Quantum Automating TC^0-Frege Is LWE-Hard

Noel Arteche
Lund University, Sweden
University of Copenhagen, Denmark

Gaia Carenini
École Normale Supérieure (ENS-PSL), Paris, France
University of Cambridge, UK

Matthew Gray
University of Oxford, UK

Abstract

We prove the first hardness results against efficient proof search by quantum algorithms. We show that under Learning with Errors (LWE), the standard lattice-based cryptographic assumption, no quantum algorithm can weakly automate $\mathbf{TC}^0$-Frege. This extends the line of results of Krajíček and Pudlák (*Information and Computation*, 1998), Bonet, Pitassi, and Raz (*FOCS*, 1997), and Bonet, Domingo, Gavaldà, Maciel, and Pitassi (*Computational Complexity*, 2004), who showed that Extended Frege, $\mathbf{TC}^0$-Frege and $\mathbf{AC}^0$-Frege, respectively, cannot be weakly automated by classical algorithms if either the RSA cryptosystem or the Diffie-Hellman key exchange protocol are secure. To the best of our knowledge, this is the first interaction between quantum computation and propositional proof search.

2012 ACM Subject Classification Theory of computation → Proof complexity; Theory of computation → Quantum complexity theory

Keywords and phrases automatability, post-quantum cryptography, feasible interpolation

Digital Object Identifier 10.4230/LIPIcs.CCC.2024.15

Related Version *Full Version*: `https://eccc.weizmann.ac.il/report/2024/029/` [7]

Funding *Noel Arteche*: This work was supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

Acknowledgements The question of the quantum non-automatability of strong proof systems was suggested to us by three different people. We thank Vijay Ganesh for bringing it up during the Dagstuhl Seminar 22411 *Theory and Practice of SAT and Combinatorial Solving*. We thank Susanna F. de Rezende for bringing our attention to the problem later and for insightful comments and careful proofreading. We would also like to thank Ján Pich for pointing us to the problem and discussing many details with us. We are particularly grateful for him directing us to the work of Soltys and Cook on LA. We also thank Rahul Santhanam for his insights and conversations and Yanyi Liu for pointing us to the existence of the certificates of injectivity. We also thank María Luisa Bonet, Jonas Conneryd, Ronald de Wolf, Eli Goldin, Peter Hall, Russell Impagliazzo, Erfan Khaniki, Alex Lombardi, Daniele Micciancio, Angelos Pelecanos and Michael Soltys for useful comments, suggestions and pointers. A preliminary version of this work was presented at the poster session of QIP 2024. We are thankful to the anonymous reviewers and their suggestions. We are also grateful to the anonymous CCC reviewers for their comments and particularly for observing that some crucial axioms were missing from the definition of $\mathsf{LA}_{\mathbb{Q}}$ in an earlier version of this work.
This work was done in part while the authors were visiting the Simons Institute for the Theory of Computing at UC Berkeley during the spring of 2023 for the *Meta-Complexity* and *Extended Reunion: Satisfiability* programs.

1 Introduction

Originally, propositional proof complexity has been primarily concerned with proving lower bounds for the length of proofs in propositional proof systems, with the ultimate goal of settling whether $\mathbf{NP} = \mathbf{coNP}$ [21]. In parallel, a growing line of research has focused on the computational hardness of finding propositional proofs. Efficient proof search is formally captured by the notion of *automatability*, introduced by Bonet, Pitassi, and Raz [17]: a propositional proof system S is automatable if there exists an algorithm that given as input a tautology φ, outputs an S-proof of φ in time polynomial in the size of the shortest proof. By relating proofs and computation, automatability connects proof complexity to central areas of theoretical computer science such as automated theorem proving, SAT solving and combinatorial optimization [18], learning theory [44], and Kolmogorov complexity [36].

Except for very weak proof systems like Tree-like Resolution, automatable in quasipolynomial time [12], most natural systems appear to be non-automatable under standard hardness assumptions. Existing hardness results can be split into two broad categories. Work from the late 90s and early 00s showed that stronger proof systems are non-automatable under cryptographic assumptions, while more recent work has shown that weaker proof systems are non-automatable under the optimal assumption that $\mathbf{P} \neq \mathbf{NP}$.

The cryptography-based approach was initiated by the seminal work of Krajíček and Pudlák [37], who showed that Extended Frege is not automatable unless factoring can be solved efficiently, although the notion of automatability would only be defined slightly later by Bonet, Pitassi, and Raz [17], who showed that $\mathbf{TC}^0$-Frege is hard to automate unless Blum integers can be factored by polynomial-size circuits. Finally, Bonet, Domingo, Gavaldà, Maciel, and Pitassi [16] extended the existing result from $\mathbf{TC}^0$-Frege to $\mathbf{AC}^0$-Frege under the stronger assumption that Blum integers cannot be factored by subexponential-size circuits.

Building on a long line of work [19, 30, 46, 8, 4, 5, 38], the first **NP**-hardness result was shown in 2019, when Atserias and Müller [10] proved that Resolution is not automatable unless $\mathbf{P} = \mathbf{NP}$. This is optimal, as $\mathbf{P} = \mathbf{NP}$ implies the automatability of any proof system. Their proof uses a clever reduction from SAT that requires showing a specific lower bound for this system. The technique has since been adapted to other weak proof systems such as Regular and Ordered Resolution [14], k-DNF Resolution [24], Cutting Planes [25], Nullstellensatz and Polynomial Calculus [23], the OBDD proof system [29] and, more recently, even $\mathbf{AC}^0$-Frege [41].

Though the latter works prove non-automatability under the optimal hardness assumption, their strength is incomparable to the cryptography-based results. The **NP**-hardness results all rely on proving specific super-polynomial proof complexity lower bounds for each system, meaning this strategy fails for $\mathbf{AC}^0[2]$-Frege and systems above, for which no lower bounds are known. In contrast, the cryptographic hardness results work by ruling out *feasible interpolation* for these systems, a property which allows one to extract computational content from proofs. For a proof system S proving its own soundness (such as $\mathbf{TC}^0$-Frege), feasible interpolation is equivalent to the notion of *weak* automatability introduced by Atserias and Bonet [8], the latter meaning that no proof system simulating S is automatable. The question of whether weak systems such as Resolution are weakly automatable remains one of the major open problems in the field. In short, there exists a trade-off between the strength of the hardness assumption involved ($\mathbf{P} \neq \mathbf{NP}$ versus cryptographic) and the generality of the result (automatability versus weak automatability).

Our work is the first new contribution to the non-automatability of strong proof systems[1] in more than two decades. The early results [37, 17, 16] relied on the assumption that factoring is hard, which does not hold for quantum models of computation due to Shor's breakthrough algorithm [49]. This raises the question of whether a quantum machine could carry out proof search efficiently for some strong proof system. Grover's search algorithm [27] already provides a quadratic speed-up over brute-force proof search for any system. While this is not enough to achieve automatability, the possibility of more powerful algorithms motivates the interest in new conditional hardness results. The **NP**-hardness results outlined above imply that $\mathbf{NP} \not\subseteq \mathbf{BQP}$ suffices to rule out automatability for weak systems, but for stronger systems no widely believed assumption had yet been proven sufficient.

In this work, we formally define the notion of *quantum automatability* and show the first hardness results. We prove that $\mathbf{TC}^0$-Frege is not quantum automatable unless lattice-based cryptography can be broken by polynomial-size quantum circuits. Our results follow from the relationship between automatability and feasible interpolation suitably generalized to the quantum setting. This means that we also rule out quantum feasible interpolation and weak quantum automatability under the same cryptographic assumptions.

Contributions

Our main contribution is proving the hardness of quantum automatability under the assumption that lattice-based cryptography is secure against quantum computers.

In 1996, Ajtai [2] gave the first worst-case to average-case reductions for lattice problems. In 1997, in joint work with Dwork [3], the worst-case hardness of such lattice problems was used to design public-key cryptosystems. Building on similar principles, the Learning with Errors (LWE) assumption of Regev [48] has become the standard post-quantum cryptographic assumption. The LWE assumption is simple to state, surprisingly versatile, and does not seem susceptible to the period-finding technique crucial to Shor's algorithm.

In this work we show that any quantum algorithm that automates $\mathbf{TC}^0$-Frege can be used to break LWE.

▶ **Theorem** (Main theorem, informal). *If there exists a polynomial-time quantum algorithm that weakly automates* $\mathbf{TC}^0$*-Frege, then LWE can be broken in polynomial time by a quantum machine.*

We then exploit the simulation of $\mathbf{TC}^0$-Frege by $\mathbf{AC}^0$-Frege proofs of subexponential size to extend the result to $\mathbf{AC}^0$-Frege under a slightly stronger assumption, in the style of [16].

▶ **Corollary.** *If there exists a polynomial-time quantum algorithm that weakly automates* $\mathbf{AC}^0$*-Frege, then LWE can be broken in subexponential time by a quantum machine.*

In order to properly state and prove these results, we first formally define the notion of quantum automatability for quantum Turing machines. Note that a quantum algorithm might provide a wrong answer with small probability, so we need to be careful in choosing the right definitions. We show that our definition is equivalent to a similar one over uniform quantum circuits, and we verify that it is robust by reproving Impagliazzo's observation that weak automatability implies feasible interpolation, suitably translated to the quantum setting.

[1] We use the terms *weak* and *strong* informally throughout the paper. Traditionally, a strong proof system is a system that proves its own soundness, though it is often also intended to be a system for which lower bounds are lacking. For our purposes, *strong* refers to anything simulating $\mathbf{TC}^0$-Frege, for which both of the previous conditions apply.

Techniques

The overall structure of the proofs follows the strategy of the previous non-automatability results of Krajíček and Pudlák [37] and Bonet, Pitassi, and Raz [17], but the technical details are quite different due to certain complications arising from lattice-based cryptography. We outline below the main hurdles and the techniques used to overcome them.

Quantum feasible interpolation

Our result follows from conditionally ruling out feasible interpolation by quantum circuits. As observed by Impagliazzo, weak automatability implies feasible interpolation. We use this observation contrapositively. Suppose that a proof system can prove the injectivity of a candidate one-way function. In the presence of feasible interpolation, we are guaranteed the existence of small circuits capable of inverting the one-way function one bit at a time. If one believes in the security of the cryptographic object, one must conclude that the proof system does not admit feasible interpolation, and in turn that it is not weakly automatable.

For this strategy to work the candidate one-way function should fulfill two important conditions. First, its definition must be simple enough that the proof system can easily reason about it. For example, RSA requires modular exponentiation to be defined, which is conjectured not to be computable in $\mathbf{TC}^0$. This forced Bonet, Pitassi, and Raz to use instead the Diffie-Hellman protocol. Second, the candidate one-way function must be injective. The rather technical reason for injectivity is that feasible interpolation allows one to carry out the inversion bit by bit, which does not guarantee retrieving a correct preimage if there are multiple ones.

A few injective one-way functions based on lattice geometry have been proposed throughout the literature, e.g., see [43, 26, 40]. However, we consider instead a simple scheme for worst-case lattice-based functions that closely resembles the one described by Micciancio [39]. Such a scheme has the advantage that its injectivity can be easily verified, and that its worst case one way-ness is guaranteed by the assumed hardness of Learning with Errors, which we will now discuss.

Learning with Errors and certificates of injectivity

We base our construction directly on the Learning with Errors assumption. The assumption is simple to define: roughly speaking, a vector x should be hard to recover after being multiplied by some public matrix A, and summed with some Gaussian noise, $Ax + \varepsilon$. While the most naive functions based on LWE are not necessarily injective, we can bound the magnitude of the error vectors to construct a family of functions where almost all of the functions are injective. For most matrices A, the corresponding function f_A in this family is worst-case one-way assuming the hardness of LWE [39].

However, the functions being injective and worst-case one-way is not sufficient, because their injectivity needs to be provable inside $\mathbf{TC}^0$-Frege. Unlike with the Diffie-Hellman construction, where a single proof showed the injectivity of the protocol for all generators, here each injective f_A may require a tailored proof of injectivity. Fortunately, most of these f_A can have their injectivity certified by a left-inverse of A together with a short basis for the dual lattice of the q-ary lattice spanned by A. These short bases not only certify injectivity, but can also be used as trapdoors to invert the function [42]. Though we do not exploit this directly, one may think of the automating algorithm as extracting such trapdoors from proofs. Instead, we use these certificates to prove the injectivity of most f_A inside $\mathbf{TC}^0$-Frege.

With these properties, we can show that feasible interpolation can be used to invert almost all f_A, which is sufficient to break LWE and its associated worst-case lattice problems.

Formal theories for linear algebra

The most technical component of the previous work on $\mathbf{TC}^0$-Frege and $\mathbf{AC}^0$-Frege was the formalization of many basic properties of arithmetic directly inside the propositional proof systems, which can be quite cumbersome. While we can borrow a large part of the existing formalization of Bonet, Pitassi, and Raz [17], putting it together to carry out arguments about lattice geometry would still be quite convoluted.

Instead, we follow the approach of Krajíček and Pudlák, who showed the injectivity of RSA in Extended Frege by reasoning in Buss's theory S^1_2 of bounded arithmetic. Universal theorems of this first-order theory translate into propositional tautologies with succinct proofs in Extended Frege. For $\mathbf{TC}^0$-Frege and its sequent calculus formalism PTK, the corresponding first-order theory of bounded arithmetic is the two-sorted theory VTC^0 introduced by Cook and Nguyen [20]. This theory is quite expressive and can reason even about analytic functions, as shown by Jeřábek [32]. However, since we are mostly interested in statements of matrix algebra, we use the more convenient formal theory LA for linear algebra introduced by Soltys and Cook [50].

The theory LA is quantifier-free and operates directly with matrices. It is strong enough to prove their ring properties, but weak enough to allow all theorems in LA to translate into propositional tautologies with short $\mathbf{TC}^0$-Frege proofs. In order to handle all the concepts required in our arguments, we work over a conservative extension of LA over the rationals which we show still propositionally translates into $\mathbf{TC}^0$-Frege.

Open problems

To the best of our knowledge, this is the first interaction between quantum computation and propositional proof search, and we believe further exploration of connections between the two fields is worthwhile. We outline below three open lines of research, ranging from the interaction between quantum computation and proof complexity to a classical problem in the theory of automatability.

Positive results?

While hardness of proof search in most natural proof systems is now conditionally ruled out under different assumptions, there exists a handful of systems for which no non-automatability results are known. This is the case for the Res($\oplus$), Res(log), Sherali-Adams and Sum-of-Squares proof systems. Could quantum algorithms automate any of these systems efficiently?

Even for proof systems where worst-case hardness is known, could quantum algorithms provide a significant speed-up over brute-force search? Clearly, Grover's algorithm already achieves a quadratic speed-up, but could this be pushed further in some cases?

Quantum proof complexity

Hardness results in automatability involve three key elements: the proof system, the hardness assumption and the model of computation for the automating algorithm. In this work we shifted the latter two to the quantum setting, by choosing a post-quantum cryptographic assumption and a quantum model of computation, but the proof systems considered remain classical.

What would it mean to have an inherently quantum proof system? In the same way that Extended Frege can be seen as $\mathbf{P}$/poly-Frege, could we define a proof system where lines are quantum circuits? This could open the door to a quantum analogue of the Cook-Reckhow

program, where showing lower bounds on quantum proof systems would be related to the question of whether **QMA** = **coQMA**. We note that an analogous approach exists in the field of parameterized complexity, starting with the work of Dantchev, Martin, and Szeider [22], who defined parameterized proof complexity as a program to gain evidence on the **W**-hierarchy being different from **FPT**. As an intermediate step, it would make sense to consider the case of randomized proof systems and the relationship between **MA** and **coMA**, though this has proven to be challenging so far.

We remark that while Pudlák [45] already defined the notion of quantum derivation rules for propositional proof systems and defined the quantum Frege proof system, his approach is orthogonal to ours, in that those systems are still designed to derive propositional tautologies. In fact, he showed that classical Frege systems simulate quantum Frege systems, though classical Frege proofs cannot be extracted from quantum proofs by a classical algorithm unless factoring is in **FP**.

Towards generic hardness assumptions

Like the original works on weak automatability, our proof requires *concrete* cryptographic assumptions. That is, we assume that some specific candidate one-way function or cryptographic protocol is secure. The reason is that in order to obtain the upper bounds on which to apply feasible interpolation we need concrete formulas to manipulate inside the different proof systems.

A major open problem in the theory of automatability is to disentangle these results from concrete families of candidate one-way functions. That is, can we prove that $\mathbf{TC}^0$-Frege is not (weakly) automatable under the assumption that, say, one-way functions exist? Even better, can one obtain **NP**-hardness of automating strong proof systems without the need to prove lower bounds first, in a way different from the strategy of Atserias and Müller [10]? This seems to require conceptual breakthroughs.

Structure of the paper

The paper is structured as follows. Section 2 recalls the necessary concepts in proof complexity and lattice-based cryptography needed in the rest of the paper. Section 3 defines automatability for quantum Turing machines and uniform quantum circuits and proves the equivalence between both models to then reprove Impagliazzo's observation on the relation between automatability and feasible interpolation, now in the quantum setting. Section 4 states and proves the main theorem of the paper. The section first presents a detailed overview of the main argument, while the subsections contain all the necessary technical work.

2 Preliminaries

We assume basic familiarity with computational complexity theory, propositional logic and quantum circuits. We review the main concepts needed from proof complexity and refer the reader to standard texts like [35] for further details. We also recall some relevant notions from linear algebra and lattice geometry useful in our arguments.

2.1 Proof complexity

Following Cook and Reckhow [21], a *propositional proof system* S for the language Taut of propositional tautologies is a polynomial-time surjective function $S : \{0,1\}^* \to \textsc{Taut}$. We think of S as a proof checker that takes some proof $\pi \in \{0,1\}^*$ and outputs $S(\pi) = \varphi$, the

theorem that π proves. Soundness follows from the fact that the range is exactly TAUT, and implicational completeness is guaranteed by the fact that S is surjective. One may alternatively define proof systems for refuting propositional contradictions. We move from one setup to the other depending on context.

We denote by $\mathsf{size}_S(\varphi)$ the *size* of the smallest S-proof of φ plus the size of φ. We say that a proof system S is *polynomially bounded* if for every $\varphi \in \text{TAUT}$, $\mathsf{size}_S(\varphi) \leq |\varphi|^{O(1)}$. We say that a proof system S *polynomially simulates* a system Q if for every $\varphi \in \text{TAUT}$, $\mathsf{size}_S(\varphi) \leq \mathsf{size}_Q(\varphi)^{O(1)}$. For a family $\{\varphi_n\}_{n\in\mathbb{N}}$ of propositional tautologies, we write $S \vdash \varphi_n$ whenever $\mathsf{size}_S(\varphi_n) \leq |\varphi|^{O(1)}$. Finally, a proof system S is said to be *closed under restrictions* if whenever S proves a formula φ in size s, for every partial restriction ρ to the variables in φ, there exists a proof of the restricted formula $\varphi_{\restriction\rho}$ in size $s^{O(1)}$.

The focus of this work is on a specific class of proof systems known as *Frege systems*. A Frege system is a finite set of axiom schemas and inference rules that are sound and implicationally complete for the language of propositional tautologies built from the Boolean connectives negation ($\neg$), conjunction ($\wedge$), and disjunction ($\vee$). A Frege proof is a sequence of formulas where each formula is obtained by either substitution of an axiom schema or by application of an inference rule on previously derived formulas. As long as the set of inference rules is finite, sound and implicationally complete, the specific choice of rules does not effect the size of the proofs up to polynomial factors, as all Frege systems polynomially simulate each other [35, Theorem 4.4.13].

We can make gradations between Frege systems by restricting the complexity of their proof lines. For a circuit class $\mathcal{C}$, the system $\mathcal{C}$-Frege is any Frege system where lines are restricted to be $\mathcal{C}$-circuits (see [31] for a formal definition). In this setup, a standard Frege system amounts to $\mathbf{NC}^1$-Frege. We are mostly interested in the weaker systems $\mathbf{AC}^0$-Frege and $\mathbf{TC}^0$-Frege, where the proof lines are, respectively, circuits of constant-depth and unbounded fan-in, and threshold circuits of constant-depth and unbounded fan-in. A *threshold circuit* is a Boolean circuit where gates can be the usual $\neg, \vee, \wedge$ as well as the threshold ones $\mathsf{Th}_k(x_1, \ldots, x_n)$, where Th_k is true if at least k of its inputs are true.

It is often convenient to consider an alternative formalism of $\mathbf{TC}^0$-Frege in the style of Gentzen's sequent calculus. The *Propositional Threshold Calculus* PTK [20, Chapter X.4.1] is a version of the propositional sequent calculus where the cuts are restricted to threshold formulas of constant depth. We refer to [17, Section 2] for a complete rendering of the derivational rules of PTK.

2.2 Lattice geometry

We recall some basic definitions from lattice geometry. For a linearly independent set of n vectors $\mathcal{B} = \{b_1, \ldots, b_n\} \subseteq \mathbb{R}^m$, which we often treat simply as an $m \times n$ matrix, the *lattice* over $\mathcal{B}$ is defined to be the set of all integer linear combinations of vectors in $\mathcal{B}$,

$$\mathcal{L}(\mathcal{B}) \coloneqq \{x \in \mathbb{R}^m \mid \text{there is } a \in \mathbb{Z}^n \text{ such that } x = \mathcal{B}a\}.$$

When the vectors in $\mathcal{B}$ belong in $\mathbb{Z}_q^m$ for some modulus q, we can further define a *modular lattice* over $\mathcal{B}$, denoted $\mathcal{L}_q(\mathcal{B})$, to be the set of all integer linear combinations of the basis modulo q,

$$\mathcal{L}_q(\mathcal{B}) \coloneqq \{x \in \mathbb{Z}_q^m \mid \text{there is } a \in \mathbb{Z}_q^n \text{ such that } \mathcal{B}a \equiv x \bmod q\},$$

where the mod function is applied element-wise in the vector.

We define the length of a vector x in $\mathcal{L}_q(\mathcal{B})$ to be the Euclidean norm of the shortest vector in $\mathbb{Z}^m$ that is congruent to x modulo q. Note that these shortest vectors will always fall in the domain $[-\lfloor q/2 \rfloor, \lfloor q/2 \rfloor]^m$.

A *q-ary lattice* $\Delta_q(\mathcal{B})$ can be thought of as an extension of a modular lattice back to $\mathbb{Z}^m$ and is the set of all vectors $x \in \mathbb{Z}^m$ congruent to members of the modular lattice,

$$\Delta_q(\mathcal{B}) \coloneqq \{x \in \mathbb{Z}^m \mid \text{there is } a \in \mathbb{Z}^n \text{ such that } \mathcal{B}a \equiv x \bmod q\}.$$

Note that because for all $x \in \{0, q\}^m$, $x \in \Delta_q(\mathcal{B})$, we have that all q-ary lattices have rank m.

From the definitions above it is clear that $\mathcal{L}_q(\mathcal{B}) \subseteq \Delta_q(\mathcal{B})$. Consequently a proof that no vector in $\Delta_q(\mathcal{B})$ has length less than ℓ also proves that no vector in $\mathcal{L}(\mathcal{B})$ has length less than ℓ.

Another important concept in lattice geometry is that of a *dual lattice*. Given a lattice $\mathcal{L}(\mathcal{B})$, its dual lattice $\mathcal{L}^*(\mathcal{B})$ is defined to be the set of vectors within the subspace spanned by $\mathcal{B}$ whose inner product with any element in $\mathcal{L}$ is an integer. Formally,

$$\mathcal{L}^*(\mathcal{B}) \coloneqq \{y \in \mathbb{R}^m \mid \text{there is } z \in \mathbb{R}^n \text{ such that } y = \mathcal{B}z \text{ and for all } x \in \mathcal{L}(\mathcal{B}), \langle x, y \rangle \in \mathbb{Z}\},$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product. The dual lattice is also a lattice, whose basis admits a closed form.

▶ **Lemma 1.** *For a basis $\mathcal{B} \in \mathbb{R}^{m \times n}$, $\mathcal{L}^*(\mathcal{B}) = \mathcal{L}(\mathcal{B}(\mathcal{B}^\intercal\mathcal{B})^{-1})$.*

Note that, if $\mathcal{B} \in \mathbb{Z}^{m \times n}$, it is easy to show that $\mathcal{B}(\mathcal{B}^\intercal\mathcal{B})^{-1} \in \mathbb{Q}^{m \times n}$, and, therefore, that any $x \in \mathcal{L}^*(\mathcal{B})$ belongs to $\mathbb{Q}^m$. This lemma is standard and can be found, for example, in [39].

Modular lattices and q-ary lattices are fairly different mathematical objects, but we can show that given a matrix $\mathcal{B} \in \mathbb{Z}_q^{m \times n}$ such that $\operatorname{rank}(\mathcal{B}) = n$, there exists a closed form for a matrix $\mathcal{B}'$ such that $\mathcal{L}(\mathcal{B}') = \Delta_q(\mathcal{B})$.

▶ **Lemma 2** (Full-rank modular lattices have q-ary lattice bases)**.** *Let $\mathcal{B} \in \mathbb{Z}_q^{m \times n}$ and define $C \in \{0,1\}^{m \times m}$ to be the permutation matrix that swaps the appropriate rows so that the first n rows of $C\mathcal{B}$ are linearly independent. Let $\mathcal{B}_1 \in \mathbb{Z}^{n \times n}$ and $\mathcal{B}_2 \in \mathbb{Z}^{m-n \times n}$ be matrices such that $\mathcal{B} = [C\mathcal{B}_1 \mid C\mathcal{B}_2]^\intercal$. Then, for $\mathcal{B} \in \mathbb{Z}_q^{m \times n}$, if $\operatorname{rank}(\mathcal{B}) = n$, $\Delta_q(\mathcal{B}) = \mathcal{L}(\mathcal{B}')$, where*

$$\mathcal{B}' = C \begin{bmatrix} I_n & 0 \\ (C\mathcal{B})_2 (C\mathcal{B})_1^{-1} & qI_{m-n} \end{bmatrix} C^{-1},$$

and where the inverses M^{-1} are defined over the modular lattice $\mathbb{Z}_q^m$.

Note that we can combine this corollary with Lemma 1 to get a closed form for $\mathcal{B}'$ such that $\Delta_q^*(\mathcal{B}) = \mathcal{L}(\mathcal{B}')$.

The *i-th successive minimum* of a lattice $\mathcal{L}$ is $\lambda_i(\mathcal{L}) \coloneqq \inf\{r \in \mathbb{Z} \mid \dim(\operatorname{span}(\mathcal{L} \cap B(0,r))) \geq i\}$, where $B(0,r)$ is the ball of radius r around the origin. Roughly speaking, this means that $\lambda_i(\mathcal{L})$ is the length of the i-th smallest linearly independent vector in the lattice.

There exists an intimate relationship between a lattice and its dual, as captured by Banaszczyk's Transference Theorem.

▶ **Theorem 3** (Transference Theorem [11])**.** *For any rank-n lattice $\mathcal{L} \subseteq \mathbb{Z}^m$, it holds that*

$$1 \leq \lambda_1(\mathcal{L}) \cdot \lambda_n(\mathcal{L}^*) \leq n.$$

Modular lattices $\mathcal{L}_q(\mathcal{B})$ are subsets of $\mathbb{Z}_q^m$, not $\mathbb{Z}^m$, and therefore the Transference Theorem does not directly apply. However we are able to leverage the fact that $\lambda_1(\Delta_q(\mathcal{B})) = \min(q, \lambda_1(\mathcal{L}_q(\mathcal{B})))$ to indirectly apply it through the q-ary lattice.

We recall useful properties of random lattices.

▶ **Lemma 4.** *For a randomly selected matrix $A \in \mathbb{Z}_q^{m\times n}$, we have that*

(i) $\Pr_A[\mathrm{rank}(A) = n] \geq n/q^{m-n+1}$;

(ii) $\Pr_A[\lambda_1(\mathcal{L}_q(A)) < r \mid \mathrm{rank}(A) = n] \leq (2r+1)^m/q^{m-2n-1}$.

These properties are folklore. For the sake of completeness, we provide proofs in Appendix C of the full version of the paper [7].

2.3 Learning with Errors (LWE)

Learning with Errors (LWE) is a central problem of learning theory, introduced by Regev [48].

▶ **Assumption 5** (The Learning with Errors (LWE) assumption [48, 42])**.** *Let $m = n^{O(1)}$, $q \leq 2^{n^{O(1)}}$, let $s \sim \mathbb{Z}_q^n$ be a secret vector, $A \sim \mathbb{Z}_q^{m\times n}$, and $\varepsilon \in \mathbb{Z}_q^m$ a sample from the discrete Gaussian with standard deviation αq with $\alpha = o(1)$ and $\alpha \in [0,1]$. The* Learning with Errors assumption *states that there is no quantum inverter M running in time $n^{O(1)}$ such that $M(A, As+\varepsilon)$ outputs s with noticeable probability over the choice of s, A, ε, and the internal randomness of M.*

The security of this assumption relies on the existence of worst-case to average-case reductions to fundamental lattice problems conjectured to be hard. In particular, as shown by Regev [48], breaking LWE implies solving the γ-GapSVP problem for an approximation factor $\gamma = n^2$. Here, γ-GapSVP refers to the *γ-Approximate Shortest Vector Problem*: given a lattice basis $\mathcal{B} \in \mathbb{Q}^{m\times n}$ and a distance threshold $r > 0$, decide whether $\lambda_1(\mathcal{L}(\mathcal{B})) \leq r$, or $\lambda_1(\mathcal{L}(\mathcal{B})) > \gamma r$, when one of those cases is promised to hold.

The belief that γ-GapSVP is intractable is backed by the fact that the problem is **NP**-hard under randomized reductions when the approximation factor is constant [2, 42, 15]. However, for the range of γ in which the reduction to LWE works, no **NP**-hardness is known. Obtaining **NP**-hardness for polynomial approximation factors would imply the breakthrough consequence of basing cryptography on worst-case hardness assumptions. In turn, this would turn our non-automatability results into **NP**-hardness results. As appealing as this might be, it is unlikely. For $\gamma \geq \sqrt{n}$, the problem γ-GapSVP is known to be in $\mathbf{NP} \cap \mathbf{coNP}$ [1] and thus cannot be **NP**-hard unless **PH** collapses.

2.4 The formal theory LA

The theory LA is a quantifier-free theory introduced by Soltys and Cook [50] whose main objects are matrices. This is not technically speaking a first-order theory of bounded arithmetic like those used by Krajíček and Pudlák [37], but like them it admits a propositional translation into Frege systems.

The system LA operates over three sorts: *indices* (intended to be natural numbers), *field elements* (over some abstract field $\mathbb{F}$), and *matrices* (with entries over $\mathbb{F}$). Variables for these three sorts are usually denoted $i, j, k, \ldots$ for indices, $a, b, c, \ldots$ for field elements, and $A, B, C, \ldots$, for matrices. We sometimes use lower-case letters $v, w, \ldots$ for vectors, which are seen as a special case of matrices.

The language of LA consists of the following constant, predicate and function symbols, over the three different sorts:

- Index sort: $0_{\text{index}}, 1_{\text{index}}, +_{\text{index}}, \cdot_{\text{index}}, -_{\text{index}}, \mathsf{div}, \mathsf{rem}, \mathsf{cond}_{\text{index}}, \leq_{\text{index}}, =_{\text{index}}$
- Field sort: $0_{\text{field}}, 1_{\text{field}}, +_{\text{field}}, \cdot_{\text{index}}, -_{\text{index}}, ^{-1}, \mathsf{r}, \mathsf{c}, \mathsf{e}, \Sigma, \mathsf{cond}_{\text{field}}, =_{\text{field}}$
- Matrix sort: $=_{\text{matrix}}$

The meaning of the symbols is the standard one, except for $-_{\text{index}}$ that denotes the cutoff subtraction ($i - j = 0$ if $i < j$) and for a^{-1}, denoting the inverse of a field element a, with $0^{-1} = 0$. For operations over matrices, $\mathsf{r}(A)$ and $\mathsf{c}(A)$ are, respectively, the number of rows and columns in A, $\mathsf{e}(A, i, j)$ is the field element $A_{i,j}$ (with $\mathsf{e}(A, i, j) = 0$ if either $i = 0$, $j = 0$, $i > \mathsf{r}(A)$ or $j > \mathsf{c}(A)$) and ΣA is the sum of the elements in A. The function symbol $\mathsf{cond}(\alpha, t_1, t_2)$ is interpreted to mean that if α holds, then the returned value should be t_1, else t_2, where α is a formula all of whose atomic subformulas have the form $m \leq n$ or $m = n$, where m and n are of the index sort, and t_1, t_2 are terms either both of index sort or both of field sort.

The language of LA can be enriched with the following defined terms: index maximum (max), matrix sum ($+$, when sizes of the matrices are compatible), scalar product ($\cdot$), matrix transpose ($A^\intercal$), zero (0) and identity matrices (I), matrix trace (tr), dot product ($\langle \_, \_ \rangle$), and matrix product ($\cdot$). See [50, Section 2.1] for details on the definitions of these terms. In general, whenever it is clear from context, we drop the subscripts indicating the sort and we use standard linear algebra notation for the sake of readability.

The theory then consists of several groups of axioms fixing the meaning of these symbols. These are rather lengthy to state, so we relegate them to Appendix B, where we also include several theorems derived by Soltys and Cook inside LA.

Observe that the theory is field-independent, but whenever we fix the field to be either finite or $\mathbb{Q}$, LA has the robust property that every theorem translates into a family of propositional formulas with short $\mathbf{TC}^0$-Frege proofs. This is the main property of LA that we shall exploit.

3 Quantum automatability and feasible interpolation

Following Bonet, Pitassi, and Raz [17], we say that a propositional proof system S is *automatable in time t* if there exists a deterministic Turing machine A that on input a formula φ outputs an S-proof of φ, if one exists, in time $t(\mathsf{size}_S(\varphi))$. We now consider the possibility of replacing A by a probabilistic or quantum Turing machine. The main issue in the definition is now that the output of the machine may be erroneous, albeit with small probability. Note, however, that if a machine were to output an incorrect proof, we would be able to easily detect this, since we can verify the proofs in polynomial time. We may thus assume that when yielding an incorrect proof, the machine will restart and find another one. Hence, instead of asking for the error-probability of the machine to be bounded, we ask for the expected running time to be bounded. The following definition captures this idea.

▶ **Definition 6** (Quantum and randomized automatability)**.** *Let S be a propositional proof system and let $t : \mathbb{N} \to \mathbb{N}$ be a time-constructible function. We say that S is* quantum *(respectively,* random*)* automatable in time t *or simply* quantumatable in time t *if there exists a quantum Turing machine (respectively, a randomized Turing machine) that on input a formula φ outputs an S-proof of φ, if one exists, in expected time $t(\mathsf{size}_S(\varphi))$.*

In what follows, we assume t to be a polynomial and talk simply about a system being *automatable* or *quantum automatable*, without reference to t. Since quantum circuits are often more convenient than quantum Turing machines, we also define automatability in the circuit setting.

▶ **Definition 7** (Circuit automatability). *Let S be a propositional proof system. We say that S is* circuit-automatable *if there exists a constant c and a uniform multi-output circuit family $\{C_{n,s}\}_{n,s\in\mathbb{N}}$ of size $(n+s)^c$ such that $C_{n,s}$ takes as input a formula φ of size n and outputs an S-proof of size s^c if a proof of size s exists, and is allowed to output any string otherwise.*

The generalization to randomized and quantum circuits is now immediate.

▶ **Definition 8.** *Let S be a propositional proof system. We say S is* quantum circuit-automatable *if there exists a constant c and a uniform multi-output quantum circuit family $\{C_{n,s}\}_{n,s\in\mathbb{N}}$ of size $(n+s)^c$ such that $C_{n,s}$ takes as input a formula φ of size n, and outputs an S-proof of size s^c with probability at least $2/3$ if a proof of size s exists, and is allowed to output any string otherwise. We say that S is* random circuit-automatable *if the circuit is classical but also takes as input a sequence r of random bits and, for at least $2/3s$ of the choices for r, $C_{n,s}(\varphi, r)$ outputs an S-proof of size s^c if a proof of size s exists, and is allowed to output any string otherwise.*

In fact, the machine-based and circuit-based definitions are equivalent.

▶ **Proposition 9.** *Let S be a propositional proof system. The following equivalences hold:*

(i) *the system S is automatable if and only if it is circuit-automatable;*
(ii) *the system S is random automatable if and only if it is random circuit-automatable;*
(iii) *the system S is quantum automatable if and only if it is quantum circuit-automatable.*

We defer the rather simple proof to Appendix B in the full version of the paper [7].

Even if a proof system is not automatable, one might still hope for an algorithm that finds some proof efficiently, even if it is in a different proof system. We say that a proof system S is *weakly automatable* if there exists another proof system Q and an algorithm A that given a formula φ, outputs a Q-proof of φ in time $\mathsf{size}_S(\varphi)^{O(1)}$. The concept was introduced by Atserias and Bonet [8], who further showed that this is equivalent to S being simulated by a system Q that is itself automatable. Despite the fact that weak automatability has been conditionally ruled out for Resolution under hardness assumptions for certain two-player games [9, 28, 13], establishing whether weak proof systems – such as Resolution – are weakly automatable under more standard hardness conjectures remains one of the main open problems in the area. It is straightforward to extend the notion of weak automatability to the quantum setting.

Weak automatability is closely related to feasible interpolation. We recall this connection in its classical form and then move to the quantum setting.

▶ **Definition 10** (Feasible interpolation [34, 46]). *We say that a proof system S has the* feasible interpolation property *if there exists a polynomial-time computable function I such that for every tautological split formula $\varphi(x,y,z) = \alpha(x,z) \vee \beta(z,y)$, whenever a proof π in S derives φ in size s, $I(\pi)$ produces an* interpolant circuit *C_φ of size $s^{O(1)}$ that takes as input an assignment ρ to the z-variables and such that*

$$C_\varphi(\rho) = \begin{cases} 0 & \text{only if } \alpha(x,\rho) \text{ is a tautology} \\ 1 & \text{only if } \beta(\rho,y) \text{ is a tautology} \end{cases}$$

indicating which side of the conjunction is tautological.

Bonet, Pitassi, and Raz attribute the following crucial observation relating (weak) automatability and feasible interpolation to Impagliazzo. We refer to it as *Impagliazzo's observation.*

▶ **Proposition 11** (Impagliazzo's observation [17, Thm. 1.1]). *If a proof system is weakly automatable and closed under restrictions, then it admits feasible interpolation.*

Impagliazzo's observation is useful contrapositively: to rule out (weak) automatability it suffices to rule out feasible interpolation, as done in the previous works [37, 17]. We outline this strategy further in Section 4, where we instantiate it together with our cryptographic assumption.

To use feasible interpolation in our setting, we suitably adapt the definition to the quantum world.

▶ **Definition 12** (Quantum feasible interpolation). *We say that a proof system S has the* quantum feasible interpolation property *if there exists a polynomial-time computable function I such that, for every tautological split formula $\varphi(x, y, z) = \alpha(x, z) \vee \beta(z, y)$, whenever a proof π derives φ in S in size s, $I(\pi)$ prints the description of a quantum interpolant circuit C_φ of size $s^{O(1)}$ as in Definition 10. If the circuit is instead randomized, we call this property* random feasible interpolation.

Interestingly, feasible interpolation is not affected by moving from classical automatability to randomized automatability. This is essentially folklore, but we reprove it for the sake of completeness.

▶ **Proposition 13.** *If a proof system S is weakly random automatable and closed under restrictions, then it has feasible interpolation by deterministic Boolean circuits.*

Proof. The proof is essentially the same as the original proof in [17], except for having to take randomness into account. Suppose R is a probabilistic automating algorithm for S. By Proposition 9.(ii), we can instead think of a family of randomized circuits $\{C_{n,s}\}_{n,s\in\mathbb{N}}$ that, for some fixed constant c, outputs proofs of size s^c when a proof of size s exists. Furthermore, let d be the constant in the exponent that bounds the blow-up in size happening in the closure under restrictions. Given a split formula $\varphi = \alpha \vee \beta$, we want to obtain an interpolant circuit C_φ.

Use the automating algorithm to find some proof of φ. Let s_0 be the size of such a proof. We first show that it is possible to extract a polynomial-size randomized circuit that computes the interpolant with one-sided error. Consider the circuit that takes as input the restriction ρ together with some random bits and proceeds to compute $C_{|\alpha|,s_0^d}(\alpha_{\restriction\rho}, r)$. If this circuit finds a proof of $\alpha_{\restriction\rho}$ and it is checked to be correct, we output 0; else, we output 1. We claim that for at least 2/3 choices of r, this circuit is a correct interpolant (and, in fact, whenever it outputs 0, it is always correct). First, note that if we output 0 it is because a proof of $\alpha_{\restriction\rho}$ was found, in which case it is correct to say that $\alpha_{\restriction\rho}$ is a tautology. Otherwise, we will always output 1. The only problematic case is when the circuit outputs 1 while $\neg\beta_{\restriction\rho}$ is satisfiable. If such was the case, then let σ be a satisfying assignment to the z-variables such that $\neg\beta_{\restriction\rho,\sigma}$ is satisfied. Since S can prove φ in size s_0 and S is closed under restrictions, we know that S can prove $\varphi_{\restriction\rho,\sigma}$ in size s_0^d, and this proof must clearly be deriving $\alpha_{\restriction\rho,\sigma} = \alpha_{\restriction\rho}$. Since $s_0^{cd} \geq s_0^d$, for a "good" choice of r the circuit $C_{|\alpha|,s_0^d}(\alpha_{\restriction\rho}, r)$ would have found such a proof, so the only reason why we could have output 1 is that we chose a bad r. But this of course only happens with probability at most 1/3. So this randomized circuit interpolates φ, makes only one-sided error, and has size polynomial in the size of the shortest proof.

We now replicate the strategy used in Adleman's theorem ($\mathbf{BPP} \subseteq \mathbf{P}/\mathrm{poly}$) to show that in fact randomness is not needed in the circuit. One can follow here the standard argument as presented, for example, by Arora and Barak [6, Thm. 7.15]: given the interpolant circuit F_φ, perform error reduction and then argue that there must be a string of random bits that is "good" for all inputs of the same size. The circuit no longer makes mistakes and computes f_φ as desired. ◀

▶ Remark 14 (Constructive feasible interpolation). Our definition of feasible interpolation deviates from the one given in standard texts like that of Krajíček [35], and follows instead the one given by Pudlák [46], who imposes the condition that the interpolant circuit must be constructed from the given proof in polynomial time. Note that even if we adopted the non-constructive definition, the kind of feasible interpolation obtained by the construction above achieves this property anyway.

The constructivity requirement is useful to obtain a sort of converse of Impagliazzo's observation: if a propositional proof system has uniform polynomial-size proofs of its reflection principle, then it is weakly automatable (see [46, Prop. 3.6]).

Since randomness does not buy us anything when it comes to proof search, all hardness results immediately transfer to the randomized setting. In particular, for every proof system S simulating $\mathbf{TC}^0$-Frege, S is not weakly random automatable unless Blum integers can be factored by polynomial-size randomized circuits. For weak proof systems where automatability is known to be **NP**-hard, the systems cannot be automatable unless $\mathbf{NP} \subseteq \mathbf{BPP}$.

When moving to the quantum setting, unfortunately, we do not know of any way to get a deterministic circuit for the interpolant. Instead, we have the following natural version of Impagliazzo's observation.

▶ **Proposition 15.** *If a proof system is quantum automatable and closed under restrictions, then it admits feasible interpolation by quantum circuits.*

Proof. The proof follows the argument in Proposition 13, except we can no longer apply the final step to get rid of quantumness. The interpolant now is a quantum circuit, since it is simulating the quantum circuit $C_{|\alpha|,s_0^d}(\alpha_{\restriction\rho})$. ◀

4 $\mathbf{TC}^0$-Frege is hard to quantum automate

The quantum version of Impagliazzo's observation (Proposition 15) is the main tool needed for our hardness results, which we are now ready to state formally.

▶ **Theorem 16** (Main theorem). *If there exists a polynomial-time quantum algorithm that weakly automates $\mathbf{TC}^0$-Frege, then the LWE assumption (Assumption 5) is broken by a uniform family of polynomial-size quantum circuits. Furthermore, if the weak automating algorithm is classical, the LWE assumption is broken by a uniform family of polynomial-size Boolean circuits.*

We can then extend the result to $\mathbf{AC}^0$-Frege under a stronger assumption. This is done by applying the fact that $\mathbf{TC}^0$-Frege proofs can be translated into $\mathbf{AC}^0$-Frege proofs of subexponential size (see, for example, Theorems 2.5.6 and 18.7.3 in [35] or the original work on the non-automatability of $\mathbf{AC}^0$-Frege [16]).

▶ **Corollary 17.** *If there exists a polynomial-time (quantum) algorithm that weakly automates $\mathbf{AC}^0$-Frege, then the LWE assumption is broken by a uniform family of (quantum) circuits of size $2^{n^{o(1)}}$.*

We devote the rest of the paper to formally proving Theorem 16.

Suppose $h : \{0,1\}^n \to \{0,1\}^n$ is an injective and secure one-way function. Let x, y and z denote variables ranging over $\{0,1\}^n$ and assume that $\mathbf{TC}^0$-Frege is able to state and refute efficiently the following unsatisfiable formula,

$$(h(x) = z \wedge x_1 = 0) \wedge (h(y) = z \wedge y_1 = 1)\,,$$

where x_1, y_1 are respectively the first bit of x and y. The unsatisfiability follows precisely from the fact that h is injective, and hence every output has a unique preimage.

If $\mathbf{TC}^0$-Frege admits feasible interpolation, we are guaranteed the existence of a small circuit $C(z)$ such that

$$C(z) = \begin{cases} 0 & \text{if } h(x) = z \wedge x_1 = 0 \text{ is unsatisfiable} \\ 1 & \text{if } h(y) = z \wedge y_1 = 1 \text{ is unsatisfiable} \end{cases}$$

meaning that C is able to invert one bit of z. Since every output has a unique preimage, we can iterate the process to get the entire input string. This contradicts the assumption that h is one-way.

In order to instantiate the proof strategy to rule out quantum feasible interpolation, we now need a candidate one-way function that is injective and conjectured to be post-quantum secure and for which injectivity can be proven inside the proof system. Unfortunately, to the best of our knowledge, no such candidate function is currently known, or not with enough security guarantees[2]. Alternatively, we may use other cryptographic objects that do achieve some form of injectivity, such as bit commitments, but the formalization of the latter does not seem simpler than the approach we follow instead. We now explain how we avoid this issue.

The most reliable post-quantum cryptographic assumptions have their security based on worst-case reductions to lattice problems conjectured to be hard. This is the case of the Learning with Errors framework [48], on which we base the security of the following class of candidate one-way functions. For these functions, as well as the basic properties of them that we employ, we follow the treatment of Micciancio [39]. We include the details for the proof complexity readers, who may not be familiar with these constructions.

▶ **Definition 18** (The candidate functions f_A)**.** *Let $m = n^{O(1)}$, $q \leq 2^{n^{O(1)}}$, and $c = \alpha q/\sqrt{n}$, where $\alpha \in [0,1]$. For every matrix $A \in \mathbb{Z}_q^{m \times n}$, we define the function $f_A : \mathbb{Z}_q^n \times \{\varepsilon \in \mathbb{Z}_q^m : |\varepsilon| \leq 10c\sqrt{mn}\} \to \mathbb{Z}_q^m$ as*

$$f_A(s, \varepsilon) \coloneqq (As + \varepsilon) \bmod q\,.$$

At this point, we would like to show inside $\mathbf{TC}^0$-Frege that the conjunction

$$(f_A(x) = z \wedge x_1 = 0) \wedge (f_A(y) = z \wedge y_1 = 1) \tag{1}$$

is a contradiction, where A is represented by free variables and x_1 and y_1 refer to the first bits of x and y. Unfortunately, the problem concerning injectivity mentioned above remains. The formula is not necessarily a contradiction, since for some choices of A, the function f_A is not injective. We can show, however, that with high probability over the choice of A, the function f_A will satisfy two conditions that imply injectivity. Namely, A will be full rank and the shortest vector in the q-ary lattice spanned by A will be large enough.

The following proposition, which captures this idea, is standard. We reprove it here for the sake of completeness, since we shall formalize part of it inside the proof systems later.

▶ **Proposition 19.** *Let $n \in \mathbb{N}, m = n \log n$ and $q \geq n^5$. With high probability over the choice of $A \in \mathbb{Z}_q^{m \times n}$, $\operatorname{rank}(A) = n$ and $\lambda_1(\mathcal{L}_q(A)) > 20c\sqrt{nm}$. Furthermore, when these hold, the function f_A is injective.*

[2] In a previous version of this work we formalized the injectivity of several group-based post-quantum cryptographic assumptions, such as MOBS [47], as well as variants of supersingular isogeny-based Diffie-Hellman protocols, which unfortunately all happen to be now broken more or less efficiently.

Proof. From Lemma 4.i we get that $\Pr_{A\sim\mathcal{U}(\mathbb{Z}_q^{m\times n})}[\text{rank}(A) < n] \leq n/q^{m-n+1}$. By Lemma 4.ii we can see that

$$\Pr_{A\sim\mathcal{U}(\mathbb{Z}_q^{m\times n})}[\lambda_1(\mathcal{L}(A)) \leq 20c\sqrt{mn} \mid \text{rank}(A) = n] \leq \frac{(40c\sqrt{mn}+1)^m}{q^{m-2n-1}}\,.$$

The probability that a random A does not satisfy the conditions in the statement is at most the sum of the two probabilities above, which are both negligible for our choice of m and q.

For injectivity, suppose for contradiction that there exist $x, x', \varepsilon, \varepsilon'$, with either $x \neq x'$ or $\varepsilon \neq \varepsilon'$, causing a collision $f_A(x,\varepsilon) = Ax + \varepsilon = Ax' + \varepsilon' = f_A(x',\varepsilon')$. We have two cases.

(a) If $\varepsilon = \varepsilon'$, then the collision happens if and only if $\text{rank}(A) < n$, which contradicts the assumption.

(b) Suppose that $\varepsilon \neq \varepsilon'$. We have that $\varepsilon - \varepsilon' = A(x' - x)$. Since the norm of $\varepsilon - \varepsilon'$ is at most $20c\sqrt{nm}$, by transitivity we have that the length of $A(x' - x)$ is bounded by the same quantity. However, the latter belongs to the lattice and therefore we obtain a contradiction. ◀

Luckily for us, these two conditions are succinctly certifiable! Indeed, to certify that the matrix A is full rank we may provide a left-inverse A_L^{-1} such that $A_L^{-1}A = I_n$. Unfortunately, we cannot guarantee that all injective f_A have simple certificates of the second property, $\lambda_1(\mathcal{L}_q(A)) > 20c\sqrt{nm}$. Nevertheless, we show in Section 4.2 that almost all of them do. These certificates take the form of sets $W = \{w_1, \ldots, w_m\} \subseteq \Delta_q^*(A)$ of short linearly independent vectors in the dual of the q-ary lattice. We prove – using the left inequality of Banaszczyk's Transference Theorem – that such a set suffices to certify the second property, and then show – using the right side of Banaszczyk's Transference Theorem – that the certificate W exists with high probability.

▶ **Definition 20** (Certificate of injectivity)**.** *A* certificate of injectivity *for the function f_A, with $A \in \mathbb{Z}_q^{m\times n}$, is a pair (A_L^{-1}, W) such that A_L^{-1} is a left-inverse so that $A_L^{-1}A = I_n$, and $W = \{w_1, \ldots, w_m\} \subseteq \Delta_q^*(A)$ is a set of m linearly independent vectors such that $\max_{i\in[m]} ||w_i|| < 1/20c\sqrt{nm}$.*

The relation between injectivity and these certificates is made formal as follows.

▶ **Proposition 21.** *Let $n \in \mathbb{N}, m = n\log n$, $q = n^5$, and $A \in \mathbb{Z}_q^{m\times n}$. The following hold:*

(i) *if there is a certificate of injectivity (A_L^{-1}, W) for f_A, then f_A is injective;*

(ii) *if $\text{rank}(A) = n$ and $\lambda_1(\mathcal{L}_q(A)) > 20mc\sqrt{nm}$, then there exists a certificate of injectivity for f_A;*

(iii) *with high probability over the choice of A, $\text{rank}(A) = n$ and $\lambda_1(\mathcal{L}_q(A)) > 20mc\sqrt{nm}$.*

Observe that given a certificate (A_L^{-1}, W), verifying its correctness is a rather simple task: it is sufficient check that $A_L^{-1}A = I_n$, to verify that W is a set of linearly independent vectors in $\Delta_q^*(A)$, and finally to ensure that the vectors in W are small enough.

Let us return to the propositional system. We denote by $\textsc{Inj}(f_A)$ the propositional formula encoding that f_A is injective. From this formula, $\mathbf{TC}^0$-Frege can derive that (1) is a contradiction. However, $\textsc{Inj}(f_A)$ is false if we leave A as free variables. We instead prove $\textsc{Inj}(f_{A_0})$ for concrete injective f_{A_0}, where A_0 is hardwired. The concrete f_{A_0} for which we do it are the ones that admit a certificate of injectivity.

Essentially, we formalize inside $\mathbf{TC}^0$-Frege that a certificate of injectivity implies injectivity. That is,

$$\mathbf{TC}^0\text{-Frege} \vdash \textsc{Cert}(C_A) \to \textsc{Inj}(f_A)\,, \tag{2}$$

where $\textsc{Cert}(C_A)$ encodes that C_A is a correct certificate for f_A. Here C_A and A are free variables. This implication is precisely Proposition 21.i above. The proof inside the system is carried out in Section 4.3.

Now, given a concrete certificate C_{A_0} for f_{A_0}, the formula $\textsc{Cert}(C_{A_0})$ is derivable inside $\mathbf{TC}^0$-Frege, which amounts to the system verifying the certificate's correctness. From this, $\mathbf{TC}^0$-Frege proves $\textsc{Inj}(f_{A_0})$.

The rest of this section completes the missing parts in the proof. Section 4.1 sketches the known fact that f_A is worst-case one-way based on the hardness of Learning with Errors, while Section 4.2 proves Proposition 21 showing the existence of certificates. We remark that the arguments and techniques are standard in cryptography and readers familiar with the area might want to skip them. We include them for the sake of completeness and to cater to the proof complexity reader that may have never come across these ideas before, and we refer to standard texts like [39] for further details. Finally, Section 4.3 formalizes the certificate-to-injectivity implication above inside the theory $\mathsf{LA}_{\mathbb{Q}}$, which propositionally translates into $\mathbf{TC}^0$-Frege. Section 4.4 reconstructs the final argument.

4.1 Security of f_A

The functions in $\{f_A\}_{A\in\mathbb{Z}_q^{m\times n}}$ very closely resemble the standard Learning with Errors functions, the only difference being that we have set a maximum value on the magnitude of the error vectors and allowed these to be chosen as a uniform part of the input (instead of being sampled from a Gaussian distribution). We now observe that inverting these functions allows us to invert LWE with high probability over the choice of the error vector.

▶ **Lemma 22** ([39, Section 3.2])**.** *Suppose there exists an algorithm B taking as input $A \in \mathbb{Z}_q^{m\times n}$ and a string z and outputting a preimage in $f_A^{-1}(z)$ with probability p. Then, LWE can be broken with probability $0.99p$ over the choice of the error vector ε and the internal randomness of B.*

Proof. It suffices to show that with high probability the error vectors in the standard Learning with Errors functions are bounded as in our definition of f_A, and thus the same inverter for f_A will also work for most of the original LWE instances. This follows from a standard Gaussian tail bound. Thus, if we are able to invert f_A on all outputs with probability p, then we are able to invert its corresponding LWE function with probability, say, $0.99p$ over the choice of ε. ◀

Note that it is in fact possible to invert with all but negligible probability, since finding a vector whose norm is far above the expectation with high probability requires that several independently sampled coordinates all return values much larger than the expected one. For simplicity, we use this weaker result which suffices for our applications.

4.2 Existence of certificates of injectivity: Proof of Proposition 21

This section proves the three statements of Proposition 21.

▶ **Proposition 21.i** (Correctness of certificates)**.** *If there is a certificate of injectivity (A_L^{-1}, W) for f_A, then $\operatorname{rank}(A) = n$ and $\lambda_1(\mathcal{L}_q(A)) > 20c\sqrt{nm}$, and thus the function f_A is injective.*

Proof. As discussed in the proof of Proposition 19, f_A is injective if and only if both $\operatorname{rank}(A) = n$ and $\lambda_1(\mathcal{L}_q(A)) > 20c\sqrt{mn}$. By elementary linear algebra, $\operatorname{rank}(A) = n$ if and only if there exists a A_L^{-1}. As previously observed we know that $\lambda_1(\Delta_q(A)) = \min(q, \lambda_1(\mathcal{L}(A)))$ and since $20c\sqrt{mn} \le q/m$, therefore it suffices to show that the existence of W as described above implies that $\lambda_1(\Delta_q(A)) > 20c\sqrt{mn}$.

Because it is a q-ary lattice we known that $\text{rank}(\Delta_q(A)) = m$. By rearranging the left inequality of the Transference Theorem for rank-m lattices, we get that $\lambda_1(\Delta_q(A)) \geq 1/\lambda_m(\Delta_q^*(A))$. By the definition of W, we conclude that $\lambda_m(\mathcal{L}^*) < 1/20c\sqrt{nm}$, which implies that $\lambda_1(\mathcal{L}_q(A)) = \lambda_1(\Delta_q(A)) > 20c\sqrt{nm}$.

Injectivity of f_A now immediately follows from the argument in Proposition 19. ◀

▶ **Proposition 21.ii** (Conditional existence of certificates). *If* $\text{rank}(A) = n$ *and* $\lambda_1(\mathcal{L}_q(A)) > 20mc\sqrt{nm}$*, then there exists a certificate of injectivity for* f_A*.*

Proof. Since we assumed that $\text{rank}(A) = n$, there exists a left inverse A_L^{-1} for A. It therefore suffices to show that if $\text{rank}(A) = n$ and $\lambda_1(\mathcal{L}_q(A)) > 20mc\sqrt{nm}$, then there exists a set of vectors W satisfying the conditions above.

By the right inequality of the Transference Theorem for rank-m lattices, we can obtain that $\lambda_m(\Delta_q^*(A)) \leq m/\lambda_1(\Delta_q(A))$. Since $\lambda_1(\mathcal{L}_q(A)) > 20mc\sqrt{nm} \leq q$, and $\lambda_1(\Delta_q(A)) = \min(q, \lambda_1(\mathcal{L}_q(A))) = \lambda_1(\mathcal{L}_q(A))$ there must exist a set of m linearly independent vectors in $\Delta_q^*(A)$, such that $\max_{i\in[m]} ||w_i|| < 1/20c\sqrt{nm}$. ◀

▶ **Proposition 21.iii** (Existence of certificates with high probability). *Let* $n \in \mathbb{N}, m = n \log n$*,* $q \geq n^5$*,* $c \leq \sqrt{nm}/40$ *and* $A \in \mathbb{Z}_q^{m\times n}$ *be sampled uniformly at random. The probability that* $\text{rank}(A) = n$ *and* $\lambda_1(\mathcal{L}_q(A)) > 20cm\sqrt{nm}$ *is at least*

$$1 - \frac{n}{q^{m-n+1}} - m^{3m-(m-2n-1)\log_m q}.$$

This probability is at least exponentially close to 1 *for our choice of* q *and* m*.*

Proof. For the following equations we define E_{short} to be the event that $\lambda_1(\mathcal{L}_q(A)) \leq 20cm\sqrt{nm}$. We have that

$$\begin{aligned}\Pr_A[\text{rank}(A) \neq n \vee \text{E}_{\text{short}}] &= \Pr_A[\text{rank}(A) \neq n] + \Pr_A[\text{E}_{\text{short}} \wedge \text{rank}(A) = n] \\ &\leq \Pr_A[\text{rank}(A) \neq n] + \Pr_A[\text{E}_{\text{short}} \mid \text{rank}(A) = n]\,.\end{aligned}$$

By Lemma 4.i, we know that $\Pr_A[\text{rank}(A) \neq n] \leq n/q^{m-n+1}$, and by the second point of Lemma 4.ii, we have that

$$\begin{aligned}\Pr[\text{E}_{\text{short}} \mid \text{rank}(A) = n] \leq \frac{(40mc\sqrt{mn})^m}{q^{m-2n-1}} &\leq \frac{(m^2 n)^m}{q^{m-2n-1}} \\ &\leq \frac{m^{3m}}{m^{(m-2n-1)\log_m q}} = m^{3m-(m-2n-1)\log_m q}\,.\end{aligned}$$ ◀

4.3 Formalization

At this point, the only thing left is the formalization of the implication $\textsc{Cert}(C_A) \to \textsc{Inj}(f_A)$ inside the propositional system. Since this is rather cumbersome, we work instead in the more convenient theory LA of linear algebra of Soltys and Cook [50]. The theory, however, is field-independent, which means we cannot state or prove properties about the ordering of the rationals, which is needed in our arguments. Furthermore, we sometimes use the fact that certain matrices are over the integers, so we must be able to identify certain elements as integers. To solve this, we introduce a conservative extension of the theory, called $\mathsf{LA}_{\mathbb{Q}}$, which assumes the underlying field to be $\mathbb{Q}$.

4.3.1 The conservative extension $\mathsf{LA}_{\mathbb{Q}}$

On top of the existing symbols of the language of LA, we have two new predicate symbols int and $<_{\mathbb{Q}}$. The int predicate, applied to a field element q, written $\mathsf{int}(q)$, is supposed to be true whenever the rational q is an integer.

The symbol $<_{\mathbb{Q}}$, which we overload onto $<$ in what follows, is intended to represent the usual ordering relation over the rationals. For convenience, we also add the symbol $x \leq y$ together with an axiom imposing that its meaning is $x < y \vee x = y$. Recall that equality of field elements was a symbol in the base theory LA, which already equipped it with its corresponding axioms.

We now extend the axiom-set of LA with axioms for the new symbols. Recall that the original axioms of LA are listed in Appendix B.

Axioms for int

- **(Int_1)** $\mathsf{int}(0)$
- **(Int_2)** $\mathsf{int}(1)$
- **(Int_3)** $\mathsf{int}(-1)$
- **(Int_4)** $\mathsf{int}(x) \wedge \mathsf{int}(y) \to \mathsf{int}(x + y)$
- **(Int_5)** $\mathsf{int}(x) \wedge \mathsf{int}(y) \to \mathsf{int}(x \cdot y)$
- **(Int_6)** $\mathsf{int}(x) \wedge 0 < x \to 1 \leq x$

Axioms for $<_{\mathbb{Q}}$

- **(Ord_1)** $x \leq y \leftrightarrow (x < y \vee x = y)$
- **(Ord_2)** $\neg(x < x)$
- **(Ord_3)** $x < y \to \neg(x = y)$
- **(Ord_4)** $x < y \wedge y < z \to x < z$
- **(Ord_5)** $\neg(x = y) \to x < y \vee y < x$
- **(Ord_6)** $x \leq y \wedge z \leq w \to x + z \leq y + w$
- **(Ord_7)** $0 \leq x \wedge 0 \leq y \to 0 \leq x \cdot y$
- **(Ord_8)** $0 \leq x \cdot x$
- **(Ord_9)** $0 \leq x \wedge y < 0 \to x \cdot y \leq 0$
- **(Ord_{10})** $a, b, c, d \geq 0 \wedge a < b \wedge c < d \to ac < bd$

The axioms for the ordering symbols are essentially the axioms of a strict total order (Ord_2-Ord_5), together with an axiom connecting $\leq$ and $<$ (Ord_1). We then ensure the compatibility of the operations with the ordering relation, (Ord_6-Ord_{10}). Our axioms are not necessarily minimal, since we are interested in convenience rather than succinctness.

The axioms for int are more ad hoc and it might seem that they are not enough to fix the correct interpretation of the symbol. Indeed, the axioms for int only really force that addition and multiplication are closed under this predicate and that every integer in the standard model can be argued to be an integer in $\mathsf{LA}_{\mathbb{Q}}$, but they do not identify $\mathbb{Z}$ as a substructure of $\mathbb{Q}$. The reason this is not an issue is that we are only interested in $\mathsf{LA}_{\mathbb{Q}}$ for its propositional translation. For our purposes, these axioms are the only ones we need to prove the required claims about lattices, and once we translate to the propositional setting, the symbols will take the standard intended interpretation.

Crucially, theorems of $\mathsf{LA}_{\mathbb{Q}}$ admit succinct $\mathbf{TC}^0$-Frege proofs. This requires extending the propositional translation of Soltys and Cook to the new symbols and axioms. We do this in detail in Appendix A.2 of the full version [7]. In this version, Appendix A contains a sketch the construction and the statement of the result.

4.3.2 Formalization of the proofs

We are ready to present the formal proofs needed inside our theory. In what follows, LA_._ stands for the corresponding axiom in Appendix B.

First, we observe that under our new axioms, $\mathsf{LA}_{\mathbb{Q}}$ can argue that the inner product of a vector with itself is non-negative. Recall that the inner product operator $\langle u, v\rangle$ is a defined term in LA, namely $u \cdot v^{\intercal}$.

▶ **Lemma 23.** *Provably in* $\mathsf{LA}_{\mathbb{Q}}$*, for every* $v \in \mathbb{Q}^n$*,* $0 \leq \langle v, v\rangle$*.*

Proof. Unfolding the definition of the term $\langle v, v\rangle$ in $\mathsf{LA}_{\mathbb{Q}}$, $\langle v, v\rangle = \sum_{i=1}^{n} v_i \cdot v_i$, so by axiom ($\mathrm{Ord}_8$) each term in the sum satisfies $v_i \cdot v_i \geq 0$, and by repeatedly applying axiom (Ord_6), the entire sum can be proven to be non-negative. ◀

We now formalize inside $\mathsf{LA}_{\mathbb{Q}}$ the classical Cauchy-Schwartz inequality.

▶ **Lemma 24** (Cauchy-Schwartz in $\mathsf{LA}_{\mathbb{Q}}$)**.** *The theory* $\mathsf{LA}_{\mathbb{Q}}$ *proves that for every* $u, v \in \mathbb{Q}^n$, $\langle u, v\rangle^2 \leq \langle u, u\rangle \cdot \langle v, v\rangle$.

Proof. We first show that $\mathsf{LA}_{\mathbb{Q}}$ can derive the following equality,

$$\frac{1}{\langle v, v\rangle}\langle(\langle v, v\rangle u - \langle u, v\rangle v), (\langle v, v\rangle u - \langle u, v\rangle v)\rangle = \langle u, u\rangle\langle v, v\rangle - \langle u, v\rangle^2, \tag{3}$$

where $\frac{1}{\langle v,v\rangle}$ can be explicitly referred to in $\mathsf{LA}_{\mathbb{Q}}$ as $\langle v, v\rangle^{-1}$.

We do this explicitly by deriving the following chain of equalities,

$$\frac{1}{\langle v, v\rangle}\langle(\langle v, v\rangle u - \langle u, v\rangle v), (\langle v, v\rangle u - \langle u, v\rangle v)\rangle = \tag{LA7.b}$$

$$\frac{1}{\langle v, v\rangle}(\langle(\langle v, v\rangle u - \langle u, v\rangle v), (\langle v, v\rangle u)\rangle + \langle(\langle v, v\rangle u - \langle u, v\rangle v), (-\langle u, v\rangle v)\rangle) = \tag{LA7.a-c}$$

$$\frac{1}{\langle v, v\rangle}(\langle v, v\rangle^2\langle u, u\rangle - \langle v, v\rangle\langle v, u\rangle^2) = \tag{LA7.c}$$

$$(\langle v, v\rangle\langle u, u\rangle - (\langle v, u\rangle^2).$$

Observe now that from Lemma 23 above, we know that $\langle(\langle v, v\rangle u - \langle u, v\rangle v), (\langle v, v\rangle u - \langle u, v\rangle v)\rangle$ is non-negative, and it is also clear that $0 \leq \frac{1}{\langle v,v\rangle}$: we know again that $\langle v, v\rangle \geq 0$, and $\langle v, v\rangle\langle v, v\rangle^{-1}$ is either 0 or 1, by LA3.d; if it is 0, we are done, and if it is 1, then by axiom (Ord_9) we can get a contradiction.

Thus, the left-hand side of Equation (3) is positive and $\mathsf{LA}_{\mathbb{Q}}$ can derive this fact. Then, by the transitivity axiom (Ord_4), the right-hand side of Equation (3) is non-negative as well. Rearranging the inequality using (Ord_6), the inequality follows. ◀

The other technical component needed in the final proof is a weakening of the lower bound in Banaszczyk's Transference Theorem (see Theorem 3). Informally, we need to prove that for every $A \in \mathbb{Q}^{m\times n}$, every non-zero vector $v \in \mathcal{L}(A)$ and any set of linearly independent vectors $W = \{w_1, \ldots, w_n\} \subseteq \mathcal{L}^*(A)$, $\langle v, v\rangle \cdot \langle w_i, w_i\rangle \geq 1$ for some $i \in [n]$.

In order for $\mathsf{LA}_{\mathbb{Q}}$ to process the conditions of the theorem, we provide certificate-like objects ensuring all the different hypotheses. For example, when we quantify over a vector v belonging to a lattice $\mathcal{L}(A)$, we provide the vector of coefficients c_v such that $Ac_v = v$. Note as well that when we quantify over matrices with elements in $\mathbb{Z}$, we are using the int predicate under the hood to enforce the entries to be integers. As a final remark, recall that we do not have existential quantifiers in LA, but whenever we do some existential quantification in the following lemmas we are quantifying over small finite domains, meaning we can write everything as a small disjunction.

▶ **Lemma 25** (Banaszczyk's left inequality in $\mathsf{LA}_{\mathbb{Q}}$)**.** *The theory* $\mathsf{LA}_{\mathbb{Q}}$ *proves the following implication. Let* $A \in \mathbb{Z}^{m\times n}$, $B \in \mathbb{Q}^{n\times n}$, $v \in \mathbb{Q}^n$, $c_v \in \mathbb{Z}^m$, $W = [w_1|\ldots|w_n] \in \mathbb{Q}^{m\times n}$, $c_W = [c_{w_1} \mid \cdots \mid c_{w_n}] \in \mathbb{Z}^{m\times n}$, $W' \in \mathbb{Q}^{m\times n}$ *fulfilling the following conditions:*

1. *the vector* v *is non-zero,* $v \neq 0_n$*;*
2. *the vector* v *belongs to the lattice* $\mathcal{L}(A)$*,* $v = Ac_v$*;*

3. *the vectors in W belong to the dual lattice*[3] $\mathcal{L}^*(A)$, $w_i = ABc_{w_i}$ *for all* $i \in [n]$*;*
4. $(A^\intercal A)B = I_n$*;*
5. *the vectors in W are linearly independent,* $W'W^\intercal = I_n$.
Then, for some $i \in [n]$, $\langle v, v\rangle \cdot \langle w_i, w_i\rangle \geq 1$.

Proof. The proof has two steps. First, we show that for all $i \in [n]$, $\langle v, w_i\rangle \in \mathbb{Z}$. To do this we use the following chain of equalities, where w is some arbitrary column w_i of W, and where the comments on the side refer to either axioms of $\mathsf{LA}_\mathbb{Q}$ or the assumptions in the statement of the lemma:

$$\begin{aligned}
\langle v, w\rangle &= \langle Ac_v, ABc_w\rangle && \text{(by ass. 2 and 3)}\\
&= c_v^\intercal A^\intercal ABc_w && \text{(by def. of dot product)}\\
&= c_v^\intercal (A^\intercal A)Bc_w && \text{(by associativity, LA5.i)}\\
&= c_v^\intercal c_w. && \text{(by ass. 4)}
\end{aligned}$$

By assumption, the entries in both c_v and c_w have integer entries, so by the closure under integer multiplication and addition (Int_4 and Int_5) we have that $c_v^\intercal c_w$ is an integer, and thus we deduce that for all $i \in [n]$, $\langle v, w_i\rangle \in \mathbb{Z}$.

In the second step of the proof, we show that there is $i \in [n]$ such that $\langle v, w_i\rangle \neq 0$. We consider the vector $s \coloneqq W^\intercal v$. Note that by definition, the j-th entry of s is $\langle w_j, v\rangle$. We can multiply both sides by the same matrix W', leading to $W's = W'W^\intercal v$. Using associativity (LA5.i), assumption (5) and properties of the identity matrix (LA5.f), we get that $W's = v$. Suppose that for all $i \in [n]$, $\langle v, w_i\rangle = 0$. Then, by definition, $s = 0_n$. We can easily derive (using LA3.a, LA3.c and LA3.i) that $W's = 0$ and therefore $v = 0$. This contradicts assumption (1).

Finally, let i denote the particular index for which we have now derived that simultaneously $\langle v, w_i\rangle \in \mathbb{Z}$ and $\langle v, w_i\rangle \neq 0$. By axiom ($\mathrm{Ord}_8$), $\langle v, w_i\rangle^2 \geq 0$. Furthermore, it is easy to derive already in LA that for any field elements a and b, if $a \neq 0$ and $b \neq 0$, then $ab \neq 0$ (this follows immediately from axioms LA3.a-d). Thus, by axiom (Ord_1), $\langle v, w_i\rangle^2 > 0$. Recall now that by axiom (Int_6) of $\mathsf{LA}_\mathbb{Q}$, every non-zero positive integer is greater or equal than 1, so $\langle v, w_i\rangle^2 \geq 1$. Then, the Cauchy-Schwartz inequality from Lemma 24 gives us

$$1 \leq \langle v, w_i\rangle^2 \leq \langle v, v\rangle \cdot \langle w_i, w_i\rangle,$$

which together with transitivity (axiom Ord_4 together with Ord_1) yields the desired $1 \leq \langle v, v\rangle \cdot \langle w_i, w_i\rangle$. ◀

We are now ready to prove in $\mathsf{LA}_\mathbb{Q}$ that a correct certificate of injectivity implies the injectivity of f_A. Informally, we aim to prove that given a certificate of injectivity as in Definition 20, the function f_A is injective. As before, we need to provide some additional objects together with the certificate to make sure $\mathsf{LA}_\mathbb{Q}$ can reason about this conditional implication and carry out the verification of the certificate.

▶ **Lemma 26** (Certificate-implies-injectivity in $\mathsf{LA}_\mathbb{Q}$)**.** *Let* $A \in \mathbb{Z}^{m\times n}$, $B \in \mathbb{Q}^{n\times n}$, $v_1 \in \mathbb{Q}^n$, $c_{v_1} \in \mathbb{Z}^m$, $v_2 \in \mathbb{Q}^n$, $c_{v_2} \in \mathbb{Z}^m$, $\varepsilon_1 \in \mathbb{Q}^n$, $\varepsilon_2 \in \mathbb{Q}^n$, $W = [w_1|\dots|w_n] \in \mathbb{Q}^{m\times n}$, $c_W = [c_{w_1} \mid \cdots \mid c_{w_n}] \in \mathbb{Z}^{m\times n}$, $W' \in \mathbb{Q}^{m\times n}$ *fulfilling the following conditions:*
1. *the vector v_1 belongs to the lattice $\mathcal{L}(A)$,* $v_1 = Ac_{v_1}$*;*
2. *the vector v_2 belongs to the lattice $\mathcal{L}(A)$,* $v_2 = Ac_{v_2}$*;*

[3] This dual lattice, in fact, admits a closed form for its base, as in Lemma 1. In particular, B can be seen as $(A^\intercal A)^{-1}$.

3. *the vectors* v_1 *and* v_2 *are distinct,* $v_1 \neq v_2$*;*
4. *the vectors in* W *belong to the dual lattice* $\mathcal{L}^*(A)$*,* $w_i = ABc_{w_i}$ *for all* $i \in [n]$*;*
5. $(A^\intercal A)B = I_n$*;*
6. *the vectors in* W *are linearly independent,* $W'W^\intercal = I_n$*;*
7. $\langle w_i, w_i \rangle < 1/400c^2nm$ *for all* $i \in [n]$*;*
8. $\langle \varepsilon_2 - \varepsilon_1, \varepsilon_2 - \varepsilon_1 \rangle < 400c^2nm$*.*

Then, $Av_1 + \varepsilon_1 \neq Av_2 + \varepsilon_2$*.*

Proof. The proof proceeds by contradiction. Suppose that $Av_1 + \varepsilon_1 = Av_2 + \varepsilon_2$, meaning that a collision exists in the range of f_A. By simple algebraic manipulations in $\mathsf{LA}_{\mathbb{Q}}$, we derive that $A(v_1 - v_2) = \varepsilon_2 - \varepsilon_1$. Let $v \coloneqq A(v_1 - v_2)$ and $\varepsilon \coloneqq \varepsilon_2 - \varepsilon_1$, so that we have $v = \varepsilon$.

Observe now that assumptions (7) and (8), together with axiom (Ord_{10}) and (LA3.d) give us $\langle w_i, w_i \rangle \langle \varepsilon, \varepsilon \rangle < 1$ for every $i \in [n]$. With the existing assumptions of the theorem we can in fact apply Lemma 25 to v, getting that there exists i such that $\langle v, v \rangle \langle w_i, w_i \rangle \geq 1$. Since $v = \varepsilon$, we get $\langle \varepsilon, \varepsilon \rangle \langle w_i, w_i \rangle \geq 1$, but this means

$$1 \leq \langle \varepsilon, \varepsilon \rangle \langle w_i, w_i \rangle < 1.$$

We remark that while technically the transitivity axiom (Ord_4) is stated for strict orders, the existing set of axioms immediately implies the "mixed" version, namely that for field elements a, b and c, if $a \leq b$ and $b < c$, then $a < c$. Thus we now have $1 < 1$, but this contradicts axiom (Ord_2). ◀

4.4 Proof of Theorem 16

We are ready to put all the pieces together.

Proof of Theorem 16. Suppose that $\mathbf{TC}^0$-Frege is weakly quantum automatable, that is, suppose that S is a quantum automatable proof system simulating $\mathbf{TC}^0$-Frege. Let Q be the quantum algorithm automating S. We describe a quantum algorithm Q' that takes as input a matrix A defining a function f_A as in Definition 18 and an output z of this function and succeeds in finding a preimage of z with high probability.

For a specific input matrix A_0, consider the formula $\textsc{Cert}(C_A) \to \textsc{Inj}(f_A)$, where C and A are free variables. In Lemma 26 this implication was proven inside $\mathsf{LA}_{\mathbb{Q}}$, and by the propositional translation for $\mathsf{LA}_{\mathbb{Q}}$ in Theorem 28 we get an efficient proof inside $\mathbf{TC}^0$-Frege, and thus also in S. Craft now the formula $\textsc{Inj}(f_{A_0})$ for the particular A_0 received as input. By Proposition 21, for most f_{A_0} there exists a certificate of injectivity C_{A_0} such that $\textsc{Cert}(C_{A_0})$ is true and, in fact, has no free variables. Consider this certificate as a partial restriction and apply it to the implication above. Since $\mathbf{TC}^0$-Frege is closed under restrictions, there must be a polynomial-size proof of $\textsc{Inj}(f_{A_0})$, and so S also proves this efficiently. Recall that, as noted in Remark 14, Impagliazzo's observation guarantees that under the existence of an automating algorithm we get constructive feasible interpolation, so from the proof of $\textsc{Inj}(f_{A_0})$ we can get a circuit that breaks one bit of the given output. By iterating this process we can recover the entire preimage. This procedure works as long as f_{A_0} is injective and admits a certificate of injectivity, but by Proposition 21 this is the case with overwhelming probability. Then, by Lemma 22, we break LWE and get the desired conclusion. ◀

The proof above is phrased from the starting assumption of a weak automating algorithm rather than feasible interpolation. The reason is that, intuitively, feasible interpolation alone does not seem to immediately break the cryptographic assumption: for every fixed matrix

A, feasible interpolation only seems to guarantee the existence (with high probability) of a circuit breaking f_A, but this circuit seems to essentially depend on A. By starting the argument from an automating algorithm, we have a uniform way of finding the proofs of injectivity for each particular f_A to then construct the corresponding interpolating circuit.

While we find this more intuitive, we can still phrase the argument directly in terms of interpolation (and hence rule out this too under the same assumption). It suffices to argue that $\mathbf{TC}^0$-Frege can refute the contradictory formulas

$$(f_A(x) = z \wedge x_i = 0) \wedge ((f_A(y) = z \wedge y_i = 1) \wedge \textsc{Cert}(C_A)),$$

where x_i and y_i refer to the i-th respective bits, and $\textsc{Cert}(C_A)$ is the certificate predicate, as in Equation (2). Observe that this is still a split formula, since the variables encoding the certificate C_A appear only on the right-hand side. That the proof system can show this is a contradiction follows immediately from the fact that it can prove the implication in Equation (2). More importantly, the refutation of this formula is uniform and known, with A as free variables, meaning we can extract the interpolants directly. It is not hard to see that interpolating on this formula we can still break the same functions that we would break with the aid of an automating algorithm. This remark is due to Impagliazzo. Thus, the following corollary also follows from our formalization.

▶ **Corollary 27.** *If $\mathbf{TC}^0$-Frege admits feasible interpolation by (quantum) circuits, then the LWE assumption can be broken by a uniform family of polynomial-size (quantum) circuits.*

References

1 Dorit Aharonov and Oded Regev. Lattice problems in $\mathbf{NP} \cap \mathbf{coNP}$. *Journal of the ACM (JACM)*, 52(5):749–765, 2005.

2 Miklós Ajtai. Generating hard instances of lattice problems. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 99–108, 1996.

3 Miklós Ajtai and Cynthia Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 284–293, 1997.

4 Michael Alekhnovich, Sam Buss, Shlomo Moran, and Toniann Pitassi. Minimum propositional proof length is **NP**-hard to linearly approximate. In *Mathematical Foundations of Computer Science 1998: 23rd International Symposium*, pages 176–184. Springer, 2006.

5 Michael Alekhnovich and Alexander A Razborov. Resolution is not automatizable unless $\mathbf{W}[\mathbf{P}]$ is tractable. *SIAM Journal on Computing*, 38(4):1347–1363, 2008.

6 Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

7 Noel Arteche, Gaia Carenini, and Matthew Gray. Quantum automating $\mathbf{TC}^0$-Frege is LWE-hard. *Electronic Colloquium on Computational Complexity (ECCC)*, 2024. URL: `https://eccc.weizmann.ac.il/report/2024/029/`.

8 Albert Atserias and María Luisa Bonet. On the automatizability of resolution and related propositional proof systems. *Information and Computation*, 189(2):182–201, 2004.

9 Albert Atserias and Elitza Maneva. Mean-payoff games and propositional proofs. *Information and Computation*, 209(4):664–691, 2011.

10 Albert Atserias and Moritz Müller. Automating resolution is **NP**-hard. *Journal of the ACM (JACM)*, 67(5):1–17, 2020.

11 Wojciech Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296:625–635, 1993.

12 P. Beame and T. Pitassi. Simplified and improved resolution lower bounds. In *Proceedings of 37th Conference on Foundations of Computer Science*, pages 274–282, 1996.

13 Arnold Beckmann, Pavel Pudlák, and Neil Thapen. Parity games and propositional proofs. *ACM Transactions on Computational Logic (TOCL)*, 15(2):1–30, 2014.

14 Zoë Bell. Automating regular or ordered resolution is **NP**-hard. *Electronic Colloquium on Computational Complexity (ECCC)*, 2020. URL: `https://eccc.weizmann.ac.il/report/2020/105/`.

15 Huck Bennett and Chris Peikert. Hardness of the (Approximate) Shortest Vector Problem: A Simple Proof via Reed-Solomon Codes. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2023)*, volume 275, pages 37:1–37:20, 2023.

16 María Luisa Bonet, Carlos Domingo, Ricard Gavaldà, Alexis Maciel, and Toniann Pitassi. Non-automatizability of bounded-depth Frege proofs. *computational complexity*, 13:47–68, 2004.

17 María Luisa Bonet, Toniann Pitassi, and Ran Raz. On interpolation and automatization for Frege systems. *SIAM Journal on Computing*, 29(6):1939–1967, 2000.

18 Sam Buss and Jakob Nordström. Proof complexity and SAT solving. *Handbook of Satisfiability*, 336:233–350, 2021.

19 Samuel R Buss. On Gödel's theorems on lengths of proofs II: Lower bounds for recognizing k symbol provability. In *Feasible mathematics II*, pages 57–90. Springer, 1995.

20 Stephen Cook and Phuong Nguyen. *Logical Foundations of Proof Complexity*. Cambridge University Press, 2010.

21 Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *Logic, Automata, and Computational Complexity*, 1979.

22 Stefan Dantchev, Barnaby Martin, and Stefan Szeider. Parameterized proof complexity. *Computational Complexity*, 20:51–85, 2011.

23 Susanna F. de Rezende, Mika Göös, Jakob Nordström, Toniann Pitassi, Robert Robere, and Dmitry Sokolov. Automating algebraic proof systems is **NP**-hard. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 209–222, 2021.

24 Michal Garlík. Failure of feasible disjunction property for k-DNF resolution and **NP**-hardness of automating it, 2020. `arXiv:2003.10230`.

25 Mika Göös, Sajin Koroth, Ian Mertz, and Toniann Pitassi. Automating cutting planes is **NP**-hard. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 68–77, 2020.

26 Rishab Goyal, Venkata Koppula, Satyanarayana Vusirikala, and Brent Waters. On perfect correctness in (lockable) obfuscation. In *Theory of Cryptography Conference*, pages 229–259. Springer, 2020.

27 Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, 1996.

28 Lei Huang and Toniann Pitassi. Automatizability and simple stochastic games. In *International Colloquium on Automata, Languages, and Programming*, pages 605–617. Springer, 2011.

29 Dmitry Itsykson and Artur Riazanov. Automating OBDD proofs is **NP**-hard. In *47th International Symposium on Mathematical Foundations of Computer Science (MFCS 2022)*, 2022.

30 Kazuo Iwama. Complexity of finding short resolution proofs. In *Mathematical Foundations of Computer Science 1997*, pages 309–318. Springer Berlin Heidelberg, 1997.

31 Emil Jeřábek. *Weak pigeonhole principle, and randomized computation*. PhD thesis, Faculty of Mathematics and Physics, Charles University, Prague, 2005.

32 Emil Jeřábek. Elementary analytic functions in VTC^0. *Annals of Pure and Applied Logic*, 174(6):103269, 2023.

33 Jan Krajíček. *Bounded Arithmetic, Propositional Logic and Complexity Theory*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1995.

34 Jan Krajíček. Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic. *The Journal of Symbolic Logic*, 62(2):457–486, 1997.

35 Jan Krajíček. *Proof Complexity*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2019.

36 Jan Krajíček. Information in propositional proofs and algorithmic proof search. *The Journal of Symbolic Logic*, 87(2):852–869, 2022.

37 Jan Krajíček and Pavel Pudlák. Some consequences of cryptographical conjectures for S^1_2 and EF. *Information and Computation*, 140(1):82–94, 1998.

38 Ian Mertz, Toniann Pitassi, and Yuanhao Wei. Short proofs are hard to find. In *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, 2019.

39 Daniele Micciancio. *The Geometry of Lattice Cryptography*, pages 185–210. Springer Berlin Heidelberg, 2011.

40 Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 700–718. Springer, 2012.

41 Theodoros Papamakarios. Depth d Frege systems are not automatable unless $\mathbf{P} = \mathbf{NP}$. *Electronic Colloquium on Computational Complexity (ECCC)*, 2023. URL: `https://eccc.weizmann.ac.il/report/2023/121/`.

42 Chris Peikert. A decade of lattice cryptography. *Foundations and Trends in Theoretical Computer Science*, 10(4):283–424, 2016.

43 Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 187–196, 2008.

44 Ján Pich and Rahul Santhanam. Learning Algorithms Versus Automatability of Frege Systems. In *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*, volume 229, pages 101:1–101:20, 2022.

45 Pavel Pudlák. Quantum deduction rules. *Annals of Pure and Applied Logic*, 157(1):16–29, 2009.

46 Pavel Pudlák. On reducibility and symmetry of disjoint **NP** pairs. *Theoretical Computer Science*, 295(1):323–339, 2003. Mathematical Foundations of Computer Science.

47 Nael Rahman and Vladimir Shpilrain. MOBS (Matrices Over Bit strings) public key exchange, 2021. `arXiv:2106.01116`.

48 Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):1–40, 2009.

49 P.W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, 1994.

50 Michael Soltys and Stephen Cook. The proof complexity of linear algebra. *Annals of Pure and Applied Logic*, 130(1):277–323, 2004. Papers presented at the 2002 IEEE Symposium on Logic in Computer Science (LICS).

A The propositional translation for $\mathsf{LA}_{\mathbb{Q}}$

We sketch the construction for the propositional translation, and defer the details to the full version. The translation is analogous to the usual propositional translations used elsewhere in bounded arithmetic (see, for example, [20, 33]). Let φ be a formula of $\mathsf{LA}_{\mathbb{Q}}$ and let σ be an object assignment that assigns a natural number to each free index variable occurring in φ and to each term of the form $\mathsf{r}(A)$ and $\mathsf{c}(A)$ occurring in φ, and substitutes every function and predicate symbol by the corresponding $\mathbf{TC}^0$ circuit of the appropriate size. We denote by $||\varphi||_\sigma$ the propositional formula obtained by carrying out this translation process.

▶ **Theorem 28** (Propositional translation for $\mathsf{LA}_{\mathbb{Q}}$)**.** *For every theorem φ of $\mathsf{LA}_{\mathbb{Q}}$ and every object assignment σ, the propositional formula $||\varphi||_\sigma$ admits polynomial-size $\mathbf{TC}^0$-Frege proofs.*

For the proof, see Theorem A.2 in the full version of the paper [7].

B Axioms and basic theorems of LA

1. **Equality axioms**
   a. $x = x$
   b. $x = y \to y = x$
   c. $(x = y \land y = z) \to x = z$
   d. $\bigwedge_i^n (x_i = y_i) \to f(\bar{x}) = f(\bar{y})$
   e. $i_1 = j_1, i_2 = j_2, i_1 \leq i_2 \to j_1 \leq j_2$.
2. **Axioms for indices**
   a. $i + 0 = i$
   b. $i + (j + 1) = (i + j) + 1$
   c. $i \cdot (j + 1) = (i \cdot j) + i$
   d. $i + 1 = j + 1 \to i = j$
   e. $i + 1 \neq 0$
   f. $i \leq i + j$
   g. $i \leq j, j \leq i$
   h. $i \leq j, i + k = j \to (j - i = k)$
   i. $i \leq j, i + k = j \to (i \not< j \to j - i = 0)$
   j. $j \neq 0 \to \mathsf{rem}(i, j) < j$
   k. $j \neq 0 \to i = j \cdot \mathsf{div}(i, j) + \mathsf{rem}(i, j)$
   l. $\alpha \to \mathsf{cond}(\alpha, i, j) = i$
   m. $\neg\alpha \to \mathsf{cond}(\alpha, i, j) = j$
3. **Axioms for field elements**
   a. $0 \neq 1 \land a + 0 = a$
   b. $a + (-a) = 0$
   c. $1 \cdot a = a$
   d. $a \neq 0 \to a \cdot (a^{-1}) = 1$
   e. $a + b = b + a$
   f. $a \cdot b = b \cdot a$
   g. $a + (b + c) = (a + b) + c$
   h. $a \cdot (b \cdot c) = (a \cdot b) \cdot c$
   i. $a \cdot (b + c) = a \cdot b + a \cdot c$
   j. $\alpha \to \mathsf{cond}(\alpha, a, b) = a$
   k. $\neg\alpha \to \mathsf{cond}(\alpha, a, b) = b$
4. **Axioms for matrices**
   a. $(i = 0 \lor \mathsf{r}(A) < i \lor j = 0 \lor \mathsf{c}(A) < j) \to \mathsf{e}(A, i, j) = 0$
   b. $\mathsf{r}(A) = 1, \mathsf{c}(A) = 1 \to \Sigma(A) = \mathsf{e}(A, 1, 1)$
   c. $\mathsf{c}(A) = 1 \to \sigma(A) = \sigma(A^\intercal)$
   d. $\mathsf{r}(A) = 0 \lor \mathsf{c}(A) = 0 \to \Sigma(A) = 0$
5. **Theorems for ring properties**
   a. $\max(i, j) = \max(j, i)$
   b. $\max(i, \max(j, k)) = \max(\max(i, j), k)$
   c. $\max(i, \max(j, k)) = \max(\max(i, j), \max(i, k))$
   d. $A + 0 = A$
   e. $A + (-1)A = 0$
   f. $AI = A$ and $IA = A$
   g. $A + B = B + A$
   h. $A + (B + C) = (A + B) + C$
   i. $A(BC) = (AB)C$
   j. $A(B + C) = AB + CA$
   k. $(B + C)A = BA + CA$
   l. $\Sigma 0 = 0_{\text{field}}$
   m. $\Sigma(cA) = c\Sigma(A)$
   n. $\Sigma(A + B) = \Sigma(A) + \Sigma(B)$
   o. $\Sigma(A) = \Sigma(A^\intercal)$
6. **Theorems for module properties**
   a. $(a + b)A = aA + bA$
   b. $a(A + B) = aA + aB$
   c. $(ab)A = a(bA)$
7. **Theorems for inner product**
   a. $A \cdot B = B \cdot A$
   b. $A \cdot (B + C) = A \cdot B + A \cdot C$
   c. $aA \cdot B = a(A \cdot B)$
8. **Miscellaneous theorems**
   a. $a(AB) = (aA)B \land (aA)B = A(aB)$
   b. $(AB)^\intercal = B^\intercal A^\intercal$
   c. $I^\intercal = I$
   d. $0^\intercal = 0$
   e. $(A^\intercal)^\intercal = A$