Foldable, Recursive Proofs of Isogeny Computation with Reduced Time Complexity

1st Krystal Maughan
College of Engineering and Mathematical Sciences
University of Vermont
Burlington, United States of America
Krystal.Maughan@uvm.edu

2nd Joseph P. Near
College of Engineering and Mathematical Sciences
University of Vermont
Burlington, United States of America
jnear@uvm.edu

3rd Christelle Vincent
College of Engineering and Mathematical Sciences
University of Vermont
Burlington, United States of America
Christelle.Vincent@uvm.edu

***Abstract*—The security of certain post-quantum isogeny-based cryptographic schemes relies on the ability to provably and efficiently compute isogenies between supersingular elliptic curves without leaking information about the isogeny other than its domain and codomain. Earlier work in this direction give mathematical proofs of knowledge for the isogeny, and as a result when computing a chain of n isogenies each proceeding node must verify the correctness of the proof of each preceding node, which is computationally linear in n.**

In this work, we empirically build a system to prove the execution of the circuit computing the isogeny rather than produce a proof of knowledge. This proof can then be used as part of the verifiable folding scheme Nova, which reduces the complexity of an isogeny proof of computation for a chain of n isogenies from $O(n)$ to $O(1)$ by providing at each step a single proof that proves the whole preceding chain. To our knowledge, this is the first application of this type of solution to this problem.

***Index Terms*—post-quantum cryptography, isogenies, zero-knowledge**

I. Introduction

Isogeny-based cryptographic protocols rely on computations over a series of nodes, which are supersingular elliptic curves, connected by arrows which are isogenies between these curves. The security of these protocols is based on variations of the hardness of computing a path between two given nodes, which amounts to an isogeny between two given supersingular elliptic curves, since the composition of isogenies is an isogeny. In its strongest form, this is a problem for which at present there is no polynomial-time solution either on classical or quantum computers, which makes it ideal for post-quantum cryptographic applications.

The first such protocol to be proposed was the Charles-Goren-Lauter hash function [CGL09]. The idea is to use the string to be hashed to create a path navigating the nodes, and releasing the (j-invariant of the) endpoint as the hash of the string. Unfortunately, it was found [KLPT14], [EHL$^+$18] that this hash function could only be made secure if instantiated with a supersingular elliptic curve with unknown endomorphism ring. Since then, more protocols requiring such a curve as a starting point have been proposed [DFMPS19], [ADFMP20], [BDF21], [LGDdSG21], [Ste22], [Bas24].

The current state of knowledge about supersingular elliptic curves allows one to, given any prime p, give a supersingular elliptic curve defined over $\overline{\mathbb{F}}_p$, either using the curve given in [Brö09] or the CM (short for complex multiplication) method. In both cases, the resulting supersingular elliptic curve either has known or efficiently computable endomorphism ring. Starting with such a supersingular elliptic curve, say E_0, one can then take a random walk in the **supersingular isogeny graph** to reach a new, random supersingular elliptic curve, say E_1. However, knowledge of the endomorphism ring of E_0 as well as of the path joining E_0 to E_1, which is the data of an isogeny from E_0 to E_1, allows the efficient computation of the endomorphism ring of E_1. At this time there is no known algorithm to generate a random supersingular elliptic curve with unknown endomorphism ring [BBD$^+$24], [MMP22].

One solution to this problem is to employ a sequential multiparty computation [BDF21] starting at some known supersingular curve E_0, and where the ith party to the computation generates a random walk from the elliptic curve E_{i-1} to another E_i, and reveals only the endpoint E_i to the next party to the computation. If every party behaves honestly and the parties do not collude, there is no known isogeny from E_0 to the final curve E_n, whose endomorphism ring is then unknown.

Because of the possibility for the parties to the computation to collude, this technique does not fully solve the problem of generating a random supersingular elliptic curve with unknown endomorphism ring. However, if each party can prove knowledge of an isogeny from E_{i-1} to E_i without revealing that isogeny, the final curve E_n can at least be trusted by the parties to the computation, who would know that they did not themselves collude. For this reason, efficiently generating proofs of knowledge for supersingular isogenies is one key to solving the problem of generating supersingular elliptic

curves with unknown endomorphism ring, and techniques allowing the efficient verification of sequential computations of isogenies are especially interesting for this application.

In this article we propose to use a **folding scheme** to combine the proofs of computation provided by each party into a single proof that can be verified in (asymptotically) the same time as it takes to verify each of the proofs individually. This is done via the Nova [KST22] recursive Succinct Non-Interactive Argument of Knowledge (SNARK) setup. Nova is a recursive zero-knowledge SNARK that uses incrementally verifiable computation (IVC) via a folding scheme. For compatibility with existing toolchains and future folding schemes, our implementation encodes the isogeny computation as an arithmetic circuit; see below for details. We also empirically build and perform an evaluation of our recursive proof. As far as we are aware, this is a novel application of folding schemes to isogeny-based protocols.

II. Previous Work on Proofs of Isogeny Knowledge

The first proof of isogeny knowledge for supersingular elliptic curves was based on the Supersingular Isogeny Diffie-Hellman (SIDH) protocol [FJP14], which was subsequently found to be insecure and then fixed [GKPV21], [DFDGZ23]. While the recent polynomial-time attacks on the SIDH problem [CD23], [Rob23], [MMP+23] have made some versions of the resulting protocols insecure, in some cases they can still be made secure using ternary challenges [BKW20], [DFDGZ23]. A different proof of isogeny knowledge, relying on pairing assumptions, was given in [BDF21], but unfortunately the security is only proved heuristically.

At present the state of the art in this area is [BCC+23], in which the authors give a method to compute a statistically zero-knowledge proof of isogeny knowledge that is compatible with any base field. To prove knowledge of an isogeny $\phi\colon E_0 \to E_1$, the method relies on constructing a random isogeny $\psi\colon E_0 \to E_2$, and then completing the SIDH square to obtain $\phi'\colon E_2 \to E_3$ and $\psi'\colon E_1 \to E_3$. The authors show that when ψ is of large-enough degree, ϕ' does not leak information about ϕ. As a result, the prover can commit to the tuples (ψ, E_2), (ψ', E_3), and (ϕ', E_2, E_3) (roughly speaking) and reveal any of the three upon a challenge to prove knowlegdge of ϕ. The protocol can be made non-interactive using the Fiat-Shamir heuristic [FS87].

By the very nature of the proof of knowledge provided, to trust in the security of the resulting final supersingular elliptic curve, every party to the computation must verify every other party's computation. This cannot be avoided, and therefore the cost of the computation of the final, secure supersingular elliptic curve increases linearly with the number of parties who wish to participate in constructing this curve.

III. Isogeny Walk Computation and Proof

As previously stated, our proposed proof of isogeny knowledge is in fact a proof of isogeny **computation**. For this reason, contrary to previous work presented above, the prover only needs to compute the isogeny walk that needs to be verified. We begin by describing this walk.

A. Preliminaries

Throughout, let $p \neq \ell$ be primes. The **supersingular ℓ-isogeny graph over $\mathbb{F}_{p^2}$** is the graph whose nodes correspond to the set of isomorphism classes of supersingular elliptic curves over $\overline{\mathbb{F}}_p$ and whose arrows correspond to ℓ-isogenies defined over $\mathbb{F}_{p^2}$. For every pair $p \neq \ell$, this graph is an **expander graph** and in fact a **Ramanujan graph** [Piz90].

B. Parameters for the walks

Though *a priori* the techniques presented in this article should be applicable to any pair of distinct primes $p \neq \ell$, in practice our implementation requires $p \equiv 3 \pmod 4$ and $\ell = 2$. We defer discussion of further restrictions on p to the Current Challenges and Limitations of Our Implementation section below.

Our choice of parameters is driven by the necessity to subsequently compile the isogeny walk into an arithmetic circuit: we chose the parameters that allowed for the simplest computation. Indeed, because $p \equiv 3 \pmod 4$, we can use simple algorithms to compute square roots both in $\mathbb{F}_p$ and $\mathbb{F}_{p^2} \cong \mathbb{F}_p[x]/(x^2+1)$.

In addition, we choose $\ell = 2$ for the simplicity of executing the walk in this case. For every walk, we use the curve of j-invariant 287496 with equation

$$E_0 : y^2 = x^3 - 3^{-2} \cdot 7^{-2} \cdot 11^3 \cdot x - 2 \cdot 3^{-3} \cdot 7^{-2} \cdot 11^3 \quad (1)$$

as our starting point. This curve has full rational 2-torsion (over $\mathbb{F}_{p^2}$), and therefore every curve in its $\mathbb{F}_{p^2}$-isogeny class does as well. This means that a step in this 2-isogeny graph starting at a curve $E : y^2 = x^3 + Ax + B$ corresponds to a root of $x^3 + Ax + B$, and each of these roots belong to $\mathbb{F}_{p^2}$. We note that for the purposes of generating a random supersingular elliptic curve with unknown endomorphism ring, any prime ℓ will do, so this restriction does not reduce the functionality of the code.

C. One step of the walk

Each party to the protocol must compute a random walk of length κ for some pre-determined value κ that ensures that the endpoint of this walk in the supersingular ℓ-isogeny graph will be sufficiently close to equidistributed in the graph. This walk is done inductively by taking κ non-backtracking steps in the graph, in the following manner: To begin, the party has a curve $E : y^2 = x^3 + Ax + B$ and the x-coordinate of a 2-torsion point of this curve x_{back}. As noted above, $x_{back} \in \mathbb{F}_{p^2}$ and is a root of $x^3 + Ax + B$. The point $P_{back} = (x_{back}, 0)$ is assumed to generate the kernel of the dual of the isogeny $\phi\colon E_{back} \to E$ from the previous curve in the walk to this curve.

To take a step forward determined by the random bit $b \in \{0, 1\}$, we then compute a root x_{root} of the quadratic polynomial $(x^3 + Ax + B)/(x - x_{back})$, corresponding to the 2-torsion point $P_{root} = (x_{root}, 0)$, and output x_b, the

x-coordinate of the point $P_b = P_{root} + [b]P_{back}$ (which is equal to x_{root} if $b = 0$ or to $-x_{root} - x_{back} \pmod p$ if $b = 1$). Using Vélu's formula [Vél71], we can compute the codomain E_b of the isogeny ϕ_b from E with kernel generated by P_b, and the x-coordinate x_{dual} of the point $\phi_b(P_{back})$, which generates the kernel of the dual isogeny $\hat{\phi}_b : E_b \to E$. The data (E_b, x_{dual}) can then be used as the input to the next step of the walk. In this manner, a random bit string corresponds to a deterministic (as the computation of x_{root} is determined by the input $(x^3 + Ax + B)/(x - x_{back})$) random walk in the supersingular 2-isogeny graph.

To begin the walk, we provide the x-coordinate

$$x_0 = -2 \cdot 3^{-1} \cdot 7^{-1} \cdot 11 \tag{2}$$

on the curve E_0 given by equation (1). The corresponding point P_0 in this case is the kernel of the isogeny from E_0 to the curve of j-invariant 1728.

D. Random curve generation

We now turn our attention to describing the overall protocol to generate a random supersingular elliptic curve with unknown endomorphism ring over $\mathbb{F}_{p^2}$ for $p \equiv 3 \pmod 4$. The input to the protocol are the length κ of the random walk to be taken by each party to the computation as well as the prime p. With starting point E_0 from equation (1) in the supersingular isogeny graph and backtracking direction x_0 from equation (2), each party performs the following algorithm:

Algorithm 1 Proof of Isogeny Computation

Require: $p \equiv 3 \pmod 4, \kappa, E_0, x_0$

Starting proof: $\mathcal{P}_0 =$ None
for $i \in 1...n$ **do**:
  Party i receives proof $\mathcal{P}_{i-1}$, elliptic curve E_{i-1}, backtracking direction x_{i-1}
  Party i generates a secret random string $s_i \in \{0,1\}^\kappa$
  Party i performs the walk determined by the string s_i starting at E_{i-1} with backtracking direction x_{i-1}; records endpoint E_i and final backtracking direction x_i
  Party i generates a witness w_i = `gen-witness`(walk-circuit, s_i)
  Party i generates proof $\mathcal{P}_i$ = nova-proof(walk-circuit, w_i, $\mathcal{P}_{i-1}$)
  Party i sends $\mathcal{P}_i$ to party $i + 1$
end for
return: proof of walk $\mathcal{P}_n$

Here for clarity we note that the proof $\mathcal{P}_i$ proves that party i has a witness proving the computation of an isogeny from E_{i-1} and E_i, and that party i has verified that the proof $\mathcal{P}_{i-1}$ is valid. In our proof setup, the witness for each step of the recursive proof is the party's string of random bits s_i; this string fully determines the party's walk and final isogeny E_i. All other components of the proof are public, including the starting and ending curves E_{i-1} and E_i and the starting and ending backtracking directions x_{i-1} and x_i.

IV. Implementation and Evaluation

Our implementation translates the isogeny walk computation into a circuit, then uses the Nova system to produce and verify proofs. We first wrote the code to compute one party's random walk in about 200 lines of Python. We then used PicoZK [BNM], a high-level Python library, to produce an arithmetic circuit in Circom format [BMIMT+22], a lower-level circuit representation for zero-knowledge proof statements. The Circom compiler then translated the circuit into a **rank-one constraint system** (R1CS) format, a standard intermediate representation for zero-knowledge proof statements. Finally, we leveraged the Nova Scotia [NoS] middleware to connect the output of the Circom compiler to the Nova proof system.

We evaluated our approach for several different lengths of the isogeny walk, and the results appear in Table I. The size of the circuit grows linearly with the length of the isogeny walk being computed in each folding step; the proof and verification times grow with the circuit size. Even without any optimization, our prototype implementation produces a proof for a walk of length $\kappa = 200$ in fewer than two minutes on a standard laptop, and verification of the entire walk takes only a few seconds.

Walk Length	Constraints	Prove Time (s)	Verify Time (ms)
50	16,401	76 s	3,101 ms
100	20,601	97 s	3,441 ms
200	29,201	110 s	3,950 ms

TABLE I
Results of our empirical evaluation for several different walk lengths and a 256-bit prime p. We ran our experiments on a laptop with an Intel Core i5-1240P and 16GB of RAM running Linux.

V. Current Challenges and Limitations of Our Implementation

As discussed in the Parameters for the walks section, our Python implementation relies essentially on the condition $p \equiv 3 \pmod 4$, not only in the computation of the square roots, but also in its assumption that the elliptic curves with j-invariant 1728 and the curve E_0 are supersingular. While it would be interesting to develop a more general implementation, this case does cover every SIDH prime, and therefore the vast majority of primes p used in practice.

A more serious restriction comes from our use of the Nova proof system, which requires a commitment scheme based on the hardness of the discrete log problem on cycles of elliptic curves. A consequence of this is that the characteristic p of the base field must be the scalar field of some elliptic curve belonging to a cycle of curves which is currently supported both by the Nova and the Circom implementations. We were able to find exactly one supported elliptic curve belonging to a cycle of curves and whose scalar field has cardinality

$p \equiv 3 \pmod 4$: the secq256k1 curve [Ark]. This means that currently our implementation is only available for

$$p = 115792089237316195423570985\ldots$$
$$008687907853269984665640564039\ldots$$
$$39457584007908834671663.$$

We stress that this is entirely an implementation issue, and that added support for more cycles of elliptic curves, some of whose scalar fields have cardinality $p \equiv 3 \pmod 4$ would allow the generation of a random supersingular curve over $\overline{\mathbb{F}}_p$ for any such p.

Secondly, our approach is not specific to a particular folding scheme; it only requires compatibility with generic arithmetic circuits. Our current implementation uses the Nova recursive zero-knowledge SNARK, which relies on the difficulty of the discrete log problem and is not quantum-safe. Presently, no suitable quantum-safe folding scheme exists of which we are aware, but our approach can be easily extended to any future quantum-safe folding scheme. Such an extension would also remove the restriction on the prime p arising from the limited availability of curves with scalar field of size $3 \pmod 4$.

VI. Summary and Future Contributions

We have shown a proof of execution of the circuit for isogeny-based proofs of knowledge using folding schemes. This is a novel, recursive solution that reduces existing computation for a chain of n isogenies from $O(n)$ to $O(1)$. Our future work involves computation using recursive zero-knowledge SNARKs that are provably safe from potential quantum attacks such as quantum rewinding [Wat09], [LMS22]. Our future work will involve an implementation that is both scalable and has these provable quantum-safe guarantees.

Acknowledgements

This material is based upon work supported by DARPA under Contract No. HR001120C0087. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of DARPA. The authors would like to thank Pratyush Mishra, PhD, for his conversations with us on incrementally verifiable computation. The authors would like to acknowledge that this was work done in part while one of the authors was visiting the Simons Institute for the Theory of Computing during the "Quantum Algorithms, Complexity and Fault Tolerance" Spring 2024 semester-long program.

References

[ADFMP20] Navid Alamati, Luca De Feo, Hart Montgomery, and Sikhar Patranabis. Cryptographic group actions and applications. In *Advances in Cryptology – ASIACRYPT 2020: 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7–11, 2020, Proceedings, Part II*, page 411–439, Berlin, Heidelberg, 2020. Springer-Verlag.

[Ark] Arkworks. secq256k1. https://github.com/arkworks-rs/curves/blob/master/secq256k1/src/curves/mod.rs.

[Bas24] Andrea Basso. A post-quantum round-optimal oblivious prf from isogenies. In Claude Carlet, Kalikinkar Mandal, and Vincent Rijmen, editors, *Selected Areas in Cryptography – SAC 2023*, pages 147–168, Cham, 2024. Springer Nature Switzerland.

[BBD+24] Jeremy Booher, Ross Bowden, Javad Doliskani, Tako Boris Fouotsa, Steven D. Galbraith, Sabrina Kunzweiler, Simon-Philipp Merz, Christophe Petit, Benjamin Smith, Katherine E Stange, Yan Bo Ti, Christelle Vincent, José Felipe Voloch, Charlotte Weitkämper, and Lukas Zobernig. Failing to Hash Into Supersingular Isogeny Graphs. *The Computer Journal*, page bxae038, 05 2024.

[BCC+23] Andrea Basso, Giulio Codogni, Deirdre Connolly, Luca De Feo, Tako Boris Fouotsa, Guido Maria Lido, Travis Morrison, Lorenz Panny, Sikhar Patranabis, and Benjamin Wesolowski. Supersingular curves you can trust. In *Advances in cryptology—EUROCRYPT 2023. Part II*, volume 14005 of *Lecture Notes in Comput. Sci.*, pages 405–437. Springer, Cham, 2023.

[BDF21] Jeffrey Burdges and Luca De Feo. Delay encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 302–326. Springer, 2021.

[BKW20] Dan Boneh, Dmitry Kogan, and Katharine Woo. Oblivious pseudorandom functions from isogenies. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2020*, pages 520–550, Cham, 2020. Springer International Publishing.

[BMIMT+22] Marta Bellés-Muñoz, Miguel Isabel, Jose Luis Muñoz-Tapia, Albert Rubio, and Jordi Baylina. CIRCOM: A circuit description language for building zero-knowledge applications. *IEEE Transactions on Dependable and Secure Computing*, 2022.

[BNM] Mark Blunk, Joe Near, and Kimberlee Model. PicoZK repository. https://github.com/uvm-plaid/picozk.

[Brö09] Reinier Bröker. Constructing supersingular elliptic curves. *Journal of Combinatorics and Number Theory*, 1(3):269–273, 2009.

[CD23] Wouter Castryck and Thomas Decru. An efficient key recovery attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023*, pages 423–447, Cham, 2023. Springer Nature Switzerland.

[CGL09] Denis X. Charles, Eyal Z. Goren, and Kristin E. Lauter. Cryptographic hash functions from expander graphs. *Journal of CRYPTOLOGY*, 22(1):93–113, 2009.

[DFDGZ23] Luca De Feo, Samuel Dobson, Steven D. Galbraith, and Lukas Zobernig. SIDH proof of knowledge. In *Advances in Cryptology – ASIACRYPT 2022: 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5–9, 2022, Proceedings, Part II*, page 310–339, Berlin, Heidelberg, 2023. Springer-Verlag.

[DFMPS19] Luca De Feo, Simon Masson, Christophe Petit, and Antonio Sanso. Verifiable delay functions from supersingular isogenies and pairings. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 248–277. Springer, 2019.

[EHL+18] Kirsten Eisenträger, Sean Hallgren, Kristin Lauter, Travis Morrison, and Christophe Petit. Supersingular isogeny graphs and endomorphism rings: reductions and solutions. In *Advances in cryptology—EUROCRYPT 2018. Part III*, volume 10822 of *Lecture Notes in Comput. Sci.*, pages 329–368. Springer, Cham, 2018.

[FJP14] Luca De Feo, David Jao, and Jérôme Plût. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *Journal of Mathematical Cryptology*, 8(3):209–247, 2014.

[FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology — CRYPTO' 86*, pages 186–194, Berlin, Heidelberg, 1987. Springer Berlin Heidelberg.

[GKPV21] Wissam Ghantous, Shuichi Katsumata, Federico Pintore, and Mattia Veroni. Collisions in supersingular isogeny graphs and the SIDH-based identification protocol. Cryptology ePrint

Archive, Paper 2021/1051, 2021. https://eprint.iacr.org/2021/1051.

[KLPT14] David Kohel, Kristin Lauter, Christophe Petit, and Jean-Pierre Tignol. On the quaternion ℓ-isogeny path problem. *LMS J. Comput. Math.*, 17:418–432, 2014.

[KST22] Abhiram Kothapalli, Srinath Setty, and Ioanna Tzialla. Nova: Recursive zero-knowledge arguments from folding schemes. In *Advances in Cryptology – CRYPTO 2022*, pages 359–388, Cham, 2022. Springer Nature Switzerland.

[LGDdSG21] Yi-Fu Lai, Steven D. Galbraith, and Cyprien Delpech de Saint Guilhem. Compact, efficient and uc-secure isogeny-based oblivious transfer. In *Advances in Cryptology – EUROCRYPT 2021: 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17–21, 2021, Proceedings, Part I*, page 213–241, Berlin, Heidelberg, 2021. Springer-Verlag.

[LMS22] A. Lombardi, F. Ma, and N. Spooner. Post-quantum zero knowledge, revisited or: How to do quantum rewinding undetectably. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 851–859. IEEE Computer Society, 2022.

[MMP22] Marzio Mula, Nadir Murru, and Federico Pintore. On random sampling of supersingular elliptic curves. Cryptology ePrint Archive, Paper 2022/528, 2022. https://eprint.iacr.org/2022/528.

[MMP+23] Luciano Maino, Chloe Martindale, Lorenz Panny, Giacomo Pope, and Benjamin Wesolowski. A direct key recovery attack on sidh. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023*, pages 448–471, Cham, 2023. Springer Nature Switzerland.

[NoS] Nova Scotia repository. https://github.com/nalinbhardwaj/Nova-Scotia.

[Piz90] Arnold K. Pizer. Ramanujan graphs and Hecke operators. *Bull. Amer. Math. Soc. (N.S.)*, 23(1):127–137, 1990.

[Rob23] Damien Robert. Breaking SIDH in polynomial time. In *Advances in Cryptology – EUROCRYPT 2023: 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part V*, page 472–503, Berlin, Heidelberg, 2023. Springer-Verlag.

[Ste22] Bruno Sterner. Commitment schemes from supersingular elliptic curve isogeny graphs. *Mathematical Cryptology*, 1(2):40–51, Mar. 2022.

[Vél71] Jacques Vélu. Isogénies entre courbes elliptiques. *Comptes Rendus Hebdomadaires des Séances de l'Académie des Sciences. Séries A et B*, 273:A238–A241, 1971.

[Wat09] John Watrous. Zero-knowledge against quantum attacks. *SIAM Journal on Computing*, 39(1):25–58, 2009.