Lattice-Based Proof-of-Work for Post-Quantum Blockchains

Rouzbeh Behnia[1], Eamonn W. Postlethwaite[2], Muslum Ozgur Ozmen[3*], and Attila Altay Yavuz[1]

[1] University of South Florida, Tampa, Florida, USA
{behnia,attilaayavuz}@usf.edu
[2] Information Security Group, Royal Holloway, University of London
eamonn.postlethwaite.2016@rhul.ac.uk
[3] Purdue University, West Lafayette, Indiana, USA
mozmen@purdue.edu

Abstract. Proof of Work (PoW) protocols, originally proposed to circumvent DoS and email spam attacks, are now at the heart of the majority of recent cryptocurrencies. Current popular PoW protocols are based on hash puzzles. These puzzles are solved via a brute force search for a hash output with particular properties, such as a certain number of leading zeros. By considering the hash as a random function, and fixing *a priori* a sufficiently large search space, Grover's search algorithm gives an asymptotic quadratic advantage to quantum machines over classical machines. In this paper, as a step towards a fuller understanding of post quantum blockchains, we propose a PoW protocol for which quantum machines have a smaller asymptotic advantage. Specifically, for a lattice of rank n sampled from a particular class, our protocol provides as the PoW an instance of the Hermite Shortest Vector Problem (Hermite-SVP) in the Euclidean norm, to a small approximation factor. Asymptotically, the best known classical and quantum algorithms that directly solve SVP type problems are heuristic lattice sieves, which run in time $2^{0.292n+o(n)}$ and $2^{0.265n+o(n)}$ respectively. We discuss recent advances in SVP type problem solvers and give examples of where the impetus provided by a lattice based PoW would help explore often complex optimization spaces.

Keywords: Blockchains · Proof-of-work · Post-quantum cryptography · Consensus protocols · Lattice-based cryptography · Shortest vector problem

1 Introduction

Consensus mechanisms are at the heart of the decentralized nature of blockchains. Proofs of Work (PoW), based on computational power, and Proofs of Stake (PoS), based on some notion of "stake" in the system, are amongst the most common types of consensus mechanisms. Cryptocurrencies like Bitcoin [21] and

[*]Work done in part when Muslum Ozgur Ozmen was at University of South Florida.

Ethereum [26] rely on PoW based on brute force hash computations to ensure decentralized trust, at the cost of terawatts of energy.[4] The hash functions used in such cryptocurrencies achieve desirable security properties against quantum adversaries when modelled as a random oracle [25]. Despite this, Grover's search algorithm [15] gives an asymptotic advantage to quantum computers when solving hash based PoWs. While some advantage over classical computers may agree with the nature of PoW protocols (more expensive or powerful machines should perform better), we consider it a valuable research topic to reduce this advantage, e.g. because quantum computers may exist for some time before being available to the public.

Our Contributions. *The main goal of this paper is to address the research gap in the state-of-the-art by creating a novel consensus protocol (specifically, a PoW algorithm) that reduces the advantage of quantum computers over classical ones, has fast verification, and adjustable difficulty.* To achieve this goal, we propose a new PoW protocol called `LPoW` based on the Hermite-SVP problem. Given the current understanding of SVP type problems, `LPoW` satisfies the following properties [1, Section IV]:

- `LPoW` provides little quantum advantage; the asymptotic quantum advantage against SVP is less than the quadratic speed up of Grover's algorithm.
- `LPoW` is hard to solve but easy to verify. Solving is equivalent to solving Hermite-SVP to a small approximation factor. Verifying is equivalent to calculating a norm, an n^{th} root, and some multiplications.
- The parameters of `LPoW` are easy to fine tune to adjust its difficulty. In particular increasing the dimension of the lattice has a well studied effect on the computational resources required to solve the PoW.

A secondary goal of the this paper is to create a PoW protocol that encourages further experimentation with, and understanding of, practical algorithmic improvements for solving SVP type problems. In [16], the authors suggest harnessing both the energy spent on hash puzzles, and the demand to mine cryptocurrencies, to improve the state-of-the-art in discrete log cryptanalysis. Following [16], and given that the difficulty of SVP is fundamental to the security of lattice based submissions to NIST's post quantum standardization process,[5] an SVP based PoW can similarly leverage this energy and demand to aid in the cryptanalysis of the SVP problem. In Section 4 we discuss several areas of SVP solving strategies which could benefit from increased attention.

Limitations. If we assume a given hash function is a random oracle, then Grover's algorithm gives the optimal speedup against PoW based on this hash function, and cannot be parallelized except in the trivial manner [27]. Effectively this means that the PoW parameters, e.g. the number of leading zeros required in the hash output, will only have to increase to account for increased computational strength, and not fundamentally new algorithmic techniques. This is not necessarily the case for the specific lattice problem we consider; we do not

[4] https://digiconomist.net/bitcoin-energy-consumption/.

[5] https://csrc.nist.gov/projects/post-quantum-cryptography

have any proofs of optimality for the algorithms currently used to solve it. In effect, this means that the PoW parameters, i.e. the lattice rank, may need to be increased to account for algorithmic improvements, as well as for increased computational strength. We note that the best known time complexity for solving the SVP puzzles we consider is $2^{\Theta(n)}$, for lattices of rank n, and that any change to even slightly subexponential in n would represent a huge moment in the theory of lattices. Therefore, we do not expect to have to increase the rank too much, even to account for any algorithmic improvements.

2 Preliminaries

For $n \in \mathbb{N}^+$ let $[n] = \{1, \dots, n\}$. For a finite set S, let $x \leftarrow \mathcal{U}(S)$ denote a uniform sample. Let $m(n)$ represent the cost of multiplying two n bit numbers.

Proof of work protocols enable a prover to prove to a verifier that it has executed a certain amount of work. We adopt the definition of such protocols from [5], which consists of algorithms that *generate* a challenge, *solve* such a challenge, thereby producing a proof of solution, and finally *verify* that this proof is correct. This triple of algorithms must satisfy the following.

Definition 1. *A* $(t(n), \delta(n))$*-Proof of Work (PoW) consists of three algorithms* $(\texttt{Gen}, \texttt{Solve}, \texttt{Verify})$ *that satisfy the following.*

- ***Efficiency:***
  - $\texttt{Gen}(1^n)$ *runs in time* $\tilde{O}(n)$.
  - *For any* $c \leftarrow \texttt{Gen}(1^n)$, $\texttt{Solve}(c)$ *runs in time* $\tilde{O}(t(n))$.
  - *For any* $c \leftarrow \texttt{Gen}(1^n)$, $\Pi \leftarrow \texttt{Solve}(c)$, $\texttt{Verify}(c, \Pi)$ *runs in time* $\tilde{O}(n)$.
- ***Completeness:*** *For any* $c \leftarrow \texttt{Gen}(1^n)$ *and any* $\Pi \leftarrow \texttt{Solve}(c)$,

$$\Pr[\texttt{Verify}(c, \Pi) = acc] = 1,$$

  with the probability taken over the randomness of $\texttt{Verify}$.[6]
- ***Hardness:*** *For any polynomial* l, *any constant* $\epsilon > 0$, *and any algorithm* $\texttt{Solve}_l^*$ *that runs in time* $l(n)t(n)^{1-\epsilon}$ *when given as input* $l(n)$ *challenges* $\{c_i \leftarrow \texttt{Gen}(1^n)\}_{i \in [l(n)]}$,

$$\Pr\left[\texttt{Verify}(c_i, \Pi_i) = acc, \forall i \,\middle|\, (\Pi_1, \dots, \Pi_{l(n)}) \leftarrow \texttt{Solve}_l^*(c_1, \dots, c_{l(n)})\right] < \delta(n),$$

  with the probability taken over the randomness of $\texttt{Gen}$ *and* $\texttt{Verify}$.

Efficiency ensures that verification runs in (near) linear time. Efficiency and completeness together ensure that a prover that performs roughly $t(n)$ operations can prove to the verifier that it has done so. Hardness requires that the prover has, e.g. a negligible chance, for δ some negligible function of n, to convince the verifier without performing $l(n)t(n)$ operations. This remains true, even if the prover may compute on the $l(n)$ challenges together.

[6] We note that our `Verify` is deterministic.

2.1 Lattices and Lattice Problems

An n-dimensional lattice Λ of rank $k \leq n$ is a discrete additive subgroup of $\mathbb{R}^n$. Given k linearly independent basis vectors $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k\} \subset \mathbb{R}^n$, the lattice generated by $\boldsymbol{B}$, i.e. their concatenation as column vectors, is

$$\Lambda(\boldsymbol{B}) = \Lambda(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k) = \left\{ \sum_{i=1}^{k} x_i \cdot \boldsymbol{b}_i \colon x_i \in \mathbb{Z} \right\}.$$

The volume of some Λ is defined as $\mathrm{Vol}(\Lambda) = \sqrt{\det(\boldsymbol{B}^t \boldsymbol{B})}$ for any basis $\boldsymbol{B}$ of Λ (i.e. volume is an invariant of the lattice, and independent of the choice of basis). We will consider only full rank lattices, where $n = k$ and $\mathrm{Vol}(\Lambda) = \det(\boldsymbol{B})$.

Definition 2. *The* minimum distance *of a lattice Λ is*

$$\lambda_1(\Lambda) = \min \{ \|\boldsymbol{v}\| : \boldsymbol{v} \in \Lambda \setminus \{0\} \}.$$

A solution to γ-approx-SVP is a vector $\boldsymbol{v} \in \Lambda$ such that $\|\boldsymbol{v}\| \leq \gamma \cdot \lambda_1(\Lambda)$.

An immediate corolloary of Minkowski's theorem, in the Euclidean norm, proves that $\lambda_1(\Lambda) \leq \sqrt{n} \det \mathrm{Vol}(\Lambda)^{1/n}$. The *Gaussian heuristic* estimates the number of lattice points of a lattice Λ contained in a measurable set S as $\mathrm{Vol}(S)/\mathrm{Vol}(\Lambda)$. When applied to a hypersphere it gives the following estimate for $\lambda_1(\Lambda)$.

Definition 3. *Let the Gaussian heuristic estimate, $gh(\Lambda)$, for $\lambda_1(\Lambda)$ be given using the Gamma function, as*

$$gh(\Lambda) = \frac{\Gamma(n/2+1)^{1/n}}{\sqrt{\pi}} \cdot Vol(\Lambda)^{1/n}.$$

Definition 4. *The α-Hermite-SVP, or α-HSVP problem is, given a lattice Λ, to find a vector $\boldsymbol{v} \in \Lambda$ such that*

$$\|\boldsymbol{v}\| \leq \alpha \cdot Vol(\Lambda)^{1/n}.$$

Instances of Hermite-SVP are given as a lattice basis, which much be somehow sampled. We generate Goldstein–Mayer lattices [14] as $\Lambda(B)$ for

$$B = \begin{pmatrix} p & x_2 & \cdots & x_n \\ 0 & 1 & \cdots & 0 \\ \vdots & & \ddots & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{1}$$

where p is a large prime and $x_i \leftarrow \mathcal{U}(\{0\} \cup [p-1])$ are i.i.d. uniform. These lattices have $\mathrm{Vol}(\Lambda) = p$ and provide a way to sample "uniformly" from all lattices of this volume [8, Section 2.3]. For example, the Darmstadt SVP Challenge[7] uses

[7] https://www.latticechallenge.org/svp-challenge/

$\log_2 p \approx 10n$ and sets $\alpha = 1.05 \cdot \Gamma(n/2+1)^{1/n}/\sqrt{\pi}$. This α is such that a solution to α-HSVP has length at most $1.05 \cdot \mathrm{gh}(\Lambda)$ and also $\alpha \approx 0.254 \cdot \sqrt{n}$. This is a constant factor smaller than the approximation, $\alpha = \sqrt{n}$, that guarantees a solution. However, the Gaussian heuristic on random lattices is accurate, and we expect 1.05^n lattice vectors of length at most $1.05 \cdot \mathrm{gh}(\Lambda)$ for $n \gtrsim 50$ [8, Section 3.1].

3 Proposed PoW Protocol, LPoW

Before we propose our new PoW protocol, we give a brief précis of how instances of Hermite-SVP problems are solved. One can solve SVP on Λ using a variety of families of algorithms. The family we consider is heuristic lattice sieves, which have the best known classical and quantum time complexity, standing at $2^{0.292n+o(n)}$ [6] and $2^{0.265n+o(n)}$ [18] respectively. However, it is not necessary to call lattice sieves in the full dimension of the lattice to solve SVP type problems [10]. Instead sieving in dimension $n - \Theta(n/\log n)$ suffices under certain heuristic assumptions. There also exist many further heuristic techniques that provide significant practical speedups [19,24]. Finally, a framework that collates, extends, and implements these techniques holds the record for the highest dimension SVP challenge solved [2]. The techniques mentioned above depend non trivially on the "quality" of the lattice basis being used, informally; how short and close to orthogonal its basis vectors are. Therefore lattice reduction algorithms such as BKZ are employed [23,9], which themselves require SVP oracles for lower dimensional projected sublattices. The constant supressed in $\Theta(n/\log n)$ above will depend on the methods used to determine the quality of the basis. Some experimental values may be found in [2, Fig. 3b].

The high level design of our PoW follows Definition 1. We set n as the dimension of the lattice and let α, p follow the Darmstadt SVP Challenges.

Definition 5. *Let* LPoW *be defined by the following triple* (Gen, Solve, Verify)*.*

- $\texttt{Gen}(1^n; r)$, *let the randomness r be explicit and derived from the previous block. First, sample a prime p of bitsize $10n$, then sample sample i.i.d. uniform $x_2, \ldots, x_n \leftarrow \mathcal{U}(\{0\} \cup [p-1])$, to form a basis $\boldsymbol{B}$ as in (1). Let $\alpha = 1.05 \cdot \Gamma(n/2+1)^{1/n}/\sqrt{\pi}$. Return $c = (\alpha, n, \boldsymbol{B}, p)$.*
- $\texttt{Solve}(c)$, *the miner parses c as $(\alpha, n, \boldsymbol{B}, p)$ and attempts to find a vector $\boldsymbol{v} \in \Lambda(\boldsymbol{B})$ such that $\|\boldsymbol{v}\| \leq \alpha \cdot p^{1/n}$. It outputs $\Pi = (\boldsymbol{v}, \boldsymbol{\nu})$, where $\boldsymbol{v} = \boldsymbol{B} \cdot \boldsymbol{\nu}$.*
- $\texttt{Verify}(c, \Pi)$, *parses c as $(\alpha, n, \boldsymbol{B}, p)$ and Π as $(\boldsymbol{v}, \boldsymbol{\nu})$, and outputs*

$$acc = \|\boldsymbol{v}\| \leq \alpha \cdot p^{1/n} \wedge \boldsymbol{v} = \boldsymbol{B} \cdot \boldsymbol{\nu} \wedge \boldsymbol{\nu} \in \mathbb{Z}^n.$$

We use an extendable output function, e.g. [12], to extract sufficient randomness from the previous block to sample the required quantities in Gen. In the following, we weaken ever so slightly the efficiency requirements of Gen and Verify. For Gen it is not known how to generate an n bit prime, either probably or provably, in $\tilde{O}(n)$. Indeed, the prime number theorem tells us that an n bit odd number is prime with probability approximately $1/n$ and no known primality test runs in $\mathrm{polylog}(n)$. Instead, by using the Miller–Rabin test [22] with $O(n)$ random bases on uniform odd n bit integers, we may generate a probable n bit prime in expected time $O(n^3 \cdot m(n))$ [13, Thm 12.2.2]. As `Verify` requires the multiplication of a matrix and a vector, it costs $O(n^2 \cdot m(n))$.

Theorem 1. *Let $t_c(x) = 2^{0.292x+o(x)}$ and $t_q(x) = 2^{0.265x+o(x)}$, and $\delta(n)$ be a negligible function of n, then, under current SVP solving techniques, there exists an $x(n) \in n - \Theta(n/\log n)$ such that* `LPoW` *is a $(t_c(x(n)), \delta(n))$ PoW against classical computers, and a $(t_q(x(n)), \delta(n))$ PoW against quantum computers.*

Proof. We may generate a probable $10n$ bit prime in expected time $O(n^3 \cdot m(n))$, and $n-1$ samples from $\mathcal{U}(\{0\} \cup [p-1])$ in time $O(n \log n)$, and hence a challenge c. The most efficient known algorithms `Solve` on input a challenge c call at least one, and at most $\mathrm{poly}(n)$, SVP oracles in dimension in $x(n) \in n - \Theta(n/\log n)$ [10,2]. Therefore in the classical case $t(n) = t_c(x(n))$, and in the quantum case $t(n) = t_q(x(n))$, using the most efficient known classical and quantum SVP oracles. Note that $n - cn/\log n \in \Theta(n)$ for any constant c, and while we do not prove that the SVP oracle must be called in dimension $x(n) \in \Theta(n)$, any $x(n) \in o(n)$ would imply a subexponential time algorithm for our α-HSVP problem, and therefore for α^2-approx-SVP [20]. As $\alpha^2 \in O(n)$, this would be a major breakthrough. Verifying a solution to a challenge can be performed in time $O(n^2 \cdot m(n))$. This concludes the discussion on efficiency. We expect 1.05^n solutions for a challenge, and therefore the PoW is complete with high probability. To make it perfectly complete one may take instead $\alpha = \sqrt{n}$ and set n larger as appropriate. Finally, it is not known how to use information from independent random lattices as advice for Hermite-SVP problems in other random lattices. Given $l(n) \in \mathrm{poly}(n)$ lattices generated by $\texttt{Gen}(1^n)$ the probability, under the Gaussian heuristic, that any of the them share a sufficiently short vector is in $\mathrm{poly}(n) \cdot 1.05^n/p \in \mathrm{negl}(n)$. Without knowing how to otherwise use advice from other lattices, we therefore have $\delta(n) \in \mathrm{negl}(n)$.

4 Discussion

We calculate a value of n that we expect to very roughly match the current cost of mining a bitcoin, 19.31 terahashes.[8] Assuming SHA-256, on input 64 bytes, takes approximately 1500 cycles, this gives approximately 2^{55} cycles. The top few data points of [2, Table 2], which uses identically generated random lattices, have dimensions $151, 153, 155$ and approximate cycle counts $2^{56}, 2^{57}, 2^{57}$ respectively. Therefore we suggest $n \approx 150$, at least given current methods.

We list here topics that could benefit from the attention `LPoW` may bring to Hermite-SVP. As mentioned in Section 3, heuristic techniques for solving SVP, e.g. the amount of attainable "dimensions for free" $\Theta(n/\log n)$, depend on the quality of the lattice basis. Clearly, the hidden constant is important. In [10,2]

[8] https://btc.com/stats/diff, retrieved 23/09/2020.

some analyses of attainable dimensions for free are given. However, given the public availability of G6K,[9] a more thorough survey of how the variants of BKZ, the insertion scoring functions, and the sequences of instructions e.g. `Pump` and `WorkOut`, described therein, affect these dimension saving techniques is possible. A downside of sieving is the exponential memory cost, which may lead to memory access delays that become a bottleneck. It has been suggested that this could be partially mitigated by hardware implementations of sieves [17,11]. Given the enormous resources put into developing ASICs for hash based PoW, one may expect similar advances to be feasible in the case of `LPoW`, as well as advances beyond the parallelism offered by G6K [2, App B]. In particular, one may hope for advances upon previous work on distributed sieving [7] to larger or more general contexts.

Finally, recent works on concrete quantum circuits and the application of error correction estimate the speedups attainable in practice from quantum search when used in the context of hash functions [4] and lattice sieves [3]. While the cited works suggest that, under our current understanding of quantum computers, little to no advantage would be gained from the use of a quantum computer when solving PoW today, we are considering the case where e.g. improvements in classical computational power push the required hardness of PoW into ranges where a quantum computer would provide a meaningful advantage, or where more efficient error correction is available. At worst, we have specified a new PoW based on well studied hard problems. This work ultimately derives from our desire to create a PoW that future proofs blockchains against giving a large advantage to quantum computers.

References

1. Aggarwal, D., Brennen, G.K., Lee, T., Santha, M., Tomamichel, M.: Quantum attacks on bitcoin, and how to protect against them. arXiv preprint arXiv:1710.10377 (2017)
2. Albrecht, M.R., Ducas, L., Herold, G., Kirshanova, E., Postlethwaite, E.W., Stevens, M.: The general sieve kernel and new records in lattice reduction. In: Ishai, Y., Rijmen, V. (eds.) Advances in Cryptology – EUROCRYPT 2019. pp. 717–746. Springer International Publishing, Cham (2019)
3. Albrecht, M.R., Gheorghiu, V., Postlethwaite, E.W., Schanck, J.M.: Estimating quantum speedups for lattice sieves. Cryptology ePrint Archive, Report 2019/1161 (2019)
4. Amy, M., Di Matteo, O., Gheorghiu, V., Mosca, M., Parent, A., Schanck, J.: Estimating the cost of generic quantum pre-image attacks on sha-2 and sha-3. In: Avanzi, R., Heys, H. (eds.) Selected Areas in Cryptography – SAC 2016. pp. 317–337. Springer International Publishing, Cham (2017)
5. Ball, M., Rosen, A., Sabin, M., Vasudevan, P.N.: Proofs of work from worst-case assumptions. In: Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, 2018, Proceedings, Part I. pp. 789–819 (2018)

[9] https://github.com/fplll/g6k.

6. Becker, A., Ducas, L., Gama, N., Laarhoven, T.: New directions in nearest neighbor searching with applications to lattice sieving. In: Proceedings of the Twenty-seventh Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 10–24. SODA '16, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (2016)
7. Bos, J.W., Naehrig, M., van de Pol, J.: Sieving for shortest vectors in ideal lattices: a practical perspective. Cryptology ePrint Archive, Report 2014/880 (2014)
8. Chen, Y.: RÃľduction de rÃľseau et sÃľcuritÃľ concrÃĺte du chiffrement complÃĺtement homomorphe. Ph.D. thesis, Université Paris Diderot (2013), http://www.theses.fr/2013PA077242
9. Chen, Y., Nguyen, P.Q.: Bkz 2.0: Better lattice security estimates. In: Lee, D.H., Wang, X. (eds.) Advances in Cryptology – ASIACRYPT 2011. pp. 1–20. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
10. Ducas, L.: Shortest vector from lattice sieving: A few dimensions for free. In: Nielsen, J.B., Rijmen, V. (eds.) Advances in Cryptology – EUROCRYPT 2018. pp. 125–145. Springer International Publishing, Cham (2018)
11. Ducas, L.: Shortest Vector from Lattice Sieving: a Few Dimensions for Free (talk). https://eurocrypt.iacr.org/2018/Slides/Monday/TrackB/01-01.pdf (Apr 2018)
12. Dworkin, M.J.: FIPS PUB 202: SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. Federal Inf. Process. Stds. (NIST FIPS) (2015), https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf
13. Galbraith, S.D.: Mathematics of Public Key Cryptography. Cambridge University Press (March 2012)
14. Goldstein, D., Mayer, A.: On the equidistribution of hecke points. Forum Mathematicum 15(2), 165–189 (13 Jan 2003)
15. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing. p. 212âĂŞ219. STOC '96, Association for Computing Machinery, New York, NY, USA (1996)
16. Hastings, M., Heninger, N., Wustrow, E.: Short paper: The proof is in the pudding. In: Goldberg, I., Moore, T. (eds.) Financial Cryptography and Data Security. pp. 396–404. Springer International Publishing (2019)
17. Kirchner, P.: Re: Sieving vs. enumeration. https://groups.google.com/forum/#!msg/cryptanalytic-algorithms/BoSRL0uHIjM/wAkZQlwRAgAJ (May 2016)
18. Laarhoven, T.: Search problems in cryptography. Ph.D. thesis, Eindhoven University of Technology (2015)
19. Laarhoven, T., Mariano, A.: Progressive lattice sieving. In: Lange, T., Steinwandt, R. (eds.) Post-Quantum Cryptography. pp. 292–311. Springer International Publishing, Cham (2018)
20. LovÃąsz, L.: An Algorithmic Theory of Numbers, Graphs and Convexity. Society for Industrial and Applied Mathematics (1986)
21. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf (2008)
22. Rabin, M.O.: Probabilistic algorithm for testing primality. Journal of Number Theory 12(1), 128 – 138 (1980)
23. Schnorr, C., Euchner, M.: Lattice basis reduction: Improved practical algorithms and solving subset sum problems. Mathematical Programming 66, 181–199 (1994)
24. Teruya, T., Kashiwabara, K., Hanaoka, G.: Fast lattice basis reduction suitable forÂămassive parallelization and its application to the shortest vector problem. In: Abdalla, M., Dahab, R. (eds.) Public-Key Cryptography – PKC 2018. pp. 437–460. Springer International Publishing, Cham (2018)

25. Unruh, D.: Computationally binding quantum commitments. In: Fischlin, M., Coron, J.S. (eds.) Advances in Cryptology – EUROCRYPT 2016. pp. 497–527. Springer Berlin Heidelberg, Berlin, Heidelberg (2016)
26. Wood, G.: Ethereum: A secure decentralised generalised transaction ledger. Ethereum project yellow paper 151, 1–32 (2014)
27. Zalka, C.: Grover's quantum searching algorithm is optimal. Phys. Rev. A 60, 2746–2751 (Oct 1999)