



Lightweight signature scheme to protect intellectual properties of Internet of things applications in system on chip field-programmable gate arrays

Kokila JAGADEESH*, Ramasubramanian NATARAJAN

Department of Computer Science and Engineering, National Institute of Technology, Tiruchirappalli, India

Received: 13.11.2018

Accepted/Published Online: 14.04.2019

Final Version: 18.09.2019

Abstract: Billions of smart objects in the edge devices offer advanced connectivity to networks which increase the security and complexity of the Internet of things (IoT) applications. To make such entities smarter heterogeneous intellectual property (IP) cores from multiple service providers are reused in system on chip platform. Enabling both chip and IP protection at post fabrication level is imperative. The IoT-based IP cores are signed with the hybrid physical unclonable function and finite state machine model to protect from cloning, misuse, unauthorized user access, and physical attacks. The extended finite-state machine is used to verify the signature, which reduces the space and time complexity. The goal is to design a secure plug-and-play system which supports IoT IP core for the unique chip without using any complex algorithms and huge memory storage. Experimental results show that the average overhead values of area and power are 1.05% and 1.08% less, respectively, compared with the existing IP protection.

Key words: System on chip field-programmable gate arrays, heterogeneous intellectual property cores, physical unclonable function, finite state machine, intellectual property protection

1. Introduction

System on chip field-programmable gate arrays (SoC FPGAs) are embedded with programmable software and customized hardware feature of a processor, which offers high levels of flexibility, reliability, and system performance [1]. The era of automation is filled with the Internet of things (IoT), which connects smart entities such as electronic devices, sensor, software, and actuators with the Internet to process and manage the huge amount of information. The IoT edge applications can be realized on application-specific integrated circuits (ASICs), FPGA, central processing units or any combination of the above-mentioned technology to design an IoT device. The major application of IoT is in designing the smart entities such as city, building, academics, machine control, autonomous transportation, medical science, real-time systems, and computing technologies [2]. The image processing and green computing application are implemented in SoC FPGA for satisfying IoT requirements; such efforts have been studied in a few articles [3, 4]. End users want thinner electronic devices, which are also more advanced, cheaper, powerful, easy to use, and capable of performing heterogeneous activities. The designers have to put more effort to design such a complex device using the latest software and hardware. The gap between the end user and the designer is filled by reusing optimized design, component, and verified software and hardware intellectual property (IP) cores. A semiconductor IP core is a reusable programmable logic unit or layout design or a cell. The IoT device requires designing advanced IoT IP cores, all included in one platform. These cores need to be flexible for all designs, and should be in sync with the ever-increasing

*Correspondence: jk.cse09@gmail.com

time-to-market pressure [5]. The number of researches, which focus on integrating IPs into the design with and without the license and protecting it from an unauthorized user and theft, is increasing. The recently used IP protection mechanisms are watermarking technique, fingerprinting mechanisms, embedded signature, active metering, computational forensic engineering, and patent. Almost all of these techniques started to use physically unclonable functions–finite-state machines (PUF–FSM) structure, because of its advantages and security, and advancement as mentioned in [6] and related works. Hence, we proposed a hybrid PUF and FSM to generate a signature to protect the device and IPs together, which will provide enhancements in security and reliability to PUF using FSM as considered in the [7, 8] respectively. The contributions of this work are as follows:

1. The proposed hardware signature scheme is the first implementation in SoC FPGA, which combines the two different PUF and FSM to protect IoT IPs. The framework in the proposed model is implemented completely on hardware that has features like less storage and power for signature generation and verification of IoT IP cores.
2. The signature generation modules use a hybrid model that contains arbiter (APUF) [9], butterfly PUF (BPUF) [10] along with FSM to select the unique key, private key and to calculate the signature.
3. The signature generated by the hybrid PUF and FSM for each IP is hard to clone and misuse, and it is resilient to physical attacks. The proposed hardware signature is not easily verifiable by a simple FSM; therefore, the EFSM is used in the verification phase.
4. This scheme needs to employ a strong PUF for device authentication and weak PUF for generation of keys, so that the computation can be balanced in terms of area and power.
5. The IoT application demands heterogeneous IP cores in edge computing level, so it is appropriate to compare any hardware signature scheme with the recent benchmarks. Therefore, the lightweight hardware signature generated is compared with the benchmark IPs like ISCAS, ITC, and DSP.

2. Related works

The pressure on the rising market of chip development is due to the extensive use of reusable IP cores in designing and verification phases. Designing an add-on IP core for any advanced application needs several years of research, implementation, and testing. There will be a good number of competitors in the IP core design. All these reasons make the IP core protection, infringement, and security against recent threats much more vital. For the past 5 years, many IP protection mechanisms have been proposed to support the reuse of IP cores with some limitations which is discussed in this section. The widely used mechanisms in SoC to protect hard and soft IP cores include obfuscated function and circuits, bit-stream encryption, watermarking, fingerprinting, IP metering, and computational forensic engineering [11, 12]. A watermarking scheme has been proposed to protect the IPs from unauthorized user even after IP cores are integrated and packed in a single die [13]. The aim of this article is to distinguish fraudulent IPs from the original ones and to reduce the false acceptance ratio to the minimal. An FSM-based watermarking scheme [14] has been proposed for a sequential circuit and it deals with increased resilience towards state deletion attack, copy detection, and design overhead declination. A new method to embed watermarking [15] in soft IP core was proposed for embedded systems and it is a sequential aware one. The simulation results have been analyzed for Xilinx Virtex-II Pro FPGA board. The main limitation of this method is sequence length. The hard IP cores are dynamically

watermarked with ultralow power and are an easily detectable feature of the technique. An optimized scan design in which dummy scan cells can be introduced is used for observation. This method also determines the tradeoff between design overhead and toughness of the watermark [16]. A dynamic fingerprinting method can be used in ASIC, SoC, and FPGA to identify whether an IP core is legal and to find whether the user and the owner are one and the same. This article discusses the previously unreported fingerprinting for sequential circuits which generate an unknown ownership watermark, individually permitted by each user through a blind signature protocol. Approximately 100 fingerprints were created and tested with International Symposium on Circuits and Systems (ISCAS) benchmark circuits and nominal results were obtained [17]. Watermarking and fingerprinting are the best methods to protect and trace IP cores and each has its own pros and cons based on the use case. Watermarking is better for protecting and locating specific IP cores, whereas fingerprinting is better for identifying IP core as a whole. A two-level FSM for PUF is proposed to correct erroneous bits generated by environmental variations (e.g., temperature, voltage, and aging variations) [7]. This eliminates the need for a database and makes the authentication mechanism more stable. In this article, a novel IP protection mechanism to restrict the execution of an IP only on specific FPGA devices has been proposed [8]. The pay-per-device licensing mechanism is adopted in this work. This enables the system developers to purchase IPs from the core vendors at the low price based on usage instead of paying the expensive unlimited IP license fees. PUF, customized for FPGAs, are embedded into already enrolled FPGA device by the FPGA vendors. Augmented FSM are embedded into the original IPs by the IP vendors such that the FSM can be activated by the PUF responses from the FPGA device. Security vulnerabilities of this PUF-FSM binding method is studied. The IP cores have to be authenticated to plug-and-play in any SoC device to enhance its capacity. As discussed in this article, there are many IP protection mechanisms, one of them being signature-based authentication [18]. This article presents a conflictless implementation and verification using two distinct signatures in reconfigurable scan test architecture. Security is very much essential for every customer's electronic device that will be dynamically updated each year. The IP core has to be adopted as a plug-and-play circuit, but it is not endorsed due to the presence of different vendors for ASIC, SoC, and FPGA IP cores. The demand for these reusable cores is increasing tremendously and the cost for usage is also very high. The IP core protection and infringement in the recent SoC FPGA has become more complex due to challenges such as scalability and reprogrammability along with low power and area. The PUFs are classified as strong and weak based on their applications [19]. According to a detailed study about PUFs, which has been analyzed and implemented in different technologies and applications, have its own ability to withstand or succumb to attacks. There was no concept of hybrid PUF along with the FSM, in the past, for heterogeneous IP cores. This motivated us to select the heterogeneous hybrid PUFs and FSM, which are suitable for generating hardware signature. The design is also unique because it uses FSM for signature generation and EFSM for verification. This model has been compared and tested with different benchmarks to analyze its individuality.

3. Proposed Lightweight hardware signature scheme

The system architecture, shown in Figure 1, consists of the IoT application with the proposed hardware signature scheme. The IoT application have to be activated in an edge device by different types of IPs. Starting from the IoT physical layer to application, the devices, protocols, networks, gateways, security, and so on use different services such as sensing, monitoring, power management, audio or video processing, hardware accelerator at each point for processing data. Each service has to be incorporated into a single die with lesser area, lower power, and lower cost. The IP core is a logical function module that is designed using hardware language for

activating software services on a device. In this scheme, the heterogeneous IP cores used in IoT devices are termed as ADD-ON IP cores or IoT IP cores, which is a combination of hardware, software, or firmware. Hence, plenty of such IP cores are available for the new blooming application to support advancement in technology in the market today. The IP core integration and protection is very much essential in single-die flexible cores.

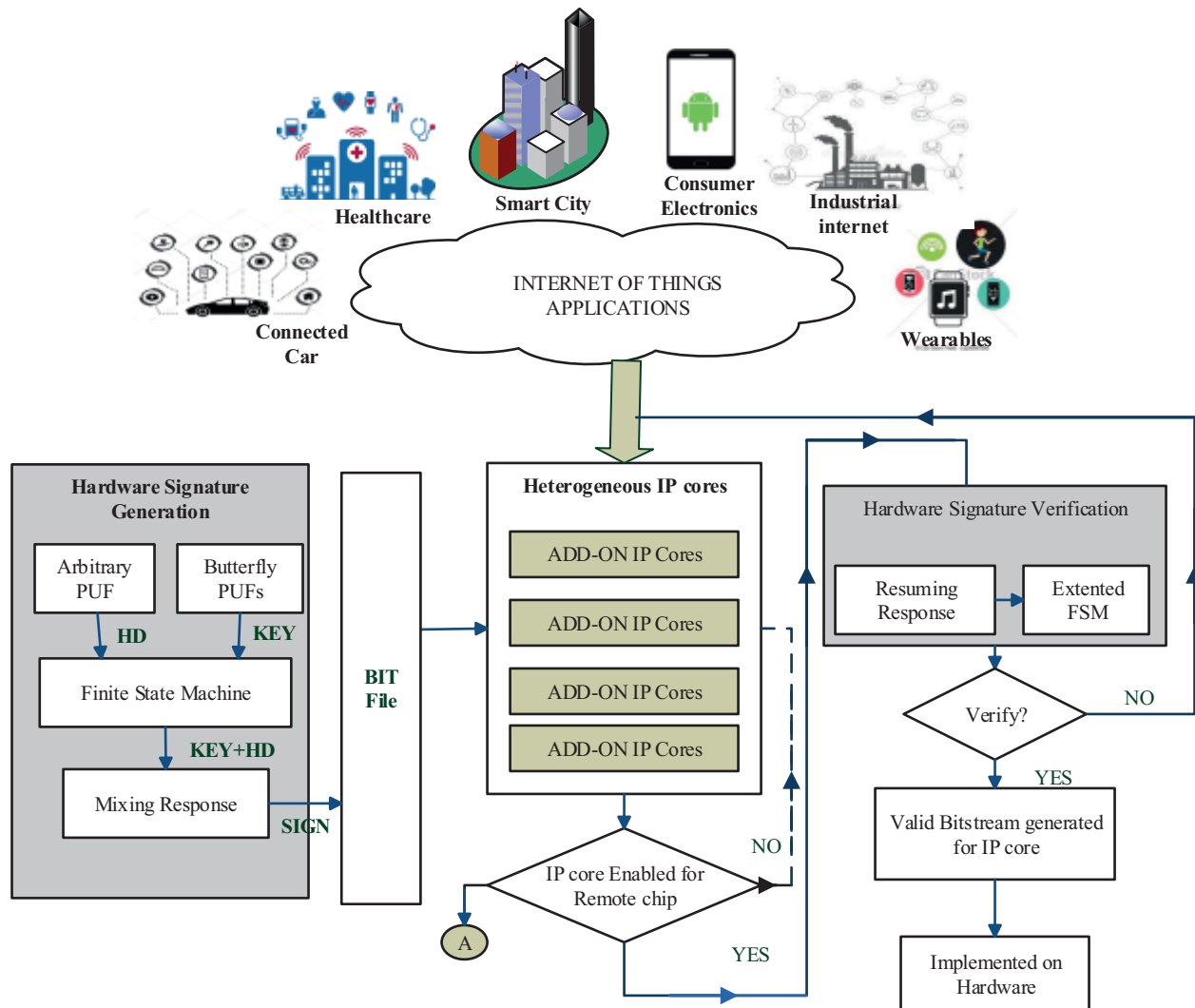


Figure 1. Overall system architecture

The applications of IoT are spread across multiple areas such as connected car, healthcare, smart cities, consumer electronics, industrial Internet, wearables, and many more. The supporting service for IoT device is converted to packable IP cores and pay-per-license is charged from each vendor. This results in high cost and long integration time. Moreover, a number of security issues, such as chip counterfeit, cloning, and antitampering mechanism, that are difficult to deal with arise. The proposed hybrid PUF and FSM models suggest a hardware signature to protect and use securely the IoT-based IP cores in the local and remote chips that utilize less area, power and time. The proposed system includes a high speed processor, programmable logic with IP cores such as hardware signature generation, heterogenous IP core which supports IoT, and verification module at local or remote systems to verify the IP cores. All these are implemented on a SoC FPGA platform.

3.1. General working principle of the proposed scheme

The signature is created using pure hardware circuits, and it has a set of features that allow the authentication of the sender and the integrity of the information to be verified. The main features of digital signatures are authentication, integrity, and nonrepudiation, which have to be achieved in any signature-based proposal [20, 21]. A pair of private and public key is used for generation and verification of hardware signatures as shown in Figure 2. As an example to explain the proposal for a small IoT device, 16-bit values are chosen for hybrid PUF and FSM. The 16-bit unique key generated by APUF circuit is hashed using the calculated hamming distance (HD) to a 4-bit HD to maintain the reliability. This HD between an APUF CRP serves as one of the inputs to the FSM. The 16-bit BPUF is used to generate a public key and the input is given to the FSM. The FSM takes the HD and the public key as inputs to make the transition to the next state. To maintain confidentiality, mixing operations such as Xor and shifting are performed in the FSM to produce a hardware signature. The signature is added to the bit stream (BIT) file and each vendor will have their own format and passcode to handle the BIT file. The passcode consists of information about the challenge (C) and time variable (V). In the verification phase, BIT file is executed using C which is supplied as the passcode to the user to extract the original hardware signature (HS) at run time. These HS and V are fed in the form of input to the extended FSM, which in turn is used to make the transition from one state to another and check whether it is valid or not. If the IP core is an authentic one, then it is allowed to execute on the platform. If it is not authentic, then the control is moved to identify the next IoT-based IP core from the stack and the entire process is repeated again based on the user requirement. The internal functioning of the generation and verification phase is explained in the next section.

3.2. Hardware signature generation

The signature generation concept used in the proposal is similar to the digital signature standard [22]. A unique key for the admin chip, which is random and never reused, is generated by APUF. For each IP core, the unique key is generated with less consumption of area and power. The signature pairs are generated with APUF unique key and BPUF public key. The 16-bit APUF response is the unique key that has to be hashed to 4-bit HD so that it is not used or seen at transit. The above case is considered to explain the proposed scheme, but the number of bits can be increased up to 256 and hashing can be more complex too. The HD is mapped for each 16-bit CRP, which may be a single value or limited range. The BPUF public key is used to identify a set of IP cores for each chipset. The steps given in algorithm 1 are followed to create the signature. The input to the algorithm is HD and KEY which are calculated from APUFs CRP and BPUF response respectively. An FSM is designed to balance the strong and weak properties of each PUF. A three-level FSM is used in this model which will produce the output hardware signature sign (HD, KEY). The FSM is used to fix the two conditions (C1 and C2), as specified in the algorithm, for limiting the dynamic values of HD and KEY. The first level of the FSM is triggered by HD of the unique key. The analysis to set the HD is made between different CRPs on the APUF for more than 10,000 iterations. The second level of the FSM is in transit if the KEY is right for the selected HD. The 16-bit BPUF will yield a minimum of 16-bit public keys, which are unique for each IoT-based IP core. The third level of the FSM is the accepting state, which will generate a hardware signature. The signature uses a simple logical combination of HD and KEY, as shown in Figure 3. Each IoT-based IP core can be selected from the network, and hardware signature can be encapsulated using the register insertion method.

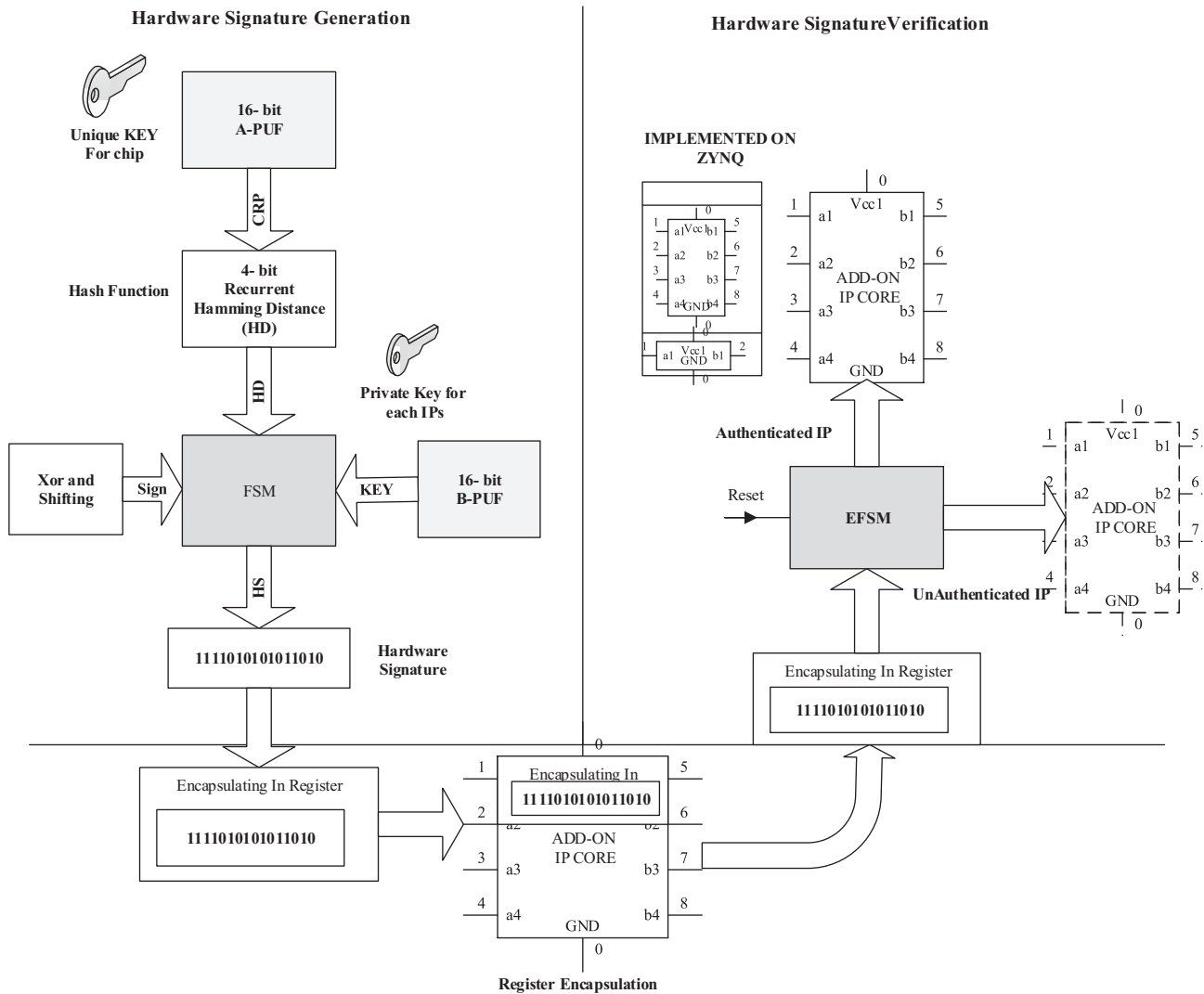


Figure 2. Working of the hardware signature scheme.

3.3. Hardware signature verification

The EFSM is mainly used to reduce the number of states in FSM and is well suited for sequential circuit verification. The verification is made robust by adopting EFSM as depicted in Figure 4. The EFSM [23] consists of less number of states ranging from the starting state to the intermediate state (E0), authenticated state (A), not authenticated state (UA); input and output variables; a transition function guarded by G, and update functions. The steps involved in EFSM are depicted in algorithm 2. The input and the output of the algorithm are signed IP core with public key and the final state respectively. The signature from ADD-ON IP core is extracted and inputted to EFSM for verification. The EFSM will change the state from starting to E0 or UA based on the RESET signal and HD. The E0 state checks the content of the register REG and a time limit is set for each HD based on APUF analysis. The Out1 and Out2 are the two variables used as the update function where simple logical operations such as Xor and shifting, based on HD, have been assumed. The EFSM will make a transition from E0 to A when the update function updates the value of Out2, which is equal to the

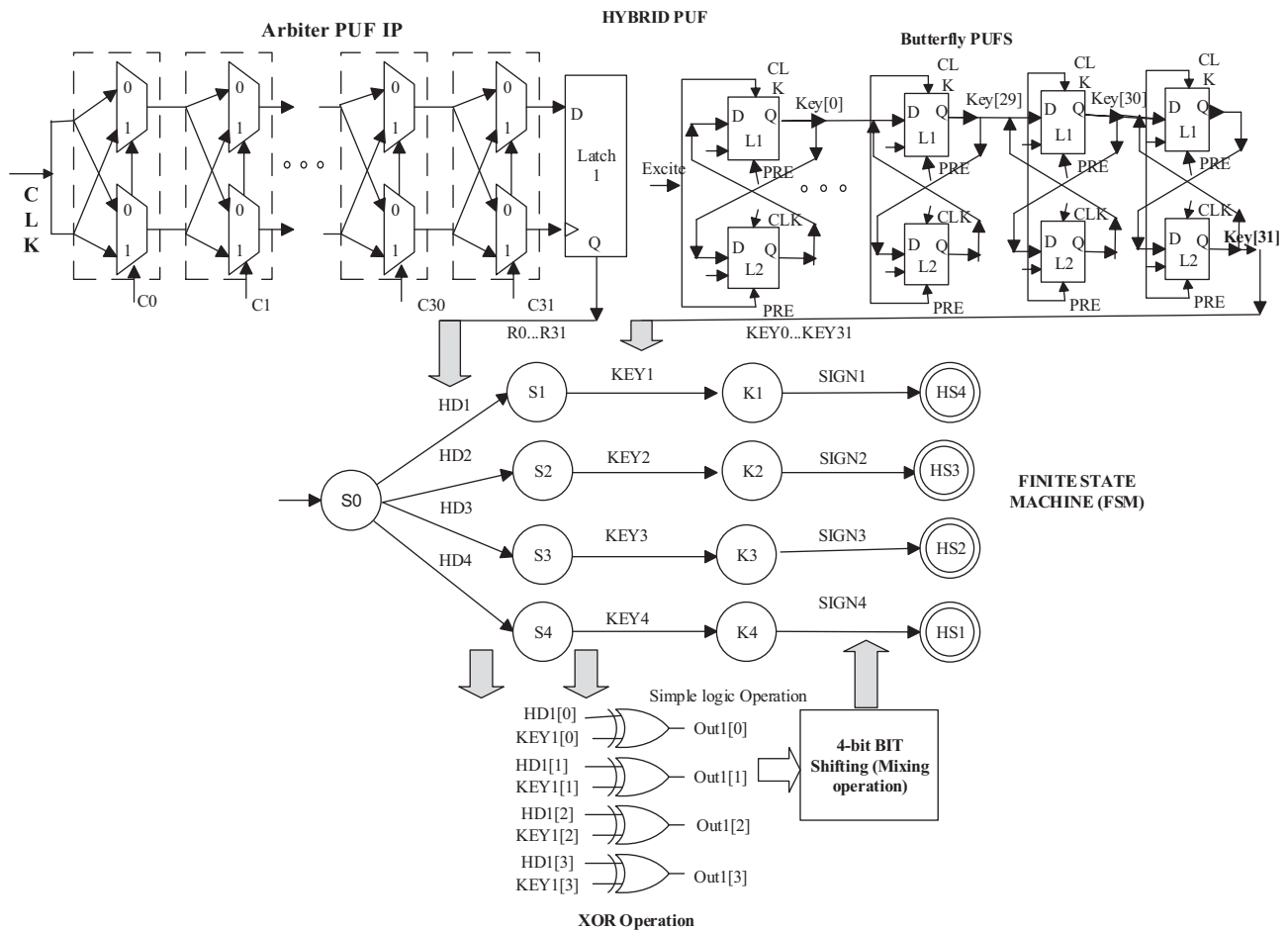


Figure 3. Hardware signature generation phase with hybrid PUFs and FSM.

public key set by BPUF within the time limit T . The UA state is reached from E0 when the Out2 is not equal to a public key and the time limit expires. The guard condition in this EFSM is designed based on the time limit and KEY value, hence t can be adjusted by the end-user chipset. The design is more restricted based on the KEY and T . The designer can modify the time limit based on the local and remote authentication.

4. Security analysis

The proposed scheme uses a PUF, which is a security primitive for chip and IP protection. The security features of strong and weak PUFs are combined with an FSM to generate HS, embedded in the initial register and verified by EFSM. The security properties of the hybrid PUF and FSM are summarized as follows:

1. The proposed HS exclusively authenticates the IP core for unique chip for which it is designed. The peculiar property of FSM and EFSM is that it dynamically fixes the states based on HD and KEY values, which prevents any attacks to illegally use the IP cores.
2. There is a possibility for the hacker to obtain the PUF response and KEY from other chip, but this information is not sufficient to obtain the IP cores. Usage of IP cores require authentication of the user,

Algorithm 1: Hardware signature generation.

Input: APUF(C, R), BPUF (KEY), IP core

Output: Sign(HD, KEY) + IP Core

Let C1 and C2 be the conditions to set values in each level of FSM respectively.

for *Generated CRPs for APUF* **do**

L1: Alter the challenges in APUF to limit the Hamming distance.

 Find messages digest $HD = HD(C, R)$, where HD may vary according to CRPs.

if $HD == C1$ **then**

 Input HD to first level of FSM.

end

else

 GOTO L1.

end

L2: Trigger the BPUFs and obtain the private key KEY.

if $KEY == C2$ **then**

 Calculate Sign(HD, KEY).

 Select a ready ADD-ON IP core from heterogeneous IP-core module.

 Calculated sign(HD, KEY) is encapsulated to the ready ADD-ON IP core.

 Mark sign(HD, KEY) as the initiator in the ADD-ON IP core.

end

else

 GOTO L2

end

end

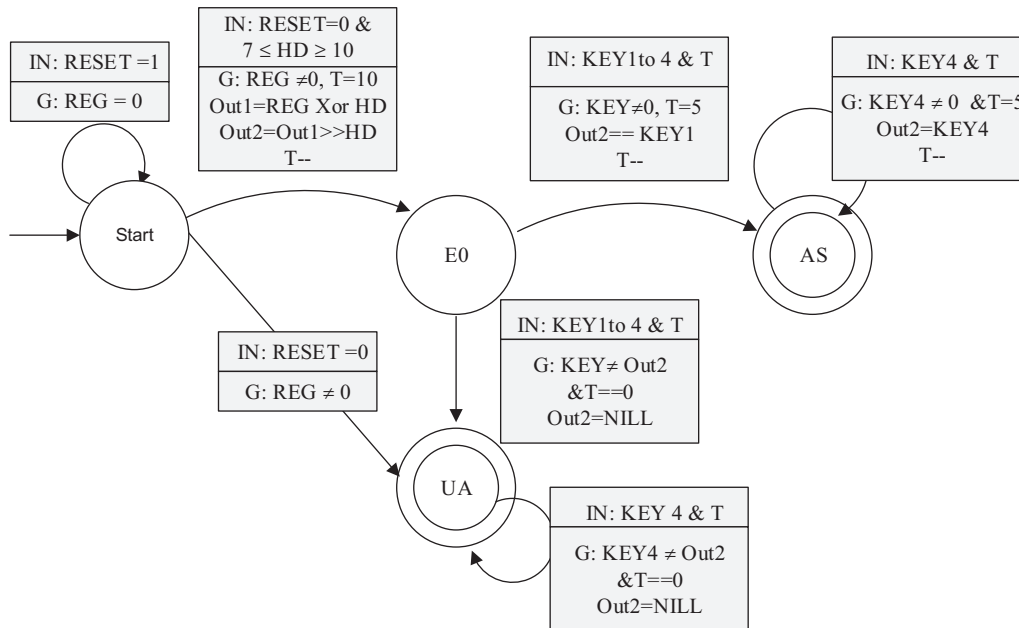


Figure 4. Hardware signature verification phase with EFSM.

which is rendered by the absolute transition in FSM and EFSM. This transition information is completely obsolete to the hackers.

Algorithm 2: Hardware Signature Verification

Input: Sign(HD, KEY) + IP Core
Output: Accept or Reject.
R1: Read the sign(HD,KEY) from IP core.
Input sign(HD, key) to E-FSM.
Input the public key t to E-FSM.
E-FSM will enter into accept or reject state based on the following steps:
if $OUT2 == KEY$ and $T = t$ **then**
 IP core is executed.
end
else if $OUT2 != KEY$ and $T != t$ **then**
 Start from R1.
end
else
 IP core is rejected and send to heterogeneous IP core module.
end

3. The HS is generated for a specific IP and the unique chip which is embedded in the IP core. Hence, the adversary cannot use the signed IP without the correct EFSM.
4. The feature of less storage area mentioned earlier is achieved by the hybrid model with FSM and EFSM for HS generation and verification respectively. The EFSM solitary is capable of verifying hardware signature.
5. This model is resilient to cloning and overbuilding, tampering, IP theft, and piracy of IoT-based IP cores.

The advantages of using FSM in our design are as follows:

1. An FSM makes the brute force attack difficult since the number of states in each level are dynamically triggered by two different PUF.
2. During verification, with the help of Hardware Signature an attacker can decrypt only one set of states in FSM. The values in other states will be different from that of the known state.

4.1. Analysis of HS attacks and PUF attacks

The basic attacks involved in the proposed HS are forgery, embedding, modification, and HS removal, which transpire between user and adversary. The vulnerability of PUFs to different types of attacks, such as brute force attack, invasive attack, modeling attack, and side channel attack are also discussed.

- In HS forgery, the adversary claims that the IP is owned by him and has adversary's HS. However, IP has been bought for higher cost and belongs to the original system developer with his signature. This is not possible since the HS has been generated by a hybrid model where the uniqueness, randomness, and reliability of the PUF are high. There is no mathematical model for the hybrid PUF and FSM; hence, HS regeneration using the same method in different chips is not practical.
- In HS embed, the hardware signature scheme is known to the adversary, and he tries to make his own HS embedded into the user's IP core. The adversary proves that the IP core is designed by him and he claims the ownership of it. Then he tries to execute the IP core in his design and generate a bitstream, but he is not a valid user. Since the hybrid model is designed for specific chip and IPs each time the EFSM is

updated, the process of embedding random values in the bit file of nonvolatile memory may not be valid at all times. Without an extended knowledge in IP design and PUF, the adversary will not be able to embed the HS and will be led to unauthenticated state.

- In HS modification, an adversary modifies the HS without the knowledge of the user; hence, the user cannot utilize the function of the IP core with the design. In such cases, the user faces a denial of service attack. The new HS has to be generated again and sent to the user based on the request. The user incurs a penalty in terms of cost for the same.
- In HS removal, the adversary tries to remove the owner's signature so that the IP will not carry proof of any authorship and the signature verification will fail. If the adversary seeks to remove the HS of the user, the IP will not be used by any of the user and IP becomes static and unused.
- In brute force attack, the hacker tries to guess the chip ID and IP key for utilizing the functionality of IP cores. The PUF responses are logically masked by FSM, which will create the possibility for an exponential trial. Hence, brute force attack is impossible.
- In invasive attack, there is no need to store the secret key in the chip. The hardware circuit itself generates a random number which is the secret key, and it is dynamically deleted after use. The keys cannot be duplicated, which will resist the invasive attacks.
- In modeling attack, a strong machine learning technique is applied to CRP of strong PUFs. Arbiter PUFs are vulnerable to this attack because of the most powerful machine learning technique. Hence, by converting linear function to nonlinear and increasing the CRP and using strong recent PUFs such as Anderson will resist this attack. The proposed model can use any recent PUFs as it enables IP reuse.

In our model, we are using FSM and EFSM along with hybrid PUF to resist against modeling attacks which is a combination of nonlinear functions. Trojan horse attacks are detected and corrected at design and hardware level [15]. PUF have full resistance towards side-channel attacks and semiresistance to Trojan horse attacks.

5. Experimental setup and evaluation approach

The proposed hybrid PUF and FSM model is successfully implemented in Zedboard, which is an SoC FPGA-based board. The strength of a high-performance processing system (PS) is combined along with flexible FPGA to form SoC FPGAs. This zedboard belongs to Zynq family and is created by Xilinx for all programmable SoC architectures. The dual-core ARM® Cortex™-A9-based PS and 28 nm Xilinx PL is integrated in a single device. The Vivado system edition is used to design the hardware module and a software development kit (SDK) to drive all the IP cores. The 16-bit APUFs, BPUFs, FSMs, and EFSMs are coded using Verilog and converted to IP blocks. The ADD-ON modules are created using verilog, VHDL, and C or C++ as it is an IoT-based IP core. The C or C++ IP cores are coded in the high-level synthesis of Vivado and tested in system edition with the proposed model. The output of this model is viewed in putty terminal. The design structure of the hardware signature generation and verification phase is diagrammatically represented in the Vivado suite depicted in Figure 5. AXI-interconnect is used to connect the IP cores to the processor in the design and a controller is used to control it. The 32-bit general-purpose AXI interface is used for low- and medium-rate data communication between the PL and PS. For high-rate communication, Zynq uses the AXI-HP interface [24].

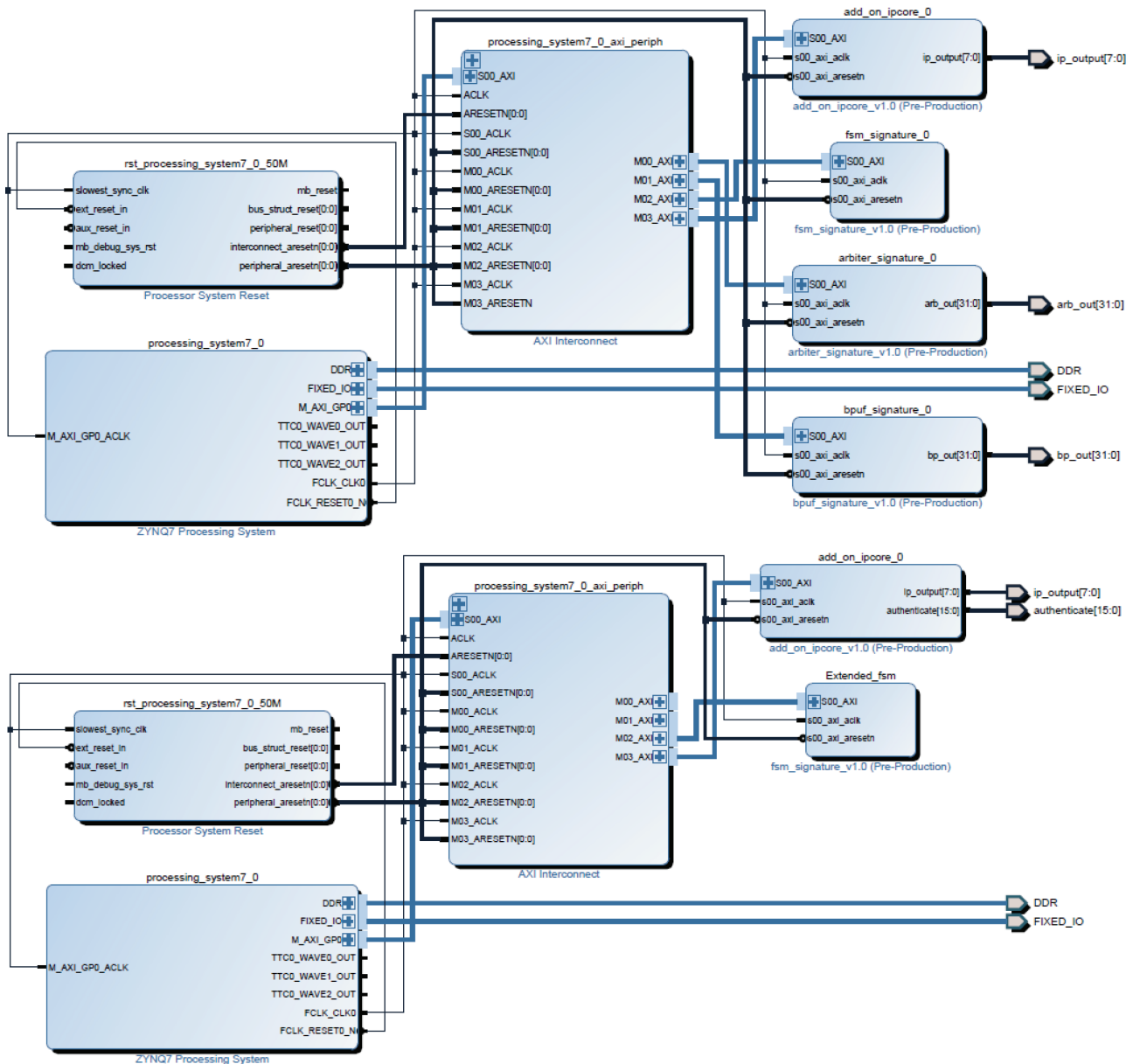


Figure 5. Vivado design for the hardware signature generation and verification.

6. Results and discussion

The results are evaluated based on the hybrid PUF security metrics, area, and power and verified on heterogeneous IP cores to show how it is well suited for the IoT device. The former evaluation is based on area, power, temperature, uniqueness, reliability, and randomness of PUFs and FSM which is demonstrated using Figures 6 and 7 and Table 1. Each APUF and BPUF is analyzed separately based on Figures 8 and 9, and its designated response is fed as input to FSM to generate HS. The later evaluation is to verify the efficiency of the proposed model with the heterogeneous modules, which are selected from different benchmarks, such as ISCAS, ITC,

and open core, and converted to IP cores. The area, power, and delay values are measured for each IP core with and without HS and are listed in Table 2. Table 2 and Figure 10 compares the utilization of area, power consumption, and delay between existing technologies and hardware signature for IP protections. Each of the above-mentioned evaluation will be discussed in detail in this section.

In Figures 6 and 7, the hybrid PUFs with FSM metric are estimated based on area and power values, for 8-, 16-, 32-, and 64-bit CRP respectively. The efficiency of the PUFs is measured using the three metrics, as mentioned in [25–27]. The area and power overhead are measured for 8-, 16-, 32- and 64-bit CRP of hybrid PUF, which depicts that the 16- and 32-bit are more prone to 28 nm technology. Since IoT edge devices have to be very small and consume less power, the proposal focuses on the 16- and 32-bit implementation. The other properties such as temperature, uniqueness, reliability, and randomness of the PUF are also considered for n-bit. The variations in power, execution time, and temperature are minimum for 2n bit increase. The area used mainly depends on the LUTs occupied by the design. The measured PUF metrics are different based on the size of the challenge and clock signals. The analysis depicts that 16-bit and 32-bit hybrid model is better for uniqueness and reliability.

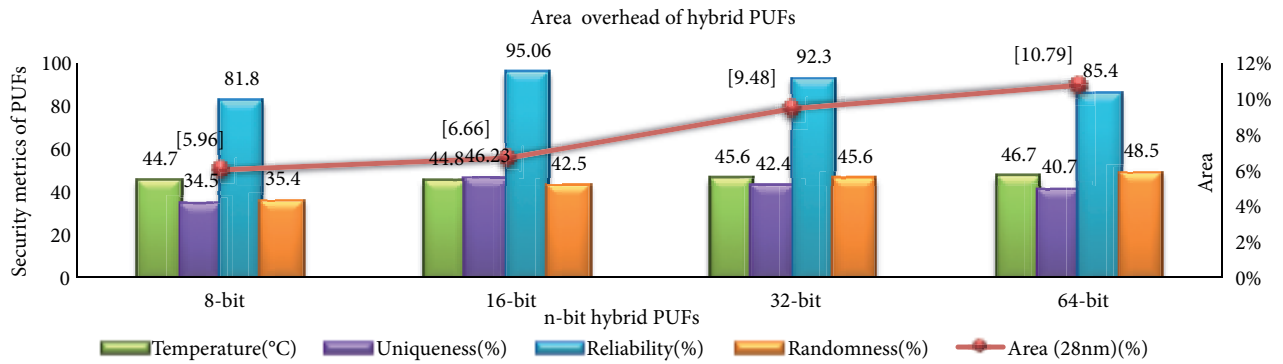


Figure 6. Area overhead of hybrid PUF with FSM for the security metrics.

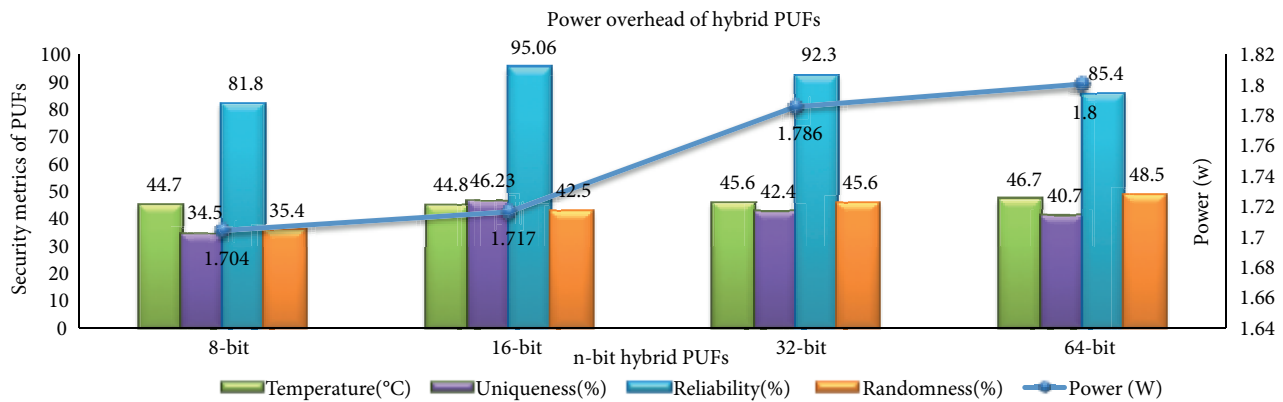


Figure 7. Power overhead of hybrid PUF with FSM for the security metrics.

The security metrics are measured for individual PUF and hybrid PUFs which are listed and compared in Table 1. While comparing with the individual PUFs, the hybrid PUF is better in terms of area, power, uniqueness, and reliability. The low-power hybrid RO PUF is implemented in 65 nm, and its power is measured in μW , and it is homogeneous. However, the proposed model is implemented in SoC FPGA, which is 28 nm with heterogeneous PUF; hence, its power and area vary and are uncomparable with those in [28].

Table 1. Comparison of A, B, and hybrid PUFs.

Measures	APUFs	BPUFs	Hybrid ABPUFs withFSM	Low-power Hybrid RO PUF [28]
Power (W)	0.959	0.098	1.717	32.3(μ W)
Process (nm)	28	28	28	65
Area (%)	56.21	55.39	22.66	250
Uniqueness (%)	45.67	36.23	46.23	50.42
Reliability (%)	98.40	91.86	95.06	97.22
Randomness (%)	49.6	30.45	42.49	NA
Temperature ($^{\circ}$ C)	43 to 125	43 to 95	44 to 46	40 to 120

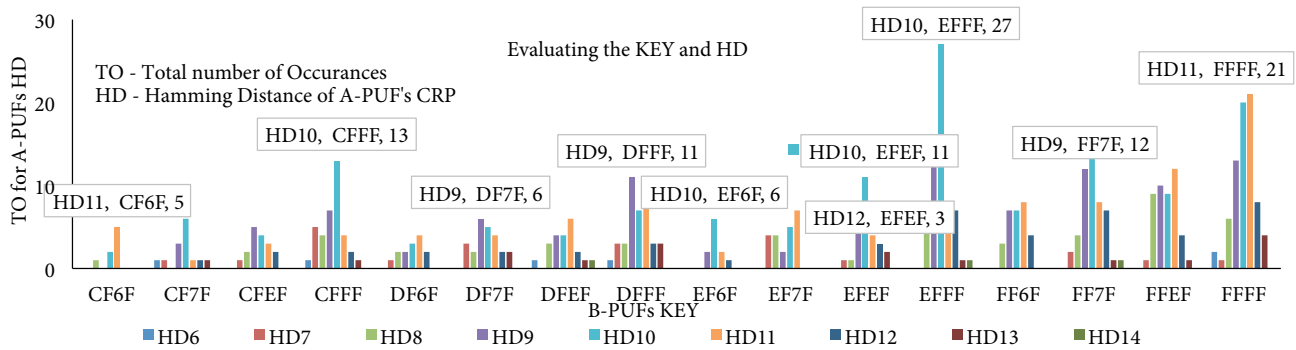


Figure 8. Evaluation of KEY and HD of hybrid PUFs with FSM for chip 1.

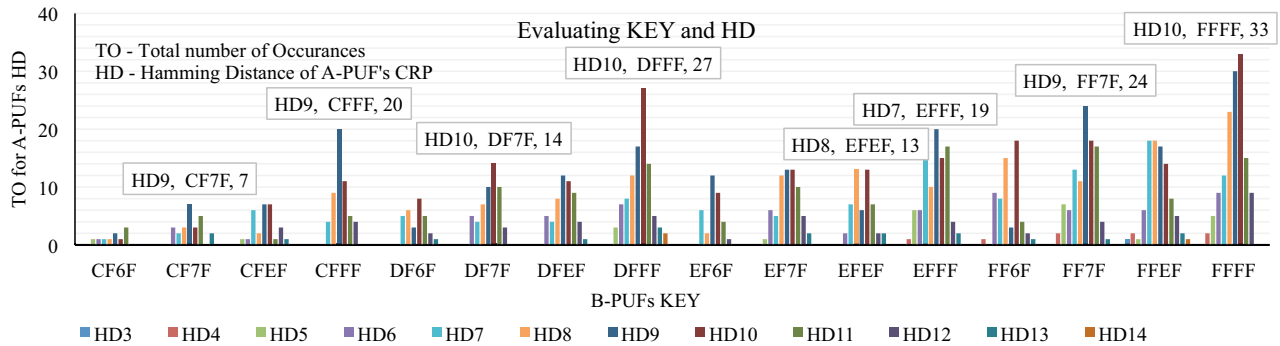


Figure 9. Evaluation of KEY and HD of hybrid PUFs with FSM for chip 2.

The proposed model is executed for 10,000 iterations, and a specific sample of 500 iterations has been selected for two different chips, and its HD and KEY values are fixed, based on Figures 8 and 9. The HD and KEY has to be chosen in such a way that it is used only once and should be unpredictable. Figures 8 and 9 include the call out for specific values which help us to select the HD and unique KEY with the highest total number of occurrences (TO) for 500 iterations. The FSM will select the KEY based on the condition set for the TO and HD. In Figure 8, the HD for FFFF, DFFF, and DF7F are 10, but the TOs are found to be 33, 27, and 14 respectively, which shows that TO is a variable which in turn lead to the observation that TO ranges can be set for the transit of FSM. The same process can be done for the value of HD 9. Figure 9 is used to fix

the HD and KEY for chip2 in FSM, in which the HD ranges from 6 to 14, and the TO will also be different for each KEY. Hence, the FSM will lead to a secure authentication state.

Table 2. Hardware signature overhead with different benchmarks; A and As, P and Ps, and D and Ds are area in utilization %, delay in s and power in W of the IP cores with and without hardware signature; ΔA , ΔP , and ΔD are the respective percentage increase.

Different Benchmarks	IP cores	A (U%)	P (w)	D (s)	As (U%)	Ps (%)	Ds (%)	ΔA (%)	ΔP (%)	ΔD (%)
ISCAS 89	S208	22	1.567	13.5	21.5	1.584	13.9	-2.27	1.08	2.96
	S1423	25	1.756	19.34	25.2	1.776	21.8	0.80	1.14	12.72
ITC 99	B17	29	1.785	25.45	28.7	1.805	25.65	-1.03	1.12	0.79
Open core	SPI	15	1.697	5.6	15.5	1.718	5	3.33	1.24	-10.71
DSP Core	FIR	18	1.709	19.89	18.8	1.723	21	4.44	0.82	5.58

The existing solutions like fingerprinting, watermarking, and security-based protocols are compared with the standard benchmark IPs from ISCAS, ITC, Open core and DSP core. Since IoT application is heterogeneous, IPs from the same benchmark is not suitable for testing this scheme. Hence, different benchmark for real-time application is selected to verify this model. Different benchmarks are taken for implementing IoT-based IP cores, and the results are analyzed in our model. On an average, the area, power, and delay vary based on IP core and the encapsulation of signature. The proposed model is not having much overhead in terms of area and power compared to the model without HS. The measured values are reported in the below table with heterogeneous benchmark IP cores.

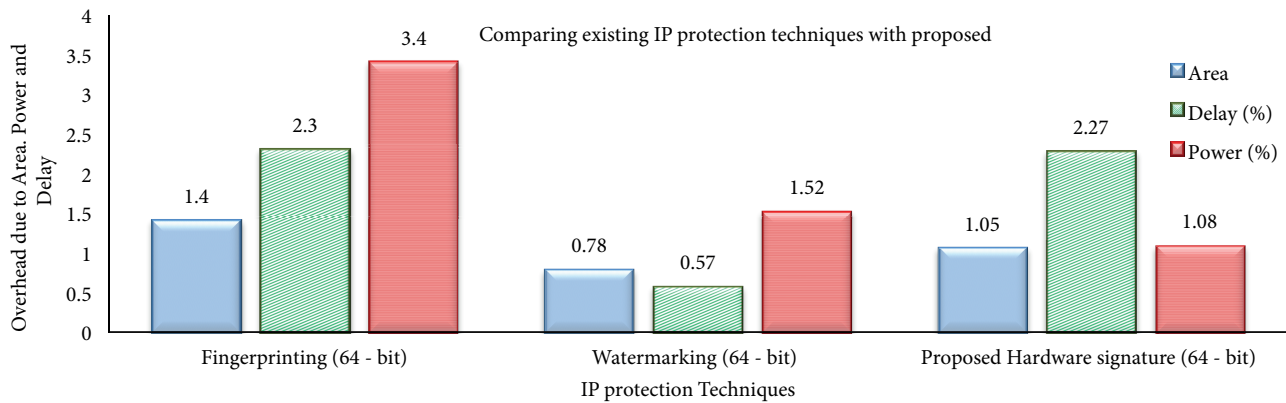


Figure 10. Area, power, and delay overhead of IP protection techniques.

The results of watermarking and fingerprinting are compared with those of the proposed hardware signature scheme in Figure 10 for 64-bit. They show that area and power ease the requirement of the IoT-based IP cores on SoC FPGA. On average, the area and power overhead were found to be 1.05 and 1.08, respectively as depicted in Figure 10. The delay is the only tradeoff, but when compared with fingerprinting concepts, it is acceptable. The delay can be reduced by using the partial reconfiguration feature of Xilinx FPGAs.

7. Conclusion

To add and protect IoT-based IP cores on the fly for SoC will depend mainly on selecting unified hardware- and software-based platform with suitable IP cores. As IoT is scattered and abounded, the selection of appropriate IP cores is very much challenging. A novel hardware signature mechanism has been proposed with hybrid PUFs and FSM to protect IoT-based IP cores. The results are analyzed for 10,000 iterations to select the CRP, HD, and KEY of the hybrid PUFs to design a secure FSM. The hybrid PUF characteristics such as uniqueness, reliability, and randomness are examined in 28 nm technology. The process of signature generation and verification is tested with the help of benchmark IP cores. The main achievement of this work is to avoid huge data storage in the chip and complex cryptographic algorithms. The existing IP protection techniques such as fingerprinting and watermarking are compared with the proposed model, which shows that the area, power, and cost overhead are reduced. A future direction is to increase the size of CRP using partial reconfiguration concepts and to analyze the qualitative effect of physical threats such as side channel attack, hardware Trojan attack, and reverse engineering in the proposed model. The extension is concerned with the reliability of the system, which will certainly lead to a better trustworthy framework.

References

- [1] Sjoval P, Virtanen J, Vanne J, Hamalainen TD. High-level synthesis design flow for HEVC intra encoder on SoC-FPGA. In: IEEE 2015 Euromicro Conference on Digital System Design (DSD); Funchal, PORTUGAL; 2016. pp. 49-56.
- [2] Synopsys. DesignWare IP for IoT SoC Designs. (White paper) East Middlefield Road Mountain View, CA, USA: Synopsys, 2016.
- [3] Dhote S, Charjan P, Phansekar A, Hegde A, Joshi S et al. Using FPGA-SoC interface for low cost IoT based image processing. In: IEEE 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI); Jaipur, INDIA; 2016. pp. 1963-1968.
- [4] Bol D, De Vos J, Botman F, de Streel G, Bernard S et al. Green SoCs for a sustainable Internet-of-things. In: IEEE 2013 Faible Tension Faible Consommation Conference (FTFC); Paris, FRANCE; 2013. pp. 1-4.
- [5] Srinivasan N. CSoC Platform / Digital Subsystem IP for IOT. (White paper) Mindtree, India: Design & reuse, 2015
- [6] Zhang J, Lin Y, Lyu Y, Qu G. A PUF-FSM binding scheme for FPGA IP protection and pay-per-device licensing. IEEE Transactions on Information Forensics and Security 2015; 10 (6): 1137-1150. doi: 10.1109/TIFS.2015.2400413
- [7] Lao Y, Yuan B, Kim CH, Parhi KK. Reliable PUF-based local authentication with self-correction. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 2017; 36 (2): 201-213. doi: 10.1109/TCAD.2016.2569581
- [8] Zhang J, Qu G. Rebuttal to Comments on 'A PUF-FSM binding scheme for FPGA IP protection and pay-per-device licensing. IEEE Transactions on Information Forensics and Security 2016; 11 (11): 2626-2627. doi: 10.1109/TIFS.2016.2553443
- [9] Machida T, Yamamoto D, Iwamoto M, Sakiyama K. A new arbiter PUF for enhancing unpredictability on FPGA. The Scientific World Journal 2015. doi: 10.1155/2015/864812
- [10] Kumar SS, Guajardo J, Maes R, Schrijen GJ, Tuyls P. The butterfly PUF protecting IP on every FPGA. In: IEEE 2008 IEEE International Workshop on Hardware-Oriented Security and Trust; Anaheim, CA, USA; 2008. pp. 67-70.
- [11] Sengupta A. Intellectual property cores: Protection designs for CE products. IEEE Consumer Electronics Magazine 2015; 5 (1): 83-88. doi: 10.1109/MCE.2015.2484745

- [12] Lao Y, Parhi KK. Obfuscating DSP circuits via high-level transformations. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 2015; 23 (5): 819-830. doi: 10.1109/TVLSI.2014.2323976
- [13] Chang CH, Cui A. Synthesis-for-testability watermarking for field authentication of VLSI intellectual property. *IEEE Transactions on Circuits and Systems I: Regular Papers* 2010; 57 (7): 1618-1630. doi: 10.1109/TCSI.2009.2035415
- [14] Cui A, Chang CH, Tahar S, Abdel-Hamid AT. A robust FSM watermarking scheme for IP protection of sequential circuit design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 2011; 30 (5): 678-690. doi: 10.1109/TCAD.2010.2098131
- [15] Kufel J, Wilson PR, Hill S, Al-Hashimi BM, Whatmough PN. Sequence-aware watermark design for soft IP embedded processors. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 2016; 24 (1): 276-289. doi: 10.1109/TVLSI.2015.2399457
- [16] Cui A, Qu G, Zhang Y. Ultra-low overhead dynamic watermarking on scan design for hard IP protection. *IEEE Transactions on Information Forensics and Security* 2015; 10 (11): 2298-2313. doi: 10.1109/TIFS.2015.2455338
- [17] Chang CH, Zhang L. A blind dynamic fingerprinting technique for sequential circuit intellectual property protection. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 2014; 33 (1): 76-89. doi: 10.1109/TCAD.2013.2282282
- [18] Saha D, Sur-Kolay S. Embedding of signatures in reconfigurable scan architecture for authentication of intellectual properties in system-on-chip. *IET Computers & Digital Techniques* 2016; 10 (3): 110-118. doi: 10.1049/iet-cdt.2015.0051
- [19] Herder C, Yu MD, Koushanfar F, Devadas S. Physical unclonable functions and applications: A tutorial. *PROCEEDINGS OF THE IEEE* 2014; 102 (8): 1126-1141. doi: 10.1109/JPROC.2014.2320516
- [20] Department of Information Technology. Guidelines for Usage of Digital Signatures in e-Governance. Ministry of Communications and Information Technology, New Delhi, Government of India, India: Rep. Government of India. Version 1.0 , 2010
- [21] National Institute of Standards and Technology. Implementation Guidance for FIPS 140-2 and the Cryptographic Module Validation Program. Canadian Centre for Cyber Security, Government office in Ottawa, Canada: Rep.FIPS, 2019
- [22] Locke G, Gallagher P. Digital signature standard (DSS). National Institute of Standards and Technology, Gaithersburg, MD, USA: Rep. FIPS PUB 186-3, 2013.
- [23] Di Guglielmo G, Di Guglielmo L, Fummi F, Pravadelli G. Efficient generation of stimuli for functional verification by backjumping across extended FSMs. *Journal of Electronic Testing*. 2011; 27 (2): 137. doi: 10.1007/s10836-011-5209-8
- [24] Crockett LH, Elliot RA, Enderwitz MA, Stewart RW. *The Zynq Book: Embedded Processing with the Arm Cortex-A9 on the Xilinx Zynq-7000 All Programmable Soc.* University of Strathclyde Glasgow, Scotland, UK: Strathclyde Academic Media, 2014.
- [25] Maiti A, Gunreddy V, Schaumont P. A systematic method to evaluate and compare the performance of physical unclonable functions. In: Athanas P, Pnevmatikatos D, Sklavos N (editors). *Embedded Systems Design with FPGAs*. New York, NY, USA: Springer, 2013. pp. 245-267.
- [26] Bhargava M. Reliable, secure, efficient physical unclonable functions. PhD, Carnegie Mellon University Pittsburgh, Pittsburgh, PA, USA 2013.
- [27] Cherkaoui A, Bossuet L, Marchand C. Design, evaluation, and optimization of physical unclonable functions based on transient effect ring oscillators. *IEEE Transactions on Information Forensics and Security* 2016; 11 (6): 1291-1305. doi: 10.1109/TIFS.2016.2524666
- [28] Cao Y, Zhang L, Chang CH, Chen S. A low-power hybrid RO PUF with improved thermal stability for lightweight applications. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 2015; 34 (7): 1143-1147. doi: 10.1109/TCAD.2015.2424955