Received: September 24, 2020. Revised: October 21, 2020. 169

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

A Non Linear PUF Circuit Design for Two Factor Authentication in IoT Cryptography

Krishna Priya Gurumanapalli[1]* **Nagendra Muthuluru[1]**

[1]*Department of Computer Science and Technology, Sri Krishnadevaraya University, India*
* Corresponding author's Email: priya.racharla@yahoo.com

Abstract: Internet-of-Things (IoT) is growing network paradigm which enables mutual communication between the user and smart devices using the internet. The IoT devices are susceptible to the security threats, due to placement of restricted computational capabilities of the computing devices in IoT. The conventional encryption algorithm utilizes the high amount of resource block in it which increases the area and power. Moreover, Two Factor Authentication (TFA) scheme based authentication protocols does not have the efficiency to secure the data. Because the random number generated by the TFA is ideal for all IoT devices which are easy to hack by the unauthorized persons. In this Research paper, the Linear Feedback Shift Register (LFSR) based Reconfigurable Physical Unclonable Function (RPUF) is proposed to overcome the security issues caused in the IoT communication. The RPUF is designed based on the LFSR to generate the random number for every clock cycle. Normally, reconfigurable process helps to generate the different output values for every clock cycle. But, it failed to generate different outputs for same input values. Here, LFSR based RPUF helps to generated the different response values even the same challenge is given to the input side. The Lightweight TFA scheme is presented for IoT, where PUF has been considered as one of the major authentication factors. At last, Spartan 6 and Virtex 6 Field Programmable Gate Array (FPGA) performances are calculated for proposed TFA-RPUF-IoT and existing TFA-PUF-IoT protocols. In Spartan 6, TFA-RPUF-IoT protocol occupied 11 slices, 31 LUTs, 42 flip flops which are less compared to conventional TFA-PUF-IoT.

Keywords: Field programmable gate array, Internet-of-things, Linear feedback shift register, Two factor authentication, Reconfigurable physical unclonable function.

1. Introduction

In the past decades, the number of IoT devices has been increasing rapidly to enable power grids, health care, control system, etc. Most of the IoT devices operate in the public places, so it is more vulnerable to hacker's attack [1]. The data losses from the various fields are avoided using the active internet and network security management systems. The corporate and government confidential information are difficult to protect due to the size of the data, so it's important to secure that information in many ways, such as privacy protection, data integrity, access control, booting, and digital forgetting [2-4]. IoT devices can be considered as an embedded system because of the following properties: First, the embedded system should have a specific task and enough memory for processing the specific task. Next, the IoT device is considered "headless" because these devices are not operated by a human. For designing these types of protocols, certain security model needs to be implemented with more communication standards, less power, and memory usage [5].

The normal security model has been proposed by many researchers such as authentication protocols [6], using hash function [7], RSA cryptosystem [8], elliptic curve [9], chaotic map [10] and bilinear pairing [11]. However, these protocols are not suitable for multi server communications and it has less randomness. To overcome these issues, two factor authentication protocols were introduced in IoT devices. In the conventional methods, some of the two factor authentication has been proposed by the researchers such as mutual authentication PUF

[12], RSA based robust authentication [13], Elliptical Curve Cryptography (ECC) [14] and key exchange protocol [15]. Aforementioned two factor authentication protocols were not generated the random number for every clock cycle and the same output data have generated for the same inputs. So, hackers can easily retrieve the operation of IoT devices. Moreover, the behaviour model elements were used to design the aforementioned conventional authentication protocols. To overcome the above mentioned problems, reconfigurable PUF is used in this research work. The major contribution of the research work is given as follows:

- All the IoT devices can communicate with each other without any data loss.
- Each IoT device has to request the server and the challenge value of each device is given to the server for setup authentication.
- With the help of reconfigurable PUF, the random data has generated which varied for each clock cycle.
- LFSR based RPUF used to generate the different response even the same challenge has given in the input terminal.
- Simple XOR based LFSR helped to design the TFA architecture with less hardware utilization.
- Due to the usage of reconfigurable PUF and the Gate level model, the hardware utilization of the two factor authentication has improved.
- Variation factor helps to show the linearity of the PUF and RPUF modules.
- LUT, flip flop, and slices are evaluated for Spartan 6 and Virtex 6 FPGA families. These performances are reduced in TFA-RPUF-IoT compared to TFA-PUF-IoT architecture.

This research paper is organized as follows: Related works are provided in section 2, the problems of existing model and solution are given in Section 3. The results and discussion are presented in Section 4, and conclusion and future scope of this research are provided in Section 5.

2. Related works

Gope and Sikdar [16] developed the lightweight and privacy-preserving two-factor authentication scheme in IoT device that used to enable the secure communication with the server installed in the data and control unit. Here, the PUF was considered as one of the important authentication factors as well as the authentication scheme was separated into phases such as Setup, and Authentication. The IoT device sent the identity along with the registration request to the server during the setup phase. In authentication phase, the IoT device was interacted with the server and the device selected the one-time alias identity to secure the communication. The inherent security features of PUFs were used to obtain the adequate security characteristics. However, the two factor algorithm was provided the security only at the physical layer.

Bui [17] presented the Advanced Encryption Standard (AES) Data path optimization for Low power multi security level IoT. The security function was not only concentrated on power, but it was also processed with less energy consumption. The multiple levels of security were developed to process the AES with high speed and ultra-low energy. This multiple level security helps to perform the key expansion and to set the key size. This algorithm has achieved 10 MHz frequency at 0.6V with the throughput of 28 Mb/s in ST FDSOI 28-nm technology. This process of the algorithm was affected by a brute force attack.

Zakaria [18] proposed the IoT security risk management model for secured practice in the healthcare environment. In the healthcare industry, IoT technologies plays an important role to interconnect medical devices and sensors. Generally, IoT was the point of a breach where attackers are able to identify the vulnerabilities and subsequently launch their attacks. IoT has segregated into three different models such as Healthcare IoT risk management, Hospital performance indicator for accountability and the implementation phases. The risk management model was affected by the side channel and related key attacks caused the data loss. The confidential information related to the patients' health history was stolen by the hackers.

Al-Asli [19] presented FPGA based symmetric re-encryption scheme to secure the data processing for cloud-integrated IoT. The proposed scheme supports various business models involving multiple parties and the data owner to give temporary access to IoT data to specific clients at a public market place (the cloud). Re-encryption scheme achieved perfect forward secrecy and it provided the FPGA authentication to establish a symmetric session key between the on-cloud FPGA and the IoT device. Due to the usage of Proxy re-encryption, 25981 LUTs and 15599 flip flops were occupied the full system which increase the design complexity.

Wazid [20] proposed user authenticated key management protocol for generic IoT networks. This paper emphasizes on the design of a new secure lightweight three factor remote user authentication scheme for Hieratical IoT networks (HIoTNs), called

the user authenticated key management protocol (UAKMP). The three factors used in UAKMP are the user smart card, password, and personal biometrics. The security of the scheme was thoroughly analyzed under the formal secure Real-Or-Random (ROR) model. The information security as well as the formal security verification was performed using the widely accepted AVISPA (Automated Validation of Internet Security Protocols and Applications) tool. Due to the remote user authentication, some of the data has not received perfectly to the receiver section.

3. TFA-RPUF-IoT protocols

The major concerns related to IoT devices securities are illegal access to information, authentication, privacy, and tracking of the data stream and data confidentiality. The User/Device authentication has become the most primary issue due to the growing concern with user IoT security and privacy leakage. The device authentication in IoT devices is one of the most critical issues encountered in the design of IoT applications. Therefore, providing robust security to IoT devices is a major challenge and it requires a brittle authentication scheme that should be robust for all kinds of side-channel attacks. For secure communication and authentication between the devices, cryptography is an essential tool. The code generated by the conventional PUF is ideal for all clock cycles, because of its unalterable internal structure of IC i.e., PUF. Hence, the identification of the code generated by the PUF is an easy task, when it is identical to the entire communication system.Therefore, the conventional PUF is required to be modified for enhancing the security against the attacks.

3.1 TFA-PUF-IoT existing model

The device authentication is a crucial security feature for IoT. Many IoT devices are deployed in the open and public places, which makes them vulnerable to physical and cloning attacks. Therefore, any authentication protocol designed for IoT devices should be robust even in cases when an IoT device is captured by an adversary. Many conventional cryptography algorithms introduced for IoT networks are password-based or secret-key based authentication schemes. These schemes consume more logical resources in integrated circuits which leads to more dynamic power dissipation. In general, IoT devices have limited power, memory storage, device area, and processing capabilities which increases the size and affects the compactness of the devices. Also, those schemes can be executable only based on the secret key exchange where there is more possibility for side channel attacks due to key leakage. The primary solution to overcome this problem is the introduction of a Lightweight and Privacy-Preserving Two-Factor Authentication Scheme in IoT devices.

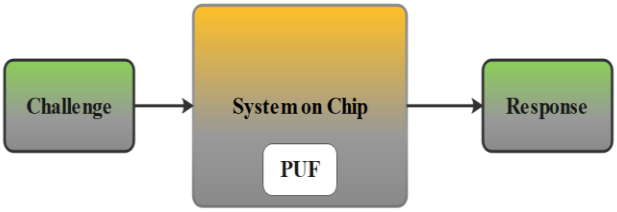

Figure. 1 A PUF model in IoT device

In this scheme, a password or a shared secret key is generated as the first authentication factor using a Fuzzy Extractor (FE) composed with two algorithms: FE Generator and FE Recovery.

1. FE. Gen is a probabilistic key generation algorithm, which takes a bit string R as input and outputs a key K and helper data hd. (K, hd) = FE. Gen(R).
2. FE. Rec is a deterministic reconstruction algorithm that recovers the key K from the noisy input variable R0 and the helper data hd i.e., K = FE. Rec (R0, hd),

The second authentication factor called PUF is introduced which can generate a unique code based on the internal microstructure of Integrated circuits. The PUF can be designed using a different combination of the digital circuit comprises transistors that can perform the digital operation to generate unique code. Each IoT devices is designed with different PUF generator circuits. A conventional IC with PUF is presented in Fig. 1.

3.1.1. Problem identification and solution

Lightweight and Two-factor authentication scheme provides a better solution for highly secured IoT device authentication. However, the drawbacks of this scheme are explained as follows:

TFA scheme is implemented mutual authentication between server and the device which is worked based on PUF secret key. This novel scheme provides robustness to security threats only at the physical layer for highly secured data transmission and it requires another cryptography scheme for data security. The distinctive feature of PUF is that it can generate a unique code, but it is fixed for each device as the internal structure of IC (PUF) cannot be modified. Nowadays, the evaluation of Artificial Intelligence (AI), machine learning techniques has been used to improve robustness in security protocols, but at the same time, those

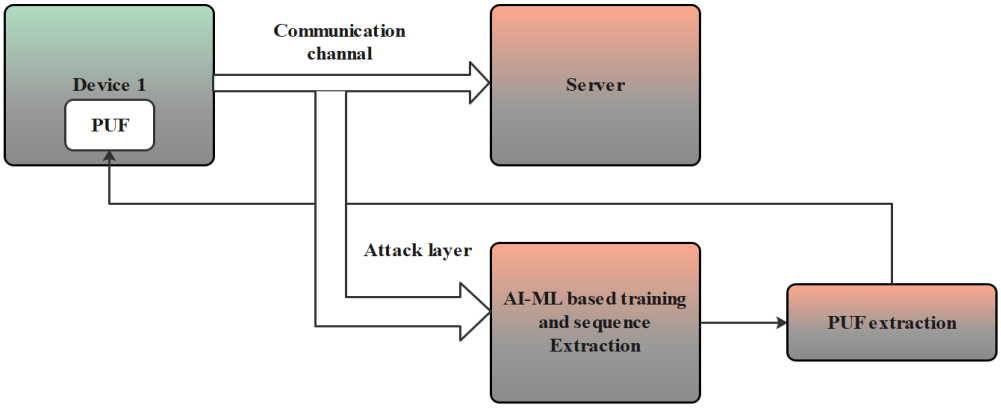

Figure. 2 AI-ML based attack model

techniques can be adopted for side channel attacks. From this view, it is possible to extract PUF code by subjecting Challenge Response Pair (CRP) into advanced AI- ML techniques which is presented in Fig. 2. So there is a need for modification in the existing TFA scheme to overcome PUF security issues in both the physical layer and communication channel.

As mentioned in the previous section, the PUF circuit can generate different functions for different IoT devices. But, the generated function values are fixed for each device. All the IoT devices are equipped with a PUF, where any attempt to tamper with the PUF will change the behaviour of the device and render the PUF useless. To overcome this problem, a dynamic path reconfiguration model is introduced in PUF with an adaptive circuit transformation mechanism based on device behaviour. By introducing this approach in the TFA scheme, the device can generate a different Uncloneable function for each interaction between server and device. For each interaction, the physical path between logic gates or circuits of PUF-IC can modify itself to generate unpredictable challenge response. To ensure data security, the CRP scheme with path reconfigurable LFSR is applied to data to ensure high differentiation between data and CRP.

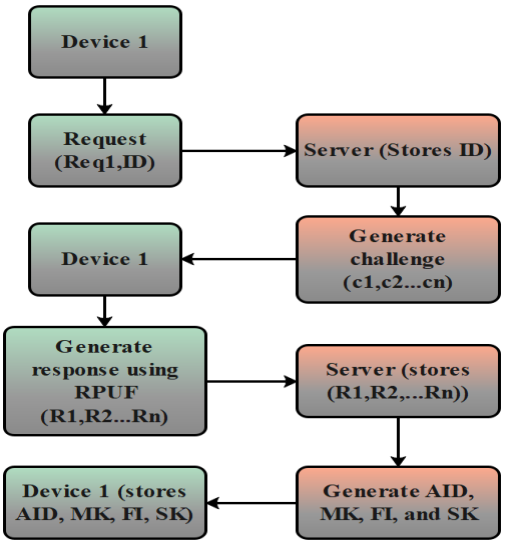

Figure. 3 Flow chart for setup phase

3.2 Reconfigurable PUF IoT set up phase:

In the setup phase, the IoT device generates the request and Identify number (ID) which are connected to the server. The flow chart of the setup phase is shown in Fig. 3. The server has stored the values which have received from the respective IoT devices. Based on the ID, the challenge has generated and connected to the same IoT device. The challenge values are helped to generate the response value based on the reconfigurable PUF which is explained in section 3.4.

The generated response values are stored in the server in the name of $R1, R2, .. Rn$. Based on the server response value, the server has generated the one-time Alias Identity (AID), Master Key (MK), Fake Identity (FI), and synchronization Key (SK) and all these values are stored in the respective IoT devices.

The AID is generated using MK and outputs from RPUF is expressed in equation Eq. (1).

$$AID = h(R||MK) \tag{1}$$

Where, one way hash function is represented as h; outputs from the RPUF is R and the server's master key is represented as MK.

Subsequently, the server also creates the unique fake identity and pairs of synchronization keys are expressed in the Eq. (2).

$$(FD, SK) = \{(fid_1, k_1), (fid_2, k_2), \dots, (fid_n, k_n)\} \tag{2}$$

Where, $fid_1, fid_2, \ldots, fid_n$ denotes the fake identify and $k_1, k_2, \ldots, k_n$ is the synchronization key pairs for n amount of IoT devices.

3.3 Authentication phase

In the authentication phase, the accessing permission has to give the perfect IoT devices where the server and device nonce has matched. The flow chart of the authentication phase is shown in Fig. 4.

In the authentication, the random number request has checked the AID, and Message ($M1$) has generated for the initial acceptance expressed in Eq. (3).

$$M1: \{AID, N_d^*\} \quad (3)$$

Where, the request message composed by the device is $M1$; $N_d^* = N_d \oplus K_{ds}$, N_d is the random number generated during the interaction and K_{ds} is secret key.

If the AID matches, master key, challenge value, and the response has loaded else the request will be rejected. The server has generated the server nonce (N_s) and hash key response composed of all the message values (M2) that is shown in Eq. (4).

$$M2: \{C, N_s^*, V_0\} \quad (4)$$

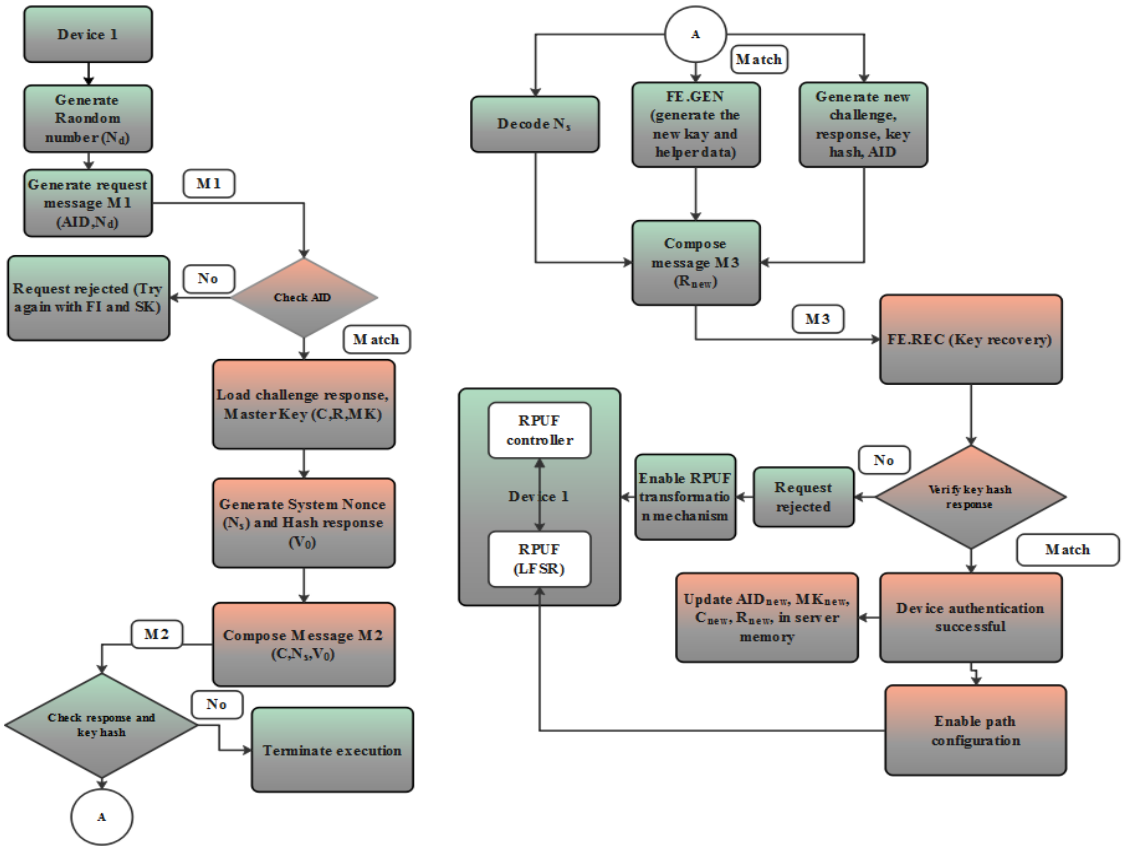

Figure. 4 Flow chart for the authentication phase

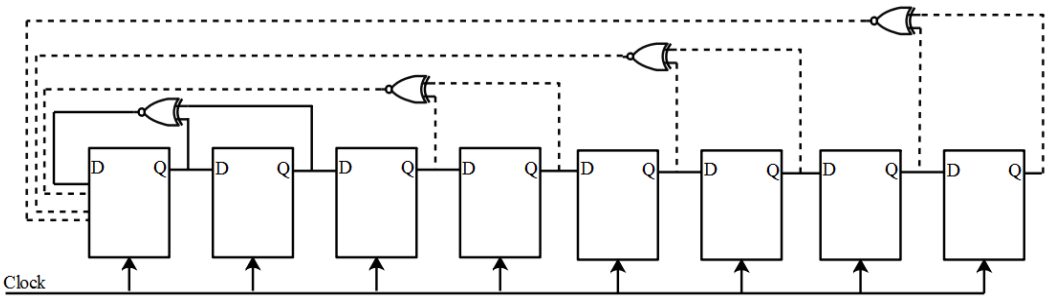

Figure. 5 Linear feedback shift register

Where, $M2$ is the response message; C is the challenge; $N_s^* = K_{ds} \oplus N_s$ and $V_0 = h(N_d||K_{ds}||N_s^*)$.

If the response is matching, the IoT device has generated the $M3$ new values which is expressed in Eq. (5).

$$M3: \{R_{new}^* , V_1, hd^*\} \quad (5)$$

Where, $R_{new}^* = k \oplus R'_{new}, R'_{new} = RPUF_{D_i}(C_{new}), C_{new} = h(C_i||K_i), hd^* = h(K_{ds}||k||hd), hd = h(K_{ds}||N_s) \oplus hd^*, k = FE.Rec(R, hd), R' = RPUF_{D_i}(C)$ and $V_1 = h(N_s||k||R_{new}^*||hd^*)$. The helper data generation algorithm is represented as $FE.Gen$, extracted RPUF output during server authentication is R', hd specifies the helper data , k specifies the key element and V_1 is the key hash response.

The generated $M3$ values are performed in the Fuzzy Extractor recovery model. Every clock cycle, the reconfigurable PUF has generated the random number which is compared with the key hash response. If the server nonce and device nonce are available in the key hash function, the respective IoT devices can get the authentication from the server.

The IoT device and the server blocks are separated by different colors. All the output values are updated in the final stage of the authentication process [16].

3.4 Reconfigurable PUF

PUF model using LFSR is shown in Fig. 5. The LFSR used in the RPUF is similar to the shift register with feedback as well as the LFSR is used due to its less computation cost, good statistical properties and less gate computation. The outputs of the flip flops used in the shift register are given as feedback to the XOR gate's input and the XOR gate's output is given as input for the 1st flip flop of the shift register. The initial value saved in the shift register is named as a seed value. The random sequence of the bits is generated by the LFSR and the feedback outputs are given to the XOR gate. The LFSR with size n (i.e., number of registers) has the capacity to generate the possible states in the period of $N = 2n - 1$ that excludes all zero state. Therefore, this LFSR is used in the design of RPUF due to its error correction and communication circuits.

Generally, the IoT devices are generating the challenge which is given to the server for getting authentication purposes. Normally, the random number has generated in the IoT device for security purposes. In previous models, the same response has been generated for all the clock cycles. This process is too easy for an unauthorized person to retrieve the data. So, a reconfigurable PUF random number has generated in this research work. The challenge input values are given in Eq. (6). In this work, the LFSR circuit has used to generate the response output which is given in Eq. (7).

$$challenge\ (C) = \{c[7], c[6], c[5], c[4], c[3], c[2], c[1], c[0]\} \quad (6)$$

$$Response\ (R) = \{r[7], r[6], r[5], r[4], r[3], r[2], r[1], r[0]\} \quad (7)$$

Here, c [0] to c [7] are represented as challenge values.

Where r [0] to r [7] values are represented as response values that are generated from LFSR.

The response generation of the IoT devices is shown as follows:

$$always\ @\ (posedgeclk)$$
$$if(rst)$$
$$Response(R) = Challenge(C)$$
$$Else$$
$$Response(R) = \{r[0]\hat{}r[3], r[7:1]\}$$

The response generation has processed on the positive edge of the clock signal. Whenever the clock signal meets the positive edge, the response has generated from the challenge. If the reset signal has enabled, the challenge values are stored in the response terminal. Whenever the reset signal has changed, the XOR operation has performed for r [0] and r [3] which are concatenated with r [7:1].The random number has generated with the help of the LFSR circuit model. The feedback process has helped to generate the numbers with more randomness. The difference between two response values are called as hamming distance which is high for every clock cycle response value. Moreover, the challenge values are repeated for different IoT devices which can communicate with different IoT devices. For example, during the initial clock cycle, the challenge

Table 1. Example of LFSR process

8-bit data								LFSR output
0	1	0	0	0	0	0	0	64
1	0	1	0	0	0	0	0	160
1	1	0	1	0	0	0	0	224
1	1	1	0	1	0	0	0	232
1	1	1	1	0	1	0	0	248
1	1	1	1	1	0	1	0	250
1	1	1	1	1	1	0	1	253

values are generated 10 decimal values which are undergone the LFSR response process. After processing the reconfigurable PUF, the response values are generated. In the next clock cycle, the response value has been varied based on the challenge values. If the same 10 decimal values are given to the different IoT devices, then the different response values will be generated. So, different kinds of random data have generated in the RPUF even the same challenge has given to the respective IoT. The process of simulation setup and results are explained in the following section.

The example process of LFSR with 8-bit data are shown in Table 1. Initially, the input of 64 is initialized as seed value which is converted into 8 bit binary value (01000000) as well as this 8 bit is given for first clock cycle. Next, the 7^{th} (0) and 6^{th} bit (1) values are XORed and the XORed value is replaced instead of 7^{th} bit value. Then the remaining values are right shifted which is equal to the 160 (i.e., 10100000). Similarly, the same XOR and right operations are carried out for the remaining clock cycles. In conventional PUF, the XOR operation has performed for $7^{th}$and 6^{th}bit for all the clock cycles. But, the proposed RPUF frequently changes the pair of bit positions to obtain the secure communication. For example, the different pair of bits considered in RPUF to perform XOR operation are (7,6), (6,5), (5,4), (4,3), (3,2), (2,1) and (1,0). During the XOR operation, the RPUF changes the pair of bits in every clock cycle which is difficult to identify by unauthorized users.

4. Simulation setup

The proposed work has been implemented using the system having i5 processor, 8 GB RAM and 256 GB SSD hard disk configuration. The authentication and setup phase has implemented using Verilog language which helped to design the logical elements. With the help of Xilinx 14.4 software, the hardware utilization of the proposed method has evaluated and verified with the summary results. Modelsim 10.5 software has used for the simulation that helps to verify the authentication phase and setup mode.

4.1 Results and discussion

Initially, the setup phase has to perform for each IoT device and the server. The setup phase has majorly processed with the help of control signals. In this phase, the control signals are considered as the clock, reset, enable. Based on the number of devices connected to the server, the enable and reset signals are varied.

The offset of the clock signal has represented as 0. The clock period and duty cycle of the clock signals are mentioned as 100ns and 50ns respectively. In this case, the overall single clock cycle required a 100ns time duration to complete the single cycle. From the 100ns, 50ns are considered as positive clock edge as well as 50ns are considered as negative clock edge respectively. The logical value and the edge type are mentioned as 1 and rising edge respectively. In the setup phase, the phase control signal is denoted as zero. Remaining all the control signals are represented as one to operate the setup phase with less number of losses.

After setting the control signals, the input of the setup phase is given to the main block. The data input is given in the form of 16 bit, which is represented as "0010101100101001". This input is given to the entire top module architecture to perform the remaining stages. The delay value of applying input is zero, which helps to operate the architecture without any delay. Two enable signals (enb1, enb2) are used to start the progress of each device. After applying the input value, each device can generate the device ID and the request is given to the server. Based on the request, some challenges have generated from the devices. This challenge is processed on the server and the server produce the response for the respective devices. The process of generating a response in the server has two types such as PUF and reconfigurable PUF.

The input of this module is considered as the control signals and the challenges values. The response output of conventional PUF is shown in Fig. 6. Clk and rst signals are used to control the entire system. In the conventional PUF, the standard response only generated based on the standard challenge value. There is no variation in the response value which helps the hackers to get the data. If the challenge value is 8, the output response is generated 143 decimal values in the response terminal. This result is the same for all the clock cycles if the challenge value is 8.

In conventional PUF, there is more possibility to identify the response value which is generated from the server. Because the response value is retained the same for all the devices when the same challenge is given to the server. The proposed reconfigurable PUF output is presented in Fig. 7. According to the reconfigurable PUF waveform, the output response is generated differently for every clock cycle. If the challenge value is 8, then the response has generated like 8, 132, 66,72,164 etc. From this output response, it is clear that the output is generated randomly even the same challenge is given to the next clock cycle.

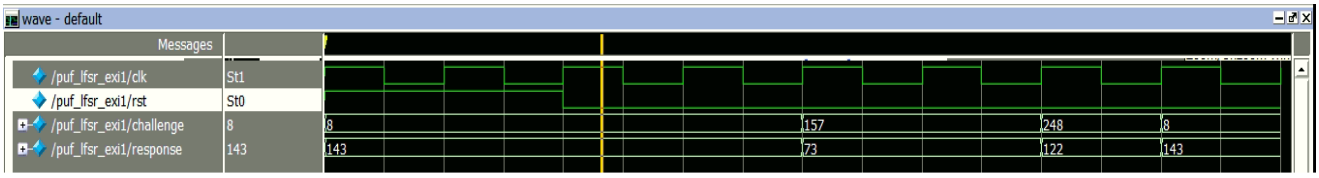

Figure. 6 Response output of conventional PUF

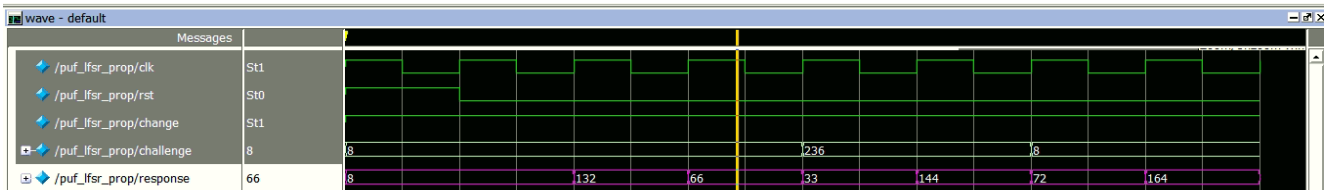

Figure. 7 Response output of Reconfigurable PUF

This random number generation is a must in the server progress to provide access to the IoT devices.

After generating the response from the server, the IoT devices get the random number which is more helpful in the authentication process. Based on the response value, remaining all the parameter values are generated which is represented as AID, FID, and Ksync in the waveform window. With the help of these values, the IoT devices can update the new values which are used for the authentication process.

When the setup phase is executed, the notification ("setup phase in progress") will be shown in the transcript window. This notification helps the designers to identify the phase model either it is in setup phase or authentication phase. Once the setup phase process has finished, all the IoT devices have registered their ID numbers to the servers and get the proper response values to from the servers. The next process of the proposed architecture is to perform the authentication phase for the stored IoT devices. For performing the authentication phase, the phase signal turns into 1. So, the stored setup phase values are progressed to the next step for getting the authentication access from the server. Device nonce and server nonce values are given the IoT devices from the server. After receiving the device and server nonce, each IoT device can check the values whether the respective device nonce is available in the received server and device nonce. Each IoT device segregates the server nonce which helps to generate the new key hash response and helper data. These generated new values are given to the server for getting the authentication.

In the server, the Key hash response contains the device nonce and server nonce. If the IoT device nonce is present in the key hash function, then the server provides the authentication for the respective IoT devices. If else, the authentication is not provided to the devices from the server. Based on the device and server nonce, the authentication has provided by the server which helped to access output. If authentication phase is executed, the transcript window shows like "Authentication Successful" and all the new values are updated in the server as well as respective IoT devices. These entire architectures are underdone the hardware utilization and authentication parameter performances which are explained as follows:

4.2 FPGA results and analysis

In this section, the hardware utilization of the TFA-RPUF-IoT architecture and TFA-PUF-IoT architecture [16] are given in detail. Both the architectures are implemented in Verilog language and the results are tabulated in Table 2. The comparison of the conventional IoT PUF authentication [16] and the proposed RPUF authentication architecture is given in Table 2. In Spartan 6 FPGA, 21.42% of Slices, 13.88% of LUT, 8.69% of Flip flops are reduced in TFA-RPUF-IoT compared to TFA-PUF-IoT [16]. Additionally, the

Table 2. Comparison of hardware utilization for a different architecture

Target Family	Device	Speed and package	Architecture	Slice	LUT	Flip flop	Frequency (MHz)	Delay (ns)
Spartan 6	XC6SLX9	-3 and CSG324	TFA-PUF-IoT [16]	14	36	46	441.92	2.263
			TFA-RPUF- IoT	11	31	42	510.06	1.961
Virtex 6	XC6VCX75T	-2 and FF484	TFA-PUF-IoT [16]	17	34	46	713.59	1.401
			TFA-RPUF- IoT	10	25	42	739.61	1.352

Table 3. Comparison of hardware utilization for conventional and proposed PUF model

Target family	Device	Speed and package	Module	Slice	LUT	Flip flop	Frequency (MHz)	Delay (ns)
Spartan 6	XC6SLX9	-3 and CSG324	PUF	4	10	8	663.46	1.507
			RPUF	3	8	8	696.45	1.436
Virtex 6	XC6VCX75T	-2 and FF484	PUF	3	10	8	1225.41	0.816
			RPUF	6	8	8	1243.54	0.804

Table 4. Comparison of two factor authentication performances for a conventional model with RPUF-IoT

Security property	Amin [13]	Han [14]	Xie [15]	TFA-PUF-IoT [16]	TFA-RPUF-IoT
Resilience to the impersonation attack	Yes	Yes	Yes	Yes	Yes
Anonymity and un traceability	Yes	No	Yes	Yes	Yes
Resilience to the password guessing attack	No	Yes	Yes	Yes	Yes
Prevents clock synchronization problem	No	Yes	No	Yes	Yes
Device security	No	No	No	Yes	Yes
Deployed security algorithm	ECC	ECC	ECC	PUF and FE	RPUF-FE
Random response for every clock cycle	No	No	No	No	Yes

Table 5. Comparison of security performances for a conventional model with RPUF-IoT

Comparison matrices	Aman [12]	TFA-PUF-IoT [16]	TFA-RPUF-IoT
Mutual authentication	Yes	Yes	Yes
Two factor secrecy	No	Yes	Yes
Privacy of the IoT devices	No	Yes	Yes
Consideration of noise in the PUF	No	Yes	Yes
Protection against physical attacks	Yes	Yes	Yes
Random response for every clock cycle	No	No	Yes

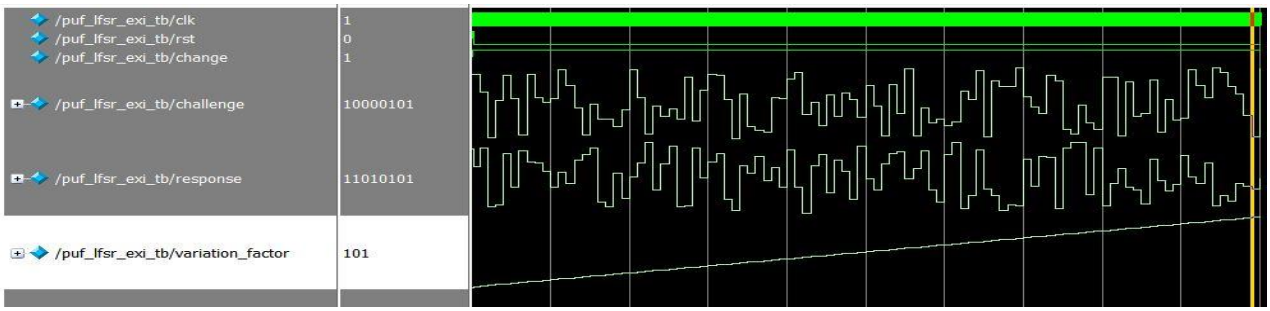

Figure. 8 Variation factor of conventional PUF model

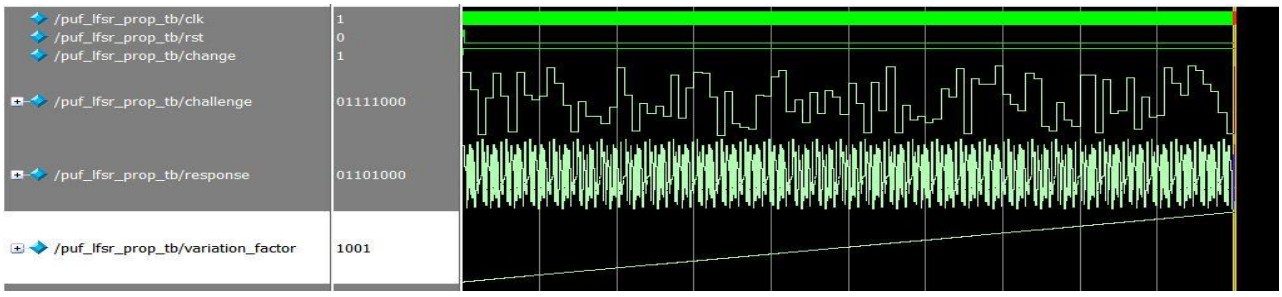

Figure. 9 Variation factor of reconfigurable PUF model

TFA-RPUF-IoT method achieved 13.35% of improved operating frequency which denoted the entire system speed.

Moreover, the separate module of PUF and RPUF module FPGA results are given in Table 3. The reconfigurable PUF consumed 6 slices, 8 LUTs, and 8 flip flops which are less compared to the existing PUF. In Virtex 6, RPUF occupied 6 slices which is higher than PUF. Because, the slice element utilization is more in XC6VCX75T device. But, the LUT, frequency, and delay performance are

improved in RPUF compared to PUF. The hardware utilization is evaluated for Virtex 6 and Spartan 6 FPGA devices which results are given in Table 2 and Table 3. The speed grade and the packages of the FPGA devices also given in tables.

Comparison of two factor authentication performances and the security performances for a conventional model with RPUF-IoT are given in Table 4 and Table 5 respectively. In this table, all the comparison models [12-16] are compared with the proposed PUF IoT architecture. In TFA, device security, secure algorithm, attacks, clock synchronization is verified and the results (Yes or No) are tabled in respective portions. In the security performances, mutual authentication, TF secrecy, PUF model, safety against the attacks are concentrated and the results are verified. In both cases, the random response for every clock cycle parameter has analyzed for PUF-IoT [16] and RPUF-IoT. Among all the conventional methods [12-16], proposed RPUF-IoT architecture only produces the random data for every clock cycle, even the same number of input is given in the next clock cycle. Due to this randomness, the unauthorized person can't retrieve the confidential information from the IoT devices.

4.3 Security analysis

In this section, the different security analysis is verified for the proposed architecture. Due to this security analysis, the proposed TFA-RPUF-IoT architecture has achieved more confidentiality than existing TFA-PUF-IoT architecture [16].

4.3.1. Session key agreement

After processing the mutual authentication, both the IoT devices and the server have shared the identical session key. In this scenario, the side channel attacks are affected in the transmission line. In the conventional method, the architecture doesn't care about the side channel attack.

But, in the proposed architecture, if the side channel attack is occurred, the secret key agreement is not secured due to the session key corruption. In case, any one of the secret key has changed, the server doesn't give the authentication for the IoT devices. Therefore, the proposed scheme is able to provide session key agreement.

The variation of the PUF and RPUF models are shown in Fig. 8 and Fig. 9. Based on the challenge value, both PUF and RPUF have generated the response values. The variation factor of the existing model is "101". But, RPUF has more variation factor

Table 6. Response values for the same challenge

Family	Clock	Challenge	Prop-PUF	Exi-PUF
Spartan 6	0	182	182	182
	1	182	91	211
	2	182	45	211
	3	182	22	211
	4	182	11	211
	5	182	5	211
	6	182	130	211
	7	182	65	211
	8	182	160	211
	9	182	90	211
	10	182	40	211

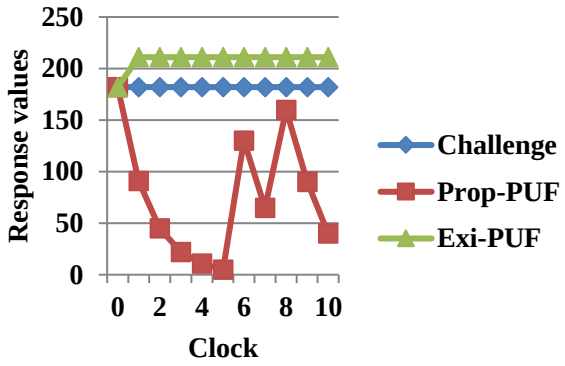

Figure. 10 Response generation based on the challenge value

(1001) which indicates the randomness of the response values.

4.3.2. Protection against related key attack

The unknown/partially identified key caused the related key attack in the IoT device communication. Since, the related key attack mainly depends on the less diffusion and symmetric property over the key expansion block. The proposed RPUF obtains the faster and nonlinear diffusion of the secret key which used to avoid the related key attack.

The same challenge has given for every clock cycle for RPUF and PUF model. The same response has generated for the PUF model, but, different kind of response values are generated in RPUF. The response values are given in Table 6 and the pictorial representation of the PUF and RPUF response value is shown in Fig. 10.

5. Conclusion

The IoT device authentication is very essential to secure the confidential data and device to device communication. In this work, reconfigurable PUF has been designed using the LFSR circuit model to generate random numbers. In the LFSR circuit, the

feedback connection has been connected differently in the registers for every clock cycle. With the help of challenge value, the response has been generated for all the IoT devices. Based on this reconfigurable operation, the unpredictable random data has been generated and perform the two factor authentication. The setup and authentication phase have been implemented in the Spartan 6 and Virtex 6 FPGA platform using Verilog language. Based on the key hash function values, the IoT device gets the authentication from the server. In Spartan 6 FPGA, 21.42% of Slices, 13.88% of LUT, 8.69% of Flip flops are reduced in the TFA-RPUF-IoT Architecture compared to TFA-PUF-IoT Architecture. Additionally, the proposed method achieved 13.35% of improved operating frequency which denoted the entire system speed. In the future, multi reconfigurable PUF will be used for each and every IoT device which increase the security level further. In future, the security in the server can be improved by using the data flipping technique which used to convert the linear generation of random number into non-linear generation.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

The paper conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing—original draft preparation, writing—review and editing, visualization, have been done by 1st author. The supervision and project administration, have been done by 2nd author.

References

[1] T. Xu, J. B. Wendt, and M. Potkonjak, "Security of IoT Systems: Design Challenges and Opportunities", In: *Proc. of IEEE/ACM ICCAD*, pp. 417-423, 2014.

[2] G. Woo, P. Kheradpour, D. Shen, and D. Katabi, "Gartner Says the Internet of Things will Transform the Data Center", *Gartner*, 2014.

[3] P. Porambage, C. Schmitt, P. Kumar, A. Gurtov, and M. Ylianttila, "Two-phase Authentication Protocol for Wireless Sensor Networks in Distributed IoT Applications", In: *Proc. of IEEE WCNC*, pp. 2728-2733, 2014.

[4] P. N. Mahalle, N. R. Prasad, and R. Prasad, "Threshold Cryptography based Group Authentication (TCGA) Scheme for the Internet of Things (IoT)", In: *Proc. of IEEE VITAE*, pp. 1-5, 2014.

[5] V. Petrov, S. Edelev, M. Komar, and Y. Koucheryavy, "Towards the Era of Wireless Keys: How the IoT Can Change Authentication Paradigm", In: *Proc. of the IEEE WF-IoT*, pp. 51-56, 2014.

[6] R. Amin and G. P. Biswas, "Anonymity preserving secure hash function based authentication scheme for consumer USB mass storage device", In: *Proc. of the Computer, Communication, Control and Information Technology (C3IT'15)*, pp. 1–6, 2015.

[7] R. Amin, "Cryptanalysis and an effcient secure ID-based remote user authentication scheme using smart card", *International Journal of Computer Applications*, Vol. 75, No. 13, pp. 43–48, 2013.

[8] R. Amin and G. P. Biswas, "Remote Access Control Mechanism Using Rabin Public Key Cryptosystem", In: *Proc. of the Information Systems Design and Intelligent Applications*, pp. 525–533, 2015.

[9] R. Amin and G. P. Biswas, "A secure three-factor user authentication and key agreement protocol for TMIS with user anonymity", *Journal of Medical Systems*, Vol. 39, No. 8, pp. 1–19, 2015.

[10] S. H. Islam, "Design and analysis of a three party password based authenticated key exchange protocol using extended chaotic maps", *Information Sciences. An International Journal*, Vol. 312, pp. 104–130, 2015.

[11] R. Amin, and G. P. Biswas, "Design and Analysis of Bilinear Pairing Based Mutual Authentication and Key Agreement Protocol Usable in Multi-server Environment", *Wireless Personal Communications*, Vol. 84, No. 1, pp. 439–462, 2015.

[12] M. N. Aman, K. C. Chua, and B. Sikdar, "Mutual authentication in IoT systems using physical unclonable functions", *IEEE Internet of Things Journal*, Vol. 4, No. 5, pp. 1327-1340, 2017.

[13] R. Amin, S. K. Islam, M. K. Khan, A. Karati, D. Giri, and S. Kumari, "A two-factor RSA-based robust authentication system for multi server environments", *Security and Communication Networks*, 2017.

[14] J. Qu, and X. L. Tan, "Two-factor user authentication with key agreement scheme based on elliptic curve cryptosystem", *Journal of Electrical and Computer Engineering*, 2014.

[15] Q. Xie, D. S. Wong, G. Wang, X. Tan, K. Chen, and L. Fang, "Provably secure dynamic ID-

based anonymous two-factor authenticated key exchange protocol with extended security model", *IEEE Transactions on Information Forensics and Security*, Vol. 12, No. 6, pp. 1382-1392, 2017.

[16] P. Gope and B. Sikdar, "Lightweight and privacy-preserving two-factor authentication scheme for IoT devices", *IEEE Internet of Things Journal*, Vol. 6, No. 1, pp. 580-589, 2018.

[17] D. H. Bui, D. Puschini, S. Bacles-Min, E. Beigné, and X. T. Tran, "AES data path optimization strategies for low-power low-energy multi security-level internet-of-things applications", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 25, No. 12, pp. 3281-3290, 2017.

[18] H. Zakaria, N. A. A. Bakar, N. H. Hassan, and S. Yaacob, "IoT Security Risk Management Model for Secured Practice in Healthcare Environment", *Procedia Computer Science*, Vol. 161, pp. 1241-1248, 2019.

[19] M. Al-Asli, M. E. Elrabaa, and M. Abu-Amara, "FPGA-Based Symmetric Re-Encryption Scheme to Secure Data Processing for Cloud-Integrated Internet of Things", *IEEE Internet of Things Journal*, Vol. 6, No. 1, pp. 446-457, 2018.

[20] M. Wazid, A. K. Das, V. Odelu, N. Kumar, M. Conti, and M. Jo, "Design of secure user authenticated key management protocol for generic IoT networks", *IEEE Internet of Things Journal*, Vol. 5, No. 1, pp. 269-282, 2017.