Inner Product Arguments

Mary Maller
Ethereum Foundation

What is an Inner Product Argument?

$Com_A = Commit(a_1, a_2, a_3, a_4)$

$Com_B = Commit(b_1, b_2, b_3, b_4)$

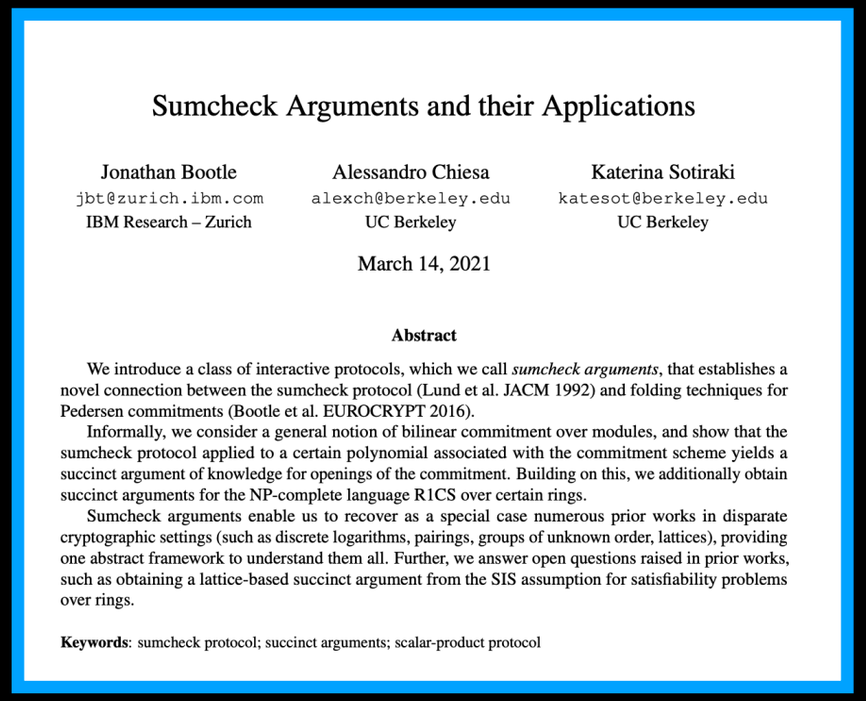
c

- Prove knowledge openings such that: $c = a_1b_1 + a_2b_2 + a_3b_3 + a_4b_4$

What is an Inner Product Argument?

$Com_A = Commit(a_1, a_2, a_3, a_4)$ | $Com_B = Commit(b_1, b_2, b_3, b_4)$ | c

- Prove knowledge openings such that: $c = a_1b_1 + a_2b_2 + a_3b_3 + a_4b_4$
- Can be seen as a special case of the SumCheck protocol.



Sumcheck Arguments and their Applications

Jonathan Bootle
jbt@zurich.ibm.com
IBM Research – Zurich

Alessandro Chiesa
alexch@berkeley.edu
UC Berkeley

Katerina Sotiraki
katesot@berkeley.edu
UC Berkeley

March 14, 2021

Abstract

We introduce a class of interactive protocols, which we call *sumcheck arguments*, that establishes a novel connection between the sumcheck protocol (Lund et al. JACM 1992) and folding techniques for Pedersen commitments (Bootle et al. EUROCRYPT 2016).

Informally, we consider a general notion of bilinear commitment over modules, and show that the sumcheck protocol applied to a certain polynomial associated with the commitment scheme yields a succinct argument of knowledge for openings of the commitment. Building on this, we additionally obtain succinct arguments for the NP-complete language R1CS over certain rings.

Sumcheck arguments enable us to recover as a special case numerous prior works in disparate cryptographic settings (such as discrete logarithms, pairings, groups of unknown order, lattices), providing one abstract framework to understand them all. Further, we answer open questions raised in prior works, such as obtaining a lattice-based succinct argument from the SIS assumption for satisfiability problems over rings.

Keywords: sumcheck protocol; succinct arguments; scalar-product protocol

Relates Sumcheck Arguments to Inner Product Arguments

- Bootle et al. compute a log sized argument in the discrete-logarithm setting without trusted setup.
- Key insight is a recursive “folding argument.”

Efficient Zero-Knowledge Arguments for Arithmetic Circuits in the Discrete Log Setting[†*]

Jonathan Bootle[1], Andrea Cerulli[1], Pyrros Chaidos[1**], Jens Groth[1], and Christophe Petit[2]

[1] University College London
{jonathan.bootle.14,andrea.cerulli.13,pyrros.chaidos.10,j.groth}@ucl.ac.uk
[2] University of Oxford
christophe.petit@maths.ox.ac.uk

Abstract. We provide a zero-knowledge argument for arithmetic circuit satisfiability with a communication complexity that grows logarithmically in the size of the circuit. The round complexity is also logarithmic and for an arithmetic circuit with fan-in 2 gates the computation of the prover and verifier is linear in the size of the circuit. The soundness of our argument relies solely on the well-established discrete logarithm assumption in prime order groups.
At the heart of our new argument system is an efficient zero-knowledge argument of knowledge of openings of two Pedersen multicommitments satisfying an inner product relation, which is of independent interest. The inner product argument requires logarithmic communication, logarithmic interaction and linear computation for both the prover and the verifier.
We also develop a scheme to commit to a polynomial and later reveal the evaluation at an arbitrary point, in a verifiable manner. This is used to build an optimized version of the constant round square root complexity argument of Groth (CRYPTO 2009), which reduces both communication and round complexity.

Keywords: Sigma-protocol, zero-knowledge argument, arithmetic circuit, discrete logarithm assumption.

Inner product argument for Pedersen Commitments

Folding Argument

$Com_A = Commit(a_1, a_2, a_3, a_4)$

$Com_B = Commit(b_1, b_2, b_3, b_4)$

c

Prove that $c = a_1b_1 + a_2b_2 + a_3b_3 + a_4b_4$

Folding Argument

$Com_A = Commit(a_1, a_2, a_3, a_4)$	$Com_B = Commit(b_1, b_2, b_3, b_4)$	c

Prove that $c = a_1b_1 + a_2b_2 + a_3b_3 + a_4b_4$

$Com'_A = Commit(a'_1, a'_2)$	$Com'_B = Commit(b'_1, b'_2)$	c'

Prove that if $c' = a'_1b'_1 + a'_2b'_2$ **then** $c = a_1b_1 + a_2b_2 + a_3b_3 + a_4b_4$

Folding Argument

$Com_A = Commit(a_1, a_2, a_3, a_4)$ | $Com_B = Commit(b_1, b_2, b_3, b_4)$ | c

Prove that $c = a_1b_1 + a_2b_2 + a_3b_3 + a_4b_4$

$Com'_A = Commit(a'_1, a'_2)$ | $Com'_B = Commit(b'_1, b'_2)$ | c'

Prove that if $c' = a'_1b'_1 + a'_2b'_2$ **then** $c = a_1b_1 + a_2b_2 + a_3b_3 + a_4b_4$

$Com''_A = Commit(a''_1)$ | $Com''_B = Commit(b''_1)$ | c''

Prove that if $c'' = a''_1b''_1$ **then** $c' = a'_1b'_1 + a'_2b'_2$

Folding Argument

$Com_A = Commit(a_1, a_2, a_3, a_4)$	$Com_B = Commit(b_1, b_2, b_3, b_4)$	c

Prove that $c = a_1b_1 + a_2b_2 + a_3b_3 + a_4b_4$

$Com'_A = Commit(a'_1, a'_2)$	$Com'_B = Commit(b'_1, b'_2)$	c'

Prove that if $c' = a'_1b'_1 + a'_2b'_2$ **then** $c = a_1b_1 + a_2b_2 + a_3b_3 + a_4b_4$

$Com''_A = Commit(a''_1)$	$Com''_B = Commit(b''_1)$	c''

Prove that if $c'' = a''_1b''_1$ **then** $c' = a'_1b'_1 + a'_2b'_2$

Open a''_1 **and** b''_1

- Bunz et al. improve efficiency by 3x and apply inner products to range proofs.

$$Com_A = Commit(a_1, a_2, a_3, a_4)$$

$$Com_B = Commit(b_1, b_2, b_3, b_4)$$

$$c$$

$$Com_{A,B,C} = Commit(a_1, a_2, a_3, a_4; b_1, b_2, b_3, b_4; c)$$

Bulletproofs: Short Proofs for Confidential Transactions and More

Benedikt Bünz*[1], Jonathan Bootle†[2], Dan Boneh‡[1],
Andrew Poelstra§[3], Pieter Wuille¶[3], and Greg Maxwell‖

[1]Stanford University
[2]University College London
[3]Blockstream

Full Version**

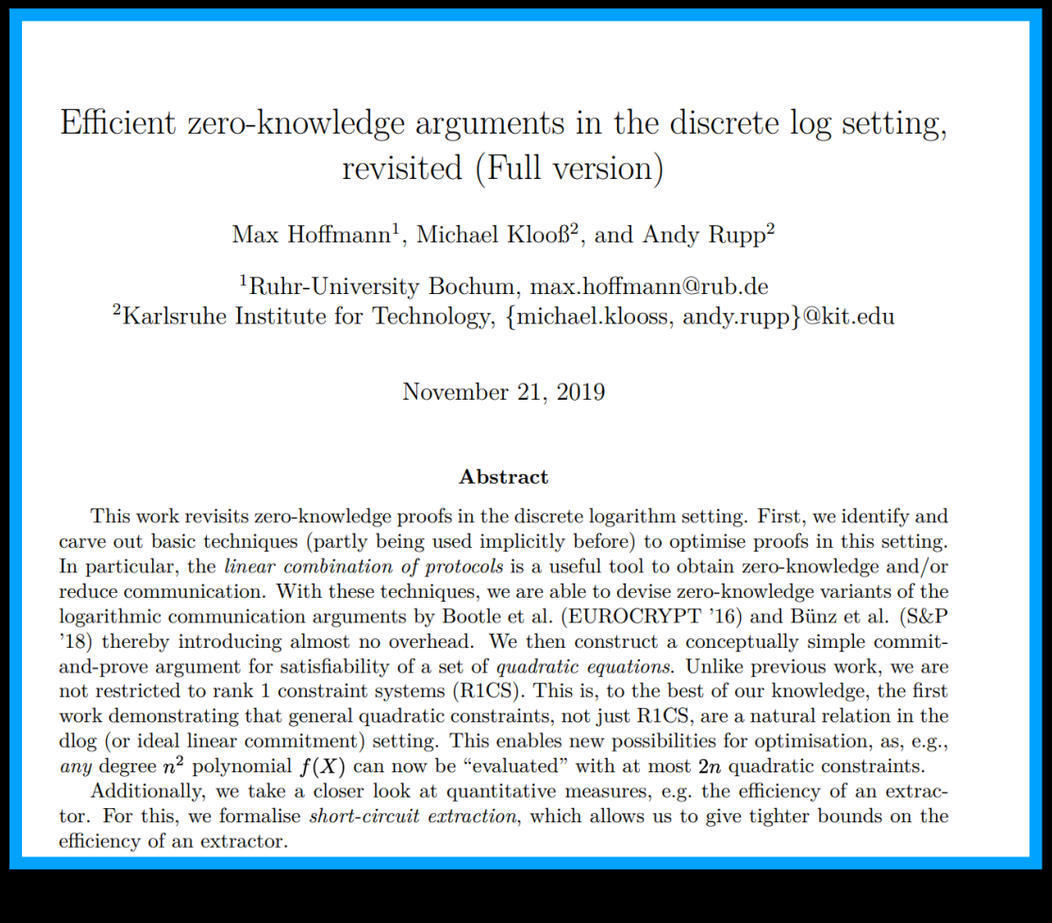
Abstract

We propose Bulletproofs, a new non-interactive zero-knowledge proof protocol with very short proofs and without a trusted setup; the proof size is only logarithmic in the witness size. Bulletproofs are especially well suited for efficient range proofs on committed values: they enable proving that a committed value is in a range using only $2\log_2(n) + 9$ group and field elements, where n is the bit length of the range. Proof generation and verification times are linear in n.

Bulletproofs greatly improve on the linear (in n) sized range proofs in existing proposals for confidential transactions in Bitcoin and other cryptocurrencies. Moreover, Bulletproofs supports aggregation of range proofs, so that a party can prove that m commitments lie in a given range by providing only an additive $O(\log(m))$ group elements over the length of a *single* proof. To aggregate proofs from multiple parties, we enable the parties to generate a single proof without revealing their inputs to each other via a simple multi-party computation (MPC) protocol for constructing Bulletproofs. This MPC protocol uses either a constant number of rounds and linear communication, or a logarithmic number of rounds and logarithmic communication. We show that verification time, while asymptotically linear, is very efficient in practice. Moreover, the verification of multiple Bulletproofs can be batched for further speed-up. Concretely, the marginal time to verify an aggregation of 16 range proofs is about the same as the time to verify 16 ECDSA signatures.

Bulletproofs build on the techniques of Bootle et al. (EUROCRYPT 2016). Beyond range proofs, Bulletproofs provide short zero-knowledge proofs for general arithmetic circuits while only relying on the discrete logarithm assumption and without requiring a trusted setup. We discuss many applications that would benefit from Bulletproofs, primarily in the area of cryptocurrencies. The efficiency of Bulletproofs is particularly well suited for the distributed and trustless nature of blockchains.

Inner product argument for Pedersen Commitments



Efficient zero-knowledge arguments in the discrete log setting, revisited (Full version)

Max Hoffmann[1], Michael Klooß[2], and Andy Rupp[2]

[1]Ruhr-University Bochum, max.hoffmann@rub.de
[2]Karlsruhe Institute for Technology, {michael.klooss, andy.rupp}@kit.edu

November 21, 2019

Abstract

This work revisits zero-knowledge proofs in the discrete logarithm setting. First, we identify and carve out basic techniques (partly being used implicitly before) to optimise proofs in this setting. In particular, the *linear combination of protocols* is a useful tool to obtain zero-knowledge and/or reduce communication. With these techniques, we are able to devise zero-knowledge variants of the logarithmic communication arguments by Bootle et al. (EUROCRYPT '16) and Bünz et al. (S&P '18) thereby introducing almost no overhead. We then construct a conceptually simple commit-and-prove argument for satisfiability of a set of *quadratic equations*. Unlike previous work, we are not restricted to rank 1 constraint systems (R1CS). This is, to the best of our knowledge, the first work demonstrating that general quadratic constraints, not just R1CS, are a natural relation in the dlog (or ideal linear commitment) setting. This enables new possibilities for optimisation, as, e.g., *any* degree n^2 polynomial $f(X)$ can now be "evaluated" with at most $2n$ quadratic constraints.

Additionally, we take a closer look at quantitative measures, e.g. the efficiency of an extractor. For this, we formalise *short-circuit extraction*, which allows us to give tighter bounds on the efficiency of an extractor.

- Hoffmann et al. show how to randomise an inner product argument such that it is zero-knowledge.

Inner product argument with zero-knowledge

Efficient zero-knowledge arguments in the discrete log setting, revisited (Full version)

Max Hoffmann[1], Michael Klooß[2], and Andy Rupp[2]

[1]Ruhr-University Bochum, max.hoffmann@rub.de
[2]Karlsruhe Institute for Technology, {michael.klooss, andy.rupp}@kit.edu

November 21, 2019

Abstract

This work revisits zero-knowledge proofs in the discrete logarithm setting. First, we identify and carve out basic techniques (partly being used implicitly before) to optimise proofs in this setting. In particular, the *linear combination of protocols* is a useful tool to obtain zero-knowledge and/or reduce communication. With these techniques, we are able to devise zero-knowledge variants of the logarithmic communication arguments by Bootle et al. (EUROCRYPT '16) and Bünz et al. (S&P '18) thereby introducing almost no overhead. We then construct a conceptually simple commit-and-prove argument for satisfiability of a set of *quadratic equations*. Unlike previous work, we are not restricted to rank 1 constraint systems (R1CS). This is, to the best of our knowledge, the first work demonstrating that general quadratic constraints, not just R1CS, are a natural relation in the dlog (or ideal linear commitment) setting. This enables new possibilities for optimisation, as, e.g., *any* degree n^2 polynomial $f(X)$ can now be "evaluated" with at most $2n$ quadratic constraints.

Additionally, we take a closer look at quantitative measures, e.g. the efficiency of an extractor. For this, we formalise *short-circuit extraction*, which allows us to give tighter bounds on the efficiency of an extractor.

Inner product argument with zero-knowledge

- Hoffmann et al. show how to randomise an inner product argument such that it is zero-knowledge.
- Their results apply to Bulletproofs but not to Bootle et al.

$$Com_A = Commit(a_1, a_2, a_3, a_4)$$

$$Com_B = Commit(b_1, b_2, b_3, b_4)$$

c → Revealed in the clear thus not zero-knowledge.

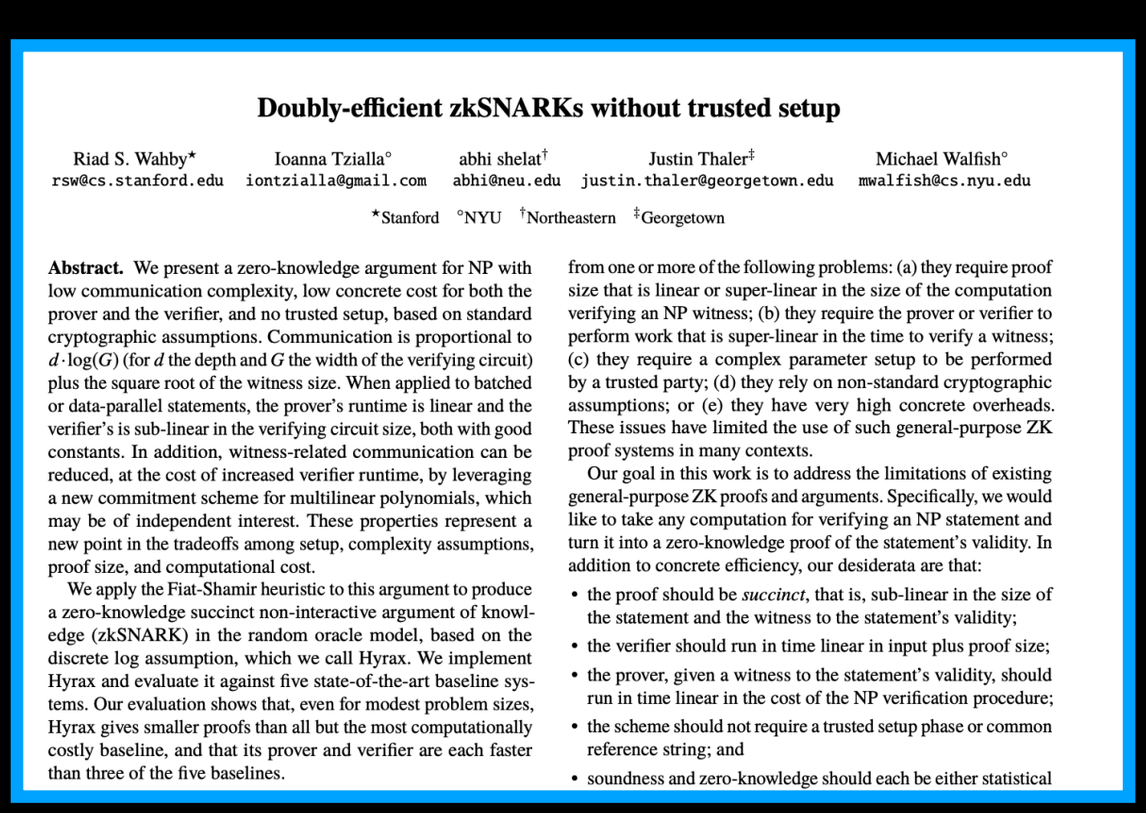
$$Com_{A,B,C} = Commit(a_1, a_2, a_3, a_4; b_1, b_2, b_3, b_4; c)$$ ← Can be randomised

Hyrax

- Wahby et al. achieve better verifier time but with worse proof size.
- Prove multiple inner product arguments hold at the same time.

Spartan

- Setty uses the inner product argument for a ZKP for R1CS.



Doubly-efficient zkSNARKs without trusted setup

Riad S. Wahby* rsw@cs.stanford.edu
Ioanna Tzialla° iontzialla@gmail.com
abhi shelat† abhi@neu.edu
Justin Thaler‡ justin.thaler@georgetown.edu
Michael Walfish° mwalfish@cs.nyu.edu

*Stanford °NYU †Northeastern ‡Georgetown

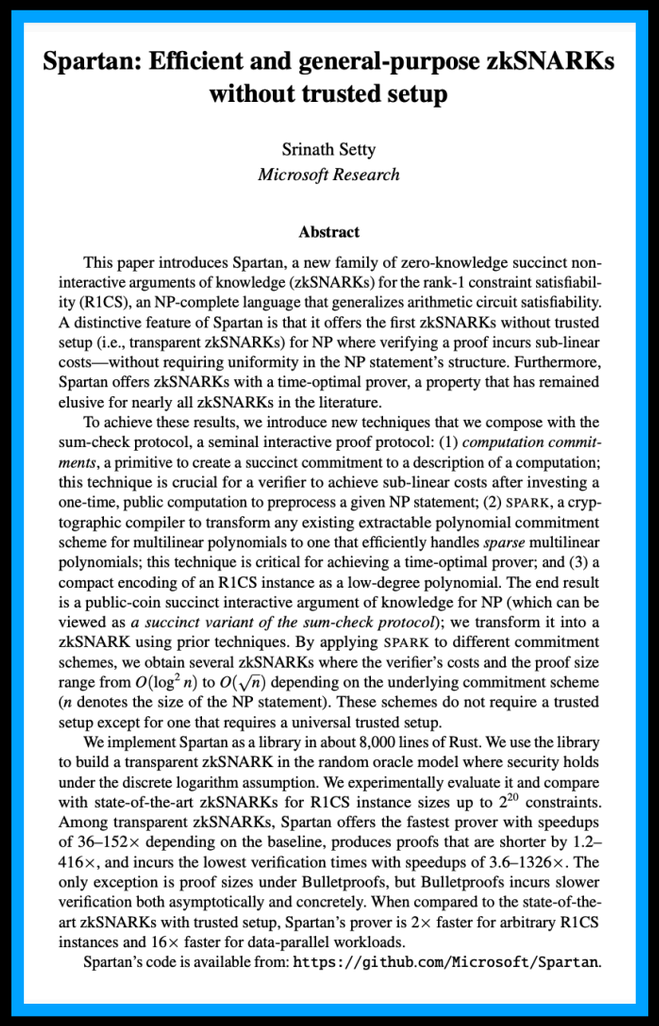
Abstract. We present a zero-knowledge argument for NP with low communication complexity, low concrete cost for both the prover and the verifier, and no trusted setup, based on standard cryptographic assumptions. Communication is proportional to $d \cdot \log(G)$ (for d the depth and G the width of the verifying circuit) plus the square root of the witness size. When applied to batched or data-parallel statements, the prover's runtime is linear and the verifier's is sub-linear in the verifying circuit size, both with good constants. In addition, witness-related communication can be reduced, at the cost of increased verifier runtime, by leveraging a new commitment scheme for multilinear polynomials, which may be of independent interest. These properties represent a new point in the tradeoffs among setup, complexity assumptions, proof size, and computational cost.

We apply the Fiat-Shamir heuristic to this argument to produce a zero-knowledge succinct non-interactive argument of knowledge (zkSNARK) in the random oracle model, based on the discrete log assumption, which we call Hyrax. We implement Hyrax and evaluate it against five state-of-the-art baseline systems. Our evaluation shows that, even for modest problem sizes, Hyrax gives smaller proofs than all but the most computationally costly baseline, and that its prover and verifier are each faster than three of the five baselines.

from one or more of the following problems: (a) they require proof size that is linear or super-linear in the size of the computation verifying an NP witness; (b) they require the prover or verifier to perform work that is super-linear in the time to verify a witness; (c) they require a complex parameter setup to be performed by a trusted party; (d) they rely on non-standard cryptographic assumptions; or (e) they have very high concrete overheads. These issues have limited the use of such general-purpose ZK proof systems in many contexts.

Our goal in this work is to address the limitations of existing general-purpose ZK proofs and arguments. Specifically, we would like to take any computation for verifying an NP statement and turn it into a zero-knowledge proof of the statement's validity. In addition to concrete efficiency, our desiderata are that:

- the proof should be *succinct*, that is, sub-linear in the size of the statement and the witness to the statement's validity;
- the verifier should run in time linear in input plus proof size;
- the prover, given a witness to the statement's validity, should run in time linear in the cost of the NP verification procedure;
- the scheme should not require a trusted setup phase or common reference string; and
- soundness and zero-knowledge should each be either statistical



Spartan: Efficient and general-purpose zkSNARKs without trusted setup

Srinath Setty
Microsoft Research

Abstract

This paper introduces Spartan, a new family of zero-knowledge succinct non-interactive arguments of knowledge (zkSNARKs) for the rank-1 constraint satisfiability (R1CS), an NP-complete language that generalizes arithmetic circuit satisfiability. A distinctive feature of Spartan is that it offers the first zkSNARKs without trusted setup (i.e., transparent zkSNARKs) for NP where verifying a proof incurs sub-linear costs—without requiring uniformity in the NP statement's structure. Furthermore, Spartan offers zkSNARKs with a time-optimal prover, a property that has remained elusive for nearly all zkSNARKs in the literature.

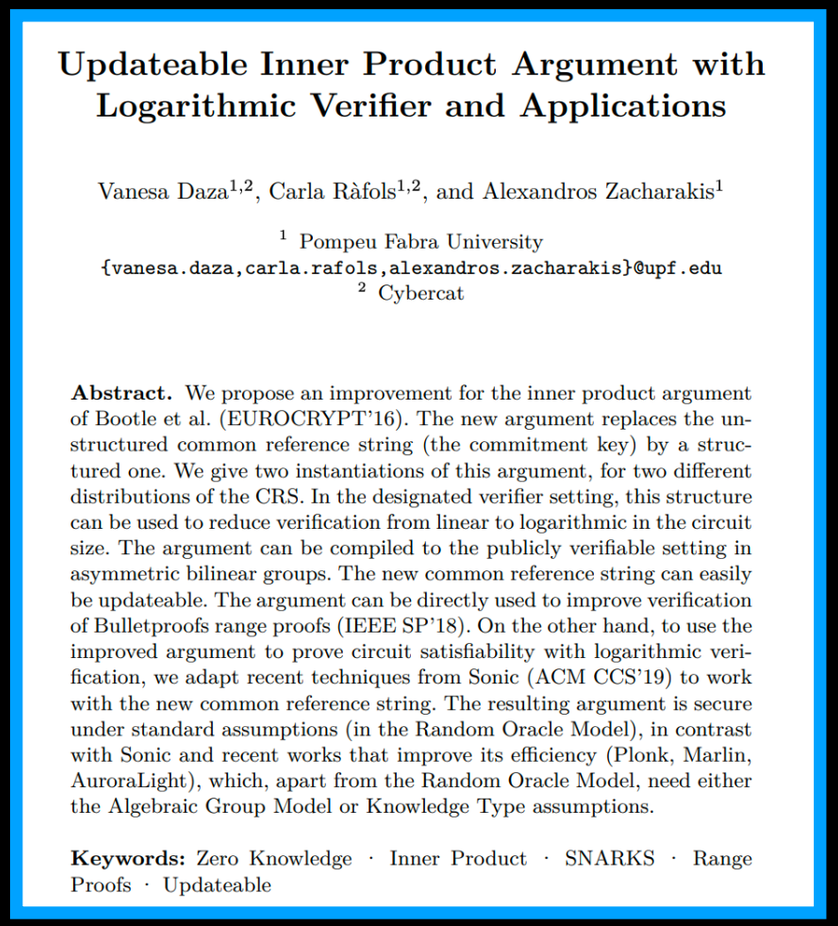
To achieve these results, we introduce new techniques that we compose with the sum-check protocol, a seminal interactive proof protocol: (1) *computation commitments*, a primitive to create a succinct commitment to a description of a computation; this technique is crucial for a verifier to achieve sub-linear costs after investing a one-time, public computation to preprocess a given NP statement; (2) SPARK, a cryptographic compiler to transform any existing extractable polynomial commitment scheme for multilinear polynomials to one that efficiently handles *sparse* multilinear polynomials; this technique is critical for achieving a time-optimal prover; and (3) a compact encoding of an R1CS instance as a low-degree polynomial. The end result is a public-coin succinct interactive argument of knowledge for NP (which can be viewed as *a succinct variant of the sum-check protocol*); we transform it into a zkSNARK using prior techniques. By applying SPARK to different commitment schemes, we obtain several zkSNARKs where the verifier's costs and the proof size range from $O(\log^2 n)$ to $O(\sqrt{n})$ depending on the underlying commitment scheme (n denotes the size of the NP statement). These schemes do not require a trusted setup except for one that requires a universal trusted setup.

We implement Spartan as a library in about 8,000 lines of Rust. We use the library to build a transparent zkSNARK in the random oracle model where security holds under the discrete logarithm assumption. We experimentally evaluate it and compare with state-of-the-art zkSNARKs for R1CS instance sizes up to 2^{20} constraints. Among transparent zkSNARKs, Spartan offers the fastest prover with speedups of 36–152× depending on the baseline, produces proofs that are shorter by 1.2–416×, and incurs the lowest verification times with speedups of 3.6–1326×. The only exception is proof sizes under Bulletproofs, but Bulletproofs incurs slower verification both asymptotically and concretely. When compared to the state-of-the-art zkSNARKs with trusted setup, Spartan's prover is 2× faster for arbitrary R1CS instances and 16× faster for data-parallel workloads.

Spartan's code is available from: `https://github.com/Microsoft/Spartan`.

Inner product arguments for many Pedersen Commitments

Improvements with Universal Setups



Updateable Inner Product Argument with Logarithmic Verifier and Applications

Vanesa Daza[1,2], Carla Ràfols[1,2], and Alexandros Zacharakis[1]

[1] Pompeu Fabra University
{vanesa.daza,carla.rafols,alexandros.zacharakis}@upf.edu
[2] Cybercat

Abstract. We propose an improvement for the inner product argument of Bootle et al. (EUROCRYPT'16). The new argument replaces the unstructured common reference string (the commitment key) by a structured one. We give two instantiations of this argument, for two different distributions of the CRS. In the designated verifier setting, this structure can be used to reduce verification from linear to logarithmic in the circuit size. The argument can be compiled to the publicly verifiable setting in asymmetric bilinear groups. The new common reference string can easily be updateable. The argument can be directly used to improve verification of Bulletproofs range proofs (IEEE SP'18). On the other hand, to use the improved argument to prove circuit satisfiability with logarithmic verification, we adapt recent techniques from Sonic (ACM CCS'19) to work with the new common reference string. The resulting argument is secure under standard assumptions (in the Random Oracle Model), in contrast with Sonic and recent works that improve its efficiency (Plonk, Marlin, AuroraLight), which, apart from the Random Oracle Model, need either the Algebraic Group Model or Knowledge Type assumptions.

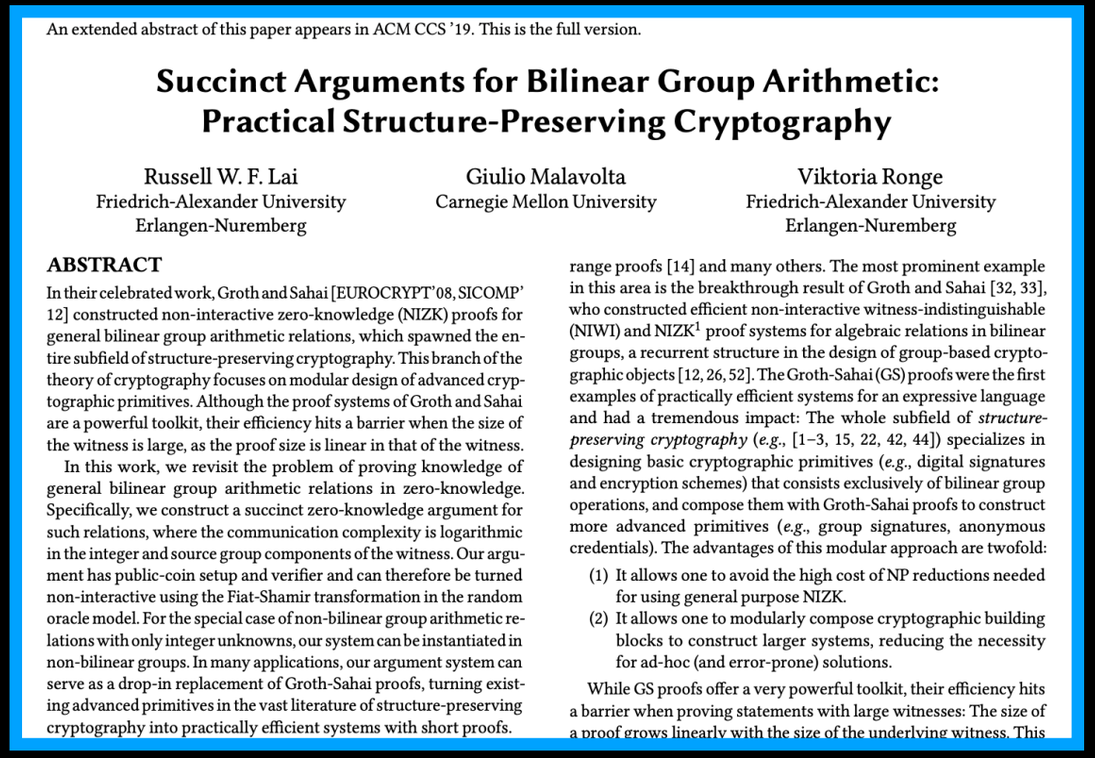
Keywords: Zero Knowledge · Inner Product · SNARKS · Range Proofs · Updateable

Logarithmic verifier in designated verifier/ helper model

- Daza et al. use a universal setup to improve the verification and assumptions in the designated verifier or helper model.

Inner Product Arguments in Bilinear Groups

- All schemes mentioned so far are based over groups in which the discrete logarithm problem holds.
- Lai et al. extend IPAs to bilinear groups (groups with pairings)



An extended abstract of this paper appears in ACM CCS '19. This is the full version.

Succinct Arguments for Bilinear Group Arithmetic: Practical Structure-Preserving Cryptography

Russell W. F. Lai
Friedrich-Alexander University Erlangen-Nuremberg

Giulio Malavolta
Carnegie Mellon University

Viktoria Ronge
Friedrich-Alexander University Erlangen-Nuremberg

ABSTRACT

In their celebrated work, Groth and Sahai [EUROCRYPT'08, SICOMP'12] constructed non-interactive zero-knowledge (NIZK) proofs for general bilinear group arithmetic relations, which spawned the entire subfield of structure-preserving cryptography. This branch of the theory of cryptography focuses on modular design of advanced cryptographic primitives. Although the proof systems of Groth and Sahai are a powerful toolkit, their efficiency hits a barrier when the size of the witness is large, as the proof size is linear in that of the witness.

In this work, we revisit the problem of proving knowledge of general bilinear group arithmetic relations in zero-knowledge. Specifically, we construct a succinct zero-knowledge argument for such relations, where the communication complexity is logarithmic in the integer and source group components of the witness. Our argument has public-coin setup and verifier and can therefore be turned non-interactive using the Fiat-Shamir transformation in the random oracle model. For the special case of non-bilinear group arithmetic relations with only integer unknowns, our system can be instantiated in non-bilinear groups. In many applications, our argument system can serve as a drop-in replacement of Groth-Sahai proofs, turning existing advanced primitives in the vast literature of structure-preserving cryptography into practically efficient systems with short proofs.

range proofs [14] and many others. The most prominent example in this area is the breakthrough result of Groth and Sahai [32, 33], who constructed efficient non-interactive witness-indistinguishable (NIWI) and NIZK[1] proof systems for algebraic relations in bilinear groups, a recurrent structure in the design of group-based cryptographic objects [12, 26, 52]. The Groth-Sahai (GS) proofs were the first examples of practically efficient systems for an expressive language and had a tremendous impact: The whole subfield of *structure-preserving cryptography* (*e.g.*, [1–3, 15, 22, 42, 44]) specializes in designing basic cryptographic primitives (*e.g.*, digital signatures and encryption schemes) that consists exclusively of bilinear group operations, and compose them with Groth-Sahai proofs to construct more advanced primitives (*e.g.*, group signatures, anonymous credentials). The advantages of this modular approach are twofold:

(1) It allows one to avoid the high cost of NP reductions needed for using general purpose NIZK.
(2) It allows one to modularly compose cryptographic building blocks to construct larger systems, reducing the necessity for ad-hoc (and error-prone) solutions.

While GS proofs offer a very powerful toolkit, their efficiency hits a barrier when proving statements with large witnesses: The size of a proof grows linearly with the size of the underlying witness. This

Inner product arguments in pairing groups

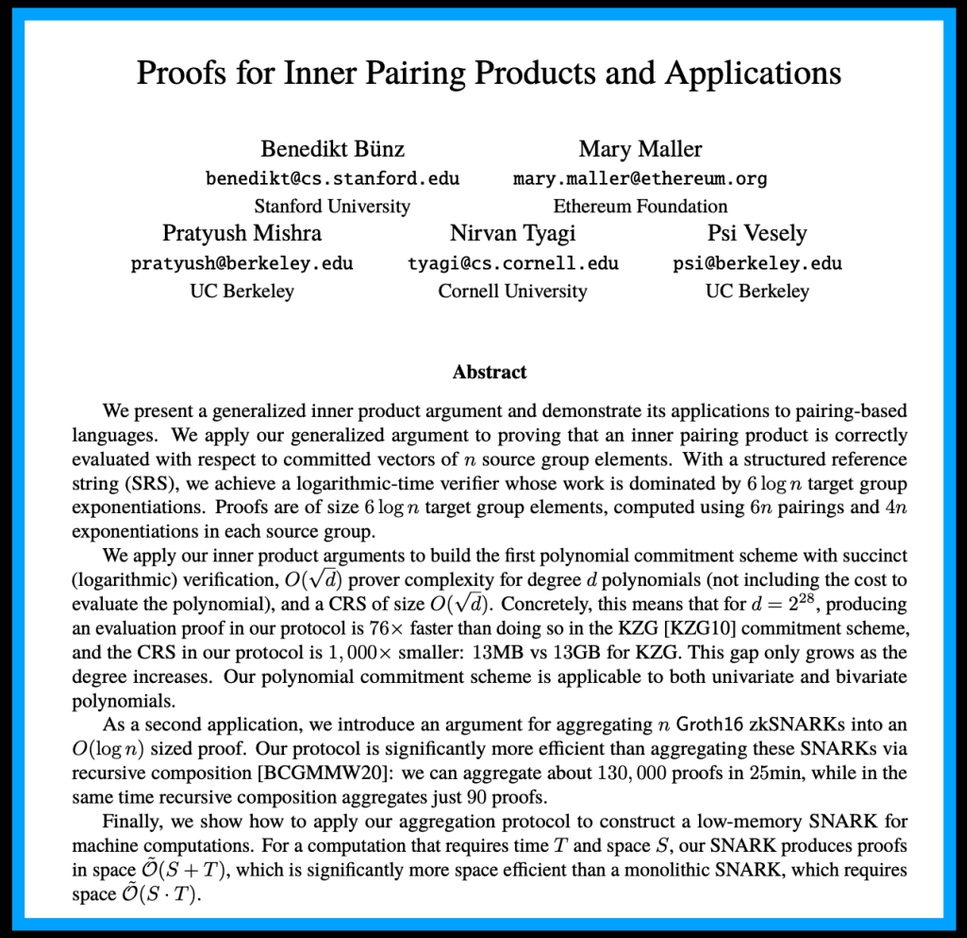
Prove that $C = e(A_1, B_1)e(A_2, B_2)e(A_3, B_3)e(A_4, B_4)$

Inner Product Arguments Applications

- We describe and prove secure generalised inner product arguments. The security proof can be used for many different IPAs.

- We implement the generalised framework in the Arkworks library.

- For applications where a universal setup is acceptable, we have better practical efficiency.



Proofs for Inner Pairing Products and Applications

Benedikt Bünz
benedikt@cs.stanford.edu
Stanford University

Mary Maller
mary.maller@ethereum.org
Ethereum Foundation

Pratyush Mishra
pratyush@berkeley.edu
UC Berkeley

Nirvan Tyagi
tyagi@cs.cornell.edu
Cornell University

Psi Vesely
psi@berkeley.edu
UC Berkeley

Abstract

We present a generalized inner product argument and demonstrate its applications to pairing-based languages. We apply our generalized argument to proving that an inner pairing product is correctly evaluated with respect to committed vectors of n source group elements. With a structured reference string (SRS), we achieve a logarithmic-time verifier whose work is dominated by $6 \log n$ target group exponentiations. Proofs are of size $6 \log n$ target group elements, computed using $6n$ pairings and $4n$ exponentiations in each source group.

We apply our inner product arguments to build the first polynomial commitment scheme with succinct (logarithmic) verification, $O(\sqrt{d})$ prover complexity for degree d polynomials (not including the cost to evaluate the polynomial), and a CRS of size $O(\sqrt{d})$. Concretely, this means that for $d = 2^{28}$, producing an evaluation proof in our protocol is $76\times$ faster than doing so in the KZG [KZG10] commitment scheme, and the CRS in our protocol is $1,000\times$ smaller: 13MB vs 13GB for KZG. This gap only grows as the degree increases. Our polynomial commitment scheme is applicable to both univariate and bivariate polynomials.

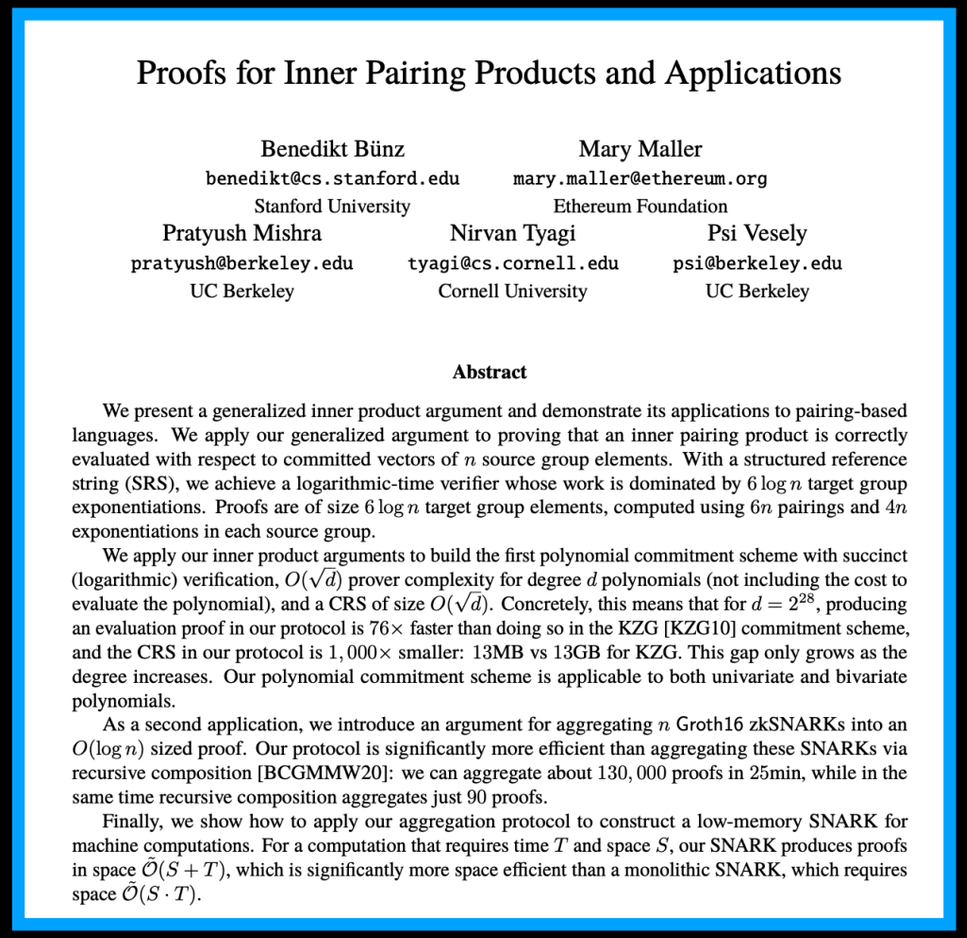
As a second application, we introduce an argument for aggregating n Groth16 zkSNARKs into an $O(\log n)$ sized proof. Our protocol is significantly more efficient than aggregating these SNARKs via recursive composition [BCGMMW20]: we can aggregate about $130,000$ proofs in 25min, while in the same time recursive composition aggregates just 90 proofs.

Finally, we show how to apply our aggregation protocol to construct a low-memory SNARK for machine computations. For a computation that requires time T and space S, our SNARK produces proofs in space $\tilde{\mathcal{O}}(S + T)$, which is significantly more space efficient than a monolithic SNARK, which requires space $\tilde{\mathcal{O}}(S \cdot T)$.

Generalised inner product arguments, polynomial commitments and proof aggregation.

Inner Product Arguments Applications

- With a universal setup, we aggregate Groth16 proofs with good prover efficiency (aggregates 130,000 proofs in 25 minutes).
- Bivariate and univariate polynomial commitment scheme with good prover efficiency, log proof sizes and verification, and small SRS.



Proofs for Inner Pairing Products and Applications

Benedikt Bünz
benedikt@cs.stanford.edu
Stanford University

Mary Maller
mary.maller@ethereum.org
Ethereum Foundation

Pratyush Mishra
pratyush@berkeley.edu
UC Berkeley

Nirvan Tyagi
tyagi@cs.cornell.edu
Cornell University

Psi Vesely
psi@berkeley.edu
UC Berkeley

Abstract

We present a generalized inner product argument and demonstrate its applications to pairing-based languages. We apply our generalized argument to proving that an inner pairing product is correctly evaluated with respect to committed vectors of n source group elements. With a structured reference string (SRS), we achieve a logarithmic-time verifier whose work is dominated by $6 \log n$ target group exponentiations. Proofs are of size $6 \log n$ target group elements, computed using $6n$ pairings and $4n$ exponentiations in each source group.

We apply our inner product arguments to build the first polynomial commitment scheme with succinct (logarithmic) verification, $O(\sqrt{d})$ prover complexity for degree d polynomials (not including the cost to evaluate the polynomial), and a CRS of size $O(\sqrt{d})$. Concretely, this means that for $d = 2^{28}$, producing an evaluation proof in our protocol is $76\times$ faster than doing so in the KZG [KZG10] commitment scheme, and the CRS in our protocol is $1,000\times$ smaller: 13MB vs 13GB for KZG. This gap only grows as the degree increases. Our polynomial commitment scheme is applicable to both univariate and bivariate polynomials.

As a second application, we introduce an argument for aggregating n Groth16 zkSNARKs into an $O(\log n)$ sized proof. Our protocol is significantly more efficient than aggregating these SNARKs via recursive composition [BCGMMW20]: we can aggregate about $130,000$ proofs in 25min, while in the same time recursive composition aggregates just 90 proofs.

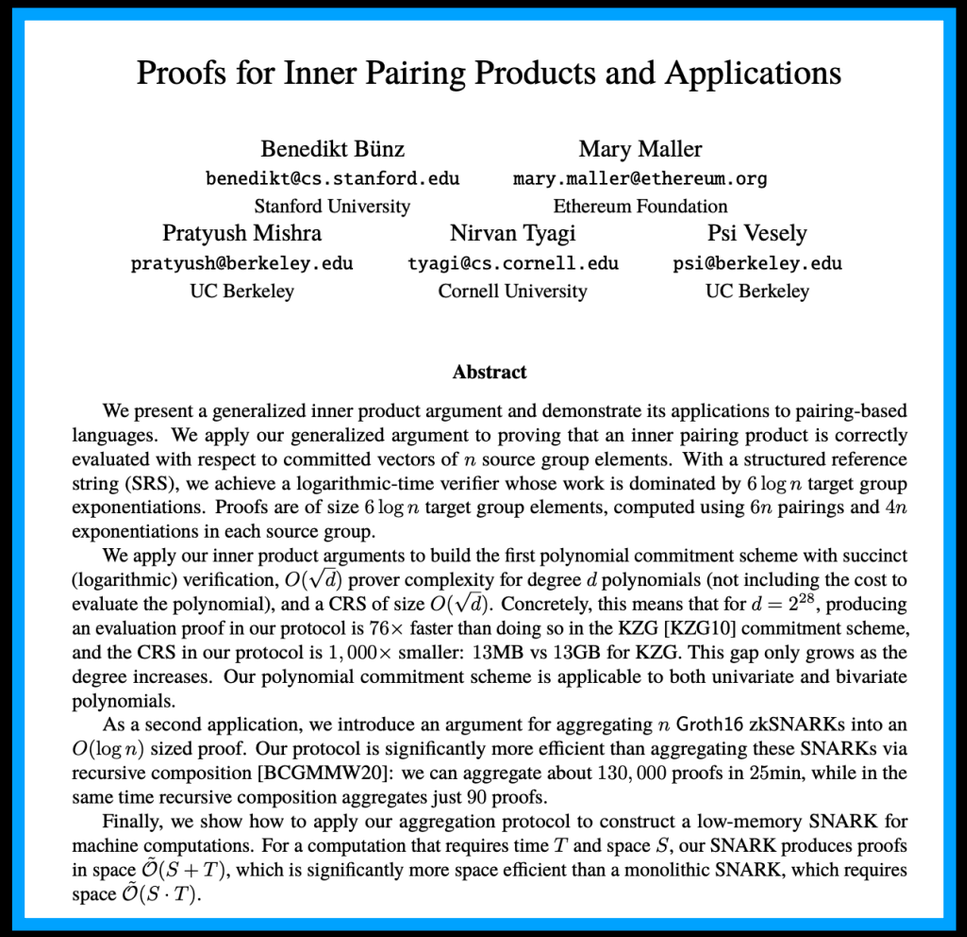
Finally, we show how to apply our aggregation protocol to construct a low-memory SNARK for machine computations. For a computation that requires time T and space S, our SNARK produces proofs in space $\tilde{\mathcal{O}}(S + T)$, which is significantly more space efficient than a monolithic SNARK, which requires space $\tilde{\mathcal{O}}(S \cdot T)$.

Generalised inner product arguments, polynomial commitments and proof aggregation.

Inner Product Arguments Applications

- With a universal setup, we aggregate Groth16 proofs with good prover efficiency (aggregates 130,000 proofs in 25 minutes).
- Bivariate and univariate polynomial commitment scheme with good prover efficiency, log proof sizes and verification, and small SRS.

Applications to machine computations.



Proofs for Inner Pairing Products and Applications

Benedikt Bünz
benedikt@cs.stanford.edu
Stanford University

Mary Maller
mary.maller@ethereum.org
Ethereum Foundation

Pratyush Mishra
pratyush@berkeley.edu
UC Berkeley

Nirvan Tyagi
tyagi@cs.cornell.edu
Cornell University

Psi Vesely
psi@berkeley.edu
UC Berkeley

Abstract

We present a generalized inner product argument and demonstrate its applications to pairing-based languages. We apply our generalized argument to proving that an inner pairing product is correctly evaluated with respect to committed vectors of n source group elements. With a structured reference string (SRS), we achieve a logarithmic-time verifier whose work is dominated by $6 \log n$ target group exponentiations. Proofs are of size $6 \log n$ target group elements, computed using $6n$ pairings and $4n$ exponentiations in each source group.

We apply our inner product arguments to build the first polynomial commitment scheme with succinct (logarithmic) verification, $O(\sqrt{d})$ prover complexity for degree d polynomials (not including the cost to evaluate the polynomial), and a CRS of size $O(\sqrt{d})$. Concretely, this means that for $d = 2^{28}$, producing an evaluation proof in our protocol is $76\times$ faster than doing so in the KZG [KZG10] commitment scheme, and the CRS in our protocol is $1,000\times$ smaller: 13MB vs 13GB for KZG. This gap only grows as the degree increases. Our polynomial commitment scheme is applicable to both univariate and bivariate polynomials.

As a second application, we introduce an argument for aggregating n Groth16 zkSNARKs into an $O(\log n)$ sized proof. Our protocol is significantly more efficient than aggregating these SNARKs via recursive composition [BCGMMW20]: we can aggregate about $130,000$ proofs in 25min, while in the same time recursive composition aggregates just 90 proofs.

Finally, we show how to apply our aggregation protocol to construct a low-memory SNARK for machine computations. For a computation that requires time T and space S, our SNARK produces proofs in space $\tilde{\mathcal{O}}(S + T)$, which is significantly more space efficient than a monolithic SNARK, which requires space $\tilde{\mathcal{O}}(S \cdot T)$.

Generalised inner product arguments, polynomial commitments and proof aggregation.

Recursive Inner Product Arguments

- Lee constructs a logarithmic size, logarithmic verifier IPA for Bilinear groups and public coin setup.

Dory: Efficient, Transparent arguments for Generalised Inner Products and Polynomial Commitments

Jonathan Lee*
*Microsoft Research, Nanotronics Imaging**

Abstract

This paper presents Dory, a transparent setup, public-coin interactive argument for proving correctness of an inner-pairing product between committed vectors of elements of the two source groups. For an inner product of length n, proofs are $6 \log n$ target group elements, 1 element of each source group and 3 scalars. Verifier work is dominated by an $O(\log n)$ multi-exponentiation in the target group. Security is reduced to the symmetric external Diffie Hellman assumption in the standard model. We also show an argument reducing a batch of two such instances to one, requiring $O(n^{1/2})$ work on the Prover and $O(1)$ communication.

We apply Dory to build a multivariate polynomial commitment scheme via the Fiat-Shamir transform. For n the product of one plus the degree in each variable, Prover work to compute a commitment is dominated by a multi-exponentiation in one source group of size n. Prover work to show that a commitment to an evaluation is correct is $O(n^{log(8)/\log 25})$ in general and $O(n^{1/2})$ for univariate or multilinear polynomials, whilst communication complexity and Verifier work are both $O(\log n)$. Using batching, the Verifier can validate ℓ polynomial evaluations for polynomials of size at most n with $O(\ell + \log n)$ group operations and $O(\ell \log n)$ field operations.

Inner product arguments in pairing groups with efficient verifier

- Setty and Lee use Dory to obtain a log size, log verifier zero-knowledge proof.

Quarks: Quadruple-efficient transparent zkSNARKs

Srinath Setty
Microsoft Research

Jonathan Lee*
Microsoft Research

Abstract

We introduce Xiphos and Kopis, new transparent zero-knowledge succinct non-interactive arguments of knowledge (zkSNARKs) for R1CS. They do not require a trusted setup, and their security relies on the standard SXDH problem. They achieve non-interactivity in the random oracle model using the Fiat-Shamir transform. Unlike prior transparent zkSNARKs, which support either a fast prover, short proofs, or quick verification, our work is the first to simultaneously achieve all three properties (both asymptotically and concretely) and in addition an inexpensive setup phase, thereby providing the first *quadruple-efficient* transparent zkSNARKs (*Quarks*).

Under both schemes, for an R1CS instance of size n and security parameter λ, the prover incurs $O_\lambda(n)$ costs to produce a proof of size $O_\lambda(\log n)$. In Xiphos, verification time is $O_\lambda(\log n)$, and in Kopis it is $O_\lambda(\sqrt{n})$. In terms of concrete efficiency, compared to prior state-of-the-art transparent zkSNARKs, Xiphos offers the fastest verification; its proof sizes are competitive with those of SuperSonic [EUROCRYPT 2020], a prior transparent SNARK with the shortest proofs in the literature. Xiphos's prover is fast: its prover time is $\approx$3.8$\times$ of Spartan [CRYPTO 2020], a prior transparent zkSNARK with the fastest prover in the literature, and is 376$\times$ faster than SuperSonic. Kopis, at the cost of increased verification time (which is still concretely faster than SuperSonic), shortens Xiphos's proof sizes further, thereby producing proofs shorter than SuperSonic. Xiphos and Kopis incur 10–10,000$\times$ lower preprocessing costs for the verifier in the setup phase depending on the baseline. Finally, a byproduct of Kopis is Lakonia, a NIZK for R1CS with $O_\lambda(\log n)$-sized proofs, which provides an alternative to Bulletproofs [S&P 2018] with over an order of magnitude faster proving and verification times.

Zero-knowledge argument with efficient verifier

Recursive Inner Product Arguments

Recursive Proof Composition without a Trusted Setup

Sean Bowe[1], Jack Grigg[1], and Daira Hopwood[1]

[1] Electric Coin Company
{sean,jack,daira}@electriccoin.co
https://electriccoin.co/

Abstract. Non-interactive arguments of knowledge are powerful cryptographic tools that can be used to demonstrate the faithful execution of arbitrary computations with publicly verifiable proofs. Increasingly efficient protocols have been described in recent years, with verification time and/or communication complexity that is sublinear in the size of the computation being described. These efficiencies can be exploited to realize *recursive proof composition*: the concept of proofs that attest to the correctness of other instances of themselves, thereby allowing large computational effort to be incrementally verified. All previously known realizations of recursive proof composition have required a trusted setup and cycles of expensive pairing-friendly elliptic curves. We obtain and implement Halo, the first practical example of recursive proof composition without a trusted setup, using the discrete log assumption over normal cycles of elliptic curves. In the process we develop several novel techniques that may be of independent interest.

Keywords: recursive proofs · incrementally verifiable computation · zero knowledge

Recursive inner product arguments in DL groups

Halo

- Bowe et al. use curve cycles in DL groups (no pairing) together with an IPA for a recursive zero-knowledge proof.
- Observes that the verification algorithm from IPAs is highly structured.
- Verifier time depends on the recursion threshold (+logarithmically in the size of the circuit).

Security Proofs without Interaction

Tight State-Restoration Soundness in the Algebraic Group Model

Ashrujit Ghoshal and Stefano Tessaro

Paul G. Allen School of Computer Science & Engineering
University of Washington, Seattle, USA
{ashrujit,tessaro}@cs.washington.edu

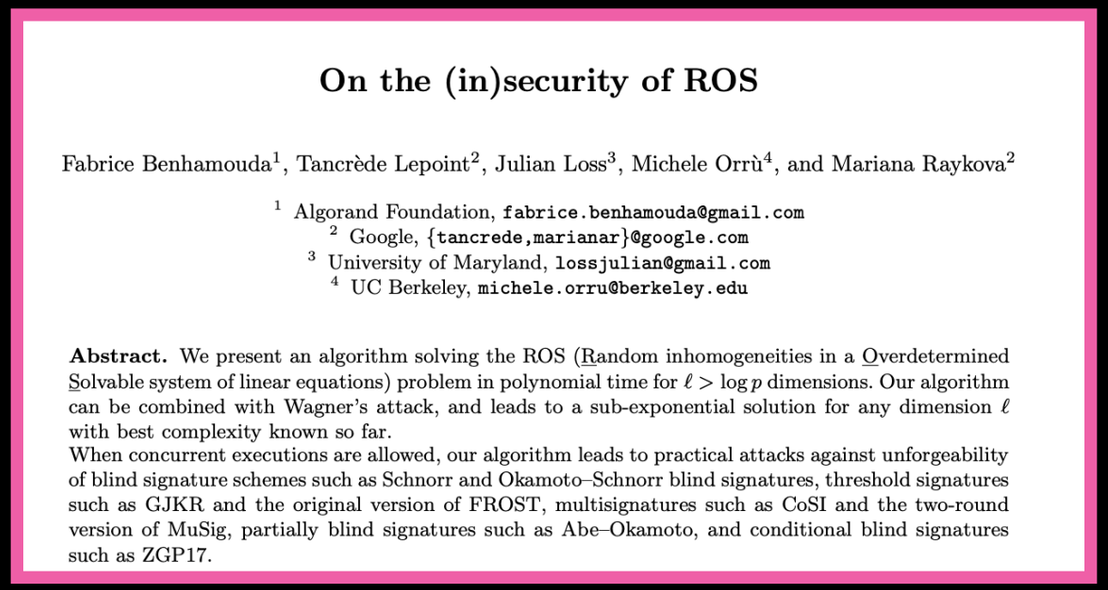
Abstract. Most efficient zero-knowledge arguments lack a concrete security analysis, making parameter choices and efficiency comparisons challenging. This is even more true for non-interactive versions of these systems obtained via the Fiat-Shamir transform, for which the security guarantees generically derived from the interactive protocol are often too weak, even when assuming a random oracle.
This paper initiates the study of *state-restoration soundness* in the algebraic group model (AGM) of Fuchsbauer, Kiltz, and Loss (CRYPTO '18). This is a stronger notion of soundness for an interactive proof or argument which allows the prover to rewind the verifier, and which is tightly connected with the concrete soundness of the non-interactive argument obtained via the Fiat-Shamir transform.
We propose a general methodology to prove tight bounds on state-restoration soundness, and apply it to variants of Bulletproofs (Bootle et al, S&P '18) and Sonic (Maller et al., CCS '19). To the best of our knowledge, our analysis of Bulletproofs gives the *first* non-trivial concrete security analysis for a non-constant round argument combined with the Fiat-Shamir transform.

Keywords. Zero-knowledge proof systems, concrete security, Fiat-Shamir transform, Algebraic Group Model, state-restoration soundness.

Algebraic group model proof for inner product arguments.

- Most inner product arguments are proven in the interactive model, but used in the non-interactive model.
- Standard Fiat-Shamir transformation does not apply to the folding argument.
- Ghoshal and Tessaro provide an algebraic group model proof.

Security Proofs without Interaction



On the (in)security of ROS

Fabrice Benhamouda[1], Tancrède Lepoint[2], Julian Loss[3], Michele Orrù[4], and Mariana Raykova[2]

[1] Algorand Foundation, fabrice.benhamouda@gmail.com
[2] Google, {tancrede,marianar}@google.com
[3] University of Maryland, lossjulian@gmail.com
[4] UC Berkeley, michele.orru@berkeley.edu

Abstract. We present an algorithm solving the ROS (Random inhomogeneities in a Overdetermined Solvable system of linear equations) problem in polynomial time for $\ell > \log p$ dimensions. Our algorithm can be combined with Wagner's attack, and leads to a sub-exponential solution for any dimension ℓ with best complexity known so far.
When concurrent executions are allowed, our algorithm leads to practical attacks against unforgeability of blind signature schemes such as Schnorr and Okamoto–Schnorr blind signatures, threshold signatures such as GJKR and the original version of FROST, multisignatures such as CoSI and the two-round version of MuSig, partially blind signatures such as Abe–Okamoto, and conditional blind signatures such as ZGP17.

- Standard Fiat-Shamir transformation does not apply to the folding argument.

- A linear sized folding argument can be seen as an ROS problem, for which there is a known efficient attack.

- Most IPAs are log sized so the attack doesn't apply (might be false with recursive protocols?).

- Should hash the full transcript with each iteration. Would not spot with an interactive proof.

Maybe not just a theoretical detail?

Thank-you for Listening