Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 8876096, 12 pages
https://doi.org/10.1155/2022/8876096

Research Article

Renovated XTEA Encoder Architecture-Based Lightweight Mutual Authentication Protocol for RFID and Green Wireless Sensor Network Applications

Manikandan Nagarajan[1], **Muthaiah Rajappa**[1], **Yuvaraja Teekaraman**[2], **Ramya Kuppusamy**[3] and **Amruth Ramesh Thelkar**[4]

[1]*School of Computing, SASTRA Deemed University, 613 401, Thanjavur, India*
[2]*Department of Electronic and Electrical Engineering, The University of Sheffield, Sheffield, S1 3JD, UK*
[3]*Department of Electrical and Electronics Engineering, Sri Sairam College of Engineering, 562 106, Bangalore City, India*
[4]*Faculty of Electrical & Computer Engineering, Jimma Institute of Technology, Jimma University, Ethiopia*

Correspondence should be addressed to Yuvaraja Teekaraman; yuvarajastr@ieee.org
and Amruth Ramesh Thelkar; amruth.rt@gmail.com

Received 10 January 2022; Revised 7 February 2022; Accepted 19 February 2022; Published 10 March 2022

Academic Editor: Rashid A Saeed

Wireless sensor networks find applications everywhere in day to day activities right from attendance entry systems to healthcare monitoring systems. The evolution of the Internet of Things (IoT) as the Internet of Everything (IoET) makes the wireless sensor network omnipresent and increases the use of Radio Frequency Identification (RFID) for the proper identification of devices and sensor nodes which are mostly battery operated. As technology evolves, security threats also increase rapidly. This mandates a strong and energy-efficient green solution. This work attempted to address these issues by effectively deploying the lightweight encryption scheme called Extended Tiny Encryption Algorithm (XTEA). Though the XTEA is lightweight and famous, it is commonly known for various attacks. Our work patches the security threats in the XTEA by applying domain-specific customization, random number utilization, and undisclosed key renewal techniques. Two custom Renovated XTEA Mutual Authentication Protocol (RXMAP) encoder architectures, namely, RXMAP-1 and RXMAP-2, are proposed based on the replacement of accurate computational blocks with approximate blocks. The proposed RXMAP protocol is evaluated for its computational and storage overhead and verified against various security threats using BAN logic formal verification and informal verification. The proposed encoder architectures are simulated for functional verification, and ASIC implementation is done with a 132 nm process node. ASIC implementation results show that the proposed designs RXMAP-1 and RXMAP-2 occupy 53.11% and 53.31% lesser area compared to XTEA I and 52.97% and 53.18% lesser area compared to XTEA II implementation. The total power consumed by the proposed encoder architectures RXMAP-1 and RXMAP-2 is 68.76% and 71.64% lesser than XTEA II implementation, respectively, while maintaining the equal throughput.

1. Introduction

Advancement in technology facilitates people to enjoy wireless devices with smart sensors. Wireless sensor networks (WSN) are finding applications in various places, viz, health monitoring, IoT (Internet of Things), logistics, and warehouses [1]. As the application areas are getting wider, the privacy and security issues are also getting complex. So, securing information exchanged in the networks is the need of the hour. There are many algorithms available in the literature for securing the communication among the sensors and systems. Most of the secured algorithms are complex and computation intensive in nature [2]. Wireless sensor networks are composed of Radio Frequency Identification (RFID) tags and low power-consuming sensors [3]. Deploying complex security algorithms is not a feasible solution for

the WSN components. This leads to the lightweight cryptography domain. Lightweight encryption schemes utilize fundamental operations like addition, rotation, and xor (ARX) operations [4]. The ARX operations are alone not sufficient for securing the data and communication. There is a chance of attack, and some complex encryption schemes are essential to protect the sensors and systems. The encryption scheme deployed for WSN is the tradeoff between the complexity and the performance [5, 6].

There are plenty of research works which satisfy the requirements of the WSN in the literature, namely, PRESENT [7], PRINCE [8], TEA [9], XTEA [10], XXTEA [11], LEA [12], HIGHT [13], and AES [14]. Each encryption scheme features and fits best for the unique applications based on the requirements and the nature. In the WSN, mutual authentication is the key part in the node registration and communication phase. In IoT applications, also, authentication is pivotal in the perception layer to address the devices and send the collected data from edge to server for processing [15]. Mutual authentication uses symmetric key encryption schemes [16, 17]. The edge sensors and RFID tags are the components of the perception layer. A separate subfield in cryptography is prevailing to establish mutual authentication between the nodes and the servers. Standard block encryption schemes need to be customized to make it fit in the RFID or sensor nodes. [18–24] implements the lightweight encoding schemes for mutual authentication.

Mutual authentication protocols (MAP) are slightly unique from conventional encryption schemes. Nondisclosure of the key is vital in the MAP which does not exactly require the conventional encryption algorithms. This work is attempted to explore the domain-specific requirements and deploy the appropriate things to ensure the security of authentication. Domain-specific architectures (DSA) are customized architectures based on the need of the specific domains and their application. DSA is a next disruptive technology alternate to the parallelism and pipelining to attain the performance [25]. In this work, we targeted the Extended Tiny Encryption Algorithm (XTEA) which is cracked by attacks such as related key attacks [26] and meet-in-the-middle attack [27].

All the encryption algorithms do have its security breaking point. This work proposes a renovated XTEA (RXTEA) which patches the reported security problems in the XTEA and optimizes the area of implementation and power requirements [6]. The requirements of the passive RFID EPC class 1 generation 2 tags [28] are that the design should have less than 10000 gate equivalent which includes security and functional handling chip components and circuit should be operated with low power which the tag receives from the reader as a trigger. This work RXTEA addresses the above mentioned issues without compromising any of the requirements of RFID EPC class 1 generation 2 tags, and it is a suitable candidate for the mutual authentication in passive tags and green wireless sensor network applications.

The proposed protocol architecture is implemented in ASIC (Application Specific Integrated Circuit) design flow with the technology process (130 nm).

1.1. The Significant Contribution of the Proposed Work. The following are the significant contributions of the proposed work:

(1) Domain-specific architecture customization for XTEA is done to make it fit for low power applications

(2) Add the features to produce delta in a random fashion which makes XTEA to withstand for various attacks

(3) Key is not shared in any form in the information used for the mutual authentication process

(4) Communication messages used in the authentication process are much lesser in size than in key size, so it is impossible to guess the key and robust against adversaries

(5) Key and identity are updated after each successful session thereby fortifying against desynchronization attack models

1.2. Organization. The remainder of the manuscript is consolidated with related works in Section 2, protocol design in Section 3, and security analysis in Section 4. Evaluation of the protocol is done in Section 5, and the ASIC implementation of the proposed protocol architecture is briefed in Section 6. Section 7 consolidates the contribution and concludes this paper.

2. Related Works

TEA family algorithms attract the interest of the researchers, and it has been evolved from the year of its invention. Few of the works modified the TEA computation to strengthen the security, and many works in the literature are focused on the performance improvement of it to get better throughput. Some of the closely related recent works which targeted RFID and IOT applications are presented below.

Mishra and Acharya [4] proposed the high throughput architectures of TEA family for high speed IoT and RFID applications. Pipelined implementation of TEA, XTEA, and XXTEA is done to improve the throughput of computation of the encryption process. By parallelizing the computation process, significant throughput improvement is attained at the expense of the more resource utilization. The authors also implemented a hybrid method by combining the TEA architecture. The designed encoders are implemented in Field Programmable Gate Array (FPGA) and Application Specific Integrated Circuit (ASIC) platform.

Kella et al. [29] attempted to modify the XXTEA architecture to make it fit for RFID tags. A serial architecture for XXTEA-192 block cipher can perform ARX operations in each clock. This works on variable length ciphers with the minimum block size of 32-bits and its multiple so that it is extended for a higher length. The design is implemented on FPGA to calculate throughput parameters.

Ragab et al. [30] claimed higher throughput while maintaining the security by adding S-box to the computation process. To add an additional strength key generation is done

with the chaotic computation method. The designed M-XXTEA is compared with AES, and it is producing 57% better efficiency than AES.

Yeo et al. [31] created the IP core for the corrected block TEA to integrate and reuse the design in various IoT applications. The developed block TEA is able to encrypt the data width from 64 bits to 256 bits in multiple of 32 bits. The IP core is developed in verilog-HDL and implemented in Vertex 4 FPGA. The design has exhibited good throughput performance.

Anusha et al. [32] drafted a pipelined architecture to implement an XTEA and TEA for the parallel computation while keeping the resources minimum to make them fit in the RFID authentication process. The developed architecture implemented in the artix-7 FPGA and computation performance is evaluated.

Khan and Zhu [33] presented a secure symmetric key-based mutual authentication protocol for the RFID authentication. In this, they used XTEA for encrypting the communication messages. An effective key updating mechanism is adopted to make the protocol stand against replay, eavesdropping, and man-in-the middle attack while keeping the computing cost minimum.

Khan et al. [34] deployed a NIOS II processor to implement the XTEA encryption process for the RFID mutual authentication purpose. The subkey of the RFID tag is updated in a random way that makes the developed protocol stable against various attacks.

3. Proposed Renovated XTEA Mutual Authentication Protocol Architecture

In this section, we propose a renovated XTEA (RXTEA) encoder architecture and a secure RXTEA-based mutual authentication protocol (RXMAP). The following changes are made in the XTEA architecture to reduce the computation and to make it robust against related key attack [26] and meet-in-the-middle attack [27]:

(1) Domain-specific customization is done to the XTEA architecture. For encryption purpose, accurate adders are not necessary, so it is replaced with approximate adders

(2) Constant delta value may lead to the guess of internal computations. It is replaced with the delta computation block which computes fresh delta for every successful authentication with the random numbers

(3) Only one-half of the ciphered text is used as a message to authenticate the server and tag. This makes it almost impossible to guess the key value

(4) Key value is updated at both server and tag side with the values internally computed. It is highly impossible to guess the new key

3.1. Renovated XTEA Encoder Architecture. The proposed RXMAP encoder architecture is presented in Figure 1. The notations used in the proposed architecture and protocol are listed in Table 1. Inputs for the RXMAP encoder architecture are 128-bit key, 64-bit IDS, 32-bit RS, and RT. With the received inputs and the computed delta value, both tag and server start to compute the cover-coded password, i.e., cipher text of two 32-bit data as in equation (1). In the cover-coded password computation, ARX process is involved. In that instead of accurate adder, approximate adders are used. Based on the approximate adders used, two proposed configurations, namely, RXMAP-1 and RXMAP-2, are constructed. Protocol governing control block produces CTRL signal to have control over the trigger generation and new IDS, key update process.

$$\text{Delta} = \{\text{RT}[31:16]||\,|\text{RS}[15:0]\}\{\text{RS}[31:16]||\text{RT}[15:0]\}. \tag{1}$$

3.1.1. Approximate Adders for RXMAP Encoder. Adders in the XTEA architecture are used to repeatedly sum up the translated plaintext with the key. Here, the concept behind the addition process is to change the plaintext value to some other value. Hence, an accurate adder is necessary to do the translation process, because they are computation intensive process in the repeated rounds. As approximate adders are noncommutative, the same logic would not work for the different domains where decryption is also mandatory. In the case of mutual authentication process, decryption is not essential and that is specified as domain-specific customization in our work.

Two 32-bit adders are proposed in this work as follows:

(1) OR only adder (OOA)—two 32-bit data are added with two inputs OR gate bit by bit

(2) XOR only adder (XOA)—two 32-bit data are added with two inputs XOR gate bit by bit

These adders are inspired from the lower part OR adder (LOA) [35]. LOA is an area and computation-efficient implementation of the adder among the various approximate adders in the literature ever found. Still, the research is progressed in the same approach for striving better accuracy [36–39]. Our work is unique in this; we hardly bother about accuracy since in this encryption domain adder is used for translation purpose.

The encoder architecture uses XOA and OOA with its addition processes called RXMAP-1 and RXMAP-2, respectively.

3.2. Renovated XTEA-Based Mutual Authentication Protocol. Our proposed Renovated XTEA Mutual Authentication Protocol is shown in Figure 2. The proposed protocol architecture is formulated for encrypting a block of 64-bit plaintext with the 128-bit key. Mutual authentication is performed by considering random numbers RT and RS. The random numbers and the key are manipulated by the proposed RXMAP encoder to produce cover-coded passwords, and it proceeds as per the protocol flow described below,

There is a certain initial set of assumptions to start the proposed protocol, and it is listed as follows:

(1) Both the tag and server knows the pseudonym IDS of the tag and the key used for authentication

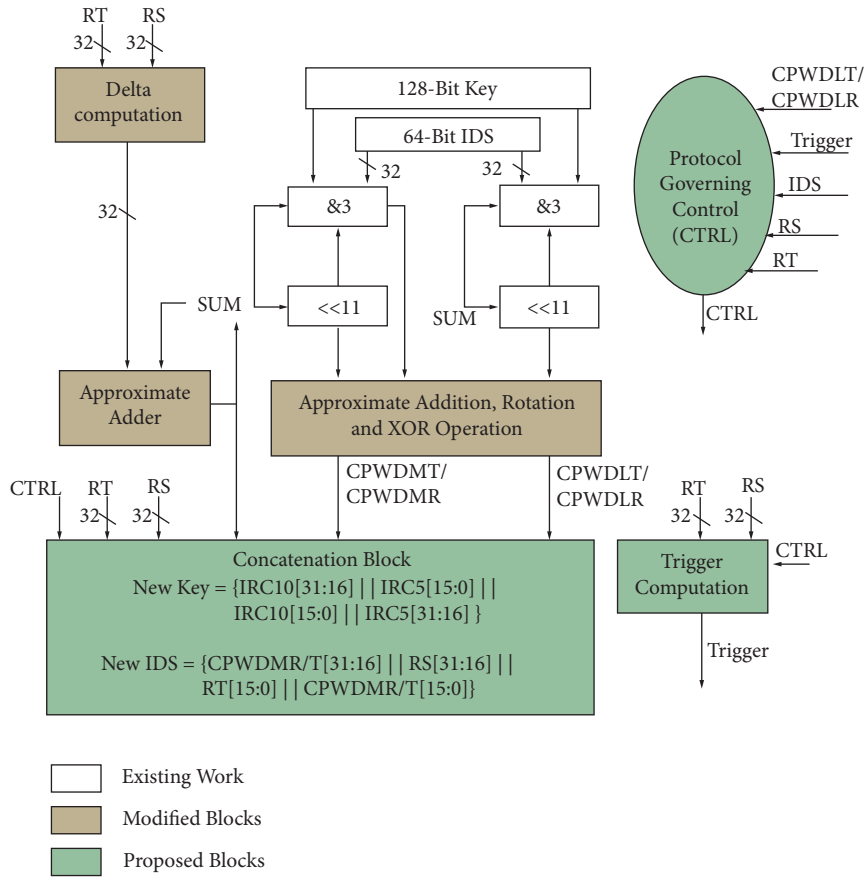


Figure 1: Proposed RXMAP encoder architecture for both the tag and reader.

Table 1: Nomenclature and symbols used in the proposed protocol.

Notions	Explanations
Req	Reader's request to the tag
ID	Identity number of the RFID tag
IDS	The pseudonym of the ID
RT	Tag's random number
RS	Reader's random number
CPWDMT	MSB 32 bits of cover-coded password generated through the proposed XTEA on the tag side
CPWDLT	LSB 32 bits of cover-coded password generated through the proposed XTEA on the tag side
CPWDMR	MSB 32 bits of cover-coded password generated through the proposed XTEA on the reader side
CPWDLR	LSB 32 bits of cover-coded password generated through the proposed XTEA on the reader side
| |	Concatenation operation
&	AND operation
^	XOR operation
<<	Left rotation
CTRL	Control signal generated by the protocol governing control module
IRC	Intermediate result in the multiple round encryptions of cover-coded password
IRC_5	Intermediate result of cover-coded password at the end of the 5^{th} round
IRC_{10}	Intermediate result of cover-coded password at the end of the 10^{th} round

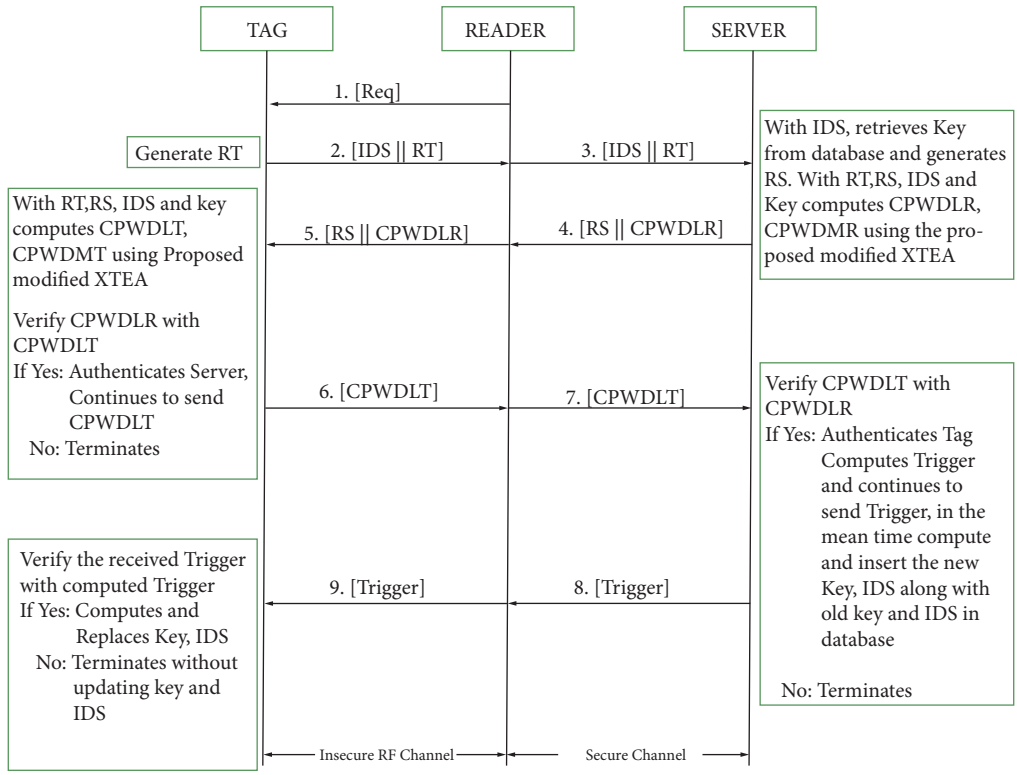


Figure 2: The proposed lightweight Renovated XTEA Mutual Authentication Protocol (RXMAP).

(2) The tag and reader have the ability to produce the random numbers

The step by step elucidation of the proposed RXMAP protocol is as follows:

(Step 1) The reader commences the protocol by sending a Req message.

(Step 2) Tag on getting Req notification generates random number RT and sends generated RT by concatenating it with its pseudonym IDS as a response to the reader Req.

(Step 3) Upon receiving IDS||RT, the reader forwards it to the server.

(Step 4) With the received IDS, the server starts searching the database for the matches. If it is found, the corresponding key will be extracted and reader side random number (RS) will be generated to use it in further steps.

With the received RT, generated RS, and extracted Key, the server computes cover-coded passwords/keys named CPWDLR and CPWDMR by using RXMAP encoder. The server sends the concatenated values RS||CPWDLR to the reader.

If the received IDS is not matched with the old or new entries in the database, then, it considers the tag as an invalid one and terminates the communication. If it matches with the new entry, then, the server deletes old entry in the database and proceeds further. Otherwise, it will check the match with the old entry; if it matches, then it senses the attack in between during the preceding communication and deletes the new entry in the database and continues the authentication process as a fresh one with the old entry. This ensures the tag's synchronization with the server all the time and also optimizes the server's storage space.

(Step 5) The reader redirects the received RS||CPWDLR to the tag.

(Step 6) With the received RS, generated RT, and the Key, the tag also computes cover-coded passwords/keys named CPWDLT and CPWDMT using RXMAP encoder.

Upon completion of the CPWDLT computation, the tag verifies CPWDLR with CPWDLT; if it is matched, then, the tag authenticates the server, and as an acknowledgement to the server authentication, the tag sends the cover-coded CPWDLT to the reader.

(Step 7) The reader redirects the received CPWDLT to the server.

(Step 8) The server verifies the received CPWDLT with CPWDLR for similarity by XNOR process; if it is equal, then the server authenticates the tag and computes the trigger value internally as per

$$\text{Trigger} = \{\text{RS}[31:16]||\,|\text{RT}[15:0]\}\&\{\text{RT}[31:16]||\text{RS}[15:0]\}. \quad (2)$$

Once the server authenticated the tag, the server sends the trigger as an acknowledgement of authentication to the tag through the reader to initiate the IDS and key replacement at the tag side. During that process, the server computes new IDS and key as per equations (3) and (4) and inserts it in its database for the corresponding tag while keeping the IDS and key used in the present session as old IDS and key. If authentication failed, the server terminates the communication without updating its database for new IDS and key.

The computation of tag's new pseudonym IDS and the corresponding key value for further processing is as follows:

$$\text{New Key} = \{\text{IRC10}[31:16]||\text{IRC5}[15:0]||\text{IRC10}[15:0]||\text{IRC5}[31:16]\}, \quad (3)$$

$$\text{New IDS} = \left\{\frac{\text{CPWDMR}}{\text{T}[31:16]}\middle|\middle|\text{RS}[31:16]||\text{RT}[15:0]||\middle|\frac{\text{CPWDMR}}{\text{T}[15:0]}\right\}. \quad (4)$$

(Step 9) The reader redirects the received trigger to the tag.

After receiving the trigger value, the tag computes the trigger value internally as per equation (2) and compares it with the received trigger value. If it matches, then, the tag replaces its IDS and key value as in equations (3) and (4) and establishes the connection. Else, it terminates the communication without altering its IDS and key values.

4. Security Analysis

In this clause, the presented RXMAP protocol is investigated for security analysis. The proposed RXMAP protocol is examined for formal and informal analyses. For formal analysis, Burrows-Abadi-Needham (BAN) logic is formulated. The informal analysis is performed on the following facets: security requirements, security threats, and cryptanalysis models.

4.1. Formal Analysis. Formal analysis is the art of analyzing a deficiency in the designed protocols that are not directly apparent through cryptanalysis. A five-step procedure was suggested for applying the formal verification functions in the RFID designs [40]. Formal analysis works under the principle of belief analysis, and based on that request/response communication between the tag and the reader is evaluated. This function examines the plain message formation and succession of communication between the transaction groups to assess the protocol's potential on epitomized level efficiently. The BAN logic is deployed for verifying our RXMAP protocol.

4.1.1. BAN Logic Analysis. This logic was formulated by Burrows et al. [41]. This logic was used to rationale authentication protocols among contenders in a distributed processing system. The precepts used in BAN logic analysis [42] are explained below with equations. The notions and corresponding descriptions of the notions that are used for BAN logic analysis [43] are tabulated in Table 2.

The BAN logic rules have been applied for the proposed RXMAP protocol for the abstract level evaluation. These rules are tabulated in Table 3.

From [42], goals for achieving mutual authentication for a protocol are derived. In the proposed protocol, the mutual authentication phase takes place only after the translated cover-coded password exchange between the tag and the reader. A successful translated cover-coded password exchange indicates a successful protocol run for one session. Therefore, if mutual authentication is verified through BAN logic, then, the protocol run is also verified simultaneously. Hence, these goals are targeted to achieve the protocol evaluation. Mutual authentication is accomplished between two principles A and B if K is existing such that

$$\begin{aligned} A| &\equiv A \overset{K}{\leftrightarrow} B, \\ B| &\equiv A \overset{K}{\leftrightarrow} B, \\ A|{\equiv}B| &\equiv A \overset{K}{\leftrightarrow} B, \\ B|{\equiv}A| &\equiv A \overset{K}{\leftrightarrow} B. \end{aligned} \quad (5)$$

The first two are believed to be essential for any protocol to be mutually authenticated. Assumptions, according to the proposed protocol in order to acquire the above goals, are formulated as mentioned below. Besides, it is assumed that the tag has RS, RT, and PWD (KEY) and the reader has RS , RT, and PWD (KEY). The server is considered a secure channel and hence not included in the analysis. PWD is retrieved with pseudonym IDS which is updated after each session. T principle denotes the tag, and R principle denotes the reader.

$$\begin{aligned} &\text{A1} : T \Rightarrow \text{RT}, \\ &\text{A2} : T| \equiv \#(\text{RT}), \\ &\text{A3} : R \Rightarrow \text{RS}, \\ &\text{A4} : R| \equiv \#(\text{RS}), \\ &\text{A5} : T \equiv \left(T \overset{\text{PWD}}{\leftrightarrow} R\right), \\ &\text{A6} : R| \equiv \left(T \overset{\text{PWD}}{\leftrightarrow} R\right). \end{aligned} \quad (6)$$

Messages that are communicated betwixt the tag and the reader in the RXMAP run are as follows:

Message 1: $R \longrightarrow T : (\{\text{CPWDLR}\}_{\text{PWD},\text{RT}}, \{\text{Trigger}\}_{\text{RT.}})$

Message 2: $T \longrightarrow R : (\text{IDS}, \{\text{CPWDLT}\}_{\text{PWD},\text{RS}})$

From the reader to tag, $CPWDLR$ is an encrypted message transmitted. This $CPWDLR$ is encrypted using

Table 2: Notions and descriptions used for BAN logic analysis.

Notions	Explanations
P or Q	P and Q are the principal that refers to a sender or receiver. For instance, the tag or the reader
$P \mid \equiv X$	P believes the statement X
$P \triangleleft X$	P sees the statement X
$\#(X)$	The formula X is fresh
$P \mid \sim X$	P once said the statement X
(X, Y)	Either formula X or Y is one section of formula (X, Y)
$P \mid \Rightarrow X$	P has control over statement X
$\langle X \rangle_Y$	X is integrated with formula Y
$P \overset{K}{\leftrightarrow} Q$	Shared key K is used for communication between P and Q. K will be known only to these two entities
NKEY	This denotes the new key generated for further updating and processing

Table 3: The BAN model rules applied for the proposed RXMAP protocol.

BAN logic rules	Rules applied for the proposed RXMAP protocol
Message-meaning rule (M)	Case (1) $T \mid \equiv R \overset{\text{RS}}{\leftrightarrow} T, T \triangleleft \{\text{CPWDLR}\}_{\text{RM}}/T \mid \equiv R \mid \sim \text{CPWDLR}$ Case (2) $R \mid \equiv T \overset{\text{RT}}{\leftrightarrow} R, R \triangleleft \{\text{CPWDLT}\}_{\text{RT}}/R \mid \equiv T \mid \sim \text{CPWDLT}$
Nonce-verification rule (NV)	Case (1) $T \mid \equiv \#(\text{CPWDLR}), T \mid \equiv R \mid \sim \text{CPWDLR}/T \mid \equiv R \mid \equiv \text{CPWDLR}$ Case (2) $R \mid \equiv \#(\text{CPWDLT}), R \mid \equiv T \mid \sim \text{CPWDLT}/R \mid \equiv T \mid \equiv \text{CPWDLT}$
Jurisdiction rule (J)	Case (1) $T \mid \equiv R \Longrightarrow \text{CPWDLR}, T \mid \equiv R \mid \equiv \text{CPWDLR}/T \mid \equiv \text{CPWDLR}$ Case (2) $R \mid \equiv T \Longrightarrow (\text{CPWDLT}), R \mid \equiv T \mid \equiv \text{CPWDLT}/R \mid \equiv \text{CPWDLT}$
Freshness rule (F)	Case (1) $T \mid \equiv \#(\text{RT})/T \mid \equiv \#(\text{CPWDLT}, \text{RT})$ Case (2) $R \mid \equiv \#(\text{RS})/R \mid \equiv \#(\text{CPWDLR}, \text{RS})$
Other inference rules (I)	Case (1) $T \triangleleft (\text{CPWDLR}, \text{RS})/T \triangleleft (\text{CPWDLR})$ Case (2) $R \triangleleft (\text{CPWDLT}, \text{RT})/R \triangleleft (\text{CPWDLT})$

PWD, *RT*, and *RS*. Trigger is sent from the reader to tag to initiate the key update process. Trigger is interleaved and translated version of *RT* and *RS*. Similarly, from the tag to reader *CPWDLT* is the encrypted message that is encrypted using *PWD*, *RT*, and *RS*.

As mentioned earlier, the RXMAP protocol should satisfy the following goals for mutual authentication analysis:

$G1$: $R \mid \equiv T \mid \equiv (T \overset{\text{NKEY}}{\leftrightarrow} R)$

G2: $T \mid \equiv R \mid \equiv (T \overset{\text{NKEY}}{\leftrightarrow} R)$

Based on the above assumptions and rules, it is witnessed that the RXMAP protocol attains the goals for mutual authentication and they are provided with multiple steps below.

S1: According to *Message* 1 and rule I (1), $T \triangleleft \{\text{CPWDLR}\}_{\text{PWD,RT,RS}}$

By assumption A1 and A2 and rule F (2),

S2: $T \mid \equiv \#\{\text{CPWDLR}\}_{\text{PWD,RT,RS}}$

From assumptions A1 and A2 and statement S1 and rule M (1), $T \mid \equiv R \mid \sim \{\text{CPWDLR}\}_{\text{PWD,RT,RS}}$

S3: $T \mid \equiv R \Rightarrow \{\text{CPWDLR}\}_{\text{PWD,RT,RS}}$

By statement S2, statement S3, and rule NV (1),

S4: $T \mid \equiv R \mid \equiv \{\text{CPWDLR}\}_{\text{PWD,RT,RS}}$

According to statement S3, statement S4, and rule J (1), $T \mid \equiv \{\text{CPWDLR}\}_{\text{PWD,RT,RS}}$.

Therefore, from statement S4 and statement S5, it can be stated as

S6: $T \mid \equiv R \mid \equiv (T \overset{\text{NKEY}}{\leftrightarrow} R)$

S7: According to the proposed protocol, $\langle NKEY \rangle_{IRC10, IRC5}$

Hence, from statement S1 to statement S7, it is evident that

$T \mid \equiv R \mid \equiv (\overset{\text{NKEY}}{\leftrightarrow} R) \longrightarrow$ (G2 i.e. goal 2 satisfied)

From Message 2 and rule I (2), it can be stated that

S8: $R \triangleleft \{\text{CPWDLT}\}_{\text{PWD,RT,RS}}$

From the assumptions A3 and A4, and from rule F (2), it can be stated that

S9: $R \mid \equiv \#\{\text{CPWDLT}\}_{\text{PWD,RT,RS}}$

By the assumptions A3 and A6 and from the statement S8 and rule M (2), it can be stated that

S10: $R \mid \equiv T \mid \sim \{\text{CPWDLT}\}_{\text{PWD,RT,RS}}$

$\text{R} \mid \equiv \text{T} \Rightarrow \{\text{CPWDLT}\}_{\text{PWD,RT,RS}}$

From statements S9 and S10 and from rule NV (2), the following statement can be obtained:

S11: $R \mid \equiv T \mid \equiv \{\text{CPWDLT}\}_{\text{PWD,RT,RS}}$

From statements S10 and S11 and from rule J (2), it can be stated that

S12: $R \mid \equiv \{\text{CPWDLT}\}_{\text{PWD,RT,RS}}$

Therefore, from statements S11 and S12, it is obvious that statement S13 can be formulated as

S13: $R \mid \equiv T \mid \equiv (T \overset{\text{NKEY}}{\leftrightarrow} R)$

S14: According to the RXMAP protocol, $\langle \text{NKEY} \rangle_{\text{IRC10,IRC5}}$

Hence, from statement S8 to statement S14, it is apparent that

$R \mid \equiv T \mid \equiv (T \overset{\text{NKEY}}{\leftrightarrow} R) \longrightarrow$ (G1 i.e. goal 1 satisfied)

Hence, from goals G1 and G2, formal analysis with BAN logic for the RXMAP protocol has been done.

4.2. Informal Analysis. In this section, the proposed RXMAP protocol is verified against variant possible security requirements, threats, and attacks.

4.2.1. Security against the Tag's Identity Reveal. In the message communication of the proposed protocol, tag's ID is not revealed in any mean. The pseudonym IDS combined with RT {IDS||RT} is only used for the transaction, and it is renewed for each process. So it is impossible to get the ID of the tag. Hence, RXMAP is secured against the tag's identity reveal.

4.2.2. Resistance to Known Session-Specific Temporary Information Attack. This attack happens when the adversary happened to get intermediate session specific information and tried to retrace the useful information of the protocol scheme. In our work, the communication messages {RS||CPWDLR}, CPWDLT, and Trigger are generated from the temporary random numbers. Moreover, messages are of different sizes that are 64 bits and 32 bits. This is not at all sufficient for tracing the tag or server information, so it is resistant to known session-specific temporary information attacks.

4.2.3. Mutual Authentication. Mutual authentication is essential for communication and data transfer to happen between tags and readers. A step by step verification of authentication of tags and server is presented in Section 3.2. It also narrates the termination of invalid tag and invalid servers at various stages. Our protocol offers mutual authentication among the valid devices in the network.

4.2.4. Forward and Backward Security. The communication messages {RS||CPWDLR}, CPWDLT, and Trigger are of sizes 64 bits, 32 bits, and 32 bits, respectively. The actual key size used in the encryption process is of 128-bit size. In addition with that, intermediate messages are generated through the session-specific random numbers. There is no possibility of finding trace of the previous or next session key. It is highly impossible to guess the 128-bit value from the 32-bit value.

4.2.5. Resistance to Replay Attack. This attack can happen when the intruder eavesdrops on any of the intermediate information to take the communication further in later point of time. This is a risky thing in many applications. In our protocol, each message at a specific time depends on the random numbers generated at that specific session. Since previous session's messages are not useful to perform replay, it is resilient to replay attack.

4.2.6. Resistance against Impersonation Attack. In order to impersonate the reader/tag adversary, we need to know the structure of the encoder in either side. Moreover, the intermediate messages are generated based on RS, RT, IDS, and key. None of the details needed for the computation and the domain-specific architecture of the encoder are available with the intruder to impersonate in between. Thus, the proposed protocol is tolerant against impersonation attack from either side.

4.2.7. Security against Man-In-The-Middle Attack. This attack may be performed by the privileged mediator trying to extract the tag or readers' identity so that the entire system becomes under the control of the intruder. This attack was there with the XTEA architecture. In our domain-specific customized RXMAP architecture, we introduced the dynamically changing "delta" value, "IDS" values, and "key" values based on the random numbers at each session. So, it is unpredictable in nature and information that the privileged mediator is subject to change for every session. Hence, the proposed architecture and protocol inherit the resistance against the man-in-the-middle attack.

4.2.8. Resistance to Masquerade Attack. An attacker may able to make the communication protocol to reveal the tag secret key by repeatedly sending the messages to the reader. This kind of repeated action will not be able to get the unique tag ID, because pseudonym was only used in each transaction also generated from the random numbers as follows:-

$\text{IDS} = \{\text{CPWDMR}/T[31:16] \| \text{RS}[31:16] \| \text{RT}[15:0] \| \text{CPWDMR}/T[15:0]\}$. Thus, no information can be cracked. The intruder cannot do replay attack, also due the IDS's freshness. Hence, the proposed scheme is resistant to masquerade attack and denial of service (DoS).

4.2.9. Resistance to Desynchronization Attack. Desynchronization attack focuses on disassociating the tag from the communication thereby leading to DoS. This is effectively handled in the proposed protocol by means of storing old and new IDs and key of the specific tag in the server and maintaining the freshness in the communication messages and values involved in by generating new random numbers RS and RT for each session. The proposed protocol uses verification message and acknowledges message to update the key and IDS values in either side. CPWDLR was used for verifying the server at the tag side, and CPWDLT was sent by the tag in response to act like an acknowledgement for the server and uses it for tag authentication. Key update will happen at the tag side upon receiving trigger acknowledge signal from the server.

Here, desynchronization attack may be applied at step 9 to disassociate the tag from the server. But in the proposed RXMAP, we stored old and new details of the tag in the

Table 4: Performance evaluation of the RXMAP against the security requirements, threats, and cryptanalysis models.

Protocol	SR1	SR2	SR3	SR4	SR5	SR6	SR7	SR8	SR9	SR10	SR11
Liu et al. [24]	Y	NA	Y	Y	Y	NA	NA	NA	Y	Y	NA
Assidi et al. [20]	Y	NA	Y	Y	Y	NA	NA	NA	Y	NA	NA
Fan et al. [23]	Y	NA	Y	Y	Y	NA	NA	NA	Y	Y	NA
Ayebie and Souidi [21]	NA	NA	Y	Y	Y	NA	NA	NA	Y	Y	NA
Izza et al. [22]	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	NA
Proposed RXMAP protocol	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

SR1: security against the tag's identity reveal; SR2: resistance to known session-specific temporary information attack; SR3: mutual authentication; SR4: forward and backward security; SR5: resistance to replay attack; SR6: resistance against impersonation attack; SR7: security against man-in-the-middle attack; SR8: resistance to masquerade attack; SR9: resistance to desynchronization attack; SR10: anonymity and untraceability; SR11: related key attack.

Table 5: Comparison of computation overhead of the tag with some lightweight protocols.

Parameter	Liu et al. [24]	Assidi et al. [20]	Fan et al. `[23]	Ayebie and Souidi [21]	Izza et al. [22]	Proposed RXMAP protocol
Types of computation	QC-MDPC encoder	AGS-based encoder	⊕, rot, per symmetric key encryption algorithm	Hash, rank metric coder	Hash, elliptic curve	XTEA approximate adders
Communication messages (in bytes)	1832	2740	192	1052	248	16

server. So once the attack happens, the tag will never get permanently disassociated from the server; rather, it will start the communication fresh with its old IDS and key. Therefore, the proposed scheme is versatile against the desynchronization attack.

4.2.10. Anonymity and Untraceability. In the proposed protocol, IDS is not separately sent. Rather, it is sent along with the random number RT as IDS||RT. It is difficult to differentiate the random number RT and the IDS since both are random and regenerated for every session. Hence, it is anonyms to the attacker if he wants to get the identity of the tag, and it is untraceable.

4.2.11. Related Key Attack. This is one of the weaknesses of the XTEA. The related key attack may be tried with the partial known part of the key in the transaction. The proposed RXMAP encoder architecture patched this weakness by randomizing the delta which involved in the calculation of sum and ciphered text. Another fact is that the key is rejuvenated in a highly random fashion since it is generated by intermediate results of cipher text processing and with unique delta value at each session.

$$\text{New key} = \{\text{IRC10}[31:16]||\text{IRC5}[15:0]||\text{IRC10}[15:0]||\text{IRC5}[31:16]\}. \quad (7)$$

Therefore, related key attack is overcome by the proposed architecture.

5. Evaluation of the RXMAP Protocol

The evaluation of the propounded RXMAP protocol is done on the basis of security requirements, computation overhead, and storage requirements to ensure its strength and lightweight to deploy it in the green energy devices.

5.1. Security Requirements. In this section, the evaluation of the RXMAP protocol for its resistance against security issues is presented. From the analysis at Section 4, it is inferred that the proposed protocol withstands various security threats and cryptanalysis models. Table 4 formed below provides the evaluation of our proposed RXMAP protocol against various security requirements, threats, and cryptanalysis models in comparison with the existing protocols.

From Table 4, it is clear that our proposed RXMAP outperforms other existing protocols with a simple architecture. Works listed for comparison in Table 4 are used complex algorithms to ensure the security, but the proposed work uses the approximate computing blocks to address the issues without adding any additional complex functions, so the architecture of the proposed RXMAP is simple.

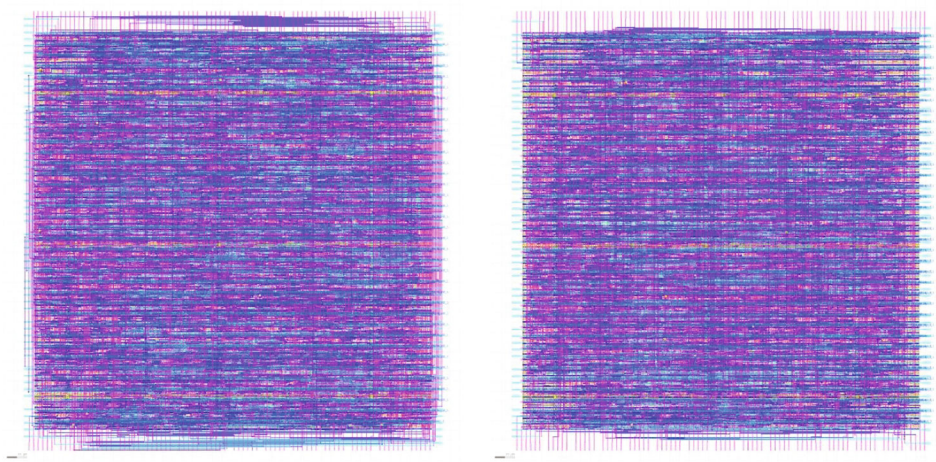
5.2. Computation Overhead. Computation over the head of the RXMAP in the tag side is given in Table 5, respectively.

From Table 5, it is certain that the computational complexity of the proposed protocol in the tag side is much less and a number of bytes transacted are also a few and it is 100 times lesser than other protocols.

5.3. Storage Requirement. The storage requirement of the tag is one of the pivotal deciding factors of the cost and size of the tag. Table 6 represents the comparison of storage

Table 6: Comparison of storage requirement of the tag with some lightweight protocols.

Parameter	Liu et al. [24]	Assidi et al. [20]	Fan et al. [23]	Ayebie and Souidi [21]	Izza et al. [22]	Proposed RXMAP protocol
Storage requirement (in bytes)	616	173	56	351	124	44



(a) (b)

Figure 3: The layout diagrams of the proposed RXMAP encoder architectures: (a) proposed RXMAP-1 encoder and (b) proposed RXMAP-2 encoder.

Table 7: Comparison of ASIC implemented proposed RXMAP architecture with the standard XTEA.

Algorithm	Block size	Logic process	GE	Cycles/block	Throughput @ 100 kHz	Throughput/area	Power (μW)
XTEA I [19]	64	130 nm	3500	240	26.7	0.008	18.8
XTEA II [19]	64	130 nm	3490	112	57.1	0.016	19.5
RXMAP-1	64	130 nm	1641	240	26.7	0.016	6.09
RXMAP-2	64	130 nm	1634	240	26.7	0.016	5.53

requirement of the proposed protocol with the recent lightweight protocols in the literature.

The proposed protocol needs 92%, 74%,21.43%, 87.5%, and 64.5% lesser storage compared to Liu, Assidi, Fan, Ayebie, and Izza et al.'s protocols, respectively.

6. ASIC Implementation of the RXMAP Encoder Architecture

The proposed RXMAP protocol is simulated with ModelSim 10.5b edition, and ASIC implementation is done in an OpenLane silicon-proven tool with the 130 nm logic process. The layout diagram of the proposed RXMAP-1 and RXMAP-2 is presented in Figure 3. ASIC implemented parameters are listed in Table 7.

Table 7 represents the through put and area in terms of gate equivalents (GE). In comparison, it is proved that the proposed designs RXMAP I and RXMAP II occupy 53.11% and 53.31% lesser area compared to XTEA I and 52.97% and 53.18% lesser area compared to XTEA II implementation.

From the parameter throughput at 100 kHz, it is clear that XTEA II overcomes other implementations, but the throughput per area of the fast implementation XTEA II and the proposed architectures RXMAP-1 and RXMAP-2 are equal while consuming 68.76% and 71.64% less power than XTEA II implementation, respectively. The proposed architectures show much improved power and throughput per area performance compared to XTEA I implementation. From this comparison, it is evident that the proposed RXMAP encoder architectures are the best fit for the portable low-energy applications.

7. Conclusion

The use of wireless sensor network applications in a day to day life is inevitable right from sophisticated systems to life-saving systems such as health care monitoring. Security and energy consumption problems are essentially addressed in this field. Our article suggested the solution for the power and security requirements of the RFID and wireless sensor network

applications. The proposed RXMAP encoder architecture patches the security issues in one of the most commonly used the lightweight XTEA algorithm and secured it from the related key attack and man-in-the-middle attack. The proposed domain-specific approximate adders OOA and XOA further made the modified XTEA architecture lighter in terms of computation and area. The proposed protocol is validated against various security threats through both formal and informal verifications. ASIC implementation results prove that the proposed designs RXMAP I and RXMAP II occupy 53.11% and 53.31% lesser area compared to XTEA I and 52.97% and 53.18% lesser area compared to XTEA II implementation. The total power consumed by the proposed encoder architectures RXMAP-1 and RXMAP-2 is 68.76% and 71.64% lesser than XTEA II implementation, respectively, while maintaining the equal throughput. These results ensure that the proposed RXMAP encoder architecture and protocol are the best fit for RFID and green wireless sensor network applications. The proposed domain-specific customization of the architectures could be extended further for other lightweight protocols to reduce the computational and storage complexity.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] D. Kwon, S. Yu, J. Lee, S. Son, and Y. Park, "WSN-SLAP: secure and lightweight mutual authentication protocol for wireless sensor networks," *Sensors*, vol. 21, no. 3, p. 936, 2021.

[2] A. Ibrahim and G. Dalkılıc, "Review of different classes of RFID authentication protocols," *Wireless Networks*, vol. 25, no. 3, pp. 961–974, 2019.

[3] H. Landaluce, L. Arjona, A. Perallos, F. Falcone, I. Angulo, and F. Muralter, "A review of IoT sensing applications and challenges using RFID and wireless sensor networks," *Sensors*, vol. 20, no. 9, p. 2495, 2020.

[4] Z. Mishra and B. Acharya, "High throughput novel architectures of TEA family for high speed IoT and RFID applications," *Journal of Information Security and Applications*, vol. 61, no. June, p. 102906, 2021.

[5] K. Mansoor, A. Ghani, S. A. Chaudhry, S. Shamshirband, S. A. K. Ghayyur, and A. Mosavi, "Securing IoT-based RFID systems: a robust authentication protocol using symmetric cryptography," *Sensors*, vol. 19, no. 21, p. 4752, 2019.

[6] S. Kerckhof, F. Durvaux, C. Hocquet, D. Bol, and F.-X. Standaert, "Towards Green Cryptography: A Comparison of Lightweight Ciphers from the Energy Viewpoint," in *International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 390–407, Springer, Berlin, Heidelberg, 2012.

[7] M. Scherer-Rath, J. Van Den Brand, C. Van Straten, L. Modderkolk, C. Terlouw, and E. Hoencamp, "Experience of contingency and congruent interpretation of life events in clinical psychiatric settings: a qualitative pilot study," *Journal of Empirical Theology*, vol. 25, no. 2, pp. 127–152, 2012.

[8] J. Borghoff, A. Canteaut, T. Güneysu et al., "PRINCE - a low-latency block cipher for pervasive computing applications," in *Advances in Cryptology – ASIACRYPT 2012*, X. Wang and K. Sako, Eds., vol. 7658 of Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2012.

[9] D. J. Wheeler and R. M. Needham, "TEA, a tiny encryption algorithm," *Lecture Notes in Computer Science*, vol. 21, pp. 363–366, 1995.

[10] R. M. Needham and D. J. Wheeler, "Tea extensions," *International Journal of Knowledge-Based and Intelligent Engineering Systems*, vol. 12, no. October 1996, pp. 3–6, 1997.

[11] D. J. Wheeler and R. M. Needham, *Correction to XTEA*, Cambridge University, England, 1988.

[12] D. Hong, J. K. Lee, D. C. Kim, D. Kwon, K. H. Ryu, and D. G. Lee, "LEA: A 128-bit block cipher for fast encryption on common processors," in *Information Security Applications. WISA 2013*, Y. Kim, H. Lee, and A. Perrig, Eds., vol. 8267 of Lecture Notes in Computer Science, pp. 3–27, Springer, Cham, 2014.

[13] D. Hong, J. Sung, S. Hong et al., "HIGHT: A New Block Cipher Suitable for Low-Resource Device," in *Cryptographic Hardware and Embedded Systems - CHES 2006. CHES 2006*, L. Goubin and M. Matsui, Eds., vol. 4249 of Lecture Notes in Computer Science, pp. 46–59, Springer, Berlin, Heidelberg, 2006.

[14] A. Moradi, A. Poschmann, S. Ling, C. Paar, and H. Wang, "Pushing the limits: a very compact and a threshold implementation of AES," in *Advances in Cryptology – EUROCRYPT 2011. EUROCRYPT 2011*, K. G. Paterson, Ed., vol. 6632 of Lecture Notes in Computer Science, pp. 69–88, Springer, Berlin, Heidelberg, 2011.

[15] K. Singh and D. D. Singh Tomar, "Architecture, enabling technologies, security and privacy, and applications of internet of things: a survey," in *Proc. Int. Conf. I-SMAC (IoT Soc. Mobile, Anal. Cloud), I-SMAC 2018*, vol. 4no. 5, pp. 642–646, New York, United States, 2019.

[16] M. Mozaffari-Kermani, K. Tian, R. Azarderakhsh, and S. Bayat-Sarmadi, "Fault-resilient lightweight cryptographic block ciphers for secure embedded systems," *IEEE Embedded Systems Letters*, vol. 6, no. 4, pp. 89–92, 2014.

[17] Q. Chai, "Securing low-cost RFID systems: a research survey," *International Journal of RFID Security and Cryptography*, vol. 3, no. 1, pp. 125–136, 2014.

[18] B. Pröll and H. Werthner, "Lecture notes in computer science," *Preface*, vol. 3590, 2005.

[19] J. Kaps, "Chai-Tea, cryptographic hardware implementations of xTEA," *Progress in Cryptology, INDOCRYPT*, vol. 5365, pp. 363–375, 2008.

[20] H. Assidi, E. B. Ayebie, and E. M. Souidi, *Based on Zero-Knowledge Proofs for RFID Systems*, Springer International Publishing, 2018.

[21] E. B. Ayebie and E. M. Souidi, "Rank-metric code-based mutual authentication protocol for RFID," *Journal of Information Security and Applications*, vol. 55, no. August, p. 102598, 2020.

[22] S. Izza, M. Benssalah, and K. Drouiche, "An enhanced scalable and secure RFID authentication protocol for WBAN within an IoT environment," *Journal of Information Security and Applications*, vol. 58, no. February, p. 102705, 2021.

[23] K. Fan, Q. Luo, K. Zhang, and Y. Yang, "Cloud-based lightweight secure RFID mutual authentication protocol in IoT," *Information Sciences*, vol. 527, pp. 329–340, 2020.

[24] Z. Liu, W. Zhang, and C. Wu, "A lightweight code-based authentication protocol for RFID systems," *Communications in Computer and Information Science*, vol. 557, pp. 114–128, 2015.

[25] J. L. Hennessy and D. A. Patterson, "A new golden age for computer architecture," *Communications of the ACM*, vol. 62, no. 2, pp. 48–60, 2019.

[26] Y. Ko, S. Hong, W. Lee, S. Lee, and J. S. Kang, "Related key differential attacks on 27 rounds of XTEA and full-round GOST," in *Fast Software Encryption. FSE 2004*, B. Roy and W. Meier, Eds., vol. 3017 of Lecture Notes in Computer Science, pp. 299–316, Springer, Berlin, Heidelberg, 2004.

[27] T. Isobe and K. Shibutani, "All subkeys recovery attack on block ciphers: extending meet-in-the-middle approach," *Selected Areas in Cryptography, SAC*, vol. 7707, pp. 202–221, 2013.

[28] J. Wang, T. Ye, and E. C. Wong, "Privacy guaranteed mutual authentication on EPCglobal class 1 Gen 2 scheme," in *2008 The 9th International Conference for Young Computer Scientists*, pp. 1583–1588, Hunan, China, 2008.

[29] C. Kella, Z. Mishra, and B. Acharya, "A compact low power architecture of XXTEA192 lightweight block cipher," in *2021 6th International Conference on Communication and Electronics Systems (ICCES)*, pp. 972–976, Coimbatre, India, 2021.

[30] A. A. M. Ragab, A. Madani, A. M. Wahdan, and G. M. I. Selim, "Design, analysis, and implementation of a new lightweight block cipher for protecting IoT smart devices," *Journal of Ambient Intelligence and Humanized Computing*, vol. 15, 2021.

[31] H. Yeo, S. Sonh, and M. Kang, "IP design of corrected block TEA cipher with variable-length message for smart IoT," *KSII Transactions on Internet and Information Systems*, vol. 14, no. 2, pp. 724–737, 2020.

[32] R. Anusha and V. Veena Devi Shastrimath, *LCBC-XTEA: High Throughput Lightweight Cryptographic Block Cipher Model for Low-Cost RFID Systems*, vol. 986, Springer International Publishing, 2019.

[33] G. N. Khan and G. Zhu, "Secure RFID authentication protocol with key updating technique," in *2013 22nd International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–5, Nassau, Bahamas, 2013.

[34] H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie, and C. Lucas, "Bio-inspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 4, pp. 850–862, 2010.

[35] G. N. Khan, X. Yu, and F. Yuan, "A novel XTEA based authentication protocol for RFID systems," in *2011 XXXth URSI General Assembly and Scientific Symposium*, vol. 2, pp. 26–29, Istanbul, Turkey, 2011.

[36] A. Dalloo, A. Najafi, and A. Garcia-Ortiz, "Systematic design of an approximate adder: the optimized lower part constant-OR Adder," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 8, pp. 1595–1599, 2018.

[37] A. Sinha Roy, R. Biswas, and A. S. Dhar, "On fast and exact computation of error metrics in approximate LSB adders," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 4, pp. 876–889, 2020.

[38] C. K. Jha, S. N. Ved, I. Anand, and J. Mekie, "Energy and error analysis framework for approximate computing in mobile applications," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 2, pp. 385–389, 2020.

[39] R. Porto, L. Agostini, B. Zatt, N. Roma, and M. Porto, "Power-efficient approximate SAD architecture with LOA imprecise adders," in *2019 IEEE 10th Latin American Symposium on Circuits & Systems (LASCAS)*, pp. 65–68, Armenia, Colombia, 2019.

[40] V. R. Vijaykumar, S. R. Sekar, S. Elango, and S. Ramakrishnan, "Implementation of $2^n - 2^k - 1$ modulo adder based RFID mutual authentication protocol," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 1, pp. 626–635, 2018.

[41] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Transactions on Computer Systems*, vol. 8, no. 1, pp. 18–36, 1990.

[42] M. Burrows, M. Abadi, and R. Needham, *Authentication: A Practical Study in Belief and Action*, University of Cambridge, Computer Laboratory, 1988.

[43] W. Mao and C. Boyd, "Towards formal analysis of security protocols," in *[1993] Proceedings Computer Security Foundations Workshop VI*, pp. 147–158, Franconia, NH, USA, 1993.