2-12-2021

A Secure and Flexible FPGA–based Blockchain System for IIoTs

Han-Yee Kim

Lei Xu
The University of Texas Rio Grande Valley, lei.xu@utrgv.edu

Weidong Shi
University of Houston

Taeweon Suh

Recommended Citation

H. -Y. Kim, L. Xu, W. Shi and T. Suh, "A Secure and Flexible FPGA-Based Blockchain System for the IIoT," in Computer, vol. 54, no. 2, pp. 50-59, Feb. 2021, doi: 10.1109/MC.2020.3022066.

A Secure and Flexible FPGA–based Blockchain System for IIoTs

Han-Yee Kim
Korea University

Lei Xu
University of Texas Rio Grande Valley

Weidong Shi
University of Houston

Taeweon Suh
Korea University

***Abstract*— Blockchain is a promising solution for Industry 4.0 due to its traceability and immutability. However, blockchain itself does not guarantee the input data integrity. The tampered data from an endpoint device can be a significant problem because it may result in a cascaded negative effect on the whole smart factory operations. In this paper, we propose an FPGA-based private blockchain system for IIoTs, where the transaction generation is performed inside the FPGA in an isolated and enclaved manner. For the key confidentiality and transaction integrity, the proposed system utilizes a PUF, soft processor, and tightly coupled sensor connections inside the FPGA fabric. Since all the critical operations are hidden under the hood, adversaries even with the root privilege cannot intervene in the transaction generation process. The implemented IIoT device provides 33 transactions per minute and consumes a 191 mW of power.**

■ **INTRODUCTION** The industrial Internet of Thing (IIoT) devices are being widely deployed in many industrial sectors, especially in smart factories. For example, the Ericsson factory in Nanjing utilizes thousands of IIoT devices, and harnesses the data generated through the connected devices. It is reported that it dramatically improves efficiency by tracking actual use of tools and dispatching services and maintenance [1]. It is expected that billions of IoT devices would be connected in the near future [1]. However, as the number of IIoTs increases, the attack surface is broadened because all the entities and their interconnections are potential targets of attacks. It is reported that there are many kinds of cyberattacks on IIoTs from Supervisory Control And Data Acquisition (SCADA) to resource-constrained IoT devices [2]. Especially, the data tampering between entities may result in a cascaded negative effect on the productivity of the smart factory. To identify the node with tampered data in the event, it is imperative to utilize a secure transaction mechanism that provides traceability and immutability. In this situation, blockchain is regarded as a promising solution because it is inherently tamper-proof, traceable, and decentralized. It can offer unique advantages in situations where trust is not guaranteed among entities by utilizing digital

signature-based authentication and verification. Blockchain is categorized into two types: public and private. Public blockchain allows anyone to join the blockchain network whereas private blockchain requires permission to participate in the network. The private blockchain is more attractive to the industrial domain because only authorized nodes can join the network.

However, adversaries may jeopardize the private blockchain system by acquiring the credential of the permissioned node. They may falsify decisive transactions by intervening in the transaction generation process. To address the security concern, processor manufacturers provide a hardware-based solution for Trusted Execution Environments (TEEs), which are typically isolated and enclaved processing features. For example, ARM introduces TrustZone, and Intel provides Software Guard Extensions (SGX). However, it is reported that there are still security breaches even with TEE, such as SgxPectre [3], which utilize the micro-architectural side-channel information. Thus, in a highly automated industrial domain, a more secure black-box model is required to completely hide its internal operations and lessen the attack surface of harming the data integrity.

In this paper, we propose a field-programmable gate array (FPGA)-based blockchain system for IIoTs. The critical blockchain operations, such as the secret key management and transaction generation, are delegated to FPGA with the bitstream protection. The FPGA system is composed of a Physically Unclonable Function (PUF), a soft processor, and an internal memory. Therefore, the side-channel-attacks and reverse-engineering against the hardware system are inherently prohibited. The remainder of this paper is organized as follows: The background section introduces modern FPGA and soft processor, and blockchain. Related works section summarizes case studies integrating IoT systems with blockchain. The Proposed architecture section details the FPGA-based blockchain system for IIoTs with the threat model and security analysis. The Implementation and evaluation section elaborates PUF and FPGA system implementation and shows the experimental results. We discuss the versatility and flexibility of the proposed FPGA system in the Discussion section. We finally conclude our paper in the Conclusion section.

BACKGROUND

This section briefs the FPGA's security feature, soft processor, and blockchain.

Modern FPGA and Soft processor

The FPGA is a field-programmable device, with which a custom hardware can be dynamically configured for the data-processing acceleration. It is widely used in applications such as the digital signal processing, artificial intelligence, and big data processing. The FPGA is also used to provide security for reducing the attack surface [4]. An FPGA is configured by a synthesized hardware design file called *bitstream*. For security, it should be guaranteed that the FPGA be safely configured with bitstream without tampering. Modern FPGA vendors are providing bitstream protection mechanisms, where hard-wired cryptographic engines such as the AES is used both to safely configure the bitstream and to cope with unauthorized use of hardware IP [5]. For decrypting the encrypted bitstream on a dedicated FPGA, the AES key is pre-stored in a non-volatile memory that cannot be read back.

A soft processor is a portable and synthesizable microprocessor, which can be configured in the different kinds of FPGAs from the entry-level to the high-end. The FPGA vendors typically provide soft processors. For example, Xilinx offers MicroBlaze and Intel provides Nios. The main reason why a developer utilizes a soft processor is flexibility. The processing work can be delegated to the soft processor in place of hard-wired modules, and on-demand modification is easily achieved by the firmware update if the system requirement changes.

Blockchain

Blockchain, in principle, is a growing linked list of records, called blocks. Figure 1 shows an abstracted overview of the blockchain architecture. As shown in Figure 1, a block structure is divided into two parts: header and body. The block header contains block number, block size, and block hashes. The hash values are used for checking the integrity of the blocks. If a block is illegally modified, it is detected by comparing the successor's previous block hash. The block body contains transactions and metadata. Each transaction includes the sender and receiver's addresses, data, and signature of the transaction generator. The signature generated with cryptographic formulas is for verifying the integrity and authenticity of transactions and for

proving the ownership. In general, the blockchain is maintained together by peers with separate storages, so that it is classified as a distributed ledger technology (DLT). In the DLT-based blockchain system, any updates in the distributed ledger must be verified by the majority of the network nodes. The verification relies on consensus algorithms such as Proof of Work (PoW), Proof of Stake (PoS), or Practical Byzantine Fault Tolerance (PBFT) [6].

To join a blockchain system, each peer should register its own asymmetric key to system administrators. In general, the Public Key Infrastructure (PKI) system is utilized for the asymmetric key authentication. In the PKI system, a certificate authority guarantees that a public key is genuine via issuing a certificate. A certificate contains key-related information such as the version number, serial number, signature algorithm identification, issuer name, and timestamp. A peer node with the certificate can join the blockchain system.

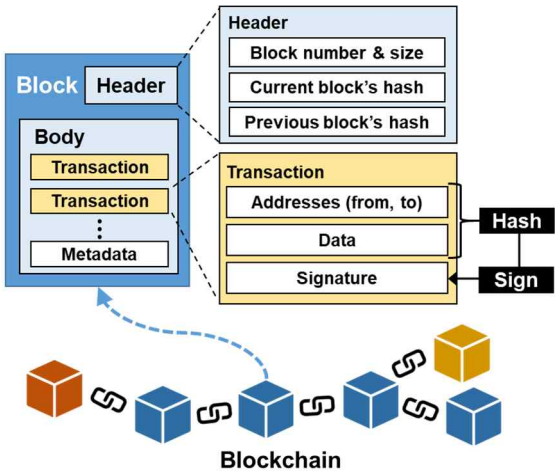

Figure 1. Blockchain architecture overview

RELATED WORKS

There are many case studies integrating IoT systems with blockchain, enhancing security and efficiency [7-12]. Huang et al. [7] proposed a credit-based consensus mechanism, which adjusts the difficulty of the PoW-based algorithm. The proposed algorithm decreases the computing burden of the honest nodes, while increasing the computing complexity against the malicious nodes. Dai et al. [8] proposed a lightweight blockchain wallet using ARM's TrustZone, which protects the payment verification process. However, the TEEs are not safe enough to assure the endpoint security due to the vulnerability to side-channel attacks, such as SgxPectre [3].

Lin et al. conceptually proposed an IoT-based blockchain system for the supply-chain traceability of food [9]. The proposed system architecture has two kinds of nodes: full-fledged nodes performing the whole blockchain functionality; IoT-based light nodes performing simple operations. Mylrea et al. proposed a blockchain system for the power grid [10]. They utilized their proprietary testbed and smart contracts for the system optimization. Mazzei et al. [11] proposed a portable and platform-agnostic blockchain solution for IIoT. They utilized an embedded system referred to as 4ZeroBox for bridging the gap between blockchain service and industrial machine. This paper is different from our work in that the focus is mainly on the system compatibility with blockchain without considering the input data protection and integrity. Florin et al. [12] proposed an FPGA based hardware system architecture for blockchain. The focus is on accelerating the blockchain jobs in IoT systems by duplicating SHA256 modules.

There are some research works addressing the security of sensors [13,14]. Taiebat et al. proposed a fault diagnosis framework for sensors [13]. The framework includes measures to enhance the fault tolerance, such as the sensor duplication and sensor network topology. Chanson et al. conceptually proposed a design methodology and requirement for a blockchain-based sensor data protection [14]. One of its goals is to create the blockchain transaction as close as possible to the sensing unit for reducing the attack vectors.

PROPOSED ARCHITECTURE

This section details the FPGA-based blockchain system for IIoTs.

Architecture overview

Figure 2. shows an overview of the proposed blockchain system architecture. Three entities are cooperating with blockchain: IIoT devices, Edge servers, and Blockchain administrators. The role of each entity is as follows:

IIoT devices: Like typical IoT gadgets, we assume that IIoT devices are light-weight and performance-restricted. Some devices are battery-operated. The role of IIoT devices is to generate (sensor) data and its blockchain transactions, and to report them to the dedicated Edge servers. Each IIoT device has attached sensors, embedded processor(s), and an FPGA. The

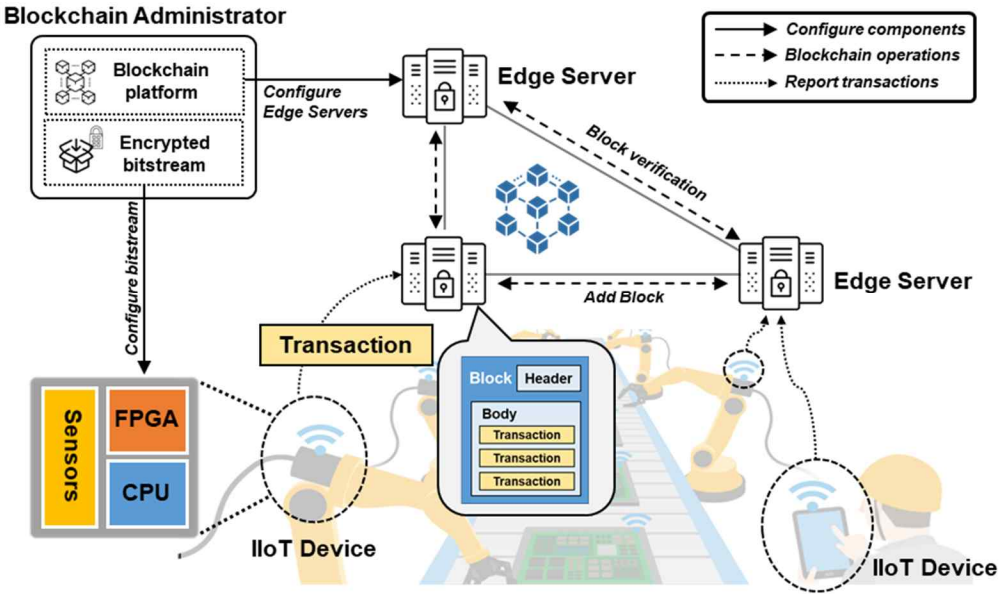

Figure 2. Overview of the proposed architecture: Three entities involved are IIoT devices, Edge servers, and Blockchain administrator.

FPGA is initially secured by the bitstream protection technology and managed by Blockchain administrator. The FPGA works as a secure black-box engine and its role is to generate blockchain transactions for the data.

Edge servers: Edge servers are high-performance computing systems or cloud elements with sufficient computing resources for Transport Layer Security (TLS), standard encryption, and recovery. The Blockchain administrator manages Edge servers, which work as full nodes executing the block operations such as block generation, verification, and consensus protocols. Especially, for the block generation, Edge servers accumulate transactions from IIoT devices and store them to the ledgers by creating blocks.

Blockchain administrator: The Blockchain administrator organizes and manages the private blockchain system. It constructs a multi-layer hierarchy from IIoT devices to Edge servers via a private blockchain platform. The administrator also generates and/or updates bitstream for FPGAs in IIoT devices. Note that, only the administrator can generate and encrypt bitstream with self-managed AES keys.

Threat model

The Blockchain administrator applies a private blockchain platform to the smart factory for traceability and immutability of the stored industrial data. We assume that Edge servers are in a secure domain because they have sufficient computing resources for security. On the other hand, individual embedded processors in endpoint IIoTs have intrinsic vulnerabilities with limited resources. There may be malicious insiders and/or outsiders in the smart factory environment. The data from IoT devices may be at the risk of being misused or tampered if adversaries intervene in the transaction generation process for the sensor data [14]. It would interrupt and/or halt the factory operations. Therefore, it is required to generate the blockchain transaction in a tightly-coupled manner from the sensor data capture to the transaction signature generation.

System architecture of IIoT device

Figure 3 shows the block diagram of the proposed IIoT device and its internal interactions. There are mainly an embedded processor, FPGA and sensors in the IoT device. Especially, sensors are tightly coupled to the FPGA via physical interfaces such as I^2C, SPI, GPIO, or CAN. The FPGA is configured with PUF, soft processor, external register, and local memory, as shown in the Figure 3. The PUF is utilized to generate a secret key. The PUF takes advantage of semiconductor process variations such as oxide thickness, metal shape, and channel length to generate a unique random value for each device with the same logical design. The soft processor works as a microcontroller for the FPGA system with local memory. Inside the local memory, the execution binary

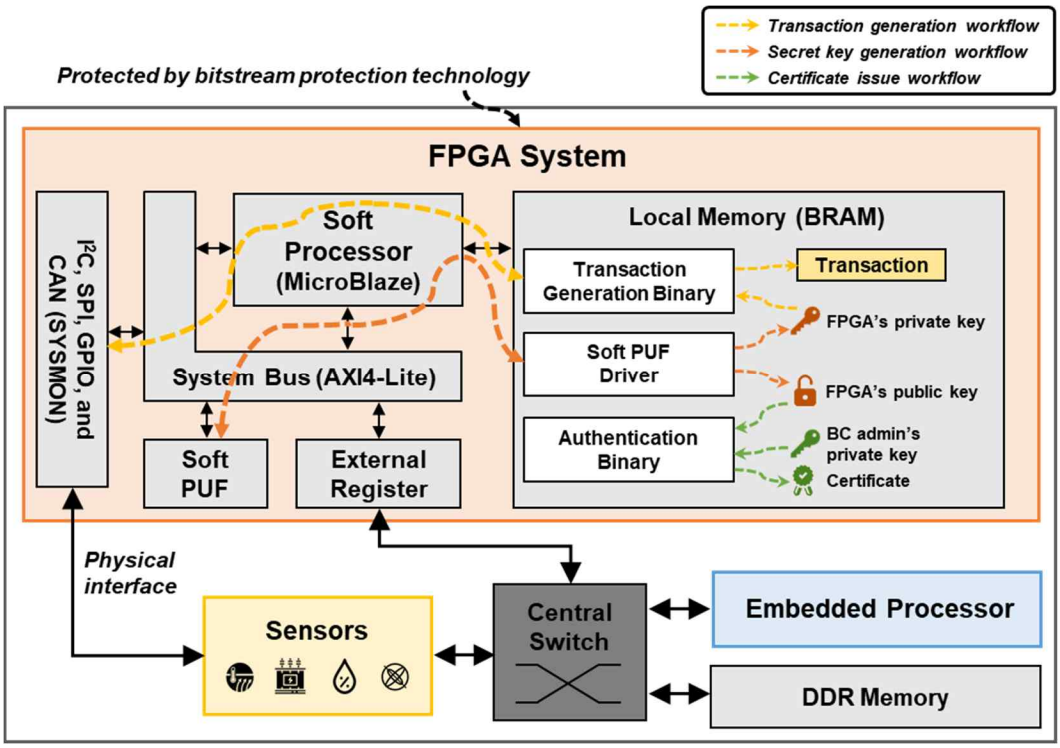

Figure 3. Detailed system architecture of IIoT device with FPGA: Sensors are directly connected to FPGA via physical interface such as I²C or SPI.

for the transaction generation, soft PUF manipulation, and key authentication is stored for the blockchain operation. The external register is a communication channel between FPGA and the embedded processor; The soft processor can write data to the external register, of which data is accessed by the embedded processor. Note that the FPGA system and its execution binary are included in a bitstream, which is safeguarded by the bitstream protection scheme during the configuration phase. The FPGA performs two fundamental operations for the data integrity in transactions: i) the key generation and management; ii) the enclaved transaction generation.

Key generation and management: The PUF inside FPGA fabric, often called soft PUF, is utilized for generating a unique private key for the blockchain operation. The PUF operates based on the *challenge* and *response* protocol, and generates a different *response* (output) with the same *challenge* (input) in each FPGA. When the bitstream is configured inside the FPGA, the soft PUF driver is automatically executed for the FPGA's private key generation. Then, the soft processor applies an arbitrary input (*challenge*) to the PUF and takes its output (*response*). The *response* is used as the FPGA's private key and its corresponding public key is calculated by the soft PUF driver. The soft PUF driver is designed to change the FPGA's private key regularly when the key lifecycle is expired. Note that a different output (*response*) can be obtained by applying a different input (*challenge*) to PUF.

After the key generation, the authentication binary takes the FPGA's public key and issues a certificate signed by the Blockchain administrator's private key. Note that, the administrator's private key is initially stored in the local memory to avoid unnecessary interactions between the administrator and FPGA. The Blockchain administrator adopts the key provisioning scheme where multiple key sets for FPGAs are utilized to minimize the impact of the key leakage. A private key is randomly selected and assigned for each FPGA in the phase of the bitstream generation. Because private keys are stored only in the FPGA's internal memory, they are not exposed to the outside of the FPGA fabric. Only the certificate is shared with the embedded processor via external registers.

Enclaved transaction generation: The enclaved transaction generation means that a blockchain transaction is directly created inside the FPGA with the sensor data encapsulated. As shown in Figure 3,

sensors are directly connected to the FPGA via physical interfaces. The sensor's raw data are converted to a digital form via analog-to-digital converter (ADC) inside the FPGA. Then, the soft processor reads the data from the ADC, and processes to generate a transaction. The signature of the transaction is computed with the FPGA's private key. The completed transaction is written to the external register. Then, the embedded processor reads the transaction and transfers it to the dedicated Edge server.

Security analysis

The proposed solution utilizes FPGA as an enclave for the critical blockchain operations. From a security perspective, the bitstream is the root-of-trust. The modern FPGAs offer the bitstream encryption scheme with AES, which can be utilized to prevent from the reverse-engineering and/or IP theft. We discuss the security concerns and countermeasures of the proposed system in terms of the key confidentiality and transaction integrity.

Key confidentiality: The FPGA's private key is generated by the PUF inside FPGA and never leaves the device. Thus, rogues even with the privileged access to the embedded processor, cannot read the key. Adversaries may conduct a brute-force attack by checking all possible private keys because they have access to the public key in the digital certificate. This kind of attack can be prevented by periodically updating the FPGA's private key. It can be achieved by designing a soft PUF driver periodically applying a new *challenge*.

Transaction integrity: The sensor data is directly gathered through physical interfaces in FPGA. The transaction construction process is hidden inside the FPGA, which cannot be intervened by attackers. An adversary may launch a Denial-of-Service (DoS) attack to paralyze the endpoint IIoT device. It can be detected by employing a strict and well-organized transaction generation policy. A simplest method is designing a periodic transaction generation policy. If the Edge server does not receive the periodic message, it is an indication of failure, malfunction, and/or compromise. Another potential attack could be the physical abuse of sensors. For the smart factory systems where the availability is the first priority, the sensor duplication with the majority voting can be used in the endpoint IIoT devices. As adopted and proved in fault-tolerant systems [13], the duplication provides high availability because it is hard to tamper multiple IIoT devices and/or sensors concurrently.

IMPLEMENTATION AND EVALUATION

For experiments, we utilized a Zynq UltraScale+ evaluation board, which has the Zynq UltraScale+ and 4GB DDR4. Zynq UltraScale+ has two sections: One section with a fused Cortex-A53 quad-core processor and the other with programmable logic (PL). A CAD tool, Vivado 2018.2, from Xilinx was used for the system development and evaluation.

Soft PUF implementation

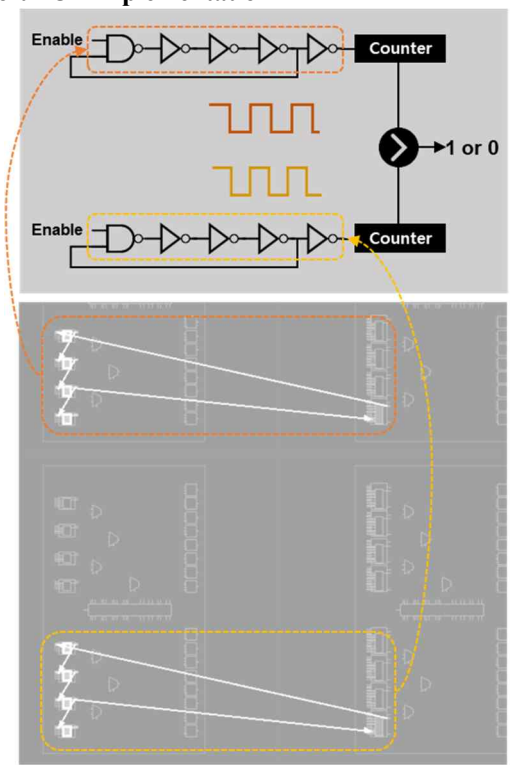

Figure 4. One-bit Ring Oscillator-based PUF on Zynq UltraScale+

We adopted Ring Oscillator (RO)-based PUF, which is composed of 2 inverter chains. Each chain has an odd number of inverters. Figure 4 shows an implementation of an 1-bit RO-based PUF in the PL section, of which purpose is to generate one random bit. As shown in Figure 4, each loop in the inverter chains generates different and unpredicted frequencies of a clock (0→1→0…), coming from each inverter's unique delay due to the manufacturing variation. Two counters were implemented to generate the final

random bit. Each counter takes the clock from its corresponding inverter chain and counts up. The counters are designed to stop when either one of them overflows. The final one random bit (either 0 or 1) is determined, depending on the counter that overflows first. There are two design considerations when implementing PUFs on an FPGA. First, the place and route of logic gates should be carefully configured to assure the uniqueness of PUF. In other words, the delays of inverter chains should be close enough to take advantage of the process variation. Second, the logic optimization in CAD tools should be turned off for the inverter chains, to prevent the logic elimination. Our experiment implemented the 32-bit RO-based PUF with the *challenge* and *response* protocol for evaluation.

FPGA system implementation

We utilize a soft processor called MicroBlaze from Xilinx for the FPGA system implementation. The MicroBlaze is connected with a monitoring module called SYSMON and a soft PUF via AXI4-Lite. The SYSMON is for capturing the sensor data. It has the analog-to-digital conversion capability with optional physical interfaces such as I^2C. The local memory for the soft processor's execution binary was implemented with Block-RAMs (BRAMs), which is internal memory in the Xilinx FPGA. The Elliptic Curve Digital Signature Algorithm (ECDSA), whose curve parameter is *secp256r1*, is used for the digital signature algorithm in blockchain. It requires a 256-bit private key. Thus, the 32-bit PUF module should be executed eight times with different inputs (*challenges)* for generating a 256-bit private key. SHA256 is used to generate a hash for the transaction. All the software codes are written in C, and the compiled execution binary is included in the bitstream.

Table 1 shows the hardware cost of the main components in the FPGA system. As shown, the system takes only a small amount of hardware resources in the Zynq UltraScale+. The PUF module consumes 1,037 Look-Up Tables (LUTs) and 1,216 Flip-Flops (FFs). The counters for the 32-bit RO-based PUF occupy most of the resources in the PUF module. The MicroBlaze processor consumes less than 0.5% of LUTs and FFs. The SYSMON consumes about 0.05% of hardware resources. The 64KB local memory was implemented with sixteen BRAMs. The AXI4-Lite system bus consumes less than 0.04%. The implemented system can be ported to Spartan-7, one of the cheapest FPGAs from Xilinx. The FPGA system is able to generate up to 33 transactions per minute. The ECDSA takes 1.804 sec, the largest portion of the execution time, and SHA256 takes only 1.668 msec. The power estimation reports a 191 mW for the FPGA system.

Table 1. Hardware cost of main components on FPGA fabric in Zynq UltraScale+.

Main Components	LUTs (274,080)	FFs (548,160)	BRAMs (912)
PUF module (32-bit)	1,037 (0.38%)	1,216 (0.22%)	0 (0%)
Soft processor (MicroBlaze 100Mhz)	1,183 (0.43%)	930 (0.17%)	0 (0%)
Sensor monitoring hardware (SYSMON)	140 (0.05%)	261 (0.05%)	0 (0%)
Local memory (64KB)	0 (0%)	0 (0%)	16 (1.75%)
System bus (AXI4-Lite)	107 (0.04%)	117 (0.02%)	0 (0%)

DISCUSSION

The proposed solution provides a secure blockchain transaction generation for IIoT systems by utilizing the FPGA. Since the role of the embedded processor is simple, it can be applied to a typical IIoT device even with a light-weight CPU. Our approach also provides versatility and flexibility because the soft-core processor is used inside the FPGA fabric; It is versatile in that it can be applied to any blockchain platform because the transaction generation binary in BRAM can be modified to follow the transaction format of each platform. It is flexible in that a transaction protocol and/or cryptographic algorithm can be changed by replacing the execution binary that is part of the bitstream. According to the experiment, the implemented FPGA system consumes a power of 191 mW. Compared with the battery-operated Cortex-M-based processor, which typically consumes hundreds of mW [15], the proposed solution does not require significant additional power to a typical IIoT device.

The performance outcome, 33 transactions per minute, is translated to roughly one transaction per every 2 seconds for each IIoT device. It means that each IIoT device can report sensors' data once every 2 seconds to a distributed ledger. With an example of the Ericsson Panda factory where more than one thousand IIoTs are deployed, it is translated to roughly more than 500 transactions per second (TPS) if the proposed approach is applied. As off-the-shelf blockchain platforms now provide thousands of TPS, and the latest

one offers 10,000 TPS [16], it means that the blockchain platform can accommodate more than 20,000 IIoT devices with our approach. For the machinery demanding blockchain transactions with a higher frequency, the additional migration of time-consuming tasks to FPGA would be an option. Glas et al. [17] reported a 7ms execution time of ECDSA and SHA operations on an FPGA. It means that the FPGA based system can generate more than 100 transactions per second, which can be applied even to autonomous vehicles [18].

CONCLUSION

This paper proposes an FPGA-based private blockchain system for enhancing the integrity and trustworthiness of the data generated from IIoT device. Inside the bitstream protected FPGA, a soft processor, PUF, external register, and local memory are integrated to generate a transaction in an isolated and enclaved manner. The PUF is utilized for the key confidentiality, and the enclaved transaction generation with tightly-coupled sensors provides the data integrity. The experiment with Zynq UltraScale+ shows that the FPGA system provides 33 transactions per minute and consumes a 191mW of power, which would be applicable to battery-operated IIoT devices. The FPGA system is versatile and flexible for various blockchain operations and platforms. In future work, we plan to extend our approach to the cluster level by organizing multiple FPGA devices with blockchain platforms.

■ REFERENCES

1. Reyna et al., “On blockchain and its integration with IoT. Challenges and opportunities,” Future generation computer systems, 88, 173-190, 2018.
2. Stellios et al., “A survey of iot-enabled cyberattacks: Assessing attack paths to critical infrastructures and services,” IEEE Communications Surveys & Tutorials 20.4, 3453-3495, 2018.
3. Chen et al. “Sgxpectre: Stealing intel secrets from sgx enclaves via speculative execution,”, IEEE European Symposium on Security and Privacy (EuroS&P), 142-157, 2019.
4. Jyothi et al., “FPGA Trust Zone: Incorporating trust and reliability into FPGA designs,” IEEE 34th International Conference on Computer Design (ICCD), 600-605, 2016.
5. Wilkinson, Kyle, “Using Encryption and Authentication to Secure an UltraScale/UltraScale+ FPGA Bitstream,” Xilinx Inc, 2017.
6. Mingxiao et al. “A review on consensus algorithm of blockchain,” IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2567-2572, 2017.
7. Huang et al., “Towards secure industrial IoT: Blockchain system with credit-based consensus mechanism,” IEEE Transactions on Industrial Informatics, 15.6 3680-3689, 2019.
8. Dai et al., “SBLWT: a secure blockchain lightweight wallet based on trustzone,” IEEE Access, 6, 40638-40648, 2018.
9. Lin et al., “Blockchain and IoT based food traceability for smart agriculture,” Proceedings of the 3rd International Conference on Crowd Science and Engineering, 1-6, 2018.
10. Mylrea et al., “Blockchain for smart grid resilience: Exchanging distributed energy at speed, scale and security,” Resilience Week (RWS), IEEE, 18-23, 2017.
11. Mazzei et al., “A Blockchain Tokenizer for Industrial IOT trustless applications,” Future Generation Computer Systems, 105, 432-445, 2020.
12. Florin et al., “FPGA based architecture for securing IoT with blockchain,” IEEE International Conference on Speech Technology and Human-Computer Dialogue (SpeD), 1-8, 2019.
13. Taiebat et al., “DISTINGUISHING SENSOR FAULTS FROM SYSTEM FAULTS BY UTILIZING MINIMUM SENSOR REDUNDANCY,” Transactions of the Canadian Society for Mechanical Engineering 41.3, 469-487, 2017.
14. Chanson et al, “Blockchain for the IoT: privacy-preserving protection of sensor data,” Journal of the Association for Information Systems 20.9, 1274-1309, 2019.
15. Boorboor S, Khorsandi M., “Development of a single-chip digital radiation spectrometer based on ARM Cortex-M7 micro-controller unit,” Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 946, 162685, 2019.
16. Insolar Team, “Insolar Technical Paper,” Insolar Technologies, 1-66, 2019.
17. Glas B et al., “Prime field ECDSA signature processing for reconfigurable embedded systems,” International Journal of Reconfigurable Computing, 2011.
18. Schoettle B., “Sensor fusion: A comparison of sensing capabilities of human drivers and highly automated vehicles,” Sustain. Worldw. Transp., 1-42, 2017.

Han-Yee Kim, is a Ph. D student in the Department of Computer Science at the Korea University. He received his B.S. degree from Korea University. His research interests include the hardware based security and cyber system acceleration by utilizing reconfigurable and embedded systems. Contact him at hanyeemy@korea.ac.kr.

Lei Xu, is an assistant professor in the Department of Computer Science, University of Texas Rio Grande Valley. He received his Ph.D. degrees in 2011 from the Chinese Academy of Sciences. His research interests include applied cryptography, cloud/mobile security, and decentralized systems. Contact him at xuleimath@gmail.com.

Weidong Shi, is an associate professor in the Department of Computer Science, University of Houston. He received his Ph.D. in computer science from Georgia Institute of

Technology, where he did research in computer architecture and computer systems. Contact him at larryshi@ymail.com.

Taeweon Suh, is a professor in the Department of Computer Science and Engineering, Korea University. He received B.S. from Korea University in 1993, his M.S. degree from Seoul National University, Korea, in 1995, and his Ph.D. degree from Georgia Institute of Technology in 2006. His research interests include hardware security, AI accelerators, and reconfigurable and embedded systems. Contact him at suhtw@korea.ac.kr.