Indonesian Journal of Electrical Engineering and Computer Science
Vol. 20, No. 3, December 2020, pp. 1460~1470
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v20.i3.pp1460-1470

A lightweight secure CoAP for IoT-cloud paradigm using elliptic-curve cryptography

Amrani Ayoub, Rafalia Najat, Abouchabaka Jaafar
LARIT, IT department, Faculty of Sciences Kenitra, IbnTofail University, Morocco

Article Info

Article history:

Received Mar 1, 2020
Revised May 16, 2020
Accepted Jun 1, 2020

Keywords:

Authentication
Avispa
CoAP
Elliptic curve cryptography
Internet of things

ABSTRACT

Cloud computing and the internet of things (IoT), two different technologies, are already part of our lives. Their impressive adoption increasing more and more, which makes them the future of the future internet. The tsunami of interconnectivity between objects and data collection is increasingly based on Cloud Computing, where data analysis and intelligence really reside. A new paradigm where the Cloud and the IoT are merged will create a new air in the world of technology, which can offer many services and applications useful to humanity. However, despite the great benefits that can bring this technology in term of new services, elasticity and flexibility, the security aspect still remains a serious constraint which hampers the expansion of this technology. This paper proposes a lightweight Mutual authentication protocol based on constrained application protocol (CoAP); that is suitable for IoT device than HTTP and using elliptic curve cryptography to secure data transmission between the Cloud and devices. We used AVISPA tool to verify our proposed scheme.

Corresponding Author:

Ayoub Amrani,
Department of Computer Engineering,
Ibn Tofail University,
242 University Road, kenitra, Morocco.
Email: amrani.ayoub@uit.ac.ma

1. INTRODUCTION

The Internet of things has become a common term in our society. By 2025, a hundred thousand objects will be connected to the Internet that communicates and share various massive data via limited resource to provide appropriate services to users, such as healthcare, smart home, transport, and smart gird. However, this large amount of data generated by these objects cannot be running by a single server environment.

Cloud computing is a distributed computing and large pool of resources for a wide-ranging of data [1]. It provides three essential services such as on-demand self-services especially data storage and computing power, rapid elasticity, ubiquitous network access, etc. This convergence between the Cloud and IoT will be welcome in order to create a homogeneous environment between the intelligibility of the objects and the robustness of the cloud, and it will bring many services that will be very useful to humanity. However, these services are facing a major security threat and are vulnerable to potential attacks because they are given through an Open channel. Mirai [2] one of the most known attacks, it's a powerful Distributed Denial of Service (DDoS), which is capable to create huge traffic using IoT devices. Surprisingly, there are other several malware that can target these devices. The IoT devices are easily compromised due to the low level of security installed. Therefore a secure and efficient scheme that guarantees confidentiality, mutual authentication, privacy, and integrity is a very essential security requirement for IoT-cloud environment. Most of the security protocols developed are based on Hypertext Transfer Protocol (HTTP) and using classical identification methods such as Rivest Shamir Adleman (RSA). HTTP is powerful and wide

used-protocol [3] but it's relatively expensive in terms of implementation code-space and network resource usage. RSA [4] is the most known cryptography algorithm, that uses public and private keys to secure communications between two entities, and for a strong complexity of the security protocol, the size of the key must be huge. Unfortunately, this cannot be handled by IoT devices because of the limitation of resources they have such as low storage capacity, computing capability, and limited battery.

This paper proposes a secure and lightweight mutual authentication scheme based on Constrained Application Protocol (CoAP), and using Elliptic curve cryptography (ECC) as a strong asymmetric cryptography method for IoT-Cloud environment considering computational costs.

1.1. Contribution

Our contributions in this paper are as follows:

a) A lightweight secure authentication scheme has been proposed Cloud of things, to secure lacks of existed scheme. Our proposed scheme guarantee mutual authentication and anonymity and resist to known attack such as session key disclosure, replay and impersonation attack.
b) A formal security analysis was performed using AVISPA to prove the validity of the proposed scheme against various attacks.

1.2. Organization

The rest of the paper is organized as follows. In Section 2 we introduce the related works. In Section 3 we will discuss the preliminaries of elliptic curve cryptography and CoAP. In Section 4, we propose a secure and lightweight mutual authentication scheme for IoT in a cloud computing environment to address the security lacks of the existed scheme. In section 5, performance analysis has been done. We verify in section 6 the proposed protocol using Automated Validation of Internet Security Protocols and Applications (AVISPA) tool. Lastly, we will conclude the section in 7 concludes.

2. RELATED WORKS

The security of hundreds of communication between IoT devices and Cloud computing requires significant resources such as storage capacity, processing, and energy. Unfortunately, the security schemes used nowadays to provide secure communication, either they uses classical cryptography methods that require huge computational resources or they are vulnerable to several attacks. Schemes developed for the IoT-Cloud paradigm can be ordered into schemes based on incorporating mutual authentication, key exchange with a trusted party, Secure-ID authentication, directed Path-based Authentication (DPAS), session key-based authentication schemes, and One-Time Password (OTP) based authentication.

Recently, to reduce the computational time for IoT devices, schemes based on the elliptic curve have been implemented. ECC has become the most used cryptography algorithm to design lightweight schemes. In 2009 Yang and Chang [5], based on Tian et al's authentication [6], developed a mutual authentication protocol with session key agreement. This method is very sweetly for IoT devices, it does not exhaust the resources of the objects, but unfortunately, this method is weak to offline password guessing, and clock synchronization [7]. Other protocols based on ECC have been proposed for IoT-Cloud by Granjal et al. [8], Ray et al. [9] and Jiang et al. [10] In 2015, a new protocol appeared, proposed by Kalra and Sood [11], they gained experience from the previous methods. They proposed a mutual authentication scheme to secure the transaction in a cloud of things using HTTP cookies, for a smart device that is HTTP clients. The use of cookies to develop mutual authentication for smart devices was very innovative. In 2017 Kumari et al. [12] after the analysis of their scheme showed that this algorithm is vulnerable against offline password guessing and insider attack. That is to say, this scheme does not provide device anonymity.

In the last few years, researchers have become more leaning to lightweight protocols for the communication on an intranet of things such as MQTT (Message Queuing Telemetry Transport) and CoaP instead of HTTP because they are suitable for IoT devices. In 2018 We developed [13] a lightweight secure mutual authentication based MQTT and using ECC, this scheme satisfies all the essential security requirements needed for this convergence. In 2019 Ankur et al. [14] proposed lightweight authentication and authorization scheme based on MQTT an ECC, unfortunately, this scheme doesn't expose the real role of the MQTT broker and how the distribution of keys was done. Firas et al. [15] designed an algorithm that implements CoAP using ECC in order to secure communication between IoT networks they simply used the ECDH (elliptic curve Diffie-Hellman) protocol, they generated private and public key to encrypt and decrypt messages. In [16] they proposed certificate key agreement protocol for IoT they replace DH and RSA certificate used by DTLS [17] (Datagram Transport Layer Security) by ECDH. In [18] they proposed a lightweight secure CoAP by compressing the standard security Protocol. They reduced the overhead of DTLS by 6LoWPAN header compressor.

3. PRELIMINARIES

As demonstrated in this document, the numbering for sections upper case Arabic numerals, then upper case Arabic numerals, separated by periods. Initial paragraphs after the section title are not indented. Only the initial, introductory paragraph has a drop cap.

3.1. Elliptic curve cryptography

In this section, we introduce the asymmetric cryptography method elliptic curves cryptography. Nowadays to maintain a good encryption strength we need a 3072-bit key size RSA instead of 1024 bit. RSA straight security required 1024-bit and 3072 bits for exceptionally valuable keys [19]. ECC with the shorter keys length can provide the same security level than RSA. Therefore the advantage of ECC over RSA is obvious. This small keys that offer ECC, allows us to design and implement strong and faster cryptographic schemes, which makes ECC a very attractive option for the limited resource-constraint device.

Introduction to elliptic curve (EC)

Elliptic curves are the simplest curves after straight lines and conics. We can define an elliptic curve E on a finite field denoted E(F) by its Equation to the form of Weierstrass:

$$E: y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \quad (1)$$

With $a_1, a_2, a_3, a_4\ et\ a_6\ \in F$

There are three main types of a finite field which can be used for the implementation of cryptography for elliptical curves: prime field, binary field, and extension field. The order of a finite field is the number of elements present in i. Count the number of points of a defined elliptic curve on a finite filed is one of the essential problems in the cryptography safe curves. In what follow we used the primer filed, with p>3. The Weierstrass equation for an elliptic curve E(F_p) on a finite prime field is represented with:

$$E: y^2 = x^3 + ax + b\ with\ a, b\ \in F_p \quad (2)$$

To be used in cryptography, the necessary condition is that the discriminant of the polynomial is equal to 0. This condition guarantees that, for any point on the elliptical curve, passes one and one single tangent.

$$f(x) = x^3 + ax + b, \Delta = 4a^3 + 27b^2 \neq 0 \quad (3)$$

This group consists of two basic operations: Point doubling (2P) and point addition (P+Q) where P and Q two different points in a curve.

Given $P = (x_p, y_p)$ et $Q = (x_q, y_q)$ two points $\neq \infty$ of an elliptic curve E(F_p) on a finite field . Geometrically, the point addition (P + Q), consists in taking the symmetric of the third point (P * Q) of intersection of the line PQ with the elliptic curve. The point doubling is the special case of addition where P = Q, we take the symmetric of the point of intersection of the tangent at P with the elliptical curve. If P and Q are symmetric relative to the x-axis, in this case the line PQ intersects the curve at the point at infinity (which is the zero of the group) and therefore Q = -P

The point addition and doubling can be calculated through the following equations and as shown in Figures 1 & 2:

$$\begin{cases} x_r = \lambda^2 - (x_p + x_q) \\ y_r = \lambda * (x_p - x_r) - y_p \end{cases}$$

$$\begin{cases} \lambda = \frac{y_p - y_q}{x_p - x_q}\ for\ point\ addition \\ \lambda = \frac{3x_p^2 + a}{2y_p}\ for\ point\ doubling \end{cases}$$

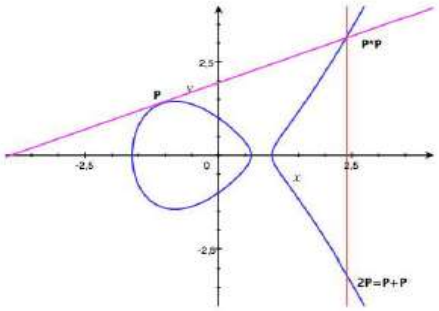

Figure 1. Point addition

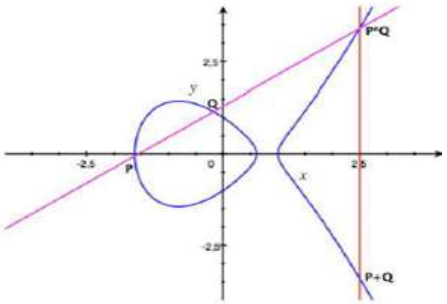

Figure 2. Point Doubling

3.2. Constrained application protocol

CoAP is a specialized web Protocol for use in an Intranet of things, that runs over the UDP (User Datagram Protocol), 3 it's designed for M2M (Machine to Machine) Applications, this protocol is a variant of the most used Protocol HTTP, that tries to get by with very limited resources, CoAP is designed to work on micro-Controllers and to use minimal resources, both on the network and the device. CoAP is based on Rest (Representational State Transfer), resources are available under a URL and Clients access these resources using GET, POST, PUSH, and DELETE methods. Figures 3 & 4 shows the CoAP header format and CoAP Transaction. Gives a brief detail description for CoAP header as shown in Table 1.

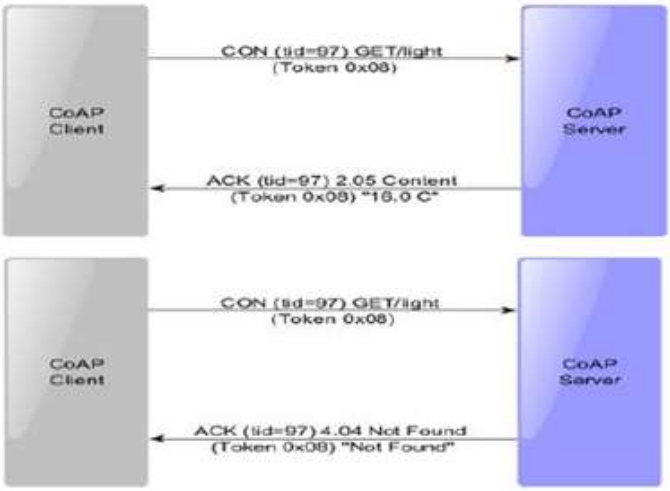

Figure 3. CoAP Header

Table 1. Gives a brief detail description for CoAP header

Header	Descriptive
ver	CoAP version
T	Type of CoAP message (Confirmable CON 0, Non-Confirmable NON 1, Acknowledgement ACK 2, Reset RST 3)
oc	Count Option

Figure 4. CoAP Transaction

3.3. Datagram transport layer security

DTLS is the primary secure protocol used to protect CoAP communications. It operates between the transport and the application layer. This protocol consists of two layer, the lower one that contains the record of the protocol and the upper one that contains the handshake, alerts and ChangeCipherSpec. To provide a strong security transmission, CoAP default choice of DTLS parameters is 3072-bit RSA key which unfortunately still huge for IoT device. Figure 5. shows the structure of a DTLS message in an IP/UDP datagram.

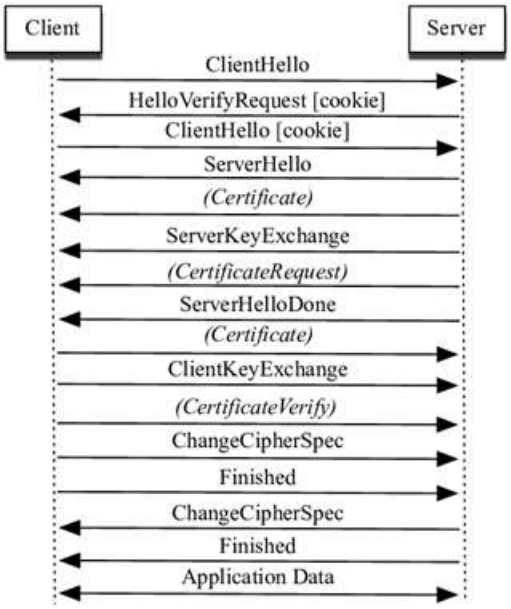

Figure 5. DTLS Handshake protocol

4. PROPOSED SCHEME

In this section, we propose a CoAP based authentication protocol for smart devices ECC cryptographic methods. We exposed in the previous section the known works to secure the transaction between an IoT device and the cloud. The idea of using CoAP for smart device authentication is novel. In this section, we propose a lightweight security mechanism based on CoAP. In this section, we discuss our proposed scheme in detail including the entities involved, system model, and mathematical formulation of the scheme.

In what follows, we will detail the proposed security scheme. Our mechanism consists of 3 parts as showen in Figure 6. The first is the initialization where the Server initiates the parameters necessary for the generation and distribution of keys. The second is the registration between the devices and the Server. And finally, is the login and authentication, it's where the IoT devices authenticated to the Cloud Server. Table 2 denotes the notations used in the proposed protocol.

Table 2. Notions used in this paper

Notation	Description
ID_i	Identity of device i
ID_c	Identity of Cloud Server
K_m	Random master key
r	Random number
Z_p	Finite field group
P	Large prime number of the order $> 2^{160}$
N_1, N_2	Random numbers generated for ECC parameters
G	Generator point of a large order n
H(.)	Cryptographic one-way hash function
K	Mutual auth key
C	Encrypted messag
$\oplus$ / $\|\|$	XOR operator // Concatenation

4.1. Initialization phase

In our design it is the Cloud Server who is in charge of the generation and distribution of keys. But before this step, the Cloud must initiates its parameters by choosing an elliptic curve equation.

$y^2 = x^3 + ax + b$ over $\mathbb{Z}_p$, where $\mathbb{Z}_p$ (p $>2^{160}$) is the finite field group. The Cloud selects two field elements $a, b \in \mathbb{Z}_p$. These two elements 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 it chooses random master secret key K_m from $\mathbb{Z}_p$ and computes public key $= G.K_m$. Finally broadcast the public parameters $K_p \leftarrow (E_p, G, p, L)$.

4.2. Registration phase

In order to register with the Cloud, the device send his $I_i \leftarrow h(ID_i \parallel rG)$

i. The Server generates K_i for every device $A_i \longrightarrow h(K_m \parallel K_i)$, where $K_i = h(N \parallel I_i \parallel ID_c)$, and stores $A_i' = A_i.G$ on the Device. The Cloud then computes the security parameters $K_s = K_m \oplus K_i$, $B_i = h(K_s \parallel I_i \parallel A_i')$ and stores $B_i' = B_i.G$, K_i corresponding to the identity ID_i of the Device i in its database.
ii. Cloud → Device: K_i , A_i'.

4.3. Login & Authentication phase

i. The Device Calculates ECC point $P_1 = N_1.G$ in order to login with the Cloud Server, then computes the $P_2 = h(N_1.A_i')$.
ii. The device sends P_1, P_2 and K_i to the Server. This last computes $A_i \longrightarrow h(K_m \parallel K_i)$ by calculating the $K_m = K_s \oplus K_i$. The Cloud then checks whether the value of
iii. $P_2^* = h(P_1.A_i)$ is equal to the received one of P_2.
iv. The Cloud checks P_2^* ?= P_2. Then selects a random number N_2, and calculates 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 Device.
v. The device computes $B_i = h(K_s \oplus I_i \oplus A_i')$ and calculate ECC point $P_4^* = P_3.B_i$, and compares the value of P_4^* with the received one of P_4.
vi. The device checks P_4^* ?= P_4, then computes $V_i = h(P_2 \parallel K)$, where $K = N_1.P_3$ and sends V_i to the Server. The Cloud 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 to authenticate the device.
vii. The Cloud checks V_i^* ?= V_i, after mutual authentication between the Ddvice and the Cloud Server.

5. SECURITY ANALYSIS

In this section, we give a informal security analysis to confirm the validity of the proposed scheme against various attacks.

5.1. Informal Security Analysis

Our proposed security scheme can hold against various types of attacks:

a) **Anonymity:** The intruder cannot obtain the identity IDi of the device because it is masked by using hash function $I_i = h(ID_i \parallel rG)$. In addition, the adversary cannot obtain random nonce r. Consequently, our scheme provides anonymity.
b) **Impersonation Attack:** When an adversary MA assumes the identity of a legitimate User. MA must successfully generate $I_i = h(ID_i \parallel rG)$. However, this can not be computed by the MA because he cannot obtain the random nonce r of the user. Therefore our scheme is well protected against the Impersonation attack.
c) **Replay Attack:** When an intruder MA may attempt to impersonate legal user eavesdropping the communication and intercepts the data and re-transmits it. This cannot be happened because the Cloud Server checks whether P_2^* ?= P_2 and the device checks whether P_4^* ?= P_4 therefor, our scheme is well protects against replay attack.
d) **Mutual Authentication:** After receiving $\{P_1, P_2, K_i\}$ from the device, the CS checks whether P_2^* ?= P_2, if it's correct the cloud authenticates the device. The same way, the device receives $\{P_3, P_4, K_s\}$ checks whether P_4^* ?= P_4, then the device authenticate the CS.Hence our scheme achieve mutual authentication among CS ans Device.
e) **Session Key Attack:** The intruder MA cannot computes the session key K because, there is noway that he can obtain the nonces of the Server and Cloud N1 & N2. In addition MA cannot computes K. Therefor, our scheme withstands the session key attack.

Brute Force: So that the intruder can deploy a brute force attack, he needs to gain the security parameters $\{P_1, P_2, P_3, P_4\}$ from the messages exchanged between the device and the CS, even he processes these parameters he cannot compute r.G. Therefore, our scheme can resist the brute force attack.

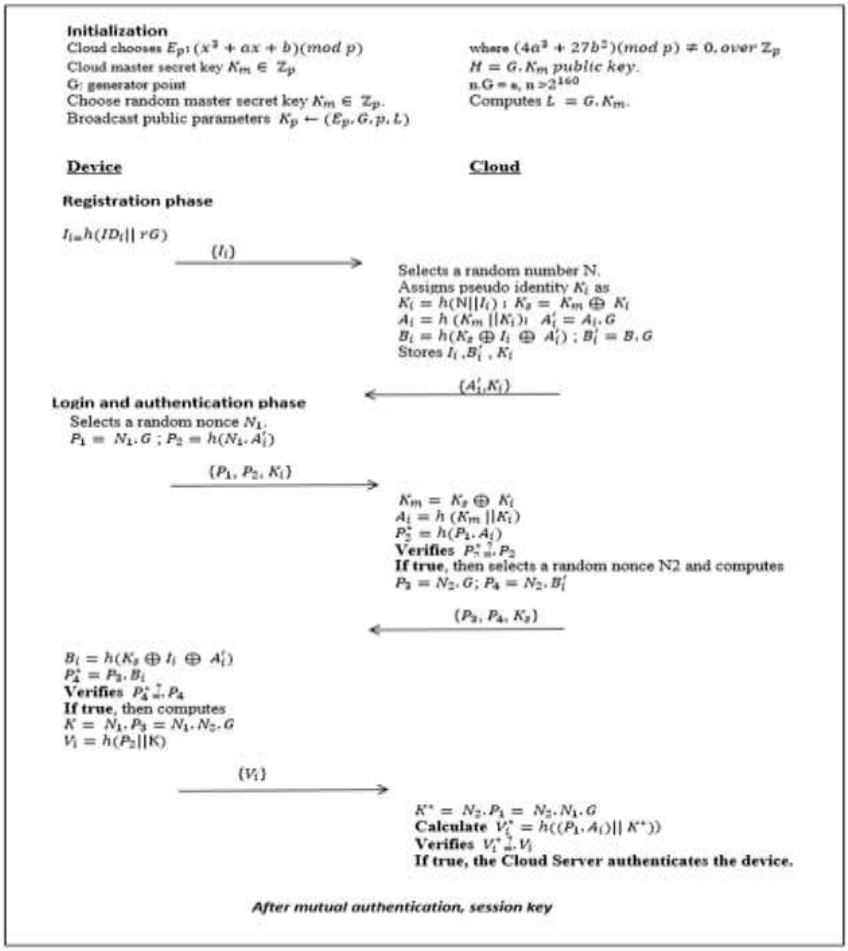

Figure 6. Mutual authentication scheme

5.2. Security feature

The proposed scheme give a better security levels compared to firas et als and other existing scheme.Table 3 shows the analysis results of the security features.

Table 3. Security Features comparison

Security feature	Firas et al.	Ours
Anonymity	X	O
Impersonation Attack	X	O
Replay Attack	O	O
Session Key Attack	X	O
Mutual Authentication	X	O
Brute Force	X	O

o , preserves the security features; x, does not preserve the security features.

6. SECURITY VERIFICATION WITH THE AVISPA TOOL

AVISPA [20] is a push-button tool for the Automated Validation of Applications and Internet Security Protocols.It was developed in 2004 by Basin et Al as part of a European project.It is an automatic analysis tool intended to assist in the validation of security protocols. He owns two major challenges: to be

efficient, while ensuring accessibility to non-specialists in the field. The tool can be used on small and medium scale protocols (as provided in the Clark / Jacob library), as well as Internet security protocols at large scale.

The protocols are implemented in High-Level Protocol Specification Language (HLPSL), which is a formal role based language using which data structures, cryptographic operators,control flow patterns and algebraic properties can be specified. The specification is rewritten in an IF intermediate format using a translator. AVISPA uses 4 other tools (back-ends) which take the IF format as input, and which offer the possibility of 4 different analyzes of the same protocol. Any language convertible to IF intermediate format can be verified by AVISPA.

6.1. HLPSL specifications

We considered two basic roles: Device and Cloud. Then, we give the session and environment role using HLPSL in Figure 9 & 10, which contains the security goals. The role specifications of device and CS are shown in Figures 7 & 8.

6.2. AVISPA simulation result

The proposed scheme is analyzed using CL-AtSe and OFMC back-ends [20]. The intruder has full control over the network in such a manner that all messages sent by the agents corresponding to the roles are available to the intruder. The results of the execution shown in Figure 11 and Figure 12. Under this model, the intruder may intercept, analyze or/and modify messages as long as he knows the required keys. The attacker can act as any entity and send any message to some other agent participating over the communication channel. The simulation results show that our scheme is secure

7. PERFORMANCE ANALYSIS

In this section we give a brief comparison between the proposed scheme and some existed mutual authentication schemes. We compared the communication cost and the computational cost of this schemes.

```
role device (Dev, CS: agent, SKedics: symmetric_key, H : hash_func, SB, RB:
channel(dy))
played_by Dev
def=
local State: nat,
    Idi , Ii, Ki, Id_C, Ks, Km: text,
    G,R, N1, P1, P2, N2,N, SK, Vi: text,
    F : hash_func
const sec1, sec2, alice_bob_r1, bob_alice_r2 : protocol_id
init State := 0
transition
%%%%%%% Registration phase
1. State = 0 /\ RB(start) =|>
          State' := 1 /\ R' := new()
                      /\ Ii' := H(Idi.F(R'.G))
                      /\ secret({Idi,F(R'.G) }, sec1, Dev)
%Send the registration request (Ii) to the CS securely
%via secure channel
                      /\ SB({Ii'}_SKedics)
%Receive registration reply { A'I,Ki } from CS
%via secure channel
2. State = 1 /\ RB({H(N'.Id_C.H(Idi.F(R'.G))).F(H(Km'.H(N'.Id_C.H(Idi.F(R'.G) ) )
).G)}_SKedics) =|>
   State' := 3 /\ secret({Id_C}, sec2, Dev)
%Login and authentication phase
     /\ N1' := new() /\ P1' := F(N1'.G)
        /\ P2' := H(N1'. F(H(Km'.H(N'.Id_C.H(Idi.F(R'.G) ) ).G)))
%Send login request {P1,P2,Ki} to CS via open channel
     /\ SB(P1'.P2'.H(N'.Id_C.H(Idi.F(R'.G) ) ) )
     /\ witness(Dev, CS, alice_bob_r1, N1')
%Receive {P3,P4,Ks} from CS via open channel
3. State = 3 /\ RB( F(N2'.G). % P3
          F(N2'.F(H(xor(
  xor(
      xor(Km', H(N'.Id_C.H(Idi.F(R'.G) ) ) ),%Ks
  H(Idi.F(R'.G) )),%Ii
                          F(H(Km'.H(N'.Id_C.H(Idi.F(R'.G) ) ) ).G )%A'i
     ).G)%B'i
  ) %P2
.xor(Km', H(N'.Id_C.H(Idi.F(R'.G) ) ) ))=|>

          State' := 5 /\ SK' := F(N1.F(N2'.G))
                      /\ Vi' := H(F(N1.F(H(Km'.Ki).G)).SK')
% Send message {Vi} to CS via open channel
                      /\ SB(Vi')
% Dev's acceptance of random nonce r2 generated for Dev by CS
                      /\ request(CS, Dev, bob_alice_r2, N2')
end role
```

Figure 7. Device Role

```
role cloud (Dev, CS: agent,
                SKedics: symmetric_key,
                H : hash_func,
                SB, RB: channel(dy))
played_by CS
def=
local State: nat,
   Idi, Id_C, Km, Ai, Aii : text,
   Ki,R, G, N1, N2,N, P3, P4: text,
   Bi, Bi1, Ks : text,
   F : hash_func
const sec1, sec2, alice_bob_r1, bob_alice_r2 : protocol_id
init State := 0
transition

1. State = 0 /\ RB({H(Idi.F(R.G))}_SKedics) =|>
        State' := 2 /\ N' := new() /\ secret({Idi,F(R.G)}, sec1, Dev)
                    /\ Km' := new()
                    /\ Ki' := H(N'.Id_C.H(Idi.F(R.G)))
                    /\ Ai' := H(Km'.Ki')
                    /\ Aii' := F(Ai'.G)
                    /\ secret({Id_C}, sec2, Dev)
                    /\ SB({Ki'.Aii'}_SKedics)
2. State = 2 /\ RB(F(N1'.G). %P1
  F(N1'. F(H(Km'.H(N'.Id_C.H(Idi.F(R.G) ) ) ).G)). %P2
  H(N'.Id_C.H(Idi.F(R.G))) ) =|>%Ki
              State' := 4 /\ N2' := new() /\ P3' := F(N2'.G)
                  /\ Bi' := H(xor(
   xor(
       xor(Km', H(N'.Id_C.H(Idi.F(R.G) ) ) ),%Ks
   H(Idi.F(R.G) )),%Ii
                                  F(H(Km'.H(N'.Id_C.H(Idi.F(R.G) ) ) ).G )%A'i
                                  ).G)
                    /\ Bi1' := F(Bi'.G)
                    /\ P4' := F(N2'.Bi1')
                    /\ Ks' := xor(Km', H(N'.Id_C.H(Idi.F(R.G) )))%Km+Ki
                    /\ SB(P3'.P4'.Ks')
                    /\ witness(CS, Dev, bob_alice_r2, N2')
3. State = 4 /\ RB(H(F(N1'.F(H(Km'.Ki).G)).F(N1'.F(N2'.G)))) =|>
% CS's acceptance of random nonce r1 generated for CS by Dev
  State' := 6 /\ request(Dev, CS, alice_bob_r1, N1')
end role
```

Figure 8. Cloud Role

```
role environment()
def=
const dev, cs    : agent,
skedics          : symmetric_key,
h, f             : hash_func,
p1, p2, ki, p3, p4, ks, vi: text,
sec1, sec2, alice_bob_r1, bob_alice_r2: protocol_id
intruder_knowledge = {dev, cs, h, f,
p1, p2, ki, p3, p4, ks, vi}
composition
  session(dev, cs, skedics, h)
/\ session (i, cs, skedics, h)
/\ session (dev, i, skedics, h)
end role

goal
secrecy_of sec1
secrecy_of sec2
authentication_on alice_bob_r1
authentication_on bob_alice_r2
end goal
environment()
```

Figure 9. Environment role

```
role session (Dev, CS: agent,
SKedics : symmetric_key,
H       : hash_func)
def=
local SB1, RB1, SB2, RB2: channel(dy)
composition
device (Dev, CS, SKedics, H, SB1, RB1)
/\ cloud (Dev, CS, SKedics, H, SB2, RB2)
end role
```

Figure.10 Session rôle

```
% OFMC
% Version of 2006/02/13
SUMMARY
 SAFE
DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
 /home/span/span/testsuite/results/Test_algo_these.if
GOAL
 as_specified
BACKEND
 OFMC
```

Figure 11. Output of OFMC back

```
SUMMARY
 SAFE

DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
 TYPED_MODEL

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

GOAL
 As Specified

BACKEND
 CL-AtSe
```

Figure 12. Output of ATSE backend

7.1. Communication cost

Before we compute the total communication costs for our scheme, we assume that the identity ID, Nonce's N, Random number R and all security parameters are 224 bit long. The output for a hash function is 256 bits (sha-2) and the elliptic curve cryptosystem is ECC - 256 bits.

The transmitted messages in the proposed scheme:

a. $\{P_1, P_2, K_i, V_i\}$ = {256 + 256 + 256 +256} = 4*256 = 1024 bits.
b. $\{P_3, P_4, K_s\}$ = {256 +256 +256} = 3*256 = 768 bits.

The total communication cost is 1792 bits. Table 4 resume the analysis results of communication cost.

Table 4. Summary of communication cost

Schemes	Total Cost
Zhou et al.[21]	4352 bits
Xue et al.[22]	3840 bits
Amin et al.[23]	3456 bits
Ours	1792 bits

7.2. Computational cost

We compared the computation overheads of our scheme with some discussed schemes [21-23]. We consider using the following parameters, to analyze computation cost. Table 5 shows the detail total cost and the analysis results of computation cost as below.

a) T_h denotes the time for the hash function.
b) T_{tem} The computational cost of a ECC point.
c) We ignore XOR operation, because it is negligible compared to the other operations.
d) multiplication operation

Table 5. Summary of computational cost

Schemes	Device	CS	Total
Zhou et al.	$13T_h$	$30T_h$	$43T_h$
Xue et al.	$12T_h$	$24T_h$	$36T_h$
Amin et al.	$12T_h$	$18T_h$	$30T_h$
Ours	$4T_h$ +$4T_{tem}$	$5T_h$+$4T_{tem}$	$9T_h$ +$8T_{tem}$

8. CONCLUSION

IoT-cloud paradigm bring many application and services to humanity in different fields such as healthcare, transportation and logistics, smart environment as well as personal and social domain. The only way that these objects can be effective and productive in large scale and with a safe use is to provide a strong secure protocol to protect the entities involved also to guarantee the integrity and the confidentiality of transmitted information.

In this paper we proposed a lightweight secure scheme based on the light-weightiness of the the ECC, to reduce the size of the key to save the resources of the IoT devices and not be exhausted. We also incorporates the concept of CoAP to reduce the overhead and also for smooth communication int IoT-Cloud environment.We presented a formal security analysis of the proposed protocol using AVISPA. Furthermore, we give a brief comparison of the security performance and features of our scheme with some already existed. We've proven that that our protocol provides better efficiency and safety than related schemes. Therefore, our scheme can be suitable for practical IoT-Cloud environment. As future work, we are working on a comparison between the scheme that we've developed using MQTT & ECC [13] and this proposed scheme based on CoAP&ECC to define which one is suitable for IoT-Cloud paradigm.

REFERENCES

[1] B. Grobauer, T. Walloschek, E. Stocker,"Understanding cloud computing vulnerabilities" *IEEE Security & Privacy*, vol. 9, pp. 50-57, 2010.
[2] M. Zane, A. Manos, et al., "Understanding the Mirai Botnet", *26th USENIX Security Symposium,* pp. 1093-1110, 2017.
[3] C Bormann, AP Castellani, Z Shelby, "CoAP: An Application Protocol for Billions of Tiny Internet Nodes", *IEEE Internet Computing,*vol 16, no. 2, pp. 62-67, 2012.
[4] P. Mahajan., A. Sachdeva, "CA study of encryption algorithms AES, DES and RSA for security", *Global Journal of Computer Science and Technology*, vol. 13, 2013.
[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", *Computers & security-elsevier*, vol 28, no. 3&4, pp. 138-143, 2009.
[6] Xiaojian Tian, D.S. Wong, and R.W. Zhu, "Analysis and improvement of an authenticated key exchange protocol for sensor networks", 2005 *IEEE Communications Letters,* vol 9, no. 11, pp. 970-972, 2005.
[7] W. Ding, M. Ying, M. Chunguang, and C. Zhen-shan, "Comments on an Advanced Dynamic ID-Based Authentication Scheme for Cloud Computing", *2012 International Conference on Web Information Systems and Mining* , pp. 246-253, 2012.
[8] G. Jorge, M. Edmundo and S. Jorge, "End-to-end transport-layer security for Internet-integrated sensing applications with mutual and delegated ECC public-key authentication.", *2013 IFIP Networking Conference,* pp. 1-9, 2013.
[9] R. Sangram, and G P. Biswas, "Establishment of ECC-based Initial Secrecy Usable for IKE Implementation", *2012 World Congress on Engineering,* vol. 1, 2012.
[10] J. Rong, L. Chengzhe, L. Jun, W. Xiaoping, and W. Hong, "EAP-Based Group Authentication and Key Agreement Protocol for Machine-Type Communications", *2013 International Journal of Distributed Sensor Networks,* pp. 2-14, 2013.
[11] K.Sheetal, and K. Sood, *"Secure authentication scheme for IoT and Cloud servers", Pervasive and mobile computing*, vol. 24, pp. 210-223, 2015.

[12] K. Saru, K. Marimuthu, K. Ashok, L. Xiong, W. Fan, and K. Neeraj, "A secure authentication scheme based on elliptic curve cryptography for IoT and Cloud servers", *The Journal of Supercomputing,*vol. 74, pp. 6428-6453, 2017.
[13] A. Amrani; N. Rafalia, J. Abouchabaka, "lightweight secure scheme for IoT-Cloud convergence based on elliptic curve", *Journal of Theoretical and Applied Information Technology,* vol. 96, no. 1, pp. 144-155, 2019.
[14] L. Ankur, Karambir, "ECC based inter-device authentication and authorization scheme using MQTT for IoT networks", *Journal of Information Security and Applications*, vol. 46, pp. 1-12, 2019.
[15] A. Firas, A. Ammar, A. Majd, A. Omar, " Security-aware CoAP Application Layer Protocol for the Internet of Things using Elliptic-Curve Cryptography", *The International Arab Journal of Information Technology*, vol. 15, pp. 550-558, 2018.
[16] R. Shahid, S. Hossein, H. Kasun, H. René, and V. Thiemo, "Lithe: Lightweight Secure CoAP for the Internet of Things", *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3711-3720, 2013.
[17] R. Shahid, T. Daniele, V. Thiemo, "6LoWPAN Compressed DTLS for CoAP" ", *2012 IEEE 8th International Conference on Distributed Computing in Sensor Systems,* pp. 287-289, 2012.
[18] Geoff Mulligan, "The 6LoWPAN Architecture", *2007 the 4th workshop on Embedded networked sensors,* pp.87.82, 2007.
[19] B. Hancock. "Security views". *COMPUTERS & SECURITY,* vol. 18, no. 7, pp. 553-564.
[20] The AVISPA Team. (2003) AVISPA. Deliverable 2.1: The High-Level Protocol Specification Language, 2003.
[21] L. Zhou, X. Li, K.H Yeh, C. Su, W. Chiu, "Lightweight IoT-based authentication scheme in cloud computing circumstance", *Future generation computer system,* vol. 91, pp. 244-251, 2019.
[22] K. Xue, P. Hong, C.A Ma, "Lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables formulti-server architecture", *Journal of Computer and System Sciences*, vol. 80, no. 1, pp. 195-206, 2014.
[23] R. Amin, N. Kumar, G.P Biswas, R. Iqbal, V.A Chang, "lightweight authentication protocol for IoT-enabled devices in distributed cloud computing environment", *Future generation computer system,* vol 78, pp. 1005-1019, 2018.
[24] K. Yokoyama, M. Yasuda, Y. Takahashi and J. Kogure, ""Complexity Bound on Semaev's Naive Index Calculus Method for ECDLP", *Number-Theoretic Methods in Cryptology 2019*.

BIOGRAPHIES OF AUTHORS

Ayoub amrani. He received his Master's degree in computer engineering from University of Ibn, Tofail, Kenitra Morocco. He is currently 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

Najat Rafalia. 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

Jaafar Abouchabaka. 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