CHAPTER 6

Development of a Blockchain Token-based Authentication of users in SDN Network

6.1 Introduction

Software-defined networking (SDN) can help in IoT networks by providing a centralized control plane that can dynamically manage and optimize network resources [21]. This allows for more efficient and automated network management, which is particularly useful in IoT networks where there are huge numbers of devices that need to be connected and managed. However, with the increasing number of devices, there is a need to protect them from unauthorized access. Authentication plays a crucial role in securing IoT networks by ensuring that only authorized users and systems can access and control the devices connected to the network. Many of the authentication protocols used to protect IoT devices are not up to industry standards, leaving them vulnerable to breaches [70]. Existing mechanisms such as two-factor authentication, and password management protocols may not be implemented on all devices. Furthermore, the centralized architecture, in which all data and resources are controlled by a single entity, can be a drawback in the modern world because it can lead to a lack of security, scalability, and decentralization. Blockchain Technology can be utilized to enhance the security and efficiency of distributed authentication by providing a decentralized and tamper-proof method for storing and sharing authentication information [40]. Additionally, the use of a consensus algorithm ensures that the stored authentication information is consistent and accurate.

The rest of this chapter is structured as follows: In Section 6.2, we describe the system architecture of token-based authentication using Smart Contract. Section 6.3 presents the experimental results and performance of the proposed work. We conclude the proposed work in section 6.4.

6.2 Proposed Model

The proposed model is a decentralized authentication mechanism for the SDN-IoT network that utilizes the Blockchain SC to provide access to IoT users. By verifying both the device and user identities, the authentication process helps prevent fraudulent activities and ensures data integrity. The process is automated through a smart contract that issues a token for each user request. The SC contains a Token Manager that plays a crucial role in maintaining secure and efficient user authentication by managing the lifecycle of access tokens. Upon successful authentication, token managers issue and sign access tokens, which are used to authenticate subsequent requests. They validate the incoming token to ensure authenticity and authorization. Token Managers can also revoke and blacklist tokens if necessary. In the following subsection, we discuss each component and interaction among themselves in detail.

6.2.1 System Architecture

The proposed architecture is depicted in Figure 6.1. Our proposed model mainly consists of two modules: the SDN module and the Blockchain module. The SDN module comprises Admins, End Users, Blockchain-enabled Controllers, Switches, and IoT devices. On the other hand, the Blockchain module comprises a User/Device Manager, Digital Token Manager, Data Storage Manager, and Consensus Algorithm. The Blockchain module components are encapsulated together in the SC. In the following, we summarize each component and its role in the system.

a) **Admins:** Network Admins are responsible for the configuration of network devices and monitoring the performances. Additionally, they also perform the enrollment of IoT devices on the blockchain. The Admin deploys the SC on the Blockchain network.

b) **Users:** In the system, End Users are the utilizers of IoT devices. The End User needs to be authenticated before accessing the device.

c) **Controllers:** The proposed model consists of SDN controllers which are distributed but logically centralized and manage the IoT network from a central point. They manage the basic configuration of switches and IoT devices. These SDN controllers are equipped with key pairs (Ethereum Address) to make transactions on the blockchain. They participate in the consensus process for creating blocks during user authentication.

d) **Switches:** These are OpenFlow-enabled switches that forward the IoT traffic to its target destination as per the flow rules installed on the device.

e) **IoT Devices:** Each IoT device connected to the SDN switch is stored on the device to switch mapping of the SC.

f) **Token Manager:** It is part of an SC. Token Manager plays a crucial role in maintaining secure and efficient user authentication by managing the lifecycle of access tokens. Upon successful authentication, token managers issue and sign access tokens, which are used to authenticate subsequent requests. They validate the incoming token to ensure authenticity and authorization. Token Managers can also revoke and blacklist tokens if necessary.

g) **Device Manager:** The device manager monitors and manages the connected devices on the network. The blockchain registers the device information on the blockchain to preserve the identity of the devices.

h) **Storage:** The user/device information and issued tokens are stored on the on-chain storage within the blockchain network. This data can be accessed by nodes on the network. We use on-chain storage because it is highly secure and resistant to tampering, but it can also be limited in terms of storage capacity and may become more expensive as the blockchain grows. Therefore, if the system requires more storage the blockchain also supports offchain storage such as IPFS (InterPlanetary File System).

i) **Consensus:** The proposed work utilize the PoW (Proof-of-Work) consensus protocol to agree upon the authentication decision.

6.2.2 Blockchain Token Based User Authentication

The authentication process is divided into four separate phases: The device registration phase, the user authentication phase, the Token Distribution Phase, and the Validation Phase.

a) **Registration Phase:** The network admin registers the IoT devices on the blockchain through the SDN controller. The global view of the SDN controller enables the network admin to monitor each device connected to the network. The SDN controller informs the network Admin about the IoT device to register on the Blockchain through $RegDev(Dev_{id}, S_{id})$ where Dev_{id} and S_{id} are the device ID and Switch ID respectively.

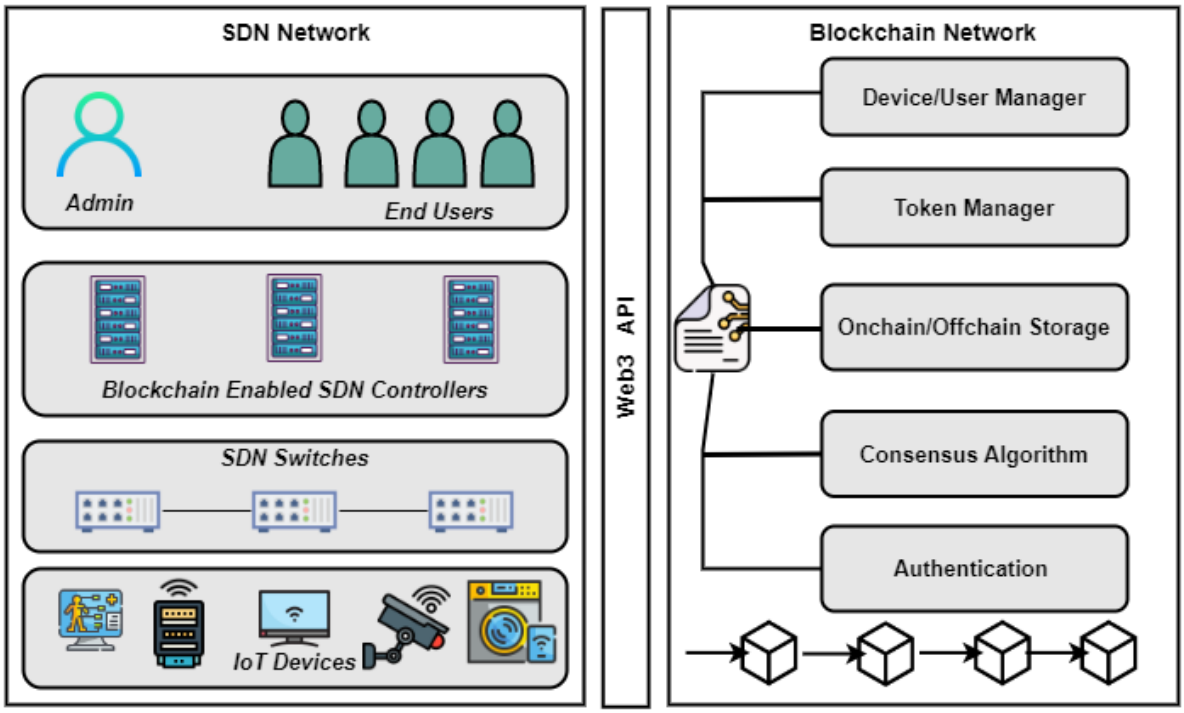

Figure 6.1: Proposed System Architecture.

b) **Authentication Phase:** The authentication phase is divided into three sub-phases: Request, Processing, and Decision. The user makes an authentication request using the public key(Ethereum Address) on the SC. First, the request is converted into a blockchain transaction and is sent through Web3 API. The request message is of the following form:

$$AuthReq = [src\_ip + Pub\_key(EA) + S_{id})] \tag{6.1}$$

where src_ip, $Pub\_key(EA)$ are the user's IP address and Public key (Ethereum Address) respectively and S_{id} is the corresponding Switch ID of the IoT device. The SC receives the request and checks with the User/Device manager if the requested IoT device is registered on the Blockchain. If the device is registered the SC invokes the *userAuth()* function and a token is issued against the user. Otherwise, the user is denied access to the IoT device. The distributed SDN controllers run the consensus to create the block.

c) **Token Distribution Phase:** After a successful User Authentication, a Blockchain Token is generated for the subsequent communication and is sent back to the end user. A Token

is of the following form:

$$Token = [T_{id} + U_{id} + C_t + Issuer_{key}] \tag{6.2}$$

where T_{id} is the token ID, U_{id} is the user ID, C_t is the timestamp of token creation and $Issuer_{key}$ is the Ethereum Address of the token issuer. Once a token is created, a new block is created and broadcasted to all the peers to preserve consistent information about the token. The Token Manager maintains a copy of the token issued for each user and their corresponding issuer which is used later in the validation process.

d) **Token Validation Phase:** Tokens are easily stolen or duplicated, resulting in unauthorized access to sensitive information. Therefore, to prevent fraud and maintain trust between users and IoT devices, authenticated tokens must be validated before accessing the IoT device. The user signs the token received from (6.2) with his/her private key $Sign(Token, Pri\_key)$ and sends the access request to the blockchain. The SC verifies the signature by decoding the token into a specified format to ensure that the token has not been tampered with. The token manager performs the revocation of access tokens by checking the token's lifetime and making sure it has not expired and is still valid. To prevent spoofing attacks, the SC verifies the address of the token issuer.

The timeline of interaction among different components for user authentication is depicted in Figure 6.2. We summarize the steps involved in the user authentication process below:

a) SC registers the IoT device information on the blockchain through the $RegDev(Dev_{id}, S_{id})$ function. Once the device is registered, an event is generated and notified to the admin.

b) Now the End User can invoke the $AuthReq()$ function to generate the digital token.

c) SC executes the UserAuth() function where it checks the validity of the IoT device and performs consensus among the Blockchain nodes.

d) Token Manager stores the token on the blockchain and a copy is sent back to the end user for device access.

e) After the SSL connection is established, the user requests data request using the signed token.

f) Then the Token Manager validates the token by executing *validateToken()* function. On successful validation, the device access is granted.

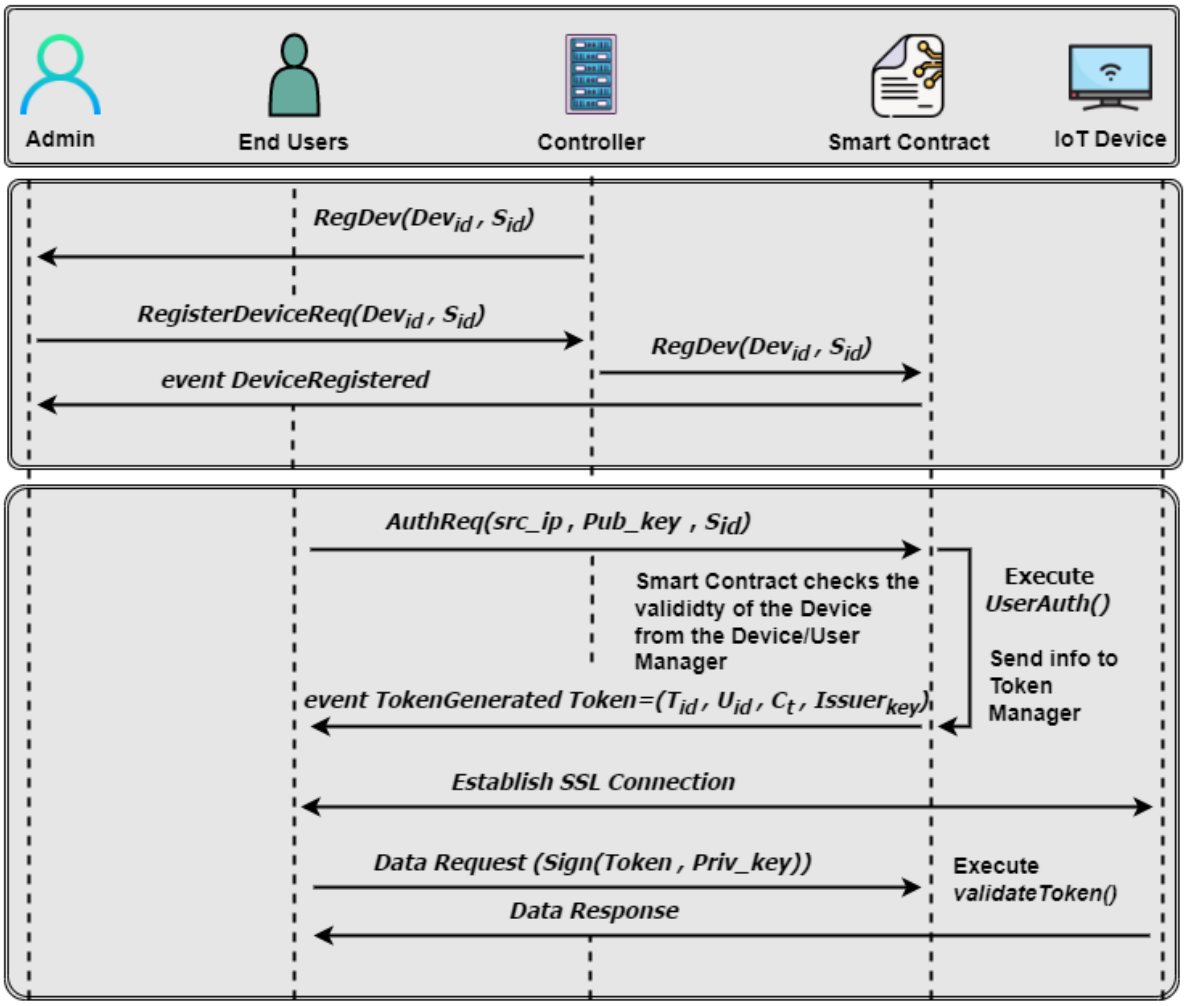

Figure 6.2: Sequence diagram of message passing among different components.

6.3 Results

In this section, we discuss the experimental result analysis of the proposed work. For this experiment, we used an online SC testing tool called Remix IDE. We implemented four functions and events associated with them in the Solidity programming language (version 0.4.22 $<$ 0.9.0). Remix IDE allows the execution of SC code and can also interact with a contract on a

blockchain network. As discussed in section 6.2.2, the token and device structures are defined in the SC.

A proof-of-concept SC is implemented to show that the blockchain token can authenticate IoT users. The token manager issues a token on successful authentication, which is stored on the blockchain. A successful SC execution of the proposed work is shown in Figure 6.3. The authentication performance is evaluated in terms of authentication success rate, latency, security, and gas cost of the authentication process. When *UserAuth()* function is executed, the SC performs the verification of user identity and IoT device mapping with a switch to confirm that the request is legitimate. The token validation shows that an unauthorized user cannot access the device. The proposed work is tested against spoofing attacks, where the user spoofs the IP address of a legitimate user, as shown in Figure 6.4. The SC can identify the authentic user by looking at the token format. The system can successfully resist the attack scenarios tested, demonstrating its high level of security.

6.3.1 Latency Analysis

The authentication method should provide a high level of security to protect against unauthorized access with minimum latency in an IoT network. Therefore, we experimented to measure the time required to execute each SC function, and the results are presented in Table 6.1.

The results of the experiment showed that the average authentication time was **69** milliseconds.

6.3.2 Cost Analysis

Since blockchain is incorporated for the validation of user identity, the transaction cost incurred during the SC function execution is another factor that needs to be analyzed. To conduct this experiment, we record the transaction cost on Remix by setting the gas price to 21000*gwei*. The result of the average transaction cost is presented in Table 6.1. The result shows that the transaction cost is feasible in a real scenario.

```
logs[
    {
        "from": "0x5A86858aA3b595FD6663c2296741eF4cd8BC4d01",
        "topic": "0xbbf285ab57a06f66efef76aacce8608661908779f200e8087ac79f32c1750421",
        "event": "Authenticated",
        "args": {
            "0": "0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2",
            "1": "0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db",
            "2": "0x78731D3Ca6b7E34aC0F824c42a7cC18A495cabaB",
            "user": "0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2",
            "device": "0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db",
            "OFswitch": "0x78731D3Ca6b7E34aC0F824c42a7cC18A495cabaB"
        }
    },
    {
        "from": "0x5A86858aA3b595FD6663c2296741eF4cd8BC4d01",
        "topic": "0xb7afdf41bfed86d8060ccda949f1dce0dc7f85757ca7565c7ed5c7c276f58c59",
        "event": "TokenCreated",
        "args": {
            "0": "1",
            "1": "0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2",
            "2": "0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db",
            "3": "0x78731D3Ca6b7E34aC0F824c42a7cC18A495cabaB",
            "uid": "1",
            "user": "0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2",
            "device": "0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db",
            "OFswitch": "0x78731D3Ca6b7E34aC0F824c42a7cC18A495cabaB"
        }
    }
]
```


Figure 6.3: Successful UserAuth() function execution result.


```
logs    [
    {
        "from": "0xd9145CCE52D386f254917e481eB44e9943F39138",
        "topic": "0xedc2b9184e4f83d4cc9079ff886f15cd4ebc064df34bb7414aa697c2d5812a57",
        "event": "NotAuthenticated",
        "args": {
            "0": "0x5B38Da6a701c568545dCfcB03FcB875f56beddC4",
            "user": "0x5B38Da6a701c568545dCfcB03FcB875f56beddC4"
        }
    }
]
```


Figure 6.4: Failed UserAuth() function execution result.

6.3.3 Discussion

The experiment results demonstrate that IoT token-based authentication is a highly effective and secure access control method. Since tokens are stored on the blockchain, they are resistant

Table 6.1: Average Latency and Transaction cost for the execution of Smart Contract functions

Function	**Latency (second)**	**Transaction Cost (Gas)**
RegDev()	0.18	110254
UserAuth()	0.67	229714
ValidateToken()	0.16	176645
RevokeToken()	0.12	167854

to tampering and can be used to track and audit all interactions with the system. The fast authentication time and high success rate make it a suitable alternative to traditional authentication methods.

Additionally, the ability to resist attack scenarios highlights the high level of security provided by the IoT token. Furthermore, the gas cost required for running the consensus protocol is significantly lower than the cost of using a centralized service, as this serves as a means of mitigating spam and ensuring the security of the network.

When a device needs to authenticate, it can query any node on the network to verify its identity, reducing the cost associated with maintaining a centralized authentication infrastructure. The use of SC in the blockchain enables automatic authentication and authorization of devices, reducing the need for human intervention.

While blockchain-based authentication has the potential to provide increased security and decentralization, it also has some disadvantages such as complexity, scalability issues, regulatory challenges, and interoperability issues that should be considered. The current scalability limitations of blockchain technology can impact its performance and efficiency, especially in large-scale deployment scenarios.

6.4 Conclusion

In this work, we have proposed a Blockchain Assisted solution for the authentication of users in the SDN-IoT network. A digital token is generated through SC to provide access to IoT devices. The decentralized nature guarantees the validity and integrity of the token transmitted

through the network. Further, this eliminates the need for a central authority to manage and secure user data. We showed that the proposed mechanism can validate the token before granting access to the device through Remix IDE execution. The experiment showed that IoT token-based authentication is a feasible and effective method of secure access control. Its high success rate, fast authentication time, and resistance to attack scenarios make it a promising solution for authentication. Additionally, the revocation of access tokens provides a higher level of security. Although our proposed method shows promising results in the Remix IDE, we acknowledge that no comparison with other methods has been conducted in this work.