PCBChain: Lightweight Reconfigurable Blockchain Primitives for Secure IoT Applications

Wei Yan, *Member, IEEE*, Ning Zhang, *Member, IEEE*, Laurent L. Njilla, *Member, IEEE*, and Xuan Zhang, *Member, IEEE*

***Abstract*— In the era of ubiquitous intelligence, the Internet of Things (IoT) holds the promise as a breakthrough technology to enable diverse applications that benefit societal problems. Yet interconnecting myriad heterogeneous IoT devices across various application domains remain a security challenge. Decentralized technology has recently emerged as a powerful primitive in building distributed applications to facilitate secure transactions between mutually distrustful parties in a trustworthy manner. Unfortunately, these decentralized protocols demand computing resources and power far beyond the reach of resource-constrained IoT devices, preventing the full adoption of distributed consensus platform in the IoT setting. In this article, we address the key bottleneck to enable blockchain in resource-constrained IoT devices. We propose a lightweight implementation of proof-of-work (PoW) mining with reconfigurable hardware primitives. By replacing the hash and cryptographic functions in classic blockchain protocol with secure and efficient hardware implementations, our proposed solution can significantly reduce hardware resources and power overheads of PoW mining, while improving the transaction speed of large-scale IoT systems. Finally, we demonstrate the algorithm by proposing an antispoofing solution for GPS navigation among lightweight IoT devices. As a replacement for position computation, a mining process generates the expected coordinates with the correct initial value and function configuration.**

***Index Terms*— Blockchain, configurable nonlinear feedback shift register (CNLFSR), Internet of Things (IoT) security, lightweight hardware primitives, physical unclonable function (PUF).**

I. Introduction

THE advent of smart and connected devices has led to the rapid growth of the Internet of Things (IoT). Many IoT-enabled applications have already started to revolutionize areas such as telemedicine, industrial controls, and smart buildings, and others are poised to address emerging societal needs like intelligent infrastructure, energy exchange, and food security [1]. As illustrated in Fig. 1(a), a typical IoT network often employs a hierarchical topology that uses various standard communication protocols to connect the local edge devices via network gateways to a central cloud service [2]. Despite IoTs' enormous potentials in enabling groundbreaking technology, there are many challenges related to its interoperability and security. For example, current IoT systems are often owned and operated by individual IoT service provider via their proprietary cloud resources, creating information silos and preventing data sharing and integration. Furthermore, when information is centralized, its security hinges on the security purview of a single service provider, which could be concerning for mission-critical systems. Even if the service provider has perfect security, it is also unclear which authority should own such centralized control. To enable the adoption of IoT systems that interconnect crosscutting sectors and application domains supported by noncooperating vendors, there is a pressing need to invoke trustworthy participation from various scales of IoTs in a distributed and decentralized manner.

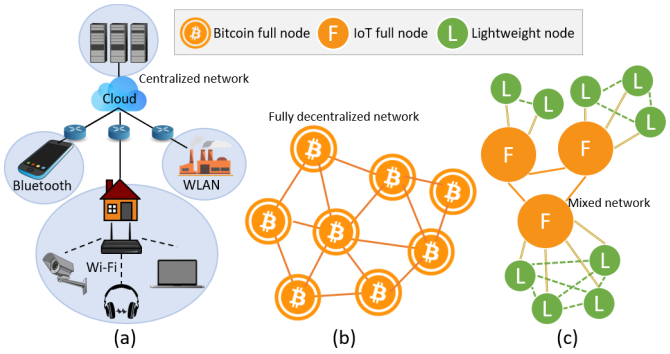

Fig. 1. (a) Network topology of a typical IoT system. (b) P2P network used in blockchain. (c) Hierarchical network topology used in the proposed PCBChain consisting of both full nodes and lightweight nodes.

Blockchain is a novel fully distributed public ledger in the form of a growing list of records linked together using cryptography that is maintained purely by a peer-to-peer (P2P) network of anonymous users/nodes. It was first introduced through the now famous cryptocurrency known as Bitcoin [3]. The use of a cryptographically linked list guarantees the integrity of the record, since, once written, the data on the blockchain are immutable and cannot be changed without breaking the cryptographic seal. While every node in the

Manuscript received February 1, 2020; revised June 21, 2020; accepted July 14, 2020. This work was supported in part by the U.S. Air Force Material Command under Award ICA2018-UP-LN and in part by the U.S. National Science Foundation under Grant CNS-1657562 and Grant CNS-1916926. *(Corresponding author: Wei Yan.)*

Wei Yan and Xuan Zhang are with the Department of Electrical and System Engineering, Washington University in St. Louis, St. Louis, MO 63130 USA (e-mail: weiyan@wustl.edu; xuan.zhang@wustl.edu).

Ning Zhang is with the Department of Computer and System Engineering, Washington University in St. Louis, St. Louis, MO 63130 USA (e-mail: zhang.ning@wustl.edu).

Laurent L. Njilla is with the U.S. Air Force Research Laboratory, Rome, NY 13441 USA (e-mail: laurent.njilla@us.af.mil).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2020.3014155

network has open access to the public ledger, only the node that can successfully solve the cryptographic puzzle, known as proof-of-work (PoW) mining, is allowed to append a new block. The design of the PoW puzzle guarantees that the chance of solving the puzzle is equally distributed among nodes that have the same computing power, and the system is secure if no single node is able to amass more than 50% of the network's computing power. Fig. 1(b) shows how blockchain is established through P2P communication between distributed nodes in the network. Through its PoW-based distributed consensus, blockchain obviates the need for a trusted intermediary and hence avoids the common pitfalls of a centralized architecture [3].

Recognizing the potential of distributed consensus in tackling the challenges in IoT systems, researchers from both academia and industry are interested in exploring the potential of incorporating blockchain in IoT applications [4], [5]. Nevertheless, the complex computation involved in the blockchain protocol poses new bottlenecks for resource-constrained IoT devices [6]. In the fully distributed blockchain network, every mining node has to handle all the cryptographic computations individually. As the system grows in scale, the computational cost increases exponentially. Indeed, energy consumption concern is not specific to IoT blockchain applications: it also presents a universal challenge to the classic blockchain network. The energy consumption of a single Bitcoin transaction can power a U.S. household for 14.46 days [7], and the overall power demand of the Bitcoin network has skyrocketed to more than 40 TWh, which is close to the amount of power consumed by the entire country of Bangladesh [7]. Although alternative proof mechanisms have been proposed to replace the power-hungry PoW mining [8], cryptography remains the most computationally dominant operation, even with these new alternatives. Specifically, the elliptic curve digital signature algorithm (ECDSA) and secure hash algorithm (SHA-256) consume major hardware resources, and dominate the power consumption in the system, making distributed consensus with decentralized architectures less efficient and less practical for IoT applications [9], [10].

As a first step to tackle this challenge, we propose a novel blockchain mining system, with new hardware primitive design, resource-friendly implementation, and secure solution in the context of IoT applications. These hardware primitives can substitute the standard cryptographic functions used in the classic blockchain protocol and significantly reduce the resource and power consumption without compromising the security level, and can be generally applicable to other types of consensus protocols. In addition to the cryptographic substitution, we propose a modified consensus protocol using these hardware primitives. Fig. 1(c) illustrates the hierarchical network topology adopted in our proposed IoT framework. We assume that typical IoT systems consist of two types of network nodes—full nodes and lightweight nodes. The former (typically an IoT gateway or microserver) is not as resource-constrained as the latter (typically low-power IoT edge devices), and therefore can fully support classic blockchain protocols and participate in its distributed mining to ensure the security of the full nodes and their connectivity. Therefore, the key problem we address in this article is establishing a distributed consensus locally among all the lightweight nodes connected to the secure full node. A discussion on ensuring the connectivity and security of the resource-rich full nodes is out of the scope of our current solution. More specifically, our solution leverages the concept of a physical unclonable function (PUF) and a configurable nonlinear feedback shift register (CNLFSR) to replace ECDSA and SHA-256 in blockchain PoW mining. The proposed PUF-CNLFSR-Blockchain (PCBChain) framework aims to achieve comparable functionality to that of blockchain PoW mining with low hardware cost and power consumption, which can enable blockchain technology in myriads of lightweight IoT devices. It opens a new avenue for exploring hardware-protocol codesign solutions for secure distributed IoT applications. To demonstrate the efficacy of this approach in real applications, we implement the PCBChain framework and the embedded functionalities of GPS defense modules on FPGAs. The design uses a pure hardware mining method to calculate the coordinates and ensures communication encryption complexity in different attack scenarios.

We make the following contributions in this article.

1) We develop the PUF and CNLFSR solution to establish new hardware-based cryptographic primitives and enable resource-efficient authentication and mining with reconfigurable security levels.
2) We propose a hardware-protocol codesign framework to facilitate distributed consensus in resource-constrained IoT devices for blockchain applications.
3) We implement PCBChain and GPS spoofing solutions in a real system, and demonstrate a significant resource saving (97%) and performance boost (50×) over the classic blockchain.

II. Background and Related Work

A. IoT and Hardware Security

IoT technology has been continuously gaining popularity in the past few years, but, their security is far from ideal. A wide variety of off-the-shelf IoT devices have been found to be missing basic security protections [11]. To better understand the threat landscape, researchers have conducted in-depth threat analysis at each communication layer and proposed numerous protection mechanisms [12]. However, many of the security mechanisms assume similar deployment environments among IoT devices and traditional servers, which can be prohibitively expensive for resource-constrained devices. To tackle the contention between security and performance, hardware security primitives have drawn attentions from both industry and academia. PUF, due to its unique physical fingerprint, shows promise for the lightweight device identification [13]. Similarly, radio frequency identification (RFID) can track and monitor objects with tags, which has been widely applied to IoTs [14]. Additionally, hardware LFSR is also popular for pseudorandom number generation in the IoT domain [15].

B. Blockchain and Distributed Consensus

Blockchain, the technology behind Bitcoin, is one of the most widely cited distributed consensus protocols [3]. Different from traditional distributed security protocols, where security is guaranteed by cryptography [16] and a fault-tolerance protocol [17], the security of Bitcoin is based on financial incentives during the new block generation process, also known as mining. It is assumed that most of the distributed consensus participants are rational and incentivized to follow the protocol. The joint computation by the consensus network remains correct as long as majority of the participants follow the protocol faithfully. This is the intuition behind the 50% rule of Bitcoin. However, since blockchains are designed to be anonymous, an adversary can simply launch a Sybil attack [18] on the network to gain the majority by impersonating multiple participants. To prevent this, the mining nodes in the network has to solve a computationally expensive crypto puzzle to propose a new block. The additional requirement to show PoW by solving a crypto puzzle not only gives new incentives for miners, but also effectively requires the attacker to own more than 50% of the collective power of the entire network to launch attacks. While PoW works effectively in defending against Sybil attacks, it is an enormous waste of energy. As a result, researchers from both academia and industry have looked into alternatives such as proof-of-useful-work [19], proof-of-stack (PoS) [20], proof-of-authority [21], and proof-of-elapsed-time [22]. Despite these emerging developments and improvements in the distributed consensus protocol, cryptographic operations such as signature verification and hashing remain the fundamental building blocks of any modern distributed consensus protocol and are basic requirements for all participants.

C. Blockchain in IoT

Recognizing the potential of blockchain-based distributed applications, several recent works have attempted to apply such distributed consensus protocols to IoT infrastructure [23], [24]. However, their mining operations are often highly centralized on full resource nodes. As a result, deployment of computation intensive distributed consensus protocols on resource-constrained IoT devices remains an open challenge [25]–[27]. The power consumption and hardware cost are not well balanced, considering the limited improvement of security. Meanwhile, not all the security problems in IoT devices can be addressed by the existing blockchain protocols [12]. Package tampering and IC counterfeiting require complementary innovations in hardware primitives [28], [29]. Hence, we shoot for the new generation of the blockchain design and protocol that can not only achieve the best trade-off of system performance but also prevent IoTs from remote attacks, hardware counterfeiting, and tampering adversaries.

III. PCBChain System Design

A. System Architecture

The newly proposed PCBChain takes a different approach to tackle the conflict between energy-intensive PoW consensus and low-energy requirement of IoT devices. As illustrated in Fig. 1(c), our proposed PCBChain is built on a hierarchical IoT network, consisting of many subsystems at the first level, each of which includes a full nodes and multiple lightweight nodes. These subsystems can exchange information via a P2P network and the standard blockchain protocol. A full node serves as the hub of a subsystem and commands its connected lightweight nodes. They constitute the second layer of the IoT network using *ad hoc* wireless communication. By applying this semidistributed topology, the PCBChain can significantly improve the scalability of the network by degrading a large scale of network, i.e., one billion nodes, to 100 000 or even one million subsystems, in which there are hundreds of lightweight nodes controlled by a full node. This architecture relieves a centralized cloud server of heavy burden through the combination of regional parallel computation and P2P communication among subsystems. The packet traffic and data management in a large amount of IoT network are now shared by all the full nodes.

Compared with a centralized IoT topology that relies on complicate challenge handshake authentication protocols (CHAPs) to ensure security, the PCBChain subsystem also simplifies the authentication and lets nodes check the behavior of each other. In this article, we present our design hardware and protocol design for each of the subsystem as well as the high-level composition. Fig. 2 presents such a PCBChain subsystem and zooms in on internal block diagrams of the full node and one of the lightweight nodes.

1) PCBChain Full Node: The full node, which is nonexistent in the traditional PoW architecture, is designed to release the resource and computation burden to enable PoW on lightweight IoTs. The full node has two roles to perform: it communicates with other full nodes in the main network with the classic PoW protocol and works as a trusted administrator in its own subsystem with our PCBChain protocol. The PCBChain part that we focus on contains six modules: a subsystem controller, true random number generator (TRNG), database, graphic user interface (GUI), user application, and *ad hoc* Wi-Fi. The controller implemented in the FPGA manages all the operations among lightweight nodes. The operating data is provided by an off-chain database, which is stored in the external memory on the FPGA platform. The TRNG in the full node leverages its PUF and CNLFSR to generate the address of challenge-response pairs (CRPs) for node authentication and symmetric key encryption [30]. Full nodes also enable auxiliary GUIs, user applications, and *ad hoc* Wi-Fi communication among nodes. With those modules, a full node can assign the proper mining tasks to its lightweight nodes.

2) PCBChain Lightweight Node: A lightweight node, which serves as a simple IoT device and mining unit, aims to minimize the extra resource utilization of PoW while sustaining the high security level, so that the attack cost is prominently higher than the transaction value to deter malicious attackers. With this purpose in mind, we apply two resource-saving techniques to these nodes: CNLFSR-based mining and bistable ring PUF (BRPUF)-based cryptographic functions to replace the SHA-256 and ECDSA in classic blockchain protocols,

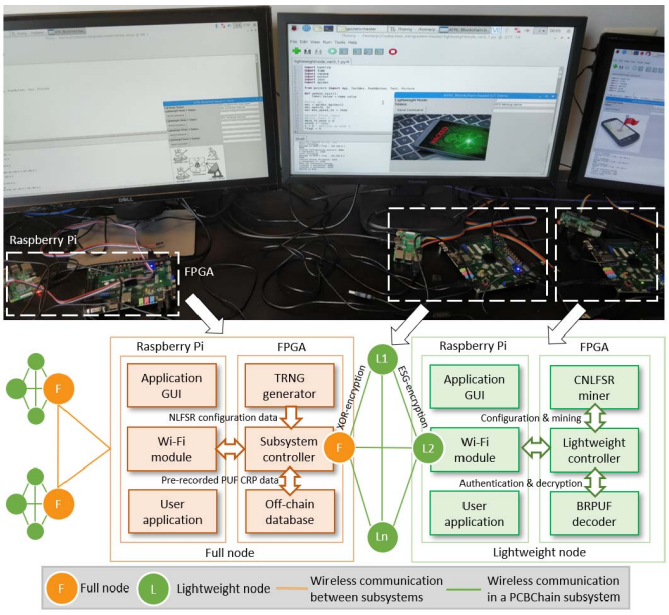

Fig. 2. Block diagram and experimental setup of the proposed PCBChain system.

as detailed in Section IV. Unlike traditional PoW, whose nodes are completely independent, PCBChain lightweight nodes work partially in slave mode, since only indispensable components are reserved internally while other functions are merged to the full node. A finite-state machine (FSM) controller combines these primitives and other modules as one essential PoW node by fetching mining information from the corresponding full node. The FSM also exchanges transactions with other lightweight nodes in the subsystem. For security reasons, one lightweight node is not allowed to communicate with nodes in other subsystems directly, but can transfer the packet to the external network through its own full node.

B. Consensus Protocol

The trust-building process of the classic fully distributed PoW protocol follows the three steps shown in Fig. 3(a): mining with the SHA-256 hash function, broadcasting the mined block, and verifying transactions in the mined block. In step 1, a mining node runs the one-way hash function (SHA-256) to exhaustively search for the nonce that results in successfully mining a unit *coin* (indicated by the light bulb) and the right to append a new block to the public ledger. In the example of Bitcoin, the block contains Bitcoin payment transactions to transfer Bitcoin values between different addresses. Then this successful node signs the freshly minted block with its private key, using ECDSA for an encrypted digital signature. This signed block is broadcast to its neighboring nodes and propagated in the network. Therefore, to support the basic PoW mining in classic blockchain, each node needs to perform repeated SHA-256 and ECDSA signature generation and verification steps, which can be very resource-intensive. To alleviate the computational demand from the lightweight node, our proposed PCBChain modifies the PoW consensus protocol into four steps that require coordination between the full

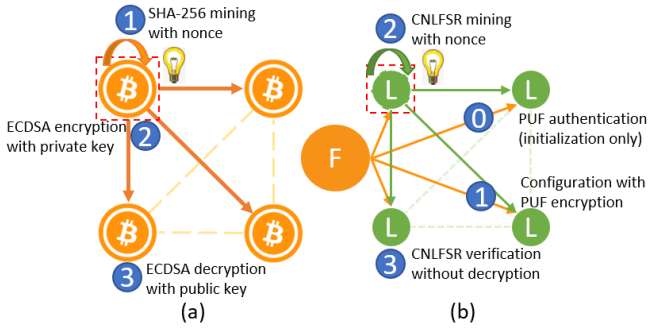

Fig. 3. (a) Classic blockchain PoW protocol. (b) Semidecentralized consensus protocol in PCBChain.

node and lightweight nodes—authentication, configuration, mining, and verification. The four-step protocol is presented in Fig. 3(b) and described as following.

1) Hardware Authentication: To set up a new network or enroll any new device, initialization authentication, marked as Step 0 in Fig. 3, is required to establish trusted communication among nodes and ensure the security of subsequent operations. During the initialization, each lightweight node equipped with PUF receives a challenge from the full node and sends a response back. Credibility depends on how many bits in the response match the corresponding data stored in the full node local memory. Even though a raw response may contain bit errors due to the instability of PUF, we avoid using additional hardware resources for error correction. Instead, the full node supports fuzzy authentication to tolerate the errors [31].

2) Mining Function Configuration: After the authentication, in step 1, we distribute the mining function to each lightweight node by configuring their CNLFSR modules, which involves two $m_{\max}$-bit binary strings: the CNLFSR polynomial function setting parameter F and the mining cycle T. F is sent to the CNLFSR lookup tables (LUTs) to configure the bitwise operation gate to build the mining function. T is used to define how many shift operations need to execute before the mining result is checked. We define m as the valid length of the configured part in the CNLFSR, but it is no larger than a maximum number $m_{\max}$. Thus, the mining cycle can be as large as $2^{T(m_{\max}-1)} + 2^{T(m_{\max}-2)} + \cdots + 2^{T(0)}$. To avoid shared function leakage during the broadcast, we apply a private PUF-based stream cipher for each lightweight node. First, the controller of the full node fetches the second pair of c and r for the target lightweight node, which should be different from the first CRP used for authentication. The secret key r can encrypt T and F, while c is sent to the target directly. Only the lightweight node that has the correct PUF can decrypt T and F. The lightweight node first uses c to generate the corresponding r from its PUF. Then the FSM can decrypt T and F using the bitwise XNOR operation as follows:

$$\begin{aligned} r &= \mathrm{PUF}(c) \\ T &= r \oplus \mathrm{enc}(T) \\ F &= r \oplus \mathrm{enc}(F). \end{aligned} \tag{1}$$

It is important to note that due to PUF instability, a few bit errors in the response cannot be avoided. This can be a serious

flaw for blockchain as the shared polynomial function would be completely different in each node. To solve this problem, we apply the bit selection strategy in the full node and helper data checking in the lightweight node [29]. Only stable CRPs of PUFs and the corresponding helper data are stored in the full node, which ensures a failure rate below 10^{-6}.

3) CNLFSR-Based Mining: After all the lightweight nodes are configured with the same function, the full node launches the mining progress by sending the lightweight node an input consisting of a transaction plus a nonce. The CNLFSR module inside each lightweight node then shifts the register value in each clock cycle until the time reaches the predefault threshold $2^{T(m-1)} + 2^{T(m-2)} + \cdots + 2^{T(0)}$ ($m \leq m_{\max}$). A successful mining result should start with a fixed number of "0"s. The values to the right are appended to a new block in a public ledger that records the transactions locally.

4) Signature Generation and Verification: To simplify the hefty signature generation algorithm in classic blockchain, public PUF (PPUF) is used to replace the asymmetric cryptography in step 3. The security is based on the unclonable feature of hardware circuits and the execution-simulation gap (ESG) principle of PPUFs. For the private PUF encryption, all the CNLFSRs in the lightweight nodes are configured to be the same polynomial function without leaking any secret information to the unauthenticated nodes. Thus, even though the original CNLFSR can be physically cloned, attackers have to try all the configuration patterns before retrieving the correct initial register values. Considering the worst case of using LFSRs, the ESG between the f_{complex} and f_{compact} is given by

$$\begin{aligned} \text{Cycles of } f_{\text{complex}} &= \sum_{i=1}^{m_{\max}} (2^i - 1) \\ \text{Cycles of } f_{\text{compact}} &= 2^m - 1 - \sum_{i=1}^{m} (2^{T(i)-1}) \\ \text{ESG} &= \text{Cycles of } f_{\text{complex}} - \text{Cycles of } f_{\text{compact}}. \end{aligned} \tag{2}$$

Hence, attackers cannot solve the puzzle within the polynomial time when the register maximum length $m_{\max}$ is large enough, while an authenticated node can find the solution quickly. Based on the ESG mechanism, we are able to broadcast the data in a secure way and avoid using any asymmetric cryptographic algorithm. Consequently, the verification process in the lightweight nodes removes the need for complex asymmetric cryptographic modules. Since the PPUF guarantees the data security, any authenticated node can decrypt the received register value by running its own CNLFSR, which is denoted as f_{compact} in (2). If the validity of the value is confirmed, a new transaction is added to the blockchain.

IV. Hardware Design and Implementation

In the hardware layer, we introduce three security primitives that are used in the PCBChain—CNLFSR, BRPUF, and PPUF.

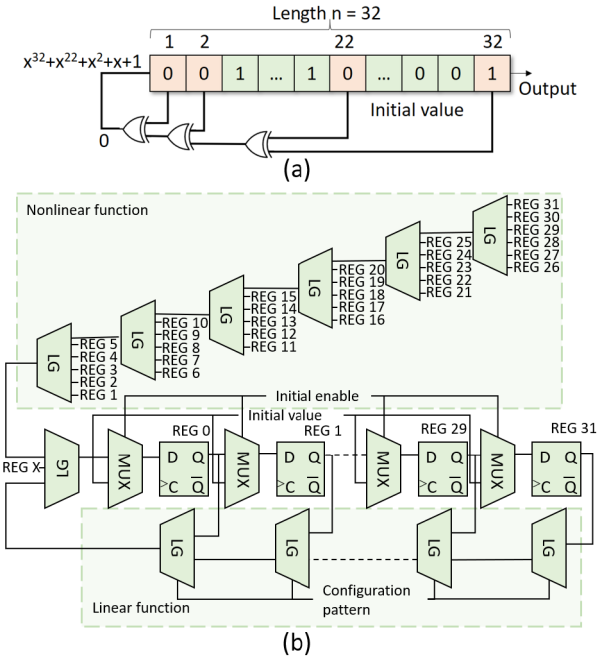

Fig. 4. (a) Fibonacci LFSR architecture. (b) Proposed 32-bit CNLFSR design.

A. CNLFSR

A mining function usually refers to a cryptographic hashing algorithm, which is a one-way function with strong security properties but high cost. Similar to other cryptographic algorithms, most secure hashes are slow in software implementation and resource-hungry in hardware implementations. Therefore, a low-cost yet secure primitive is required for the PoW in lightweight IoT devices. Due to the simple construction of circuit, LFSR is widely used for pseudorandom number generation in cryptography applications [32] and is especially attractive as a low-cost implementation in IoT devices comparing to a complex mining function such as SHA-256. The LFSR, as shown in Fig. 4(a), only requires 32 flip-flops and three gates, but has the capability to provide the maximum period of $2^{32}-1$. However, a classic Fibonacci LFSR is known to be weak against a Berlekamp–Massey attack [33], which makes a guess and simulates the LFSR up to a symbol to see if it is correct. We address the problem by designing a configurable linear feedback shift register on an FPGA. In the linear function part of Fig. 4(b), a LUT is added before each D flip-flop (DFF) [34]. The SRAM-based LUT can be configured as a multiplexer (MUX) to customize the initial value in the DFF. AN LUT based logic gate (LG) is added below each DFF, which serves as a MUX or bitwise XOR operational unit. By initializing LUT cells with the corresponding functionality and changing the selection input value, we can configure the polynomial function of a CLFSR for each mining process to increase the linear complexity [35]. The implementation uses Xilinx Vivado tool and standard Xilinx 7 series FPGAs configurable logic blocks [34], [36]. As a further security improvement, our design involves nonlinear feedback shift registers (NLFSRs) in the nonlinear function part of Fig. 4(b). Another LG combines both outputs to enable the CNLFSR

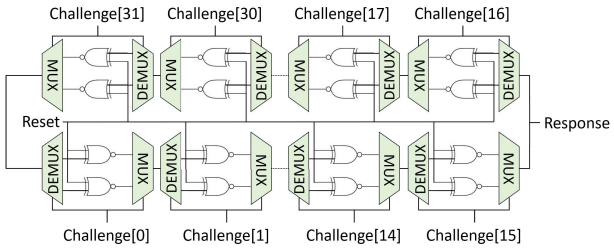

Fig. 5. 32-bit BRPUF design.

to reconfigure the function and switch between nonlinear and linear mode. Given that the linear complexity bounds between $2^{n-1}+n$ and 2^n-1 for each NLFSR, their combination offers stronger resistance against the Berlekamp–Massey algorithm attack [37].

B. BRPUF

A PUF is a physical entity that generates unique results from random uncontrollable variables in the manufacturing process. Given the same challenge, each PUF reacts with a different response, called a CRP. The BRPUF was first proposed in 2011 [38], and has a limited cost and decent reliability compared to other types of delay PUFs. Its FPGA design, with 32 stages of duplicated NOR gates, is shown in Fig. 5, which contains a 32-bit challenge and a 1-bit response. Each basic stage consists of two NOR gates: a MUX at the gate outputs and a DEMUX at the gate inputs. The MUX and the DEMUX share a select signal which is one bit out of the 32-bit challenge signal. This signal selects either the upper or the lower paths of the MUX and the DEMUX, which provide different delays and determine the 1-bit output after the stabilization process. By storing the challenge in an LFSR, we are able to generate a 32-bit response with the same challenge. The response includes 2^{32} patterns and will be the initial value of the CNLFSR. The BRPUF requires low hardware resource utilization on FPGA and application-specific integrated circuits (ASICs). For example, the NOR gate, MUX, and DEMUX can be implemented with one LUT. A 32-stage BRPUF only requires 128 LUTs in total. It is noted that the delay-based PUF exerts rigorous timing constraints to the internal wires, which make placement and routing a critical step during the implementation. The details can be found in our previous works [39], [40]. Due to the low cost and secure features of BRPUF, it can be applied in our lightweight blockchain infrastructure for node authentication, stream cipher, and random number generation.

C. PPUF

A PPUF is another concept involved in our PCBChain for low-cost asymmetric key cryptography [41]. Though it is a hardware puzzle, the circuit is public so that the delay of each gate can be simulated, which is extremely time consuming. As the ESG described in (2), the hardware circuit is regarded as a private key (f_{compact}) whereas the simulation based on the digital library is a public key (f_{complex}). When $m_{\max}$ is much larger than m, the ESG between f_{complex} and f_{compact} can establish a secure hardware-based n-to-n public key communication with low cost. To build such an asymmetric key algorithm, PCBChain requires no additional hardware, simply combining the existing BRPUF and CNLFSR. Authorized by the BRPUF, the legit-configured CNLFSR in each node provides the mining result as an "encrypted" transaction. It takes other authenticated nodes f_{compact} cycles to decrypt the packet, and further verify the correctness of the mining result. A cloned node, which contains a counterfeiting PUF and fails in the PUF authentication, have to spend much longer time trying every possible pattern on its unconfigured CNLFSR to obtain the transaction value.

V. Security Analysis

In this section, we present the security properties of PCBChain using theoretical formulations, and we illustrate how certain well-known attacks can be mitigated in our proposed solution under a typical IoT system setting.

A. PCBChain Mining Security

Generally, using LFSRs directly as hash functions or stream ciphers needs further artifices to counteract the weaknesses implicit in the linearity. Otherwise it is regarded as insecure because the Berlekamp–Massey algorithm can find the shortest LFSR length n for a given binary output sequence within $O(n^2)$ operations [42]. A common solution is to combine several LFSRs to build a keystream with the desired statistical properties, which is known as Geffe generator [43]. However, this keystream is also easily broken by correlation attacks, which significantly reduce the breaking effort compared to brute-force attacks. To improve the security, correlation immunity is commonly required between each LFSR subset and the final combination output. Mathematically, for any independent n binary random variables $x_0, \ldots, x_{n-1}$, if the array $y = f(x_0, \ldots, x_{n-1})$ is independent of any random vector $(x_{i_1}, \ldots, x_{i_k})$ with $0 < i_1 < \cdots < i_k < n$, then the function f is kth order correlation immune. When the Boolean function f holds a low-order correlation immunity, it is more susceptible to a correlation attack. To improve the resistance against cryptoanalytic attack, PCBChain uses NLFSRs with dynamic configurations, which avoids the use of the same function by issuing a new configuration pattern in the next mining process. According to Table I, while sequences generated by an LFSR of order n have a fixed linear complexity of n, an NLFSR provides a lower bound of $2^{n-1}+n$ and upper bound of 2^n-1 when n is larger than 2 [44]. When we combine two linear functions with bitwise logic operations, the output linear complexity turns out to be only the sum or product of two input linear complexity. When the nonlinear function is involved, the output linear complexity Λ is determined not only by the two shift registers, but also by the logic function. With variations in the register length, the CNLFSR guarantees the security of the mining function within the polynomial time. In Table I, we list the linear complexity of the LFSR, NLFSR, and CNLFSR. Compared to SHA-256 used in typical

TABLE I
COMPLEXITY OF MINING FUNCTIONS

	SHA-256	LFSR	NLFSR	CNLFSR
Number of functions	1	$\frac{\phi(2^n-1)}{n}$	$2^{2^{n-1}-n}$	$\sum_{i=2}^{n} 2^{2^{i-1}-i}$
Output length	256	n	n	n
Number of output	2^{256}	$\frac{2^n-1}{n}$	$\frac{2^n}{n}$	$\frac{2^n}{n}$
Maximum linear complexity	NA	n	2^n-1	$\sum_{i=2}^{n}(2^i-1)$
Minimum linear complexity	NA	n	$2^{n-1}+n$	$\sum_{i=2}^{n}(2^{i-1}+i)$
Collision complexity	$2^{28.5} \sim 2^{255.5}$	NA	NA	NA

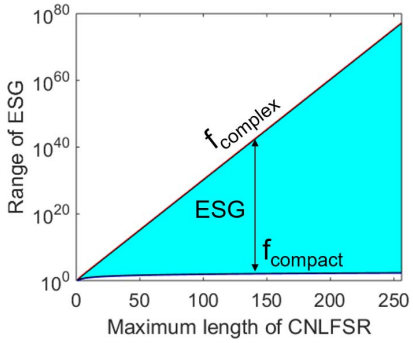

Fig. 6. ESG between the configured CNLFSR and the unauthenticated lightweight node.

blockchain protocols [45], breaking a CNLFSR can be more difficult when n is large enough.

B. PCBChain Digital Signature Security

Fundamentally, asymmetric cryptographic algorithms make a ciphertext encrypted by anyone impossible to decrypt in polynomial time, while another key holder can easily find the plaintext. The ESG generated from the PPUF determines the security level of the nodes' digital signature. Ideally, f_{complex} should be much larger than f_{compact} to ensure the complexity of the cryptographic algorithm. According to (2), f_{complex} has a fixed complexity since the register maximum length $m_{\max}$ is physically preset. f_{compact}, however, can be variable. Given a different configuration F, T in (2) has a flexible range from m to $2^m - 1 - m$. As the shaded area shown in Fig. 6, ESG grows exponentially as $m_{\max}$ increases. According to the estimate, we can safely assume that the asymmetric key cryptographic algorithm in our PCBChain is secure, given sufficient CNLFSR length.

C. System-Level Security

The 51% attack is a well-known adversary for PoW systems, where the adversary can potentially control more than 50% of the network's computing power to halt new transactions or to allow double spending. In any IoT-based blockchain architecture, it is often desirable to keep the energy cost per block/transaction as low as possible. This requirement, however, creates a fundamental contention with the requirement that it would be very difficult for the adversary to obtain more than 51% of the computation power of the network. Using the on-demand Cloud infrastructure, attackers can scale up the computation significantly with marginal cost, compared to IoT devices. As a result, the PCBChain topology cannot defend against an attacker who compromises one of the IoT sensors to delegate the mining computation to an external entity. However, the original design of crypto puzzle was to prevent the Sybil attack in the block proposing stage [18], and in our PCBChain, due to the use of PUFs, it becomes possible to detect and attribute an anomaly to an individual device. One way to build an anomaly detector is to examine the deviation of the block mining rate of a device from its expected mining power. On the other hand, since full node has the privilege of selecting the configured equation, it is able to specify a certain lightweight node with the shortest mining time. As seen in the application that we will introduce in the next section, a lightweight node with the correct CNLFSR initial value and configuration can achieve the expected output within much fewer calculation iterations, compared to other nodes with random initial values. Even if more than half of the lightweight nodes in a subsystem are controlled by attackers, a "cheating note" offered by the full node can still make a specified node win the mining competition. Consequently, the PCBChain tackles the 51% challenge from a different angle, thanks to its unique design and protocol.

VI. PCBChain Applications

Our lightweight blockchain design can be applied to many embedded systems, such as smart homes and home medical devices. PCBChain also efficiently solves the identity and integrity problems in a supply chain. Moreover, it helps build secure communication among untrusted objects, for example, unmanned drones with IoT networks. In this section, we propose a low-power and low-cost blockchain approach to protect GPS coordinates during insecure communication.

A. Military GPS Attack Assumptions and Scenarios

We simplify a military GPS navigation system to a full node and several lightweight nodes, which can be mapped to the elements in the PCBChain subsystem. The full node acts as the commander of this system, sending coordinates to other individuals in the network as their next destination. Lightweight nodes follow the instructions of the commander and move to the expected locations. Our assumptions on the GPS attacks are as follows.

1) Lightweight nodes have the PCBChain information pre-stored in the off-Chain of a full node. The data are shared only to the full node, and attackers cannot access this off-Chain physically or remotely.
2) Attackers can break any well-known or conventional cryptography during real-time communication and obtain the broadcast content among nodes. Attackers can also modify the encrypted packet in the wireless network.
3) Attackers have no knowledge of the hardware primitives in either the full node or the lightweight nodes. The lightweight node has secure tampering resistance, so physical attacks and information leakage are not taken into consideration.

4) Since the PUF and CNLFSR are secure, attackers need additional time to guess the PUF type and the keys to decrypt the authentication, configuration, and verification packets.
5) Once the lightweight node moves to the next coordinate, the previous location information becomes invalid. Thus, an efficient attack must solve the problem before the position update of the victim.

Attacks, hence, may occur under two conditions. The first case is a man-in-the-middle attack, which is based on the assumption that the conventional cryptography of communication is broken and the location content is disclosed. The expected coordinate is maliciously modified to mislead lightweight nodes about the next position. To convince the full node, the new coordinate reported by the corresponding lightweight node will also be altered to match the intended result and pass the verification. The second method of attack is to decrypt the response sent from the lightweight node directly and obtain its latest location information. According to these assumptions, the software encryption algorithms are broken fast enough before the node moves to the next position. Unfortunately, the existing solutions show poor efficiency and low security level on lightweight IoT devices [46]. Nevertheless, the proposed PCBChain exploits the hardware primitive security to find an answer against information modification and leakage.

B. PCBChain-Based GPS Antispoofing System

We apply the same architecture of Fig. 2 to the GPS security solution. The difference is that the CNLFSR initial value (x_n, y_n), configuration data $(\Delta x_n, \Delta y_n)$, and mining result (x_{n+1}, y_{n+1}) are meaningful coordinates in this application. As illustrated in Fig. 7, an isolated off-chain database provides the crypto-CRPs and coordinates on the scheduled paths for all the lightweight nodes of the subnetwork. This database helps the full node to determine which node takes the shortest time for mining while arriving the specified location. The stored data are encrypted to ensure the information security. Any remote access to the off-chain directly is not allowed as well. The full node in the subsystem is the only one that has privilege to physical access to the off-chain database. To transfer the moving instruction $(\Delta x_n, \Delta y_n)$ securely, the full node encrypts it with different symmetric keys k_s and broadcasts $\text{Enc}(\Delta x_n, \Delta y_n, k_s)$ in the subsystem. Each lightweight node can decrypt a corresponding packet with its PUF. The configured nodes start mining with the current position (x_n, y_n) and the decrypted $(\Delta x_n, \Delta y_n)$. In the example, lightweight node 1 is expected to finish the mining first. The generated (x_{n+1}, y_{n+1}) denotes the next position to go. An encrypted packet $\text{Enc}(x_{n+1}, y_{n+1}, k_a)$ will be broadcast to other nodes for verification. The time gap for different roles will be discussed in the later section.

This PCBChain application contains three prerequisites for successful navigation: 1) a full node knows the initial value of each node; 2) the mining function configured by $(\Delta x_n, \Delta y_n)$ satisfies the equation $(x_{n+1}, y_{n+1}) = (x_n, y_n) + (\Delta x_n, \Delta y_n)$; and 3) a specified lightweight node (lightweight node 1 in the example) that is expected to moving along the direction vector $(\Delta x_n, \Delta y_n)$ can calculate the result (x_{n+1}, y_{n+1}) within a shorter time than other nodes. To prove that a full node can always find a shortest path for a certain node, we actually compare the ratio between f_{compact} and f_{complex}. Because all the initial values are uniformed to fit the same CNLFSR, both functions are fixed to the same configuration. While f_{compact} defines the shortest path ① from (x_n, y_n) to (x_{n+1}, y_{n+1}), other nodes that take longer mining time must have their initial values located on the path ②. Given a random initial value, the probability that it is enclosed on the path ② is equal to $(f_{\text{compact}}/f_{\text{complex}})$. According to (2), when $m \ll m_{\max}$, $(f_{\text{compact}}/f_{\text{complex}}) \to 0$. As a result, a targeted node can be guaranteed to win a mining.

Given those constraints, a full node is able to assign verifiable moving tasks to any node in the subsystem, and any malicious modification of the packets can be detected by the transparent computation shared among all the nodes. To launch a successful man-in-the-middle attack, one have to model the hardware architecture to create the correct fault injection value and pass the verification. We will prove that such a modeling attack is not feasible for the BRPUF-CNLFSR structure. To guarantee the broadcast result security, we will also demonstrate that only the full node and configured lightweight nodes are able to verify the mining output. Any brute force attack cannot extract the encrypted coordinates within the polynomial time.

C. Cryptography Modeling Attack and CNLFSR Defense

We first discuss the cryptographic attack model and how PCBChain prevents a spoofing attack. According to the previous assumption, man-in-the-middle attackers cannot get access to the hardware devices. To pass the authentication or decrypt $\text{Enc}(\Delta x_n, \Delta y_n, k_s)$ to obtain the configuration function $(\Delta x_n, \Delta y_n)$, adversaries have to apply a modeling attack to build the digital library of a PUF and predict the key (response) according to the k_s (challenge). With the PUF response, it is easier to calculate where a lightweight node is expected to go. Furthermore, the adversary can modify $(\Delta x_n, \Delta y_n)$ to maliciously change the destination to (x'_{n+1}, y'_{n+1}). Finally, to convince the full node that the command is executed correctly, attackers have to replace the packet with the computed (x_{n+1}, y_{n+1}) instead of (x'_{n+1}, y'_{n+1}).

To build the PUF model and predict the responses of BRPUF, attackers have to know the PUF type and collect enough CRPs between a full node and lightweight nodes. Unfortunately, without knowledge of the PUF structure, remote attackers cannot build an effective attack model. Moreover, only the authentication CRP can be directly measured, which occurs once when there is a new node joining the subsystem. To obtain the other CRPs, the adversary first must solve each NLFSR problem and operate the bitwise XOR to decrypt one response in reverse. According to Table I, any NLFSR indicates a minimum complexity of $2^{n-1} + n$. Therefore, the difficulty of breaking PCBChain is much harder and more time-consuming than breaking a conventional PUF modeling attack.

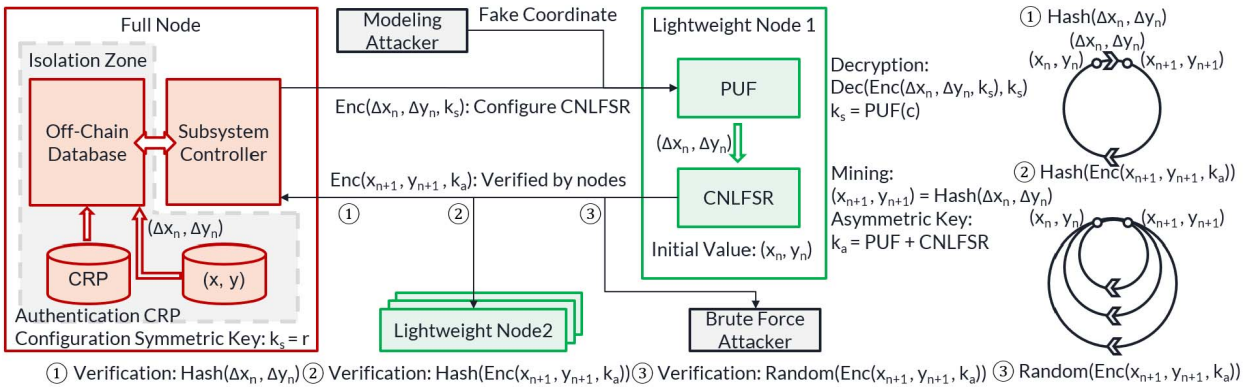

Fig. 7. Workflow of PCBChain based GPS security solution.

Due to the nonlinear property of PCBChain, traditional arbiter PUF modeling methods such as logistic regression cannot be applied here [47]. Instead, we use an open-source deep learning algorithm to attack universal PUFs and NLFSR [48]. Since PCBChain uses combined hardware primitives, we need to build our own model first. Considering a BRPUF with $m_{\max}$-bit challenges C, when an inverter ring is formed and released from an all "0" state, the stabilization process settles on a "0101," "1010," or $NOT\ READY$ state, according to the process variation and system noise, which leads to 2^n different converging results. The delay difference of odd stages is defined as $\vec{\omega}_o = (\omega_1, \omega_3, \ldots, \omega_{n-3}, \omega_{n-1})$, and that of even stages is represented as $\vec{\omega}_e = (\omega_2, \omega_4, \ldots, \omega_{n-2}, \omega_n)$. Given the corresponding challenge vector sets $\vec{\Phi}_o$ and $\vec{\Phi}_e$, the total delay time difference Δ can be expressed as $\Delta = \vec{\omega_o}^T\vec{\Phi}_o - \vec{\omega_e}^T\vec{\Phi}_e$. Hence the 1-bit response r can be calculated with a sign function

$$r = \mathrm{sgn}(\Delta) = \mathrm{sgn}\left(\vec{\omega_o}^T\vec{\Phi}_o - \vec{\omega_e}^T\vec{\Phi}_e\right). \tag{3}$$

For the CNLFSR modeling, we discuss the linear part and nonlinear part separately. The linear function is a polynomial of degree n shown in a form like $f(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_2x^2 + a_1x + a_0$. The real numbers can be replaced by a weight vector $\vec{\omega}_L = (\omega_{L_0}, \omega_{L_1}, \ldots, \omega_{L_{n-1}}, \omega_{L_n})$. We can easily build a model f_L for an LFSR containing $((\phi(2^n - 1))/n)$ functions, in which ϕ denotes the Euler phi function. However, the number of cyclically equivalent sequences defined by de Bruijn has $2^{2^{n-1}-n}$ possibilities, which makes NLFSR modeling harder than that of LFSR [49]. To follow the default form, we also use the Boolean function F to describe the NLFSR $f(x_0, x_1, \ldots, x_{n-1}) = x_0 + F(x_1, \ldots, x_{n-1})$ [49]. Considering the modeling and computing complexity, we focus on the NLFSR function f_N of order 32 or less. The combined f_C for the mining function output therefore can be expressed with linear and nonlinear parts

$$\begin{aligned} f_L &= \vec{\omega_L}^T\vec{x}_i \\ f_N &= \vec{\omega_N}^T\left(\vec{x_i}^T\vec{x}_i\vec{x}_i\vec{x}_i\right) \\ f_C &= f_L + f_N. \end{aligned} \tag{4}$$

Given a response set generated by the BRPUF as the CNLFSR initial value, the system model is no longer a simple PUF or LFSR. In an experiment shown in the later section, we will prove that the decryption of (x_{n+1}, y_{n+1}) is still a hard problem with deep learning technology.

D. Brute Force Attack and Time Gap Defense

As shown in Fig. 7, the second attack scenario is a brute force decryption of the broadcast packet from lightweight node 1. In the example, three roles are involved during the verification process: the full node, lightweight nodes, and the brute force attacker. To maintain a normal working status of the subsystem, the verification duration of the full node and lightweight nodes should be short. On the attacker side, the decryption time must be long enough to build a time gap. The security assessment depends on whether the attacker can solve the mathematical problems before the current coordinate becomes invalid.

The full node, which has all the knowledge of the subsystem, verifies $\mathrm{Enc}(x_{n+1}, y_{n+1}, k_a)$ by executing the same mining function, which is defined as $\mathrm{Hash}(\Delta x_n, \Delta y_n)$. Thus, the verification duration is equal to the mining time of the lightweight node that broadcasts the new position information. As illustrated in Fig. 7, the calculation complexity is abstracted as the shortest path from the former coordinate (x_n, y_n) to the new location (x_{n+1}, y_{n+1}), which takes the following time to verify:

$$t_1 = \sum_{i=1}^{m}(2^{T(i)-1}) \times t_{\mathrm{clk}} \tag{5}$$

where t_{clk} is the system clock frequency. Obviously, the verification duration t_1 is controlled by the valid CNLFSR length m and the corresponding configuration function $T(i)$.

On the other hand, all the lightweight nodes share the same CNLFSR configuration but do not know the initial value (x_n, y_n) generated by the PUF on lightweight node 1. Since the CNLFSR is a one-way function, we cannot reverse the function to calculate (x_n, y_n) directly. It is noted that even though the recorded process (x_{n-1}, y_{n-1}) to (x_n, y_n) in the last mining can indicate the initial position of the current mining by default, the actual (x_n, y_n) should be generated by the PUF privately and kept confidentially until verification is done. Hence it is impossible to replicate the calculation path

① as well. To prove $\mathrm{Enc}(x_{n+1}, y_{n+1}, k_a)$ as a correct mining result, we have to follow the path ② shown in Fig. 7. Given the proper m and $T(i)$, the path ② can be set with a longer time than t_1, but much shorter compared to random collisions

$$t_2 = \left(2^m - 1 - \sum_{i=1}^{m}(2^{T(i)-1})\right) \times t_{\mathrm{clk}}. \tag{6}$$

The attackers, however, are unable to decrypt the PUF response, CNLFSR configuration function $T(i)$, and the maximum length $m_{\max}$. Eventually, an adversary has to try all the configuration functions with the length varying from 2 to $m_{\max}$ until a collision is found. The average probability of collisions requires a computation time of t_3 to reach

$$t_3 = \frac{1}{2}\sum_{i=1}^{m_{\max}}(2^i - 1) \times t_{\mathrm{clk}}. \tag{7}$$

If $m_{\max}$ is large, the verification gap is sufficient to prevent brute force attacks.

VII. Evaluation Results

We evaluate the performance of PCBChain by implementing one full node and two lightweight nodes on three Zedboards, as shown in Fig. 2. Considering the measurement period in reality, the CNLFSR and BRPUF are configured to 32-bit mode. The maximum execution round is set to $2^{32} - 2^{20} = 4\,293\,918\,720$, which takes a Zedboard about 43 seconds to accomplish. All the measurements are performed at the normal working temperature and supply voltages of FPGAs. To compare the mining function and digital signature, we also implement a classic PoW with SHA-256 and ECDSA on KC705 board. As described in the manual [34], both platforms use Xilinx 7 series FPGAs, which have the same structure and cells. Additionally, SHA-256 is tested on a Raspberry Pi.

A. Mining Function Performance

We first compare the mining performance of CNLFSRs with that of the typical SHA-256 function implemented on Xilinx FPGA and Raspberry Pi platforms. The classic software SHA-256 is tested on the Cortex-A9 SoC of a Raspberry Pi. The classic hardware SHA-256 is evaluated by a Kintex-7 FPGA on a KC705 board. The CNLFSR is implemented on the embedded Artix-7 of a Zedboard. As Table II shows, although the resources for the software approach are hard to estimate, the required processor is equivalent to 26 million transistors. Between the cheaper hardware implementations, the SHA-256 requires far more LUTs and FFs than a CNLFSR does. While the Zedboard offers the slowest clock speed, the proposed CNLFSR performs with the highest mining rate. Moreover, the power consumption of both SHA-256 approaches is much more than that of the CNLFSR. To make a fair comparison of mining functions on different platforms, we evaluate not only the platform power but also the energy cost for each hash execution. Hardware approaches, especially for CNLFSR of PCBChain, are much more energy-efficient and thus more suitable for low-power IoT devices. We make thorough randomness assessments according to 15 NIST tests [50], which prove the CNLFSR output to be random. The SHA-256 can generate unique hash values approaching 50% differences with various inputs. In PCBChain, the hamming distance heavily relies on the output of the BRPUF, which varies from 45% to 55%. Finally, we provide the stability performance between the SHA-256 and CNLFSR. While both the CNLFSR and SHA-256 are deterministic hash functions with 100% output accuracy, the BRPUF output is known to generate up to 2.16% bit errors in responses [38]. With bit selection strategy, the failure rate of the mining function is reduced to 10^{-6}.

TABLE II
Mining Functions Performance Comparison

	Classic SW	Classic HW	PCBChain
MPU type	Cortex-A9	Kintex-7	Artix-7
Resource (LUT/FF/gate)	26M	2,668/2,609	73/36
Clock (MHz)	1,400	200	100
speed (hash/s)	27K	2.4M	3.13M
Power (W)	2.05	1.82	0.12
Energy per hash (J/hash)	7.59×10^{-5}	7.58×10^{-7}	3.83×10^{-8}
Randomness (NIST)	pass	pass	pass
Uniqueness	$\rightarrow 50\%$	$\rightarrow 50\%$	$45\% \sim 55\%$
Stability	100%	100%	$> 1 - 10^{-6}$

B. Digital Signature Performance

Table III presents the digital signature performance with three different approaches: ARM-based classic ECDSA [51], FPGA-based classic ECDSA, and CNLFSR/BRPUF-based ESG. Compared to the resource usage for the software and hardware ECDSA, the verification cost of PCBChain is extremely low. To meet the timing, we slow down the clock for the ECDSA implementation on the FPGA. Even so, the signature generation and verification speed are still much faster than that for the software approach. Furthermore, the PCBChain generation and verification progress requires less time than the hardware ECDSA does. Since we apply ESG methodology in PCBChain, there is no additional cryptographic process. Thus the signature generation time in PCBChain is 0. Although there is no traditional decryption for PCBChain, a lightweight node still needs to run the CNLFSR for $2^{20} - 1$ cycles to obtain the transaction and nonce. Thus the PCBChain verification takes 10 ms in our settings. Although the power consumption of classic software PoW is not given in the previous work [51], it is easy to estimate the bound on ARM7. Because the processing time is much longer than that with the other methods, the software approach shows poor energy efficiency. By comparing the overall performance of different digital signatures, PCBChain still performs with the best trade-off.

C. PCBChain System Performance

Based on the measured results, we are able to simulate the performance of PCBChain with a large scale system of IoT

TABLE III
DIGITAL SIGNATURE PERFORMANCE COMPARISON

	Classic SW	Classic HW	PCBChain
MPU type	ARM7 TDMI	Kintex-7	Artix-7
Resource (LUT/FF/gate)	74,000	13,512/15,669	201/68
Clock (MHz)	100	50	100
Generation (ms)	153.5	7.51	0
Verification (ms)	313.4	9.09	10
Power (W)	≈ 0.8	2.58	0.18
Energy per signature (J/sig)	$\approx 3.73 \times 10^{-1}$	4.28×10^{-2}	1.8×10^{-3}

TABLE IV
DIFFERENT LIGHTWEIGHT BLOCKCHAIN SYSTEM COMPARISON

	REM	Microchain	PCBChain
Hardware platform	CPU	CPU	FPGA
Protocol type	PoW	PoS	PoW
Mining function	SHA-256	SHA-256	CNLFSR
Software overhead	$5.8\% \sim 14.4\%$	Not mentioned	$0.6\% \sim 4.7\%$
Scale complexity	$O(K)$	$O(K^2)$	$O(K)$
Throughout (M/h)	$0.5 \sim 4$	$202 \sim 405$	≈ 4.8

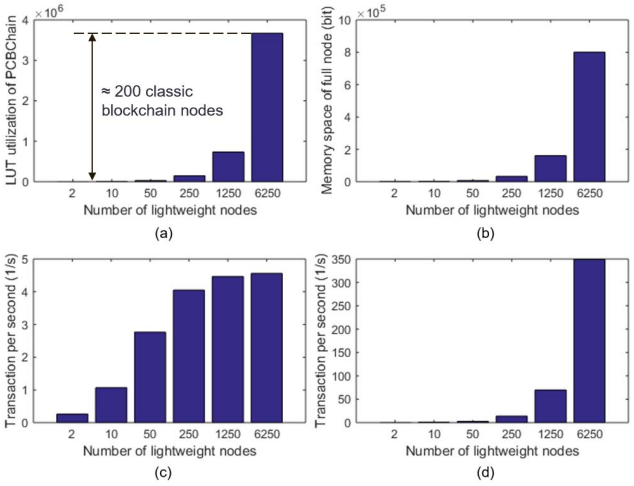

Fig. 8. Relationship between the number of lightweight nodes and (a) total LUT resource usage. (b) Full node memory space. (c) TPS of a full node PCBChain. (d) TPS of multiple full node PCBChain.

devices. Fig. 8(a) shows the LUT utilization of the whole system, which indicates that a PCBChain subsystem with over 6000 lightweight nodes demand for fewer resources than 200 classic blockchain nodes implemented on FPGAs [10]. While our design fits lightweight nodes into low-cost IoT devices, the full node requires more space to store CRPs and the blockchain information. Fig. 8(b) presents the minimum memory needed for one-time mining. Using a subsystem with 6250 lightweight nodes as an example, the full node should provide 10-MB memory for the PCBChain mining progress; there is no memory cost for lightweight nodes.

Another key performance metric of a blockchain system is its transactions per second (TPS) rate. Latencies such as mining, wireless transfer, memory operations, and other factors are considered during the simulation. As indicated in Fig. 8(c), when all the lightweight nodes rely on only one full node in a centralized configuration, the transaction rate is low, experiencing bottleneck due to limited data transfer parallelization. Instead of using the traditional IoT network, we adopt a semicentralized topology by distributing the lightweight nodes among multiple full nodes, with each full node connecting to a certain number of lightweight nodes. Fig. 8(d) indicates that the TPS rate can reach 350, which is 50 times faster than the upper bound of Bitcoin. This result suggests that the scalability issue of the blockchain-based IoT can be solved by constructing such a hierarchical topology.

We compare the key performance with two similar frameworks in Table IV. One resource efficient PoW, REM, relies on the software guard extensions (SGX) framework [19]. Another design called Microchain is built with standard ARM core [52]. Both solutions use SHA-256 as the mining function. Leveraging the hardware primitives of our PCBChain, the lightweight nodes require less software overhead compared to the REM, but the full node demands extra resources for computation. Though the hardware cost is not specified in other works, it is important to note that our PCBChain is the only one that can execute mining without microprocessors, and the cost is less than 600 LUTs. The Microchain owns a communication complexity of $O(K^2)$ during the voting process, in which K is the linear scale to the committee size. On the other hand, the PCBChain and REM have a scale complexity of $O(K)$ due to the different protocol type. When the block size varies among 512 KB, 1 MB, 2 MB, and 4 MB, the average throughout of PCBChain is 4.8 MB/h, which is higher than that of the other PoW, but slower than the PoS solution. Nevertheless, the high throughout of PoS is based on sacrificing the security level against Bribe attack, whose cost is 50× lower than PoW Bribe attack [53].

D. GPS Security Evaluation

To evaluate the security level of our GPS antispoofing solution, three 32-bit BRPUF instances are implemented on the Zynq reprogrammable logic side of different Zedboards. Each BRPUF is connected with a CNLFSR, and the response is used as the initial value of the CNLFSR. Hence the mining output is determined by different PUF responses and function configurations. In the first step, we use the 32-bit CRPs and CNLFSR outputs for training to build the PCBChain model and estimate a prediction rate of its output. Considering the limited memory resources of FPGAs, we store only 6250 reliable and random CRPs on the FPGA block RAMs as the input. With the help of bit selection algorithms for PUFs, we are able to generate deterministic initial values for the mining functions, which eliminate the system level errors in the PCBChain output. The training process thus achieves the highest prediction rate to evaluate the lower bound of the PCBChain system security against modeling attacks. Fig. 9 shows the prediction ratio after training using different models. By applying the BRPUF

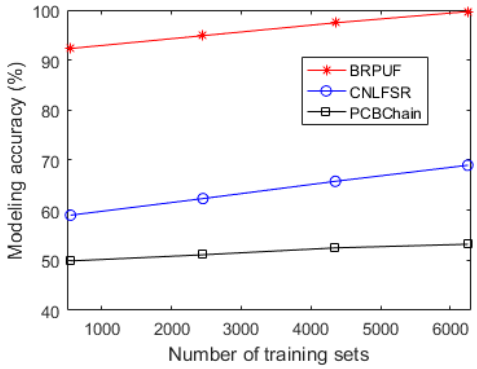

Fig. 9. Matched bit ratio of PCBChain output using modeling attacks.

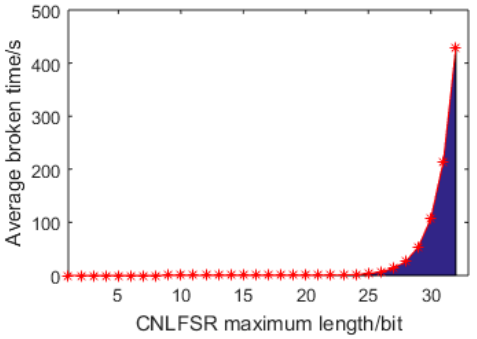

Fig. 10. Secure time threshold against the brute force attack during verification.

model proposed in (3), over 90% of the PUF outputs can be forecast. It is noted that the rate does not reflect the difficulty of decrypting a packet, since attackers cannot obtain the responses directly. Relatively, the CNLFSR model provides an accuracy of under 70% when the training samples are fewer than 6250. When we combine those two primitives, the curve becomes very flat, which means the predication rate of the model are no better than a guess from random binary strings. Hence we can safely claim that the cryptography model cannot be established with the limited samples and time, and attackers are not provided with sufficient data to decrypt the coordinate information.

Further, we evaluate the brute force attack on verification in this application. Based on our assumptions, one of the time constraints is that a lightweight node should move from (x_{n-1}, y_{n-1}) to (x_n, y_n) within t_3. Compared to the worst case shown in Fig. 6, the mining time of the lightweight node can be ignored when $T(i)$ and m are always selected to intentionally form a short path from (x_n, y_n) to (x_{n+1}, y_{n+1}). In such a case, we can test the time bound of a lightweight node being transported to avoid brute force attacks. The shaded area in Fig. 10 represents the secure area that meets the time constraint, with a regular system clock of 100 MHz. The trend indicates that a register length larger than 25 bits is more realistic for lightweight nodes to move to the next location before the current configuration is broken. For example, a 32-bit CNLFSR provides a 7-min secure time window in Fig. 10. If the distance between two coordinates requires more time to cover, we should consider extending the maximum length of CNLFSRs.

VIII. Conclusion

This article takes the first step to explore the potential of using lightweight hardware implementation for blockchain mining in IoT systems. Leveraging two proven hardware primitives—PUF and —LFSR, our solution extends their usage into BRPUF and CNLFSR to replace the key resource-hungry mining (SHA-256) and digital signature (ECDSA) algorithms in classic blockchain PoW mining. A PCBChain protocol is proposed to modify the consensus for resource-constrained lightweight nodes with cooperation from resource-rich full nodes. We also analyze security issues at the component, protocol, and system levels. With this technology, we further discuss how PCBChain helps improve the security level in a GPS antispoofing application. Empirical results reveal significant reductions in hardware resources and power compared to the classic PoW, while achieving a better TPS rate with high scalability.

Acknowledgment

The views and opinions of the authors do not reflect those of the US DoD, Air For Force, and AFRL.

References

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, Sep. 2013.

[2] M. U. Farooq, M. Waseem, A. Khairi, and S. Mazhar, "A critical analysis on the security concerns of Internet of Things (IoT)," *Int. J. Comput. Appl.*, vol. 111, no. 7, pp. 1–6, Feb. 2015.

[3] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Tech. Rep., 2008.

[4] N. Kshetri, "Can blockchain strengthen the Internet of Things?" *IT Prof.*, vol. 19, no. 4, pp. 68–72, 2017.

[5] *Iota News Volkswagen Over the Air Update Poc With Iota at Cebit 2018*. [Online]. Available: https://iota-news.com/volkswagen-over-the-air-update-with-iota-at-cebit-2018

[6] A. Polianytsia, O. Starkova, and K. Herasymenko, "Survey of hardware IoT platforms," in *Proc. 3rd Int. Scientific-Practical Conf. Problems Infocommun. Sci. Technol. (PIC S&T)*, Oct. 2016, pp. 152–153.

[7] *Bitcoin Energy Consumption Index*. [Online]. Available: https://digiconomist.net/bitcoin-energy-consumption

[8] M. Pilkington, "11 blockchain technology: Principles and applications," in *Research Handbook on Digital Transformations*, vol. 225, 2016.

[9] A. Satoh and T. Inoue, "ASIC-hardware-focused comparison for hash functions MD5, RIPEMD-160, and SHS," *Integration*, vol. 40, no. 1, pp. 3–10, Jan. 2007.

[10] A. Abidi, B. Bouallegue, and F. Kahri, "Implementation of elliptic curve digital signature algorithm (ECDSA)," in *Proc. Global Summit Comput. Inf. Technol. (GSCIT)*, Jun. 2014, pp. 1–6.

[11] V. Sivaraman, H. H. Gharakheili, A. Vishwanath, R. Boreli, and O. Mehani, "Network-level security and privacy control for smart-home IoT devices," in *Proc. IEEE 11th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2015, pp. 163–167.

[12] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," *Future Gener. Comput. Syst.*, vol. 82, pp. 395–411, May 2018.

[13] U. Chatterjee, R. S. Chakraborty, and D. Mukhopadhyay, "A PUF-based secure communication protocol for IoT," *ACM Trans. Embedded Comput. Syst.*, vol. 16, no. 3, p. 67, 2017.

[14] X. Jia, Q. Feng, T. Fan, and Q. Lei, "RFID technology and its applications in Internet of Things (IoT)," in *Proc. 2nd Int. Conf. Consum. Electron., Commun. Netw. (CECNet)*, Apr. 2012, pp. 1282–1285.

[15] A. B. Orúe, L. H. Encinas, V. Fernández, and F. Montoya, "A review of cryptographically secure prngs in constrained devices for the IoT," in *Proc. Int. Joint Conf. SOCO CISIS ICEUTE León, Spain*. Springer, Sep. 2017, pp. 672–682, 2017.

[16] D. Chaum, C. Crépeau, and I. Damgard, "Multiparty unconditionally secure protocols," in *Proc. 20th Annu. ACM Symp. Theory Comput. STOC*, 1988, pp. 11–19.

[17] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *Proc. OSDI*, vol. 99. 1999, pp. 173–186.

[18] J. R. Douceur, "The sybil attack," in *Proc. Int. Workshop Peer Peer Syst.*, Springer, 2002, pp. 251–260.

[19] F. Zhang, I. Eyal, R. Escriva, A. Juels, and R. Van Renesse, "REM: Resource-efficient mining for blockchains," in *Proc. 26th USENIX Secur. Symp. (USENIX Secur.)*, 2017, pp. 1427–1444.

[20] V. Zamfir. (2015). *Introducing Casper 'the Friendly Ghost*. Ethereum Blog. [Online]. Available: https://blog.ethereum.org/2015/08/01/introducing-casper-friendly-ghost

[21] S. De Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone, "Pbft vs proof-of-authority: Applying the cap theorem to permissioned blockchain," Tech. Rep., 2018.

[22] K. Olson, M. Bowman, J. Mitchell, S. Amundson, D. Middleton, and C. Montgomery, "Sawtooth: An introduction," *Linux Found.*, to be published.

[23] A. Dorri, S. S. Kanhere, and R. Jurdak, "Blockchain in Internet of Things: Challenges and solutions," 2016, *arXiv:1608.05187*. [Online]. Available: http://arxiv.org/abs/1608.05187

[24] K. Rahim, H. Tahir, and N. Ikram, "Sensor based PUF IoT authentication model for a smart home with private blockchain," in *Proc. Int. Conf. Appl. Eng. Math. (ICAEM)*, Sep. 2018, pp. 102–108.

[25] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *Int. J. Web Grid Services*, vol. 14, no. 4, pp. 352–375, 2018.

[26] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, "A survey on the security of blockchain systems," *Future Gener. Comput. Syst.*, vol. 107, pp. 841–853, Jun. 2020.

[27] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, "A survey of distributed consensus protocols for blockchain networks," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 1432–1465, 2nd Quart., 2020.

[28] W. Yan, F. Tehranipoor, and J. A. Chandy, "A novel way to authenticate untrusted integrated circuits," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2015, pp. 132–138.

[29] J. Delvaux, D. Gu, D. Schellekens, and I. Verbauwhede, "Helper data algorithms for PUF-based key generation: Overview and analysis," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 6, pp. 889–902, Jun. 2015.

[30] P. Wortman, W. Yan, J. Chandy, and F. Tehranipoor, "P2M-based security model: Security enhancement using combined PUF and PRNG models for authenticating consumer electronic devices," *IET Comput. Digit. Techn.*, vol. 12, no. 6, pp. 289–296, Nov. 2018.

[31] W. Yan, F. Tehranipoor, and J. A. Chandy, "PUF-based fuzzy authentication without error correcting codes," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 36, no. 9, pp. 1445–1457, Sep. 2017.

[32] H. Krawczyk, "LFSR-based hashing and authentication," in *Proc. Annu. Int. Cryptol. Conf.*, Springer, 1994, pp. 129–139.

[33] F. Masoodi, S. Alam, and M. Bokhari, "An analysis of linear feedback shift registers in stream ciphers," *Int. J. Comput. Appl.*, vol. 46, no. 17, pp. 46–49, 2012.

[34] *7 Ser. FPGAs Configurable Log. Block User Guide Rev. 1.7*, Xilinx, San Jose, CA, USA, Nov. 2014.

[35] *Efficient Shift Registers, LFSR Counters, Long Pseudo-Random Sequence Generators Rev. 1.1*, Xilinx, San Jose, CA, USA, Jul. 1996.

[36] *Vivado Design Suite Tutorial: Implement. Rev. 2018.2*, Xilinx, San Jose, CA, USA, Jun. 2018.

[37] M. A. Orumiehchiha, J. Pieprzyk, R. Steinfeld, and H. Bartlett, "Security analysis of linearly filtered NLFSRs," *J. Math. Cryptol.*, vol. 7, no. 4, pp. 313–332, Dec. 2013.

[38] Q. Chen, G. Csaba, P. Lugli, U. Schlichtmann, and U. Ruhrmair, "The bistable ring PUF: A new architecture for strong physical unclonable functions," in *Proc. IEEE Int. Symp. Hardw.-Oriented Secur. Trust*, Jun. 2011, pp. 134–141.

[39] W. Yan, C. Jin, F. Tehranipoor, and J. A. Chandy, "Phase calibrated ring oscillator PUF design and implementation on FPGAs," in *Proc. 27th Int. Conf. Field Program. Log. Appl. (FPL)*, Sep. 2017, pp. 1–8.

[40] W. Yan, F. Tehranipoor, X. Zhang, and J. Chandy, "FLASH: FPGA locality-aware sensitive hash for nearest neighbor search and clustering application," in *Proc. 30th Int. Conf. Field Program. Log. Appl. (FPL)*, 2020.

[41] N. Beckmann and M. Potkonjak, "Hardware-based public-key cryptography with public physically unclonable functions," in *Proc. Int. Workshop Inf. Hiding*. Springer, 2009, pp. 206–220.

[42] G.-L. Feng and K. K. Tzeng, "A generalization of the berlekamp-massey algorithm for multisequence shift-register synthesis with applications to decoding cyclic codes," *IEEE Trans. Inf. Theory*, vol. 37, no. 5, pp. 1274–1287, Sep. 1991.

[43] L.-N. Lee and F. Hemmati, "Nonlinear random sequence generators," U.S. Patent 4 852 023, Jul. 25, 1989.

[44] C. E. Vivelid, "Nonlinear feedback shift registers and generating of binary de Bruijn sequences," M.S. thesis, Univ. Bergen, Bergen, Norway, 2016.

[45] S. K. Sanadhya and P. Sarkar, "New collision attacks against up to 24-step SHA-2," in *Proc. Int. Conf. Cryptol. India*, Springer, 2008, pp. 91–103.

[46] J. Cathalo, B. Libert, and J.-J. Quisquater, "Cryptanalysis of a verifiably committed signature scheme based on GPS and RSA," in *Proc. Int. Conf. Inf. Secur.*, Springer, 2004, pp. 52–60.

[47] U. R Ührmair, F. Sehnke, J. S Ölter, G. Dror, S. Devadas, and J. Ü. Schmidhuber, "Modeling attacks on physical unclonable functions," in *Proc. 17th ACM Conf. Comput. Commun. Secur. CCS*, 2010, pp. 237–249.

[48] P. H. Nguyen, D. P. Sahoo, C. Jin, K. Mahmood, U. Rührmair, and M. van Dijk, "The interpose PUF: Secure PUF design against state-of-the-art machine learning attacks," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, pp. 243–290, Aug. 2019.

[49] T. Rachwalik, J. Szmidt, R. Wicik, and J. Zabłocki, "Generation of nonlinear feedback shift registers with special-purpose hardware," in *Proc. Mil. Commun. Inf. Syst. Conf. (MCC)*, Aug. 2012, pp. 1–4.

[50] *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications Rev. 1a*, National Institute of Standards and Technology, Gaithersburg, MD, USA, Apr. 2010.

[51] M. Aydos, T. Yantk, and C. K. Koc, "A high-speed ECC-based wireless authentication on an ARM microprocessor," in *Proc. 16th Annu. Comput. Secur. Appl. Conf. (ACSAC)*, Dec. 2000, pp. 401–409.

[52] R. Xu and Y. Chen, "Microchain: A light hierarchical consensus protocol for IoT system," 2019, *arXiv:1912.10357*. [Online]. Available: http://arxiv.org/abs/1912.10357

[53] E. Deirmentzoglou, G. Papakyriakopoulos, and C. Patsakis, "A survey on long-range attacks for proof of stake protocols," *IEEE Access*, vol. 7, pp. 28712–28725, 2019.

Wei Yan (Member, IEEE) received the master's degree in electronic engineering from the University of Chinese Academy of Sciences, Beijing, China, in 2014, and the Ph.D. degree from the University of Connecticut, Storrs, CT, USA, in 2018.

He is currently an Assistant Professor with Clarkson University, Potsdam, NY, USA. Previously, he was a Postdoctoral Researcher at Washington University in St. Louis, St. Louis, MO, USA. His research interests are FPGA-based digital systems, hardware security, and blockchain. His projects include flash memory tester, solid-state PCIe cards, high-speed image collection, embedded flash translation layers, fault-tolerant physical unclonable functions, true random number generators, automatic electrical power system verification, lightweight blockchain on IoTs, GPS spoofing defender, and memory fingerprints.

Ning Zhang (Member, IEEE) received the Ph.D. degree from Virginia Tech, Blacksburg, VA, USA, in 2016.

He is currently an Assistant Professor with the Department of Computer Science and Engineering, Washington University in St. Louis, St. Louis, MO, USA. Before that, he was with an industry as a Cyber Engineer and Technical Lead for over ten years. His research focus is system security, which lies at the intersection of security, embedded systems, and computer architecture and software.

Laurent L. Njilla (Member, IEEE) received the B.S. degree in computer science from the University of Yaoundé-1, Yaoundé, Cameroon, the M.S. degree in computer engineering from the University of Central Florida (UCF), Orlando, FL, USA, in 2005, and the Ph.D. degree in electrical engineering from Florida International University (FIU), Miami, FL, USA, in 2015.

He joined the Cyber Assurance Branch, U.S. Air Force Research Laboratory (AFRL), Rome, NY, USA, as a Research Electronics Engineer in 2015. As a Basic Researcher, he is responsible for conducting and directing basic research in the areas of cyber defense, cyber physical systems, cyber resiliency, hardware security, and the application of game theory, category theory, and blockchain technology. He is the Program Manager of the Center of Excellence (CoE) in Cyber Security for the Historically Black Colleges and Universities and Minorities Institutions (HBCU/MI) and the Program Manager of the Disruptive Information Technology Program at AFRL/RI. He has coauthored over 70 peer-reviewed journal articles and conference papers with best paper awards. He is a co-inventor of two patents and three patent applications. He has co-edited two books *Blockchain for Distributed System Security* and *Modeling and Design of Secure Internet of Things* (Wiley–IEEE Press).

Dr. Njilla is a member of the National Society of Black Engineer (NSBE).

Xuan "Silvia" Zhang (Member, IEEE) received the B.Eng. degree in electrical engineering from Tsinghua University, Beijing, China, in 2006, and the M.S. and Ph.D. degrees in electrical and computer engineering from Cornell University, Ithaca, NY, USA, in 2009 and 2012, respectively.

She is currently an Assistant Professor with the Preston M. Green Department of Electrical and Systems Engineering, Washington University in St. Louis, St. Louis, MO, USA. She works across the fields of VLSI, computer architecture, and cyber physical systems. Her research interests include adaptive power and resource management for autonomous systems, hardware/software co-design for machine learning and artificial intelligence, and efficient computation and security primitives in analog and mixed-signal domains.

Dr. Zhang was a recipient of the NSF CAREER Award in 2020, the DATE Best Paper Award in 2019, and the ISLPED Design Contest Award in 2013, and her work has also been nominated for the Best Paper Award at DATE 2019 and DAC 2017.