complexity. My contributions in this paper were the adaptations of the 5-move argument from [9] to the new lattice-based setting, new

security proofs for the protocol, and a novel technique boosting the soundness of the zero-knowledge protocol by simulating operations in finite field extensions over integer modules, building on work by [41] and [42].

Work in this Thesis This thesis focusses on contributions from the following papers.

1. [9], which contains the square-root and logarithmic communication arguments for arithmetic circuit satisfiability based on the discrete logarithm assumption. We use all of the arguments but the polynomial commitment sub-protocol from this paper.

2. [12], which contains the relation-framework for low-degree polynomials, and efficient batched protocols for low-degree polynomial relations based on the discrete logarithm assumption. We use all of the arguments from this paper.

3. [37], which defines the Ideal Linear Commitment model, and gives linear-time zero-knowledge protocols for arithmetic circuit satisfiability based on hash-functions and error-correcting codes. From this paper, we use the ILC model and compilation of ILC protocols into real protocols using hashes and error-correcting codes.

4. [40], which contains a square-root communication argument for arithmetic circuit satisfiability based on the Short Integer Solution problem. From this paper, we use the soundness-boosting techniques over finite field extensions.

We also include the following pieces of unpublished work.

1. A novel argument for arithmetic circuit satisfiability with three moves and a sub-linear communication complexity.

2. Small modifications to the ILC model and compiler [37], for efficiency reasons.

3. A compiler from ILC protocols to real protocols based on the discrete logarithm assumption.

1.4 Outline and Recipes

Outline Chapter 2 gives a detailed survey of related work. Chapter **??** provides full and formal definitions of zero knowledge proofs and arguments and arithmetic circuits, and describes the Ideal Linear Commitment Model. Chapter **??** gives definitions for the discrete logarithm assumption, and collision resistance for cryptographic hash functions. Chapter **??** contains some lemmas, and their proofs, which will be useful for proving security of our Ideal Linear Commitment protocols and compiled protocols. Chapter **??** presents various Ideal Linear Commitment protocols. Chapter **??** compiles Ideal Linear Commitment protocols into real zero-knowledge protocols based on the discrete logarithm assumption or collision-resistant hash-functions. Chapter **??** presents some special protocol optimisations. Chapter 9 contains conclusions and ideas for future investigation.

Recipes The results in this thesis are fairly modular. That is, if one is only interested in a particular type of zero-knowledge argument, it is possible to restrict their attention to particular parts of the thesis.

To construct a hash-based argument for arithmetic circuits, one can use either the three-move or five-move arithmetic circuit argument, with the compiler based on hash functions and error-correcting codes. With the three-move argument, one can use the argument for small fields in order to boost the soundness of the resulting protocol.

To construct discrete-logarithm based arguments for arithmetic circuits, one can use the three-move or five-move arithmetic circuit argument, with the compiler based on Pedersen commitments. With the five-move argument, one can then apply the recursive argument for scalar products to obtain a protocol with logarithmic communication complexity.

To construct arguments for specialised languages, such as polynomial evaluation arguments, membership arguments, and range proofs, one can use the low-depth circuit argument with either compiler.

Chapter 2

Background and Related Work

Zero-knowledge proofs were invented by Goldwasser, Micali, and Rackoff [43]. In defining zero-knowledge proofs, the authors solved several important conceptual problems.

Firstly, defining what is meant by interactive protocols between two parties required Interactive Turing Machines to put the concept on a rigourous theoretical footing. This lead to the new computational complexity classes IP and ZK of languages which can be recognised by interactive proofs and zero-knowledge proofs, respectively. They gave a zero-knowledge proof for quadratic residuosity, the first zero-knowledge proof, showing that the class ZK was non-empty.

Secondly, they solved the problem of what it means for some party to know something. The knowledge of a party, or computing device, was captured by whatever it is possible for that party to compute, given the information available to it, and its own computational constraints.

Finally, the problem of what it means to gain no knowledge from an interaction was captured using a simulation-based definition. That is, if it is possible to efficiently simulate the contents of an interactive protocol without taking part in the protocol or knowing any secret information that the participants are privy to, then observing the interaction cannot confer any new knowledge. What can be computed from viewing the execution of the protocol is exactly the same as what can be computed without seeing it, and using a simulated execution instead.

Goldreich et al. [44] later showed that all languages in NP have zero-knowledge

proofs, so that NP is contained inside ZK. Informally, this means that for any problem for which one can check the solution efficiently, one can also convince somebody else that you hold the solution, without giving away any information about the solution. In fact, more is true. By taking several results together, we know that there are zero-knowledge proofs for every language in IP [45, 46, 47], assuming the existence of one-way functions.

Feige et al. [48] introduced zero-knowledge proofs-of-knowledge. These are different to the original proposal of zero-knowledge proofs. Let us consider the difference for NP languages L, with an instance u and a witness w. The original zero-knowledge proofs could be referred to as 'zero-knowledge proofs of membership' in this context. They prove that u lies in L, without leaking any further information, such as w. So the verifier learns that some valid w exists, but this does not guarantee that the prover actually knows a witness. In a proof of knowledge, the verifier learns that the prover knows a valid w, and nothing more. These are useful for identification schemes, for example, where the prover might authenticate themselves by proving that the know the secret key corresponding to a particular public key.

We can classify zero-knowledge proofs according to the number of rounds of interaction that take place between the prover and the verifier. Non-interactive zero-knowledge proofs were introduced in [49]. In these proofs, the proof consists of a single message sent from the prover to the verifier, who then accepts it or rejects it. Non-interactive zero-knowledge proofs require a common reference string as input to the protocol, to be used by both the prover and the verifier. Without a common reference string, it is only possible to construct non-interactive zero-knowledge proofs for languages in the complexity class BPP, which are seen as trivial, as the verifier can efficiently decide whether an instance is in a BPP-language without receiving any help from the prover.

The soundness and zero-knowledge properties of zero-knowledge proofs usually come in three different flavours:

- Perfect; the property is always satisfied, even against computationally unbounded adversaries.

- Statistical; the property fails to be satisfied with at most negligible probability, even against computationally unbounded adversaries.
- Computational; the property fails to be satisfied with at most negligible probability, against computationally bounded adversaries.

So far, we have discussed zero-knowledge proofs, which have perfect or statistical soundness. However, these can only have computational zero-knowledge. Protocols with computational soundness and perfect or statistical zero-knowledge are called zero-knowledge *arguments*. Brassard et al. [50] showed that all languages in NP have zero-knowledge arguments with perfect zero-knowledge. Micali [51] introduced the related notion of CS (computationally-sound) proofs, for which false proofs for true statements exist, but are computationally difficult to find.

Gentry et al. [52] used fully homomorphic encryption to construct zero-knowledge proofs where the communication complexity corresponds to the size of the witness. For circuit satisfiability, for example, the scheme works by encrypting the witness using a symmetric key encryption scheme, and using fully homomorphic encryption to decrypt the witness and evaluate the circuit homomorphically in the witness while still in encrypted form. This result gives somehow optimal communication complexity, as proofs cannot in general have communication that is smaller than the witness size unless surprising results about the complexity of solving SAT instances hold [53, 54].

Kilian [55] showed that in contrast to zero-knowledge proofs, zero-knowledge arguments can have very low communication complexity. His construction relied on the PCP theorem. Probabilistically checkable proofs, or PCPs, are proofs consisting of strings of many elements, whose correctness can be checked by examining only a small number of elements, sampled at random. Kilian's scheme has an excellent polylogarithmic communication complexity, but does not yield a practical scheme due to the large computational overhead required to convert statements into PCPs. In his scheme, the prover converts the statement to be proved into a PCP consisting of bits, commits to each of the bits using a single commitment, and then hashes all of the commitments in a Merkle tree. The verifier chooses a few bits of the PCP

to verify. The prover reveals those commitments from the Merkle tree, and uses a simple, auxiliary zero-knowledge proof system to prove that the committed bits will pass the verifier's checks.

Ishai et al. [56] introduce commitment schemes with linear decommitment. After the committer commits to several values, they can open a linear combination of the commitments, in a verifiable manner. This is a weakening of the property of homomorphic commitments, where one can use the homomorphic property to compute the correct linear combination of the commitments, and then open them. These commitments are closely related to the ILC model. One could view parts of our compilation of ILC protocols into zero-knowledge protocols based on hash functions and error-correcting codes as a proof that one can construct a commitment scheme with linear decommitment from these ingredients; one with an interactive decommitment phase.

If an interactive protocol is zero-knowledge, then for any verifier, even a malicious one, there exists an efficient simulator for the protocol. Honest verifier zero-knowledge [57] is a weaker property, which only guarantees that there exists a simulator for the interaction between an honest prover and an honest verifier. On introducing the notion, [57] show that any protocol with statistical honest-verifier zero-knowledge can be converted into a fully statistical zero-knowledge protocol under the discrete logarithm assumption. [58] show the same result under general one-way permutations, and [59] improved the result to one-way functions.

Another result by Damgaard [60] showed that public-coin honest verifier zero-knowledge protocols with a constant number of rounds can be transformed into a fully zero-knowledge protocol without any complexity assumptions. [61] show that statistical honest-verifier zero-knowledge proofs can be converted into statistical fully zero-knowledge proofs without any complexity assumptions.

Assuming a common reference string and relying on trapdoor commitments, Damgård [62] gave a transformation yielding concurrently secure protocols for Σ-Protocols. The transformation can be optimized [63] using the idea that for each public-coin challenge x, the prover first commits to a value x', then the verifier sends

a value x'', after which the prover opens the commitment and uses the challenge $x = x' + x''$. The coin-flipping can be interleaved with the rest of the proof, which means the transformation preserves the number of rounds and only incurs a very small efficiency cost to do the coin-flipping for the challenges.

If one does not wish to rely on a common reference string for security, one can use a private-coin transformation where the verifier does not reveal the random coins used to generate the challenges sent to the prover (hence the final protocol is no longer public coin). One example is the Micciancio and Petrank [64] transformation (yielding concurrently secure protocols) while incurring a small overhead of $\omega(\log \lambda)$ with respect to the number of rounds as well as the computational and communication cost in each round. The transformation preserves the soundness and completeness errors of the original protocol; however, it does not preserve statistical zero-knowledge as the obtained protocol only has computational zero-knowledge.

There are other public-coin transformations to general zero-knowledge e.g. Goldreich et al. [65]. The transformation relies on a random-selection protocol between the prover and verifier to specify a set of messages and restricting the verifier to choose challenges from this set. This means to get negligible soundness error these transformations require $\omega(1)$ sequential repetitions so the round complexity goes up.

The Fiat-Shamir transformation [66] is a method of converting public-coin interactive zero-knowledge arguments into non-interactive zero-knowledge proofs. The new non-interactive protocol include a hash function in the common reference string. The prover replaces the verifier's messages with a hash of the protocol transcript up to that point. The resulting arguments are highly efficient in practice and are provably secure in the random oracle model [67]. In the random oracle model, even if the initial interactive proof only has honest verifier zero-knowledge, the resulting argument will have full zero-knowledge.

However, it has been shown [**?**, **?**] that there are interactive protocols which are sound in the random oracle model, but which are insecure for any choice of hash function. Despite this theoretical problem, the Fiat-Shamir heuristic is still used to produce arguments for practical applications, where the hope is that it does give

sound arguments for "natural" problems.

Schnorr [5] and Guillou and Quisquater [68] gave early examples of practical zero-knowledge arguments for concrete number theoretic problems. Schnorr's protocol proves knowledge of a discrete logarithm, and Guillou-Quisquater's protocol proves knowledge of the message corresponding to an RSA encryption. Extending Schnorr's protocols, there have been many constructions of zero-knowledge arguments based on the discrete logarithm assumption. Cramer and Damgård [6] gave a zero-knowledge argument for arithmetic circuit satisfiability, which has linear communication complexity. The argument uses homomorphic commitments to all wire values in the circuit, using the homomorphic property to verify that that the addition gates in the circuit are satisfied, and giving a protocol to verify multiplications which is used for each multiplication gate in the circuit.

Before the logarithmic protocol presented in this thesis, the most efficient discrete logarithm based zero-knowledge arguments for arithmetic circuits were the ones by Groth [7] and Seo [8], which are constant move arguments with a communication proportional to the square root of the circuit size. Both of these protocols fit into the Ideal Linear Commitment model. The square-root communication cost comes from the fact that all of the wire values in the arithmetic circuit are arranged into a matrix, and the prover sends the verifier a commitment to each row, and a linear combination of the rows. Balancing the number of rows and columns in the matrix gives a square-root communication cost in total. This thesis also gives two arguments for arithmetic circuit satisfiability based on the discrete logarithm assumption. One has fewer rounds of interaction than these previous works, and the other has lower concrete communication costs, and fewer verification equations.

Using pairing-based cryptography instead of just relying on the discrete logarithm assumption, Groth [69] extended these techniques to give a zero-knowledge argument with a cube-root communication complexity. In this argument, the prover arranges the wire-values into a cuboid. Each slice of the cuboid is a matrix, and the prover commits to each row of each matrix using a Pedersen commitment, which collapses the cuboid of wire-values into a matrix of Pedersen commitments. Then,

the prover uses a pairing-based commitment scheme to collapse the matrix of Pedersen commitments into a vector of pairing-based commitments. The cube-root communication complexity comes from the fact that the prover has to send values to the verifier for each dimension of the cuboid, and balancing the dimensions gives the cube-root. The argument requires only a constant numbers of moves. It is the ability to use multiple related commitment schemes that allows the compression.

Our logarithmic protocol for arithmetic circuit satisfiability employs a similar concept, but works in a slightly different way. Wire-values can be arranged in a hypercube. All elements are committed to using a Pedersen commitment. At each step in the argument, the prover receives a random challenge from the verifier, and takes a random linear combination of lower dimensional hypercubes to reduce the dimension of the hypercube by one. This operation is compatible with the Pedersen commitment scheme, up to some correction factors, and results in a Pedersen commitment to fewer elements. It is the sequential interaction over many rounds, and the special properties of the Pedersen commitment scheme, that allows the compression. This led to the protocol given in [9], which shows that not only is the commitment scheme compatible with the compressing operation, but one can reduce a scalar-product check on the original elements to a check on the new compressed elements, leading to a highly efficient protocol for verifying the scalar product of committed vectors.

[1] observes that the original protocol uses separate Pedersen commitments for each input vector to each scalar product, and optimises the argument by giving a new argument where are values are contained in a single commitment. Hyrax [70] uses the same scalar-product argument in a different way, combined with a multivariate polynomial commitment scheme, to give efficient proofs for highly structured circuits, which achieve sub-linear proof size, linear prover time, and sub-linear verification time when giving proofs for a highly parallelisable circuit, or computing a batch proof for a large number of identical circuits. This approach performs best for circuits of low depth.

[40] adapts the square-root communication protocol of [9] to the post-quantum

setting, using commitments based on the hardness of the shortest vector problem for lattices. This work can be seen as the compilation of a particular ILC protocol into the new lattice-based setting. However, the commitment scheme used does not seem to admit the same special properties needed to replicate the argument with logarithmic communication complexity which is possible in the discrete logarithm setting. Like the compilation of ILC protocols based on hash functions, the lattice-based protocol requires an extra proof-of-knowledge on all commitments. The main difference between that protocol and our work is the adaptation of the same techniques to a new algebraic setting where the size of elements is important, and which is not a field, so that proof-techniques based on linear algebra become much more difficult to apply.

An exciting line of research [29, 30, 31, 32, 33, 34, 35, 36, 2] has developed many proposals for succinct non-interactive arguments (SNARGs) yielding pairing-based constructions where the arguments consist of a constant number of group elements. The arguments have a constant size, and a constant verification time, which allowed for effective recursive composition cite and exciting proof-carrying-data techniques. The disadvantage of these arguments is the super-linear computational complexity of the prover. The techniques have also been extended to give proofs with simulation-extractability [71, 72].

Like the ILC model's relationship with many discrete-logarithm-based protocols, the pairing-based protocols above can all be captured using the model of Linear Interactive Proofs and Linear PCPs [73]. In linear interactive proofs, the prover and verifier send vectors of field elements to one another. The verifier makes linear queries on a proof vector created by the prover. However, the prover can only send linear (or affine) transformations of the verifier's previously sent vectors, which distinguishes these systems from ILC protocols, in which the prover is allowed to perform more general computation. It is possible to convert ILC protocols into linear interactive proofs, and vice-versa, but the resulting protocols are usually inefficient, reflecting the fact that the models were tailored to different use-cases. Furthermore, linear interactive proofs usually involve a verifier of algebraic degree two. The ILC protocols presented in this work sometimes have a higher algebraic degree, so they

could not be converted into linear interactive proofs and compiled under the same methods.

The idealised linear interactive proofs are compiled into real arguments in pairing groups such as [36, 2] by creating a common reference string with all of the verifier's queries embedded into the exponents of group elements. Since it is believed to be difficult to manipulate group elements in these groups except by using the group operations, the resulting protocols are secure. However, due to the nature of the verifier's queries in the idealised proofs, the common reference strings are highly structured and must either be generated by a trusted third party or an expensive multiparty computation protocol. Other works such as [74] attempt to mitigate the problem. They give a protocol which updates common reference strings, so that the updated common reference strings are trustworthy, even if the old ones were not. [75, 76, 77] present secure multi-party computation protocols used to generate the common reference strings of pairing-based SNARKs.

Furthermore, non-falsifiable knowledge extractor assumptions are used to guarantee security. In contrast, the arguments we develop here are based solely on the discrete logarithm assumption, or collision resistant hash functions, and use a small common reference string which is independent of the circuit. This is because we choose to compile our idealised protocols under these alternative assumptions.

Bootle et al [78] used error-correcting codes and linear-time collision-resistant hash functions to give the first zero-knowledge proof and argument systems for arithmetic circuit satisfiability with constant computational overhead. The prover uses a linear number of field multiplications, and verification is even more efficient, requiring only a linear number of additions. They proposed the ILC model, which forms the basis of this work. Their methodology was also similar, desigining an ideal protocol with the correct security properties and compiling it into a real proof. The result hinges on choices of particularly efficient linear-time-computable hash functions and codes. This work includes similar compilations, modified to account for changes in the ILC model, optimisations specific to Reed-Solomon codes, and the discrete logarithm setting.

STARKs [79] give an argument with logarithmic communication costs, and logarithmic verification costs. Computational complexity for the prover is quasilinear, but the constants give this approach higher computational time for the prover than other cryptographic proof implementations such as [1, 70] for giving proofs about practical instances.

An another effective way to construct efficient zero-knowledge proofs is to follow the so-called MPC-in-the-head paradigm of [80]. This approach proved itself to give very efficient constructions both theoretically and practically. In this approach, when the prover wants to prove, for example, that a circuit is satisfiable, they simulate a secure multi-party computation protocol to evaluate the circuit on secret-shared inputs. They commit to the view of each party in the multi-party computation protocol. The verifier randomly selects some fraction of the views, and checks that they are consistent. ZKBOO [81] and subsequent optimisation ZKB++ [82] use hash functions to construct zero-knowledge arguments for the satisfiability of boolean circuits. Their communication complexity is linear in the circuit size, but the use of symmetric primitives gives good performances in practice.

Ligero [83] provides another implementation of the MPC-in-the-head paradigm and used techniques similar to [78] to construct sublinear arguments for arithmetic circuits. The approach used in Ligero is similar to the approach used in this work. One difference is that Ligero uses the multiplicative properties of Reed-Solomon codewords to help verify multiplications. This work does not require codewords to have any multiplicative properties.

Jawurek et al. [84] gave a different approach to zero-knowledge proofs derived from multiparty computation protocols, using garbled circuits.

All of the arguments mentioned above which rely on collision-resistant hash-functions for security only require a simple common reference string including a description of the hash-function. This makes them suitable for blockchain applications where a trusted-setup procedure is particularly undesirable.

Other works give a composite approach. [85] uses two approachs to interactive zero-knowledge. They use the garbled-circuit techniques of [84] to prove

non-algebraic statements, and algebraic protocols based on the discrete logarithm assumption and RSA assumption to prove algebraic statements. They leverage both techniques at the same time in order to construct efficient privacy-preserving credentials. [86] uses similar ideas to give non-interactive zero-knowledge proofs, this time by combining discrete-logarithm-based interactive zero-knowledge proofs made interactive through the Fiat-Shamir heuristic, and pairing-based SNARKs.

Another model for protocols is that of interactive oracle proofs, introduced in [87]. Interactive oracle proofs are interactive proofs between a prover, and a verifier, in which the verifier only has query access to the prover's messages. They simultaneously generalise interactive proofs and PCPs. Intuitively, one way to view interactive oracle proofs is as PCPs where the interaction between the prover and the verifier means that the prover only has to compute a small part of the PCP for the verifier to check. In fact, Ideal Linear Commitment protocols can also be seen as interactive oracle proofs with some extra restrictions.

As well as general proof proof systems, various works give protocols with low communication complexity for specific languages. For example, in a membership argument [88, 89], a prover demonstrates that a secret committed value λ is an element of a list $\mathscr{L} = \{\lambda_0, \ldots, \lambda_{N-1}\}$, without revealing any other information about λ. In a polynomial evaluation argument [90, 89], a prover demonstrates that a secret committed value v is the evaluation of a public polynomial $h(U)$ at another secret committed value u. In a range proof [91, 92], a prover demonstrates that a secret committed value a is an element of the interval $[A;B]$.

The goals of membership arguments are related to those of zero-knowledge sets [93]. Membership arguments allow a prover to commit to a secret value and show that it lies in a public set, without leaking information on the value. On the other hand, zero-knowledge sets allow the prover to commit to a secret set, and handle membership and non-membership queries in a verifiable manner, without leaking information on the set.

Herranz constructs attribute-based signatures [94] using what is essentially a set membership argument for multiple values. The argument relies only on the discrete

logarithm assumption, but the communication complexity is much high; linear in the size of the set. Camenisch et al. [95] also provide set membership proofs with logarithmic communication complexity, and Fauzi et al. [96] construct constant size arguments for more complex relations between committed sets. The latter two works both rely on pairing-based assumptions.

Groth and Kohlweiss [2], and a follow-up work [10] show that one can prove that one out of N commitments contain 0 with logarithmic communication complexity, based on the discrete logarithm assumption. For homomorphic commitments, it is easy to reduce the task of a membership argument to the task of checking that one commitment contains a zero, by dividing every commitment in the public set by the prover's own commitment. Both arguments work by arranging the values in the list into a tree, and having the prover commit to a sequence of bits which describe a path from the root of the tree to the leaf which is equal to the prover's own commitment. [10] optimises the protocol given in [2] by generalising to an n-ary tree rather than a binary one.

Range arguments can be seen as a special case of membership arguments, where $\mathscr{L}$ is simply a list of consecutive integers. Many are based on the strong RSA assumption, and use Lagrange's Four-Square Theorem. Couteau et al. show that this assumption can be replaced by an RSA-variant which is much closer to the standard RSA assumption [97]. Examples are [98, 92]. The work [99] gives an argument with sub-logarithmic communication complexity in the size of the list, which is comparable to the efficiency we achieve, and also relies on the hardness of the discrete logarithm problem, but uses pairings for verification.

Membership arguments also generalise arguments that a committed value lies in a linear subspace such as [100, 101, 102], which all make use of pairings. Peng [103] achieves a square-root complexity. Some existing protocols [11], [14] even achieve logarithmic communication complexity. Our single-value membership proof is an extension of the latter works where we reduce the number of commitments from logarithmic to constant.

Cryptographic accumulators,[104, 105, 106, 107], can also be used to give mem-

bership proofs. The members of a set are absorbed into a constant-size accumulated value. Witnesses for set-membership can then be generated and verified using the accumulated value. Efficient instantiations of accumulators exist and often rely on the Strong RSA assumption or pairing-based assumptions. An RSA modulus has to be $\frac{\lambda^3}{\text{polylog}\lambda}$ bits to provide security against factorisation using the General Number Field Sieve. Security of pairing-based schemes with constant embedding degree scale similarly due to sub-exponential algorithms for attacking the discrete logarithm problem in the target group. Furthermore, such schemes require a trusted setup. By contrast, when instantiating our proofs using Pedersen commitments or collision resistant hash functions, we only require commitments of size $O(\lambda)$ bits for security against discrete logarithm attacks in elliptic curve groups, or collision-finding attacks against the hash functions.

Some of the schemes can be adapted to give zero-knowledge arguments for non-membership, from a variety of settings. For example, [11, 103] also give non-membership arguments in the discrete logarithm setting. Accumulators that support non-membership arguments have been constructed, based on both pairing assumptions ([108]) and the strong RSA assumption ([109]).

Our polynomial commitment protocol is a key part of our zero-knowledge argument. Polynomial commitments were first introduced by Kate et al. [110], who give a construction using bilinear maps. The original construction has also been extended to the multivariate case [111, 112]. Libert et al. [113] also gave a construction relying on much simpler pairing-based assumptions. Our polynomial commitment protocol builds on the polynomial commitment protocol presented in [13], and gives a square-root communication complexity when instantiated with compact commitments. Later, Hyrax [70] gives a commitment scheme for multi-linear polynomials using the scalar-product argument of [9, 1], with a logarithmic communication complexity. The same idea can be incorporated into our batch protocol for low-degree polynomials, but does not improve asymptotic performance, so for ease of exposition, we do not discuss this.

Some zero-knowledge proofs and arguments use the idea of embedding many

statements into a single polynomial using Lagrange interpolation polynomials in a challenge x. The idea originates in the quadratic arithmetic programs of Gennaro et al. [114]. It was used in the context of interactive zero-knowledge arguments by Bayer [15]. The technique was originally applied to construct a Hadamard product argument and batched polynomial evaluation argument. Earlier work by Gennaro et al. [115] batches Schnorr proofs using simple powers of x.

Other batch arguments in the literature use methods from [116] and multiply different instances of the proof by small exponents before compressing the proofs together. This approach may be used to trade soundness for efficiency. The batch argument in this thesis proves and verifies the logical AND of many statements simultaneously. There are also batch proofs for OR statements [117], and k-out-of-N batch proofs [118]. Finally, Henry and Goldberg [118] define a notion of conciseness to characterise batch proofs.

Camenisch and Stadler [119] also propose a general framework of relations for zero-knowledge proofs based on the discrete logarithm assumption. Their notation is useful for describing large and complex statements. We take some inspiration from their notation, but use different notation since the ILC model describes general relations over fields, and values committed using a generic commitment scheme.

Chapter 3

Conclusion

In this thesis, we introduced the ILC model, modified it to bring it closer to real protocols, and gave compilations from ILC protocols to real zero-knowledge protocols based on hash functions and error-correcting codes. The compilations separated the cryptographic and non-cryptographic parts of the design process and simplify the protocol design process. In particular, designing our ILC protocols and proving them secure was a matter of applying linear algebra and simple lemmata about polynomial identity testing. Proving that the ILC protocols could be securely compiled into real arguments was more complicated, but was done once and for all, and the compilations can be reused for many ILC protocols in the future. We presented protocols with state-of-the-art communication complexity and round complexity, and showed that the ILC model is powerful enough to reason about both general NP-Complete statements like arithmetic circuit satisfiability. We also gave ILC protocols for simpler statements such as polynomial evaluation or range proofs, in a manner that leads to highly efficient protocols. This included the framing of general relations to capture a class of zero-knowledge protocols characterised by low-degree polynomials, formalising the techniques used in such protocols, and providing a generic protocol for reasoning about such relations, which can be used to give batch-proofs for many statements at the same time.

We found techniques used in interactive zero-knowledge protocols in the discrete logarithm setting, and rewrote many of those protocols in the ILC model. Thus, this work addresses our first goal, and shows that a great many discrete logarithm

arguments follow the same basic design paradigms. Surprisingly, the same style of protocol and design techniques extend beyond the discrete logarithm setting to another commitment scheme which is not homomorphic!

Using only this methodology, we were able to present some protocols with state-of-the-art communication complexity. Examples include a discrete-logarithm based polynomial evaluation argument, with a better asymptotic communication complexity than observed prevously, and a discrete-logarithm based membership argument, whose asymptotic communication complexity has improved constants over previous work, and which has highly tuneable parameters. These are of practical significance as they can be used as part of membership and non-membership arguments both in the designs of other primitives, like group and ring signatures, and in applications such as preventing double-spending in cryptocurrencies. Since ILC protocols can also be compiled based on hash functions and error-correcting codes, we also obtain some completely new arguments for polynomial evaluation and membership based on the existence of collision resistant hash functions. This shows that our second goal of designing efficient protocols was also addressed.

We also presented some extra techniques which fall outside the ILC model, namely, a recursive argument to show that committed values have a particular scalar product, and a field extension technique which boosts the soundness of ILC protocols over small fields. This is at once a strength and a weakness of using idealised communication models. Protocols inside such models are highly constrained, which makes them easier to design and reason about, but may also limit their performance and utility. The fact that the most efficient protocol in this thesis, the logarithmic-communication argument for arithmetic circuit satisfiability, does not lie within the main model of communication, is a limitation. However, once a suitable model has been identified, one can also try to design useful protocols by attempting to create protocols outside the model.

There are other zero-knowledge protocols [**?**], some based on lattices, and some based on the Strong RSA assumption, which seem to work on the same basis as ILC protocols. That is, the prover commits to certain vectors, and the verifier picks a

random challenge, and uses structured linear combinations of the committed vectors in a number of verification equations. Unlike in the ILC model, in which all elements belong to a field and the notion of size is not important, these settings require careful consideration of the size of committed elements to ensure zero-knowledge, and often for soundness too. The model falls short of capturing these protocols. Improving the model to take this into account, in particular for lattice-based protocols which may enjoy post-quantum security guarantees, is an attractive target for future research.

Another avenue that was not investigated is restricting the verifier's ILC queries. In all of the ILC protocols presented in this thesis, the coefficients of the verifier's linear queries are given by a linearly-independent set of polynomials evaluated at uniformly random challenges chosen by the verifier. The queries have a carefully chosen algebraic structure. For every protocol that we give, the query matrix appears to be a form of strongly universal hash function. The compilation from ILC protocols to discrete-logarithm based protocols requires restrictions on the rank and dimension of the matrix, and that a related system of linear equations can be solved. These conditions are treated in an ad-hoc manner outside of the proofs that the protocols are secure in the idealised model. There is still a gap between the model and the compiled protocols, and the communication model can be refined further. One could hope that such strong algebraic restrictions lead to interesting results, such as lower bounds on the communication complexity of ILC protocols, as linear algebra is an old discipline with many results that one could hope to apply to the structure of the query matrices.

Bibliography

[1] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. Bulletproofs: Short proofs for confidential transactions and more. *IACR Cryptology ePrint Archive*, 2017:1066, 2017.

[2] Jens Groth and Markulf Kohlweiss. One-out-of-many proofs: Or how to leak a secret and spend a coin. In *Advances in Cryptology – EUROCRYPT 2015*, page 764, 2014.

[3] Stephanie Bayer and Jens Groth. Zero-knowledge argument for polynomial evaluation with application to blacklists. volume 7881, pages 646–663, 2013.

[4] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). pages 291–304. ACM, 1985.

[5] Claus-Peter Schnorr. Efficient signature generation by smart cards. *J. Cryptology*, 4(3):161–174, 1991.

[6] Ronald Cramer and Ivan Damgård. Zero-knowledge proofs for finite field arithmetic; or: Can zero-knowledge be for free? pages 424–441. Springer, 1998.

[7] Jens Groth. Linear algebra with sub-linear zero-knowledge arguments. In *Advances in Cryptology – CRYPTO 2009*, pages 192–208, 2009.

[8] Jae Hong Seo. Round-efficient sub-linear zero-knowledge arguments for linear algebra. In *Public Key Cryptography - PKC 2011*, pages 387–402, 2011.

[9] Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Jens Groth, and Christophe Petit. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II*, pages 327–357, 2016.

[10] Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Essam Ghadafi, Jens Groth, and Christophe Petit. Short Accountable Ring Signatures Based on DDH. In *ESORICS*, pages 243–265, 2013.

[11] Stephanie Bayer and Jens Groth. Zero-Knowledge Argument for Polynomial Evaluation with Application to Blacklists. In *EUROCRYPT*, pages 646–663, 2013.

[12] Jonathan Bootle and Jens Groth. Efficient batch zero-knowledge arguments for low degree polynomials. In *Public-Key Cryptography - PKC 2018 - 21st IACR International Conference on Practice and Theory of Public-Key Cryptography, Rio de Janeiro, Brazil, March 25-29, 2018, Proceedings, Part II*, pages 561–588, 2018.

[13] Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Jens Groth, and Christophe Petit. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In *EUROCRYPT*, pages 327–357, 2016.

[14] Jens Groth and Markulf Kohlweiss. One-out-of-Many Proofs: Or How to Leak a Secret and Spend a Coin. In *EUROCRYPT*, pages 253–280, 2015.

[15] Stephanie Bayer. *Practical zero-knowledge Protocols based on the discrete logarithm Assumption*. PhD thesis, University College London, 2014.

[16] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE Symposium on Security and Privacy, SP 2018,*

Proceedings, 21-23 May 2018, San Francisco, California, USA, pages 315–334, 2018.

[17] Stephanie Bayer and Jens Groth. Efficient zero-knowledge argument for correctness of a shuffle. In *Advances in Cryptology – EUROCRYPT 2012*, pages 263–280, 2012.

[18] Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Essam Ghadafi, and Jens Groth. Foundations of Fully Dynamic Group Signatures. In *ACNS*, pages 117–136, 2016.

[19] Mihir Bellare, Haixia Shi, and Chong Zhang. Foundations of group signatures: The case of dynamic groups. In *Topics in Cryptology - CT-RSA 2005, The Cryptographers' Track at the RSA Conference 2005, San Francisco, CA, USA, February 14-18, 2005, Proceedings*, pages 136–153, 2005.

[20] Aggelos Kiayias and Moti Yung. Secure scalable group signature with dynamic joins and separable authorities. *IJSN*, 1(1/2):24–45, 2006.

[21] Jonathan Bootle, Mehdi Tibouchi, and Keita Xagawa. Cryptanalysis of compact-lwe. In *Topics in Cryptology - CT-RSA 2018 - The Cryptographers' Track at the RSA Conference 2018, San Francisco, CA, USA, April 16-20, 2018, Proceedings*, pages 80–97, 2018.

[22] Dongxi Liu, Nan Li, Jongkil Kim, and Surya Nepal. Compact-lwe: Enabling practically lightweight public key encryption for leveled iot device authentication. *IACR Cryptology ePrint Archive*, 2017:685, 2017.

[23] Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal gaussians. In *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, pages 40–56, 2013.

[24] Thomas Espitau, Pierre-Alain Fouque, Benoît Gérard, and Mehdi Tibouchi. Side-channel attacks on BLISS lattice-based signatures: Exploiting branch

tracing against strongswan and electromagnetic emanations in microcontrollers. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 1857–1874, 2017.

[25] Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, and Jens Groth. Efficient zero-knowledge proof systems. In *Foundations of Security Analysis and Design VIII - FOSAD 2014/2015/2016 Tutorial Lectures*, pages 1–31, 2016.

[26] Jens Groth. Short non-interactive zero-knowledge proofs. volume 6477, pages 341–358, 2010.

[27] Jens Groth, Rafail Ostrovsky, and Amit Sahai. New techniques for noninteractive zero-knowledge. *Journal of the ACM*, 59(3):11:1–11:35, 2012.

[28] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-dnf formulas on ciphertexts. In *Theory of Cryptography, Second Theory of Cryptography Conference, TCC 2005, Cambridge, MA, USA, February 10-12, 2005, Proceedings*, pages 325–341, 2005.

[29] Jens Groth. Short pairing-based non-interactive zero-knowledge arguments. pages 321–340. Springer, 2010.

[30] Helger Lipmaa. Progression-free sets and sublinear pairing-based noninteractive zero-knowledge arguments. In *Theory of Cryptography Conference – TCC 2012*, pages 169–189, 2012.

[31] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. In *Innovations in Theoretical Computer Science – ITCS 2012*, pages 326–349, 2012.

[32] Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic span programs and succinct NIZKs without PCPs. pages 626–645. Springer, 2013.

[33] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. Recursive composition and bootstrapping for SNARKS and proof-carrying data. In *Symposium on Theory of Computing Conference – TCC 2013*, pages 111–120, 2013.

[34] Bryan Parno, Jon Howell, Craig Gentry, and Mariana Raykova. Pinocchio: Nearly practical verifiable computation. In *IEEE Symposium on Security and Privacy*, pages 238–252. IEEE Computer Society, 2013.

[35] Eli Ben-Sasson, Alessandro Chiesa, Daniel Genkin, Eran Tromer, and Madars Virza. SNARKs for C: verifying program executions succinctly and in Zero Knowledge. In *Advances in Cryptology – CRYPTO 2013*, pages 90–108, 2013.

[36] Eli Ben-Sasson, Alessandro Chiesa, Eran Tromer, and Madars Virza. Succinct non-interactive zero knowledge for a von neumann architecture. In *USENIX Security Symposium 2014*, pages 781–796, 2014.

[37] Jonathan Bootle, Andrea Cerulli, Essam Ghadafi, Jens Groth, Mohammad Hajiabadi, and Sune K. Jakobsen. Linear-time zero-knowledge proofs for arithmetic circuit satisfiability. In *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part III*, pages 336–365, 2017.

[38] Jens Groth. Linear algebra with sub-linear zero-knowledge arguments. pages 192–208. Springer, 2009.

[39] Stephanie Bayer and Jens Groth. Zero-knowledge argument for polynomial evaluation with application to blacklists. In *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, pages 646–663, 2013.

[40] Carsten Baum, Jonathan Bootle, Andrea Cerulli, Rafaël del Pino, Jens Groth, and Vadim Lyubashevsky. Sub-linear lattice-based zero-knowledge arguments for arithmetic circuits. In *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part II*, pages 669–699, 2018.

[41] Ronald Cramer, Ivan Damgård, and Valerio Pastro. On the amortized complexity of zero knowledge protocols for multiplicative relations. In *Information Theoretic Security - 6th International Conference, ICITS 2012, Montreal, QC, Canada, August 15-17, 2012. Proceedings*, pages 62–79, 2012.

[42] Ronald Cramer, Ivan Damgård, and Marcel Keller. On the amortized complexity of zero-knowledge protocols. *J. Cryptology*, 27(2):284–316, 2014.

[43] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proofs. *SIAM Journal on Computing*, 18(1):186–208, 1989.

[44] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM*, 38(3):691–729, 1991.

[45] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity and a methodology of cryptographic protocol design (extended abstract). In *27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986*, pages 174–187, 1986.

[46] Russell Impagliazzo and Moti Yung. Direct minimum-knowledge computations. In *Advances in Cryptology - CRYPTO '87, A Conference on the Theory and Applications of Cryptographic Techniques, Santa Barbara, California, USA, August 16-20, 1987, Proceedings*, pages 40–51, 1987.

[47] Adi Shamir. Ip=pspace. In *31st Annual Symposium on Foundations of Computer Science, St. Louis, Missouri, USA, October 22-24, 1990, Volume I*, pages 11–15, 1990.

[48] Uriel Feige, Amos Fiat, and Adi Shamir. Zero knowledge proofs of identity. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, New York, USA*, pages 210–217, 1987.

[49] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications. pages 103–112. ACM, 1988.

[50] Gilles Brassard, David Chaum, and Claude Crèpeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37(2):156–189, 1988.

[51] Silvio Micali. Computationally sound proofs. *SIAM Journal on Computing*, 30(4):1253–1298, 2000.

[52] Craig Gentry, Jens Groth, Yuval Ishai, Chris Peikert, Amit Sahai, and Adam Smith. Using fully homomorphic hybrid encryption to minimize non-interative zero-knowledge proofs. *Journal of Cryptology*, pages 1–24, 2014.

[53] Oded Goldreich and Johan Håstad. On the complexity of interactive proofs with bounded communication. *Information Processing Letters*, 67(4):205–214, 1998.

[54] Oded Goldreich, Salil P. Vadhan, and Avi Wigderson. On interactive proofs with a laconic prover. *Computational Complexity*, 11(1-2):1–53, 2002.

[55] Joe Kilian. A note on efficient zero-knowledge proofs and arguments. pages 723–732. ACM, 1992.

[56] Yuval Ishai, Eyal Kushilevitz, and Rafail Ostrovsky. Efficient arguments without short pcps. In *22nd Annual IEEE Conference on Computational Complexity (CCC 2007), 13-16 June 2007, San Diego, California, USA*, pages 278–291, 2007.

[57] Mihir Bellare, Silvio Micali, and Rafail Ostrovsky. The (true) complexity of statistical zero knowledge. In *Proceedings of the 22nd Annual ACM*

Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA, pages 494–502, 1990.

[58] Rafail Ostrovsky, Ramarathnam Venkatesan, and Moti Yung. Interactive hashing simplifies zero-knowledge protocol design. In *Advances in Cryptology - EUROCRYPT '93, Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, May 23-27, 1993, Proceedings*, pages 267–273, 1993.

[59] Tatsuaki Okamoto. On relationships between statistical zero-knowledge proofs. *J. Comput. Syst. Sci.*, 60(1):47–108, 2000.

[60] Ivan Damgård. Interactive hashing can simplify zero-knowledge protocol design without computational assumptions (extended abstract). In *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, pages 100–109, 1993.

[61] Oded Goldreich, Amit Sahai, and Salil P. Vadhan. Honest-verifier statistical zero-knowledge equals general statistical zero-knowledge. In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 399–408, 1998.

[62] Ivan Damgård. Efficient concurrent zero-knowledge in the auxiliary string model. Springer, 2000.

[63] Jens Groth. *Honest verifier zero-knowledge arguments applied*. BRICS, 2004.

[64] Daniele Micciancio and Erez Petrank. Simulatable commitments and efficient concurrent zero-knowledge. pages 140–159. Springer, 2003.

[65] Oded Goldreich, Amit Sahai, and Salil Vadhan. Honest-verifier statistical zero-knowledge equals general statistical zero-knowledge. pages 399–408. ACM, 1998.

[66] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings*, pages 186–194, 1986.

[67] Mihir Bellare and P Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. *Proceedings of the 1st ACM conference on . . .*, (November 1993):1–21, 1993.

[68] Louis C. Guillou and Jean-Jacques Quisquater. A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory. In *Advances in Cryptology - EUROCRYPT '88, Workshop on the Theory and Application of of Cryptographic Techniques, Davos, Switzerland, May 25-27, 1988, Proceedings*, pages 123–128, 1988.

[69] Jens Groth. Efficient zero-knowledge arguments from two-tiered homomorphic commitments. In *Advances in Cryptology – ASIACRYPT 2009*, pages 431–448, 2009.

[70] Riad S. Wahby, Ioanna Tzialla, Abhi Shelat, Justin Thaler, and Michael Walfish. Doubly-efficient zksnarks without trusted setup. In *2018 IEEE Symposium on Security and Privacy, SP 2018, Proceedings, 21-23 May 2018, San Francisco, California, USA*, pages 926–943, 2018.

[71] Jens Groth and Mary Maller. Snarky signatures: Minimal signatures of knowledge from simulation-extractable snarks. In *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part II*, pages 581–612, 2017.

[72] Sean Bowe and Ariel Gabizon. Making groth's zk-snark simulation extractable in the random oracle model. *IACR Cryptology ePrint Archive*, 2018:187, 2018.

[73] Nir Bitansky, Alessandro Chiesa, Yuval Ishai, Rafail Ostrovsky, and Omer Paneth. Erratum: Succinct non-interactive arguments via linear interactive proofs. Springer, 2013.

[74] Jens Groth, Markulf Kohlweiss, Mary Maller, Sarah Meiklejohn, and Ian Miers. Updatable and universal common reference strings with applications to zk-snarks. In *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part III*, pages 698–728, 2018.

[75] Eli Ben-Sasson, Alessandro Chiesa, Matthew Green, Eran Tromer, and Madars Virza. Secure sampling of public parameters for succinct zero knowledge proofs. In *2015 IEEE Symposium on Security and Privacy, SP 2015, San Jose, CA, USA, May 17-21, 2015*, pages 287–304, 2015.

[76] Sean Bowe, Ariel Gabizon, and Matthew D. Green. A multi-party protocol for constructing the public parameters of the pinocchio zk-snark. *IACR Cryptology ePrint Archive*, 2017:602, 2017.

[77] Sean Bowe, Ariel Gabizon, and Ian Miers. Scalable multi-party computation for zk-snark parameters in the random beacon model. *IACR Cryptology ePrint Archive*, 2017:1050, 2017.

[78] Jonathan Bootle, Andrea Cerulli, Essam Ghadafi, Jens Groth, Mohammad Hajiabadi, and Sune K. Jakobsen. Linear-time zero-knowledge proofs for arithmetic circuit satisfiability. In *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part III*, pages 336–365, 2017.

[79] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Scalable, transparent, and post-quantum secure computational integrity. *IACR Cryptology ePrint Archive*, 2018:46, 2018.

[80] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Zero-knowledge from secure multiparty computation. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007*, pages 21–30, 2007.

[81] Irene Giacomelli, Jesper Madsen, and Claudio Orlandi. Zkboo: Faster zero-knowledge for boolean circuits. In *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016.*, pages 1069–1083, 2016.

[82] Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, and Greg Zaverucha. Post-quantum zero-knowledge and signatures from symmetric-key primitives. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 1825–1842, 2017.

[83] Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. Ligero: Lightweight sublinear arguments without a trusted setup. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 2087–2104, 2017.

[84] Marek Jawurek, Florian Kerschbaum, and Claudio Orlandi. Zero-knowledge using garbled circuits: how to prove non-algebraic statements efficiently. In *2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4-8, 2013*, pages 955–966, 2013.

[85] Melissa Chase, Chaya Ganesh, and Payman Mohassel. Efficient zero-knowledge proof of algebraic and non-algebraic statements with applications to privacy preserving credentials. In *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part III*, pages 499–530, 2016.

[86] Shashank Agrawal, Chaya Ganesh, and Payman Mohassel. Non-interactive zero-knowledge proofs for composite statements. In *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part III*, pages 643–673, 2018.

[87] Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. Interactive oracle proofs. pages 31–60. Springer, 2016.

[88] Emmanuel Bresson and Jacques Stern. Efficient revocation in group signatures. In *PKC*, pages 190–206, 2001.

[89] Stefan Brands, Lisa Demuynck, and Bart De Decker. A practical system for globally revoking the unlinkable pseudonyms of unknown users. In *ACISP*, volume 4586, pages 400–415, 2007.

[90] Eiichiro Fujisaki and Tatsuaki Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. volume 1294, pages 16–30, 1997.

[91] Fabrice Boudot. Efficient proofs that a committed number lies in an interval. In *EUROCRYPT*, pages 431–444, 2002.

[92] Helger Lipmaa. On diophantine complexity and statistical zero-knowledge arguments. In *ASIACRYPT*, pages 398–415, 2003.

[93] Silvio Micali, Michael O. Rabin, and Joe Kilian. Zero-knowledge sets. In *FOCS*, pages 80–91, 2003.

[94] Javier Herranz. Attribute-based versions of schnorr and elgamal. *Appl. Algebra Eng. Commun. Comput.*, 27(1):17–57, 2016.

[95] Jan Camenisch and Rafik Chaabouni. Efficient protocols for set membership and range proofs. *Advances in Cryptology-ASIACRYPT . . .*, 2008.

[96] Prastudy Fauzi, Helger Lipmaa, and Bingsheng Zhang. Efficient Non-Interactive Zero Knowledge Arguments for Set Operations. In *Financial Cryptography and Data Security*, pages 216–233, 2014.

[97] Geoffroy Couteau, Thomas Peters, and David Pointcheval. Removing the strong RSA assumption from arguments over the integers. In *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part II*, pages 321–350, 2017.

[98] Jens Groth. *Honest verifier zero-knowledge arguments applied*. PhD thesis, Aarhus University, 2004.

[99] Rafik Chaabouni, Helger Lipmaa, and Abhi Shelat. Additive combinatorics and discrete logarithm based range protocols. In *ACISP*, volume LNCS 6168, pages 336–351, 2010.

[100] Charanjit Jutla and Arnab Roy. Shorter {Q}uasi-{A}daptive {NIZK} {P}roofs for {L}inear {S}ubspaces. In *ASIACRYPT*, volume LNCS 8269, pages 1–20, 2013.

[101] Charanjit S. Jutla and Arnab Roy. Switching lemma for bilinear tests and constant-size NIZK proofs for linear subspaces. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8617 LNCS(PART 2):295–312, 2014.

[102] Eike Kiltz and Hoeteck Wee. Quasi-adaptive NIZK for linear subspaces revisited. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9057(339563):101–128, 2015.

[103] Kun Peng. A general, flexible and efficient proof of inclusion and exclusion. *Trusted Systems*, pages 33–48, 2012.

[104] Josh Benaloh and Michael de Mare. One-way accumulators: A decentralized alternative to digital signatures. *Advances in Cryptology, EUROCRYPT93*, 1994.

[105] Lan Nguyen. Accumulators from bilinear pairings and applications to ID-based ring signatures and group membership revocation. In *CT-RSA*, pages 275–292, 2005.

[106] Jan Camenisch, Markulf Kohlweiss, and Claudio Soriente. An accumulator based on bilinear maps and efficient revocation for anonymous credentials. *Public Key Cryptography - PKC*, 2009.

[107] Jan Camenisch and Anna Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In *CRYPTO*, pages 61–76, 2002.

[108] Ivan Damgård and Nikos Triandopoulos. Supporting Non-membership Proofs with Bilinear-map Accumulators. IACR ePrint archive report 538, 2008.

[109] Jiangtao Li, Ninghui Li, and Rui Xue. Universal Accumulators with Efficient Nonmembership Proofs. *Proceedings of the 5th international conference on Applied Cryptography and Network Security (ACNS)*, pages 253–269, 2007.

[110] Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg. Constant-size commitments to polynomials and their applications. In *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings*, pages 177–194, 2010.

[111] Charalampos Papamanthou, Elaine Shi, and Roberto Tamassia. Signatures of correct computation. In *Theory of Cryptography - 10th Theory of Cryptography Conference, TCC 2013, Tokyo, Japan, March 3-6, 2013. Proceedings*, pages 222–242, 2013.

[112] Yupeng Zhang, Daniel Genkin, Jonathan Katz, Dimitrios Papadopoulos, and Charalampos Papamanthou. vsql: Verifying arbitrary SQL queries over dynamic outsourced databases. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pages 863–880, 2017.

[113] Benoît Libert, Somindu C. Ramanna, and Moti Yung. Functional commitment schemes: From polynomial commitments to pairing-based accumulators from simple assumptions. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 30:1–30:14, 2016.

[114] Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic Span Programs and Succinct NIZKs without PCPs. In *EUROCRYPT*, pages 626–645, 2013.

[115] Roario Gennaro, Darren Leigh, Ravi Sundaram, and William Yerazunis. Batching Schnorr identification scheme with applications to privacy-preserving authorization and low-bandwidth communication devices. In *ASIACRYPT*, volume LNCS 3329, pages 276–292, 2004.

[116] Mihir Bellare, Juan A. Garay, and Tal Rabin. Batch Verification with Applications to Cryptography and Checking. In *EUROCRYPT*, pages 236–250, 1998.

[117] Kun Peng and Feng Bao. Batch ZK Proof and Verification of OR Logic. In *Inscrypt*, volume LNCS 5487, pages 141–156, 2008.

[118] Ryan Henry and Ian Goldberg. Batch proofs of partial knowledge. In *ACNS*, pages 502–517, 2013.

[119] Jan Camenisch and Markus Stadler. Proof systems for general statements about discrete logarithms. Technical Report 260, ETH Zurich, 1997.

[120] Nicholas Pippenger. On the evaluation of powers and monomials. *SIAM J. Comput.*, 9(2):230–250, 1980.