International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-4, November 2019

A Mutual Authentication Based on Multi Agent System for IoT-Cloud Paradigm

Amrani Ayoub, Rafalia Najat, Abouchabaka Jaafar

Abstract*: With the revolution of cloud computing and Internet of Things (IoT), users can access IoT services in different domain such as smart home, smart healthcare and smart factory. The connected device using sensors generates hundreds of real time communication that transfers sensitive user's data to the cloud server. This sensitive information is transferred via an open and insecure channel. The Cloud-IoT paradigm has opened a new window of security challenges related to it. Therefore secure authentication schemes are of utmost importance to provide a secure access data and IoT services for legal users. This paper propose a secure distributed mutual authentication protocol based on our previous published scheme "Amrani et Al.", suitable for IoT environment. Moreover we implement the proposed scheme using Multi Agent System (MAS) in JADE platform, analyze the results and compare its performance with "Hanaoui et Al." scheme. The proposed protocol can also protect a Mobile Agent (MA) while carrying sensitive information, from IoT platform to Cloud Server.

Keywords: Security, Multi agent System, Cloud Computing, Internet of Things, JADE

I. INTRODUCTION

Internet of things (IoT) refers to the networked interconnection of everyday objects. Through this concept we can develop hundreds of applications that can be implemented in different areas and industries as such as transport and logistics, health and smart environment... An IoT device has very interesting characteristics, it must be able to communicate and interact with its environment, and it would be worth saying that it is an object having the ability to communicate via various modes of communication with the world. With all this inter communication and exchanging data, IoT devices at this stage need cloud resources to run this mass of exchanged data, and to ensure good decision-making. Things in IoT are passive devices that do not inherit any form of intelligence. The Cloud is fully responsible for the intelligence of these objects.

Manuscript published on November 30, 2019.
* Correspondence Author
Amrani Ayoub*, IT department, faculty of science, Ibn Tofail University, Kenitra, Morocco. Email: amrani.ayoub@uit.ac.ma
Rafalia Najat, IT department, faculty of science, Ibn Tofail University, Kenitra, Morocco. Email: najat.rafalia@uit.ac.ma.
Abouchabaka Jaafar, IT department, faculty of science, Ibn Tofail University, Kenitra, Morocco. Email: amrani.ayoub@uit.ac.ma.

Multi-Agent Systems is a set of agents, that interact with each other, situated in a common environment, with intelligent behavior, and have the ability to achieve a certain goal. MAS can act and provide as a fitting solution for realizing embedded intelligence [1]. Agents can help objects to make autonomic decisions and reduce the amount of communication between object to objects and objects to cloud.

Security still remains the major issue while getting connected to the cloud for using its resources. Indeed due to the limited resources of connected objects such as processing capability and memory size, the use of cryptographic systems such as RSA will not be practicable to be implemented on IoT devices, because their security level depends on the length of the keys. In [2], we developed an interesting lightweight secure scheme based on elliptic curve for IoT-Cloud that ensures a mutual authentication. In this paper we learned from our previous scheme and use it to develop a new distributed security protocol with JADE platform, using multi agent system (MAS) that guarantees a mutual authentication based on elliptic curve, between the IoT and Cloud. The rest of this paper is structured as follows. In Section 2 we'll put the light Cloud-IoT security threats. In Section 3, we'll discuss some related works and security issues. Section 4, we'll summarize the preliminaries of elliptic curve and some notions used. In section 5 we will model the authentication protocol with MAS. An implementation of the proposed solution is provided using JADE [3] in section 6. Finally, section 7 concludes the paper.

II. SECURITY CONSTRAINTS AND REQUIREMENTS

A. Security Constraints

As we approach an increasingly interconnected society with progressively savvy items associated with the cloud, the dangers likewise increment and vulnerabilities that should be recognized and corrected duplicate. Be that as it may, this situation is additionally extremely alluring to cybercriminals, who find in the expansion of devices and applications an extraordinary motivating force for their activities. IoT security issues can be of various nature and happen at various levels. In what follow some of the most significant threats in IoT-Cloud environment:

Computer attacks: Computer assaults are the most widely recognized danger in a cloud situation. They can be Denial of Service (D-DOS) assaults, malware spread in IoT gadgets, abuses, and assaults on the client's security or even change of the electronic parts of the gadget.

Software vulnerabilities: Another real security challenge lies in the vulnerabilities of IoT applications and programming. These must remain refreshed, broke down, tried and designed effectively to anticipate security issues, both in stage and backend.

Data interception: Interchanges between IoT gadgets are another level where cyber security dangers may happen. Session kidnappings, or correspondence conventions and getting system information are some threats to which it is basic to receive security measures.

In order to fully enjoy the benefits of connected objects, it is imperative to ensure the security and privacy of users. We can resume in what follows some challenges, which we should working on:

Data privacy: Data collected by IoT devices must be secured.

Vulnerabilities in authentication: IoT device interact with other entities, and collect sensitive information. It's important to work in authentications mechanism

Data encryption: Sending sensitive data by unencrypted is a major security problem.

B. Security Requirements

The Security requirements that are relied upon to be met by the IoT security plans are as per the following [19]:

Authentication: The authentication process allows to a smart device to ensure the identity of the entity with which it communicates. It must also be verified that only the validated user has access to the IoT devices.

Authorization: It guarantees that only the approved objects have access to the resources and network services.

Access control: It is the process that ensures that the authenticated objects have only access to the resources of which one has permission.

III. RELATED WORKS

To reduce the time computing of an IoT device, recently schemes based on elliptic curve has been proposed and others have been implemented. The use of the elliptic curve comes for multiple reasons. The key size is one of the major reasons, which is very small compared to other cryptosystem keys. In 2009 Yang and Chang [5], an interesting mutual authentication scheme and a session key agreement between the user and the server was proposed, this algorithm does not exhaust the resources of the device, since it is the server that does all the work, but unfortunately this algorithm suffers from the clock synchronization and the offline password guessing [6]. In 2012 Hafizul et al. [7] by learning from Debiao et al's. He proposed a scheme consisting of four steps. Initialization phase, client registration, mutual authentication with key agreement and finally changing and updating the local private key phase, but again this scheme suffers from the password guessing and does not hide the identity of the client. Other protocols based on ECC have been proposed by Granjal et al. [8], Ray et al. [9] and Jiang et al. [10].

Not too long ago in 2015, a novel scheme appeared, proposed by Kalra and Sood [11], who have gained experience from other previously discussed. It's a very strong algorithm; it proposes a mutual authentication to secure the communication between IoT devices and the Cloud using HTTP cookies. It's a protocol that has been verified by AVISPA tool, but that has not been implemented in real case. In 2017 Kumari et al. [12] they proved that the Kalra and

Sood scheme is vulnerable against offline password guessing and insider attack.

In [13] proposed an authentication broker that integrates MAS and the single sign-on, to satisfy the confidentiality, integrity and non-repudiation requirement In [14] an approach of how to apply MAS to serve the authentication service in the multi-clients and multi-application environment has been proposed, and finally in [15] have developed an interesting security protocol based on a combination of TLS, RSA, ECC signature in order to create a secure channel that guarantees authentication, confidentiality and integrity for migration of a mobile agent from a platform to another while carrying sensitive information. Unfortunately this convergence requires a huge time of calculation, for the generation, the distribution of key and for the signature which increases the execution time. In an IoT environment the time execution is an element not to be neglected, because it is exhaust the resource of the devices.

IV. PRELIMINARIES OF ELLIPTIC CURVE CRYPTOGRAPHY

In this section we introduce the concept of ECC (elliptic curve cryptography). To get started, the RSA keys that have the recommended size, keep increasing to maintain sufficient encryption strength, from 1024 bits to 2048 bits a few years ago, are the most common used for SSL certificates. An alternative to RSA keys are the ECC keys. These two types of master keys share the same important property of being asymmetric algorithms (a key to Encrypt and a key to decrypt). However, ECC can offer the same level of encryption power for much shorter keys, providing better security while reducing computing requirements. The use of ECC short keys can help to save obvious costs. This reduced key size also allows us to implement and design faster and strong cryptographic operation, which makes ECC a very attractive option for devices with storage or processing power is limited, which is becoming increasingly common in the era of the Internet of Things. Table 1 give a comparison between ECC and RSA based on key size for same security level [16].

Table- I. Comparison of RSA and ECC based on key size

ECC key size (bits)	RSA key size	key size ratio
163	1024	1:16
256	3072	1:12
384	7680	1:20
512	15360	1:30

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Comparing to the existing PKC's [30, 31] (Public Key Cryptosystem). ECC offers better performance and more security. This concept was proposed first by Koblitz and Miller. An elliptic curve E(a ,b) over a finite prime field K, where a,b ∈ K is given in Eq. (1) bellow :

$$E(a,b): y^2 = x^3 + ax + b\ (modp) \quad (1)$$

Where p>3, a and b will have to fulfil the following condition $4a^3 + 27b^2 \neq 0$. K can be in the following fields{$\mathbb{R}, \mathbb{Q}, \mathbb{C}, \mathbb{Z}/p\mathbb{Z}$}. Let E be an elliptic curve defined on a field K, and two points P, Q E (K), L the line connecting P to Q (the tangent to E if P = Q) and R the third intersection point of L to E. Let L be the vertical line passing through R. We define P + Q E (K) as the second point of intersection of L' with E. With the law of composition (E (K), +) is an Abelian group whose neutral element is the point to infinity (O). Fig. 1 shows the basic curve for ECC [20].

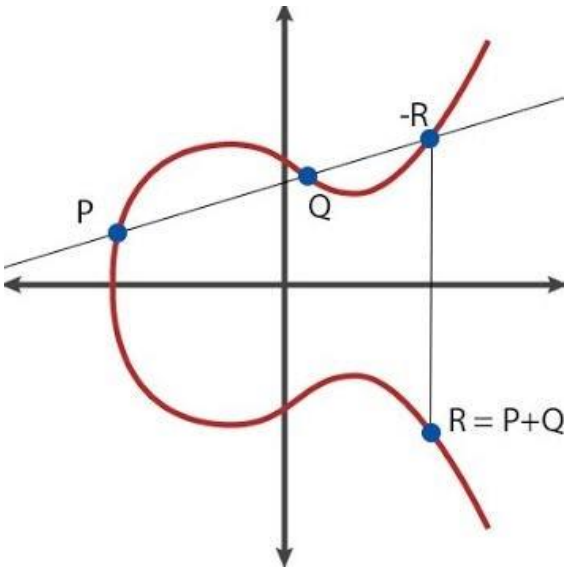

Fig. 1.Weierstrass elliptic curve

Scalar addition and scalar multiplications are two common operation on elliptic curves. Scalar multiplication nP over an elliptic curve E (a, b) is defined as repeated additions, as given in Eq. (2) below:

nP = P + P + P + … + P (nTimes) (2)

Where n ∈ K^*. Security of ECC depends on the complexity and difficulty of solving following problems:

Problem 1: Elliptic Curve Discrete Logarithm Problem (ECDLP).

We suppose a curve E($\mathbb{Z}$/n$\mathbb{Z}$). By giving a Q,K ∈ E($\mathbb{Z}$/n$\mathbb{Z}$), with Q a multiple of P. We need to find K that solves the following equation Q = KP. It is a difficult problem to solve. This is called, the discrete logarithm problem or (ECDLP).

Problem 2: Computational Diffie–Hellman Problem (CDHP)

Given three points P, u and v over an elliptic curve E(a,b), where u,v ∈ K^*. It's difficult to solve uvP over E(a,b).in polynomial time.

V. PROPOSED PROTOCOL

It is sure that an agent in an intranet of things will add several benefits to these devices, the aspect of intelligence, the autonomy of tasks and reducing the number of communication between object and especially between object and cloud. But with all these advantages, this merged technological, will not see the day if the security aspect has been neglected. We are talking about very sensitive information, circulating in the network that can be exposed and exploited. Recently the research in the MAS has become very active, the development and implementation of agents in different sectors. But rarely, where we found a community that deals with the security aspect. Authentication is a process that ensures and confirms a user's identity. And it is the most essential requirement as the agent from the IoT device and the Cloud must authenticate each other for secure communication. In this paper we have designed a mutual authentication scheme which has gained experience from [2] based on elliptic curves and using Multi agent system.

A. Multi agent system Model

In this section we'll detail our MAS model and we'll explain the role of each used agent. Our model consists of three phases, initialization phase, registration phase and finally the login and authentication phase.

1. Initialization Phase -

In our design, it's the Cloud who's in charge to initiate the parameters of the parameters necessary for the generation and distribution of keys as shown in figure 2. We assume that the Cloud has sufficient storage and calculation capacity to generate and distribute the keys to the sub for each session. First the Cloud Computing chooses an elliptic curve equation

$$y^2 = x^3 + ax + b \text{ Over } \mathbb{Z}_p \text{ , where } \mathbb{Z}_p \quad (3)$$

(p $>2^{160}$) is the finite field group. He selects two field elements $a, b \in \mathbb{Z}_p$. a and b must satisfy the following condition $4a^3 + 27b^2 \neq 0$. G is the base point of the elliptic curve with a prime order (n > 2^{160}), and O be the point at infinity such that n x G = O. Then the Cloud chooses random master secret key K_m from Z_p and computes public key

$L = G.K_m$. Finally broadcast the public parameters

$K_p \leftarrow (E_p, G, p, L)$.

2. Registration phase –

i. $device_i$ to $Cloud$ –

In order to register with the Cloud. At the device level, we need an agent named registration agent who will be in charge of calculating and presenting $I_i \leftarrow h(ID_i \parallel rG)$.

ii. $Cloud\ to\ Device_i$ –

After the Cloud receives I_i. An agent named "S_P_ag" computes the security parameters $K_i = h(N \parallel I_i \parallel ID_c)$, $K_s = K_m \oplus K_i$ and transfer it to "Co_ag". This last is an agent who's in charge of generating and computing $A_i \longrightarrow h(K_m \parallel K_i)$ and $A_i^{'} = A_i.G$ of each device. Then he computes other security settings $B_i = h(K_s \parallel I_i \parallel A_i^{'})$ and stores $B_i^{'} = B_i.G$, K_i corresponding to the identity ID_i of the device i in its database.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

3. Login & Authentication phase -

In this phase, we're going to need too agents authentication agent "Auth_ag" and login agent "Log_ag" in the both side, the cloud and the embedded device.

i. $device_i$ *to Cloud* -

The agent "Auth_ag" from the device platform computes $P_1 = N_1.G$ and $P_2 = h(N_1.A_i^{'})$ send it with the K_i to the Cloud through.

ii. $Cloud\ to\ Device_i$ -

The Cloud receives P_1 andP_2. This last computes $A_i \longrightarrow h(K_m \parallel K_i)$ by calculating $K_m = K_s \oplus K_i$ then computes the poin$P_2^* = h(P_1.A_i)$t. The "Auth_ag" checks whether the value of P_2^* is equal to the received value of P_2. If it's true, he computes the ECC point $P_3 = N_2.G$,$P_4 = N_2.B_i^{'}$, and sends P_3 , P_4 and K_s to the IoT device.

iii. $device_i$ *to Cloud* -

The "Log_ag" from the device platform receivesP_3 , P_4 then he computes $B_i = h(K_s \oplus I_i \oplus A_i^{'})$ and calculates ECC point $P_4^* = P_3.B_i$ and compares the value of P_4^* with the received value of P_4 if it's true the "Log_ag" calculates $V_i = h(P_2 \parallel K)$, where $K = N_1.P_3$ and sends V_i to the cloud.

iv. $Cloud\ to\ Device_i$ -

The "Log_ag" from the Cloud platform receivesV_i. Then he computes $V_i^* = h((P_1.A_i) \parallel K^*)$ where$K^* = N_2.P_1$. And compares the value to the received value of V_i.

B. Algorithm presentation

Algorithm 1: Initialization

Input: Ep : EC equation, K_m: Cloud platform secret key

Output: H: Reception platform public key, G: Generator point.

1. Chooses $E_p: (x^3 + ax + b)(mod\ p)$
2. Generates the generator point G.
3. Computes $H = G.K_m\ public\ key$

Algorithm 2: Registration

Input:

ID_i : Identity of the device i

ID_c : Identity of the cloud , K_m **:** secret key of the cloud

K_i : Key for device I , r : random number

And other security parameters : $A_i^{'}$,A_i, B_i , $B_i^{'}$

Output:

$I_{i=}$: The hashed identity of the device

1. Computes$I_i \leftarrow h(ID_i \parallel rG)$.
2. Sends I_i to the cloud
3. Select a random number N.
4. Computes $K_i = h(N \parallel I_i \parallel ID_c)$
5. Computes $K_s = K_m \oplus K_i$
6. Computes $A_i \longrightarrow h(K_m \parallel K_i)$
7. Computes $A_i^{'} = A_i.G$
8. Computes $B_i = h(K_s \parallel I_i \parallel A_i^{'})$
9. Computes $B_i^{'} = B_i.G$
10. Stores $B_i^{'}$,K_i andID_i.
11. Sends $A_i^{'}$ and K_i to the device i.

Algorithm 3: Login and Authentication

Input:

Select random number r_1; r_2 , K_m: Cloud platform secret key

G: Generator point.

Output:

ECC points : P_1, P_2, P_3, P_4

A_i : security parameter , K : session key , and set of parametersV_i, V_i^*

1. Computes $P_1 = N_1.G$.
2. Computes $P_2 = h(N_1.A_i^{'})$
3. Sends P_1 , P_2, K_i to the cloud
4. Computes $K_m = K_s \oplus K_i$
5. Computes $A_i \longrightarrow h(K_m \parallel K_i)$
6. Computes $P_2^* = h(P_1.A_i)$
7. Verifies P_2^* (? =) P_2,
8. Computes $P_3 = N_2.G$ and $P_4 = N_2.B_i^{'}$
9. Verifies P_4 (? =) P_4^*
10. Computes $K = N_1.P_3$
11. Computes $V_i = h(P_2 \parallel K)$
12. Sends V_i to the Cloud
13. Computes $K^* = N_2.P_1$
14. Computes $V_i^* = h((P_1.A_i) \parallel K^*)$
15. Verifies V_i (? =) V_i^*
16. Generates a session key

VI. IMPLEMENTATION AND RESULTS.

Here we expose our model that we implemented and the results got through implementing the proposed solution. In this work we used JADE [3] an open source framework for the development of peer-to-peer applications of intelligent agents. JADE offer a platform with FIPA [17] specifications and an API for developing agents with java. The main characteristics of Jade platform is

- Deployed on one or more machines.
- Hosts a set of uniquely identified agents that can communicate bidirectionally with other agents, by sending messages whose content is expressed in ACL.
- Each agent runs in a container that provides it with its runtime environment, he can migrate inside the platform.
- Any platform must have a main container that registers the other containers.
- A platform is a set of active containers.

We present in figure-2, our model with a diagram, the different internal and external communication between agents that we have implemented, to establish a mutual authentication between the smart IoT Device and the Cloud

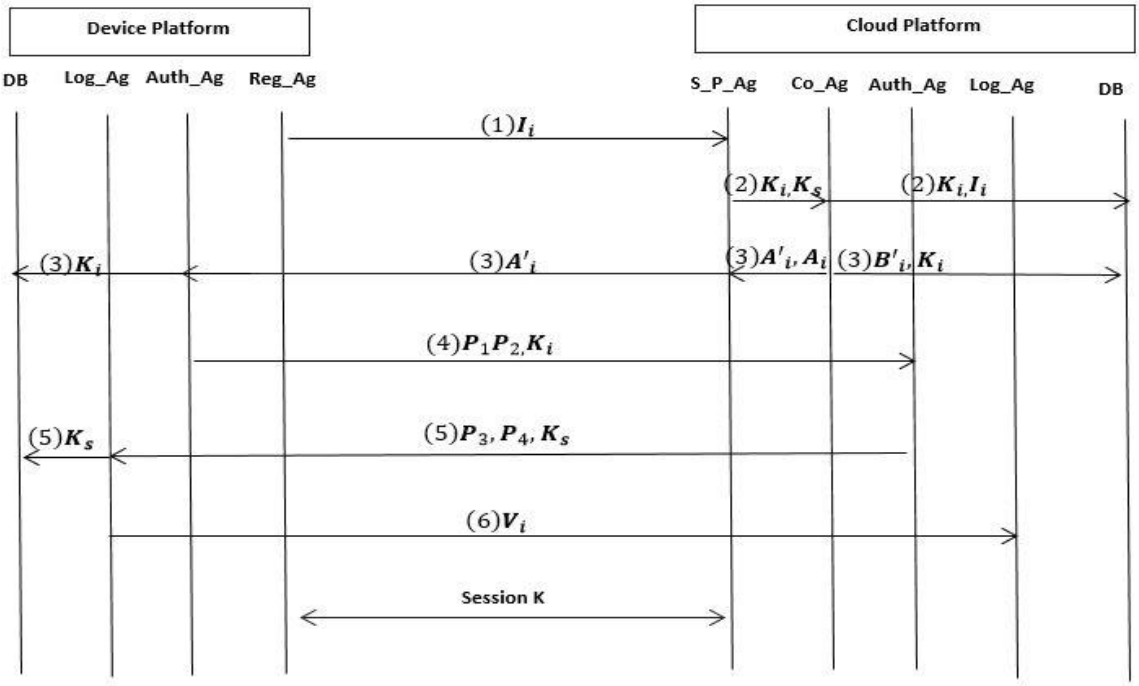

Fig. 2. MAS Model

Server. Each agent will perform his task, according to the scenario. The practical tests of the implementation are carried out in a machine which contains two containers first represent the IoT machine and the second the Cloud server. The technical characteristics of the machine are: Intel Core i5 processor has 2.2 GHz with 6 GB of RAM..

A. Parameter used

In this subsection we're going to present some major parameter used during our implementation, namely the parameters of the elliptical curve, the hush function used and we'll give also an agent code snippet.

According to section 4. In order to generate the point G, we must first define our elliptic curve

$$E: (x, y) | \, y^2 \equiv x^3 + ax + b \; with \; a, b \in K. \quad (4)$$

In [18] they published a useful standard of recommended set of elliptic curve domain parameters. In our work we used the P-192 Curve which equals to 1536 bit RAS Key length. In what follow we go through each parameter {P,G,a,b,q} in decimal form:

i. The Prime P-

P_192="6277101735386680763835789423207666416083908700390324961279".

ii. a and b –

a="6277101735386680763835789423207666416083908700390324961276".

b="2455155546008943817740293915197451784769108058161191238065".

iii. The base Point G:

X_G="602046282375688656758213480587526111916698976636884684818".

Y_G = "174050332293622031404857552280219410364023488927386650641".

iv. Order q of the point G:

q="627710173538668076383578942317605901376719477 3182842284081".

Point addition and Point doubling are two operations which can be applied to a base point G on the elliptic curve. All the parameters exchanged in our protocol between the IoT device and the cloud servers in order to establish a mutual authentication were hashed with the SHA-256.

B. Evaluation and results

In figure-3, we present our RMA (remote agent management), it's a GUI that groups the two containers created. The first container "Main-container" represent IoT-platform, and the second represent the cloud-platform. Each container, brings together a multi agent system that communicates with each other in order to establish mutual authentication between the IoT and Cloud.

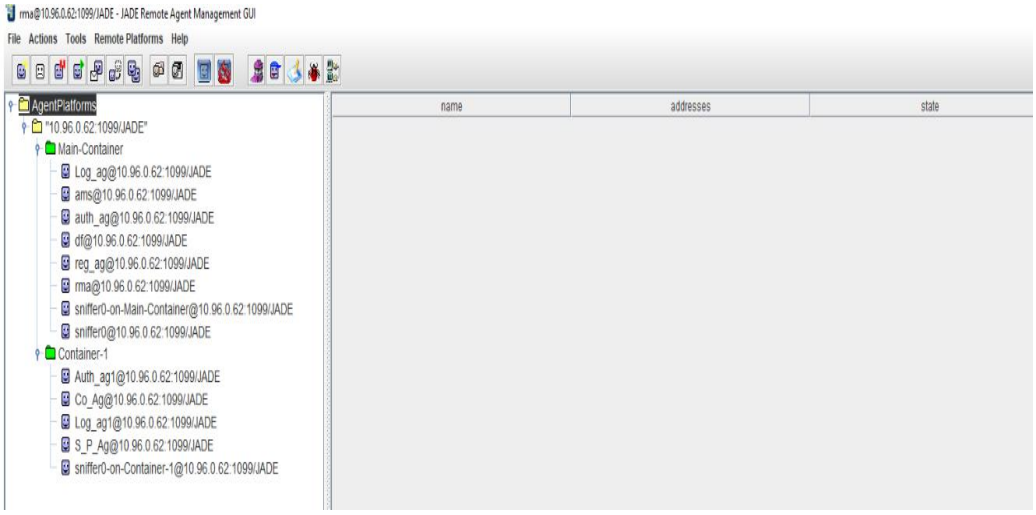

Fig3. Remote Agent Management

The sniffer agent of the JADE platform allows us to visualize the sequence of messages exchanged between the agents of the two platforms. In figure-4 summarizes in a scenario, the communication established between the two containers. And that only approves and validates the model that we have established in figure-3.

To have a better visibility on the routing of the parameters between the two platforms, the figures bellow shares the results of the execution. We have thoroughly tested the results of the test following the model on Figures 3 in order to understand how it works. The first phase of the test was to initialize the parameters of the Elliptic curve in the cloud platform, then in order to establish a mutual authentication we respected the three phases: registration, the login and the authentication phase.

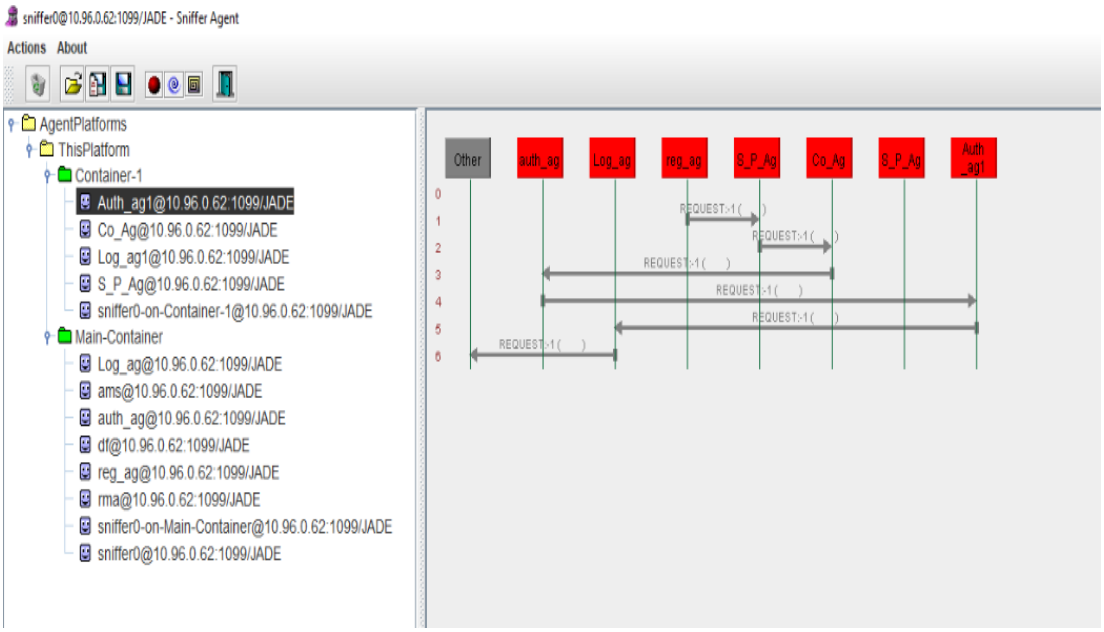

Fig4. Remote Agent Management

```
**************************** Begin REG_AG  ****************************
Ii = 8b86788c62f5c10e3d7c35041cef4c8f38064ef11606fbe18f7f94ca6edfe5ae
Ii -----> S_P_Ag
**************************** END REG_AG    ****************************
```

Fig 5. Reg_Ag

```
************************* Begin S_P_AG ***************************************
Ii recu (1)  -> 8b86788c62f5c10e3d7c35041cef4c8f38064ef11606fbe18f7f94ca6edfe5ae
-> KM calcule = 5f24ca773d0c79c7acc3fe906ded6aed82096bb3d152f9a4a18be308cb525196
-> KI calcule = 579c934e4087a908de98da00e1003e2a29ee5f5af566ea6354cc3e328c8cc19e
-> KS = KM + KI =>08B859397D8BD0CF725B24908CED54C7ABE734E9243413C7F547DD3A47DE9008
KS + KI Send to -----> Co_AG
Ii stored in DB
************************* ENG S_P_AG    ***************************************
```

Fig 6. S_P_Ag

```
****************************** begin CO_AG ***************************************
KS + KI recu (2) -> 08B859397D8BD0CF725B24908CED54C7ABE734E9243413C7F547DD3A47DE9008-
                    579c934e4087a908de98da00e1003e2a29ee5f5af566ea6354cc3e328c8cc19e
-> AI = h(Km||Ki) => 7bd4c7be1e5aa8ed80e414cfaec35a3f19f7947c35baa3db71dabf7e4080d91f
-> A'i = Ai.G     => 1040067971350406560486125745745076404398323416624402989021
                     744127839171648146032702772591989233247268293638878650970
-> Bi = h(KS + Ii + A'i) => 18510696afb4c47a120b941e35c0747ff0868f77a6a3683c68c3dfd32e6bf3f2
-> B'i = Bi.G            => 1594984437765437069024055179892454292356122734783857332625
                            397958137854398843123506498478715312171057219235322077200 3
A'i envoye -------> Auth_Ag
Stored  KS , et B'i in DB
****************************** End CO_AG    ***************************************
```

Fig 5. Co_Ag

```
***************************** begin Auth_AG ***************************************
 A'i recu (3) -> 1040067971350406560486125745745076404398323416624402989021
                 744127839171648146032702772591989233247268293638878650970
 -> P1 = N1.G      => 3447117460266499760388376087438356979190075617130986385895
 -> P2 = h(N1.A'i) => 6351047d59352d12e86d42f47f234ec29a8339b65eec9589202079698fdbd2d4
 P1,P2,Ki Send to -----> Auth_ag1
***************************** End Auth_AG ***************************************
```

Fig 6. Auth_Ag

```
***************************** begin Auth1_AG ***************************************
P1 + P2 + Ki recu (4) ->  3447117460266499760388376087438356979190075617130986385895
                          9420500640415137854577959946285312414709478216007700827 37
                          6351047d59352d12e86d42f47f234ec29a8339b65eec9589202079698fdbd2d4
-> New KM = Ks + Ki   => 5F24CA773D0C79C7ACC3FE906DED6AED82096BB3D152F9A4A18BE308CB525196
       P2x = 6351047d59352d12e86d42f47f234ec29a8339b65eec9589202079698fdbd2d4
-> P3 = N2.G    => 24597500196220474257396157037402832588286689413460022155 89
                   189974623546792213343252611568737478471840585117956464321 3
-> P4 = N2.B'i => 4747916186628511358568165725601338303591066434623215151865
P3, P4 et KS Send to  -------> Log_ag
***************************** END Auth1_AG ***************************************
```

Fig 7. Auth1_Ag

```
***************************** begin Log_AG ***************************************
P3, P4 and KS recu (5)-> 24597500196220474257396157037402832588286689413460022155 89
                         189974623546792213343252611568737478471840585117956464321 3-
                         474791618662851135856816572560133830359106643462321515186 5-
                         08B859397D8BD0CF725B24908CED54C7ABE734E9243413C7F547DD3A47DE9008
-> new Bi = 18510696afb4c47a120b941e35c0747ff0868f77a6a3683c68c3dfd32e6bf3f2
-> P'4    = 4747916186628511358568165725601338303591066434623215151865
true !! Then
-> K  => 18256296063733412251477051762212243874623150437087295433 0
-> Vi => 4e3aecd76a1bef5eb19868359463c521a0aac547298023da451757375e7322f5
Vi envoye ----->  Log_ag
***************************** END Log_AG ***************************************
```

Fig 8. Log_Ag

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

```
***************************** begin Log1_AG ****************************************
Message recu 6 final-> 4e3aecd76a1bef5eb19868359463c521a0aac547298023da451757375e7322f5
                       6D19CFF44938CF30DEED1B15F515AA4FD38AB7225CF350B3653474147D5001D6
-> K'  => 18256296063733412251477051762212243874623150437087295433O
-> V'i =>4e3aecd76a1bef5eb19868359463c521a0aac547298023da451757375e7322f5

    session eshtabilished .....
***************************** END Log1_AG ******************************************
```

Fig 9. Log1_Ag

In this part we will present the execution time of our solution composed of several security mechanisms to guarantee the mutual authentication between IoT device and the Cloud server. Before presenting our time test result, let's consider the total time for our solution.

$$T_{total} = T_{registration} + T_{authentication} + T_{login} \quad (5)$$

Figure 10 we expose the detail of the execution time for each agent

Finally, according to the equation (5) the total time for executing our solution is:

$$T_{total} = 34 + 420 + 125 = 579$$

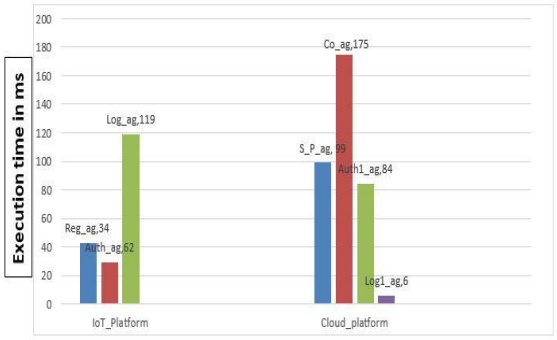

Fig 10. Execution time in ms for each agent

In what follows we compare our results with that of Hanaoui et al [15] protocol. Both solution are developed in Jade Platform, and both on containers located in the same platform.

The graph in the figure 14, resumes the execution time for both protocol. Regarding the execution time of Hanaoui's we neglected the time for the migration of the agent between the
two platforms and we focused only on the execution time that was required to establish the secure channel. In our proposed protocol each agent is responsible for performing a specific task that requires time, as shown in the following figure. The total period of the proposed protocol is defined in 579ms, while the Hanaoui's protocol is in 2213ms.

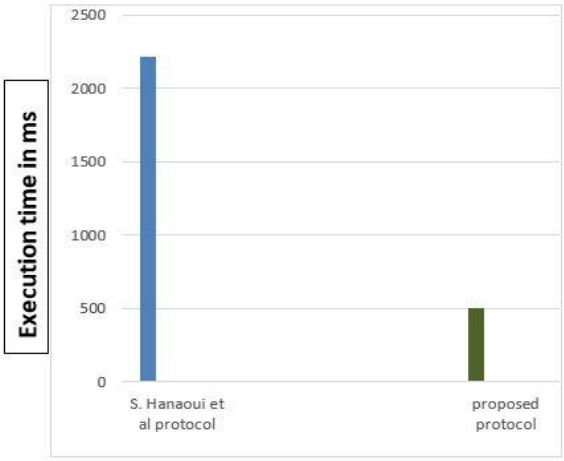

Fig 11. Comparison of the two solutions

VII. CONCLUSION

In this paper, we have detailed the failures and the security needs that a Cloud-IoT environment requires, especially with the low-capacity of the intelligent devices. Also we mentioned some works in the literature that tried with various protocol to secure the communication between IoT and Cloud. Later we have summarized the prerequisites of the elliptic curve needed to develop a mutual authentication algorithm. While emphasizing the utility and benefits that multi agent systems can bring in such a smart environment, we proposed a solution based on our scheme [2], that protect IoT-Cloud environment form different types of attacks like, man-in-the-middle, Cookie theft, Offline password guessing attack, Replay attack and Stolen-verifier attack. In the last section, we compared our results with those of Hanaoui's, we found that our scheme is better whether it is at the security level or the execution time The result of the execution confirms the performance, efficiency and adaptability. As future work, we intend to secure the migration of a mobile agent from one platform to another using this protocol.

REFERENCES

1. Semwal, T.; Nair, S.B. "AgPi: Agents on Raspberry Pi". Electronics 2016, 5,72.
2. A. Ayoub, N.Rafalia, A.Jaafar, "lightweight secure scheme for iot-cloud convergence based on elliptic curve" JATIT vol. 97 (13) ,Jan 15. 2019.
3. A. Rimassa G Bellifemine, F. Poggi. "a pa 2000 compliant agent development environment" Proceedings of the fifth international conference on Autonomous agents, pages 216-217,2001.

4. Danny B. Lange and Mitsuru Oshima. "Seven good reasons for mobile agents" 42(3):88-89, 1999-03-01. Available: http://www.moe-lange.com/danny/docs/7reasons.pdf.
5. Jen-Ho Yang and Chin-Chen Chang. "An ID-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem" Elsevier, J. Computer and Security 28(3):138-143 2009-05.
6. Ding wang, ying mei, chunguang ma, and zhen-shan cui. "Comments on an advanced dynamic id-based authentication scheme for cloud computing in wism", springer, Web Information Systems and Mining, pp 246-253, 2012.
7. Sk ha zul islam and g. p. biswas. "An improved id-based client authentication with key agreement scheme on ecc for mobile client-server environments" J. Theoretical and applied informatics, 24(4), january 2012.
8. Jorge granjal, edmundo monteiro, and jorge sa silva. "End-to-end transport-layer security for internet-integrated sensing applications with mutual and delegated ecc public-key authentication". In IP networking conference, 2013 pages 1-9 ieee, 2013.
9. Sangram ray and g p. biswas. "Establishment of ecc-based initial secrecy usable for ike implementation" Proceedings of the World Congress on Engineering. july 2012.
10. Rong jiang, chengzhe lai, jun luo, xiaoping wang, and hong wang. "Eap-based group authentication and key agreement protocol for machine-type communications". International journal of distributed sensor networks, 9(11):304-601. November 2013.
11. Sheetal kalra and sandeep k. sood. "Secure authentication scheme for iot and cloud servers". Elsevier, Pervasive and mobile computing, 24:210-223, december 2015.
12. Saru kumari, marimuthu karuppiah, ashok kumar das, xiong li, fan wu, and neeraj kumar. "A secure authentication scheme based on elliptic curve cryptography for iot and cloud servers". The journal of super computing, april 2017.
13. Deok-gyu lee, seo-il kang, dae-hee seo, im-yeong lee: "Authentication for single/multi domain in ubiquitous computing using attribute certification". Computational Science and Its Applications - ICCSA 2006 (4).
14. Nicolae Constantinescu and Claudiu Ionut Popirlan. "Authentication model based on multi-agent system" Mathematics and Computer Science Series. pp 11, 2011.
15. Sanae Hanaoui. "On the security communication and migration in mobile agent systems". Advanced Intelligent Systems for Sustainable Development (AI2SD'2018) pp 302-313.
16. B. hancock, Security views, computer & security. 18(7) (1999) 553-564.
17. Foundation for intelligent physical agents. "Pa agent management support for mobility specification", document number dc00087c.technical report Geneva, Switzerland, may(2002).
18. Mathematical routines for the nist prime elliptic curves,april 05, 2010. Available : http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.204.9073&rep=rep1&type=pdf
19. R. Hassan, M. Houssain, M. Fotouhi"Towards an Analysis of Security Issues, Challenges, and Open Problems in the Internet of Things" , IEEE 11th World Congress on Services. 2019-07-08.
20. Peter Reid "Is Elliptic Curve Cryptography (ECC) a step towards something more — Understanding ECC". Available : https://medium.com/@peterreid_12788

AUTHORS PROFILE

Amrani Ayoub, he received her Master's degree in computer engineering from University of Ibn Tofail, Kenitra Morocco. He is actually a PhD student in the IT laboratory, Computer and Telecommunications Research (LaRIT) in Kenitra-Morocco. His research interests include: information security, cryptography, multi-agent systems, distributed computing.

Pr. Rafalia Najat, she has obtained three doctorates in Computer Sciences at Mohammed V University, Rabat, Morocco by collaboration with ENSEEIHT, Toulouse, France, and at Ibn Tofail University, Kenitra, Morocco. Currently she is a professor at Ibn Tofail University, Department of Computer Sciences, Kenitra, Morocco. Her research interests are in distributed systems, multi agent systems, concurrent and parallel programming, communication, Security, Big data and Cloud Computing.

Pr. Abouchabaka Jaafar, he has obtained two doctorates in Computer Sciences applied to mathematics at Mohammed V University, Rabat, Morocco. Currently he is a professor at Ibn Tofail University, Department of computer Sciences, Kenitra, Morocco. His research interests are in concurrent and parallel programming, distributed systems, multi agent systems, Genetics Algorithms, Big Data and Cloud Computing.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)