Enhanced Mutual Authentication Scheme for Cloud of Things

Bhuvaneshwari S\textsuperscript{1}, Anantha Narayanan V\textsuperscript{2},

\textsuperscript{1,2} Dept. of Computer Science and Engineering, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, India

\textsuperscript{1}cb.en.p2cse16007@cb.students.amrita.edu, \textsuperscript{2}v_ananthanarayanan@cb.amrita.edu

Abstract. In recent years, technologies play a vital role in day to day activities, of which cloud computing and Internet of Things form a primary role. The integration of these two technologies would be a huge advancement wherein there is a large pool of embedded devices communicating with the cloud. However, there are lot of issues when we go for this integration. One of the main concerns is security, where device authentication and data privacy plays a vital role. In this paper, a mutual authentication scheme using Elliptic Curve Cryptography has been proposed. The proposed scheme has been analyzed against the work of Kalra et al. and the possible drawbacks with resolutions have been presented. The proposed scheme is verified using AVISPA and is proved to be strong against several attacks which prove it is more secure, powerful and efficient.

Keywords: Authentication, Cloud Computing, Elliptic Curve Cryptography, Internet of things.

1 Introduction

Internet of things (IoT) includes not only physical objects that are embedded with software and sensors but also includes humans in the network who are connected with these objects. IoT exchanges data among them to achieve a common goal in a smart way [1]. In recent years IoT has embraced different technologies and services in different fields such as healthcare, automation, manufacturing and household making it smart by undergoing multi fold advances. Unquestioningly the strength of IoT has been increased because of its involvement in everyday activity and since they operate without any human intervention.

The large scale of IoT projects has been doubled in the past 12 months as the number of organizations using IoT has increased with more than 50 thousand connected devices. These devices produce data which could be analyzed using cloud technologies, which has infinite storage and processing power. One of the biggest challenges in IoT is storage of multi-source high heterogeneity data with low semantics, so we move forward towards combining IoT with Cloud [2].

The users of cloud can be relaxed on the management, maintenance and storage of resources and in terms of cost, its pay-as-you-use policy has been an advantage. The usage of public cloud for IoT usage has been on increase for the past 2 years. Cloud
The trend shows more usage in future as there is a constant increase in the usage of embedded things which provides huge volumes of data.

![Cloud of Things](image)

**Fig. 1. Cloud of Things**

The two different worlds, cloud and IoT complement each other well and has been used to obtain a lot of benefits in different applications [3][4][5]. Cloud can give an effective solution when it comes to management and also build real-time applications [19] and assist in the processing of the data produced [6]. This integration of cloud computing and IoT is known as “Cloud of Things”.

On integration, the cloud will act as a layer in the middle of the things and the applications cover the complexities in its functionalities to realize them. The cloud offers connectivity through different means such as IoT applications in mobiles to access the data stored in it.

Security is a major concern when it comes to integration of these two technologies, where authentication plays a major role. There is a possibility of a lot of attacks on the cloud side causing an issue in data privacy. There are different encryption techniques used to address data integrity, data confidentiality and authenticity [7].

The embedded devices which will act as an HTTP (Hyper Text Transfer Protocol) client can communicate to a cloud server (an HTTP server) as shown in Fig. 1. Embedded devices can access data storage and computational facilities from the cloud storage. Here mutual authentication of the device and the cloud plays an important role, before further processing. Since the embedded devices are constrained in all categories, an asymmetrical cryptographic technique called Elliptic Curve Cryptography (ECC) is recommended [8].

The rest of the paper is organized as follows. Section 2 deals with related works, followed by the proposed method in Section 3. Section 4 and 5 provides the security analysis and cost analysis respectively. The concluding remarks are presented in Section 6.
2 Related Works

Recent years have seen a lot of authentication schemes being proposed, which uses integration of IoT and cloud technologies. Table 1 gives a summary of the most relevant techniques to our work. The table also shows the different vulnerabilities for each of the authentication schemes.

Table 1. Authentication Schemes

<table>
<thead>
<tr>
<th>Proposed Methodology</th>
<th>Authentication Scheme</th>
<th>Description and vulnerabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wu et al. [9]</td>
<td>ID-based remote authentication with smart cards on open distributed system from elliptic curve cryptography</td>
<td>All the client information is warehoused in the server, so that only the server can authenticate the client. This leads to lot of threats such as man in the middle and insider attack.</td>
</tr>
<tr>
<td>Tian et al. [10]</td>
<td>Analysis and improvement of an authenticated key exchange protocol for sensor networks</td>
<td>They use communal authentication using certificates which upsurge the cost.</td>
</tr>
<tr>
<td>Abichar et al. [11]</td>
<td>A fast and secure elliptic curve based authenticated key agreement protocol for low power mobile communications</td>
<td>Certificate based authentication scheme is used, which is not that cost effective.</td>
</tr>
<tr>
<td>Yang et al. [12]</td>
<td>An id-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem</td>
<td>ECC centered authentication aimed at smart devices is proposed.</td>
</tr>
<tr>
<td>Moosavi et al. [14]</td>
<td>An elliptic curve-based mutual authentication scheme for RFID implant system</td>
<td>ID verifier transfer protocol using ECC is proposed.</td>
</tr>
<tr>
<td>Liao et al. [15]</td>
<td>A secure ECC-based RFID authentication scheme integrated with ID-verifier transfer protocol</td>
<td>An authentication method centered on ECC for RFID is proposed.</td>
</tr>
<tr>
<td>Muhamed et al. [16]</td>
<td>A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks based on the Internet of Things notion</td>
<td>Authentication along with key agreement for heterogeneous system has been proposed.</td>
</tr>
<tr>
<td>Persson et al. [17]</td>
<td>Merging cloud and IoT</td>
<td>A model for unification IoT and cloud in an incorporated framework is presented.</td>
</tr>
<tr>
<td>Sood et al. [18]</td>
<td>Dynamic identity-based single password anti-phishing protocol</td>
<td>Single password anti-phishing method using cookie is presented.</td>
</tr>
</tbody>
</table>
3 Proposed Method

The proposed method provides an enhancement of the necessary security features for mutual authentication of Cloud of Things platform. In the proposed methodology, the mutual authentication scheme runs with HTTP cookies. There are 3 phases designed, namely, Registration, Pre-computation and log in, and Authentication. The whole scheme here is centered on the ECC, which is present in the Initialization phase [8]. Fig. 2 shows the proposed authentication scheme and Table 2 shows the different representations used in the proposed protocol. This section briefs the proposed algorithm.

Table 2. Representations used in the protocol

<table>
<thead>
<tr>
<th>Device</th>
<th>Embedded device</th>
</tr>
</thead>
<tbody>
<tr>
<td>D_IDi</td>
<td>Identity of the device</td>
</tr>
<tr>
<td>Cloud_server</td>
<td>Cloud server</td>
</tr>
<tr>
<td>Pwi</td>
<td>Password of the device</td>
</tr>
<tr>
<td>RS</td>
<td>First random number produced in the server</td>
</tr>
<tr>
<td>ran1, ran2</td>
<td>Random numbers used in ECC</td>
</tr>
<tr>
<td>Hash()</td>
<td>One-way hash function</td>
</tr>
<tr>
<td>X</td>
<td>Server’s private key</td>
</tr>
<tr>
<td>G</td>
<td>Base point of ECC</td>
</tr>
<tr>
<td></td>
<td>Concatenation</td>
</tr>
<tr>
<td>E_p</td>
<td>Finite field group</td>
</tr>
<tr>
<td>Cookie</td>
<td>HTTP cookie information</td>
</tr>
<tr>
<td>SK</td>
<td>Session key</td>
</tr>
<tr>
<td>⊕</td>
<td>XOR operation</td>
</tr>
<tr>
<td>EXP_TIME</td>
<td>Cookie’s expiration time</td>
</tr>
</tbody>
</table>

3.1 Registration phase

Step R1: For recording with Cloud_server, the client directs a unique D_IDi and Pwi password to the server. Here the embedded device used is considered to be intruder proofed so the password is created by itself. The embedded device directs the password to the Cloud_server as a hashed value N=Hash(D_IDi|Pwi).

Step R2: The server generates random number RS and with private key of server, it computes Bi,

\[ Bi = RS \oplus \text{Hash}(X) . \]

and security parameters,

\[ \text{Cookie} = \text{Hash} (B_i|X|\text{EXP\_TIME}|N) \]

\[ \text{Cookie'} = \text{Cookie} \ast G \]

\[ Ai = \text{Hash} (B_i \oplus N \oplus \text{Cookie'}) \]

Step R3: Cookie’ is passed from the server to the device and is stored there.
3.2 Login phase

Step L1: The device generates a random number ran1 and uses the generator point G, Num1 = ran1 * G.

Step L2: Num2 = Hash (ran1 * Cookie') is calculated with already stored Cookie' value.

Step L3: {N, Num1, Num2} is passed from the device to the server.

Step L4: The server computes Num2' = Hash (Num1 . Cookie) with the values stored in it.

Step L5: In server, if Num2 == Num2'

Then it selects a random number, ran2
Calculates, Num3 = ran2 * G * RS
Num4 = Num3 * Ai

Step L6: Server passes {Num3, Num4, Ti} to the device.

---

**Fig. 2. Proposed authentication scheme**
### 3.3 Authentication phase

**Step A1:** The device calculates $A'_1$ with already stored value from the previous phase.

$$A'_1 = \text{Hash}(\text{Bi} \oplus \text{Cookie'} \oplus \text{Hash (D_IDi|Pwi)})$$

**Step A2:** $\text{Num4}' = \text{Num3} \ast A'_1$ is calculated by the device.

**Step A3:** In the device, if $\text{Num4} = \text{Num4}'$ is true,

Then it calculates

$$S = \text{ran1} \ast \text{ran2}$$

$$V = \text{Hash}((\text{Num1} \ast \text{Cookie'})|S)$$

**Step A4:** The calculated $V$ is passed to the server from the device.

**Step A5:** In the server, $S = \text{Num1} \ast \text{ran2} \ast \text{RS}$ is calculated. Using the new calculated $S$, the server computes $V' = \text{Hash}((\text{Num1} \ast \text{Cookie'})|S)$

**Step A6:** In server, if $V = V'$ is true, then

Session key $SK = \text{Hash}(X|S)$ is generated and shared with the device.

Else,

Reject the device.

### 4 Security Analysis

The proposed work is simulated using HLPSL in AVISPA. The simulation is used to demonstrate that the proposed method is secure against various threats. Figures 3-5 show the HLPSL code developed for simulation. The different threats observed in mutual authentication are:

```hlpsl
begin
  role device, Cloud_server, agent, KEx, symmetric_key, Message_func, RS, channel_id;

  struct HLPSL
  
  3.3 Authentication phase

  Step A1: The device calculates $A'_1$ with already stored value from the previous phase.

  $A'_1 = \text{Hash}(\text{Bi} \oplus \text{Cookie'} \oplus \text{Hash (D_IDi|Pwi)})$

  Step A2: $\text{Num4}' = \text{Num3} \ast A'_1$ is calculated by the device.

  Step A3: In the device, if $\text{Num4} = \text{Num4}'$ is true,

  Then it calculates

  $S = \text{ran1} \ast \text{ran2}$

  $V = \text{Hash}((\text{Num1} \ast \text{Cookie'})|S)$

  Step A4: The calculated $V$ is passed to the server from the device.

  Step A5: In the server, $S = \text{Num1} \ast \text{ran2} \ast \text{RS}$ is calculated. Using the new calculated $S$, the server computes $V' = \text{Hash}((\text{Num1} \ast \text{Cookie'})|S)$

  Step A6: In server, if $V = V'$ is true, then

  Session key $SK = \text{Hash}(X|S)$ is generated and shared with the device.

  Else,

  Reject the device.

Fig. 3. HLPSL code for role specification of device
```
Insider attack: This attack is impossible because the Cloud_server does not have the device details as the device identity and the password is hashed and before it is passed to the server from the intruder proofed device.

Replay attack: Here the intercepted messages are retransmitted to perform the attack but in the system proposed, if an attacker tries and intercept the message \{Num1, Num2, N\} which is transmitted from D_IDi to Cloud_server. But even if the login phase computation continues the device will not be authenticated in the authentication phase where the verification takes place only if Num4 == Num4', the device proceeds to calculate S=ran1*ran2 using which the mutual authentication variable V = Hash ((Num1*Cookie')|S) is computed but when it is passed to the server side it is again verified with the values stored in the server only if I holds true the device would be authenticated. Hence a false device cannot compute the V and SK session key. So the replay attack is impossible.

Man-in-the-middle attack: Since here there is joint authentication among the D_IDi and Cloud_server there can be no possibility of Man-in-the–middle attack.

Fig. 4. HLPSL code for role specification of server
Fig. 5. HLPSL code for role specification of session, goal and environment

- Brute force attack: Only if the invader is able to take out all the security constraints \{Bi,Num1,Num2,Num3,Num4\} calculated, he can test with the brute force method but even then guessing the password Pwi and also the random numbers ran1 and ran2 is impossible and to access the attacker should know \{X\}, so brute force attack cannot be performed.
- Cookie-theft attack: Embedded device D_IDi is intruder proofed, the cookie stored in the smart device cannot be stolen.
- Offline password guessing attack: Hence the device id and password are only hashed and passed to the seer, there is no way to perform this attack, even if password would be guessed the attacker will not be able to find the identity of the device.
4.1 System Security Requirements

In order to propose an authentication scheme or any security measures, there are few security requirements that need to be achieved.

- Mutual authentication: The device and server is mutually authenticated using the variable $\text{Num4}$ and also $V$, which is exchanges during authentication phase only if it is valid then the $SK$ is calculated and shared else the device is rejected.
- Confidentiality: During all the transmission no important variable is directly transmitted to the other side without hashing and even throughout the communication the device details and the private key of the server is kept very confidential.
- Anonymity: The illegitimate server cannot know any device details in the proposed scheme as because the password and identity of the device are always hashed before exchanging any messages between the server and the device.

4.2 Comparison with Kalra et al.

Recently, Kalra et al. presented a mutual authentication method in order to link the cloud and embedded devices using ECC and appealed that their method gains all the security necessities and is very resilient to several types of threats. But the authentication method suffers from certain drawbacks [8] and is deliberated in detail in Table 3. The solutions for these shortcomings are also provided. The shortcomings have been addressed in the proposed method.

<table>
<thead>
<tr>
<th>Drawbacks</th>
<th>Definition</th>
<th>Reason</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absence of device anonymity</td>
<td>Anonymity is to secure the device specific information so the data in cloud server would remain secure.</td>
<td>Login phase $D_{IDi}$ is passed in plain text, so attacker can easily get the $D_{IDi}$ by monitoring requests.</td>
<td>The ID_I s not passed at any phase in the system. N only the hashed value of PW and $D_{IDi}$ are passed.</td>
</tr>
<tr>
<td>Mutual authentication is impossible</td>
<td>Mutual authentication between cloud server and the embedded device.</td>
<td>Authentication phase While registering the device, the cloud server creates the pwI but it is never shared to the device.</td>
<td>The pwI is created by device itself and hashed with $D_{IDi}$ and passed to the server.</td>
</tr>
</tbody>
</table>
5 Cost Analysis

The computation and storage cost is discussed in Table 4 and Table 5 respectively., where, CC is the computational cost (hash operation) and CCEcc is the computational cost (ECC point multiplication operation). The experiment results of Kilinc et al. shows the computation costs i.e. execution time of CC and CCEcc are 2.3μs and 22.26 × 10² μs. Since all the operations are lightweight the system is inexpensive when compared to other methods as shown in Table 4.

<table>
<thead>
<tr>
<th>Schemes</th>
<th>Embedded device</th>
<th>Cloud server</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kalra et al.</td>
<td>4 CC + 3 CCEcc</td>
<td>5 CC + 4 CCEcc</td>
<td>9 CC + 7 CCEcc = 15.603 × 10³ μs</td>
</tr>
<tr>
<td>Kumari et al.</td>
<td>3 CC + 4 CCEcc</td>
<td>4 CC + 4 CCEcc</td>
<td>7 CC + 8 CCEcc = 17.824 × 10³ μs</td>
</tr>
<tr>
<td>Proposed Scheme</td>
<td>3 CC + 3 CCEcc</td>
<td>5 CC + 3 CCEcc</td>
<td>8 CC + 6 CCEcc = 13.374 × 10³ μs</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Schemes</th>
<th>Number of messages</th>
<th>Number of bits</th>
<th>Storage cost (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kalra et al.</td>
<td>3</td>
<td>1760</td>
<td>320</td>
</tr>
<tr>
<td>Kumari et al.</td>
<td>3</td>
<td>1760</td>
<td>480</td>
</tr>
<tr>
<td>Proposed Scheme</td>
<td>3</td>
<td>1760</td>
<td>320</td>
</tr>
</tbody>
</table>

The communication and storage cost is lesser when compared to the Kumari et al. method and is equal to the Kalra et al. method but with lesser computational cost. The simulation results in AVISPA tool is given in Fig. 6 were both the results using OFMC back-end and using CL-AtSe back-end are shown. Both the results show that the system is safe even when the intruder has full access to the network and the different types of attacks are tried out but it is impossible to break the authentication scheme.
Fig 6. OFMC and ATSE output of the proposed scheme

6 Conclusion

The proposed ECC based mutual authentication scheme is proved to be protected against several threats. Moreover this method shows that it satisfies all the security requirements such as mutual authentication, anonymity and confidentiality. The results of the implemented system shows that the computational cost is efficient and it can be implemented on any system which has HTTP protocol and the system is proved to be safe. The future work of this paper is to use the same authentication scheme on a real-time Cloud of Things system.

References


