Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 11, No. 01, March 2025.

BLOCKCHAIN-BASED DEVICE AUTHENTICATION IN EDGE COMPUTING USING QUANTUM APPROACH

Vinayak A. Telsang[1], Mahabaleshwar S. Kakkasageri[2] and Anil D. Devangavi[3]

(Received: 26-Aug.-2024, Revised: 25-Oct.-2024, Accepted: 28-Oct.-2024)

ABSTRACT

The Internet of things (IoT) emerged as a new technology, where everything is connected. Large amounts of data need to be stored for processing; hence, edge computing can reduce the storage of data in a distributed environment, which enhances processing speed and low usage of bandwidth. With an ever- increasing use of IoT devices, issues such as authentication of devices, privacy of data stored and integrity of data have also increased. The authentication of devices is a major concern for edge-connected IoT devices. The problem was solved by using classical cryptographic algorithms such as Elliptic Curve Cryptography (ECC), Rivest-Shamir-Adleman (RSA) and Diffie-Hellman (DH) for message encryption by using public and private keys that need to be stored. These keys need to be stored on a server for device authentication. In device authentication, storing many keys leads to more computation and storage costs and to an increase in delay. With quantum computing and quantum algorithms, such as Shor's and Grover's, it becomes easy to break the keys of cryptographic algorithms, making the system vulnerable. The proposed work Blockchain-based Device Authentication in Edge Computing Using Quantum Approach (BDAEC-QA) provides authentication for IoT devices using context information, quantum key distribution (QKD) and blockchain. The proposed scheme uses the smart contracts to store an information of the IoT devices on the server side, which is used by blockchain to provide secure authentication between the edge server and the IoT devices. The proposed scheme also provides communication between IoT devices across the network. The proposed work is compared with "Lightweight Two-factor-based User Authentication Protocol for IoT-Enabled Healthcare Ecosystem in Quantum Computing" (LTBA) and "A Blockchain-based Mutual Authentication Scheme for Collaborative Edge Computing" (BBMA) and has less registration, key generation and authentication delay, respectively. The BDAEC-QA scheme uses less computation and storage costs as compared with other existing schemes. The proposed scheme is simulated using the AVISPA tool, to provide the security proofs and analysis that indicate that the BDAEC-QA scheme is resistant to well-known attacks.

KEYWORDS

IoT, 5G, Data security, Cryptic algorithms, Blockchain.

1. INTRODUCTION

The traditional cloud-computing technology is a centralized server that allows users to access resources as and when needed [1]. But, centralized computing technology suffers from denial of service (DoS) and distributed denial-of-service (DDoS) attacks. With the growth of connected devices, cloud computing suffers from latency, quality of service (QoS) and time delays. Hence, edge computing emerged as a new alternative to compute, store and process data at the edge of the network [2]. In the era of the internet of everything (IoE) and the advent of industry 4.0, edge-computing technology has become very popular in the Industrial Internet of Things (IIoT) [3]. In IIoT devices, such as sensors, mobile phones, …etc. are connected to an edge server where computation occurs, reducing transmission time and network traffic and improving QoS [4].

With the rapid development of the IoT, the security and privacy issues of IoT devices are issues of concern [5]. Issues, such as authentication, confidentiality and integrity, need to be addressed. The authentication of connected devices is a real challenge, because if the authentication of the connected device does not occur in the network, it leads to leakage of sensitive data. To solve this problem, many authors have proposed schemes that are based on classical cryptosystems, using public and private keys. The private key is used for encrypting the data, while the public key is used for decrypting the

1. V. A. Telsang is with Biluru Gurubasava Mahaswamiji Institute of Technology, Department of Computer Science and Engineering, Mudhol-587317, Visvesvaraya Technological University, Belagavi, Karnataka, India. Email: v.telsang@gmail.com
2. M. S. Kakkasageri is with Basaveshwar Engineering College, Department of Electronics and Communication Engineering, Bagalkote587102, Visvesvaraya Technological University, Belagavi, Karnataka, India. Email: mahabalesh_sk@yahoo.co.in
3. A. D. Devangavi is with Basaveshwar Engineering College, Department of Artificial Intelligence and Machine Learning, Bagalkote587102, Visvesvaraya Technological University, Belagavi, Karnataka, India. Email: anildevangavi_s@yahoo.co.in

data and *vice versa*. Other techniques include hashing of the data by using public or private keys [6], one-time password [7], secret key mechanisms [8], biometric verification [9], third-party servers for key generation, distribution and verification [10]. All such cryptosystems are guaranteed by the hardness of the discrete logarithmic problem that they have adopted. But, if any advanced system overcomes the hardness employed by those cryptosystems, then the system will be compromised.

With the growth of quantum computing, solving the hardness of mathematical problems of traditional cryptographic algorithms will not be an issue. The security of such a cryptosystem, which is guaranteed by the keys used by classical cryptographic algorithms, may become easy to crack [11]. Since the use of quantum algorithms, such as Shor's and Grover's algorithms which can break the cryptosystems in the near future. So one should incorporate the principles of Quantum Cryptography (QC), based on photons and their quantum properties, the photons have different quantum states measured at any time, which helps in developing a secure cryptosystem [12]. The cryptosystem, which is developed using QC mechanics, is believed to be more secure and nearly impossible to break [13].

The classical algorithms use keys to be generated and stored on the server for IoT device verification during the authentication phase. The security of keys is again a major concern, because it can lead to impersonation, man in middle and eavesdropping attacks [14]. Since IoT devices are distributed in nature and due to the growth of blockchain technology as a result of its decentralized nature, cryptographic properties and improved reliability, as well as fault tolerance and unforgeability makes it suitable for providing a solution to store data along with those generated keys [15].

Blockchain technology supports peer-to-peer networking; whenever a transaction occurs, it is verified by the node before being added to the blockchain. The blockchain contains a number of blocks and each block contains a set of transactions that are structured in the merkle hash tree. Whenever a new block is added, the previous block's hash value is stored in the new block to create the structure of the entire blockchain, which ensures that data cannot be modified. The node maintains the ledger and is updated whenever transaction occurs. The ledger maintains the number of blocks that are chained together with a hash mechanism by storing information, like time stamp, hash of current block and hash of previous block. The blockchian is implemented through different consensus algorithms, such as: Proof of Work (PoW) in which the node solves mathematical calculation to add to the blockchain. Proof of Stack (PoS) uses cryptocurrency validation to select the node. Byzantine Fault Tolerance (BFT) is used in the network of nodes where nodes exchange messages and reach consensus [16].

Smart contracts are unchangeable or immutable computer codes that carry out terms according to the occurrence of a set of pre-determined events. Leveraging blockchain technology, smart contracts enable trusted transactions and agreements among anonymous entities without the help of a central authority or an additional enforcement mechanism [17].

Quantum cryptography is an area that helps develop the cryptosystem using the rules of quantum mechanics. The quantum mechanics uses the smallest unit called the qubit, which is in two quantum states: 0 or $\{|0\rangle\}$ or 1 or $\{|1\rangle\}$. Quantum cryptography is based on using photons and their qubit properties to develop unbreakable cryptosystems. The photon exists in more than one state simultaneously and the state is changed when measured [18]. Quantum key distribution is a technique used in quantum cryptography, where a stream of photons is used to transmit data. These photons have a property called a spin, which is of 3 types: Horizontal (⟷), Vertical (↕) and Diagonal (↗) or (↖). Whenever a message is transferred from party A to party B, A sends the polarized bits by using the randomly chosen bases (+) (×). On receiving the polarized bits, B chooses the basis and calculates the polarized bits. The polarized bits, which are similar to parties A and B, are used as a quantum key between them [19].

1.1 Objectives

The objectives of the proposed BDAEC-QA scheme are as follows:

- To reduce the cryptographic attack by using quantum mechanics by storing context information of IoT devices.
- To enhance the security of IoT device information by storing it at the edge server using the blockchain.
- To reduce computation and storage costs at the edge server by using a quantum key (QKey).

1.2 Contributions

The authentication of IoT devices in an edge-based network is solved through quantum mechanics, which includes three phases: initialization, key generation, distribution and authentication. In addition, a security analysis of the proposed scheme is also discussed.

- The proposed quantum-based authentication scheme identifies IoT devices using their context information and QKey.
- The use of blockchain to store information of IoT devices at the edge server using smart contracts.
- Establishing communication between IoT devices within the vicinity of the edge server and outside the edge server.

The rest of the paper is organized as follows. Related research works are presented in Section 2. The proposed scheme for authentication of IoT devices is presented in Section 3. Simulation and analysis result are discussed in Section 4. The result discussion of the proposed scheme with different schemes is given in Section 5. Section 6 presents a conclusion.

2. RELATED WORKS

The authentication scheme discussed in [20] has initialization phases consisting of system registration and device registration. Each device has an ID (EID) once it registers and the system ID (SID) is provided by system admin and gets the registration token. The token is stored in the blockchain by using a smart contract with information about the SID, EID and device address (EIP) and an authpass is given to each device. Whenever the authentication of a device is requested, it sends the authpass to fog nodes by encrypting the request using its private key; decryption of the request is carried out using the public key of the device at the fog node. The blockchain enabled fog node verifies the EID present in the blockchain as well as the smart contract. If verification is successful, then authentication is successful; otherwise, the device request is rejected. A computation time of 1.06 ms and a power consumption of 7.24 mW are achieved.

The blockchain-based authentication mechanism is discussed in [21]. It uses smart contract to store the user’s request. The miner nodes are used to check the smart contracts of IoT devices. The miner node generates the token for the device upon a token-generation request from the device. The token is signed with its private key and sent to the requested device. During the verification phase, the signed token is issued to the blockchain and if verified successfully, authentication is successful. The scheme achieves a communication delay of 1.6 sec and a communication overhead of 3 sec. A post-quantum fuzzy commitment scheme is provided in [22] and used for the healthcare system. Here, the user must register and authenticate herself/himself with the medical server to access the medical data. The medical data is collected and measured using a smart card and biometric data. The verification of the medical data is successful if the extracted value matches the biometric data and the smart card. The system is complex and it becomes difficult for device authentication with more parameters. The work achieves a computational cost of 20 msec.

The quantum communication authentication for drones discussed in [23] uses a database server to store pre-shared private information with both the ground station and the legitimate drones. The private information of the drone, random key and quantum states is encoded with a private key and sent to the ground station. A random key is used, which guarantees the security of the secret messages. The drone and ground stations authenticate themselves through the secret messages. The schemes provide the secure communication by solving information leakage by detecting the probability of attacks as 0.998. According to the hybrid authentication mechanism based on the vehicle-access network scheme discussed in [24], the scheme identifies information and uses a hash function. The vehicle-resource utilization is efficient, since it uses a multi-vehicle task-management model. For messages between 10-80, the scheme achieves an authentication time of 10-45 msec with a loss rate of 15% and a latency of 35 msec. The resource consumption of the scheme can be optimized by using the master node.

The static and mobile IoT devices using certificate-less cryptography provided in [25] elaborated on the key-generation procedures, lightweight key negotiation and mutual authentication for IoT devices between inter-edge and intra-edge servers. The scheme overcomes most security attacks and achieves an authentication time and a registration time of 0-2.2 msec and 0.2-1.4 sec, respectively (for 10-100 devices) with a CPU usage time of 28%. The multi-party protocol based on lattice-based cryptography discussed in [26] generates a pair of master keys by using the security parameters by the server; i.e., master secret key and master public key. The user who wants to be part of the network has to request the server by sending her/his public key. The server generates an identity for the user by using the master secret key and the user's public key. The scheme is power efficient and secures communication by eliminating public certificates. A power consumption of 40mW and a CPU usage of 40% are achieved by this scheme.

The two-factor authentication scheme for medical server provided in [27] has a server where the user requests registration with a user ID and a password and if the user ID does not exist, the server responds with the smart card, which contains the hashed values of the user information. Whenever the user wants to communicate, he/she can use a smart card along with a user ID and a password. The scheme achieves a communication cost of 320-800 bits, with an execution time of 0.095 msec. The scheme is secure with a session key generated for each user and with the use of two-factor authentication.

The protean authentication scheme based on minimal initialization vectors provided in [28] uses an edge server to store initialization vectors (V). The gateway maintains hardware (H) information for the edge along with initialization vectors. During each authentication cycle, H and V are used by the gateway to generate a random number in each cycle as an authentication key and securely transfer that information to the edge server, making it virtually impossible to arrive at the authentication keys. The key is generated at each cycle, which makes the scheme more secure, but it is resource intensive with a voltage drainage at edge and router occurring for every 4 and 3 hours, respectively.

Lattice-based device to device authentication discussed in [29] uses edge computing and blockchain technology to reduce the computation overhead on IoT devices. The decentralized blockchain is used for public-key management which simplifies key revocation and enhances security. The scheme uses: registration phase, where the IoT device is registered by its edge server and its public keys are added to the blockchain ledger. In the authentication and key-agreement phase, registered IoT devices can authenticate each other and generate a shared session key. The distributed ledger ensures that the edge servers verify the authenticity and validity of public keys of IoT devices. The protocol uses less communication cost as compared with other lattice-based schemes and a storage cost of 1536 bits.

The lattice-based authentication for vehicular communications provided in [30] uses the registration phase, where edge nodes register with the cloud server with public keys stored in the blockchain. The blockchain uses hyper-ledger fabric with smart contract for adding edge node public key. During the authentication phase, the edge nodes mutually authenticate each other using session key. The revocation phase involves the raft consensus algorithm which ensures transaction integrity and ensures that the public keys can be modified by authorized edge nodes. A computation cost of 11,046 µsec and a storage cost of 2112 bits were obtained during the analysis of the scheme.

3. PROPOSED SCHEME

3.1 Network Architecture

The cloud servers are placed far from the IoT devices and moving data for computation requires more time. Despite the cloud server's processing power, time-intensive applications could not be dealt with, since they suffer from latency and bandwidth-consumption problems. The edge server can provide a solution to these problems when used in combination with a cloud server. The general 3-layer edge architecture is shown in Figure 1. It consists of physical devices at the device layer, edge servers at the edge layer and service providers at the cloud layer. The physical devices that are in proximity to the edge server are connected to that edge server for information exchange and computation. The edge server collects the information from IoT devices through edge controllers, analyzes it using emerging technologies implemented as generic capabilities, called Application Programming Interface (API) and provides the result. The edge server also implements algorithms, data-security techniques and machine-learning algorithms for computation, analyzing and storing the results. The edge layer is close to the device layer and is more suitable for time-intensive applications and intelligent processing. Hence, it is more secure and efficient as compared with cloud computing.

3.2 Preliminaries

- $ES = \{ES_1, ES_2, ...ES_n\}$, where ES is the set of edge servers.
- $IoT_{dev} = \{IoTdev_1, IoTdev_2, ...IoT\ dev_n\}$, where IoT_{dev} is the set of IoT devices.
- Each edge server and IoT device have their pair of public and private keys for encryption and decryption, respectively. ES has its pair of keys $\{k_{pues}, k_{pres}\}$ and IoT_{dev} has its pair of keys $\{k_{puit}, k_{prit}\}$.
- Each IoT device is assigned with unique device ID (DID) by the edge server.
- The hash value of the *input* is calculated by using one-way hash function h (*input*).
- The encryption function *Encrpt (pu$_{key}$, message)* is used to encrypt the message by using public key.
- The decryption function *Decrpt (pr$_{key}$, message)* is used to decrypt the message by using private key.
- Context information (*CI*) is the information of the device which consists of its MAC address (*M ACadd*), location information (*locinf o*) and timestamp, as shown in Equation (1).

$$CI = \{MACadd, locinfo, timestamp\} \quad (1)$$

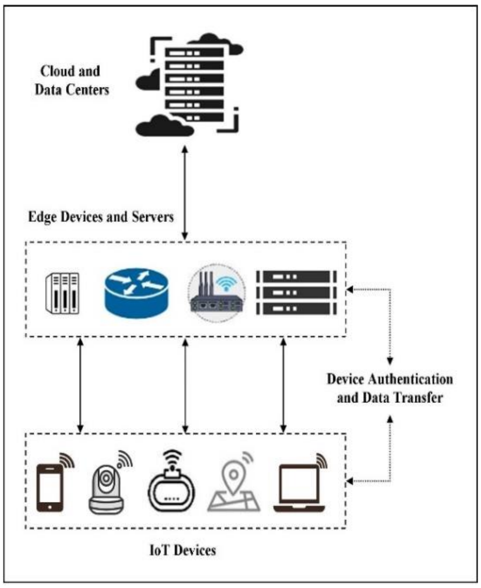


Figure 1. Edge architecture.

3.3 Notations

The notations considered in the proposed BDAEC-QA scheme are listed in Table 1.

Table 1. Notations.

Notations	Description
ES_i,ES_j,ES_m	*Edge server*
IoTdev	*IoT device*
CI	*Context information*
k_{pues},k_{puesi},k_{puesj},k_{puesm}	*Public key of ES*
k_{pres},k_{presi},k_{presj} ,k_{presm}	*Private key of ES*
kpuit	*Public key of IoTdev*
kprit	*Private key of IoTdev*
|	*Concatenation operation*
Mreg	*Registration request*
Mpinfo	*Quantum sequence information Device*
DID	*ID of IoTdev*
QKey	*Quantum key*
Sk	*Session key*
Mesv	*Verification message within edge*
Mesoev	*Verification message outside edge*

3.4 Proposed Architecture

The proposed BDAEC-QA scheme for the authentication of IoT devices is shown in Figure 2. It consists of an IoT layer, an edge layer and a blockchain layer. The IoT layer consists of IoT devices that provide context information that needs to be processed by the edge server. The edge layer consists of edge servers, which store and apply computation to generate the device ID and quantum key. It also communicates with the blockchain layer to store the IoT-device information along with the quantum key using a smart contract. The proposed authentication architecture has three phases: registration, key generation and distribution and authentication, as shown in Figure 2. In the registration phase, the edge server broadcasts its public key in the network. IoT device in the vicinity of the edge server uses the edge server's public key to send its context information to the edge server. After registration, the edge server generates the quantum sequence using quantum bits and basis information, as shown Table 2 and sends the quantum-bit information to the IoT device to begin quantum key generation. In the key generation and distribution phase, the IoT device also generates a quantum sequence using quantum bits and choosing a random basis, where the basis information of the IoT device is sent to the edge server. The edge server, upon receiving basis information, matches and extracts the matched sequence number from its quantum sequence and sends only the matched quantum-sequence number to the IoT device to generate a quantum key (QKey) between the respective IoT device and the edge server. Each edge server stores the information of the requested IoT device in the blockchain by creating the markle tree by using the information sent by the device along with the QKey. In the authentication phase, the IoT device sends an authentication request to the edge server. The edge server verifies the authentication request stored in the blockchain. Based on the verification, the IoT device is either authenticated or unauthenticated. The operations involved in the 3 phases are explained in detail below.

Table 2. Quantum-sequence generation.

Bases	1	0
+	↕	↔
×	↗	↖

3.4.1 Registration

- **IoT-device Registration:** The following steps are involved in the registration process of the IoT device to the edge server.
    - Each edge server has its pair of $\{k_{pues}, k_{pres}\}$ keys. Each edge server broadcasts its public key k_{pues} in the network, so that any IoT device can send a registration-request message M_{reg} by using k_{pues}.
    - The IoT device sends an encrypted message M_{reg} to the edge server by using its public key k_{pues}. The registration message sent from the IoT device consists of context information and its public key k_{puit}.

$$M_{reg} = Encrpt(k_{pues}, CI \mid k_{puit}) \qquad (2)$$

    - The edge server decrypts the M_{reg} by using its k_{pres} as: Decrpt (k_{pres}, M_{reg}) and gets the context information of the IoT device and its public key; i.e., (CI | k_{puit}).
    - The edge server then generates a unique device ID (DID) for each IoT device and registers it along with its CI information. After registering, the ES generates the quantum sequence, as shown in Table 2 by randomly choosing the quantum bits and basis. The quantum-sequence information (M_{pinfo}) is encrypted and sent to the IoT device by using its key k_{puit} as shown in Equation 3. After sending the information, the ES initiates the quantum-key (QKey) generation and distribution phase.

$$M_{pinfo} = Encrpt(k_{puit}, sequence(\updownarrow, \leftrightarrow, \nearrow, \nwarrow)) \qquad (3)$$

- **Edge-server Registration:** The edge server registers with the cloud server by using the public key of the cloud k_{pucs} and sends the registration message as $M\ es_{es}$=*Encrypt* (k_{pucs}, (ID_{es} $\mid \boldsymbol{n_1}$)). The cloud server, after receiving the message ($M\ es_{es}$), replays with a session key (Ski) to the edge server to confirm registration, as shown in Equation 4.

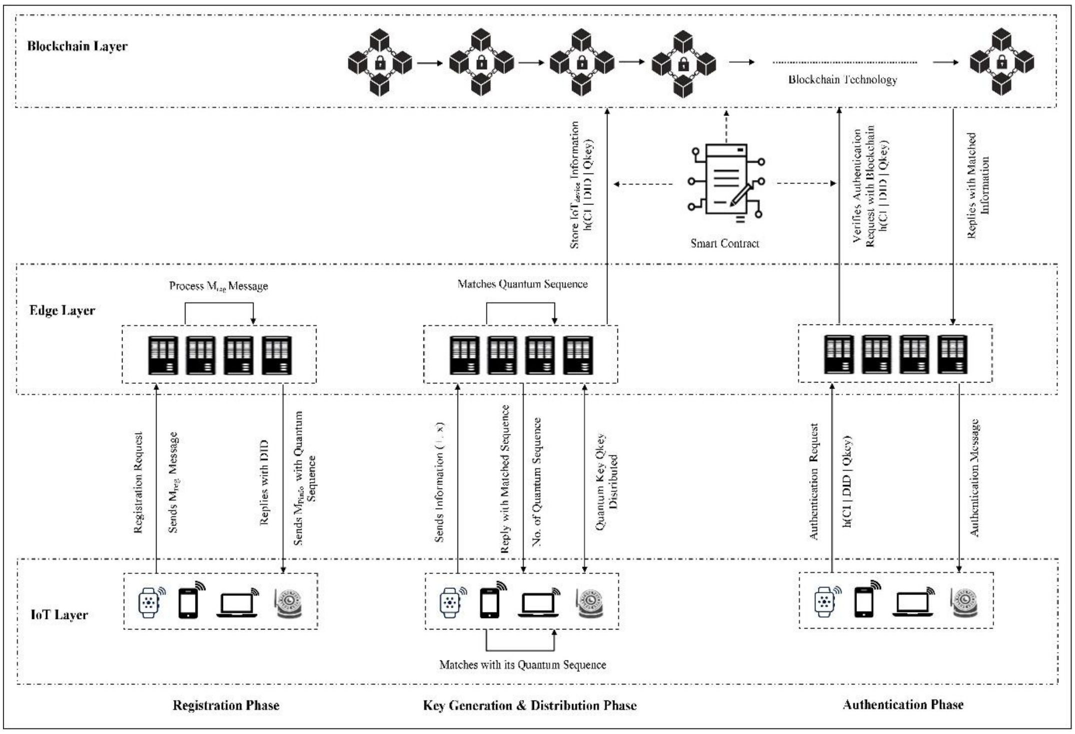

Blockchain Layer
Blockchain Technology
Store IoT$_{device}$ Information h(CI | DID | Qkey)
Smart Contract
Verifies Authentication Request with Blockchain h(CI | DID | Qkey)
Replies with Matched Information
Edge Layer
Process M$_{reg}$ Message
Matches Quantum Sequence
Registration Request
Sends M$_{reg}$ Message
Replies with DID
Sends M$_{Pinfo}$ with Quantum Sequence
Sends Information (1, x)
Reply with Matched Sequence
No. of Quantum Sequence
Quantum Key Qkey Distributed
Authentication Request
h(CI | DID | Qkey)
Authentication Message
IoT Layer
Matches with its Quantum Sequence
Registration Phase
Key Generation & Distribution Phase
Authentication Phase

Figure 2. BDAEC-QA IoT-device authentication scheme.

$$Msg_{es} = Encrpt(k_{pues}, (S_{ki} \mid n1)) \tag{4}$$

The edge server then decrypts the message to get Ski as Decrypt (k_{pres}, $M\ sg_{es}$) to complete the registration process.

3.4.2 Key Generation and Distribution

- The IoT device decrypts the message M_{pinfo} using its private key k_{prit} to receive the quantum-bit sequence information as *Decrpt* (k_{prit}, M_{pinfo}). The IoT device uses this quantum-bit sequence and the randomly generated basis to generate the quantum sequence, as shown in Table 2. Equation 5 represents the generated quantum sequence and Equation 6 represents the four states of the qubits used to generate the QKey.

$$|\psi\rangle_{q1,q2,\ldots qk} = (|000 \ldots 00\rangle_{q1,q2,\ldots qk} + |111 \ldots 11\rangle_{q1,q2,\ldots qk})/\sqrt{2} \tag{5}$$

$$\{|0\rangle, |1\rangle, |+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}, |-\rangle = (|0\rangle - |1\rangle)/\sqrt{2} \tag{6}$$

Algorithm 1. IoT-device Registration

```
Input: Context information(CI), $k_{pues}$, $k_{pres}$, $k_{puit}$
Output: Generating DID for IoT devices
if $IoT_{dev}$ in the vicinity of ES then
    Send CI to ES as shown in Eq.(1)
end if
while true do
    if ES receives CI then
        for each CI received from $IoT_{dev}$ do
            Decrypt $M_{reg}$ as in Eq.(2)
            Generate DID for each $IoT_{dev}$
        end for
        for each $IoT_{dev}$ with DID do
            Send $M_{pinfo}$ as shown in Eq.(3)
        end for
    end if
endwhile
```

- The IoT device then sends its randomly chosen basis-sequence information to the edge server by encrypting it with using k_{pues} as: *Encrpt* (k_{pues}, *sequence* (+, ×)) for quantum-key mapping, as shown in Table 2.
- The edge server decrypts the basis sequence using *kpres* and gets the information as: *sequence* (+, ×) sent by the IoT device. The ES uses IoT_{dev} basis and matches it with its generated quantum sequence. The ES extracts the quantum-sequence number from a matched pair of ES quantum sequences and the IoT quantum sequences to generate an QKey.
- The ES stores the IoT-device information as: $h\ (CI \mid DID \mid QKey)$ in the blockchain by creating a new block. This information is stored for each requested IoT device separately in the ES; i.e., the information uses the CI, DID and the Qkey of the respective device.
- The ES sends the matched quantum-sequence number to the respective IoT_{dev} device by using its DID. Upon receiving IoT_{dev} it is matched with its quantum sequence to get the QKey.
- After the QKey is generated and distributed such that both IoT_{dev} and ES have an Qkey, which is a unique and symmetric key between each other, respectively. The same key is used for authentication between the respective IoT_{dev} and the ES.

3.4.3 Authentication

- Whenever IoT_{dev} wants to communicate, it should be authenticated. For authentication, IoT_{dev} sends an authentication request to the ES as: *Encrpt* (*QKey*, *CI* | *DID*).
- The edge server, upon receiving the authentication request from IoT_{dev} decrypts it using the QKey (already obtained in Phase-2) and matches the information stored in the blockchain. If the authentication request matches, then IoT_{dev} is authenticated; otherwise, it is not authenticated.

Algorithm 2. Key Generation and Distribution

```
Input: Quantum sequence (Qseq) and $M_{pinfo}$
Output: QKey distribution at both ES and $IoT_{dev}$
Edge server sends $M_{pinfo}$ to $IoT_{dev}$
if $IoT_{dev}$ registered then
    Generate Qseq and basis
    Send $M_{pinfo}$ to $IoT_{dev}$
end if
$IoT_{dev}$ decrypts $M_{pinfo}$ and generates Qseq and basis and sends basis information to ES for
  QKey generation
for each $IoT_{dev}$ registered do
    if Qseq of $IoT_{dev}$ = = Qseq of ES then
        ES extracts the matched quantum sequence number ($Qseq_{num}$)
    end if
end for
At ES: The matched quantum sequence number is QKey (generated)
ES sends the quantum sequence number information to $IoT_{dev}$
The IoTdev matches the ES $Qseq_{num}$ to its generated $Qseq_{num}$
for each $IoT_{dev}$ : ES $Qseq_{num}$ do
    if Qseq of $IoT_{dev}$ = = ES $Qseq_{num}$ then
        $IoT_{dev}$ extracts the matched $Qseq_{num}$
    end if
end for
At IoT device: The matched $Qseq_{num}$ is QKey (distributed)
At ES: Stores the $IoT_{dev}$ information in blockchain using smart contract
for each $IoT_{dev}$ : QKey generated do
    create a block information as: $h\ (CI \mid DID \mid QKey)$
    Store the device information in blockchain
end for
```

3.4.4 Communication of IoT Devices

In this phase, IoT devices want to communicate with other IoT devices within the edge network or outside the network. Edge servers interact with each other to validate the IoT devices. The edge servers also share the registered information with the cloud server for communication outside the edge network. The registered device information is shared with the cloud server by using the Ski along with the context information of the IoT device, as shown in Equation 7.

$$M\,sgreg_{IoT} = Encrpt\,(S_{ki}, h\,(CI \mid DID \mid QKey) \mid ID_{esi}) \qquad (7)$$

Algorithm 3. IoT-device Authentication

```
Input: Authentication request ($CI \mid DID, QKey$)
Output: Authentication message
for each Authentication request from $IoT_{dev}$ do
    if Request matches with information stored in blockchain $h\,(CI \mid DID \mid QKey)$ then
        Authentication Successful
    else
        Authentication Unsuccessful
    end if
end for
```

The cloud server then decrypts the information and updates the registered device information for the respective edge server as: Decrypt (S_{ki}, $M\,sgreg_{IoT}$).

- **Within the Same Edge Network**

Whenever an IoT device moves from one edge server (ES_i) to another edge server (ES_j), then the validity of IoTdev has to be checked to communicate within the network of ES_j. The IoTdev sends the its registered information to ES_j as: $M_{\,esv} = Encrypt\,(k_{puesj}, M1)$, where $M1 = (CI \mid DID \mid QKey) \mid kpuesi$. The ES_j then decrypts $M_{\,esv}$ and sends the M1 information to edge server ES_i as: Encrypt (k_{puesi}, $M\ 1$). The server ES_i matches M1 with its registered IoT_{dev} information and replies with a message as "valid" to ES_j, then IoT_{dev} can communicate within the ES_j network.

- **Outside the Edge Network**

When the IoT device IoT_{dev} moves from the edge network, the validity of the IoT_{dev} is not verified outside the edge network. When it sends the message to ES_m as: $M\ es_{oev} = Encrypt\,(k_{puesm},(CI \mid DID \mid QKey) \mid kpuesi)$, ES_m in turn sends the message (cs_{msg}) to the cloud server, as shown in Equation 8.

$$cs_{msg} = Encrypt\,(S_{ki}, (CI \mid DID \mid QKey) \mid k_{puesi}) \qquad (8)$$

The cloud server, after receiving the cs_{msg} decrypts and sends the validity of IoT_{dev} if the information is updated by the edge servers, IoT_{dev} can communicate outside the edge network.

3.5 Case Study Discussion

The QKD-based system uses power, but provides more security while used during key exchange and encryption. Traditional QKD systems use quantum transmitters and receiver components in the network infrastructure, causing more power consumption in large-scale IoT networks. This power hungry nature of the QKD can be optimized by developing quantum hardware, where IoT devices can perform minimum cryptographic operations and quantum operations can be lifted to cloud servers. As the technologies mature and there may be development of chips that can be integrated into low power IoT devices, this makes them consume less power and provide more security with the QKD approach.

Some of the Real World Solutions Using QKD Approaches

The SwissQuantum network testbed deployed in Geneva uses the BB84 protocol for secure communication using QKD [31]. The project shows the feasibility of implementing QKD with regular telecom infrastructure by using quantum encryption. The network guarantees the secure transfer of government and financial data and shows that the QKD can be implemented to provide solutions to real-world problems. Toshiba Europe Ltd. has developed the chip-based Quantum Key Distribution (QKD) system with focus on reducing the size, weight and power consumption of QKD systems, by integrating them into semiconductor chips [32]. These chips are more power-efficient and can be mass-produced with significantly lower cost. These chips with QKD are used to provide a robust level of security for highly-sensitive data.

The proposed scheme is based on QKD approach with blockchain to authenticate IoT devices. The BDAEC-QA scheme considers the aspect security rather than power consumption at the IoT device. Our simulation results show that the proposed scheme performs better in terms of various delays, but also resists different attacks. With research going on, quantum-based solutions, the QKD approach and PQC can be implemented to provide more security with less network resources.

4. Simulation Model

This section describes simulation settings, different performance parameters and different security threats applicable to the BDAEC-QA scheme.

4.1 Simulation Settings

The proposed BDAEC-QA scheme is simulated using the Eclipse platform with Java SDK 11. Three edge servers were created and each was registered with three IoT devices. We set the communication distance at 50 metres. We also used the public blockchain to store the device information. The metamask is used to fetch the information from the blockchain in real time during device authentication.

4.2 Performance Analysis

We have simulated BDAEC-QA scheme and compared it with LTBA [22] and BBMA [25] schemes. Different performance parameters mentioned below are analyzed to test the effectiveness of the proposed scheme.

- **Registration delay:** It is the time taken by the IoT device to register to the edge server. It is measured in milliseconds. We observe that from Figure 3, as more devices register for different edge servers randomly, there is an increase in the registration time. The BDAEC- QA scheme used 3 edge servers and devices can register with any edge server. There is liner growth, which shows that the proposed scheme is stable.
- **Key-generation delay:** It is the time taken by the edge server or the IoT device to generate the QKey and is denoted as T_{qk}. It is measured in milliseconds. The key-generation delay of the BDAEC-QA scheme w.r.t the number of devices is shown in Figure 4. The BDAEC-QA scheme key-generation delay is reduced by 14% and 15% than in LTBA and 10% and 11% than in BBMA, when the edge devices considered are 50 and 100, respectively. The BDAEC-QA scheme generates quantum keys using bases and quantum sequence information rather than complex mathematical computations and hence takes less time.
- **Encryption delay:** It is the time taken by the edge server or the IoT device to encrypt the message and is denoted as T_e. It is measured in milliseconds. Figure 5 shows the encryption delay w.r.t key size and the number of devices. The BDAEC-QA scheme encryption delay is decreased by4% and 4.9% than in LTBA and 2.8% and 5.6% than in BBMA respectively. BDAEC- QA uses the quantum key which is a symmetric key and hence the key-generation delay is lower, resulting in a lower encryption delay.
- **Decryption delay:** It is the time taken by the edge server or the IoT device to decrypt the message and is denoted as T_d. It is measured in milliseconds. The BDAEC-QA scheme decryption delay is 5.2% and 10% less than in LTBA and 2.9% and 5.8% less than in BBMA, respectively. Figure 6 shows the decryption delay w.r.t varying key size and the number of devices. Since the key-generation delay is less because BDAEC-QA uses the quantum key which is a symmetric key, this results in a lower decryption delay.
- **Authentication delay:** It is the time taken by the edge server to authenticate the registered IoT device. It is measured in milliseconds. Figure 7 shows the authentication delay of an IoT-device with key size. We observed that BDAEC-QA scheme takes 16% and 7% less authentication time than LLBA and BBMA schemes, respectively.

- **Storage cost:** It is the number of bits required to store the information at the IoT device and the edge server during the operations discussed in the proposed scheme. It is denoted as S_{cost}.
- **Computation cost:** It is the number of bits required to complete the operations discussed in the proposed scheme by the edge server and the IoT device. It is denoted as C_{cost}.

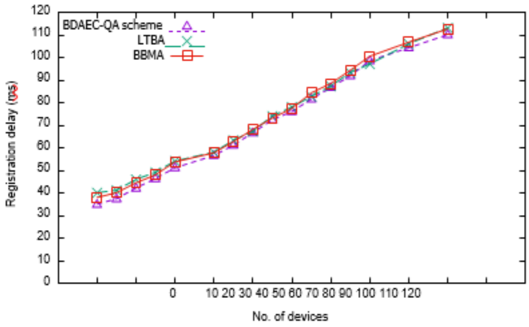


Figure 3. Registration delay.

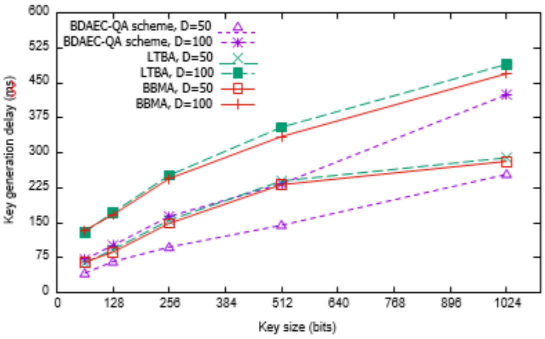


Figure 4. Key-generation delay.

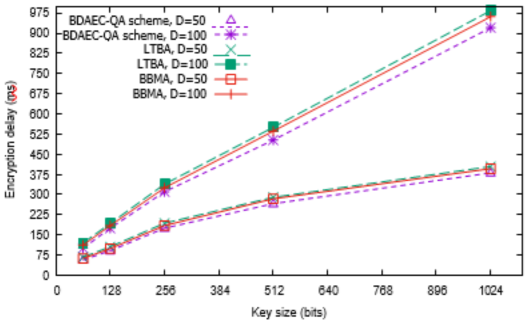


Figure 5. Encryption delay.

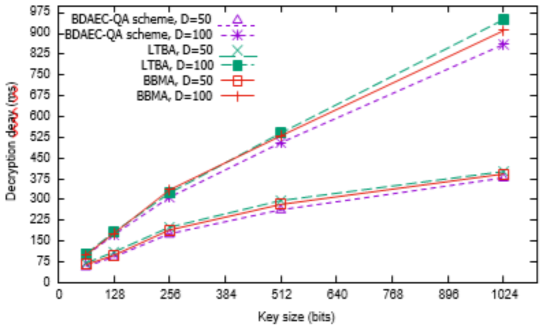


Figure 6. Decryption delay.

4.3 Adversary Model

The Cannetti-Krawczyk (CK) adversary model [33] evaluates the proposed authentication protocol. In the CK model, the adversary is provided with the information about the messages exchanged between authorized parties. The adversary uses the information and impersonates the authorized users. The goal of CK model is to determine the level of security that the protocol should provide and withstand against various attacks. Along with the CK model, additional security requirements are discussed.

- **Usual attacks:** An attacker can steal the information of a device by stealing its identity. If an attacker can impersonate a device, he/she can change the authentication process. The OFMC report provided in Fig. 8 suggest that in the session role, the keys of device and server are made available to the intruder, but still the system is "SAFE" as shown in Fig. 9. In BDAEC- QA scheme, the authentication of IoT devices is carried out by using the context information of the device, QKey, which is stored in the blockchain on the server side. These pieces of information are difficult to steal and hence, the proposed scheme resists to reply and impersonation attacks.
- **Ephemeral Secret Leakage (ESL) attack:** It refers to the preservation of identity of the IoT device privacy. In BDAEC-QA scheme, only registered devices can be authenticated. The confidentiality of the BDAEC-QA scheme is checked by disposing the partial information of the edge device, such as ID, public key and private key. During registration, the context information is passed as Encrpt (k_{pues}, CI$|k_{puit}$). Also, during authentication, the IoT device sends the authentication request as: h (CI $|DID$ $|QKey$), in encrypted format and only IoT devices can manipulate the information. Thus, it guarantees the confidentiality of the proposed BDAEC-QA scheme.
- **Conditional anonymity (CA):** The CI of the IoT device and the private key of the edge server is provided to the attacker, but in the BDAEC-QA scheme, it is not possible to impersonate, since the device information is stored in the blockchain along with its Qkey. The device information is stored in the blockchain with h (CI $|DID$ $|QKey$); hence, in the proposed scheme, the device is not revealed to any server.

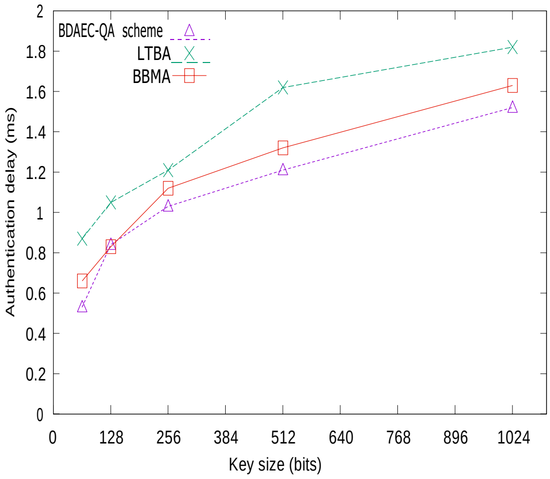


Figure 7. Authentication delay.

- **No Key Escrow:** The proposed BDAEC-QA scheme ensures that the information of the IoT devices is known by storing the hashed information of the IoT devices at the edge server.
- **Integrity:** It refers to the authentication information of an IoT device stored at the edge server, which can not be modified once it is stored unless this done by the IoT device itself. In the BDAEC-QA scheme, once the context information is accepted and the QKey is generated at the edge server, these are stored in the blockchain using a smart contract as h (CI |DID |$QKey$). Once the information is stored, it cannot be modified, thus guaranteeing the integrity of the proposed scheme.
- **Device capture or (Man-in-middle attack):** A device-capture attack happens when an intruder acquires the information of communicating devices and behaves as an authenticated device. It either steals or alters the data as required, which affects the communication between the devices. The proposed BDAEC-QA scheme prevents such attacks by using QKey and blockchain, because the context information is hashed by using QKey: h (CI |DID |$QKey$) and can not be easily compromised. Hence, it entrusts the message only to legitimate devices.
- **Resistance eavesdropping attack:** Information leakage is crucial for any authentication protocol; otherwise, an attacker can deduce the message exchanged between an IoT device and an edge server and extract information. In the BDAEC-QA scheme, the public and private keys are used during the initialization phase and QKey is used after the key-generation phase. In proposed scheme, it is not possible to extract the IoT-device information such as: context information, QKey and timestamp as easily. Even if the intruder tries to extract the quantum bits and basis information, the QKey information cannot be found due to the randomness of the QKD protocol.
- **Blockchain-data transfer:** In the BDAEC-QA scheme, we preserve the device data by storing it in the blockchain at the edge server. The blockchain ensures that data cannot be modified once it is stored, by using the previous block's hash value while creating the structure of the blockchain. Hence, the proposed scheme is more secured.
- **Quantum-attack resistance:** The BDAEC-QA scheme uses QKey which is generated by using the quantum bits and basis at the edge server and the IoT devices. It is not possible to detect the state of the quantum key. If there is any modification to the quantum bits, then a new quantum sequence is generated and hence a different QKey, which is detected by our authentication scheme, thereby preventing quantum attacks.

4.4 Security Analysis

The BDAEC-QA scheme is analyzed with fortifcation against different attacks to ensure that the proposed scheme is well protected. Table 4 provides the security properties comparison between the proposed BDAEC-QA scheme and existing schemes. The proposed work is analyzed using the Automated Validation of Internet Security Protocols and Applications (AVISPA) simulation tool in order to verify the authentication protocol against various attacks, including MIM, impersonation, replay and key secrecy [40]. AVISPA is based on High-level Protocol Specification Language

(HLPSL), which is an expressive, modular, role-based, formal language that allows for specification. HLPSL uses the temporal logic of action for specified semantics, converting the latter into operation semantics as an Intermediate Format (IF) and the output is in Output Format (OF). IF specifications are input into the 4 back-end models: On-the-Fly Model Checker (OFMC), Constraint Logic-based Attack Searcher (CL-AtSe), SAT-based Model Checker (SATMC) and Tree Automata based on automatic approximations for analysis of security protocols (TA4SP) used by AVISPA. Figure 8 shows the roles of device, edge server and session role. Figure 9 shows the OFMC report, which performs protocol falsification and bounded verification, CL-AtSe report applies constraint solving and implements redundancy elimination techniques and simplification heuristics and the SATMC report, which represents a violation of the security properties of the protocol. The OFMC and CL-AtSe models use formal verification and if the result of an authentication protocol is safe, then security requirements are met.

The comparison of storage and computation costs with various schemes is provided in Table 3. The T_{0+1}, T_{c+cs}, T_{4+9} use 128-bit computation cost. T_h generates 128-bit hashed output. T_e and T_d also generate 128-bit encrypted and decrypted outputs. T_b uses 20 bits to store the block information. T_{qk} uses 64 bits as key size. The existing schemes [34]-[35],[37]-[38] and [39] use more C_{cost} and [36] uses less C_{cost} compared with the proposed scheme. The S_{cost} is used by the scheme discussed in [34]-[36] and [39] uses more as compared with the proposed scheme.

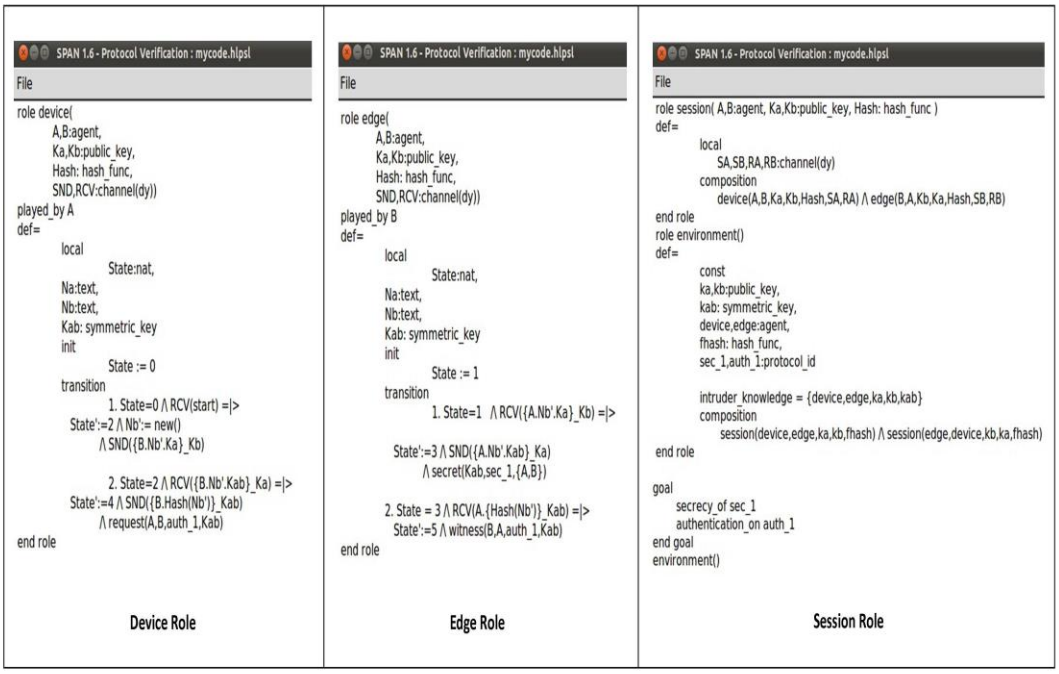
Table 3. Comparison of computation cost and storage cost with existing schemes.

Schemes	C_{cost} **(in bits)**	S_{cost} **(in bits)**
Multimodal biometric [34]	$T_{0+1} + T_{c+cs} + T_{4+9}$=512	$6Mn\phi$=23.3 kb
Remote registration and group authentication [35]	$T_k + 2T_h + 2T_e$=576	$T_k + 2T_h + 2T_d$=576
Lightweight Three-Factor Authentication [36]	$4T_{mp} + 2T_{add} + T_h$=320	$2klogk(4k^2log^2k + 4klogk + 7)$=861
Secure user authentication and key agreement [37]	$7T_h + 2T_{e/d}$=1152	-
Secure authentication key exchange [38]	$26T_h + 11T_{pm}$=4736	-
Light authentication key agreement [39]	$19T_h$=2432	$3T_h + 3T_{fe} + 3T_d + K_{fe}$=1280
BDAEC-QA Scheme	$3T_e + 2T_{qk} + 2T_d + T_b$=468	$2T_e + 2T_{qk} + 2T_d + T_b$=532

Table 4. Comparison of security properties with existing schemes.

Security Properties	**[34]**	**[35]**	**[36]**	**[37]**	**BDAEC-QA scheme**
Confidentiality	✓	✓	✓	✓	✓
Integrity	✓	✓	✓	**NA**	✓
Man-in-middle attack	×	✓	✓	✓	✓
Impersonation attack	✓	✓	✓	✓	✓
Anonymity	×	✓	✓	✓	✓
Eavesdropping attack	×	✓	✓	✓	✓
Blockchain data transfer	×	×	✓	**NA**	✓
Quantum attack	**NA**	**NA**	**NA**	**NA**	✓

The quantum-key approach uses symmetric key between the IoT device and the edge server. The symmetric-key exchange is faster as compared with asymmetric-key exchange. The main objective is to implement the quantum mechanics for generating Qkey and the blockchain approach to enhance the security of the proposed scheme. The Qkey is used to exchange messages between edge device and edge server. The CI and QKey of the device are stored in the blockchain with smart contract to provide the extra layer security at the edge server. The blockchain is an immutable ledger which provides the integrity of data stored in it. The proposed protocol is feasible by not only resisting the major attacks, but also by performing better compared with other schemes. The BDAEC-QA scheme is simulated and compared with existing schemes with respect to various delays, computation cost and storage cost and it performs better. The scheme is also validated with the CK adversary model with different attacks and analyzed using AVISPA tool to meet the security requirements.


SPAN 1.6 - Protocol Verification : mycode.hlpsl
File
SPAN 1.6 - Protocol Verification : mycode.hlpsl
File
SPAN 1.6 - Protocol Verification : mycode.hlpsl
File

```
role device(
        A,B:agent,
        Ka,Kb:public_key,
        Hash: hash_func,
        SND,RCV:channel(dy))
played_by A
def=
          local
                    State:nat,
          Na:text,
          Nb:text,
          Kab: symmetric_key
          init
                    State := 0
          transition
                    1. State=0 /\ RCV(start) =|>
            State':=2 /\ Nb':= new()
                  /\ SND({B.Nb'.Ka}_Kb)

                    2. State=2 /\ RCV({B.Nb'.Kab}_Ka) =|>
            State':=4 /\ SND({B.Hash(Nb')}_Kab)
                  /\ request(A,B,auth_1,Kab)
end role
```

Device Role

```
role edge(
        A,B:agent,
        Ka,Kb:public_key,
        Hash: hash_func,
        SND,RCV:channel(dy))
played_by B
def=
          local
                    State:nat,
          Na:text,
          Nb:text,
          Kab: symmetric_key
          init
                    State := 1
          transition
                    1. State=1   /\ RCV({A.Nb'.Ka}_Kb) =|>

            State':=3 /\ SND({A.Nb'.Kab}_Ka)
                  /\ secret(Kab,sec_1,{A,B})

          2. State = 3 /\ RCV(A.{Hash(Nb')}_Kab) =|>
            State':=5 /\ witness(B,A,auth_1,Kab)
end role
```

Edge Role

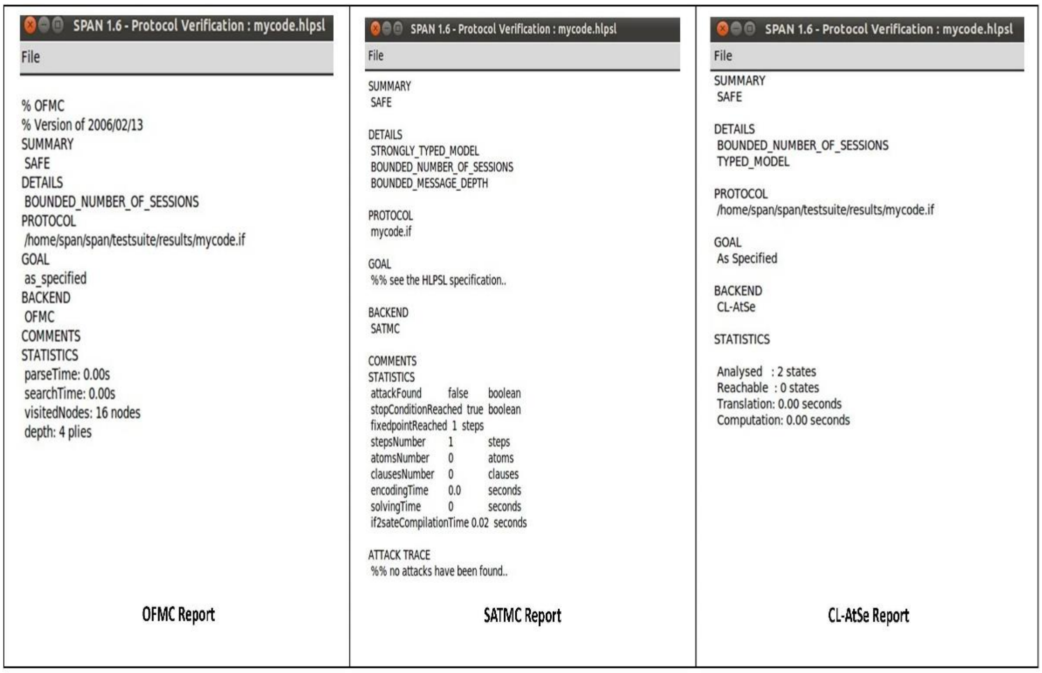
```
role session( A,B:agent, Ka,Kb:public_key, Hash: hash_func )
def=
          local
              SA,SB,RA,RB:channel(dy)
          composition
              device(A,B,Ka,Kb,Hash,SA,RA) /\ edge(B,A,Kb,Ka,Hash,SB,RB)
end role
role environment()
def=
          const
          ka,kb:public_key,
          kab: symmetric_key,
          device,edge:agent,
          fhash: hash_func,
          sec_1,auth_1:protocol_id

          intruder_knowledge = {device,edge,ka,kb,kab}
          composition
              session(device,edge,ka,kb,fhash) /\ session(edge,device,kb,ka,fhash)
end role

goal
     secrecy_of sec_1
     authentication_on auth_1
end goal
environment()
```

Session Role

Figure 8. Device, edge and session roles.


SPAN 1.6 - Protocol Verification : mycode.hlpsl
File
SPAN 1.6 - Protocol Verification : mycode.hlpsl
File
SPAN 1.6 - Protocol Verification : mycode.hlpsl
File

```
% OFMC
% Version of 2006/02/13
SUMMARY
 SAFE
DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
 /home/span/span/testsuite/results/mycode.if
GOAL
 as_specified
BACKEND
 OFMC
COMMENTS
STATISTICS
 parseTime: 0.00s
 searchTime: 0.00s
 visitedNodes: 16 nodes
 depth: 4 plies
```

OFMC Report

```
SUMMARY
 SAFE

DETAILS
 STRONGLY_TYPED_MODEL
 BOUNDED_NUMBER_OF_SESSIONS
 BOUNDED_MESSAGE_DEPTH

PROTOCOL
 mycode.if

GOAL
 %% see the HLPSL specification..

BACKEND
 SATMC

COMMENTS
STATISTICS
 attackFound        false     boolean
 stopConditionReached true  boolean
 fixedpointReached  1  steps
 stepsNumber        1         steps
 atomsNumber        0         atoms
 clausesNumber      0         clauses
 encodingTime       0.0       seconds
 solvingTime        0         seconds
 if2sateCompilationTime 0.02  seconds

ATTACK TRACE
 %% no attacks have been found..
```

SATMC Report

```
SUMMARY
 SAFE

DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
 TYPED_MODEL

PROTOCOL
 /home/span/span/testsuite/results/mycode.if

GOAL
 As Specified

BACKEND
 CL-AtSe

STATISTICS

 Analysed   : 2 states
 Reachable  : 0 states
 Translation: 0.00 seconds
 Computation: 0.00 seconds
```

CL-AtSe Report

Figure 9. AVISPA simulation results.

5. DISCUSSION

The existing schemes use the complex mathematical computations for key generation and hence, they take more time to encrypt and decrypt the data. In time-sensitive application, IoT devices are deployed and need less time to communicate and authenticate themselves. The proposed scheme is compared with LTBA scheme which is based on two-factor authentication which stores biometric details using random oracle model and BBMA scheme, based on blockchain-based mutual authentication between IoT devices and edge server by using different cryptographic algorithms. The BDAEC-QA scheme takes less delay as compared with the LTBA and BBMA schemes with respect to different performance parameters. The lattice-based solutions for device-to-device authentication [29] provide the post-quantum solutions at the cost of communication overhead. The proposed work BDAEC-QA uses less overhead by considering Qkey and only the blockchain is used to store the device information as compared with [29] where a consortium-blockchain network among edge servers is used to maintain a copy of the ledger in each edge server. The membership-service provider is added to manage access level to the ledger, hence the authentication is provided with more overhead cost. The anonymous authentication for vehicular communication [30] takes complex operations and may impact the communication overhead. The paper in [30] discusses the blockchain with smart contract and stores the public keys, which reduces the cost at the edge node. The proposed BDAEC-QA scheme also stores the IoT-device information in the blockchain using smart contract with less communication cost. The storage and computations cost of the proposed scheme are also compared with different existing schemes, as shown in Table 3 and the BDAEC-QA scheme performs better by storing less bits to store and compute the data at the edge server. As compared with LTBA scheme, our scheme uses the blockchain to store the IoT-device information at the server side; hence, it provides more security. The security analysis of the proposed scheme is done using AVISPA as compared with LTBA, BBMA and all the security requirements of the proposed scheme are met as shown in Table 4.

6. Conclusion

The blockchain-based device authentication using quantum approach focuses on authenticating IoT devices within and outside the edge network using quantum-key mechanism. The main objective is to implement the quantum mechanics for generating Qkey and the blockchain approach to enhance the security in case of authenticating the IoT devices. The proposed scheme works in 3 phases: IoT-device registering with the edge server and storing the context information using the quantum key. The Qkey is used to exchange messages between edge device and edge server. The CI and QKey of device are stored in the blockchain with smart contract to provide the extra-layer security at the edge server. The blockchain is an immutable ledger which provides the integrity of data stored in it. The IoT device is authenticated when the device sends the authentication message to the edge server. The proposed work is feasible, not only by resisting the major attacks, but also by performing better compared with other schemes. The BDAEC-QA scheme is simulated and compared with existing schemes with respect to various delays, computation cost and storage cost and it performs better. The scheme is also validated with CK adversary model with different attacks and analyzed using AVISPA tool to check the safety of the proposed scheme to meet the security requirements.

References

[1] A. Kumar et al., "A Comprehensive Survey of Authentication Methods in Internet-of-Things and Its Conjunctions," Journal of Network and Computer Applications, vol. 204, Page 103414, 2022.

[2] W. Z. Khan et al., "Edge Computing: A Survey," Future Generation Computer Systems Journal, vol. 97, pp. 219–235, 2019.

[3] K. Cao, Y. Liu, G. Meng and Q. Sun, "An Overview on Edge Computing Research," IEEE Access Journal, vol. 8, pp. 85714–85728, 2020.

[4] Q. Fan et al., "A Secure and Efficient Authentication and Data Sharing Scheme for Internet of Things Based on Blockchain," Journal of Systems Architecture, vol. 117, Page 102112, 2021.

[5] P. M. Chanal and M. S. Kakkasageri, "Security and Privacy in IoT: A Survey," Wireless Personal Communications Journal, vol. 115, pp. 1667—1693, 2020.

[6] P. Memarmoshrefi, R. Seibel and D. Hogrefe, "Autonomous Ant-based Public Key Authentication Mechanism for Mobile *Ad-hoc* Networks," Journal of Mobile Networks and Applications, vol. 21, pp. 149–160, 2016.

[7] S. S. Rani, S. Pradeep, R. M. Dinesh and S. G. Prabhu, "OTP Based Authentication Model for Autonomous Delivery Systems Using Raspberry Pi," Proc. of the Int. Conf. on Intelligent Controller and Computing for Smart Power (ICICCSP), pp. 1–5, Hyderabad, India, 2022.

[8] M. Mitev et al., "Authenticated Secret Key Generation in Delay-constrained Wireless Systems," EURASIP Journal of Wireless Communication and Networking, vol. 2020, Article no. 122, 2020.

[9] Y. Wang, T. Nakachi and H. Ishihara, "Edge and Cloud-aided Secure Sparse Representation for Face Recognition," Proc. of the 27th IEEE European Signal Processing Conf. (EUSIPCO), pp. 1–5, A Coruna, Spain, 2019.

[10] H. Goumidi et al., "Lightweight Secure Authentication and Key Distribution Scheme for Vehicular Cloud Computing," Journal of Symmetry, vol. 13, no. 3, Article no. 484, pp. 1-29, 2021.

[11] J. Mulholland, M. Mosca and J. Braun, "The Day the Cryptography Dies," IEEE Security Privacy, vol. 15, no. 4, pp. 14–21, 2017.

[12] N. Gisin, G. Ribordy, W. Tittel and H. Zbinden, "Quantum Cryptography," Reviews of Modern Physics, vol. 74, pp. 145–195, 2002.

[13] C. H. Bennett and G. Brassard, "Quantum Cryptography: Public Key Distribution and Coin Tossing," Theoretical Computer Science, vol. 560, pp. 7–11, 2014.

[14] H. Zeyu et al., "Survey on Edge Computing Security," Proc. of the IEEE Int. Conf. on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), pp. 96–105, Fuzhou, China, 2020.

[15] X. Wang et al., "Survey on Blockchain for Internet of Things," Journal of Computer Communications, Elsevier, vol. 136, pp. 10–29, 2019.

[16] M. A. Uddin, A. Stranieri, I. Gondal and V. Balasubramanian, "A Survey on the Adoption of Blockchain in IoT: Challenges and Solutions," Blockchain: Research and Applications, vol. 2, no. 2, pp. 1–49, 2021.

[17] Y. Zhang et al., "Smart Contract-based Access Control for the Internet of Things," IEEE Internet of Things Journal, vol. 6, no. 2, pp. 1594– 1605, 2018.

[18] K. Ekerta, "Quantum Cryptography Bases on Bell's Theorem," Physical Review Letters, vol. 67, pp. 661–664, 1991.

[19] V. Scarani et al., "Security Aspect of Practical Quantum Key Distribution," Reviews of Modern Physics, vol. 81, no. 3, pp. 1301–1350, 2009.

[20] U. Khalid et al., "A Decentralized Lightweight Blockchain-based Authentication Mechanism for IoT Systems," Cluster Computing, vol. 23, pp. 2067–2087, 2020.

[21] K. Hameed, S. Garg, M. B. Amin and B. Kang, "A Formally Verified Blockchain-based Decentralized Authentication Scheme for the Internet of Things," Journal of Supercomputing, vol. 77, pp. 14461—14501, 2021.

[22] A. A. Al-Saggaf, T. Sheltami, H. Alkhzaimi and G. Ahmed, "Lightweight Two-factor-based User Authentication Protocol for IoT-enabled Healthcare Ecosystem in Quantum Computing," Arab Journal for Science and Engineering, vol. 48, pp. 2347–2357, 2023.

[23] H. Abulkasim et al., "Authenticated Secure Quantum-based Communication Scheme in Internet-of-Drones Deployment," IEEE Access Journal, vol. 10, pp. 94963–94972, 2022.

[24] J. Wu, Z. Jin, G. Li, Z. Xu, C. Fan and Y. Zheng, "Design of Vehicle Certification Schemes in IoV Based on Blockchain," World Wide Web Journal, vol. 25, pp. 2241—2263, 2022.

[25] G. Cheng, Y. Chen, S. Deng, H. Gao and J. Yin, "A Blockchain-based Mutual Authentication Scheme for Collaborative Edge Computing," IEEE Transactions on Computational Social Systems, vol. 9, no. 1, pp. 146–158, 2022.

[26] A. K. Sahu, S. Sharma and D. Puthal, "Lightweight Multi-party Authentication and Key Agreement Protocol in IoT-based E-Healthcare Service," ACM Transactions on Multimedia Computing, Communications and Applications, vol. 17, pp. 1–20, 2021.

[27] P. Nag, P. Chandrakar and K. Chandrakar, "An Improved Two-factor Authentication Scheme for Healthcare System," Procedia Computer Science, vol. 218, pp. 1079–1090, 2023.

[28] S. Sathyadevan, K. Achuthan, R. Doss and L. Pan, "Protean Authentication Scheme – A Time-bound Dynamic KeyGen Authentication Technique for IoT Edge Nodes in Outdoor Deployments," IEEE Access Journal, vol. 7, pp. 92419–92435, 2019.

[29] A. Shahidinejad and J. Abawajy, "Decentralized Lattice-based Device-to-device Authentication for the Edge-enabled IoT," IEEE Systems Journal, vol. 17, no. 4, pp. 6623–6633, 2023.

[30] A. Shahidinejad, J. Abawajy and S. Huda, "Anonymous Lattice-based Authentication Protocol for Vehicular Communications," Vehicular Communications, vol. 48, Page 100803, 2024.

[31] D. Stucki et al., "Long-term Performance of the SwissQuantum Quantum Key Distribution Network in a Field Environment," New Journal of Physics, vol. 13, no. 12, Page 123001, 2011.

[32] Toshiba, "QKD Technology to Semiconductor Chip," [Online], Available: https://news.toshiba.com/press-releases/press-release-details/2021/Toshiba-Shrinks-Quantum-Key-Distribution-Technology-to-a-Semiconductor-Chip/default.aspx.

[33] R. Canetti and H. Krawczyk, "Analysis of Key-exchange Protocols and Their Use for Building Secure Channels," Proc. of the International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2001), pp. 453–474, Springer, 2001.

[34] N. D. Sarier, "Multimodal Biometric Authentication for Mobile Edge Computing," Information Sciences, vol. 573, pp. 82–99, 2021.

[35] H. Goswami and H. Choudhury, "Remote Registration and Group Authentication of IoT Devices in 5G Cellular Network," Computers Security Journal, vol. 120, Page 102806, 2022.

[36] A. M. Almuhaideb and K. S. Alqudaihi, "A Lightweight Three-factor Authentication Scheme for WHSN Architecture," Sensors Journal, vol. 20, no. 23, Page 6860, 2020.

[37] S. Uppuluri and G. Lakshmeeswari, "Secure User Authentication and Key Agreement Scheme for IoT Device Access Control Based Smart Home Communications," Wireless Network Journal, vol. 29, pp. 1333—1354, 2023.

[38] Wu, Tsu-Yang, Zhiyuan Lee, Lei Yang, Jia-Ning Luo and Raylin Tso, "Provably Secure Authentication Key Exchange Scheme Using Fog Nodes in Vehicular *Ad Hoc* Networks," The Journal of Supercomputing vol. 77, no. 7, pp. 6992-7020, 2021.

[39] M. Hamada, S. A. Salem and F. M. Salem, "LAMAS: Lightweight Anonymous Mutual Authentication Scheme for Securing Fog Computing Environments," Ain Shams Engineering Journal, vol. 13, no. 6, p. 101752, 2022.

[40] I. Aciobanitei, R. I. Guinea and M. L. Pura, "AVISPA *versus* AVANTSSAR in the Model Checking of Secure Communication Protocols," Proc. of the 15th Int. Joint Conf. on e-Business and Telecomm. (ICETE 2018), vol. 2: SECRYPT, pp. 520-525, DOI: 10.5220/0006887905200525, 2018.

ملخص البحث:

ظهرت إنترنت الأشياء كتكنولوجيا جديدة، حيث كلُّ شيء متّصل. وينبغي تخزين كمّياتٍ ضخمة من البيانات، لذا فإنّ حوسبة الحافة يمكنها التّقليل من تخزين البيانات في بيئةٍ موزَّعة، الأمر الّذي من شأنه أن يحسن من سرعة المعالجة ويؤدّي إلى استخدام قدْر أقلّ من عرض النّطاق. ومع ازدياد استخدام أجهزة إنترنت الأشياء، ازداد ظهور قضايا مثل المُصادقة على الأجهزة، وخصوصية البيانات المخزّنة، وتكامل البيانات. وتُعدّ المُصادقة على الأجهزة مسألةً مهمّةً بالنّسبة إلى أجهزة إنترنت الأشياء المتّصلة باستخدام حوسبة الحافَة. وقد تمّت معالجة هذه المسألة عن طريق خوارزميات ترميزٍ كلاسيكية لترميز الرّسائل باستخدام المفاتيح العامّة والمفاتيح الخاصّة الّتي يتعيّن تخزينها. ويتطلب الأمر تخزين هذه المفاتيح في جهاز خادم من أجل المُصادقة على الأجهزة، علماً بأنّ تخزين عددٍ كبيرٍ من المفاتيح يؤدّي إلى ازدياد تكلفة الحوسبة وتكلفة التّخزين وإلى ازديادٍ في التّأخير. وباستخدام الحوسبة الكمّيّة والخوارزميات الكمّيّة، يُصبح من السّهل كَسْرُ خوارزميات التّرميز، الأمر الّذي يجهل النّظام هشّاً.

إنّ الآلية المقترحة في هذا البحث من شأنها أن توفِّر المُصادقة على أجهزة إنترنت الأشياء باستخدام معلومات السِّياق وتوزيع المفاتيح الكمّيّة وسلاسل الكُتل. وتستخدم الطّريقة المقترحة "العُقود الذّكية" لتخزين المعلومات الخاصّة بأجهزة إنترنت الأشياء في جهاز الخادم الّذي تستخدمه سلسلة الكُتل من أجل تأمين المُصادقة المتبادلة بين أجهزة إنترنت الأشياء عبر الشّبكة.

لقد جرت مقارنة الطّريقة المقترحة بعددٍ من الطُّرق المشابهة الواردة في أدبيات الموضوع، وامتازت الطّريقة المقترحة بقيم أقلّ لكلٍّ من زمن التّسجيل وزمن توليد المفاتيح وزمن تأخير المُصادقة. كذلك تميّزت الطّريقة المقترحة بقدْرٍ أقلّ لتكلفة الحوسبة وتكلفة التّخزين، مقارنةً بغيرها من الطُّرق الواردة في دراسات سابقة أخرى. وقد تمّت محاكاة الطّريقة المقترحة باستخدام أداة (AVISPA) للبرهنة على أمانها ومُقاومتها للهجمات.