Post-Quantum and Blockchain-Based Attestation for Trusted FPGAs in B5G Networks

Ilias Papalamprou*, Nikolaos Fotos†, Nikolaos Chatzivasileiadis†, Anna Angelogianni†,
Dimosthenis Masouros*, Dimitrios Soudris*
*National Technical University of Athens, Greece
†Ubitech Ltd., Digital Security & Trusted Computing Group, Greece
*{ipapalambrou, dmasouros, dsoudris}@microlab.ntua.gr, †{nfotos, nchatzivasileiadis, angelogianni}@ubitech.eu

***Abstract*—The advent of 5G and beyond has brought increased performance networks, facilitating the deployment of services closer to the user. To meet performance requirements such services require specialized hardware, such as Field Programmable Gate Arrays (FPGAs). However, FPGAs are often deployed in unprotected environments, leaving the user's applications vulnerable to multiple attacks. With the rise of quantum computing, which threatens the integrity of widely-used cryptographic algorithms, the need for a robust security infrastructure is even more crucial. In this paper we introduce a hybrid hardware-software solution utilizing remote attestation to securely configure FPGAs, while integrating Post-Quantum Cryptographic (PQC) algorithms for enhanced security. Additionally, to enable trustworthiness across the whole edge computing continuum, our solution integrates a blockchain infrastructure, ensuring the secure storage of any security evidence. We evaluate the proposed secure configuration process under different PQC algorithms in two FPGA families, showcasing only 2% overheard compared to the non PQC approach.**

***Index Terms*—FPGA, Secure Configuration, Post-Quantum Cryptography, Remote Attestation, Blockchain**

I. Introduction

The rise of 5G and beyond 5G (B5G) networks promises unprecedented improvements in network performance, offering ultra-low latency [1], higher bandwidth [2], and support for real-time applications such as autonomous vehicles, augmented reality (AR), and others [3]. However, traditional network infrastructures, which rely on specialized hardware optimized for specific tasks (e.g., data packet processing and routing), struggle to meet these demands due to their lack of flexibility, scalability, and manual management complexity [4]. As a result, network infrastructures are transitioning from specific-purpose hardware to general-purpose processing nodes that leverage virtualization technologies and software-defined networking (SDN) [5]. This new infrastructure not only hosts network-related services but also supports user-developed applications to be deployed closer to the *network edge*, thus, enabling real-time processing required to meet the performance requirements of modern applications [6].

In this landscape, hardware accelerators such as GPUs [7] and FPGAs [8], DPUs [9] and others are being adopted to ensure that performance and real-time requirements are met for latency-sensitive applications deployed at the network edge. Among these, FPGAs are particularly notable for their reconfigurability and energy efficiency, making them ideal for dynamic edge environments where workloads can vary significantly. However, despite their performance advantages, the incorporation of accelerators along with the distributed nature of B5G architectures introduces a range of security challenges that must be addressed to safeguard their deployment.

One of the main security challenges lies in the vulnerability of FPGAs, which are often deployed in environments without adequate protection, leaving them exposed to threats such as malware injections [10], hardware trojans [11] and others. Even existing security features, such as AMD's bitstream encryption have been found to be unreliable [12]. The challenge of securing FPGAs becomes even more critical with the advent of *quantum computing*, which introduces new security threats. If quantum computers achieve their anticipated performance, they could potentially break widely used cryptographic algorithms such as the Rivest-Shamir-Adleman (RSA) system and Elliptic Curve Cryptography (ECC) schemes [13], both of which underpin many secure configuration mechanisms today.

Beyond hardware vulnerabilities, the distributed and multi-entity nature of 5G/B5G networks introduces additional security challenges related to trust and authentication. In these networks, multiple parties – such as *infrastructure providers*, *application developers*, and *network operators* – are involved in the operation of edge computing environments. Each entity plays a distinct role, creating a complex ecosystem where verifying the integrity and authenticity of every participant is crucial to ensuring overall network security. To address these challenges, robust attestation is needed to verify the actions and trustworthiness of each party, along with immutable storage systems to securely gather evidence.

Although previous research has developed various methods to enhance secure configuration of FPGAs, these approaches do not offer countermeasures against quantum attacks or provide a reliable storage system for collected security evidence. For instance, researchers have concentrated on FPGA Trusted Execution Environments (TEEs) [14]–[16] to ensure secure configuration, with some introducing additional functionalities such as a netlist scanner [17], for detecting malicious modules. However, all of these solutions rely on ECC-based algorithms, leaving them vulnerable to quantum attacks. Additionally, all of the previously mentioned solutions overlook the collection and management of attestation evidence, thereby lacking a trustworthy data consistency mechanism. While some recent

efforts have begun integrating such mechanisms into generic edge environments [18], they have yet to focus on FPGAs.

In this paper, we propose a remote attestation protocol that integrates Post-Quantum Cryptography (PQC) with blockchain technology to establish a quantum-resistant, trusted attestation framework for FPGAs in B5G networks. Our approach addresses both the quantum security vulnerabilities of traditional cryptographic methods and the need for reliable, immutable storage of security evidence in multi-entity environments. The key contributions of this work are: *i)* We introduce a hybrid hardware and software solution based on remote attestation with PQC algorithms, ensuring that the attestation process remains secure against future quantum threats; *ii)* We integrate a blockchain infrastructure, for collecting the security evidence from the deployed FPGA-based edge nodes, providing a decentralized, tamper-resistant ledger for storing attestation records; and *iii)* We evaluate the performance in terms of execution time of our system, based on different PQC algorithms on two FPGA families. Our results shows that for a particular selection of PQC algorithms, minimal overhead is added ($\sim 2\%$) compared to the regular non-PQC approach.

II. Proposed Remote Attestation Scheme

Our solution leverages a custom *remote attestation* scheme, enhanced with PQC to ensure quantum-resilient integrity verification and Blockchain technology to provide an immutable and transparent record of attestation evidence. Remote attestation enables a system and its components to prove their trustworthiness to a remote *verifier*. After an offline preparation, the process starts with the *verifier* sending a challenge to the system, which then sends evidence back to the *verifier* for validation [19]. Afterwards, all attestation evidence is stored in a secure storage mechanism i.e., the Blockchain.

A. System's Architecture

Figure 1 shows the architecture of our system, that consists of three entities: i) the application provider, ii), the infrastructure provider, and iii), the service provider.

1) Application Provider: The application provider refers to the entity offering the application (bitstream) to be deployed on the FPGA. This could be either an application related to network functionality (e.g., packet processing) or an end-user application requiring real-time processing (e.g., autonomous driving). The objective is to ensure that the bitstream executing on the FPGA is an unaltered and authenticated version of the original bitstream provided by the application provider, with no modifications introduced by unauthorized entities.

2) Infrastructure Provider: The infrastructure provider is responsible for delivering and maintaining the underlying hardware and software resources necessary to support the edge computing environments. Unlike other entities, it focuses specifically on the physical and virtual infrastructure to ensure efficient, secure and reliable service operations.

Edge Node: The edge computing node equipped with the FPGA, with its architecture shown in Fig. 1. It contains hardware (i.e., AES Kernel) and software components (i.e., attestation service) supplied by the infrastructure provider. The *attestation service* is the core of the edge node and is responsible for managing all tasks for the secure programming of the FPGA; including connection with the attestation server, gathering the security evidence for the remote attestation (Digital Signature Algorithm (DSA), hash functions, communication with AES Kernel), as well as loading the target application onto the FPGA (Key Encapsulation Mechanism (KEM), bitstream decryption). In contrast to prior works, we focus on enhancing the system's security by utilizing PQC algorithms. Specifically, for the KEM and the DSA we avoid regular ECC-based approaches. Their security relies on the difficulty of solving particular mathematical problems (e.g., Discrete Logarithm Problem), which can be solved by quantum computers. In contrast, PQC algorithms rely on different mathematical properties (e.g., lattices), which remain resistant even to quantum attacks [20]. Furthermore, given the diversity of devices in edge computing environments, our solution is versatile, targeting two distinct FPGAs: (i) PCI-Express (PCI-E) FPGAs with an x86 CPU as the Processing System (PS), which are better suited for larger-scale edge nodes, (ii) System-on-Chip (SoC) FPGAs, which integrate both the Programmable Logic (PL) and the PS within a single system. We note that for the PS/PL communication the AXI-4 protocol is used. These are ideal for far-edge environments with tighter energy constraints.

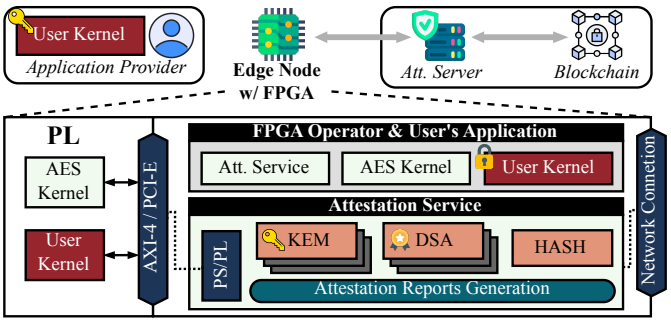


Fig. 1: Overview of our proposed approach for secure configuration of edge nodes with FPGAs.

3) Service Provider: This entity is responsible for managing the security and validation services within 5G/B5G network infrastructure, particularly in the role of the *verifier* in the attestation process. In this context, Mobile Network Operators (MNOs) commonly fulfill this role.

Attestation Server: An external server acting as the Verifier in the remote attestation. Is responsible for validating the values received by the edge node, with pre-stored reference values, that have been acquired from the application and infrastructure provider. Lastly, it is connected with the blockchain.

Blockchain: Consitues the decentralized storage for storing any evidence collected from the attestation requests. The blockchain architecture is based on Hyperledger BESU [21], enabling the design of an Ethereum-based permissioned ledger.

4) Trust model: We assume the presence of an secure attestation server, that manages the secure storage and acquisition of reference values. We do not blindly trust the edge node

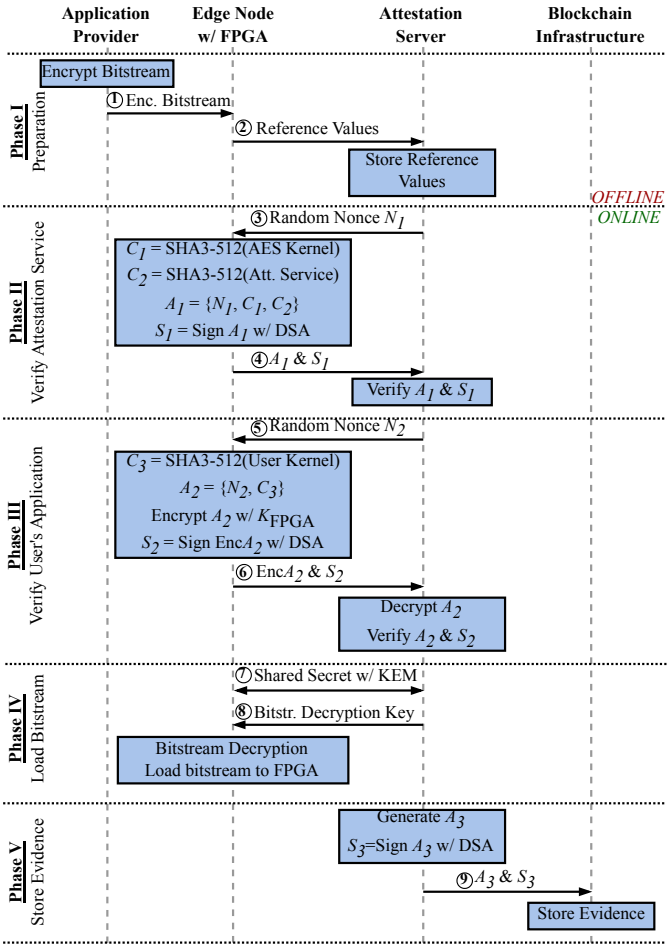


Fig. 2: Proposed remote attestation protocol.

provider; we verify the infrastructure prior to each user's application deployment. Lastly, although we acknowledge the risk of physical attacks within the FPGA (e.g., side-channel attacks), they are beyond the scope of this work.

B. Remote Attestation Protocol

Fig. 2 shows in detail each step of the proposed remote attestation protocol. The entire process for securely deploying applications in the edge can be broken down into five phases:

Phase I: Initially, an offline preparation is required: In step ① the application provider using K_{btstr} encrypts the application's bitstream and transfers it to the edge node. In step ② the attestation server acquires from the infrastructure provider all the reference values required for attestation. Additionally, the infrastructure provider is responsible for generating key pair $\{K_{s,pub}, K_{s,priv}\}$ for the DSA, as well as sharing the public key $K_{s,pub}$ with the attestation server. Lastly, the attestation server collects from the infrastructure provider K_{FPGA} key for the corresponding edge node.

Phase II: To deploy an application to the edge, the application provider sends a request to the infrastructure provider for initiating the verification process. First, we ensure the integrity of the attestation service of the edge node. In step ③ the attestation server generates a random nonce N_1 and sends it to the edge node. There, the attestation report A_1 is produced, containing two SHA3-512 checksum values: for the software (C_1) and for the hardware components (C_2) of the attestation service. The received nonce N_1 is appended to A_1, producing $A_1 = \{N_1 \| C_1 \| C_2\}$ and signed using $K_{s,priv}$, generating S_1. In step ④ A_1 and S_1 are transferred to the attestation server, which verifies each received data. Upon successful validation, all the components of the attestation service are verified and the remote attestation continues.

Phase III: Following a similar process as in phase II, we proceed with verifying the application's provider bitstream. In step ⑤, the attestation server generates a fresh nonce N_2 and sends it to the edge node, where a new attestation report A_2 is generated. It contains the received nonce N_2 and the user's encrypted bitstream SHA3-512 checksum C_3. Afterwards, the attestation report is encrypted with AES256-CBC in the FPGA, using a pre-installed key K_{FPGA} by the provider, eventually producing $EncA_2 = Enc_{K_{FPGA}}\{N_i \| C_3\}$. It is then signed with $K_{s,priv}$, producing S_2 and in step ⑥, $EncA_2$ and S_2 are transferred to the attestation server. Upon receiving all data, the attestation server decrypts $EncA_2$ using the pre-stored K_{FPGA} associated with the respective edge node, to obtain the encrypted bitstream's checksum. If the verification of both the checksum and the received signature is successful, the protocol proceeds with loading the user's bitstream onto the FPGA. Note that for higher security assurance, a Physical Unclonable Function (PUF) can be integrated in the FPGA (e.g., from [22]), for securely generating keys.

Phase IV: With successful attestation of all individual components, the attestation server sends to the edge node the bitstream decryption key K_{btstr}. For securely transferring K_{btstr}, a Key Encapsulation Method (KEM) is used for generating the shared secret K_{ss} that acts as the key for the AES256-CBC encryption/decryption of K_{btstr}. The edge node consequently receives the $Enc_{K_{ss}}(K_{btstr})$, decrypts the user's bitstream with K_{btstr} and programs the FPGA.

Phase V: After each remote attestation is completed, whether successful or not, the attestation server forwards the results to the Blockchain. Specifically, the attestation server generates a new report A_3, that includes a fresh nonce N_3, as well as the attestation results for both the attestation service (phase II) and the application's provider bitstream (phase III). The report is signed producing S_3 and then in step ⑨ both A_3 and S_3 are forwarded to the Blockchain.

III. Evaluation & Discussion

A. Experimental Setup

In Table I the specifications of our experimental setup are listed. As outlined in Section II, two distinct FPGA families are evaluated; (a) PCI-E FPGA for larger scale edge nodes, and (b) SoC FPGAs for far-edge nodes. The software components are a combination of Python 3 and C++, while the hardware modules utilize both High Level Synthesis (HLS) and Register-Transfer Level (RTL) code and are implemented using Vivado/Vitis 2021.1. As a proof of concept we utilized an AI analytics accelerator developed in HLS for the deployed

TABLE I: Edge Node & Far Edge Node Specifications

Specification	Edge Node	Far-Edge Node
FPGA	PCI-Express FPGA	ZCU104 MPSoC
PL	ALVEO U280	XCZU7EV
Utilization†	LUT=9%, FF=6% DSP< 1%, BRAM=12%	LUT=15%, FF=12% DSP=0%, BRAM=17%
PS	Intel Xeon Gold 6530 @ 2.1GHz (x86) & 256GB RAM	ARM Cortex-A53 @ 1.2GHz (aarch64) & 2GB RAM
Host OS	Ubuntu Server 22.04 (x86)	Petalinux 2021.1 (aarch64)

† Includes the AES kernel and the communication between PS/PL.

TABLE II: Cryptographic Algorithms for each Configuration

Configuration	DSA		KEM	
	Algorithm	Security†	Algorithm	Security
No-PQ	ECDSA	○	ECDH	○
PQ-I	Falcon 1024	◑	Kyber 1024	●
PQ-II	Falcon 1024	◑	McEliece-348864	◑
PQ-III	Dilithium 5	●	Kyber 1024	●
PQ-IV	Dilithium 5	●	McEliece-348864	◑

† ○ Low Security Level, ◑ Mid Security Level, ● High Security Level. The security levels against quantum attacks are based on [23], [24].

application. The external attestation server is a general purpose x86-based PC, while the Blockchain is deployed on an Ubuntu 22.04 Virtual Machine (VM).

B. Performance Evaluation

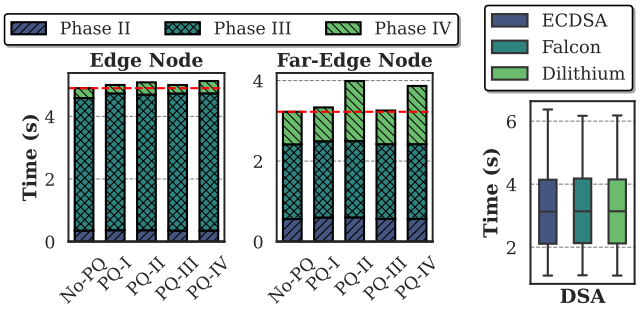
The proposed solution is evaluated over five configurations based on the cryptographic algorithms selected, as shown in Table II. We analyze the execution time of (a) the individual phases of the remote attestation (Fig. 3a) and (b) pushing data to the blockchain (Fig. 3b). As a baseline for comparing the PQC algorithms, we choose widely used ECC algorithms.

Edge Node: Both PQ-I&III configurations perform similarly with the baseline, with PQ-III showing the lowest overhead at $\sim$ 2% over the No-PQ configuration. When analyzing each attestation phase individually, phase II shows similar performance across all configurations, regardless of the DSA. Phase III accounts for the majority of the execution time, due to the lengthy process of configuring the Alveo with the AES kernel, which takes $\sim$ 3.8sec. In phase IV, where different KEMs are employed, Kyber outperforms McEliece.

Far-Edge Node: The PQ-I&III configurations have the closest performance to the baseline, with PQ-III obtaining the best results. All configurations in phases II&III have similar execution times, while in phase IV significant overhead is noticed when using the McEliece method. As with the larger edge node, Kyber emerges as the fastest PQ KEM, while for DSA, both Falcon and Dilithium deliver similar performance.

Blockchain: In the box plot shown in Fig. 3b, the time required to upload the results is nearly the same across all DSAs. Additionally, we observe some variation in each data push time, as the time needed to finalize each block's processing in the blockchain is not constant.

Overall, based on the experimental results and the security levels in Table I, we conclude that the most prominent configuration is PQ-III, that uses Dilithium as DSA and Kyber as KEM. This selection incurs approximately 2% overhead compared to the non-PQC solution in both FPGA families.



(a) Remote attestation protocol (b) Blockchain

Fig. 3: Execution time for 100 requests.

C. Security Assessment

1) PQC Algorithms Selection: We have chosen Falcon [25] and Dilithium [26] as our DSA, and Kyber [27] and McEliece [28] for our KEM. While PQC standards are still in the process of being finalized, these algorithms are considered strong candidates for the upcoming NIST standards. For the hash function (SHA3-512) and the encryption/decryption algorithm (AES-256), we have opted for classical cryptographic methods (with high bit-width) rather than PQC alternatives, as for our requirements no PQC options are available. This selection still maintains a high level of security, as based on research works [29], [30], they are effective in countering quantum attacks.

2) Third party attacker: In case an adversary attempts to load a malicious kernel to the edge node, since the reference values in the attestation server differ, the attestation fails and the malicious kernel isn't programmed to the FPGA. Furthermore, since the user's bitstream is encrypted, it prevents any reverse engineering attempts by malicious third parties. Additionally, by signing each attestation report A_i before it is shared, we mitigate the risk of man-in-the middle attacks that could alter the report's contents. Lastly, by including a random nonce in every transaction we prevent any replay attacks, in which an adversary would attempt to obtain and re-transmit previous attestation reports (targeting either the attestation server or the blockchain infrastructure).

3) Malicious FPGA Operator: In case an attacker attempts to insert malicious functionalities in the provided infrastructure that could compromise the attestation process, by verifying the attestation service deployed by the operator (phase II, steps ③-④), we guarantee the integrity of the edge nodes.

IV. Conclusions

Ensuring the secure configuration of FPGA-based edge nodes in the PQ-era poses new challenges. In this work, we presented a solution for ensuring the secure deployment of applications in FPGAs. It is built upon remote attestation and combines software and hardware modules to ensure the authenticity of each component. Furthermore, we integrate a blockchain infrastructure to our system, as a trusted storage for any evidence collected. Our evalution over two FPGA families shows that by employing PQC algorithms for the DSA and KEM steps, the execution time overhead is negligible (2%).

REFERENCES

[1] P. Popovski, Č. Stefanović, J. J. Nielsen, E. De Carvalho, M. Angjelichinoski, K. F. Trillingsgaard, and A.-S. Bana, "Wireless access in ultra-reliable low-latency communication (urllc)," *IEEE Transactions on Communications*, vol. 67, no. 8, pp. 5783–5801, 2019.

[2] S. F. Abedin, M. G. R. Alam, S. A. Kazmi, N. H. Tran, D. Niyato, and C. S. Hong, "Resource allocation for ultra-reliable and enhanced mobile broadband iot applications in fog network," *IEEE Transactions on Communications*, vol. 67, no. 1, pp. 489–502, 2018.

[3] K. Samdanis and T. Taleb, "The road beyond 5g: A vision and insight of the key technologies," *IEEE Network*, vol. 34, no. 2, pp. 135–141, 2020.

[4] T. Benson, A. Akella, and D. A. Maltz, "Unraveling the complexity of network management.," in *NSDI*, pp. 335–348, 2009.

[5] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2014.

[6] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing—a key technology towards 5g," *ETSI white paper*, vol. 11, no. 11, pp. 1–16, 2015.

[7] M. Abbasi, A. Najafi, M. Rafiee, M. R. Khosravi, V. G. Menon, and G. Muhammad, "Efficient flow processing in 5g-envisioned sdn-based internet of vehicles using gpus," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 5283–5292, 2020.

[8] V. Chamola, S. Patra, N. Kumar, and M. Guizani, "Fpga for 5g: Re-configurable hardware for next generation communication," *IEEE Wireless Communications*, vol. 27, no. 3, pp. 140–147, 2020.

[9] L. Barsellotti, F. Alhamed, J. J. V. Olmos, F. Paolucci, P. Castoldi, and F. Cugini, "Introducing data processing units (dpu) at the edge," in *2022 International Conference on Computer Communications and Networks (ICCCN)*, pp. 1–6, IEEE, 2022.

[10] T. La, K. Pham, J. Powell, and D. Koch, "Denial-of-service on fpga-based cloud infrastructures—attack and defense," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 441–464, 2021.

[11] R. S. Chakraborty, I. Saha, A. Palchaudhuri, and G. K. Naik, "Hardware trojan insertion by direct modification of fpga configuration bitstream," *IEEE Design & Test*, vol. 30, no. 2, pp. 45–54, 2013.

[12] M. Ender, A. Moradi, and C. Paar, "The unpatchable silicon: a full break of the bitstream encryption of xilinx 7-series {FPGAs}," in *29th USENIX Security Symposium (USENIX Security 20)*, pp. 1803–1819, 2020.

[13] D. J. Bernstein and T. Lange, "Post-quantum cryptography," *Nature*, vol. 549, no. 7671, pp. 188–194, 2017.

[14] M. Zhao, M. Gao, and C. Kozyrakis, "Shef: Shielded enclaves for cloud fpgas," in *Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, pp. 1070–1085, 2022.

[15] H. Oh, K. Nam, S. Jeon, Y. Cho, and Y. Paek, "Meetgo: A trusted execution environment for remote applications on fpga," *IEEE Access*, vol. 9, pp. 51313–51324, 2021.

[16] Y. Wang, X. Chang, H. Zhu, J. Wang, Y. Gong, and L. Li, "Towards secure runtime customizable trusted execution environment on fpga-soc," *IEEE Transactions on Computers*, 2024.

[17] S. Zeitouni, J. Vliegen, T. Frassetto, D. Koch, A.-R. Sadeghi, and N. Mentens, "Trusted configuration in cloud fpgas," in *2021 IEEE 29th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, pp. 233–241, IEEE, 2021.

[18] S. Zhang, Z. Yan, W. Liang, K.-C. Li, and B. Di Martino, "Bcae: A blockchain-based cross domain authentication scheme for edge computing," *IEEE Internet of Things Journal*, 2024.

[19] G. Coker, J. Guttman, P. Loscocco, A. Herzog, J. Millen, B. O'Hanlon, J. Ramsdell, A. Segall, J. Sheehy, and B. Sniffen, "Principles of remote attestation," *International Journal of Information Security*, vol. 10, pp. 63–81, 2011.

[20] L. Chen, L. Chen, S. Jordan, Y.-K. Liu, D. Moody, R. Peralta, R. A. Perlner, and D. Smith-Tone, *Report on post-quantum cryptography*, vol. 12. US Department of Commerce, National Institute of Standards and Technology . . . , 2016.

[21] "Welcome — Besu documentation."

[22] F.-J. Streit, P. Krüger, A. Becher, J. Schlumberger, S. Wildermann, and J. Teich, "Choice–a tunable puf-design for fpgas," in *2021 31st International Conference on Field-Programmable Logic and Applications (FPL)*, pp. 38–44, IEEE, 2021.

[23] H. Nejatollahi, N. Dutt, S. Ray, F. Regazzoni, I. Banerjee, and R. Cammarota, "Post-quantum lattice-based cryptography implementations: A survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 6, pp. 1–41, 2019.

[24] L. Malina, P. Dobias, J. Hajny, and K.-K. R. Choo, "On deploying quantum-resistant cybersecurity in intelligent infrastructures," in *Proceedings of the 18th International Conference on Availability, Reliability and Security*, pp. 1–10, 2023.

[25] P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Prest, T. Ricosset, G. Seiler, W. Whyte, Z. Zhang, *et al.*, "Falcon: Fast-fourier lattice-based compact signatures over ntru," *Submission to the NIST's post-quantum cryptography standardization process*, vol. 36, no. 5, pp. 1–75, 2018.

[26] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé, "Crystals-kyber: a cca-secure module-lattice-based kem," in *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 353–367, IEEE, 2018.

[27] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé, "Crystals-dilithium: A lattice-based digital signature scheme," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 238–268, 2018.

[28] R. J. McEliece, "A public-key cryptosystem based on algebraic," *Coding Thv*, vol. 4244, pp. 114–116, 1978.

[29] K. Jang, S. Lim, Y. Oh, H. Kim, A. Baksi, S. Chakraborty, and H. Seo, "Quantum implementation and analysis of sha-2 and sha-3," *Cryptology ePrint Archive*, 2024.

[30] C. A. Roma, C.-E. A. Tai, and M. A. Hasan, "Energy efficiency analysis of post-quantum cryptographic algorithms," *IEEE Access*, vol. 9, pp. 71295–71317, 2021.