A Lightweight Trust Based Secure Authentication Mechanism for IoT Devices

Aditya Thakur (✉ thakuraditya.cs@gmail.com)  
Dr B R Ambedkar National Institute of Technology Jalandhar

Prashant Kumar  
Dr B R Ambedkar National Institute of Technology Jalandhar

Nisha Chaurasia  
Dr B R Ambedkar National Institute of Technology Jalandhar

Research Article

Keywords: Internet of Things, authentication schemes, security, AVISPA, verification, OFMC

Posted Date: August 21st, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3224058/v1

License: © This work is licensed under a Creative Commons Attribution 4.0 International License.  
Read Full License

Additional Declarations: No competing interests reported.
A Lightweight Trust Based Secure Authentication Mechanism for IoT Devices

Aditya Thakur  
Department of Computer Science and Engineering  
Dr B R Ambedkar National Institute of Technology  
Jalandhar, Punjab, India  
thakuraditya.cs@gmail.com

Prashant Kumar  
Department of Computer Science and Engineering  
Dr B R Ambedkar National Institute of Technology  
Jalandhar, Punjab, India  
prashantk@nitj.ac.in

Nisha Chaurasia  
Department of Information Technology  
Dr B R Ambedkar National Institute of Technology  
Jalandhar, Punjab, India  
chaurasian@nitj.ac.in

Abstract - The Internet of Things, which is in the next phase of communication, is quickly overtaking all other technologies. IoT enables seamless data exchange, interaction, and communication between various physical objects. IoT brings automation and intelligence to a variety of industries and fields, including agriculture, transportation, industry, and health. Improving user efficiency and comfort is the goal of IoT applications. The security of internet-connected devices has recently become more important because of global cyber-attacks. Authentication is one of the most important network security principles, whether for small networks like local servers or large networks like central cloud servers. For IoT applications, several solutions have also been put forth, but they are not at all efficient and as well secure. In this paper, we have proposed a trust-based authentication method, which is not only lightweight but as well is secure. For verification of the security of the protocol AVISPA tool has been used in various modes.

Keywords: Internet of Things, authentication schemes, security, AVISPA, verification, OFMC

1 INTRODUCTION

Internet of Things (IoT) has emerged as a revolutionary concept [16], in the realm of technology, connecting physical devices and enabling them to communicate and exchange data with each other over the internet. IoT has many uses since it is easily adapted to other technologies, such as wearables [18] such virtual glasses, fitness bands, and GPS belts. Hospital beds may readily include IoT to monitor vital signs, blood pressure, oximeter readings, and body temperature [19], [22]. Information is provided as traffic monitoring via Google Maps. Gathering data on a field's moisture content, acidity, presence of certain nutrients, and temperature might assist farmers [25]. Smart cameras and doorbells are common in smart homes [17].

While IoT offers numerous benefits and opportunities, it also brings forth significant security concerns that need to be addressed [28]. The interconnected nature of IoT devices and their reliance on data exchange make them vulnerable to various threats, potentially compromising privacy, data integrity, and even physical safety. Understanding and mitigating these security challenges is crucial to ensure the trustworthiness and reliability of IoT systems.

Authentication plays a critical role in the Internet of Things (IoT) environment, where a vast network of interconnected devices communicates and exchanges data. It ensures that only authorized devices and users can access and interact with IoT systems, thereby safeguarding data privacy, maintaining system integrity, and preventing unauthorized activities which also maintain Confidentiality [30] and Integrity [29]. The importance of authentication in the IoT environment cannot be over stated, as it establishes trust and enables secure interactions among devices and the broader IoT ecosystem.

There are various single factor authentication, and as well multi-factor authentication methods [20] that allow us to combine two one factor authentications to create a two-factor authentication. To securely transfer data from one device to another across an insecure channel cryptography [21] can also be employed, however owing to resource constraints in IoT devices, cryptography [26] is not feasible.so we need an authentication method which performances well in the limited resource IoT Devices and also able to secure the IoT network.

Considering these factors, in this paper we propose LTSAM

The contribution can be summarized as follows:

1. This research work presented a lightweight trust based secure authentication mechanism (LTSAM) by updating the challenge response based secure vault authentication scheme [1].

2. LTSAM has two different authentication schemes depending upon the security requirement of the authentication which is decided by the server.
3. Security analysis is done in two different sections as non-formal and formal. Moreover, for formal security analysis, section AVISPA tool simulation is used.

Rest of the paper is organized as follows: Section 2 discusses related work. Section 3 talks about the proposed LTSAM in a very detailed manner. Section 4 describes the non-formal security analysis of LTSAM. Section 5 presents the formal security analysis using AVISPA tool simulation of LTSAM. Section 6 draws the conclusion and talks about the future work for LTSAM.

2 Related Work

T. Shah et al. [1] introduced secure vaults which contain n, m-bit keys, the key in the secure vault is used to authenticate the IoT device to the IoT server to securely send the data. The authentication takes place in three steps and is driven by the Challenge-Response Mechanism, the duration of the secure communication determines the security of the session. The smaller the sessions provide greater security and larger the session compromises with the security of the session, the security of this protocol is observed as it provides security from many attacks.

B. T. Asare et al. [2] represent the Edward curve digital signature algorithm, is used as an authentication method. It includes three stages and eleven input parameters. Key Generation, Signature Generation. The signature is void if the decoding fails at any stage. Decentralized authentication process, integrity, privacy, and availability of IoT node data are all guaranteed by this technology, which also assures protection from different threats.

A. Esfahani et al. [3] introduce a simple authentication technique between the IoT device (sensor) and router. Through this approach, authentication is accomplished by two processes: registration and authentication. Both processes employ hash and XOR operations to establish security. This technique offers reciprocal authentication, secrecy about the identity of the sensor, protection against replay attacks, protection against man-in-the-middle attacks, protection against impersonation attacks, protection against modification attacks, and little computational and storage overhead.

B. B. Gupta et al. [4] introduced a technique of identity-based authentication consisting of three steps and is a lightweight authentication protocol. Setup phase, Decryption phase, validation phase. This approach protects against DDOS, and flooding attacks and employs access control to offer security.

H. Deng et al. [5] introduced Flexible privacy-preserving data sharing (FPDS) scheme, this system allows the end user the ability to encrypt data using identification as well as regulate data access. It is a reliable identity-based authentication technique that allows the sender to control who has access to the data.

J. Zhang et al. [6] introduced a very unique way to authenticate through the proximity-based mechanism and have named it Move2Auth. This process is carried out in seven steps and requires a smartphone due to the limited resources in IoT device. It is a secured method as there are variations in the movement of the smartphone as well this technique can protect against many attacks like Eavesdropping, Impersonating IoT Device, and Denial-of-Service.

B. Kim et al. [7] introduces key hiding technology for authentication between IoT devices. The method is followed in ten steps. Once authentication gets completed device clears its memory and all the variable that has been used to create the authentication pair. Software-based WBC and hardware-based PUF techniques are used for authentication.

Kumar, U. et al. [8] introduce mutual authentication technique which is able to detect and remove the harmful node in the IoT network, and hence there are two different algorithms for both of them. The algorithm used for detection is using three way handshake. The second algorithm used to detect DDoS attack. This algorithm uses identity management and network control restriction, and with the help of this algorithm the network can be secured from the harmful nodes.

D. K. Sharma et al. [9] introduced more than one authentication methods. The proposed model uses MQTT method to transfer the authentication data to the cloud, MQTT is a messaging protocol using three aspects. It uses different schemes for different levels of security. The various schemes discussed in this paper are ID and Password, OTP-Based, Certificate-Based.

A. Badhib et al. [10] introduce authentication method in a different way as to mitigate DDoS attack. The authentication method consists of the following phases: Initialization, the static authentication phase, and the continuous authentication phase. Initialization phase is carried out using a separate secure channel. In the authentication phase the device and the gateway authenticate each other. The continuous phase is performed after a regular interval of ‘T’ so to maintain the security.

M. Adil et al. [11] introduced a method, called Hash-MAC-DSDV, uses hash functions, message authentication codes (MACs), and a distance-vector routing protocol (DSDV) to provide secure communication between IoT devices. The idea is to use hash function and MACs for authentication and DSDV for a secure communication channel between devices.

A. Shahidinejad et al. [12] introduces a lightweight protocol, which uses a lightweight hash function and a distributed key generation mechanism to provide secure authentication. They also evaluate the performance of the protocol, demonstrating its low overhead and high efficiency in terms of computation and communication. The protocol consists of 12 steps. AVISPA tool is used to evaluate the protocol. This protocol ensures mutual authentication, confidentiality, anonymity, accessibility, scalability.

S. Garg et al. [13] introduced a robust lightweight authentication scheme which is of three phases. The proposed protocol allows for mutual authentication and key
agreement between two Internet of Things (IoT) nodes by using a server in a hierarchical manner. Without this hierarchy, the IoT nodes can communicate directly with each other, but this will increase the computational and communication overhead. The server evaluates the authenticity of each node before forwarding its request to the other node. The proposed mechanism is designed to be extensible and flexible for use in IoT setups, allowing the nodes to agree on a session key only after successful authentication.

### 2.14 Challenge Response

Jabbari and J. B. Mohasefi [14], introduced a mutual method to authenticate two parties which allows them to create a session key which not only is secure but also don’t need to keep the trust in the network server. This authentication scheme is used for LoRaWAN network consist of four phases. Vulnerability assessment presents that the mentioned method can withstand various attacks and is robust, while being efficient.

### 2.15 Fingerprint

A. Bedari et al. [15] introduces a secure fingerprint authentication system for IoT. Two-stage transformation system is used to transform features, for this a fusion technique is used which considers weight in the first step and convolution to remove elements for the second stage. The proposed system has been evaluated using six public fingerprint databases and has been shown to have competitive performance compared to existing cancelable fingerprint templates, energy efficient and requires low computational costs, making it suitable for use on resource limited IoT devices. The system has been evaluated using six public fingerprint databases and has been found to have competitive performance compared to existing cancelable fingerprint templates.

### 3 PROPOSED METHODOLOGY

In this section we will discuss the proposed methodology. The proposed methodology consists of two phases. Both phases are used for authentication. The first phase has more mathematical calculations and has more message transfer to authenticate and the second phase is like a lightweight version which has a smaller number of message transfers and helps to authenticate in a faster way with less calculations. Server will decide on which phase the IoT device will authenticate.

#### 3.1 Assumptions

As HLPSL comes with a restrictions like array and other data structures, Loops, functions, Pointers, and various other programming features cannot be used. So as to cope up with the implementation of algorithm we have assumed some values like values of secure Vault, Value obtained by XOR of multiple keys.

#### 3.2 First Phase:

In this section we are going to talk about the First phase. This phase requires transfer of 4 messages as shown in figure 3.1 and each message requires some amount of mathematical calculation which begins with an IoT device sending its unique Device ID to the server, in a message like

\[ M_1 = \{ \text{DeviceID} \} \quad \text{(1)} \]

In this manner the authentication will start. After Receiving M1 from the device, Server responds by generating a challenge C1 which consists of indexes from the secure vault and a random number R1 so M2 looks like

\[ M_2 = \{ C_1, R_1 \} \quad \text{(2)} \]

Which is like a response to M1 and a challenge for the IoT device. Upon receiving M2 from the server the Device will then generate K1 which is obtained by performing xor operation on all the keys whose index are present in C1, such that

\[ K_1 = V_1 \oplus V_2 \oplus V_3 \oplus V_4 \oplus V_5 \oplus V_6 \ldots \ldots \quad \text{(3)} \]

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>Message</td>
</tr>
<tr>
<td>⊕</td>
<td>XOR operation</td>
</tr>
<tr>
<td>C</td>
<td>Challenge, contains the indexes of the keys from the vault</td>
</tr>
<tr>
<td>R</td>
<td>Random number</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>(K, M)</td>
<td>Message encryption using K as the key</td>
</tr>
<tr>
<td>V</td>
<td>Value from the secure vault</td>
</tr>
<tr>
<td>T</td>
<td>Session key</td>
</tr>
<tr>
<td>T_i</td>
<td>Random number used to compute session key</td>
</tr>
<tr>
<td>P</td>
<td>Positive authentication count</td>
</tr>
<tr>
<td>N</td>
<td>Negative authentication count</td>
</tr>
<tr>
<td>{a, b}</td>
<td>Message with 2 elements</td>
</tr>
<tr>
<td>E</td>
<td>Encryption</td>
</tr>
</tbody>
</table>
In response to $M_2$, the IoT device generates a message which contains a challenge $C_2$ alongside the random number $R_1$ which was in $M_2$ and another random number $R_2$, $T_1$ for the session key. This whole message is encrypted using $K_1$ as the key,

$$M_3 = E \left( K_1, \{ R_1, T_1, \{ C_2 \parallel R_2 \} \} \right) \ldots (4)$$

Upon receiving of $M_3$, Server gets the first part of the session key that is $T_1$ and gets a challenge $C_2$ which contains the indexes of keys which is used to calculate $K_2$ such that

$$K_2 = V_1 \oplus V_2 \oplus V_3 \oplus V_4 \oplus V_5 \oplus V_6 \ldots \ldots (5)$$

For the final message $M_4$ it contains $T_2$ for the session key, $M_4$ has been encrypted using the key obtained by performing logical ‘or’ of $K_2$ and $R_2$, such that

$$M_4 = E \left( K_2 + R_2, \{ T_2 \} \right) \ldots (6)$$

Once the device receives $M_4$ session key can be generated now as both $T_1$ and $T_2$ are shared among the server and IoT device session key is generated such that.

$$T = T_1 \oplus T_2 \ldots (7)$$

Table 2 Authentication using First Phase

<table>
<thead>
<tr>
<th>Protocols First Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> Device ID; Challenge and Response; Random Number for session key; Secure Vault; <strong>Output:</strong> Successful authentication; Generation of Session key for communication; Increment/Decrement of P/N value; <strong>First Algorithm:</strong></td>
</tr>
<tr>
<td>I. From IoT to the Server, $M_1 = {UID}$</td>
</tr>
<tr>
<td>II. From Server to the IoT, $M_2 = {C_1, R_1}$</td>
</tr>
<tr>
<td>III. Generate $K_1$ such that, $K_1 = V_1 \oplus V_2 \oplus V_3 \oplus V_4 \oplus V_5 \oplus V_6 \ldots$</td>
</tr>
<tr>
<td>IV. By IoT to the server, $M_3 = E \left( K_1, { R_1, T_1, { C_2 \parallel R_2 } } \right)$</td>
</tr>
<tr>
<td>V. Generate $K_2$ such that, $K_2 = V_1 \oplus V_2 \oplus V_3 \oplus V_4 \oplus V_5 \oplus V_6 \ldots$</td>
</tr>
<tr>
<td>VI. From Server to IoT, $M_4 = E \left( K_2 + R_2, { T_2 } \right)$</td>
</tr>
<tr>
<td>VII. Compute Session key such that, $T = T_1 \oplus T_2$</td>
</tr>
<tr>
<td>VIII. Increment/Decrement of P/N depending upon the success of this session.</td>
</tr>
</tbody>
</table>
### 3.3 Second Phase:

This phase consists of 3 message transfer presented in figure 3.2 and each message can be generated by lesser and more easy calculations on comparison to the previous phase, this phase is rather the lightweight and a faster method for authentication this phase begins with IoT device sending its Device Id to the server in a message named $M_1$, such that

$$M_1 = \{DeviceID\} \ldots (8)$$

After server receives the Unique ID, it verifies the Id and prepares a message $M_2$ which is encrypted using the first key in the secure vault such that

$$M_2 = E(V_1, \{C_1, T_1\}) \ldots (9)$$

Here $C_1$ is the challenge which is generated by the server containing the indexes of the keys from the secure vault, $T_1$ is a part of the session key which will be used to generate the session key. When the IoT device receives $M_2$, it extracts $T_1$ and $C_1$ and generates $K_1$ such that

$$K_1 = V_1 \oplus V_2 \oplus V_3 \oplus V_4 \oplus V_5 \oplus V_6 \ldots \ldots (10)$$

Which is used to encrypt $M_3$, $M_3$ contains the second part of the session key $T_2$ which is the first part of the session key which is a generated number by the IoTD so $M_3$ looks like

$$M_3 = E(K_1, \{T_2\}) \ldots (11)$$

Upon receiving $M_3$ exchange of $T_1$ and $T_2$ is done, and generation of session key takes place by calculating $T$ such that. After this further communication will be encrypted using $T$.

$$T = T_1 \oplus T_2 \ldots (12)$$

<table>
<thead>
<tr>
<th>Table 3 Authentication using second phase</th>
</tr>
</thead>
</table>

**Protocols Second Phase:**

**Input:** Device ID;
Challenge and Response;
Random Number for session key;
Secure Vault;

**Output:** Successful authentication;
Generation of Session key for communication;
Increment/Decrement of P/N value;

**Second Algorithm:**

I. From IoT to the server. $M_1 = \{UID\}$

II. From Server to the IoT. $M_2 = E(V_1, \{C_1, T_1\})$

III. Generate $K_1$ such that, $K_1 = V_1 \oplus V_2 \oplus V_3 \oplus V_4 \oplus V_5 \oplus V_6 \ldots$

IV. $M_3 = E(K_1, \{T_2\})$

V. $T = T_1 \oplus T_2$

VI. Increment/Decrement of P/N depending upon the success of this session.
3.4 Deciding Factor:
This is a server-side calculation which decides to which phase the IoT device shall choose to authenticate, this is done by checking two variables named ‘P’ and ‘N’ which are assigned to a device while registration of the device represented by figure 3.3. Initially these values are assigned as ‘0’ and depending upon the future session these values are either incremented or decremented, server follows this set of rules to decide whether to Increment/Decrement the value and whether to follow First/Second phase for authentication, the rules are as follows.
If for the first time an IoT device comes for authentication, then both its ‘P’ and ‘N’ values will be ‘0’ and so it will go to the First phase for authentication and after each successful session ‘P’ value will increment by ‘1’. An IoT device must go through the First phase until its ‘P’ value reaches to ‘3’ once it exceeds ‘3’ then the Device is allowed to authenticate through the Second phase. Due to any reason if any session is not able to complete properly then ‘N’ value is incremented by ‘3’. Upon which if the device goes for authentication then it has to go through the First phase and after each successful session its N value will decrease by ‘1’, device has to go through this First phase until ‘N’ value reaches to ‘0’, When ‘N’ value is greater than ‘0’ then this is called as a suspicious state and under suspicious state if there is another miss communication then the device will be blocked from the server and will not be able to do any further communication until it is manually unblocked by user, whereas once the device is out of this suspicious state then depending upon its ‘P’ value authentication is been decided.

Updating keys: Following a successful session, the secure vault undergoes an update by performing an XOR operation between session key and all the existing keys within the vault. This process ensures that the secure vault remains updated and incorporates the changes introduced by the session.

Table 4 Deciding Factor

<table>
<thead>
<tr>
<th>Input:</th>
<th>Device ID; Success of a session; P/N value;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output:</td>
<td>Increment/Decrement of P/N value;</td>
</tr>
<tr>
<td>Second Algorithm:</td>
<td></td>
</tr>
<tr>
<td>I.</td>
<td>Check P and N value of the requesting IoT device</td>
</tr>
<tr>
<td>II.</td>
<td>If (N &gt; 0) &amp; (N &lt;= 3): Use First Phase, Decrease N by 1 after one successful session</td>
</tr>
<tr>
<td>III.</td>
<td>If (N &gt; 0) &amp; (N &lt;= 3): Block the IoT device in case of one unsuccessful session</td>
</tr>
<tr>
<td>IV.</td>
<td>If (N = 0) &amp; (P &lt; 3): Use First Phase, Increase P by 1 after one successful session</td>
</tr>
<tr>
<td>V.</td>
<td>If (N = 0) &amp; (P &gt; 3); then continue to Second Phase for authentication</td>
</tr>
<tr>
<td>VI.</td>
<td>If (N = 0) &amp; (P&lt;=3</td>
</tr>
</tbody>
</table>

4 Unformal Security Analysis
This section discusses the informal security verification of a proposed TBA (To Be Announced) scheme to assess its ability to withstand potential attacks and protect system security. The verification covers passive attacks where an adversary eavesdrops on communication, as well as active attacks where the adversary manipulates or injects malicious data. The objective is to determine if the TBA scheme effectively prevents the adversary from deriving secret keys and identities, ensuring data integrity, confidentiality, and secure communication. The analysis includes evaluating the scheme's resilience against message forgery, replay attacks, masquerade attacks, device compromise, Denial-of-Service (DoS) threats, and Man-in-the-Middle (MITM) attacks. By identifying vulnerabilities and weaknesses, improvements can be made to enhance the scheme's security. Overall, the verification instills confidence in the TBA scheme's ability to provide robust security in practical deployments.
4.1 MITM
In TBA, we implement a mechanism to generate a key that can be used to secure the message by encrypting it. This generated key is created using two separate random numbers securely transmitted between the server and the IoT device through encrypted messages. These messages employ encryption keys derived from a secure vault exclusively shared between the server and the IoT device. As a result, a Man-in-the-Middle (MITM) attacker is unable to access the generated key from the messages transmitted between the server and the IoT device. Consequently, the attacker is unable to retrieve or manipulate any messages exchanged between the server and the IoT device once the authentication process is completed.

4.2 SCA
TBA employs a unique approach where the encryption key is formed by combining multiple keys through XOR operations. This design ensures that the individual keys utilized in the encryption process cannot be deduced from the encryption key alone. Consequently, an attacker attempting a side channel attack cannot acquire information about the specific key values involved in the authentication process. Therefore, the attacker is unable to retrieve the secure vault in its entirety, replicate the IoT device, or inject falsified messages into the communication channel.

4.3 DoS
The server or IoT device can be targeted by an attacker through flooding it with a significant volume of fraudulent requests, causing it to crash due to resource limitations. However, in our architecture, resources are not allocated prior to authentication, thus preventing the possibility of a Denial of Service (DoS) attack.

4.4 DIA
DIA (Device Impersonation Attack) in this type of attacks an attacker or adversary, attempts to carry out an impersonation attack by sending an unauthorized message request to the server in order to gain the trust of a legitimate user. The adversary calculates the DID (Device ID) and current timestamp but fails in their attempt because they lack a secure vault and the corresponding keys. The attacker has the ability to manipulate secret identities and credentials, as the server verifies the timestamp and identities every time.

4.5 Masquerade Attack
A masquerade attack in the IoT environment involves an attacker impersonating a legitimate entity, such as an IoT device or user, to gain unauthorized access or manipulate the system. These attacks pose risks to the integrity, confidentiality, and availability of IoT devices and networks. Mitigating masquerade attacks requires implementing secure authentication mechanisms, managing credentials securely, using encrypted communication protocols, deploying intrusion detection systems, applying regular software updates, and raising user awareness. By adopting these security measures, organizations can reduce the risks associated with masquerade attacks, safeguard IoT systems, and protect sensitive data. For instance, when the Device starts with the first message (DID) to the server, the adversary can intercept the message and pretend to be an authentic

---

**Figure 4 Work Flow for deciding factor**

```
Start
N <= 3
YES
P < 3
YES
First Phase
N = N + 3
YES
N > 3
P = P + 1
N > 0
YES
N = N - 1
NO
END
SECOND PHASE
N = N + 3
YES
N > 3
END
```

---

**Table of Contents**

- 4.1 MITM
- 4.2 SCA
- 4.3 DoS
- 4.4 DIA
- 4.5 Masquerade Attack
trusted user by sending a fabricated first message (DID\textsubscript{Adversary}). Initially, the server receives the message from the attacker. However, the attacker is unable to generate a response for the challenge posed by the server, thereby unable to circumvent the security measures since they cannot produce the required response for the server's challenge.

### 4.6 Eavesdrop Attack
Suppose there is an attacker who attempts to eavesdrop and monitor the messages exchanged between the Device and the Server. However, the attacker is unable to successfully eavesdrop because all the messages are encrypted using the session key established during the authentication process. Additionally, the use of timestamps with each packet requested provides resistance against such attacks, ensuring the communication remains secure, and addressing concerns related to anonymity and untraceability threats.

#### 5 Formal Security Analysis
This section focuses on the security verification process, which involves utilizing the Automated Validation of Internet Security Protocols and Applications (AVISPA) tool to assess the safety of the TBA scheme. AVISPA is an automated tool used for validating internet security protocols and applications. It employs the High-Level Protocol Specification Language (HLPSL) to describe the protocols. However, this paper specifically focuses on the analysis mechanisms provided by the OFMC and CL-AtSe backends. The output generated by the tool is presented using the If format. The output format consists of five phases: SUMMARY, DETAILS, PROTOCOL, GOAL, and the BACKEND, providing comprehensive information about the analysis results.

The proposed scheme is designed with two primary roles: IoT Device (IoTD) and Server (S). To ensure the security and effectiveness of the scheme, various phases of the proposed scheme are tested using the High-Level Protocol Specification Language (HLPSL).

In order to validate the security of the proposed scheme, the AVISPA tool is utilized with the OFMC and CL-AtSe backends. These backends enable the trusted user to perform security validations even in the presence of potential attackers, effectively mitigating the risk of replay attacks.

The scheme's resilience against Man-in-the-Middle (MITM) attacks is also verified using the AVISPA tool. The design protocol script ensures that such attacks are unable to compromise the security measures in place.

Furthermore, the session and environment roles within the proposed approach are thoroughly tested. All the designed roles within the scheme successfully satisfy the requirements and objectives when analyzed using the OFMC and CL-AtSe model checker tools.
TBA scheme is been tested under OFMC and ATSE mode under the AVISPA tool and the result simulation for the first phase in OFMC is been presented in the Figure 11 and ATSE simulation results are presented in Figure 12. During the OFMC simulation the summary comes out to be safe with a depth of 9 piles, VisitedNodes 205 and in a SearchTime of 0.10 sec. In case of ATSE mode the summary comes out be safe.

Whereas for the second phase the result simulation under OFMC mode is been presented in Figure 13, and the summary comes out to be safe with a depth of 6 piles, VisitedNodes 55 and in a SearchTime of 0.02 sec. In case of ATSE mode the summary comes out to be safe presented in Figure 14.

### 6 Conclusion and Future Work

This paper has presented a trust-based authentication method that encompasses two distinct phases based on the level of trust established with the IoT devices. Both phases have been designed with robust security measures to ensure the integrity and confidentiality of data exchanges. The key differentiating factor between the two phases lies in the calculations involved. By adopting this approach, the proposed method provides a secure and efficient solution for authentication in IoT applications.
7 Declaration

7.1 Ethics Approval – This manuscript is not been submitted to any other journal or at any other conference.

7.2 Conflict of Interest – There is no conflict of interest as state by all the authors.

7.3 Data Availability – Not Available, as there is no such data or dataset.

7.4 Author Contribution - In this manuscript Aditya has contributed for conceiving and designing the proposed algorithm, collecting the data, collecting data for analysis tool, performed the analysis, wrote the paper. Dr. Prashant Kumar and Dr. Nisha Chaurasia has contributed for Conceiving and designing the proposed algorithm, Manuscript editing, data validation, Analysis validation, Reviewing of the manuscript.

7.5 Funding – No funding is been received while making this manuscript.

7.6 Consent to publish – All the publisher have the consent to publish this manuscript.

REFERENCES


