Enhanced MQTT Architecture for Smart Supply Chain

Raouya AKNIN, Youssef Bentaleb
Ibn Tofail University, Engineering Sciences Laboratory, Kenitra, Morocco

***Abstract*—In industry 4.0, the use of smart supply chains has become necessary in order to overcome the shortcomings of traditional supply chains, such as overstocking, delivery delays, and stock out. However, the use of smart supply chains has introduced new security challenges because of the internet of things (IOT) constraint nature. Thus, the problem raised is ensuring the supply chain security requirements while taking into consideration the properties of the constraint environment. For this purpose, this paper aims to strengthen the authentication and data transmitting processes of the Message Queuing Telemetry Transport(MQTT) protocol, as the most used protocol for communication in the IOT environment, using blockchain and smart contracts. The new MQTT architecture allows to avoid a single point of failure, to ensure data immutability and to automatize the authentication mechanism as well as the publishing and the subscribing processes. In addition, the use of a one-time password (OTP) instead of a permanent one is another security measure used to protect the architecture from identity spoofing. The new architecture comprises three phases: Registration, Connection, and Publishing. Each phase is automatically controlled by a smart contract. For attack simulation tests, the smart contracts are implemented in a remix environment. The results of the simulation tests show that the new architecture is robust and resistant to different attacks.**

Keywords*—Smart supply chain; internet of things; MQTT protocol; blockchain; smart contracts; Ethereum; solidity; one-time password

I. Introduction

Industry 4.0 has proven its effectiveness in managing and optimizing the value chain. It thus enables better production with minimal cost and increased accuracy. The use of various technologies such as internet of things (IOT), cloud computing, artificial intelligence, and big data in order to interconnect different production and control units, collect data and analyze it has offered better visibility, quick decision-making, and intervention in the best time frame. The successful transformation of industries towards 4.0 requires first a review of supply chain management as a crucial element in managing the relationships between the different stakeholders involved in the product life cycle from raw materials acquisition until the delivery of the final product to the customer. Indeed, traditional supply chains have created several challenges, such as overstocking, delivery delays, and stockouts. Although the integration of IOT in supply chain management has solved the above challenges, security issues have arisen and have become more and more worrisome. Indeed, smart supply chains are subjects of several attacks that threaten data confidentiality, integrity, and availability due to security policies that have been breached to meet the constrained environment's requirements. Research works addressing the supply chain security challenges have either used the IOT protocols' native security mechanisms or have proposed a robust solution without considering the constrained environment nature. Hence, the challenge is to fit the supply chains security requirements while taking into consideration the constrained environment properties. This paper will propose a security enhancement of the Message Queuing Telemetry Transport (MQTT) protocol, one of the most used protocol in the IOT environment, thanks to its lightness in terms of resources and bandwidth, without affecting the protocol's overall operation and its performances . The new architecture is based on smart contracts and consortium blockchain in order to automatize access control, publishing, and subscribing processes of devices involved in different stages of the product life cycle as well as to ensure the security of shared data between the supply chain's stakeholders. It is composed of three phases: Registration, Connecting, and Publishing phases. Each phase is controlled by a smart contract.In the registration phase, stakeholders' trusted administrators register the devices in the blockchain, define the topics that they have the rights to publish and subscribe on and recover the necessary keys for connection and transmitting data. Before, publishing or subscribing, the devices must first be connected to the broker network using one-time password (OTP) for authentication. Once the authentication data are approved, they can publish or register in a topic according to the rights granted by their administrators. To further strengthen the architecture, this paper will propose some security measures to protect the supply chain from different attacks. Indeed, it uses a broker network instead of one to avoid a denial of service. It uses the OTP for connection to the broker network in order to protect the device from identity spoofing. It also automatically verifies the packet number before any transaction to protect the architecture from reply attacks. The simulation tests attack will show that the new architecture is resilient to these attacks. The remainder of this paper is structured as follows: Section II will discuss research works that address the smart supply chains security challenges. Section III will report the main smart supply chain management challenges. Section IV will review some preliminary concepts needed for the proposal. Section V will describe the proposed MQTT architecture for the supply chain. Section VI will implement the smart contracts used for the registration, the connection, and the publishing phases using remix IDE. Section VII will show the results of attack simulation tests. Section VIII will describe the contribution of this paper in addressing smart supply chains challenges. Section IX will summarize the main ideas discussed in this article.

II. Related Works

In this section, we will introduce research works addressing the security challenges in supply chain management.

Article[23] proposed an architecture and an implementation of automated supply chain management for position and shipment tracking using global positioning system (GPS) and Radio Frequency Identification (RFID) technologies respectively. The proposed architecture has used MQTT protocol for communication, however, it used native security mechanisms which are not suitable for transmitting a critical data. Article[22] proposed a conceptual framework for supply chain management using blockchain and smart contracts in order to reduce the involvement of third parties and improve data security. However, it focused only on product purchasing and the agreement between buyer and seller while the supply chain management requires the involvement of different stakeholders participating in the product life cycle. Article[20] proposed an implementation of a food tracing system based on permissioned blockchain for the food supply chain. The security supply chain requirements regarding data privacy, confidentiality and access control are fully respected. However,because the supply chain is composed of different stockholders, it will be preferable to use a consortium blockchain instead of a permissioned blockchain. Moreover, the solution used a blockchain in constrained environment which can cause problems of resources and bandwidth. Article [21] applies Cyber Threat Intelligence with Machine Learning techniques to analyze and predict threats against supply chains. However, it focused only on analyzing and predicting the threats without offering the necessary countermeasures. Hence,the proposed solution will be efficient for an already secured architecture. After analyzing the aforementioned research works, it turns out that the papers either use the IOT protocols native security mechanisms for supply chain management which does not fit the security requirements of supply chains, or propose robust security solutions without considering the constrained environment properties. The architecture proposed in this paper aims to improve the security of the MQTT protocol in order to fit the security supply chain's requirements without affecting the overall operations neither the lightness of the protocol.The solution is based on consortium blockchain to allow the interaction between different stakeholders and smart contracts to automatize the authentication, the publication, and the subscription processes as well as to ensure data confidentiality, integrity, and availability. The advantage of this architecture is that only the broker's network interacts with the blockchain and the smart contracts which eliminate the use of the blockchain in the constrained environment. To avoid using the Transport Layer Security (TLS) protocol in this environment, the parameters for calculating the OTP are exchanged on different channels, making it impossible for a hacker to retrieve it. Verification of the packet numbers by smart contracts is another security measure used to avoid replay attacks.

III. Smart Supply Chain Issue

The goal behind supply chain management is to supervise the product life cycle from the purchase of the raw material until the product's deliverance to a customer. The supply chain is composed of many independent entities that must share the data with each other. The use of IOT devices in the supply chain management such as sensors, Radio Frequency Identification (RFID), and Global Positioning System (GPS) improves the performances as well as the transparency in the whole process product life cycle. The gathered data in each step must be shared with other entities. Hence, the MQTT protocol is the fittest protocol thanks to its publish-subscribe model. Indeed, the IOT components can share the data in a specific topic and the subscribers in this topic can receive the data thanks to the intermediate server called a broker. However, the native security measures are insufficient to ensure the security supply chain requirements, and it can make it a subject of many attacks, namely Man in the middle attack, Denial of service, reply attacks, identity spoofing, Information Disclosure, Privilege Escalation, Tampering Data, and so on [15]. In fact, MQTT Messages are not natively encrypted and the data is exchanged in plaintext. Hence any network sniffer can acquire valuable information such as : IP broker, credentials, Name of a topic, Data payload, MQTT port number, and so on. By having this information, several attacks can be carried out. First, a hacker can steal the credentials during the connection establishment phase and then publishes the wrong data on behalf of the legitimate publisher. Moreover, since the credentials are permanent, the hacker can use them forever until the client changes them. Data integrity is also a challenge for the current MQTT architecture since the hacker can modify the MQTT messages content including the topic name that the legitimate publisher had published in. Another attack scenario targeting data integrity can happen when a hacker sends malicious firmware to subscribers in order to transform them into botnets [16] [17] [18]. The proposed solution for this challenge is the use of the Transport Layer Security (TLS) protocol. However, it doesn't seem to be a good alternative since it increases the computational overhead on resource-constrained devices. Another attack type that MQTT is facing is a Denial of service (DoS). Since the Broker presents a single point of failure in MQTT architecture, it can be a target of many DOS attacks. Hence, when a broker is broken down, communication between publisher and subscriber is no longer possible. The Slow DoS against Internet of Things Environments (SlowITe) attack is one among other DOS attacks that target the MQTT protocol. Indeed, it tries to saturate the broker with the maximum possible connections in order to deprive the legitimate clients to connect to the broker [19]. In order to overcome these challenges,the proposed architecture will use a blockchain and smart contract in order to automatize and strengthen the authentication process as well as to ensure data confidentiality and integrity.It will base on a consortium blockchain since the supply chain entities are independent . Indeed, each entity which composed the supply chain has a trusted node called device administrator who registers the IOT devices in the blockchain as well as the topics that has the right to publish or subscribe on a.For instance, the transportation company register the GPS device in the blockchain in order to publish in the Geolocalisation topic. The entities concerned by product tracking such as the customer, the Retailer and the manufactory, are registered in the blockchain as subscribers on this topic.

IV. Preliminaries

This section is a reminder of the main concepts and the technologies used in this proposal.

A. Supply Chain

Supply Chain can be defined as a connected set of resources and processes that are involved in providing goods to customers. It establishes a multistakeholder collaboration environment between different entities that are involved in the product life cycle(manufacturers, suppliers, distributors, retailers and transportation, information and other logistics management service providers) from the raw materials sourcing until the deliverance of finished goods to the end consumer [1][2]. In industry 4.0, the term "smart supply chain" or "digital supply chain" is used to describe the adoption of innovative technologies across all supply chain stages in order to increase the performances and improve customer service [3][4].

B. MQTT Protocol

Message Queuing Telemetry Transport (MQTT) is a publish-subscribe protocol used as an alternative to Hypertext Transfer Protocol (HTTP) in the constraint environment. As depicted in Fig. 1, The MQTT architecture is composed of three components: the publisher that sends the data related to a specific topic, the subscriber that is registered in a specific topic in order to receive a notification when this topic is updated and the broker is an intermediary server that gets the data from different publishers and sends them to subscribers that are already registered in this topic.MQTT protocol used a Transmission Control Protocol (TCP) as an underlying transport protocol [5] [6]. It uses port 1883 for unencrypted messages and port 8883 for encrypted ones. Before publishing or subscribing to any topic, the clients must be connected to a broker. The native authentication method used by the MQTT protocol is the login and password transited in plaintext format [6]. MQTT protocol offers to publishers and subscribers a feature of choosing quality of service (QOS) levels depending on the network condition,the device characteristics and the application criticality: in QOS 0, the message is delivered at most once without any acknowledgment; in QOS1, the message is delivered at least one time. Hence, the sender keeps the message stored until the reception of an acknowledgment; in QOS 2, the message is delivered exactly one time without any duplication[7].

C. Blockchain

Blockhain is a distributed database composed of a list of ordered blocks and it runs on a Peer to peer network [9]. Each block is structured as follows: A header that contains information related to the block namely Block version, Timestamp, Merkle tree root hash, Parent Block hash and nonce, which is a random number for verifying the hash.This information varies based on the blockhain network provider. A body that contains transactions. Each block is linked to the previous one thanks to the Parent block hash field which makes the blockchain immutable from frauds. Special nodes in the network, called Miners, hold the responsibility of adding a block to a blockhain. For this purpose, a consensus algorithm is used in order to reach a common agreement between untrusted nodes and define the winner miner. Proof of work and Proof of stake are the most used consensus algorithms[9] [6].

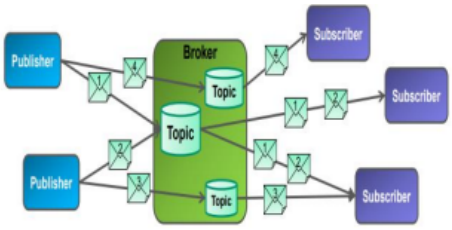

Fig. 1. The MQTT architecture [8].

D. Smart Contract

Smart contract was first introduced by Szabo in the mid-1990s in order to minimize contracting cost between transacting parties and to avoid accidental exceptions or malicious actions during contract performance.He suggested translating a contract into code that will be self-executed when predetermined conditions are met. The advent of the blockchain technology has made the implementation of smart contract possible [10]. Nowadays,the term of smart contract is popularly used to refer [9] to code scripts that run synchronously on multiple nodes of a distributed ledger namely blockchain. Ethereum is the most popular platform to implement a smart contract.

E. Ethereum

Ethereum is an open-source blockchain platform that enables the development of smart contracts [12] using a Turing-complete programming language called "solidity" [11]. Then a solidity compiler transforms a source code into bytecode in order to be interpreted by Ethereum Virtual Machine (EVM). The interaction between Ethereum smart contracts and the users is through transactions. Indeed, the Ethereum platform supports two types of accounts: user accounts and smart contract accounts. This latter is assigned to the contract, once it is deployed.User account is assigned to the Ethereum users in order to deploy contracts and to interact with them. Ether is the cryptocurrency used in the Ethereum platform. The transaction cost is defined according to two parameters: Gas limit and Gas price. Gas limit is the maximum amount of gas that the user can pay and the Gas price is the amount of Ether that the user is willing to pay for one unit of gas [13].

V. The Enhanced MQTT Architecture

The architecture presented in this section is a security enhancement of the current MQTT protocol architecture.The goal is to strengthen the protocol's security to meet supply chain requirements without disrupting its normal operation.This architecture is based on blockchain and smart contracts to automatize authentication, publication, and subscription.it aims also to ensure data confidentiality and integrity .As depicted in Fig. 2, it is composed of clients (Publisher and subscriber) which represent the IOT devices used by each entity in supply chain and the broker network which execute automatically smart contracts.The choice of consortium blockchain allows each supply chain entity to have a trusted administrator who registers the devices in the blockchain as well as the topics which can publish or subscribe on. The communication between the components is divided into three phases: The

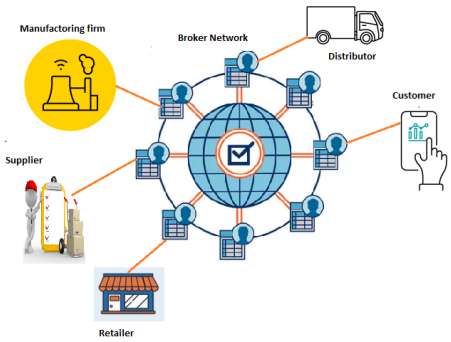

Fig. 2. The enhanced MQTT architecture.

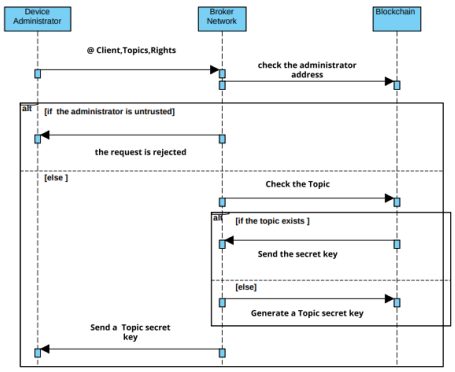

Fig. 3. Registration sequence diagram.

registration phase,the Connecting phase and the Publishing phase. Each phase is automatically controlled by a smart contract.To address the drawbacks of permanent password,the new architecture is based on OTP (One Time password) for authentication. This password that becomes invalid in few minutes protect the architecture from the identity spoofing.

A. Registration Phase

During this phase, the device administrator (Trusted node) calls the registration smart contract in order to register his devices in the blockchain and to recover the necessary keys for OTP calculating and for messages encryption. These keys are communicated to the devices in an out-of-band mode.Since the exchanged information is critical and the administrator has no computation and memory constraints, the TLS protocol can be used in this phase.After this transaction, for each device, a token containing the device address as well a list of topics that the device has the right to publish or subscribe on is generated and stored in the blockchain. Also,for each new topic a key is automatically generated and stored in the blockchain. The main steps of the registration phase are shown in Fig. 3.

B. Connecting Phase

As the current MQTT architecture, before publishing or subscribing in the topic the client must first connect to the broker.In this architecture,the device calls the connecting smart contract to connect to the broker network.Before allowing the communication,first the smart contract verifies the device registration and the Packet Number in order to avoid a Reply attack then it sends a challenge necessary for the one-time password (OTP) calculating.The device calculates the OTP and sends the a hash OTP. The smart contract verifies the hash OTP and then allows the connection to the broker network. The main steps of the connecting phase are shown in Fig. 4.

1) OTP calculating: The authentication process is based on one-time password (OTP) in order to avoid identity spoofing.The OTP calculation is based on or function .The inputs of this function are respectively :the list of publication's key Topics, the list of subscription's key Topics and the challenge.The detail of OTP Calculating is depicted in the equation (1):

$$OTP = F(LKP, LKS, challenge) \quad (1)$$

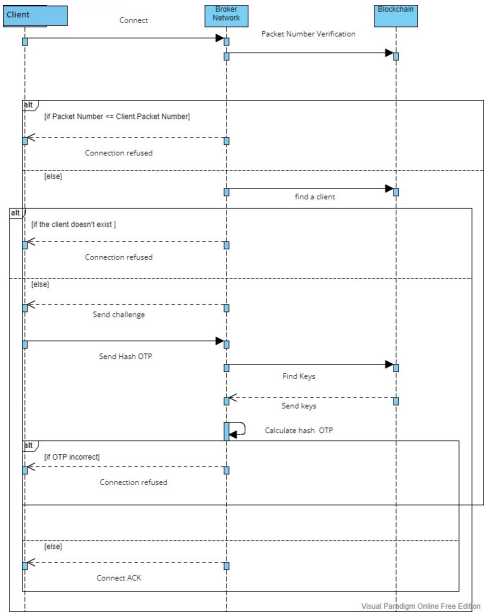

Fig. 4. Connection sequence diagram.

- F :is OR function.
- LKP: the list of publication's key Topics.
- LKS: the list of subscription 's key Topics.

N.B List of Keys and function are communicated to the device in outband mode.

C. Publishing Phase

As depicted in Fig. 5, the publishing process is the same as the current MQTT architecture in overall. However, we have added some security measures in order to strengthen the architecture. Indeed,in this phase, the device first calls the publishing smart contract. This latter checks its rights on publishing in this topic before allowing it to publish. Then, it notifies all the subscribers in that topic. In order to ensure data confidentiality, the published messages are encrypted with the topic key.

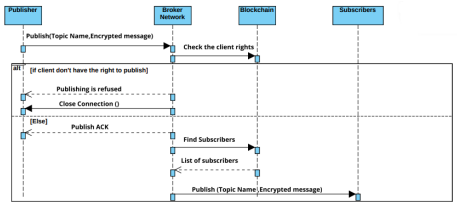

Fig. 5. Publishing sequence diagram.

VI. Smart Contracts Implementation

The authentication mechanism as well as the publication and the subscription processes are implemented in smart contracts for automatization. They are implemented in consortium blockchain and executed by broker network.The smart contracts proposed in this paper are developed using solidity language in a remix environment.

A. Registration Smart Contract

In order to register his device, the Device administrator calls the resgistrationMain function depicted in Fig. 6, This function allows the device administrator to interact with the registration smart contract. The administrator communicates to the function the following information: client address as well as the topics that the device has the right to publish or subscribe. Before allowing device registration,the smart contract first verifies the administrator privileges. Then it stores in the blockchain the required device information for authentication, publication and subscription. It also creates a topic if it doesn't exist. Finally, it sends to the administrator the necessary keys for OTP calculating as well as for messages encryption.

B. Connecting Smart Contract

. The functions used to interact with the connecting smart contract are: connectionMAin Fig. 7 and OTPVerification Fig. 8. When a device sends a connecting message to the broker network, the connectionMAin function verifies the packet number in order to avoid a reply attack and then, it checks the existence of the device in the blockchain. After the previous verifications, the function sends a challenge to a device. In its turn, the device calculates the Hash OTP using the method detailed in the connecting phase section and calls the OTPVerification function. This latter verifies the hash OTP to allow or deny the connection.

C. Publishing Smart Contract

The interaction between the device and the publishing smart contract is through the publish function Fig.9. Indeed, the device calls this function for a publishing request. The function verifies the client's rights. Once the publishing is allowed, it sends a notification to all subscribers on this topic.

VII. Simulation Tests

A. Case Study

In this section, we will perform simulation tests attack through a case study. Considering the simplified supply chain

```
function resgistrationMain(address client_address,Topic memory client_right)
public returns (uint){
bool check_administrator=checkingAdministrator(msg.sender);
if (check_administrator==false){
        emit ConnectionMessage("You don't have a privilege to add devices!The
connection is closed");
    return (0);}
    else {
        adddevice(client_address, client_right) ;
      bool ckeck=CheckTopic(client_right.name);
        if (ckeck==false){
          uint key_Topic=  addingTopic(client_right.name);
          emit ConnectionMessage("The client is added to the blockchain and
          the key is generated");
          return(key_Topic);

        }
        else
          for (uint h=0;h<i;i++) {
              bool
comparaison=stringComparison(Keys[h].name,client_right.name);
                if (comparaison==true){
        emit ConnectionMessage("The client is added to the blockchain and
        the key is sent");



                return(Keys[h].key) ;}}
    }
}
```

Fig. 6. Resgistration main function.

```
function connectionMAin(uint Packet_Number) public returns(uint) {   infinite gas
   bool  check_packet=Packet_verification(Packet_Number);
    if (check_packet==false){
        emit ConnectionMessage("Reply attack is detected and the connection is refused");
        return(0); }
    else { Clients[msg.sender].lastPacketNumber=Packet_Number;
        if(Clients[msg.sender].taille_mapping==0){
        emit ConnectionMessage("Connection is unauthorized for unregistered device");
        return(1);}
        else {
            emit ConnectionMessage("The challenge is sent");
            challenge =Calculchallenge();
            return(challenge);} } }
```

Fig. 7. connection main function.

```
function OTP_Verification(uint hashOTP_Client) public returns (bool) {   infinite gas
    uint l=Clients[msg.sender].taille_mapping;
 for(uint j=0;j<l;j++){
     string memory Topic_name= Clients[msg.sender].Topics[j].name;
     bool  Topic_read=Clients[msg.sender].Topics[j].read;
     bool   Topic_write=Clients[msg.sender].Topics[j].write;
     uint Key=reccuperercle (Topic_name);
     if (Topic_read==true){keysPublish.push(Key);}
     if (Topic_write==true){Keyssubscribe.push(Key);}}
uint keyPublish=0;uint KeySubscribe=0;uint otp;
for (uint h=0;h<keysPublish.length;h++){ keyPublish=keysPublish[h]+keyPublish;}
for (uint m=0;m<Keyssubscribe.length;m++){KeySubscribe=Keyssubscribe[m]+KeySubscribe;}
otp=keyPublish+KeySubscribe+challenge;
uint hash_OTP = uint(keccak256(abi.encodePacked(otp)));
if(hash_OTP==hashOTP_Client)
{ emit ConnectionMessage("OTP is correct and the connection is accepted");
    return(true);}
else {emit ConnectionMessage("OTP is incorrect and the connection is refused");
    return (false);}}
```

Fig. 8. OTP verification function.

```
function publish(string memory Topic_name) public {   infinite gas
    uint p;
    uint y;
  for( p=0;p<Clients[msg.sender].taille_mapping;p++){
      bool comparaison=stringComparison(Clients[msg.sender].Topics[p].name,Topic_name);
      if(comparaison==true){
        if (Clients[msg.sender].Topics[p].write==false){
          emit ConnectionMessage("You don't have a right to publish and connection is closed!");
          break;  }
        else{ emit ConnectionMessage("Your message will be sent to all subscribers");
            break;} }}
  if(p==Clients[msg.sender].taille_mapping)
     emit ConnectionMessage("The topic doesn't exist in your list and the connection is closed");
  else {for(uint t=0;t<taille_mapping_Clients;t++){
          address o= Client[t];
              for ( y=0;y<Clients[o].taille_mapping;y++){
              bool comparaison1=stringComparison(Clients[o].Topics[y].name,Topic_name);
              if (comparaison1==true && Clients[o].Topics[y].read==true )
              Subscribers.push(o); } }
  Display();}
```

Fig. 9. Publish function.

depicted in Fig.10 and supposing that the products are sensitive to temperature and humidity. Hence, it is important to monitor these parameters in the whole product life cycle through respectively temperature and humidity sensors. The Manufactory adds product information such as Fabrication date,end date and product ingredients in RFID Tag. This information will be published through a RFID reader. In the distribution phase a manufacturer,a retailer as well as the customer need to know the product location that is why a GPS device is used in this phase. Each entity in the supply chain has an end

Fig. 10. Simplified supply chain [14].

device in order to monitor the product manufacture.Device Administrator of each entity ensures the configuration and the maintenance of devices belonging to it. In order to collect customers' feedback, the trusted retailer administrator adds the end device customer in the blockchain to publish his feedback later . The topics are structured in hierarchical way. Tables I, II, III, IV, V, VI, VII, VIII, IX,X, XI, XII, XIII, XIV, XV bellow summarizes the topics and the devices using in this case as well as the Devices' rights for each topic.

B. Attack Simulation Tests

This section describes, for each phase, the nominal scenario as well as scenarios of possible attacks.

1) Registration phase:

1) Scenario 1:The nominal scenario
   - A trusted Aministrator (supplier Administrator for example) calls the registrationMain function to register his device (Supplier Temperature sensor for example). The device has the following address (0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db). Table I depicts the device's rights (Fig. 11).
   - The smart contract verifies the administrator address, then it registers the device in the blockchain and sends the secret topic key to the administrator (Fig. 12).
2) Scenario 2: Device is added by an untrusted administrator:
   - An unknown administrator tries to add a new Device. As depicted in Fig. 13, an error message is displayed.

2) Connecting phase:

1) Scenario 3:The nominal scenario
   - A registered Device (Supplier Temperature sensor for example) calls connectionMain function to connect to the broker network.
   - The Smart contract verifies that the device is already registered, checks the packet number to avoid a reply attack and then sends a challenge Fig. 14.
   - After receiving the challenge, the device calculates the OTP and sends the hash OTP Fig. 15.
   - The OTPVerification function verifies that the sent OTP is correct, then it allows the connection to the broker network Fig.16.
2) Scenario 4:The Device sends a wrong Hash OTP
   - A registered device (Supplier Temperature sensor for example) calls connectionMain function to connect to the broker network.
   - The Smart contract verifies that the device is already registered, checks the packet number to avoid a reply attack and then sends a challenge.
   - The device sends a wrong a hash OTP;The network broker refuses the connection Fig.17.
3) Scenario 5:A malicious device connection:
   - When a malicious device sends a connect message and it is not registered the connection is refused and an error message is displayed Fig.18.

TABLE I. TOPIC 1

Topic Name	Device	Publish	Subscribe
Temperature /raw Material	Supplier Temperature sensor	X	-
	End device Manufactory firm	-	X
	End device Retailer	-	X
	End device customer	-	X

TABLE II. TOPIC 2

Topic Name	Device	Publish	Subscribe
Humidity /raw Material	Supplier humidity sensor	X	-
	End device Manufactory firm	-	X
	End device Retailer	-	X
	End device customer	-	X

TABLE III. TOPIC 3

Topic Name	Device	Publish	Subscribe
Product information	RFID Reader	X	-
	End device Retailer	-	X
	End device customer	-	X

TABLE IV. TOPIC 4

Topic Name	Device	Publish	Subscribe
Temperature / warehouse	Manufactory Temperature sensor	X	-
	End device Manufactory firm	-	X
	End device Retailer	-	X
	End device customer	-	X

TABLE V. TOPIC 5

Topic Name	Device	Publish	Subscribe
humidity /warehouse	Manufactory humidity sensor	X	-
	End device Manufactory firm	-	X
	End device Retailer	-	X
	End device customer	-	X

TABLE VI. TOPIC 6

Topic Name	Device	Publish	Subscribe
Geolocation	GPS	X	-
	End device Manufactory firm	-	X
	End device Retailer	-	X
	End device customer	-	X

TABLE VII. TOPIC 7

Topic Name	Device	Publish	Subscribe
Temperature /Distribution	Distribution Temperature sensor	X	-
	End device Manufactory firm	-	X
	End device Retailer	-	X
	End device customer	-	X

TABLE VIII. TOPIC 8

Topic Name	Device	Publish	Subscribe
humidity /Distribution	Distribution humidity sensor	X	-
	End device Manufactory firm	-	X
	End device Retailer	-	X
	End device customer	-	X

TABLE IX. TOPIC 9

Topic Name	Device	Publish	Subscribe
Temperature/warehouse Retailer	Retailer Temperature sensor	X	-
	End device Manufactory firm	-	X
	End device Retailer	-	X
	End device customer	-	X

TABLE X. TOPIC 10

Topic Name	Device	Publish	Subscribe
Humidity /warehousse Retailer	Retailer humidity sensor	X	-
	End device Manufactory firm	-	X
	End device Retailer	-	X
	End device customer	-	X

TABLE XI. TOPIC 11

Topic Name	Device	Publish	Subscribe
Customer feedback	End device customer	X	-
	End device Manufactory firm	-	X
	End device Retailer	-	X

TABLE XII. TOPIC 12

Topic Name	Device	Publish	Subscribe
Configuration/ raw Material	Supplier administrator Device	X	-
	Supplier Temperature sensor	-	X
	Supplier humidity sensor	-	X

TABLE XIII. TOPIC 13

Topic Name	Device	Publish	Subscribe
Configuration/ Manufactory	Manufactory administrator Device	X	-
	Manufactory Temperature sensor	-	X
	Manufactory humidity sensor	-	X

TABLE XIV. TOPIC 14

Topic Name	Device	Publish	Subscribe
Configuration/ Distribution	Distribution administrator Device	X	-
	Distribution Temperature sensor	-	X
	Distribution humidity sensor -	X	
	GPS -	X	

TABLE XV. TOPIC 15

Topic Name	Device	Publish	Subscribe
Configuration/ Retailer	Retailer administrator Device	X	-
	Retailer Temperature sensor	-	X
	Retailer humidity sensor	-	X

4) Scenario 6:Reply attack simulation:
   - A registered device calls connectionMain function to connect to the broker network.
   - A hacker forwards the same device's message to deceive the broker network.
   - The broker network refuses the connection

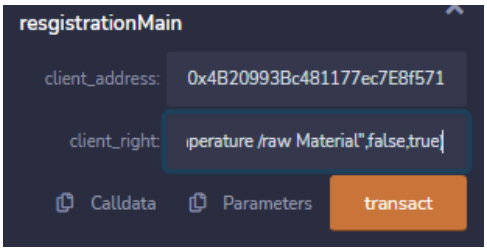

Fig. 11. Device registration transaction (Scenario 1).

Fig. 12. Device registration transaction output (Scenario 1).

Fig. 13. An error message transaction output (scenario2).

Fig. 14. Connection main transaction's output (Scenario3).

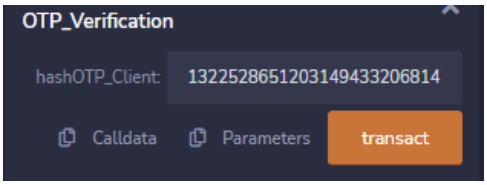

Fig. 15. OTP verification transaction (Senario 3).

Fig. 16. Log message accepted connection (Senario 3).

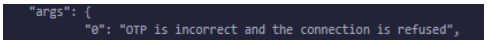

Fig. 17. Log message: Refused connection (Scenario 4).

and an error message is displayed Fig.19

3) Publishing phase:

1) Scenario 7:The nominal scenario:
   - A registered Device (Supplier Temperature sensor for example)who has address (0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db) calls publish function in order to publish in "Temperature /raw Material"Fig. 20 .
   - The smart contract verifies the Supplier Temperature sensor's rights. Then it returns the addresses of all subscribers in the Topic Fig. 21.
2) Scenario 8: The topic doesn't exist in the Devices's list or it doesn't have the right to publish on:
   - A registered Device (End device Manufactory firm for example) publishes in "Temperature

Fig. 18. Log Message: Unregistered client (Scenario 5).

Fig. 19. Log Message: Reply attack (Scenario 6).

/raw Material" topic (doesn't have the right to publish on) .

- As depicted in Fig. 22, the network broker refuses the publishing in the topic, the connection is closed and an error message is displayed.

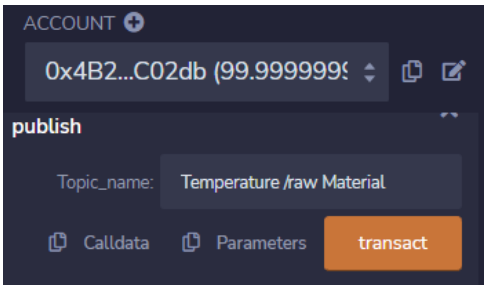

Fig. 20. Device publication transaction (Scenario7).

Fig. 21. List of subscribers (Scenario7).

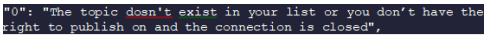

Fig. 22. Log message: Publishing is refused (Scenario 8).

C. Discussion and Studies

The solution proposed in this paper responds to the problem raised, which consists in designing an architecture that meets the supply chains security requirements while taking into consideration the constrained environments properties. The attack simulation test scenarios have been designed to test the resistance of the new architecture to the attacks and security issues already mentioned in section III. Denial of service (DOS) and distributed denial-of-service (DDOS) attacks that jeopardize data availability have been addressed through the decentralized architecture of the blockchain and the use of a broker network instead of one. The use of the consortium blockchain perfectly meets the need of the supply chain, which stipulates the communication and sharing of data between several independent entities. Each entity designates an administrator who registers the entity's devices as well as the topics on which they are allowed to publish or subscribe on(Scenario 1). Unlike the standard MQTT architecture where any device can connect, publish or subscribe in the broker, the new architecture allows connection only for already registered devices (Scenario 5).The devices publication is also allowed only for topics designated by administrators in the registration phase (Scenario 8).To avoid MAN in the middle attacks and preserve data confidentiality in a constrained environment, the authentication data exchange is carried out in several stages. Firstly, when registering devices, the administrator collects the keys necessary for calculating the OTP, which will be communicated to the device in out-of-band mode (Scenario 1). Then, during the connection phase, the broker network sends a challenge, another parameter used in the OTP calculation, to a device (Scenario 3). After calculating the OTP, the device sends the OTP hash for verification. Thus, even if a hacker is positioned between the device and the broker network, he does not have all the information needed to spoof the device's identity. Reply attacks are avoided thanks to packet number verification (Scenario 6).

VIII. The Contribution of the Article

The goal of this paper is to enhance the MQTT security protocol in order to fit the supply chain requirements and without affecting the overall operation of this protocol. It proposed a holistic solution based on blockchain and smart contracts to automatize the authentication, connecting and publishing processes.It used a consortium blockchain since supply chain requires interaction between independent entities. Each entity appoints a trusted administrator who registers the devices in the blockchain as well as the topics that it has the right to publish or subscribe on.The TLS protocol can be used in this phase since the devices are not constrained. All the necessary keys used for OTP calculating are communicated in this phase. The paper proposed several security measures to protect the architecture from the most common attacks and to ensure data confidentiality, integrity and availability. Indeed,it used a broker network instead of one to avoid a denial of service.It used the OTP hash for connection to the broker network in order to protect the device from identity spoofing .it also automatically verified the packet number before any transaction to protect the architecture from reply attacks. The simulation tests attack showed that the new architecture is resilient to these attacks.

IX. Conclusion

In summary, this paper proposed a MQTT security improvement in order to fit supply chains requirements.It aims to address security challenges without affecting the overall operation or performances.The new architecture is based on blockchain and smart contracts in order to avoid a single point of failure, to ensure data immutability and to automatize the authentication mechanism as well as the publishing and the subscribing processes. The paper proposed several security measures to strengthen the MQTT architecture.Indeed,it requires the device registration before network integration.This task is performed by a trusted administrator of each supply chain entity.To lighten the protocol, the architecture has used a TLS only in this phase when the required keys for computing OTP and encrypting messages are communicated to device administrator. This later communicates this critical information to the device in out band mode. It is also based on OTP for authentication which protects the architecture from man-in-the-middle attacks. Packet number verification is another security measure used in this architecture in order to avoid a reply

attack. The attack simulation tests are shown the resistance of our architecture against malicious attacks. On the other hand, this paper is limited only to the implementation and the test of the smart contracts side, that is why in our future work, we will try to implement the end-to-end mechanism and perform the attack tests simulation on the whole architecture.

REFERENCES

[1] Janvier-James, Assey Mbang. "A new introduction to supply chains and supply chain management: Definitions and theories perspective." International Business Research 5.1: 194-207 (2012).

[2] Helo, Petri, and Yuqiuge Hao. "Blockchains in operations and supply chains: A model and reference implementation." Computers & Industrial Engineering 136: 242-251(2019).

[3] Ageron, Blandine, Omar Bentahar, and Angappa Gunasekaran. "Digital supply chain: challenges and future directions." Supply Chain Forum: An International Journal. Vol. 21. No. 3. Taylor & Francis, 2020.

[4] Wu, Lifang, et al. "Smart supply chain management: a review and implications for future research." The International Journal of Logistics Management (2016).

[5] Buccafurri, Francesco, Vincenzo De Angelis, and Roberto Nardone. "Securing mqtt by blockchain-based otp authentication." Sensors 20.7 (2020).

[6] Aknin Raouya, and Youssef Bentaleb. "Securing MQTT Architecture Using a Blockchain." Advances in Information, Communication and Cybersecurity: Proceedings of ICI2C'21. Springer International Publishing, (2022).

[7] Al Enany, Marwa O., Hany M. Harb, and Gamal Attiya. "A New Back-off Algorithm with Priority Scheduling for MQTT Protocol and IoT Protocols." International Journal of Advanced Computer Science and Applications 12.11 (2021).

[8] Zorkany, M., K. Fahmy, and Ahmed Yahya. "Performance evaluation of iot messaging protocol implementation for e-health systems." International Journal of Advanced Computer Science and Applications 10.11 (2019).

[9] Elgendy, Mohamed Abdel Kader Mohamed, Mohamed Aborizka, and Ali Mohamed Nabil Allam. "A Blockchain-based Model for Securing IoT Transactions in a Healthcare Environment." International Journal of Advanced Computer Science and Applications 13.9 (2022).

[10] Zou, Weiqin, et al. "Smart contract development: Challenges and opportunities." IEEE Transactions on Software Engineering 47.10: 2084-2106 (2019).

[11] Wang, Zeli, et al. "Ethereum smart contract security research: survey and future research opportunities." Frontiers of Computer Science 15: 1-18(2021).

[12] Hu, Teng, et al. "Transaction-based classification and detection approach for Ethereum smart contract." Information Processing & Management 58.2: 102462(2021).

[13] Oliva, Gustavo A., Ahmed E. Hassan, and Zhen Ming Jiang. "An exploratory study of smart contracts in the Ethereum blockchain platform." Empirical Software Engineering 25: 1864-1904(2020).

[14] Stadtler, Hartmut. "Supply chain management: An overview." Supply chain management and advanced planning: Concepts, models, software, and case studies : 3-28 (2015).

[15] Chen, Fu, et al. "A review on the study on MQTT security challenge." 2020 IEEE International Conference on Smart Cloud (SmartCloud). IEEE, (2020).

[16] Bhawiyuga, Adhitya, Mahendra Data, and Andri Warda. "Architectural design of token based authentication of MQTT protocol in constrained IoT device." 2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA). IEEE, (2017).

[17] Laghari, ShaA, et al. "Cyberattacks and vociferous implications on SECS/GEM communications in industry 4.0 ecosystem." International Journal of Advanced Computer Science and Applications 12.7 (2021).

[18] Andy, Syaiful, Budi Rahardjo, and Bagus Hanindhito. "Attack scenarios and security analysis of MQTT communication protocol in IoT system." 2017 4th International conference on electrical engineering, computer science and informatics (EECSI). IEEE, (2017).

[19] Vaccari, Ivan, Maurizio Aiello, and Enrico Cambiaso. "SlowITe, a novel denial of service attack affecting MQTT." Sensors 20.10: 2932(2020).

[20] Wu, Hanqing, et al. "Data management in supply chain using blockchain: Challenges and a case study." 28th International Conference on Computer Communication and Networks (ICCCN). IEEE, (2019).

[21] Yeboah-Ofori, Abel, et al. "Cyber threat predictive analytics for improving cyber supply chain security." IEEE Access 9 : 94318-94337 (2021).

[22] Turjo, Manoshi Das, et al. "Smart supply chain management using the blockchain and smart contract." Scientific programming 2021 :1-12 (2021).

[23] Laxmi, Aishwarya Raj, and Ayaskanta Mishra. "Automation in supply chain management system using Internet of Things (IoT)." International Journal of Engineering Technology 7.2 :777-783(2018).