Hindawi
Wireless Communications and Mobile Computing
Volume 2024, Article ID 5998163, 14 pages
https://doi.org/10.1155/2024/5998163

Research Article

An Elliptic Curve Menezes–Qu–Vanston-Based Authentication and Encryption Protocol for IoT

Susovan Chanda,[1] **Ashish Kr. Luhach**,[2] **J. Sharmila Anand Francis**,[3] **Indranil Sengupta**,[4] **and Diptendu Sinha Roy**[1]

[1]*Department of Computer Science and Engineering, National Institute of Technology Meghalaya, Shillong, Meghalaya, India*
[2]*Department of Electrical and Communication Engineering, The Papua New Guinea University of Technology, Lae, Papua New Guinea*
[3]*Department of Computer Science, Rejal Alma'a Campus, King Khalid University, Abha, Saudi Arabia*
[4]*Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, Kharagpur, India*

Correspondence should be addressed to Ashish Kr. Luhach; ashish.kumar@pnguot.ac.pg

Received 22 February 2023; Revised 7 February 2024; Accepted 8 February 2024; Published 22 March 2024

Academic Editor: Manuel Fernandez-Veiga

The exponential growth of the Internet of Things (IoT) has led to a surge in data generation, critical for business decisions. Ensuring data authenticity and integrity over unsecured channels is vital, especially due to potential catastrophic consequences of tampered data. However, IoT's resource constraints and heterogeneous ecosystem present unique security challenges. Traditional public key infrastructure offers strong security but is resource intensive, while existing cloud-based solutions lack comprehensive security and rise to latency and unwanted wastage of energy. In this paper, we propose a universal authentication scheme using edge computing, incorporating fully hashed Elliptic Curve Menezes–Qu–Vanstone (ECMQV) and PUF. This approach provides a scalable and reliable solution. It also provides security against active attacks, addressing man-in-the-middle and impersonation threats. Experimental validation on a Zybo board confirms its effectiveness, offering a robust security solution for the IoT landscape.

1. Introduction

The Internet of Things (IoT) has assimilated into our daily life. IoT solutions are pervasive in many facets of daily life, including remote sensing, agriculture, healthcare systems, smart cities, smart homes, etc. [1–3]. The number of IoT devices are expected to exceed 75 billion by the year 2025 [3]. Such massive number of IoT devices produces a large amount of data which requires massive computational power to analyze and process these data. Since IoT devices have limited resources, they cannot store and compute such huge volumes of data generated by the IoT devices. Cloud computing is employed to meet such demands. Depending upon the applicability, IoT devices collect and transmit the raw data to the cloud server for subsequent analysis and processing. Massive amounts of raw data are transferred straight to cloud servers, which substantially degrades network performance and frequently creates a single point of failure and causes high latency. To avoid these problems, a three-tier IoT-edge-cloud architecture is commonly used in recent times. In this three-tier architecture, IoT nodes offload the heavy computation tasks to a nearby edge nodes. IoT devices transfer the raw data to edge nodes which then forward the processed data to cloud servers after analyzing them. Thus, it improves the overall performance of the network. Edge nodes typically work in a public network and are connected to IoT nodes using wireless networks. This makes the entire ecosystem vulnerable to various cyber attacks, namely, man-in-the-middle attack, repudiation attack, and eavesdropping [1]. Since processing of these data plays a direct role in some very critical day-to-day operations, tampering of the data may cause catastrophic failures. So, it is essential to ensure that tamper-free data are passed through unsecured channels [4]. Inclusion of cloud computing and edge computing with IoT devices make the ecosystem heterogeneous in nature and a generalized security solution is essential to meet the security requirements [5]. In contrast to the expansive capabilities of cloud, IoT devices are characterized

by constrained resources. Due to these contradicting characteristics of IoT ecosystems, conventional security solutions do not present a wholesome solution for the entire ecosystem [6]. Public-key infrastructure is one of the widely used conventional security systems that can address overall network security. Similarly, identity-based public key cryptography (ID-PKC) is another security solution which is used by traditional desktop-based computing systems [7]. The IoT ecosystem, when considered holistically, cannot achieve the requisite level of security through the utilization of current communication protocols tailored for IoT devices. They are susceptible to various types of attacks, such as credential disclosure attacks, unprotected pairing attacks, etc. [8–10].

In the pursuit of creating comprehensive security protocols for IoT ecosystems, researchers have faced the challenge of developing a one-size-fits-all solution [11, 12]. While several security protocols have been designed for IoT systems, a significant proportion of them rely on two fundamental techniques: Physical Unclonable Function (PUF) and elliptic curve cryptography (ECC) [13, 14]. These protocols have gained popularity because ECC, unlike traditional public-key cryptography algorithms, offers heightened security while maintaining relatively small key sizes. Notable example of such ECC-based solution can be found in Chanda et al.'s [15] study. However, these protocols have their own limitations. One of the serious problems that these solutions suffer from are the amount of energy they consume. These protocols exhibit more energy which impact the longevity of the device [16]. Another problem that these methods are unable to resolve is the man-in-the-middle attack since they use Deffie Hellman elliptic curve to generate the key pair. Man-in-the-middle attacks can be eliminated by using ECMQV, a variation of ECC [17]. ECMQV also offers defence against impersonator attacks [18]. Another major of the existing problem is that they are using cloud server to assist the IoT nodes generate the public and private key pair. Main drawback of this approach is increased latency, as data must travel to and from remote cloud servers. Bandwidth usage is another area of concern, particularly with large number of IoT devices participating in the ecosystem which produces large-scale data transfers between devices and the cloud. Another problem with this approach is compliance issue. Compliance with data regulations may be challenging.

Protocols mentioned in Chanda et al.'s [15, 19–23] study have used ECC and PUF. PUF uses the physical property of the integrated circuit (IC) to recognize the IC. Chatterjee et al. [19] suggested a method for generating session keys that makes use of an extra IC to produce the PUF. And PUF generated by the IC is used for authentication purposes only. Moreover, the registration process in these solutions are very tedious and has to be done manually by capturing challenge–response pair (CRP) for all participating devices, which becomes prohibitively inconvenient with massive number of IoT devices. The complicated bilinear pairing of this method is another drawback. To create key pairs for IoT nodes, a sophisticated bilinear pairing technique is applied. Additionally, it is unable to prevent impersonation attacks and man-in-the-middle attack [21]. Recent efforts have been directed toward developing a secure protocol for IoT networks using Blockchain and PUF circuit [24].

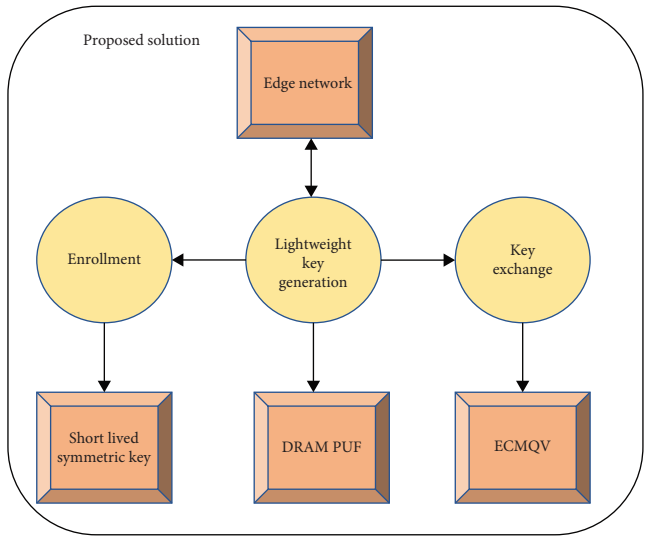

Figure 1: Schematic diagram of the proposed method.

Here we presented a novel authentication-cum-key generation protocol for seamless communication across an IoT network. Our primary contributions include:

(1) We propose a secure authentication protocol for IoT edge computing using PUF. Existing works in the domain either focus exclusively on the IoT network or devote their investigation on cloud networks only, thus leaving a void in finding generalized solutions that address the heterogeneity of IoT ecosystems. Usage of edge computing offers distinct advantages over the other solutions especially over cloud computing. By processing data closer to its source, proposed solution reduces latency, ensuring faster response times. This approach also optimizes bandwidth usage, as only relevant data or processed results are transmitted, enhancing efficiency. Additionally, the resilience of edge computing shines in scenarios with intermittent or unreliable cloud connectivity. Cost savings are achievable through reduced data transfers and optimized resource usage. Scalability is facilitated by the ability to horizontally scale resources by adding more edge devices. For industries with stringent data regulations, it provides compliance by keeping data closer to its origin.

(2) We also propose a novel Key generation scheme based on a simple MQV elliptic curve which consumes less energy.

(3) We have used underlying circuit of devices to define trust relationships using special hardware property called PUF (Figure 1).

(4) The proposed solution provides security from man-in-the-middle attack. We also provide a formal security proof of the protocol.

Table 1: Key sizes (in bits) as per NIST recommendation.

Symmetric key	RSA-key bits	ECC bits
80	1,024	160
112	2,048	224
128	3,072	256
192	7,680	384
256	15,360	521

(5) We provide performance analyses of the protocol in terms of communication overhead, storage, and computation requirements.

The rest of the paper is organized as follows: Challenges related to IoT key generation have been presented in Section 2. We also discussed possible ways of solving such problems therein. The proposed protocol has been discussed in Section 3. Experimental evaluation has been presented in Section 4. Section 5 discusses the security analyses. Finally, Section 6 presents the concluding remarks and potential future works.

2. Background

The section has explored the realm of ECC and its variant, Menezes–Qu–Vanstone (ECMQV), as well as delved into the concept of PUF. Following this discussion, the focus has shifted to research efforts related to lightweight public key infrastructure (PKI) solutions.

2.1. Elliptic Curve Cryptography and Menezes–Qu–Vanstone. Discrete logarithm problem is an unique feature of an elliptic curve which is the basis of ECC. Miller [25] described how asymmetric keys can be built from it. Any point in the elliptic curve is treated as a public key. And when the given point is multiplied with a random number it produces correspondence private key.

Elliptic curve in cryptography (ECC) has its advantage in key size. It provides better security protection than other asymmetric key cryptography using small key size. Table 1 presents the key size recommendation fixed by NIST for different cryptography Algorithm 1.

The IoT protocols mentioned in Chatterjee et al.'s [19, 26–30] study have used elliptic curve-based cryptography scheme. Menezes–Qu–Vanstone (MQV) is an authenticated protocol based on the Diffie–Hellman scheme. Like other DH schemes, MQV provides protection against an active attacker. Modified form of MQV is known as elliptic curve MQV (ECMQV). ECMQV provides more protection against active attacker and provides less computation overhead compared to ECC [16, 18]. We have used the hash variant of ECMQV in our proposed method.

2.2. Physical Unclonable Function (PUF). PUF employs the complex physical device features to carry out a set of challenge–response pairs. Such pairs are specific to the device. It can not be represented using any mathematical model. Due to this, it is very difficult for adversaries to guess or produce the secret by themselves. Moreover, no storage is required to keep the secret [31]. Since PUF secret rely on the physical features of an integrated circuits, it is not possible to produce them [32].

2.3. Related Works. In the context of authentication protocols for IoT devices within a three-tier IoT-edge-cloud architecture, they can be broadly categorized into two groups as discussed in Aziz et al.'s [33] study. The first category employs symmetrical keys, while the second utilizes PKI-based ECC protocols. A symmetric key intercloud authentication protocol has been introduced in Seifelnasr et al.'s [34] study. Although symmetric keys demand less computational power, a significant drawback lies in their secure distribution. Securely transmitting symmetric keys over a public channel is a challenging task. Yang et al. [35] proposed a decentralized edge-based authentication protocol in which authentication processes occur at the edge nodes. Additionally, an ECC-based anonymous mutual authentication protocol is put forth in Li et al.'s [36] study. Challenges associated with these protocols include device identification and random number generation, which are crucial in defining elliptic curves. To address these issues, the utilization of PUF technology provides a solution.

Main applications of PUF are low-cost authentication and key generation [19–21, 31, 37]. PUF can be used as a seed to a key generation algorithm and generate the key pair successfully [31, 37]. It can also be used as a symmetric key to secure the communication between two nodes [31]. Identity-based public key cryptography scheme based on PUF has been discussed by Chatterjee et al. [19] and Yang et al. [27]. This scheme has resolved the key distribution problem by using ID-based public key cryptography. But these methods have many shortcomings. Two major such problems are as follows: bilinear function has been used to realize these scheme and bilinear function is a complex operation which consumes significant amount of computing resources. Moreover, IBE cryptography inherently suffer from key escrow issue. Certificateless cryptography can be used for generating the key pair for small devices and it has been discussed by Ma et al. [28] and Seo et al. [38]. However, main issues with these scheme is the complexity of the process where end nodes need to do multiple handshaking before establishing a secure session. Chatterjee et al. [19] proposed a key generation and key exchange protocol for IoT devices. As discussed in the previous section, this scheme has quite a few number of limitations such as additional area overhead, manual enrollment process, and complex key generation process. Moreover, it is suffering from man-in-the-middle attack and replay attack. This work has been further improved by Braeken [21] and Chatterjee et al. [20]. A notable work mentioned in Braeken's [21] study tried to avoid man-in-the-middle attack, whereas Chatterjee et al. [20] proposed for the removal of the CRP database from key generation node to a offline secured database.

Boneh and Franklin [39] proposed ID-based public key cryptography. This method gets rid of tedious certificate distribution process by using the user's identity such as email ID, name, etc., to generate public key. When one node wants to communicate with another nodes, sender uses identity of

Table 2: Comparison of state-of-the-art IoT security solutions.

State of art solution	Traditional PKI	Certificateless PKI	ID-based PKI
Existing cloud-based solutions transferred huge amount of data directly to cloud that brings down the network performance severely and pose a single point of failure	Digital certificate requires computation heavy process to build and validate	It avoids certain complex PKI operations. But it still uses partial key generation, revocation process	Key escrow problem
These solutions added extra circuit to produce the PUF	Computation intensive process	Computation intensive process	Computation intensive process
It requires more time to generate PUF mapping	Memory consumption is high	Memory consumption is high	Memory consumption is high

the recipient node and get the public key from a trusted third party called key generation center (KGC) to encrypt the message. Upon receiving the encrypted message, recipient gets the corresponding private key from KGC after successfully proving its valid identity and decrypt the message. Though this scheme solve the key distribution issue but it suffers some major issues. One of them is key escrow issue.

Public key cryptography mentioned above has resolved a critical issue, i.e., complex certificate exchange and validation of digital certificate. These two operations are resource intensive. That is why these solutions may be used in resource constrained environment such as IoT ecosystem.

An ID-based public key cryptography protocol has been proposed by Chatterjee et al. [19] and Yang et al. [27]. In this scheme, key distribution issue has been resolved by introducing the identity-based public key cryptography. However, it has couple of significant issues. At first, a complex bilinear function is used to implement the protocol. Bilinear pairing is computation heavy and complicated process. Second, IBE cryptography inherently suffers from key escrow issue [28]. Seo et al. [38] explained how certificateless cryptography can be used for generating the key pair for small devices. However, main issue with these scheme is the complexity of the process where end nodes need to do multiple handshaking before establishing a secure session.

Identity-based encryption (IBE) proposal have been implemented by Chen [30] and Boneh and Franklin [40]. Original IBE had two major issues—it was prone to repudiation attack and it was suffering from key escrow problem. Chatterjee et al. [19] has overcome these two issues. In addition to this it has replaced public known identity such as email address with the PUF response as public identity of the IoT node. Major disadvantage of this scheme is that it uses bilinear pairing to achieve the above requirement. Bilinear pairing is computation heavy and takes a significant amount of CPU cycle to compute. This can be clearly visible in the result shared in Chatterjee et al.'s [19] study. And it also proposed these computation heavy task to be performed at the resource constrained IoT node. Combination of traditional PKI and IBE protocol has been proposed for IoT environment by Yang et al. [27].

PUF is another research area which is being considered for fulfilling the security needs of the IoT device. PUF uses the manufacturing variations of ICs to derive unique information [31] from a device. PUF has many interesting features. It can uniquely identify a billions of ICs even though they are manufactured by the same vendor using same design specification. Another noteworthy feature is it can be used as private–public key generation [41]. Asymmetric key generation process using PUF has been explained in Marchand et al. [22] and Park et al.'s [42] studies. It has generated seed by hashing the PUF output and use the seed to generate the public/private key using key generation protocol. Each IC can have exponential number of unique set of CRP and it is next to impossible to model the CRP generation process using any mathematical model [37]. Due to the noise, output of the PUF response is not accurate every time it is generated for a given challenge.

PUF circuitry can be categorized into two types—intrinsic- and FPGA-based PUF [43]. FPGA-based PUF such as arbiter PUF needed additional circuitry to implement. However, in case of intrinsic PUFs, challenge-response pair (CRP) can be found within the device hardware itself. Intrinsic PUFs can be based on static random access memory (SRAM) and dynamic random access memory. Disadvantage of PUF based on SRAM is that CRP must be extracted during boot stage. However, CRP can be retrieved during run time for PUF implemented using DRAM. Until recently, IoT security solution proposed using PUF [19] requires additional FPGA hardware circuit. Proposed protocol has used DRAM and it is inbuilt to the device. That is why in this protocol, we do not need additional hardware. Moreover, it can retrieve CRP information at run time. Table 2 describes the comparative study among the existing security solutions for resource constrained device.

Protocol mentioned in this paper has removed the challenges mentioned above. At the same time, it takes less computation time and dissipates less amount of power. These two features are suitable for IoT ecosystems.

3. Proposed Method

This paper enhances the solution proposed by Chanda et al. [15]. It introduces a lightweight PKI protocol using an edge network. And it replaces the traditional elliptic curve with more secured MQV elliptic curve. Usage of edge computing offers distinct advantages over the other solutions especially over cloud computing. By processing data closer to its source,

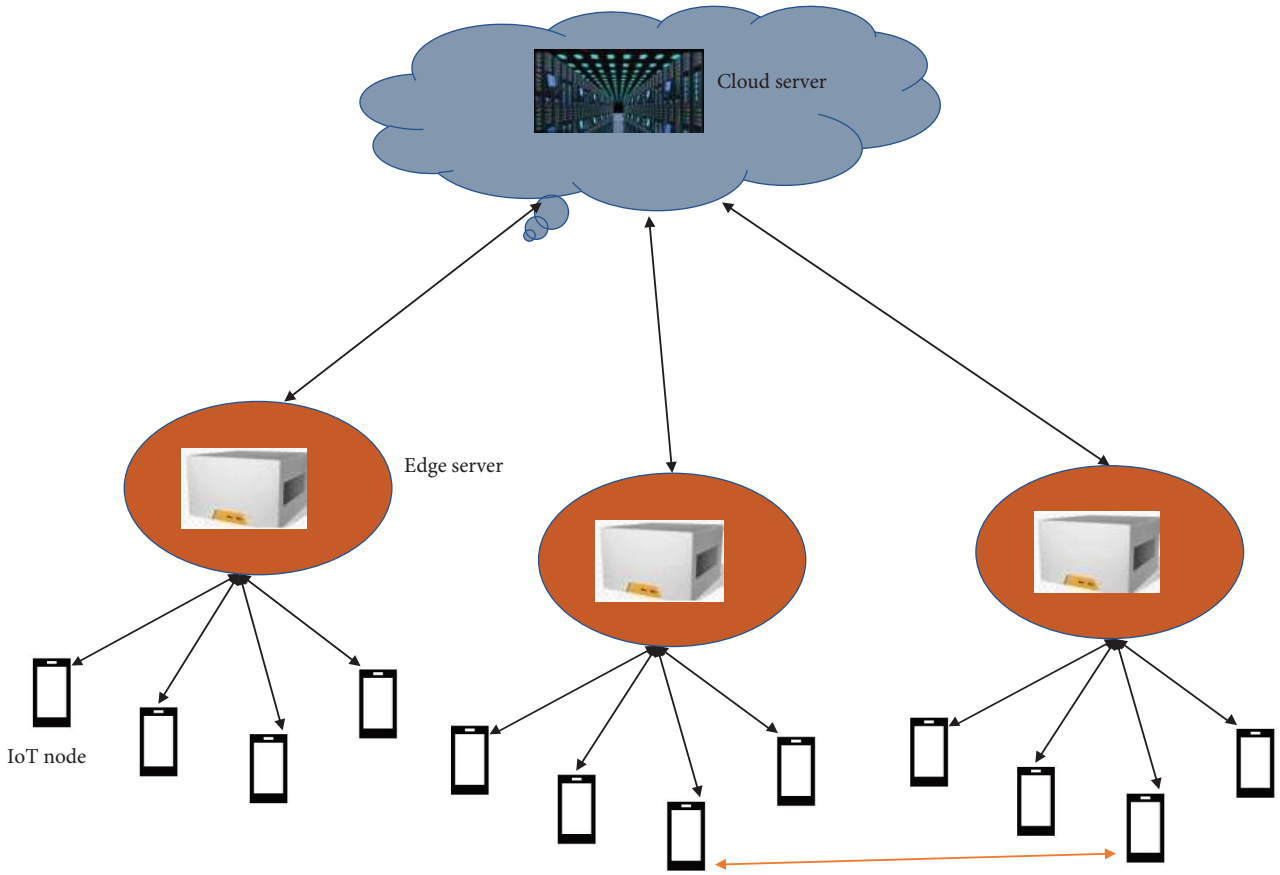

Figure 2: Overall architecture of the proposed method.

proposed solution reduces latency, ensuring faster response times. This approach also optimizes bandwidth usage, as only relevant data or processed results are transmitted, enhancing efficiency. Additionally, the resilience of edge computing shines in scenarios with intermittent or unreliable cloud connectivity. Cost savings are achievable through reduced data transfers and optimized resource usage. Scalability is facilitated by the ability to horizontally scale resources by adding more edge devices [41]. For industries with stringent data regulations, it provides compliance by keeping data closer to its origin. MQV on elliptic curves represents a sophisticated enhancement to traditional ECC for secure key agreement. The key idea behind MQV on elliptic curves is to establish a shared secret between two parties over an insecure communication channel, utilizing the mathematical properties of elliptic curves. Like Chanda et al. [15] proposed solution is also built on fundamental principles of the well-established PKI system, where nodes exchange their public keys before data transfer. In contrast to traditional PKI systems, the proposed protocol simplifies the digital certificate structure and eliminates resource-intensive operations like certificate validation and certificate revocation processes. Despite these simplifications, it manages to offer security features comparable to PKI. The following sections provide a detailed explanation of the protocol's process.

3.1. Trust Model. In the traditional PKI system, trust relationships are established between end nodes and a Certificate Authority (CA). End nodes rely on the CA to validate certificates of other nodes through a complex certificate validation process. In the proposed scheme, a two-layer trust model is employed. The first layer establishes trust between an IoT node and a designated special node known as the Key Center (KC), which is hosted within the edge network. The second layer of trust is established between the KC and a Cloud Server. Instead of storing key information within digital certificates, this scheme utilizes PUF. PUF generates unique CRPs, which are employed for authenticating devices. A node is authenticated when it correctly provides the CRP requested by the KC. Additionally, CRPs generated by PUF serve as symmetric keys for distributing public keys among user nodes. This innovative approach enhances the security of the system.

The architectural framework of the proposed method is depicted in Figure 2, presenting a structure comprising three primary components involved in the communication process. These components are defined as follows:

(1) Storage Server (SS): Hosted in the cloud, the SS maintains the CRPs for all nodes enrolled in the IoT ecosystem.

(2) KC: Situated within the edge network, the KC serves as an intermediary node responsible for coordinating between end nodes and facilitating the production and distribution of public keys. It plays a crucial role in mediating communications between user

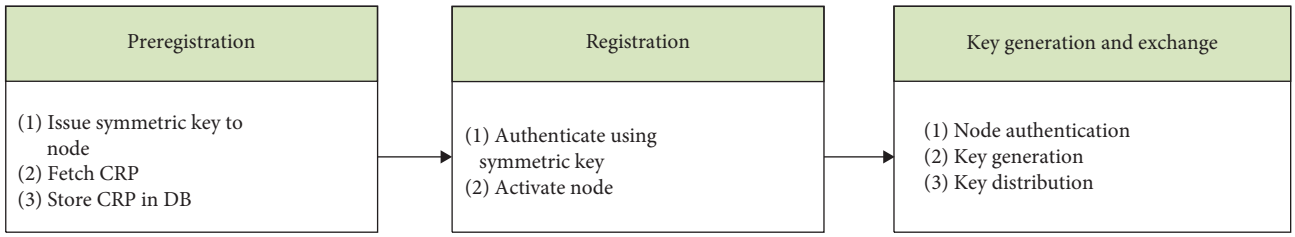

Figure 3: Different phases of the proposed solution.

nodes and the SS. Multiple KC nodes can be deployed as needed to accommodate the number of user nodes in the ecosystem, ensuring adequate support for user nodes' operations.

(3) IoT nodes: These are the end-user nodes within the IoT ecosystem. IoT nodes establish trust relationships and communicate with the KC for authentication and other interactions.

The interaction within this framework involves a structured communication flow, where the SS maintains a direct connection with the KC. User nodes, on the other hand, do not directly communicate with the SS. During the registration phase, the SS issues short-lived symmetric keys to each user node for enrollment, enhancing security. The proposed scheme introduces two distinct trust relationships within this architecture, which will be further elaborated upon.

(1) Permanent relationship—It is defined between SS and KC. PKI system has been used to establish this relationship.

(2) Temporary relationship—This is established during registration phase where SS issues symmetric key to each user node for enrollment.

Protocol mentioned here comprised of three steps as shown in Figure 3. Each of these steps has been discussed in the following subsection.

3.2. Pre-Enrollment Phase. SS generates the symmetric using the algorithm based on the reciprocity of wireless channels, as mentioned in Haroun and Gulliver's [44] study. Then, it distributes the symmetric key to each user node. Symmetric key is valid for a given duration t and user node needs to finish the enrollment by this time.

3.3. Enrollment Phase. Enrollment phase started after the pre-enrollment phase. In this phase, KC sends a set of challenges to the user node. Upon receiving the challenges from KC, user node produces the response from in-built PUF circuit. It involves encrypting PUF data with the symmetric key received from the KC. The resultant encrypted response is then transmitted to the KC. Subsequently, the KC forwards the received request from the user node to the SS. Within this process, the SS undertakes the decryption of the data utilizing the symmetric key provided to the user node during the pre-enrollment phase. Following the decryption of the message transmitted by the user node, the SS retrieves the response stored in the repository. This retrieved response is then validated against the responses received from the user node, thus completing the symmetric key encryption and decryption cycle with a focus on secure data transmission:

(1) $\texttt{Response}_{\text{User}} = D(E(\text{Response}_{\text{User}}), \text{SymmetricKey})$.

(2) $\texttt{Validation} = \texttt{Validate}(\texttt{Response}_{\text{SS}}, \texttt{Response}_{\text{User}})$.

3.4. Key Generation and Exchange. Process of the authentication-cum-communication is initiated by two user nodes who want to communicate between them. KC will assist the user nodes to generate the key pair and distribute between them. It has three subprocesses.

3.4.1. Authentication. KC authenticates the user nodes before it assists them in key generation and exchange process. It reads the CRP from SS and then sends the challenge to the user nodes and ask them to provide the corresponding response. After receiving the response from the user nodes, it validates them with the response it retrieves from the RS node. If it finds the response as valid then KC initiate the key generation process.

(1) Two nodes can exchange data between them only when they have the public key of each other. Otherwise, they would not be able to encrypt the data needed for secured communication. It consists of two steps. Initially, it generates public and private key. And in the second step, it exchanges the public among each other through KC. Since KC supervise the key generation and exchange process it first checks if the given nodes have already generated the key pair as part of a different communication. In case, KC identifies any such key pair then it skips the key generation process and initiate the key distribution.

(2) Node $\texttt{Node}_A$ that wants to initiate the communication sends request to KC to provide them the public key of $\texttt{Node}_B$. Also it sends a request to KC for helping it in key generation and exchange process. $\texttt{R}, \texttt{Node}_B, \texttt{Node}_A, E_{\text{PUF}(C_1)}(\texttt{Node}_A)$, and C_1. It sends the operation name as part of the request (R here) and other details it sends are identification of $\texttt{Node}_B$, identification of $\texttt{Node}_A$. It also sends the encrypted form of identification of $\texttt{Node}_A$ using PUF response and the equivalent challenge. CRP of

C_1 is used to validate if the request has been sent by the correct node. This is as follows.

(i) KC retrieves challenge–response from SS for C_1 of $Node_A$ to validate the authenticity of the node.

(ii) Following equation is used to validate the authenticity $Node_A = Node_A \oplus PUF(C_1)$ $PUF_{RS}(C_1)$

$$Node_A = D_{PUF_{RS}}(E_{PUF(C_1)}(Node_A)).$$

(iii) After $Node_A$ is successfully authenticated, it initiates validation process for $Node_B$ in the next steps.

(iv) KC retrieves set of CRP for $Node_B$ from SS. And then sends a request to $Node_B$ to provide the corresponding response. Request KC that is sent to $Node_B$ is given as follows:

$AUTH, Node_B, E_{PUF_{RS}}(C_2)$, and C_2.

(v) KC then validates the authenticity of $Node_B$ by the equation given as follows:

$$C_2 = D_{PUF_{C_2}}(E_{PUF_{RS}(C_2)}).$$

(vi) A nonce is generated and sent to KC after the successful validation of $Node_B$:

$AUTH, NONCE_B, E_{PUF_{C_2}}(NONCE_B), C_2$, and $Node_B$.

(vii) KC validates $Node_B$'s authentication using the following formula:

$$NONCE_B = D_{PUF_{C_2}}(E_{PUF_{RS}}(NONCE_B)).$$

(viii) After both the nodes are authenticated successfully, KC begins the key pair generation and exchange process.

3.4.2. Key Generation and Distribution

(1) KC retrieves the CRP of the given node on random basis. Then, it performs the necessary error corrections on the response using a hash function. After error correction is done on the response bit, it generates the generic parameters for the elliptic curve using the following equation:

$H: 0, 1^n$;

$G \rightarrow H(C_1 \oplus C_2)$.

After generating the generic parameters for elliptic curve, KC sends them to be used for key generation to both nodes. Then, each node uses ECC and the generic parameters provided by the KC to generate the key pair by themselves.

(2) User nodes use the following equation to create key pair based on the generic parameters provided by the KC:

$s \rightarrow$ random number;

$P_{pub} \rightarrow s \times G$; public key;

$s \rightarrow$ private key.

(3) Once the key pairs are generated by the individual node then they would send the public key to KC so that it can distribute it to the node who is interested.

$Msg1 = E_{PUF_{C_1}}(Node_B \oplus Pub_A)$;

$SHARE, Node_B, Pub_A, Msg1, C_1, Node_A$.

(4) KC uses the following equation to validates the public key it receives from the user nodes:

$Pub_A = D_{PUF_{RS}}(Msg1)$ $Pub_A = D_{PUF_{RS}}(E_{PUF_{C_1}}(Node_B \oplus Pub_A))$.

(5) After validating the public key, KC distribute the public key to the intended node using the following equation:

$Msg2 = E_{PUF_{C_2}}(Node_B \oplus Pub_A)$;

$SHARE, Node_B, Pub_A, Msg2, C_2, Node_A$.

(6) Individual node also validates the public sent by KC. $Node_B$ validates if it has received correct key from $Node_A$ using given equation.

(i) A nonce messages is created by $Node_B$ to encrypt the public key sent by $Node_A$.

$Data = ENC_{Pub_A}(NONCE)$

$KEYVER, Node_A, NONCE, Data, Node_B$.

(ii) $Node_A$ validates the request after getting the key verification request from $Node_B$.

$NONCE' = DEC_{Pvt_A}(Data) = NONCE$.

(iii) One more nonce is generated by $Node_A$ to encrypt the public key of $Node_B$ and transfer back the request to $Node_A$.

$Data = NONCE + NONCE_1$;

$Data = ENC_{Pub_A}(Data)$;

$KEYVER, Node_B, NONCE_1, Data, Node_A$.

(iv) Next, $Node_B$ verifies the request using the mentioned equations:

$Data_1 = NONCE + NONCE_1$;

$Data_1 = DEC_{Pvt_B}(Data_1) = Data$.

(v) It completes the key generation and exchange steps.

The above mentioned functionalities are shown in Algorithm 1.

4. Experimental Evaluation

In the absence of standardized tools for measuring the security of communication protocols, a custom experimental setup has been meticulously designed to validate the proposed method. This setup comprises essential hardware and software components. A personal computer, boasting a Core i5 processor and 8 GB of RAM, is designated as the central control unit (RS) for managing and overseeing the experiment. Two Zybo Zynq-7000 boards serve as user nodes, chosen for their unique fusion of ARM-based software programmability and FPGA-based hardware programmability. The core of the experiment involves the implementation of a DRAM-based

Input: $H1:\{0,1\}^n \in \{0,1\}^n$,
$G \leftarrow H1(C_a \oplus C_b)$,
$E \leftarrow$ MQV Elliptic curve E over finite field F_q.
Output: $\text{Pub}_A \rightarrow$ Public Key of A
$\text{Pub}_B \rightarrow$ Public Key of B
$\text{Pvt}_A \rightarrow$ Private Key of A
$\text{Pvt}_B \rightarrow$ Private Key of B
1 $<\text{enc}(G, \text{puf}(c_a)), G, H1, c_a> \leftarrow$ Encrypt for A
2 $<\text{enc}(G, \text{puf}(c_b)), G, H1, c_b> \leftarrow$ Encrypt for B
3 $x_a \rightarrow$ random number
4 $\text{Pub}_A \rightarrow x_a.G$
5 $\text{Pvt}_A \rightarrow x$
6 $y_b \rightarrow$ random number
7 $\text{Pub}_B \rightarrow y_b.G$
8 $\text{Pvt}_B \rightarrow y$
9 $<\text{enc}(\text{Pub}_A, \text{PUF}(c_3)), \text{Pub}_A, c_3>$
10 $\text{Pub}_A = \text{enc}(\text{Pub}_A, \text{PUF}(c_3))$
11 $<\text{enc}(\text{Pub}_B, \text{PUF}(c_5)), \text{Pub}_B, c_5>$

Algorithm 1: MQV-based key generation protocol.

PUF using Xilinx technology. PUF generates distinct CRPs essential for device authentication and enhancing security. To manage and store these CRPs, a MYSQL database is employed as a secure repository. The generation of elliptic curve key pairs is facilitated by the Miracle SDK library. Code development using Miracle SDK culminates in the creation of efl files. These files, once cross-compiled for the ARM processor architecture, are installed on the Zybo Zynq-7000 boards. Subsequently, the code is executed on these boards, and the results are retrieved. This integrated experimental setup provides a robust platform for the thorough validation of the proposed method. It leverages the Zybo boards' versatility, the DRAM-based PUF, and the capabilities of the Miracle SDK library, all while ensuring data integrity and secure management through the MYSQL database. It is a key component in assessing the security of the communication protocol under scrutiny.

In order to make a comparative study with existing solutions, we implemented two IoT communication protocols published recently. Algorithm 1 shows the implementation.

4.1. Implementation of Proposed Method. The proposed scheme has been implemented using C language and Miracle SDK. MySQL server has been used to simulate the repository server.

The proposed scheme has been realized through C language and Miracle SDK. The implementation incorporates MySQL server to emulate the functionalities of the SS. In this simulation environment, HP i-5 laptops play a dual role, effectively mimicking both the cloud server and edge node. To replicate the diverse characteristics of IoT nodes, we employed various Xilinx boards in conjunction with child processes spawned using C language, creating a dynamic ecosystem that closely mimics the intricacies of IoT devices. This innovative integration ensures a holistic representation of our proposed solution, with each component contributing to the emulation of real-world scenarios. The synergy between C language, Miracle SDK, MySQL, and the diverse hardware components employed underscores our commitment to creating a robust, versatile, and scalable implementation.

4.2. Implementation of IBE Cryptography. We have used the algorithm mentioned for certificateless cryptography with the following modification:

(1) Removed Weil pairing method.
(2) Added partial key extraction phase.

4.3. Results. The proposed method, alongside two other communication protocols—certificateless cryptography and cloud-based IBE cryptography—has undergone comprehensive evaluation within the previously described experimental environment. The evaluations involved the analysis of results derived from elliptic curves with varying key sizes, specifically 160, 192, and 224 bits. These results serve as a critical measure of the performance and security attributes of each protocol, shedding light on their effectiveness and suitability for the intended communication tasks.

4.3.1. Parameters. Parameters used in the analysis are given below:

(1) Execution time: It represents the duration taken to perform the code block.
(2) Energy consumption: It is determined by deducting the energy consumed in the previous sample from the present sample. A sample denotes the amplitude of remaining energy in a battery.
(3) Storage size: The memory size needed during the execution of process:

$\text{E}_{\text{Enrgy}R} \rightarrow$ `Power dissipated by CPU per unit time;`

$\text{E}_{\text{ExcTime}P} \rightarrow$ `Time needed by process P to finish the execution;`

$\text{E}_{\text{Enrgy}P} \rightarrow$ `Power dissipates by process P in each execution cycle; and`

$\text{E}_{\text{Enrgy}P} \rightarrow \text{Enrgy}_R \times \text{ExcTime}_P$

4.3.2. Computation Overhead. In Table 3, we present the execution times for the creation of a single key pair, which is a crucial metric in assessing the efficiency of the given protocol. This table also includes execution times for the other two existing algorithms for reference. In Column 3 of Table 3, the execution times of the proposed method and the two other communication protocols are compared. Additionally, the power dissipation, shown in Column 4 of Table 3, is documented for the proposed method and the other two communication protocols. It is important to note that the evaluation encompasses various elliptic curve key sizes, including 160, 192, and 224 bits, to provide a

Table 3: Evaluation of single key-pair generation.

Methods	ECC-bits	Time (μs)	Energy (μJ)
Traditional PKI based	P-160	9.493	6.1
	P-192	13.25	18
	P-224	16.2425	27.34
Cloud-based protocol	P-160	2.002	0.797
	P-192	6.258	1.13
	P-224	5.767	2.01
Edge base proposed protocol	P-160	1.305	0.634
	P-192	3.893	0.93
	P-224	4.538	1.89

Table 4: Evaluation of generating scheme with concurrent keypair.

Methods	ECC-bits	Time (ms)	Energy (J)
Traditional PKI	P-160	13.444004	7.355072
	P-192	17.535036	18.40548
	P-224	22.75026	32.761354
Cloud-based protocol	P-160	3.6036	0.897609
	P-192	14.3934	1.321412
	P-224	14.4175	2.112316
Edge base proposed protocol	P-160	2.4312	0.634586
	P-192	8.2825	0.905251
	P-224	9.3746	1.602835

comprehensive view of the performance across different security levels.

Table 4 explains the measurements taken when the protocols are executed on concurrent key creation requests for 250 users. From the outcome mentioned in the aforesaid tables, we can conclude the following:

(1) Due to MQV elliptic scalar operation given protocols outsmart the existing protocols. Given protocol is $2-3\times$ faster than the existing protocols. It may be recall that reason for slowness of the existing protocols is they use bilinear pairing for key generation process. Our findings is corroborated with the result mentioned in Cao et al.'s [45] study where it is shown that peer protocols are $5\times$ slower with respect to the proposed protocol.

(2) Comparison made on energy consumption explains that the given protocols consumes $3-4\times$ less power than the peer protocols.

(3) Outcome mentioned here clearly indicates that execution time, energy consumption changes with respect to the size of ECC. For example, time needed to produce a single key pair for a 160 bits elliptic curve by the given protocol is $1.3\times$ quicker than 192 bits and $2\times$ quicker than 224 bits. Likewise, power dissipation varies with respect to the key size.

(4) Execution time linearly increased when number of users are increased for both single-user and concurrent users.

Figures 4 and 5 present the outcome for execution time and power dissipation in graphical chart. The vertical axis in Figure 4 denotes to the time requires to complete the operations in ms, whereas elliptic curve size is mentioned in horizontal axis. Figure 4 represents that time needed to produce a pair of private and public key using the given protocol as well as the other communication protocols method has grown elliptic curve size. However, the given protocol yields far superior outcome compared to the state-of-the-art protocols. In Figure 5, the vertical axis indicates the power dissipation in microjoule, and horizontal axis represents the size of the elliptic curve. Figure 5 concludes that power consumption of the given protocol and the existing schemes is growing with respect to elliptic curve size. Nevertheless, the given method yields a far superior result than the existing protocols. Mathematical comparison indicated that the given protocol is $4-7\times$ quicker than the other two protocols. It is also clear that the power dissipation is $5-10\times$ better than the other protocols.

4.4. Memory Consumption. Memory size occupied by the given protocol at run time is shown in Table 5. It also describes the memory size consumed by the other two existing protocols. Clearly given method takes less memory than the existing protocols.

5. Security Analysis

5.1. Adversarial Model. Characteristics of an adversary are defined as follows:

(1) Adversary will have full authority over the communication channel, i.e., adversary will be able to replay, add, update the data exchanged over the channel. It can also monitor the data transmitted over the channel.

(2) The adversary can perform the following operations in polynomial time.

(i) Invoke(timestamp, challenge, response, node)—Adversary can execute this task to ask for the public key of a particular node.

(ii) Read (KC, channel, node)—Message that is exchanged between KC and a node can be intercepted using this operation.

(iii) MemoryRead (node)—Data stored in the memory of a specific node can be read using this method.

(iv) Get (node, KC)—Public key of a specific node can be obtained using this function.

(v) CorruptNode()—It simulates corrupted node.

(vi) CorruptKC()—It simulates the compromised KGC.

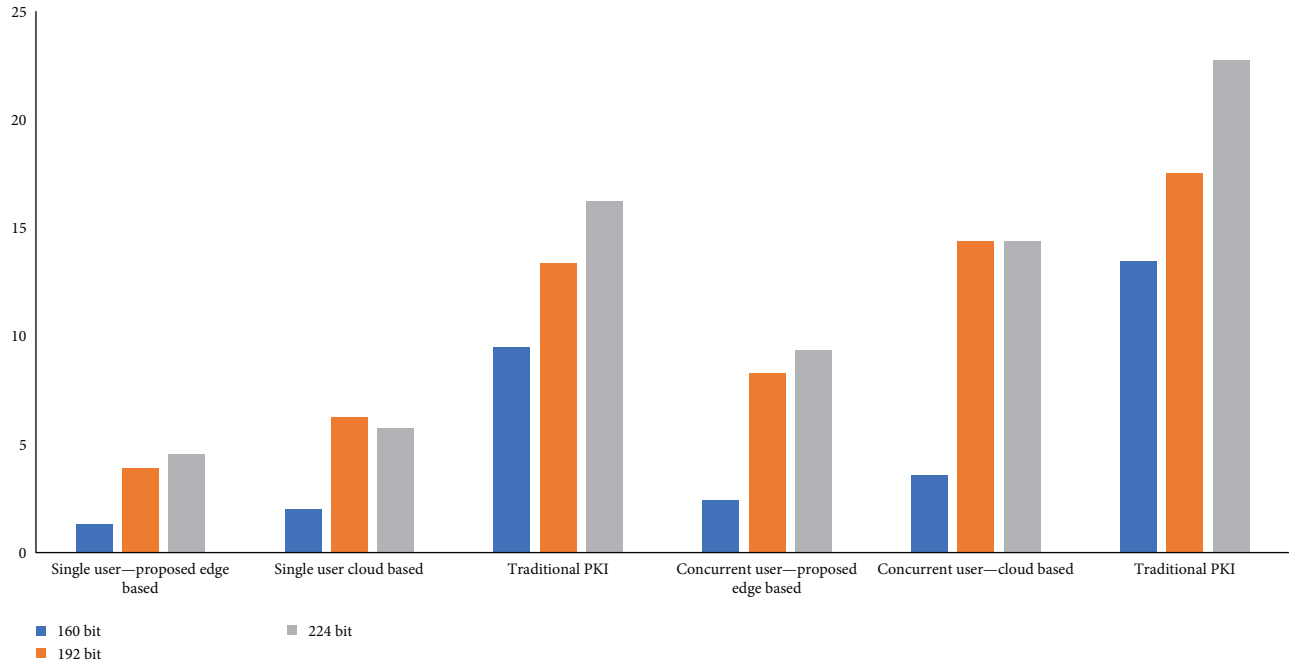

Figure 4: Evaluation of processing time (ms).

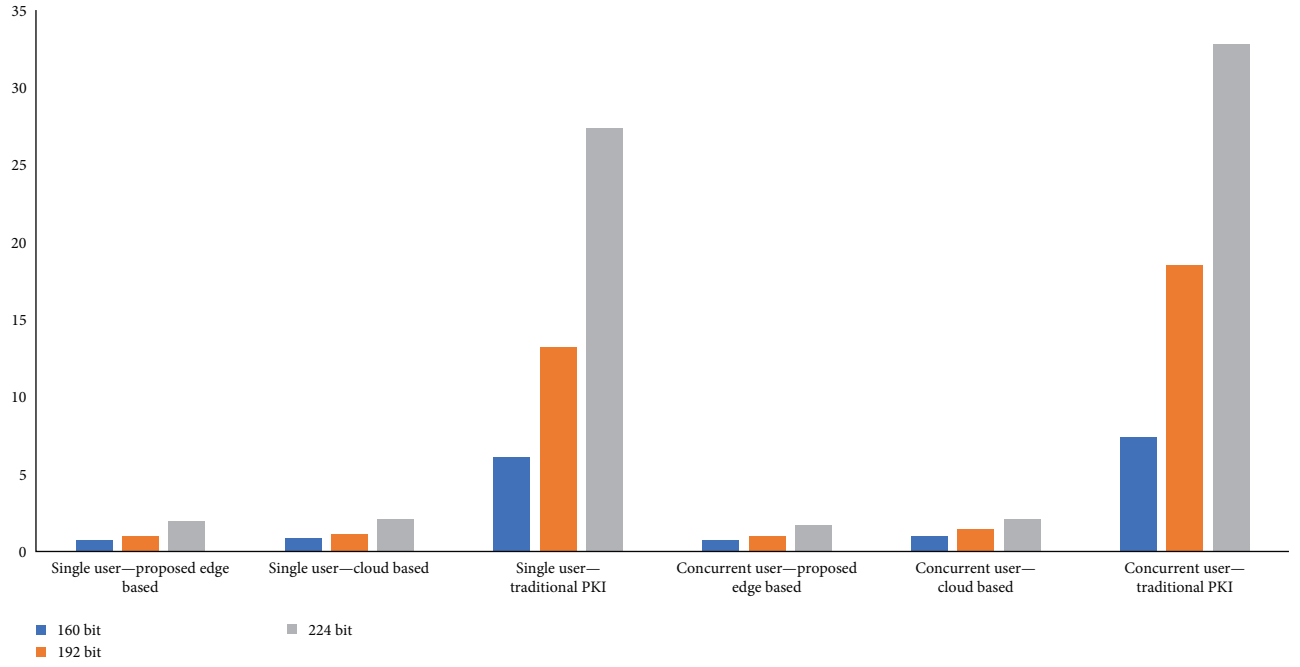

Figure 5: Evaluation of power consumption (J).

5.2. Assumptions. We made the following assumptions:

(1) It is not possible to mathematically model the physically unclonable function (PUF).

(2) MQV ECC problem cannot be broken.

5.3. Formal Security Analysis Using Real or Random Model. Formal security of the given method has been explained by employing ROR model. We have considered S as an adversary, M^{n} as $\boldsymbol{n}$_th instance of a participant. Few such participants are IoT nodes such as Inode_i and Inode_k. Adversary S can talk to participant M^n. M^k-Inode_k and M^{n}-Inode_i indicate $\boldsymbol{k}$_th and $\boldsymbol{n}$_th instances of Inode_i and Inode_k. $\mathrm{h}(\cdot)$ is modeled collision resistant one-way hash function. This hash function can be invoked by Adversary S and other entities of the protocol.

Table 5: Memory consumption in KB.

Methods	ECC bits	Single user	Concurrent users
Traditional PKI	P-160	3,809.92	24,794.56
	P-192	9,313.11	75,827.24
	P-224	12,347.93	93,290.05
Cloud-based protocol	P-160	160.05	1,125.57
	P-192	1,125.57	5,303.21
	P-224	1,165.64	8,490.73
Edge-based proposed protocol	P-160	138.32	890.89
	P-192	632.84	4,987.64
	P-224	1,093.43	78,297.50

Theorem 1. *Proposed method fulfills the SK-security. It is proved in the following way. An Adversary S run the protocol in polynomial time t. Advantage of S to get the secret key SK_ik by breaking the protocol's security while the key exchange phase can be estimated as follows:*

$$\mathrm{Adv}^{\mathrm{PMethod}}{}_A(t) \leq (q^2{}_{\mathrm{hash}}/|\mathrm{Hash}|) + 2\mathrm{Adv}^{\mathrm{ECMQV}}{}_A(t). \tag{1}$$

Proof. We have taken help of three games namely G1, G2, and G3 to prove the above mentioned theorem. An event is denoted as follows:

$$\mathrm{SUCCESS}^{Gj}{}_S, \tag{2}$$

where Adversary S can guess the random bit c in the G_j correctly.

Advantage probability winning the game G_j for S is as follows:

$$\mathrm{Adv}^{\mathrm{PMethod}}{}_{S,Gj} = \Pr[\mathrm{SUCCESS}^{Gj}{}_S]. \tag{3}$$

Scheme explained in Wazid et al.'s [46] study has been used to prove this game. □

Game G0: We have simulated the actual attack by Adversary S against the given method using ROR model. Here bit c is selected randomly before start of the game G0. So it can be deducted from semantic security as follows:

$$\mathrm{Adv}^{\mathrm{PMethod}}{}_A(t) = \left[2.\mathrm{Adv}^{\mathrm{PMethod}}{}_{A,\mathrm{G0}} - 1\right]. \tag{4}$$

Game G1: Eavesdropping attack has been included in the game. In this game, Adversary S can eavesdrop all the messages exchanged during key exchange. Secret key generated by adversary is compared with the real keys exchanged between nodes using reveal and test query.

$\mathrm{PUF}(C_k)$ and $\mathrm{PUF}(C_i)$ define the security of the key SK_{ik}. Since PUF responses are unknown to S so eavesdropping of the messages will never increase S's winning probability in the game G1. Also the games G0 and G1 are indistinguishable so we got the following result:

$$\mathrm{Adv}^{\mathrm{PMethod}}{}_{A,\mathrm{G1}} = \mathrm{Adv}^{\mathrm{PMethod}}{}_{A,\mathrm{G0}}. \tag{5}$$

Game G2: Game G2 used hash function H. An active attack has been modeled in this game. DATA transferred between two IoT nodes are protected using collision resistant hash function which $h(\mathrm{Node}_B + \mathrm{Pub}_A + \mathrm{PUF}(\mathrm{C}_2))$or $h(\mathrm{Node}_B + \mathrm{Pub}_A + \mathrm{PUF}(\mathrm{C}_1))$ a computationally infeasible task for the Adversary S. Moreover, $\mathrm{PUF}(\mathrm{C}_1)$ and $\mathrm{PUF}(\mathrm{C}_2)$ are unknown to S. Also G1 and G2 are "indistinguishable." Game G2 has included Hash query. The birthday paradox deduces the following equation [47]:

$$\begin{aligned}&\left|\mathrm{Adv}^{\mathrm{PMethod}}{}_{A,\mathrm{G1}} - \mathrm{Adv}^{\mathrm{PMethod}}{}_{A,\mathrm{G2}}\right| \\ &\leq (q^2{}_{\mathrm{hash}}/2|\mathrm{hash}|) + \mathrm{Adv}^{\mathrm{ECMQV}}{}_A(t).\end{aligned} \tag{6}$$

Adversary S modeled all the scenarios. Only item remaining is to guess the bit c to win the game. Therefore, we have the following equation:

$$\mathrm{Adv}^{\mathrm{PMethod}}{}_{A,\mathrm{G2}} = 1/2. \tag{7}$$

By combining all the above equations:, we get the following equation:

$$\begin{aligned}&1/2\mathrm{Adv}^{\mathrm{PMethod}}{}_A(t) \\ &= \left[\mathrm{Adv}^{\mathrm{PMethod}}{}_{A,\mathrm{G0}}\$ - \$1/2\right] \\ &= \left|\mathrm{Adv}^{\mathrm{PMethod}}{}_{A,\mathrm{G1}} - \mathrm{Adv}^{\mathrm{PMethod}}{}_{A,\mathrm{G2}}\right| \leq (q^2{}_{\mathrm{hash}}/2|\mathrm{hash}|) \\ &\quad + \mathrm{Adv}^{\mathrm{ECMQV}}{}_A(t).\end{aligned} \tag{8}$$

5.3.1. Informal Security Analysis

Lemma 1. *Proposed protocol is free from Denial of service attack (DoS). It allows any IoT node to request for keys from other IoT nodes.*

Proof. Key generation center (KGC) hosted in cloud server validates the request received by a particular node using CRP value store in its database. KC will reject the message before initiating the key generation process if it finds message with random CRP. All the request traffic are protected by PUF challenge. Response of the corresponding challenge is kept in

edge server (ES) only. It is impossible for the adversary to get the access of ES and fetch a valid CRP to protect the request. So, it is not possible for adversary to deny KC and perform the DoS attack. □

Lemma 2. *Public key of a node cannot be modified.*

Proof. Messages communicated between a node and KC can be intercepted by an adversary. But it can not modify the public key. Because public key information is XORed using PUF response. □

Lemma 3. *Replay attack cannot be conducted by the adversary in the proposed protocol.*

Proof. In the adversarial model, messages transmitted between any node and KC can be requested, read, and received by an adversary. These messages can be intercepted by the adversary and sent them back to the KC to carry out replay attack. Each messages in the proposed scheme contains a timestamp which is encrypted using PUF response, KC will verify the timestamp before approving a request. The KC would remove a message if it contains an expired timestamp. So an adversary cannot carry out the replay attack. □

Lemma 4. *Given solution maintains untraceability.*

Proof. Nodes are identified using PUF responses. This identity is also included in the messages transferred between any two entities. Since it is not possible for an adversary to guess the PUF response of a particular challenge, adversary would not be able to identify the node that has sent the message.□

6. Conclusion and Future Works

This paper introduced a generalized PKI-based ECC authentication and key generation protocol that enhances communication within IoT ecosystems by employing ECMQV. The protocol's primary advantages lie in its ability to significantly reduce energy consumption and execution time. It achieves this through the incorporation of a unique elliptic curve known as MQV elliptic curve, which enhances both security and performance in terms of execution time and memory usage. The protocol's design leverages in-built circuits in user nodes to generate the MQV elliptic curve. The experimental results corroborate the effectiveness of the proposed protocol, demonstrating that it outperforms its peer processes by being 1.5 times faster and exhibiting lower energy dissipation. These attributes are particularly valuable for resource-constrained IoT devices. Future extensions of this work may involve automating the enrollment process to eliminate the need for manual intervention in symmetric key distribution to individual user nodes. Additionally, a revocation mechanism could be introduced for enhanced security. Furthermore, it is worth considering the impact of environmental temperature on the quality of secure keys generated from PUF. Investigating whether the proposed method consistently delivers high-quality PUF output under varying ambient temperatures could be a valuable avenue of research for designing a robust authentication scheme.

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] A. Adeel, M. Ali, A. N. Khan et al., "A multi-attack resilient lightweight IoT authentication scheme," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 3, Article ID e3676, 2022.

[2] W. He, G. Yan, and L. D. Xu, "Developing vehicular data cloud services in the IoT environment," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1587–1595, 2014.

[3] G. Yang, L. Xie, M. Mäntysalo et al., "A health-IoT platform based on the integration of intelligent packaging, unobtrusive bio-sensor, and intelligent medicine box," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2180–2191, 2014.

[4] J. Zhang, S. Rajendran, Z. Sun, R. Woods, and L. Hanzo, "Physical layer security for the internet of things: authentication and key generation," *IEEE Wireless Communications*, vol. 26, no. 5, pp. 92–98, 2019.

[5] Y. Sun, Y. Cui, Y. Huang, and Z. Lin, "SDMP: a secure detector for epidemic disease file based on DNN," *Information Fusion*, vol. 68, pp. 1–7, 2021.

[6] Y. Sun, K. Yu, A. K. Bashir, and X. Liao, "B*l*-IEA: a bit-level image encryption algorithm for cognitive services in intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 1, pp. 1062–1074, 2023.

[7] X. Yao, H. Kong, H. Liu, T. Qiu, and H. Ning, "An attribute credential based public key scheme for fog computing in digital manufacturing," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2297–2307, 2019.

[8] C. Osborne, "Researchers discover over 170 million exposed IoT devices in major US cities—zdnet," 2018.

[9] A. Shamir, A. Biryukov, and L. P. Perrin, "Summary of an open discussion on IoT and lightweight cryptography," in *Proceedings of ESC'17 —Early Symmetric Crypto Workshop*, University of Luxembourg, 2017.

[10] E. J. Marinissen, Y. Zorian, M. Konijnenburg et al., "IoT: source of test challenges," in *2016 21th IEEE European Test Symposium (ETS)*, pp. 1–10, IEEE, Amsterdam, Netherlands, May 2016.

[11] V. Sureshkumar, S. Mugunthan, and R. Amin, "An enhanced mutually authenticated security protocol with key establishment for cloud enabled smart vehicle to grid network," *Peer-to-Peer Networking and Applications*, vol. 15, pp. 2347–2363, 2022.

[12] K. Mahmood, J. Ferzund, M. A. Saleem, S. Shamshad, A. K. Das, and Y. Park, "A provably secure mobile user authentication scheme for big data collection in IoT-enabled maritime intelligent transportation system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 2, pp. 2411–2421, 2023.

[13] N. N. Anandakumar, M. P. L. Das, S. K. Sanadhya, and M. S. Hashmi, "Reconfigurable hardware architecture for authenticated key agreement protocol over binary Edwards curve," *ACM Transactions on Reconfigurable Technology and Systems*, vol. 11, no. 2, pp. 1–19, 2018.

[14] S. Shamshad, K. Mahmood, S. Hussain et al., "An efficient privacy-preserving authenticated key establishment protocol for health monitoring in industrial cyber–physical systems," *IEEE Internet of Things Journal*, vol. 9, no. 7, pp. 5142–5149, 2022.

[15] S. Chanda, A. K. Luhach, W. Alnumay, I. Sengupta, and D. S. Roy, "A lightweight device-level public key infrastructure with DRAM based physical unclonable function (PUF) for secure cyber physical systems," *Computer Communications*, vol. 190, pp. 87–98, 2022.

[16] S. Garg, K. Kaur, G. Kaddoum, J. J. P. C. Rodrigues, and M. Guizani, "Secure and lightweight authentication scheme for smart metering infrastructure in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3548–3557, 2020.

[17] N. N. Anandakumar, M. S. Hashmi, and S. K. Sanadhya, "Field programmable gate array based elliptic curve Menezes–Qu–Vanstone key agreement protocol realization using physical unclonable function and true random number generator primitives," *IET Circuits, Devices & Systems*, vol. 16, no. 5, pp. 382–398, 2022.

[18] A. P. Sarr, P. Elbaz-Vincent, and J.-C. Bajard, "A secure and efficient authenticated Diffie–Hellman protocol," in *Public Key Infrastructures, Services and Applications. EuroPKI 2009.*, F. Martinelli and B. Preneel, Eds., vol. 6391, pp. 83–98, Springer, Berlin, Heidelberg, 2010.

[19] U. Chatterjee, R. S. Chakraborty, and D. Mukhopadhyay, "A PUF-based secure communication protocol for IoT," *ACM Transactions on Embedded Computing Systems*, vol. 16, no. 3, pp. 1–25, 2017.

[20] U. Chatterjee, V. Govindan, R. Sadhukhan et al., "Building PUF based authentication and key exchange protocol for IoT without explicit CRPs in verifier database," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 3, pp. 424–437, 2019.

[21] A. Braeken, "PUF based authentication protocol for IoT," *Symmetry*, vol. 10, no. 8, Article ID 352, 2018.

[22] C. Marchand, L. Bossuet, U. Mureddu, N. Bochard, A. Cherkaoui, and V. Fischer, "Implementation and characterization of a physical unclonable function for IoT: a case study with the TERO-PUF," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 1, pp. 97–109, 2018.

[23] S. U. Hussain, M. Majzoobi, and F. Koushanfar, "A built-in-self-test scheme for online evaluation of physical unclonable functions and true random number generators," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 2, no. 1, pp. 2–16, 2016.

[24] K. Mahmood, S. Shamshad, M. A. Saleem et al., "Blockchain and PUF-based secure key establishment protocol for cross-domain digital twins in industrial Internet of Things architecture," *Journal of Advanced Research*, 2023.

[25] V. S. Miller, Use of elliptic curves in cryptographyin *Conference on the Theory and Application of Cryptographic Techniques*, pp. 417–426, Springer, Berlin, Heidelberg, December 2000.

[26] Z. Liu, X. Huang, Z. Hu, M. K. Khan, H. Seo, and L. Zhou, "On emerging family of elliptic curves to secure Internet of Things: ECC comes of age," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 3, pp. 237–248, 2017.

[27] L. Yang, P. Yu, W. Bailing, B. Xuefeng, Y. Xinling, and L. Geng, "IOT secure transmission based on integration of IBE and PKI/CA," *International Journal of Control and Automation*, vol. 6, no. 2, pp. 245–254, 2013.

[28] M. Ma, D. He, M. K. Khan, and J. Chen, "Certificateless searchable public key encryption scheme for mobile healthcare system," *Computers & Electrical Engineering*, vol. 65, pp. 413–424, 2018.

[29] X. Yao, X. Han, and X. Du, "A light-weight certificate-less public key cryptography scheme based on ECC," in *2014 23rd International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–8, IEEE, Shanghai, China, August 2014.

[30] W. Chen, "An IBE-based security scheme on Internet of Things," in *2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems*, pp. 1046–1049, IEEE, Hangzhou, China, 2012.

[31] G. E. Suh, C. W. O'Donnell, and S. Devadas, "Aegis: a single-chip secure processor," *IEEE Design & Test of Computers*, vol. 24, no. 6, pp. 570–580, 2007.

[32] Q. Zhang, J. Wu, H. Zhong, D. He, and J. Cui, "Efficient anonymous authentication based on physically unclonable function in industrial Internet of Things," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 233–247, 2023.

[33] M. F. Aziz, A. N. Khan, J. Shuja, I. A. Khan, F. G. Khan, and A. R. Khan, "A lightweight and compromise-resilient authentication scheme for IoTs," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 3, Article ID e3813, 2022.

[34] M. Seifelnasr, R. AlTawy, and A. Youssef, "Efficient inter-cloud authentication and micropayment protocol for IoT edge computing," *IEEE Transactions on Network and Service Management*, vol. 18, no. 4, pp. 4420–4433, 2021.

[35] A. Yang, J. Weng, K. Yang, C. Huang, and X. Shen, "Delegating authentication to edge: a decentralized authentication architecture for vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 1284–1298, 2022.

[36] Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng, and Y. Zhang, "Consortium blockchain for secure energy trading in industrial Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3690–3700, 2018.

[37] Z. S. Paral and S. Devadas, "Reliable and efficient PUF-based key generation using pattern matching," in *2011 IEEE International Symposium on Hardware-Oriented Security and Trust*, pp. 128–133, IEEE, San Diego, CA, USA, June 2011.

[38] S.-H. Seo, M. Nabeel, X. Ding, and E. Bertino, "An efficient certificateless encryption for secure data sharing in public clouds," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 9, pp. 2107–2119, 2014.

[39] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," *SIAM Journal on Computing*, vol. 32, no. 3, pp. 586–615, 2003.

[40] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Annual International Cryptology Conference. CRYPTO 2001: Advances in Cryptology—CRYPTO 2001*, pp. 213–229, Springer, Berlin, Heidelberg, August 2001.

[41] S. Shamshad, M. F. Ayub, K. Mahmood, M. Rana, A. Shafiq, and J. J. P. C. Rodrigues, "An identity-based authentication protocol for the telecare medical information system (TMIS)

using a physically unclonable function," *IEEE Systems Journal*, vol. 16, no. 3, pp. 4831–4838, 2022.

[42] S.-Y. Park, S. Lim, D. Jeong, J. Lee, J.-S. Yang, and H. Lee, "PUFSec: device fingerprint-based security architecture for Internet of Things," in *IEEE INFOCOM 2017—IEEE Conference on Computer Communications*, pp. 1–9, IEEE, Atlanta, GA, USA, May 2017.

[43] W. Xiong, A. Schaller, N. Anagnostopoulos et al., "Run-time accessible DRAM PUFs in commodity devices," in *International Conference on Cryptographic Hardware and Embedded Systems*, pp. 432–453, Springer, Berlin, Heidelberg, August 2016.

[44] M. F. Haroun and T. A. Gulliver, "Secret key generation using chaotic signals over frequency selective fading channels," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 8, pp. 1764–1775, 2015.

[45] X. Cao, W. Kou, and X. Du, "A pairing-free identity-based authenticated key agreement protocol with minimal message exchanges," *Information Sciences*, vol. 180, no. 15, pp. 2895–2903, 2010.

[46] 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, 2018.

[47] S. Malani, J. Srinivas, A. K. Das, K. Srinathan, and M. Jo, "Certificate-based anonymous device access control scheme for IoT environment," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9762–9773, 2019.