Design of programmable hardware security modules for enhancing blockchain based security framework

Devika Kalathil Nandalal, Ramesh Bhakthavatchalu
Department of Electronics and Communication Engineering, Amrita Vishwa Vidyapeetham, Amritapuri, India

Article Info

Article history:

Received Aug 3, 2022
Revised Sep 15, 2022
Accepted Oct 1, 2022

Keywords:

Digital signature
Hardware security modules
Blockchain
Physical unclonable function
SHA-256

ABSTRACT

Globalization of the chip design and manufacturing industry has imposed significant threats to the hardware security of integrated circuits (ICs). It has made ICs more susceptible to various hardware attacks. Blockchain provides a trustworthy and distributed platform to store immutable records related to the evidence of intellectual property (IP) creation, authentication of provenance, and confidential data storage. However, blockchain encounters major security challenges due to its decentralized nature of ledgers that contain sensitive data. The research objective is to design a dedicated programmable hardware security modules scheme to safeguard and maintain sensitive information contained in the blockchain networks in the context of the IC supply chain. Thus, the blockchain framework could rely on the proposed hardware security modules and separate the entire cryptographic operations within the system as stand-alone hardware units. This work put forth a novel approach that could be considered and utilized to enhance blockchain security in real-time. The critical cryptographic components in blockchain secure hash algorithm-256 (SHA-256) and the elliptic curve digital signature algorithm are designed as separate entities to enhance the security of the blockchain framework. Physical unclonable functions are adopted to perform authentication of transactions in the blockchain. Relative comparison of designed modules with existing works clearly depicts the upper hand of the former in terms of performance parameters.

Corresponding Author:

Devika Kalathil Nandalal
Department of Electronics and Communication Engineering, Amrita Vishwa Vidyapeetham
Amritapuri, India
Email: devikanandalal@gmail.com

1. INTRODUCTION

The huge expense to fabricate complex integrated circuits (ICs) has urged top very large-scale integration (VLSI) companies to go fabless and rely on third-party intellectual properties (IPs), foundries, and testing facilities. Such globalization in the VLSI industry paved the way for serious security issues that emerge due to the varying entities involved starting from the design stage, up to its delivery to end users. Hardware security is thus becoming a serious concern in the design, manufacturing, and testing of ICs. IC supply chain is exposed to many forms of attacks such as IP theft, malicious trojan insertion, IC counterfeiting, overproduction, reverse engineering, and many more [1], [2]. Many IC protection schemes have been introduced to address security threats. But all existing security methods focus on a single problem rather than providing a unified solution to all the attacks prevailing in the supply chain. Thus, it is crucial to employ an integrated technique to safeguard the integrity and authenticity of ICs during varied phases of chip design to curb security breaches caused by fraudulent components.

Blockchain affirms a holistic solution in addressing all sorts of security issues and provides a tamper-resistant database for design protection. The idea of timestamped-based decentralized technology was first conceived by the scientist “Satoshi Nakamoto” to track, trace and authenticate bitcoin transactions. Bitcoin is a digital currency that relies on cryptographic techniques for the creation and authentication of transactions. In recent years most commercial applications entrust blockchain technology to perform safe and secure data sharing and broadcasting about anything of value (known as transactions in the blockchain). The shared and decentralized database maintenance coupled with the absence of third-party organizations to monitor and validate the data transactions add to its attractiveness.

Blockchain is encountered serious security challenges despite its decentralized immutable framework due to the reliance of this technology on software platforms and units such as wallets to store digital resources and secret keys. Furthermore, the involvement of online networks and the feeble security of apps, websites, and software on which the entire technology runs increases the probability of hacking. The security aspect of blockchain can be strengthened through the medium of efficient hardware security modules (HSM) that provide a safe environment to deal with sensitive data associated with the blockchain. HSMs bridge the inconsistency existing about the safe storage of digital assets and key parameters which are pre-requisite for efficient peer-to-peer shared communication. A few of the valuable characteristics of HSMs used for blockchain security solution includes key generation and storage, encryption, IP protection, signature authentication, and dedicated hardware memory. Such a kind of dedicated hardware element can allow for efficient management of information maintaining both data security and privacy, completely eradicating the dependency on online storage. Blockchain should readily rely on HSMs as the defensive channel for quantum-safe cryptographic operations to safeguard and protect sensitive information and code.

This work focuses on the design of efficient hardware elements inside HSMs to enhance the security feature of blockchain and effectively utilize this technology in the context of the IC supply chain. The components within the proposed HSM scheme include secure hash algorithm-256 (SHA-256), elliptic curve digital signature algorithm (ECDSA), hybrid physical unclonable functions (PUF), and hardware memory. HSMs act as a communicating module between the computing system (node) and the user to do all cryptographic computations as and when required. Thus, it relieves the blockchain framework from resource-intensive and exhausting operations and thus saves a significant amount of power and energy. Hardware encryptions also speed up the entire process when related to software equivalent. Each module is designed to be programmable and validated separately to work for any bit input. The research considers security at the cost of hardware overhead caused by these modules of HSMs upon blockchain techniques.

The remaining portion of the paper is organized as: section 2 elaborates on the background research done that paved way for the novel approach. HSM and different cryptographic elements are discussed in section 3. Section 4 illustrates the proposed HSM scheme combined with blockchain inclusive of designed cryptographic modules for HSM. Results of different modules implemented in field-programmable gate array (FPGA) are interpreted and analyzed in section 5. Section 6 concludes the paper together with future scope.

2. BACKGROUND RESEARCH

2.1. Threats in IC supply chain

The complexity of ICs has risen to a greater extent on account of the constant and vigorous scaling happening in the semiconductor industry. As per Moore’s law, since 1960 the transistor count in a single IC has been twice its number in every two years. Correspondingly chip cost has also diminished at the same pace. As a result, consumer devices, military infrastructures, and other critical systems started to utilize more electronic devices and ICs from the higher end to the lower end. The rapid growth in the semiconductor industry has in turn brought serious challenges in assuring the security and integrity of ICs [3]. The hardware devices were always presumed to be resistant to attacks disregarding the vulnerabilities associated with the integrated circuit supply chain.

An untrustworthy IC supply chain facilitates adversaries to insert cloned and counterfeit chips as authorized ones into the market. If the forged copies of ICs go undetected or prevented, it ends up in a system with potential security vulnerabilities. Furthermore, joint test action group (JTAG) which is a widely adopted standard to perform in-field debugging and IC testing can be exploited by the attackers to acquire IP ownership, design, or secret data hidden inside the core [4]–[6].

So far researchers have proposed numerous techniques to detect and prevent IP piracy, trojan insertion, reverse engineering, and counterfeiting of ICs that occurs during different stages of IC design. Unseemly, the suggested prevention/protection techniques focus to impede threats selectively and also to a limited extend. For instance, IP watermarking and fingerprinting help to claim design ownership and track illegal use of IPs respectively but fail to prevent reverse engineering and trojan insertion, while hardware-based logic and FSM-based locking/obfuscation effectively prevent the above threats by preserving the design functionality by locking it with a secret key [7], [8]. Similarly, IC metering methods control IC postfabrication and hinder overproduction by enabling chip activation using a confidential key that can be generated/accessed by the IP owner alone [9]. Nonetheless, they fail in offering a comprehensive solution to secure IC during various stages of its design. Most of the counterfeit prevention methods confide in the reliability of secret keys stored in tamper-proof memory. The existing techniques are unsuccessful to detect and record the illicit users who may corrupt the IP design during the design, fab, and testing phases due to the ignorance of the IP owner. Another critical affair is the lack of trusted and decentralized databases to record confidential information such as keys/passwords that can be accessed and verified by authorized users only amidst deceitful situations [10], [11].

2.2. Blockchain technology as the comprehensive solution

The ever-increasing and tenacious attempt to secure the integrity and authentication of data transmission has led to the evolution of the most prominent technology in the past decade known as blockchain [12]. The modern IC supply chain demands blockchain as the only solution that can comprehensively address all forms of security threats and provide a tamper-proof database to ensure the security of design IPs [11], [13]. Blockchain constitutes a chain of blocks wherein each block is interlinked by means of a cryptographic hash of the preceding block in association with a timestamp. All transactions within a block are hashed individually and later hashed in pairs until it computes to a single hash known as Merkle root hash. Besides hash functions, public key cryptosystems and cryptography conjointly act as the backbone of blockchain technology [14], [15]. The significance of blockchain is owed to the decentralized feature of ledgers that are shared among every participating node in the network. Blockchain keeps account of all transactions across different nodes (computers) with the goal to impede any changes to the database related to a specific block without the updating of subsequent blocks. In this scheme, each node maintains a copy of the ledger that contains all sensitive information related to the application recorded as transactions. Even with the distribution of ledgers to the public, the intensive encryption process involved safeguards the entire proceedings. The admirable properties of blockchain are the following:

- Immutability: any data once entered into the blockchain database is impervious to modifications.
- Decentralization: each participant in the blockchain network possesses a copy of the information database and is not held under a central authority.
- Transparency: everyone in the network can monitor all transactions meanwhile protecting the integrity of the user.

2.3. Vulnerabilities in blockchain security

In blockchain-based operations, each participant is provided with unique key pairs generated using ECDSA algorithms. Therefore, individuals who have possession of the secret key have complete ownership of his/her account. At the same time, most blockchain users make use of software platforms such as wallets to safely store secret keys and digital passwords. However, the security of such software depends on its ease of access. Even the hardware wallets, for instance, Trezor and Keepkey [16] claim to provide an enhanced level of security in holding keys and digital assets, still they are prone to fault injection attacks. In fault injection attacks, a fault is injected into the computing device that can hinder the execution of current instructions or even corrupt the data under operation. This may lead to key leakage compromising the security offered by hardware wallets [17], [18]. Once the key is stolen anyone could exploit the valuables for malevolent purposes despite the immutable secure database of blockchain. Bitfinex-bitcoin exchange company reported in 2015 and 2016 about the security breach they encountered wherein the private keys held by a multi-signature wallet were jeopardized. In the absence of HSMs, it is peculiarly insecure to rely on software and online sources for key generation. Storing of confidential information in the unprotected shared memory of computing systems certainly leads to a security breach. Even the storage of keys in bank vaults is found to be unreliable. Unfortunately, this is factual for every blockchain-related application.

2.4. Significance of hardware security modules

Hardware security modules are one kind of crypto processor that is particularly devoted to the safety and protection of crypto keys in its entire life cycle. It appears mostly in the form of plug-in cards that can be directly connected to a computer or server [19]. Tamper-resistant feature of HSMs enables secure management, execution, and storage of cryptographic keys for most security-aware organizations. In addition to key storage, it also carries out secure authentication, signature generation, encryption, decryption and other cryptographic functions [20]. The involvement of HSMs in the blockchain networks eliminates the need for the system’s software to have a copy of private keys and data in the shared memory of the web server. Entire cryptographic operations are performed within the boundaries of the HSM domain. Since web servers are used to perform various applications, it increases the risks of information leakage that cyber attackers could misuse. HSMs act as an interacting module between the participant and the node so that the user can provide inputs and receive outputs without accessing or interfering with internal components of the security device. This ensures that the data inside HSMs can never be extracted, manipulated, or withdrawn in their entire lifecycle.

3. HARDWARE SECURITY MODULES

In the business sector where data protection relies on key-dependent digital signatures and encryptions, protecting these private keys is significant. HSMs are solutions to these business problems. HSMs are available in various forms ranging from universal serial bus (USB) devices to plug-in cards to perform various cryptographic operations and key protection all through their life process.

3.1. Use-case of HSMs

The use of HSMs sustains to advance and enlarge into more industries and scenarios. For a long time, HSMs have been used to safeguard the secure socket layers (SSL) and transport layer security (TLS) encryption keys that allow for secure transactions over the web. Today HSMs are used in powering emerging use cases as in internet of things (IoT) devices, digital watermarking, and blockchain applications. HSMs are mainly used for generating strong credentials for IoT devices such as gaming consoles, interacting vehicles, and medical-related areas. The contents of video streaming are digitally watermarked and safeguarded against piracy. They are used as a strong foundation for building advanced blockchain applications. In blockchain, HSMs are utilized to protect the root keys to ensure the authenticity of participants and for generating signatures and providing a secure environment for performing cryptographic operations. It acts as a preventive measure against adversaries from stealing keys, accessing data, or faking the users' ID in business transactions.

3.2. Hardware security components

3.2.1. Hash functions

Hash functions are basically algorithms that perform mathematical operations on the input message. The output obtained after passing the plain text through hash is called a digest. In contrast, normal encryption hashes are irreversible in nature. No decryption key can convert a hash digest back to its original message. Secure hash algorithms are a family of hash functions passed as Federal Information Processing Standard (FIPS) that includes SHA-0, SHA-1, SHA-2, and SHA-3. SHA-256 belongs to the SHA-2 family that produces 256-bit value as a digest. Hash algorithms are formulated to satisfy two criteria: It is infeasible for an attacker to retrieve the original message from a given hash output. It is impossible for an attacker to generate two messages with identical hash values. Even a single bit change should trigger a significant change in the digest produced. In blockchain, each block is chained to the previous blocks using SHA-256 hash pointers that contain the hash of all the data inside the preceding block. It is also used for mining purposes in the blockchain.

3.2.2. Digital signatures

Digital signatures are analogous to traditional hand-written signatures in many aspects but properly implemented digital signatures are more difficult to forge than handwritten ones. Digital signatures come under the FIPS standard which was proposed in 1991 and got standardized in 1994. Even though they work based on the public key infrastructure unlike it, digital signatures use the private key for encryption and the public key for decryption. Digital signature algorithm (DSA) makes use of a public key for authenticating the signature, but the authorization process is much more complicated when compared with Rivest, Shamir, and Adleman (RSA). DSA also provides three benefits that include message authentication, integrity verification, and non-repudiation. Figure 1 displays a typical digital signature algorithm.

In DSA the input message is passed on to a hash function where the digest is generated is passed on to a signing function. The signing function also has other parameters like global variable G, random variable K, and the private key of the sender (Pk). The two outputs obtained from the signing function are called r and s. On the receiver end, we pass the plaintext to the same hashing function to regenerate the digest. It is passed on to the verification function again depending on G, K, and Pk in addition to r and s received from the sender. The value generated by the function is then compared to r if they match then the verification process is complete and data integrity is verified.

3.2.3. Physical unclonable functions

A physical unclonable function (PUF) is a function that is based on certain physical systems that produce random values as output. The PUF output known as a response to the input challenge is unpredictable even for attackers with physical access to the system. The manufacturing variations of a design during the fabrication process are uncontrollable and unpredictable due to which no two identical circuits can have the same electrical and delay parameters. Thus, it is impossible to clone or reproduce another copy of the same physical system despite knowing the functionality of a path. They are different categories of PUF based on the source of physical randomness that includes delay-based PUFs such as arbiter PUF and ring oscillator PUF while memory-based PUFs comprise of static random-access memory-based PUF (SRAM PUF), butterfly PUF, latch PUF [21], [22]. PUF has a wide variety of applications as mentioned:

- Device authentication: Since the same PUF circuitry generates unique PUF responses on different chips, this property can be utilized for authenticating devices.
- Key generation and storage: Unique responses generated using PUF structures during run-time are more secure than keys stored in memory.
- IP protection: PUF can be combined with finite state machines (FSM) to perform active IC metering.
- Device authentication: Device can be authenticated with the help of distinct challenge-response pairs created for each design.

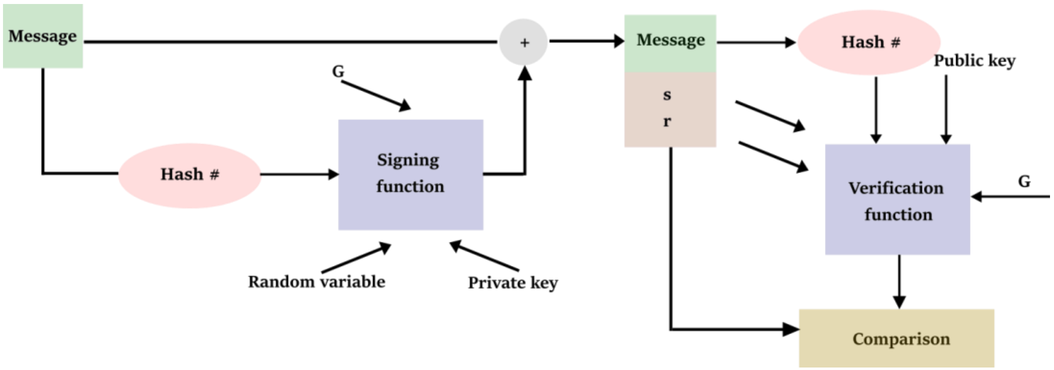


Figure 1. Typical digital signature algorithm

4. PROPOSED HARDWARE SECURITY MODULE SCHEME INTEGRATED WITH BLOCKCHAIN

This work introduces a unique idea of integrating HSM with blockchain to protect on-chip cores against malicious JTAG invasions, and IP threats and to track IP distribution. It enhances the security aspect of blockchain by acting as an interface between the decentralized framework and the user. The entire data processing phase in the blockchain is performed using a dedicated hardware system comprising of hash functions, PUF, and digital signatures. The blockchain ledger of the computing node keeps track of all the information computed by HSM. All these data are stored and processed by the proposed HSM and communicated with the blockchain as and when needed. However, secret keys and passwords are stored exclusively by the designed HSM. In this structure, except for the blockchain database, all cryptographic components such as hashing function SHA-256, crypto module, ECDSA keypair generation, and PUF modules are customized and designed to be programmable to take in any bit-size input. Figure 2 represents the proposed HSM scheme that can be adopted for enhancing blockchain security in the context of the IC supply chain.

4.1. Working of HSM with blockchain

In HSM, the ECDSA module generates private-public key pairs for each participant in the network. This public key acts as the enroll ID for the users involved in the IC supply chain. Algorithm 1 portrays the IP registration phase while Algorithm 2 illustrates the IC access and verification mechanism.

IP verification unit stores a local database that holds the information details pertaining to each IC such as enroll ID, IC ID, and challenge-response pairs of PUFs. Enroll ID lets the IP owner know which participant is currently seeking entry to acquire IP. IC ID uniquely identifies each IC enrolled by the designer. CRPs are also unique, and it is associated with the PUF of each IC. When the communicating node sends the enroll ID or other credentials, they are verified by the verification unit, and the corresponding PUF challenge is sent to the crypto module. This module consists of PUF and SHA-256 modules. PUF is indeed a hybrid combination of arbiter and butterflies PUF with high reliability and security. The response generated by PUF for the given challenge goes as input to hashing unit to render the hash ID of the device (HashID).

A typical blockchain database contains a hash of all transactions, Merkle root hash, block header, and timestamp. This database will be maintained by each node, but the hash computations needed to produce all these data elements are done through HSM and it is updated in the local ledger of each node. To access the sensitive data stored in the blockchain database, the participant should have knowledge about the block to which a particular IC belong and its equivalent header value. The header value, block number, and other metadata related to the block are known only to the trusted participants within the blockchain network. Thus, it ensures that only authorized users can access information confidential for an IC. If the correct values of block ID and block header are entered, then the hash of chip ID stored in the hardware memory is compared with the hash computed by the crypto module from the PUF response. Block header input is also checked for a match with the computed header value. If a match is found Hash ID is retrieved. Once the following values are provided enroll ID is cross-checked with the passcode given for users such as system integrators or testers to identify the purpose.

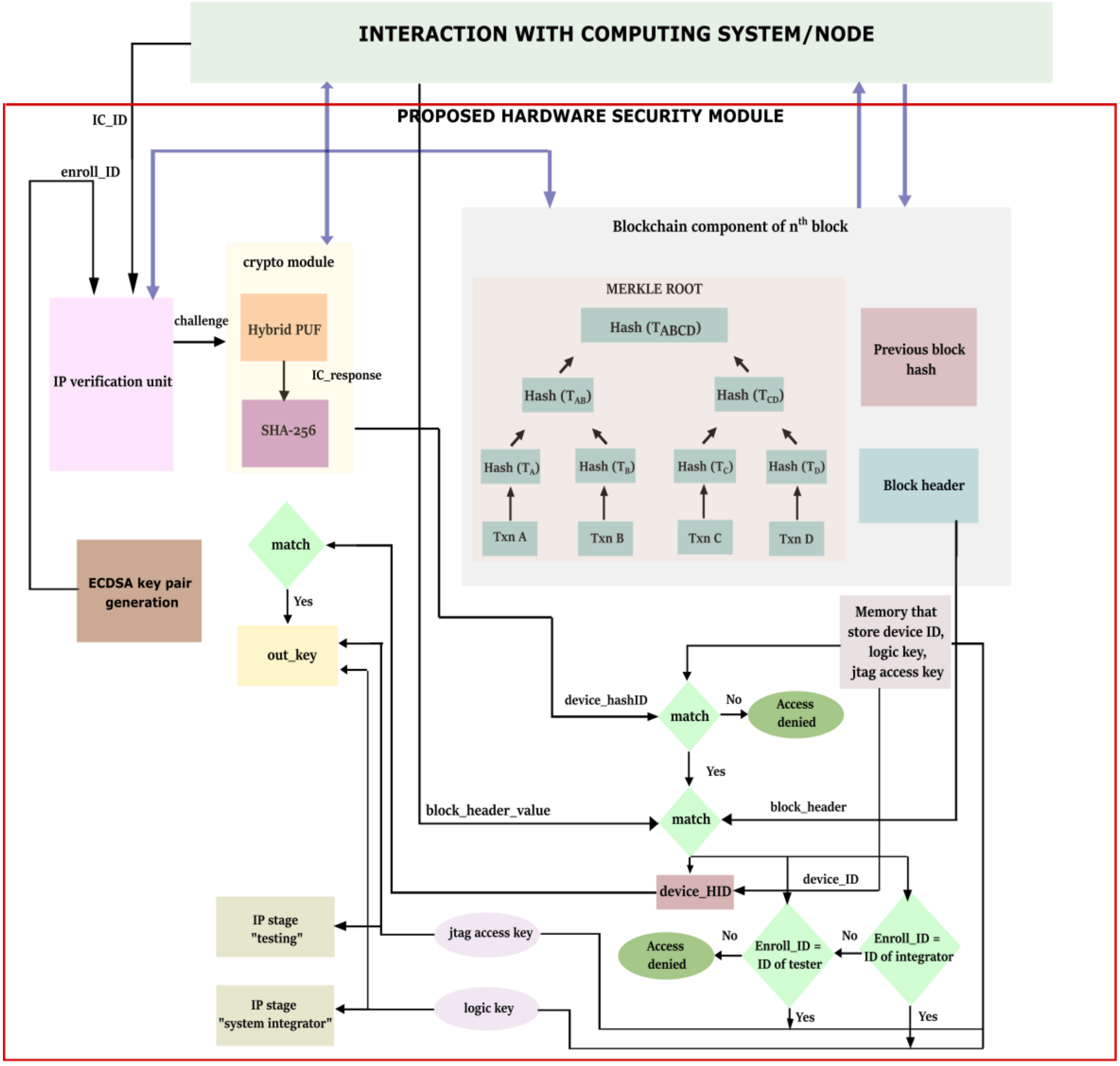


Figure 2. Proposed HSM scheme along with blockchain node

Algorithm 1. Pseudocode for IP registration phase

```
Input: Enroll ID, Device ID
$PUF_{ch_i}$ ← PUF Challenge input for $IC_i$
  If Current participant is not design owner Then
    Access denied and error message is shown
End
Else
    IC details added to the memory of verification unit
      $Verification\_unit.participantID = Enroll\ ID$
      $Verification\_unit.IC\_ID = Device\ ID$
      $Verification_{unit}.PUF_{ch_i} = unique\ value$
End
```

Algorithm 2. Pseudocode for IP/IC access and verification mechanism

```
Input: Enroll ID, Device ID
$PUF_{ch_i} \leftarrow$ PUF Challenge input for $IC_i$
  $Hash_{ID_i} \leftarrow$ hash ID of the IC
  $Device_{ID_i} \leftarrow$ hash ID of the IC stored in memory
  $BH_{ID_i} \leftarrow$ Block header of the IC
  $BH_{mem} \leftarrow$ Block header of the IC stored in memory
  If current participant is not system integrator or tester Then
   Access denied and error message is shown
End
Else
   If input IDs matches with corresponding IC details Then
     $PUF_{ch_i}$ is selected and given as input to crypto-module and $Hash_{ID_i}$ is generated
       Level 1 security accessed
        If $Hash_{ID_i} = Device_{ID_i}$ Then
          Level 2 security accessed
           If $BH_{ID_i} = BH_{mem}$ Then
             Level 3 security accessed
              If $Enroll\ ID = Tester_{ID}$ stored in memory Then
                JTAG access key is provided to the participant IC stage is updated as
                testing
              End
              If $Enroll\ ID = Integrator_{ID}$ stored in memory Then
                Logic key is provided to the participant IC stage is updated as integration
              End
           End
           Else
               Access denied
           End
       End
       Else
          Access denied
       End
   End
   Else
       Access denied
   End
End
```

Thus, the correct access key can be provided from HSM memory. The memory of HSM also holds passwords and keys for accessing different segments of an IC. If enroll ID matches with the passcode of the system integrator block identifies the current user as a system integrator and entrusts with a logic key and the status of the IP stage is changed to “system integrator”. The logic key helps to unlock the IC functionality that is obfuscated for security reasons. If enroll ID is validated to be from the test engineer, the tag access key is provided with the current IP status changed to “tester”. In this manner, the IP owner can track and trace all IP designed by him and protect the designs from IP theft, counterfeit, and trojan insertion.

The framework ensures multi-level security with the aid of a verification unit, PUF, and passcode provided to each intermediate user of the supply chain. PUF challenge-response pairs associated with each IC are utilized to do the verification process in place of ECDSA. The PUF module embedded within the IC should produce the same challenge-response pairs as the one stored in the local database of each node. The equivalence between the two CRP values confirms the chip authenticity and identifies the IC exclusively. The designed HSMs act as a medium for communication and processing of confidential data between the computing device and blockchain network.

4.2. SHA-256

Secure hash algorithm SHA 256 is the most widely used hashing function in blockchain technology, particularly in bitcoin. SHA-256 is preferred over other hashes on account of its high collision resistance coupled with the trade-off between computational expense and security. SHA-256 is used to hash every transaction, for computing Merkle root hash, and block header and to link each block with the predecessor using block header hash. Since this algorithm forms the cryptographic backbone in this distributed technology, the hash function should be implemented to compute the hash digest for any input entered. This enables the same SHA-256 design to be adopted in each phase that requires hashing process [23].

SHA-256 is designed to accept any N-bit message and generate fixed 256-bit hash digest. Programmability is the unique feature of this hash architecture and message block converts the random size plaintext into 512-bit blocks through the process of padding and segregation. The block diagram of SHA-256 is portrayed in Figure 3.

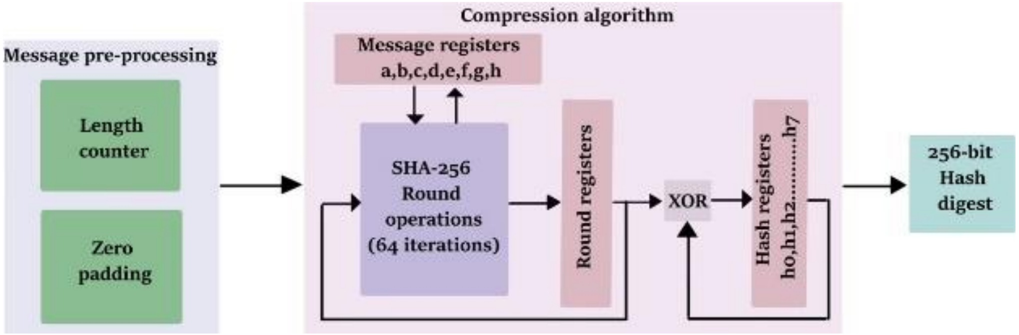

Message pre-processing
Length counter
Zero padding
Compression algorithm
Message registers a,b,c,d,e,f,g,h
SHA-256 Round operations (64 iterations)
Round registers
XOR
Hash registers h0,h1,h2........h7
256-bit Hash digest

Figure 3. Block diagram of SHA-256

Each 512-bit block thereafter goes through 64 rounds of operation as mentioned in the compression function of the sha-256 algorithm. The updated values in 32-bit message register a, b, c, d, e, f, g, h modify 32-bit hash registers H0, H1, H2, H3, H4, H5, H6, H7 for every i^{th} 512-bit block. The algorithm is performed for all message blocks with modified values of round registers to get the final 256-bit hash output.

4.3. Elliptic curve digital signature algorithm

Elliptic curve digital signature algorithm (ECDSA) is the most popular digital signature algorithm amid other security techniques that utilizes elliptic curve cryptography in generating key pairs for the process of signing and verification of any information. The powerful and unbreakable cryptographic framework combined with speedy computations done by ECDSA makes them popular among the existing signature schemes [24]. High efficiency and greater security offered by ECDSA at reduced key size have increased its recognition in the era of blockchain technology. It is the private key and public key generated by this asymmetric key algorithm that acts as the password and addresses respectively for each transaction in the blockchain.

Various complex mathematical functions involved with the Elliptic curve digital signature process comprise modular multiplication, multiplicative inverse, point addition, point doubling, and point multiplication. All these modules are separately designed and implemented to enhance the effectiveness of the overall structure. Interleaved modular multiplication is adopted to implement multiplication operations while the extended Euclidean algorithm does the work of the inversion function. The ECDSA architecture is also made to take in any bit value and provide the appropriate output as a signature or for verification. The proposed HSM makes use of ECDSA to generate key pairs in order that the public key computed will act as the enroll ID of the participants. The verification process is completely taken over by the PUF component present in HSM.

4.4. Hybrid PUF

PUF is a phenomenal hardware primitive that can be applied for secret key generation and authentication that provides a unique device fingerprint based on the given input challenge. The unique ID generated by PUF is as per the random uncontrollable variations that occur during the IC manufacturing process. The challenge-response pair produced by PUF is specific for each IC instance and is unclonable. Usually, secret keys are embedded in non-volatile memories such as electrically erasable programmable read-only memory (EEPROM). But the presence of these storage devices adds to the manufacturing complexity. It also becomes easier for attackers to extract confidential data from such memory elements. PUF on the contrary generates key value at the time of ciphering so that the key is neither stored inside the memory nor could be leaked by the attackers. The secret value once used will be deleted after the encryption phase. For blockchain-related applications, lightweight PUF can replace resource-demanding ECDSA to do the job of authentication and verification and the same has been adopted in the proposed scheme to strengthen blockchain.

A typical arbiter PUF encounters routing differences due to the asymmetric crisscross paths of the multiplexer pairs and the intricate routes to the arbiter flip-flop. This increases the static delay component of the PUF when compared to the random delay factor. So, the response of arbiter PUF will be static in nature by virtue of which the value can be anticipated by the attackers. This drawback of arbiter PUF is overpowered by the new PUF structure which is a hybrid form of butterfly and arbiter PUF. The two different kinds of PUF are combined here to improve the randomness, uniqueness, and reliability of the arbiter PUF element. Butterfly PUF component enables the design to be used in any type of FPGA and is stable under any environmental condition with negligible fabrication cost. The PUF is also designed to be programmable to be flexible enough to encrypt any N-bit data to the same bit-length output. This unique ID created can never be reverse-engineered or cloned by adversaries. The minimal structure of PUF also contributes to a reduction in area and power overhead since in this setup authentication process is done by PUF instead of ECDSA. Figure 4. demonstrates the architecture of a 1-bit hybrid PUF for an input challenge of 2-bits. This 1-bit hybrid PUF response circuit is replicated ‘N’ times to generate an N-bit response for an M-bit challenge.

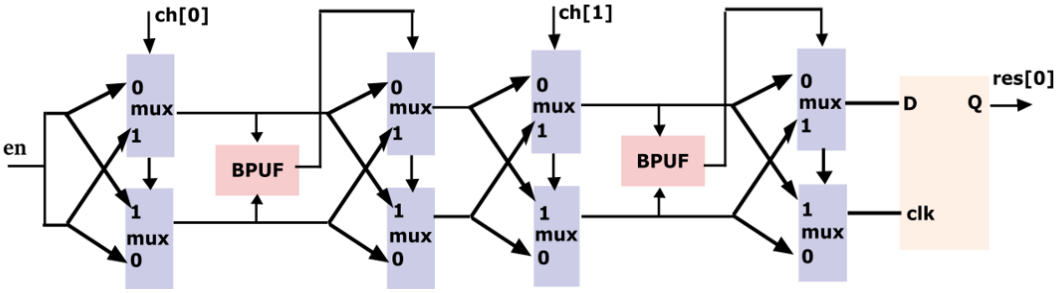


Figure 4. Hybrid PUF architecture for 2-bit challenge

4.4.1. PUF metric evaluation

- Uniqueness: The ability to distinguish two PUF instances uniquely is called uniqueness. The efficacy of uniqueness is measured by calculating the inter-hamming distance (Inter HD). Hamming distance between two equal bit-length binary values evaluates the count of positions at which the equivalent bits differ. The average inter-hamming distance among ‘m’ chips is given as (1) [25]:

  $$InterHD = \frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \frac{HD(R_i(N),Rj(n))}{} \quad (1)$$

  where $R_i(n)$ and $R_j(n)$ are the responses of i^{th} and j^{th} instances for the same challenge. For ideal cases, the value of inter HD will be 50%.
- Reliability: Reliability measures the consistency of each PUF instance to generate the same response under different external conditions such as temperature variations and changes in supply voltage. Intra-Hamming distance is calculated to estimate reliability. The average intra HD of an i^{th} chip for j different conditions is given by (2) [25].

  $$IntraHD = \frac{1}{m} \sum_{j=1}^{m} \frac{HD(R_i(n),R_{i,j}(n))}{n} \quad (2)$$

  $$Reliability + 100\% - IntraHD \quad (3)$$

- Uniformity: uniformity determines the symmetry of 0’s and 1’s in a PUF response. It indicates the randomness of the response, and the optimal value is 50%. Hamming weight (HW) is calculated to measure the uniformity and it is formulated as [25]:

  $$HW = \frac{1}{n} \sum_{j=1}^{i} ri,j$$

  where $r_{i.j}$ corresponds to the j^{th} bit of i^{th} chip’s response.

5. RESULTS AND ANALYSIS

This section discusses the implemented designs of cryptographic elements such as SHA-256, ECDSA, and PUF that form the components in the HSM proposed. The feature of reconfigurability is added to each design to make them flexible enough to work for any bit values. The research also puts in efforts to improve the design performances of the suggested security modules in comparison to the existing works in similar hardware primitives. To serve this purpose, a comparative analysis of the FPGA implementation of the proposed designs with other related works is performed. All the designs are implemented using the Xilinx Vivado tool in Virtex 7 FPGA and the functionality is verified using the ModelSim simulator. PUF designs are simulated using the LT-spice tool to get a unique response based on changes in the properties of the PUF instance. MATLAB tool is also used in the research work to plot the values to validate ECDSA modules and also for validating PUF metrics.

Programmable SHA-256 structure is fed with varying size inputs to check on the variation in area utility with an increase in bit-length. Table 1 highlights the resource utilization, speed, and power usage of different bit-size message inputs to programmable SHA-256 circuits. The results exhibit that for the rise of bit-length from 8 to 448 there occurs merely a 1.4% and 0.64% increase in LUT and register utilization respectively for a slight decrease in speed. The same structure is designed to be flexible enough to do encryption as well as to perform blockchain mining processes by just varying the input parameter alone, and the results depict that the design works for any input value.

Table 1. Comparative analysis of performance parameters of SHA-256 design for varying input size

Input size	#Slice LUTs	#Slice Registers	#IOBs	LUT-FF pairs	Speed (MHz)	Dynamic Power (mW)
8	7029	2156	267	1906	176.978	224
16	7135	2204	275	1956	176.679	248
32	7234	2262	291	1990	176.678	267
48	7338	2344	307	2066	176.356	289
64	7391	2381	323	2101	175.538	328
128	7553	2445	387	2167	175.711	350
256	7844	2573	515	2287	175.830	397
300	7965	2617	559	2346	175.828	415
448	14444	4549	707	4198	171.863	556

The prevailing works on SHA-256 have used different FPGA devices to implement hash algorithms for better efficiency but none of the designs have been added with nor discussed the flexibility in converting any bit-input to fixed digest. Table 2 shows the relative comparison of existing hash designs with the proposed SHA design for the same FPGA implementation. The results show that the latter displayed superior performance in terms of speed with moderate area utilization.

Table 2. Comparative analysis of proposed SHA-256 with other existing designs

Designs	Platform	Frequency (MHz)	Area (Slices)
Proposed design	Virtex-7	175.583	525
[26]	Virtex-5	79	391
[27]	Virtex-5	64.45	139
[28]	Virtex-3	83	1322
[29]	Virtex-3	88	1261

ECDSA architecture has been designed to improve the adaptability of each subcomponent in this algorithm such as modular multiplication, inversion, or point multiplication. Every module is designed to receive N-bit input and provide M-bit output. Figure 5 put on view the MATLAB plot of ECDSA verification validated for about 10 different messages of 24-bit size. The ECDSA output after FPGA implementation was cross-checked with MATLAB output to confirm the proper execution of the design code. Since the ECDSA module is concerned with key pair generation in the proposed HSM scheme the design to generate key pairs are implemented in FPGA to analyze the performance parameters. Table 3 manifests the comparative analysis of the ECDSA key-pair module with existent works in the same domain. The results clearly depict the trade-off achieved between area and speed relative to the cited works.

A new hybrid form of PUF architecture has been put forth to be applied in HSM design. So, the PUF is designed at the transistor level using the LT-spice simulation tool to generate different PUF instances. The PUF was designed to take in the 4-bit challenge and give out an 8-bit response. Table 4 exhibits the response obtained from 20 PUF instances for the same 4-bit challenge value ‘0001’. Transistor parameters such as the channel length (L), channel width (W), and threshold voltage (Vth) were minimally varied in each PUF instance to bring in the effect of uncontrollable manufacturing variations.

The uniqueness metric is calculated based on the responses obtained from all 20 instances of PUF. The uniqueness factor gained by the hybrid structure of PUF is around 49.02% and the uniformity component is estimated to be around 55%. Figures 6(a) and 6(b) display the intra-Hamming distance and uniformity of the proposed PUF respectively.

Hybrid PUF is relatively compared with other existing works in silicon PUF designs such as arbiter PUF, ring oscillator (RO) PUF, butterfly PUF, and SRAM PUF. Table 5 displays a comparative analysis of having a higher uniqueness factor that is close to 50% in contrast to normal arbiter PUF or ring oscillator (RO PUF). The uniqueness of the suggested PUF with other delay-based and memory-based PUFs. The table reveals that the new architecture value acquired is almost equivalent to that of memory based PUFs. Therefore, hybrid PUF falls under the category of strong PUFs since it can generate a large number of CRPs but at the same time is able to achieve uniqueness close to weak PUFs. Results clearly depict that hybrid PUF offers an improved performance than typical arbiter and other kinds of PUF. Table 6 shows the FPGA implementation of PUF for varying bit responses when a fixed 8-bit challenge is provided as input. These outcomes indicate that hybrid PUF is suitable for FPGA implementation with a balance in respect of speed, power, and area contrary to arbiter PUF.

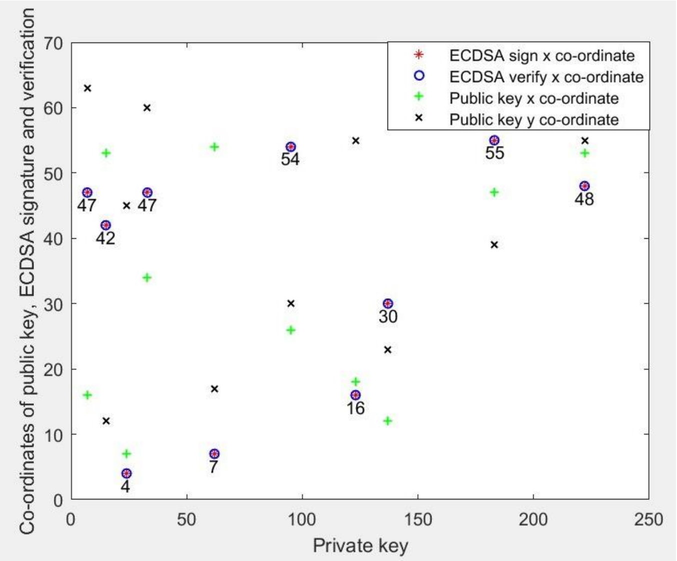


Figure 5. Validation of output obtained from ECDSA design using MATLAB

Table 3. Comparative analysis of proposed ECDSA key-pair module with other existing designs

Work	Platform	Reported Area (slices)	Frequency (Mhz)
Proposed ECDSA	Virtex-7	6030	136.77
[30]	Virtex-7	8900	177.7
[31]	Virtex-7	24200	72.9
[32]	Virtex-5	10200	66.7
[33]	Virtex-4	35700	70

Table 4. Response obtained for various PUF instances through LTspice simulation

PUF instance	Response
PUF1	11011010
PUF2	01010011
PUF3	11001101
PUF4	00101000
PUF5	11101011
PUF6	01010111
PUF7	01110010
PUF8	11000001
PUF9	10110111
PUF10	11111110
PUF11	11011011
PUF12	11001001
PUF13	10001111
PUF14	10111001
PUF15	01111010
PUF16	10011101
PUF17	01110011
PUF18	01011110
PUF19	01101010
PUF20	11101100

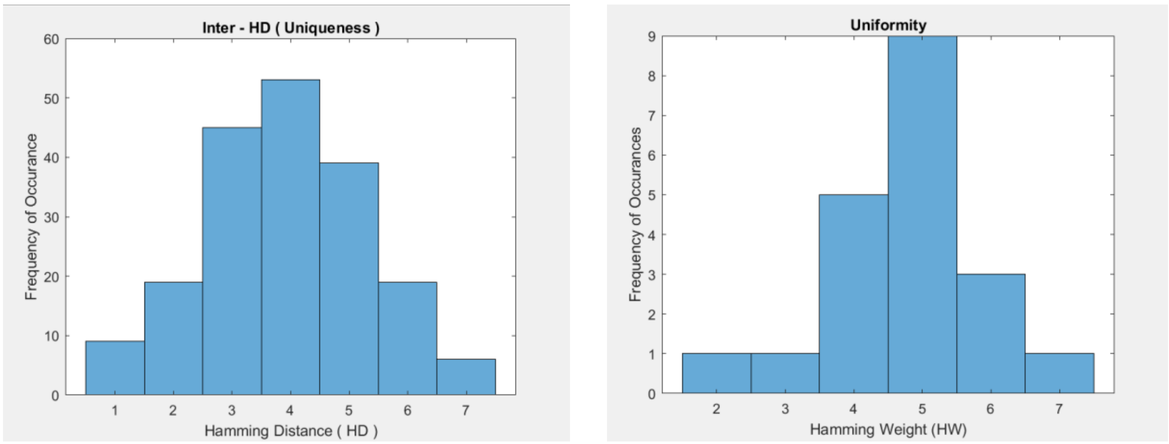
Table 5. Comparative analysis of proposed PUF with existing PUF designs

PUF type	Simulation/FPGA/Silicon	Technology	#PUF instance	Uniqueness (%)	Uniformity (%)
Arbiter [34]	Silicon	180 nm	37	23	NR
Arbiter [35]	Silicon	45 nm	40	38.9	NR
RO [36]	Simulation	90 nm	100	46.5	NR
SRAM [37]	FPGA	NR	17	49.97	≈ 50
Butterfly PUF (BPUF) [38]	FPGA	NR	36	50	≈ 60
Proposed design	Simulation	50 nm	20	49.02	55

NR=Not Reported

Table 6. Hybrid PUF implementation results for various bit-length response

Bit-size	8	16	32	64	128
#LUTs	167	354	706	1792	3573
#Registers	136	272	544	1088	2176
#IOBs	34	58	106	202	394
Speed (MHz)	257.73	246.37	219.78	209.48	208.73
Dynamic power (mW)	12	23	45	108	216



(a) (b)

Figure 6. PUF characteristics (a) intra Hamming distance of various generated responses and (b) uniformity of various generated responses

6. CONCLUSION AND FUTURE SCOPE

This research work focuses on the design of reconfigurable HSMs to supplement the blockchain cryptographic operations and thereby make this decentralized technology impervious to any form of security attack. The hardware security unit coupled with distributed public ledgers assures an advanced blockchain security system that is both reliable and cost-effective to be enforced for any security-oriented applications. The novel technique suggested tracking and controlling the sharing of IP from its creation in addition to the security against JTAG and IP attacks. This helps the IP owner to trace if any malicious modification, IP theft, or trojan insertion occurs during the entire lifetime of the IC. This work put forth an approach to utilize this ingenious technology in the VLSI domain with the integration of HSMs to safeguard the rights of an IP owner and aids him in the efficient management of sensitive information. Confidential cryptographic design for blockchain enables users to have ultimate authority over their hardware and provides a better-decentralized hardware solution that consumers could entrust. The proposed architecture is discussed as a miniature model for lower-order bit- size. However, this can be generalized and extended to an industrial rank so that it could be applied to all blockchain applications where the security of data transactions is of critical concern.

REFERENCES

[1] H. Salmani, T. Hoque, S. Bhunia, M. Yasin, J. J. Rajendran, and N. Karimi, “Special session: countering IP security threats in supply chain,” in *2019 IEEE 37th VLSI Test Symposium (VTS)*, Apr. 2019, pp. 1–9, doi: 10.1109/VTS.2019.8758633.

[2] J. Knechtel, S. Patnaik, and O. Sinanoglu, “Protect your chip design intellectual property,” in *Proceedings of the International Conference on Omni-Layer Intelligent Systems*, May 2019, pp. 211–216, doi: 10.1145/3312614.3312657.

[3] G. Bloom, E. Leontie, B. Narahari, and R. Simha, “Hardware and security,” in *Handbook on Securing Cyber-Physical Critical Infrastructure*, Elsevier, 2012, pp. 305–331.

[4] X. Ren, V. G. Tavares, and R. D. (Shawn) Blanton, “Detection of illegitimate access to JTAG via statistical learning in chip,” in *Design, Automation and Test in Europe Conference and Exhibition (DATE), 2015*, 2015, pp. 109–114, doi: 10.7873/DATE.2015.0558.

[5] K. Rosenfeld and R. Karri, “Attacks and defenses for JTAG,” *IEEE Design and Test of Computers*, vol. 27, no. 1, pp. 36–47, Jan. 2010, doi: 10.1109/MDT.2010.9.

[6] R. Bhakthavatchalu, S. K. Kannan, and M. Nirmala Devi, “Verilog design of programmable JTAG controller for digital VLSI IC’s,” *Indian Journal of Science and Technology*, vol. 8, no. 17, Aug. 2015, doi: 10.17485/ijst/2015/v8i17/62664.

[7] K. Shamsi, M. Li, K. Plaks, S. Fazzari, D. Z. Pan, and Y. Jin, “IP protection and supply chain security through logic obfuscation,” *ACM Transactions on Design Automation of Electronic Systems*, vol. 24, no. 6, pp. 1–36, Nov. 2019, doi: 10.1145/3342099.

[8] A. Baumgarten, A. Tyagi, and J. Zambreno, “Preventing IC piracy using reconfigurable logic barriers,” *IEEE Design and Test of Computers*, vol. 27, no. 1, pp. 66–75, Jan. 2010, doi: 10.1109/MDT.2010.24.

[9] F. Koushanfar, “Provably secure active IC metering techniques for piracy avoidance and digital rights management,” *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 1, pp. 51–63, Feb. 2012, doi: 10.1109/TIFS.2011.2163307.

[10] A. Kulkarni, N. A. Hazari, and M. Niamat, “A blockchain technology approach for the security and trust of the IC supply chain,” in *2019 IEEE National Aerospace and Electronics Conference (NAECON)*, Jul. 2019, pp. 249–252, doi: 10.1109/NAECON46414.2019.9058027.

[11] X. Xu, F. Rahman, B. Shakya, A. Vassilev, D. Forte, and M. Tehranipoor, “Electronics supply chain integrity enabled by blockchain,” *ACM Transactions on Design Automation of Electronic Systems*, vol. 24, no. 3, pp. 1–25, Jun. 2019, doi: 10.1145/3315571.

[12] B. K. Mohanta, D. Jena, S. S. Panda, and S. Sobhanayak, “Blockchain technology: A survey on applications and security privacy Challenges,” *Internet of Things*, vol. 8, Dec. 2019, doi: 10.1016/j.iot.2019.100107.

[13] P. Cui, J. Dixon, U. Guin, and D. Dimase, “A blockchain-based framework for supply chain provenance,” *IEEE Access*, vol. 7, pp. 157113–157125, 2019, doi: 10.1109/ACCESS.2019.2949951.

[14] W. Yan, N. Zhang, L. L. Njilla, and X. Zhang, “PCBChain: lightweight reconfigurable blockchain primitives for secure IoT applications,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 10, pp. 2196–2209, Oct. 2020, doi: 10.1109/TVLSI.2020.3014155.

[15] S. P. Mohanty, V. P. Yanambaka, E. Kougianos, and D. Puthal, “PUFchain: A hardware-assisted blockchain for sustainable simultaneous device and data security in the internet of everything (IoE),” *IEEE Consumer Electronics Magazine*, vol. 9, no. 2, pp. 8–16, Mar. 2020, doi: 10.1109/MCE.2019.2953758.

[16] Tomshwom, “Lessons from the Trezor Hack,” Steemit, 2017. https://steemit.com/bitcoin/@tomshwom/lessons-from-the-trezor-hack (accessed on January 10, 2022).

[17] J. Redman, “A Def Con 25 demonstration claims to ‘Break bitcoin hardware wallets,’” *Bitcoin.com*, 2017. https://news.bitcoin.com/def-con-25-demonstration-break-bitcoin-hardware-wallets/ (accessed Jan. 10, 2022).

[18] J. Redman, “Small Ethereum clones getting attacked by mysterious “51 Crew”,” *Bitcoin.com*, 2016. https://news.bitcoin.com/ethereum-clones-susceptible-51-attacks/ (accessed Jan. 09, 2022).

[19] Utimaco, “Why HSM is vital to the blockchain technologies,” *Utimaco*, 2020. https://utimaco.com/current-topics/blog/why-hsm-vital-blockchain-technologies (accessed Jan. 10, 2022).

[20] C. Crane, “What is a hardware security module? HSMs explained,” *hashedout*, 2021. https://www.thesslstore.com/blog/what-is-a-hardware-security-module-hsms-explained/ (accessed Jan. 10, 2022).

[21] S. Khan *et al.*, “Utilizing manufacturing variations to design a tri-state flip-flop PUF for IoT security applications,” *Analog Integrated Circuits and Signal Processing*, vol. 103, no. 3, pp. 477–492, Jun. 2020, doi: 10.1007/s10470-020-01642-9.

[22] S. Morozov, A. Maiti, and P. Schaumont, “An analysis of delay based PUF implementations on FPGA,” *International Symposium on Applied Reconfigurable Computing*, vol. 382–387, 2010.

[23] K. N. Devika and R. Bhakthavatchalu, “Parameterizable FPGA implementation of SHA-256 using blockchain concept,” in *2019 International Conference on Communication and Signal Processing (ICCSP)*, Apr. 2019, pp. 370–374, doi: 10.1109/ICCSP.2019.8698069.

[24] D. K. N and R. Bhakthavatchalu, “Efficient hardware prototype of ECDSA modules for blockchain applications,” *Telecommunication Computing Electronics and Control (TELKOMNIKA)*, vol. 19, no. 5, Oct. 2021, doi: 10.12928/telkomnika.v19i5.19416.

[25] A. Maiti, I. Kim, and P. Schaumont, “A robust physical unclonable function with enhanced challenge-response set,” *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 1, pp. 333–345, Feb. 2012, doi: 10.1109/TIFS.2011.2165540.

[26] F. Kahri, B. Bouallegue, M. Machhout, and R. Tourki, “An FPGA implementation and comparison of the SHA-256 and Blake-256,” in *14th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering-STA’2013*, Dec. 2013, pp. 152–157, doi: 10.1109/STA.2013.6783122.

[27] N. C. Iyer and S. Mandal, “Implementation of SHA-256 algorithm in FPGA based processor,” *International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)*, vol. 4, pp. 334–344, 2015.

[28] L. Bai and S. Li, “VLSI implementation of high-speed SHA-256,” in *2009 IEEE 8th International Conference on ASIC*, Oct. 2009, pp. 131–134, doi: 10.1109/ASICON.2009.5351591.

[29] K. K. Ting, S. C. L. Yuen, K. H. Lee, and P. H. W. Leong, “An FPGA based SHA-256 processor,” in *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2002, pp. 577–585.

[30] M. M. Islam, M. S. Hossain, M. K. Hasan, M. Shahjalal, and Y. M. Jang, “FPGA implementation of high-speed area-efficient processor for elliptic curve point multiplication over prime field,” *IEEE Access*, vol. 7, pp. 178811–178826, 2019, doi: 10.1109/ACCESS.2019.2958491.

[31] S. Asif, M. S. Hossain, and Y. Kong, “High-throughput multi-key elliptic curve cryptosystem based on residue number system,” *IET Computers and Digital Techniques*, vol. 11, no. 5, pp. 165–172, Sep. 2017, doi: 10.1049/iet-cdt.2016.0141.

[32] H. Marzouqi, M. Al-Qutayri, and K. Salah, “An FPGA implementation of NIST 256 prime field ECC processor,” in *2013 IEEE 20th International Conference on Electronics, Circuits, and Systems (ICECS)*, Dec. 2013, pp. 493–496, doi: 10.1109/ICECS.2013.6815461.

[33] K. Javeed and X. Wang, “FPGA based high speed SPA resistant elliptic curve scalar multiplier architecture,” *International Journal of Reconfigurable Computing*, pp. 1–10, 2016, doi: 10.1155/2016/6371403.

[34] J. W. Lee, Daihyun Lim, B. Gassend, G. E. Suh, M. van Dijk, and S. Devadas, "A technique to build a secret key in integrated circuits for identification and authentication applications," in *2004 Symposium on VLSI Circuits. Digest of Technical Papers (IEEE Cat. No.04CH37525)*, 2004, pp. 176–179, doi: 10.1109/VLSIC.2004.1346548.

[35] L. Lin, S. Srivathsa, D. K. Krishnappa, P. Shabadi, and W. Burleson, "Design and validation of arbiter-based PUFs for Sub-45-nm low-power security applications," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 4, pp. 1394–1403, Aug. 2012, doi: 10.1109/TIFS.2012.2195174.

[36] M. T. Rahman, F. Rahman, D. Forte, and M. Tehranipoor, "An aging-resistant RO-PUF for reliable key generation," *IEEE Transactions on Emerging Topics in Computing*, pp. 1–1, 2015, doi: 10.1109/TETC.2015.2474741.

[37] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, "FPGA intrinsic PUFs and their use for IP protection," in *Cryptographic Hardware and Embedded Systems-CHES 2007*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 63–80.

[38] S. S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen, and P. Tuyls, "Extended abstract: The butterfly PUF protecting IP on every FPGA," in *2008 IEEE International Workshop on Hardware-Oriented Security and Trust*, Jun. 2008, pp. 67–70, doi: 10.1109/HST.2008.4559053.

BIOGRAPHIES OF AUTHORS

Devika Kalathil Nandalal is a research scholar in Electronics and Communication department at Amrita School of Engineering with a Master of Engineering degree in VLSI design from Amrita university, Kollam, India (2017). She received her B. Tech degree in electronics and communication engineering from MG University in 2015. Her research interests include blockchain concepts, hardware security, cryptography, VLSI testing, field programmable gate arrays (FPGA), digital system design, and logic synthesis. She can be contacted at devikanandalal@gmail.com.

Ramesh Bhakthavatchalu is currently working as a professor in the Department of Electronics and Communication Engineering at Amrita Vishwa Vidyapeetham since 2005. He completed his M.E in applied electronics from College of Engineering, Guindy. In 2016, he received his Ph.D. degree in electronics and communication engineering from Amrita University. His current research interests include hardware security, automatic test pattern generation (ATPG), LBIST, cryptography, logic testing, VLSI testing, formal verification, signal processing, logic design, and field programmable gate arrays. He has more than 10 years of experience in core VLSI industries across the world. He worked as a design engineer at Arasan Chip Systems Inc. for 4 years and as an application engineer at SysTest Technologies, Inc. for 2 years. He is an active reviewer of IEEE access journals and other scientific publications. He can be contacted at rameshb@am.amrita.edu.