Survey of Cryptographic Offloading Techniques for Blockchain Systems

Sebastian Fritsch, Richard von Seck*, Filip Rezabek*,
*Chair of Network Architectures and Services
School of Computation, Information and Technology, Technical University of Munich, Germany
Email: s.fritsch@tum.de, seck@net.in.tum.de, frezabek@net.in.tum.de

***Abstract*—The rise of blockchain systems in recent years has posed challenges to their energy efficiency and scalability. Proposed solutions to these problems have been cryptographic offloading techniques, which we want to present in this paper. Firstly, we introduce blockchain systems based on the blockchain stack and the blockchain trilemma. Secondly, we present the offloading techniques found for each layer and group them into five categories: Proof of Work offloading, Signature Verification offloading, Zero-Knowledge Proof offloading, offloading to the network and Trusted Execution offloading. It was observed that most of the actual offloading of cryptographic operations occure in the context of Proof of Work or Zero-Knowledge Proofs, and despite successful research results, the other offloading categories are often not yet applied in practice and further research is needed.**

***Index Terms*—blockchain, cryptographic offloading, hardware acceleration, FPGA**

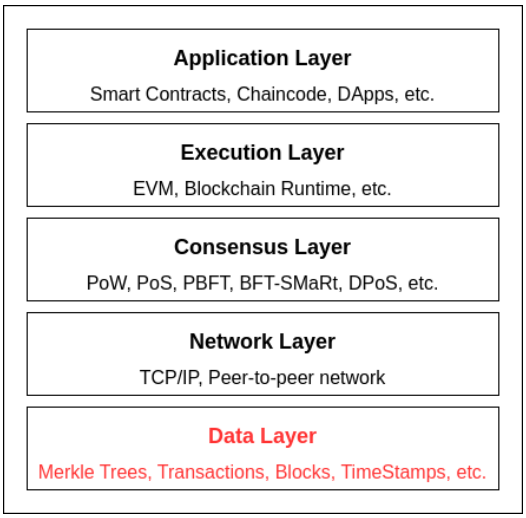

Figure 1: The blockchain stack as presented by [3] [4]

1. Introduction

Blockchain technology has gained immense popularity in recent years for its ability to provide secure and decentralized computation, transfer, and storage of data. Starting with Bitcoin [1] as a proof of concept in 2008, a huge ecosystem of different blockchain systems and technologies has emerged over the last decade.

However, one of the major bottlenecks of further adoption of state-of-the-art blockchain technology into real-world systems is low throughput performance and high latency [2]. To overcome this challenge, researchers have proposed various cryptographic offloading techniques that aim to shift some of the heavy computational tasks from blockchain nodes to other specialized hardware devices.

In this paper we conduct a survey of cryptographic offloading techniques for blockchain systems. It covers various approaches proposed to reduce the computational burden on blockchain nodes, which are categorized into five main categories: Proof of Work (PoW) offloading, Signature Verification offloading, Zero-Knowledge Proof (ZKP) offloading, and Trusted Execution offloading. The objective of this survey is to provide an overview of cryptographic offloading techniques for blockchain systems and to help researchers and practitioners understand the state-of-the-art in this area.

2. Background

Figure 1 shows the blockchain stack as laid out by Fan et al. [3] and Li et al. [4], which provides a basic introduction to the technical side of blockchain systems.

The data layer is the foundational layer of the blockchain, which defines data structures, models, block layout, and storage policies. All higher layers in the blockchain stack use and work on these data structures and in that way the data layer forms a meta-layer, which is symbolized by the red coloring. A very crucial structure are transactions, which contain information signed by a user on how to alter the distributed ledger data in accordance with the runtime rules. Those, in turn, get grouped into blocks, which then get committed together onto the ledger through a consensus mechanism. The exact layout of these data structures varies from one blockchain implementation to another, but they all share some common concepts. For example, the linkage of the different blocks (which coined the term block-chain), through the inclusion of the previous block header in the current block, is essential [4].

A blockchain consists of multiple nodes that communicate in a peer-to-peer network. If everybody can participate in the network, we speak of a public blockchain, otherwise the blockchain system is called permissioned. Similarly to the OSI/ISO model, the network layer facilitates the operation of all higher layers. The network layer is responsible for connections to peers and the propagation of blocks and transactions. The peer-to-peer layout is the basis for the decentralized structure of the blockchain, as there is no machine or network participant that can exclude a peer by itself [3].

 doi: 10.2313/NET-2023-11-1_05

The core of a blockchain is formed by the consensus layer. On this layer, blockchain nodes run an agreed-upon consensus algorithm that decides on the validity of the current state of the ledger and on which blocks to include in it. In the center of most consensus mechanism is a sybil resistance mechanism, that differentiates between trustworthy and non-trustworthy peers, like PoW in Bitcoin [1]. In PoW, miners must find a random value to include in the block so that the block hash has the required number of zero bits. As it is not computationally feasible to solve this efficiently, miners must resort to brute force and invest money in the form of energy to mine the block and "prove their work". In return, miners receive a reward for their efforts, which in most public blockchains is a token share. PoW has been criticized harshly for its vast energy consumption and high latency, so modern blockchains tend to switch to other consensus mechanisms such as Proof of Stake (PoS), Proof of Authority (PoA) or Practical Byzantine Fault Tolerance (PBFT) [2]. Those algorithms do not rely on the computational power of the nodes, but rather on the stake or their identity.

An interesting area of research on novel consensus mechanisms has been in the field of rollups [5]. Rollups attempt to solve the scalability issues of many layer-one chains such as Ethereum by introducing a second layer, that handles the transaction execution and only stores state and proofs on the layer-one chain. We distinguish between two types of rollups: Optimistic rollups and zero-knowledge rollups.

For cryptographic offloading we focus on zero-knowledge rollups, which publish a validity-ZKP to the layer-one to attest the correct execution of transactions. ZKPs are a technology that enable proving the validity of a statement without revealing information about the statement iself. They are also used for applications in Smart Contracts (SCs) such as privacy preserving transfer [5].

The application layer forms the user facing site of the blockchains. Starting with Ethereum in 2015, blockchains enable the execution of SCs and Chaincode on the execution layer [6]. The execution layer offers SCs an execution environment, in which they can interact with blockchain assets and data in ways allowed by the runtime. Examples for such SCs include simple transfers, decentralized exchanges, decentralized autonomous organizations or privacy preserving transfers through ZKPs.

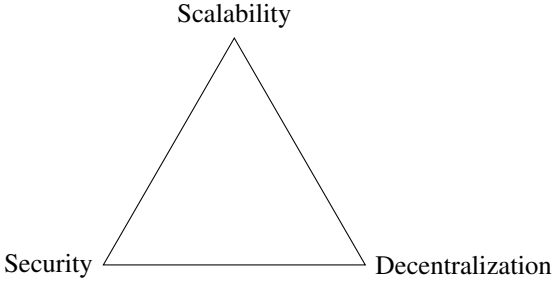

Figure 2: The blockchain trilemma [7]

Figure 2 shows the blockchain trilemma as first presented by Vitalik Buterin [7]. It states that there is an inevitable trade-off between the three properties of Scalability, Security, and Decentralization. As for many public blockchains, security and decentralization are crucial, scalability becomes a major issue. This problem led to more research into scalability solutions like sharding, rollups, or the Lightning Network [4], but also to more attention and research on offloading cryptographic operations [8], which we are going to outline in this work.

3. Overview of Categories

Nearly every layer in a blockchain system involves cryptographic operations, which – depending on their computational effort – may be suitable for cryptographic offloading. In Table 1, we have categorized our findings for cryptographic offloading in five categories and mapped them to the layers, where the offloading technique is used. In the following we are going to outline our findings in detail associated with every blockchain layer.

3.1. Data Layer

Starting with the data layer, we find data structures like Merkle Trees, which are computed using a tree-like structure of hashes. While it is theoretically possible to accelerate these hash computations, we did not find recent research on hardware or network offlading of the data layer.

3.2. Network Layer

Next up with the network layer, some blockchains (especially permissioned blockchains) like Quorum, Ripple and Corda use TLS to secure the data transfer between network participants [29] [30]. Hyperledger Fabric also offers support for TLS, but it is turned off by default [30]. Research by Kim et al. [27] has explored offloading the TLS handshake to SmartNICs, which demonstrated a 5.9x speedup compared to using a CPU. However, there has not been any evidence of adoption of such offloading techniques in blockchain systems.

3.3. Consensus Layer

The consensus layer is home to most cryptographic operations in a blockchain system and therefore designated for offloading them. For the sake of readability we are going to present each offloading category in an own subsection.

3.3.1. Proof of Work Offloading. PoW is a popular sybil-resistance algorithm and the most widespread example of cryptographic offloading to hardware in blockchain systems. The best-known example of this is Bitcoin, which incentivizes block mining through block rewards and transaction fees [9]. In order to mine more correct blocks and save energy, the first Bitcoin GPU miner was introduced in 2010, followed by subsequent developments such as FPGA (Field Programmable Gate Array) miners in 2011 and the first specialized ASIC (Application-Specific Integrated Circuit) miner in 2013 [9]. These advancements led to an increase in mining power from 33 MH/s on Intel Core i7-990x to 13.5 TH/s on an Antminer S9 ASIC. Currently, the two state-of-the-art ASIC miners are

Category	Affected Layers	References
Proof of Work Offloading	Consensus Layer	[9], [10], [11], [12]
Signature Verification Offloading	Consensus Layer, Execution Layer, Application Layer	[13], [14], [15], [16]
Zero-Knowledge Proof Offloading	Consensus Layer, Application Layer	[17], [18], [19], [20], [21], [14]
Trusted Execution Offloading	Consensus Layer, Execution Layer	[22], [23], [24], [25], [26]
Network Offloading	Consensus Layer	[15], [27], [28]

TABLE 1: Categories of Cryptographic Offloading Techniques

BitFury and Bitmain, both of which operate on a 16nm die size and have an energy consumption of about 0.07W per GH/s [9].

Korotkyi and Sachov [12] presented a design for a PoW accelerator for the IOTA cryptocurrency by implementing the Curl hash function on a FPGA, and they were able to achieve a latency speedup of 2100x compared to the software implementation.

On the other hand, efforts have been made to prevent offloading of PoW to specialized hardware by introducing ASIC-resistant algorithms such as multi-hash PoW (e.g., X11, X14), memory-hard PoW (e.g., Ethash, Scrypt, Crypt-Night), or programmatic PoW (e.g., ProgPoW) [31].

Another interesting field of blockchain application is the use in IoT networks. As most IoT nodes do not have access to plentiful energy or computation resources, researchers have proposed offloading the PoW to the edge. This involves taking the computational effort away from the node itself and delegating it to a trusted machine on the edge network [28].

3.3.2. Signature Verification Offloading. As modern blockchains, such as Ethereum, shift their consensus away from PoW due to energy and efficiency concerns, other cryptographic operations become increasingly important. In particular, signature verification was found to be a major performance bottleneck in PoS/PoA chains. In 2020, Toyoda et al. showed that about 5.91 seconds out of 30 seconds blockchain runtime were spent on `crypto.Ecrecover` in a PoA Ethereum blockchain [32]. The function `crypto.Ecrecover` recovers the public key of a given ECDSA signature.

In a work published by Javaid et al. [15], they measured that about 40% of the total execution time on their Hyperledger Fabric blockchain is spent on `ecdsa_verify` and 10% on the SHA256 hashing algorithm. The signature verification is needed on the consensus layer, but also partly on the application layer, as SCs and decentralized Apps (dApps) also often perform cryptographic operations. Therefore, this subsection can also be applied to Section 3.4. These examples signify the optimization speedup in consensus gained by offloading asymmetric crypto operations.

Ikeda [16] published work on offloading ECDSA verification to a FPGA that is twice as fast as a 64-thread AMD EPYC 7601 but uses about 33 times less energy. There is a tradeoff between energy consumption and performance of the implementation and most research we found is focused on energy efficiency. While this is a valid concern, the performance aspect is more important in the context of blockchain systems. A commercial solution offered by SilexInsight and marketed as "Blockchain Hardware Accelerator" [13] focuses on performance and claims 500,000 signature verifications on the `secp256k1` curve per second, which would correspond to a verification time of 2 microseconds. It can be used on FPGAs but is also sold as a proprietary core for ASICs. However, there is no public record of blockchains using the Silex Insight accelerator, and it is unclear whether the 500,000 signatures are measured on a FPGA or an ASIC [13].

Devlin [14] published work on accelerating the Zcash blockchain, focusing not only on ZKP acceleration but also on a signature verification core. He achieved a 1.5x speedup in signature verification with one core and the ability to add more FPGA cores for parallel processing.

The most sophisticated research we found was performed by Javaid et al. [15], who implemented a custom network card on a FPGA that filters blockchain messages and verifies signatures for transactions and blocks, as well as calculates block hashes directly on the FPGA. This design is similar to a SmartNIC, as it can operate on the data before it reaches the CPU. To speed up signature verification itself, they used a propriertary core provided by Mercury Systems. In addition to offloading signature verification, they also accelerated computation of tx and block hashes using "3 stream-based SHA-256 hash calculators" and validation/sanitization of transactions. This enabled a 4.4x speedup in block validation compared to software implementation.

3.3.3. Trusted Execution Offloading. Another interesting area of research has been Trusted Execution Environments (TEEs). TEEs are implemented as a hardware feature by CPU manufacturers and offer a way to protect the confidentiality and integrity of computations that take place inside them. In addition, many TEE vendors offer a way for trusted communication channels to retrieve computing results from the TEE [25]. As the primitives of Trusted Computing are based on cryptography, and the use of them offers ways to accelerate operations, we consider the use of TEEs as a cryptographic offloading technique.

In an overview published by Bao et al. [26], they presented various consensus algorithms that make use of the TEE implementation called "Intel Software Guard Extensions (SGX)". These include Proof-of-Useful-Work (PoUW), Proof-of-Elapsed-Time (PoET), or Secure Proof of Stake (SPoS), which is resistant against the "nothing at stake attack" and the "long-range attack".

3.3.4. Zero-Knowledge Proof Offloading. As explained in Section 2, zero-knowledge proofs are used in the consensus layer for rollup layer-two chains. There are multiple approaches and setups for zero-knowledge proofs. The most famous ones are zkSNARKs, zkSTARKs, and Bulletproofs [33]. At the current time, every known zero-knowledge setup suffers from some performance or decentralization issues, such as creation complexity, validation complexity, proof size, or the requirement for a trusted

setup [34]. This makes it difficult to achieve a performant software implementation. Therefore, multiple researchers have set out to offload the computation of verifications or proofs onto suited hardware.

Weiliang et al. [17] presented GZPK, a proof system for zkSNARKs, that offloads many of the underlying ZKP operations to a GPU. Such underlying functions include Multiscalar Multiplication (MSM) or Number-theoretic transformations (NTT). A similar approach was presented by Lu et al. [18], which focused on the MSM operation. They were able to achieve a speedup of about 2x in comparison to other state-of-the-art GPU implementations. Apart from GPU acceleration, there is research on FPGA offloading. For instance, Peng et al. [21] designed a hardware implementation of the Groth16 zk-SNARK algorithm, which managed to achieve a 10x speedup for creating a proof compared to a reference software implementation. Additionally, some commercial companies are researching on and developing hardware implementations of zero-knowledge proofs, most notably Ingonyama[1] and Cysic[2]. There has also been the "ZPrize"[3] competition in 2022, which focused on accelerating the underlying functions. For example, Ray et al. [20] accelerated the MSM by 15% compared to a FPGA implementation of ZCash, which we reference in Section 3.4. Another prize category was the acceleration of NTT on a FPGA, which was won by team "Supranational". They achieved a performance of 2.47 milliseconds for a transform of the required 2^{24} points [19].

3.4. Execution and Application Layer

The execution and application layers of blockchain systems involve multiple cryptographic operations. For instance, the Solana Sealevel Runtime provides syscalls for cryptographic operations like `sol_secp256k1_recover` or `sol_curve_validate_point` [35]. The Ethereum Virtual Machine (EVM) provides the `ECDSAPUBKEY`, `ECDSASIGN`, and `SNARKV` precompiled contracts [6] that could potentially be offloaded to hardware. Additionally, SCs are free to implement any cryptographic operations they require, leading to a variety of applications. One such application are the ZKPs used for privacy-preserving transfers of digital assets. Tornado Cash and Zcash are two of the most prominent examples of this [36]. Research has been conducted on offloading some of the computations involved in Zcash to FPGAs. For example, Ben Devlin [14] designed a zk-SNARK accelerator by implementing a `bls12-381` coprocessor on a FPGA. He achieved a 2.9x speedup by implementing $\mathbb{F}_p$, $\mathbb{F}_p^2$, $\mathbb{F}_p^6$, $\mathbb{F}_p^{12}$ arithmetic over the `bls12-381` curve, as well as several higher-level operations such as inversion, calculating powers, calculating Frobenius maps, Miller loops, final exponentiation, and optimal ate pairing [14].

Several researchers have proposed leveraging TEEs to execute SCs over private data and to verify the correct execution of these contracts. Bowman et al. [24] employed Intel SGX to achieve these goals for mutually untrusted parties. Meanwhile, Yuan et al. [23] presented ShadowEth, which utilized a similar approach to execute private SCs while preserving existing public blockchains. In this method, only the verification of the correct execution of SCs is performed on the public chain.

In contrast, Das et al. [22] proposed FastKitten, which utilizes TEEs to execute SCs on public blockchains that lack native support for smart contract execution but have a means of storing data. Unlike the previous approaches, FastKitten does not focus on privacy but rather on extending the functionality of blockchains.

4. Discussion

In Section 3, we explored various approaches to offloading cryptographic operations and how they can be applied to different layers of the blockchain stack. We can further categorize these methods based on their scope and impact on the blockchain system.

PoW offloading provides a significant performance improvement in terms of energy costs and speed for the node itself. Without using PoW offloading to ASICs, mining Bitcoin would no longer be profitable due to the high energy costs associated with using GPUs and CPUs [9]. This is likely the reason why PoW offloading is widely used in practice. ZKP offloading to GPUs is currently deployed in Filecoin [4] and there is a lot of research going into offloading to FPGAs and ASICs.

On the other hand, approaches like Signature Verification offloading often only provide a performance gain for the blockchain if the majority of nodes required for consensus adopts them. Otherwise, the blockchain's speed is bottlenecked by the slowest machine required to reach consensus. While there may be potential for energy savings, to our knowledge, this approach has not been widely adopted yet.

While achieving majority adoption of specialized hardware is challenging in permissionless blockchains like Ethereum, there is potential for permissioned blockchains like Hyperledger Fabric. In those settings, the nodes are known and can be required to use a certain hardware or software. This is the focus of the work done by Javaid et al. [15] on Hyperledger Fabric, although they acknowledge that further research and development is needed for the technology and especially for the database connection between specialized hardware and the CPU. This example can be mapped to the blockchain trilemma presented in Figure 2, where one trades decentralization of the blockchain against scalability and security.

Another interesting aspect is the adoption of TEEs in blockchains. Bao et al. [26] have laid out more than 18 different research proposals and projects that make use of Intel SGX. However, there has also been criticism of TEEs. For example, many of these projects are vulnerable to single-point-of-failure attacks, meaning a single compromised SGX enclave could compromise the entire network. Additionally, many solutions depend on "trusted functions", like random number generators, which are provided by the environment and must be trusted by the node operator [26].

1. https://ingonyama.com
2. https://cysic.xyz, https://hackmd.io/@Cysic
3. https://www.zprize.io
4. https://github.com/filecoin-project/rust-fil-proofs

5. Conclusion and future work

In conclusion, offloading cryptographic operations to specialized hardware or TEEs can significantly improve the performance and efficiency of blockchain systems. PoW offloading is already widely used in practice, while other offloading techniques such as signature verification require wider adoption to have a significant impact. Other interesting research has been done on SCs in the Execution and application layer, which can benefit from TEEs and ZKPs for privacy and functionality. Overall, performance and energy improvements are an important step forward but wider adoption and research are needed for some of these techniques.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008, accessed: 2023-04-01. [Online]. Available: "https://bitcoin.org/bitcoin.pdf"

[2] A. G. Gad, D. T. Mosa, L. Abualigah, and A. A. Abohany, "Emerging trends in blockchain technology and applications: A review and outlook," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 9, pp. 6719–6742, Oct. 2022. [Online]. Available: https://doi.org/10.1016/j.jksuci.2022.03.007

[3] C. Fan, S. Ghaemi, H. Khazaei, and P. Musilek, "Performance evaluation of blockchain systems: A systematic survey," *IEEE Access*, vol. 8, pp. 126 927–126 950, 2020. [Online]. Available: https://doi.org/10.1109/access.2020.3006078

[4] W. Li, M. He, and S. Haiquan, "An overview of blockchain technology: Applications, challenges and future trends," in *2021 IEEE 11th International Conference on Electronics Information and Emergency Communication (ICEIEC)2021 IEEE 11th International Conference on Electronics Information and Emergency Communication (ICEIEC)*. IEEE, Jun. 2021. [Online]. Available: https://doi.org/10.1109/iceiec51955.2021.9463842

[5] L. T. Thibault, T. Sarry, and A. S. Hafid, "Blockchain scaling using rollups: A comprehensive survey," *IEEE Access*, vol. 10, pp. 93 039–93 054, 2022. [Online]. Available: https://doi.org/10.1109/access.2022.3200051

[6] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, pp. 1–41, 2022, accessed: 2023-03-30. [Online]. Available: https://ethereum.github.io/yellowpaper/paper.pdf

[7] V. Buterin, "Why sharding is great: demystifying the technical properties," 2021, accessed: 2023-03-30. [Online]. Available: https://vitalik.ca/general/2021/04/07/sharding.html

[8] A. Hafid, A. S. Hafid, and M. Samih, "Scaling blockchains: A comprehensive survey," *IEEE Access*, vol. 8, pp. 125 244–125 262, 2020. [Online]. Available: https://doi.org/10.1109/access.2020.3007251

[9] M. B. Taylor, "The evolution of bitcoin hardware," *Computer*, vol. 50, no. 9, pp. 58–66, 2017. [Online]. Available: https://doi.org/10.1109/mc.2017.3571056

[10] N. T. Courtois, M. Grajek, and R. Naik, "Optimizing SHA256 in bitcoin mining," in *Communications in Computer and Information Science*. Springer Berlin Heidelberg, 2014, pp. 131–144. [Online]. Available: https://doi.org/10.1007/978-3-662-44893-9_12

[11] J. A. Dev, "Bitcoin mining acceleration and performance quantification," in *2014 IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE, May 2014. [Online]. Available: https://doi.org/10.1109/ccece.2014.6900989

[12] I. Korotkyi and S. Sachov, "Hardware accelerators for IOTA cryptocurrency," in *2019 IEEE 39th International Conference on Electronics and Nanotechnology (ELNANO)*. IEEE, Apr. 2019. [Online]. Available: https://doi.org/10.1109/elnano.2019.8783449

[13] "Silex insight blockchain hardware accelerator product sheet," accessed: 2023-03-30. [Online]. Available: https://www.silexinsight.com/content/uploads/BA452-Blockchain-Hardware-Accelerator_Web.pdf,https://www.silexinsight.com/blockchain-hardware-accelerator/

[14] B. Devlin, "Zcash fpga acceleration engine," 2019, accessed: 2023-03-30. [Online]. Available: https://github.com/ZcashFoundation/zcash-fpga/blob/master/zcash_fpga_design_doc_v1.4.2.pdf

[15] H. Javaid, J. Yang, N. Santoso, M. Upadhyay, S. Mohan, C. Hu, and G. Brebner, "Blockchain machine: A network-attached hardware accelerator for hyperledger fabric," in *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*. IEEE, Jul. 2022. [Online]. Available: https://doi.org/10.1109/icdcs54860.2022.00033

[16] M. Ikeda, "Hardware acceleration of elliptic-curve based crypto-algorithm, ECDSA and pairing engines," in *2021 IEEE 14th International Conference on ASIC (ASICON)*. IEEE, Oct. 2021. [Online]. Available: https://doi.org/10.1109/asicon52560.2021.9620402

[17] W. Ma, Q. Xiong, X. Shi, X. Ma, H. Jin, H. Kuang, M. Gao, Y. Zhang, H. Shen, and W. Hu, "GZKP: A GPU accelerated zero-knowledge proof system," in *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*. ACM, Jan. 2023. [Online]. Available: https://doi.org/10.1145/3575693.3575711

[18] T. Lu and L. Peng, "BPU: A blockchain processing unit for accelerated smart contract execution," in *2020 57th ACM/IEEE Design Automation Conference (DAC)*. IEEE, Jul. 2020. [Online]. Available: https://doi.org/10.1109/dac18072.2020.9218512

[19] "Accelerating ntt operations on an fpga," 2022, accessed: 2023-03-30. [Online]. Available: https://github.com/z-prize/2022-entries/tree/main/open-division/prize2-ntt

[20] F. Y. Q. Andy Ray, Ben Devlin and R. Yesantharao, "Hardcaml zprize competition," 2022, accessed: 2023-03-30. [Online]. Available: https://zprize.hardcaml.com

[21] B. O. Peng, Y. Zhu, N. Jing, X. Zheng, and Y. Zhou, "Design of a hardware accelerator for zero-knowledge proof in blockchains," in *Lecture Notes in Computer Science*. Springer International Publishing, 2021, pp. 136–145. [Online]. Available: https://doi.org/10.1007/978-3-030-74717-6_15

[22] P. Das, L. Eckey, T. Frassetto, D. Gens, K. Hostáková, P. Jauernig, S. Faust, and A.-R. Sadeghi, "Fastkitten: Practical smart contracts on bitcoin," USA, p. 801–818, 2019.

[23] R. Yuan, Y.-B. Xia, H.-B. Chen, B.-Y. Zang, and J. Xie, "ShadowEth: Private smart contract on public blockchain," *Journal of Computer Science and Technology*, vol. 33, no. 3, pp. 542–556, May 2018. [Online]. Available: https://doi.org/10.1007/s11390-018-1839-y

[24] M. Bowman, A. Miele, M. Steiner, and B. Vavala, "Private data objects: an overview," 2018. [Online]. Available: https://arxiv.org/abs/1807.05686

[25] K. Rabimba, L. Xu, L. Chen, F. Zhang, Z. Gao, and W. Shi, "Lessons learned from blockchain applications of trusted execution environments and implications for future research," in *Workshop on Hardware and Architectural Support for Security and Privacy*. ACM, Oct. 2021. [Online]. Available: https://doi.org/10.1145/3505253.3505259

[26] Z. Bao, Q. Wang, W. Shi, L. Wang, H. Lei, and B. Chen, "When blockchain meets SGX: An overview, challenges, and open issues," *IEEE Access*, vol. 8, pp. 170 404–170 420, 2020. [Online]. Available: https://doi.org/10.1109/access.2020.3024254

[27] D. Kim, S. Lee, and K. Park, "A case for SmartNIC-accelerated private communication," in *4th Asia-Pacific Workshop on Networking*. ACM, Aug. 2020. [Online]. Available: https://doi.org/10.1145/3411029.3411034

[28] S. Wadhwa, S. Rani, Kavita, S. Verma, J. Shafi, and M. Wozniak, "Energy efficient consensus approach of blockchain for IoT networks with edge computing," *Sensors*, vol. 22, no. 10, p. 3733, May 2022. [Online]. Available: https://doi.org/10.3390/s22103733

[29] M. Benji and M. Sindhu, "A study on the corda and ripple blockchain platforms," in *Advances in Intelligent Systems and Computing*. Springer Singapore, Dec. 2018, pp. 179–187. [Online]. Available: https://doi.org/10.1007/978-981-13-1882-5_16

[30] N. Storublevtcev, "Cryptography in blockchain," in *Computational Science and Its Applications – ICCSA 2019*. Springer International Publishing, 2019, pp. 495–508. [Online]. Available: https://doi.org/10.1007/978-3-030-24296-1_39

[31] H. Cho, "ASIC-resistance of multi-hash proof-of-work mechanisms for blockchain consensus protocols," *IEEE Access*, vol. 6, pp. 66 210–66 222, 2018. [Online]. Available: https://doi.org/10.1109%2Faccess.2018.2878895

[32] K. Toyoda, K. Machi, Y. Ohtake, and A. N. Zhang, "Function-level bottleneck analysis of private proof-of-authority ethereum blockchain," *IEEE Access*, vol. 8, pp. 141 611–141 621, 2020. [Online]. Available: https://doi.org/10.1109/access.2020.3011876

[33] X. Sun, F. R. Yu, P. Zhang, Z. Sun, W. Xie, and X. Peng, "A survey on zero-knowledge proof in blockchain," *IEEE Network*, vol. 35, no. 4, pp. 198–205, Jul. 2021. [Online]. Available: https://doi.org/10.1109/mnet.011.2000473

[34] Y. Gong, Y. Jin, Y. Li, Z. Liu, and Z. Zhu, "Analysis and comparison of the main zero-knowledge proof scheme," in *2022 International Conference on Big Data, Information and Computer Network (BDICN)*. IEEE, Jan. 2022. [Online]. Available: https://doi.org/10.1109/bdicn55575.2022.00074

[35] R. Patel, "Sealevel syscalls," 2022, accessed: 2023-03-30. [Online]. Available: https://bpf.wtf/sol-0x04-syscalls/

[36] T. Chen, A. Lu, J. Kunpittaya, and A. Luo, "A review of zero knowledge proofs," 2021, accessed: 2023-03-30. [Online]. Available: https://timroughgarden.github.io/fob21/reports/r4.pdf