Journal of Positive School Psychology http://journalppw.com
2022, Vol. 6, No. 5, 3007-3019

A Lightweight Iot Application Security System To Ensure Secure Authentication, Integrity And Secure Session Among Iot Devices

Tarun Dhar Diwan[1] , Dr. Siddhartha Choubey[2] , Dr. H.S. Hota[3]

[1]*CSE CSVTU, Bhilai Chhattisgarh,India tarunctech@gmail.com*
[2]*CSE SSTC, Bhilai Chhattisgarh, India sidd25876@gmail.com*
[3]*CS ABVV, Bilaspur Chhattisgarh, India proffhota@gmail.com*

Abstract

The IoT is the new era technology in digital world. IoT gives physical devices the ability to process data in real time. It allows physical things to be interactive and responsive without the need for human involvement. These programmers are really promising and provide the greatest possible service. This number encourages academics to research the Internet of Things in aspects of its capabilities, effectiveness, responsiveness, problems, risks, and protection. As a result, having strong security, privacy, authentication, and attack recovery is an option. In a networked world, the most pressing security and privacy issues must be addressed. It focuses on security, privacy, and safety throughout the whole IoT value chain, including devices, networks, cloud, infrastructure, apps, and services. In this paper, a framework is proposed, termed as IoT application security system (IASS), a IoT device application security using mutual authentication, integrity checking and cryptography for establishing secure communication over insecure communication channel. IoT device application security protocol was designed and verified on AVISPA tool and then secure communication was established. The proposed multi-layer IoT security framework shows its robustness with high accuracy level to detect malicious activities. The model also proved to be lightweight with less communication overhead and complexity.

Keywords: Internet of Things (IoT), Security, Authentication, Integrity, Cryptography, AVISPA.

1. Introduction

In Internet-of-Things (IoT) scenario, client and cloud server are connected in a secure network to share application securely. This cloud-based platform distributes safe data resources to users connected via IoT, allowing information to be transferred through one device to the other without being replicated. They may safely transfer data between cloud platform and Internet-of-Things (IoT) users by enabling them. Because cloud computing is not entirely safe, a secure and authenticated structure is necessary to prevent security threats [1]. Several identification techniques have been proposed in past years. Information security, user request, and authentication mechanisms, in addition towards the security obligations, are important factors to enable the communication continuously. In terms of maintaining privacy protection, the devices identify must be distinct from those of another, especially adversaries [2]. Numerous approaches have recently been provided to recognize anonymity relevance. Several conventional methods are ineffective in providing user anonymity in IASS. According to the various security vulnerabilities that have been brought to light, a secure cloud computing-based IASS must address the core concerns given in Fig 1.

The application layer defines all apps that use IoT technologies or where IoT has already being implemented. It's feasible to create smart

homes, smart cities, smart health, animal tracking, and other IoT applications. It is in charge of delivering services to the apps. Because offerings are based on data acquired by detectors, they may vary depending on the Applications. A variety of challenges in the applications layer revolve around security. When IoT is utilised to create a smart house, it brings several dangers and weaknesses from both inside and outside. One of the major challenges to ensure security in IoT-based intelligent home is that these devices consists of low computer power and the low level of storage that devices are used in smart homes like ZigBee. Therefore, it is needed to design an lightweight security system to ensure overall security aspects.

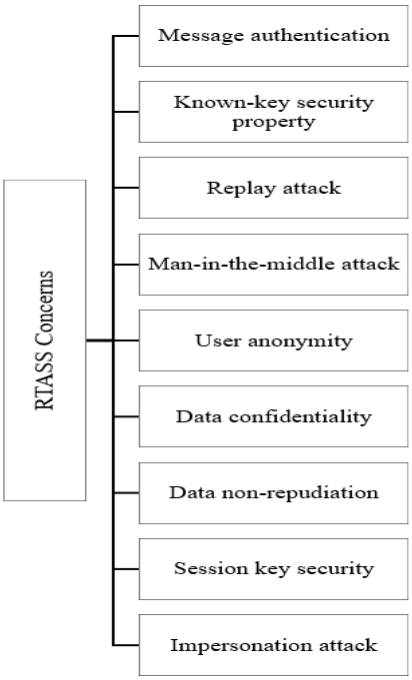

Figure 1: Security Concerns of IoT Application layer

The following are the writer's main elements:

- State-of-the-art about IoT application security frameworks to ensure secure sessions with authentication and integrity checking services.
- This paper also presented a model for IoT application security intending to design a lightweight algorithm for IoT devices.
- Formal and informal investigation of the IASS is presented in this paper.
- Finally, this research compares the current state of the art to established methodologies.

The following are the illustrations for the remainder of this paper: The background information for IoT security, as well as its approaches, are covered in Section 2. The article provides an explanation of the technique and strategies in section 3. The IASS result analyses are reported in section 4. This part also includes a formal and informal security study of IASS, as well as a comparison of the current state-of-the-art with existing approaches. Ultimately, the conclusions and upcoming study objectives are described in section 5.

2. Literature Review

Nawaf Almolhis et al. [1] discussed the security problems of the IoT cloud. The fundamentals of the IoT cloud are examined, followed by an analysis of the security issues that consumers may face while utilizing smart devices that are connected to the cloud. In addition, solutions from the literature are researched and given. Open security research challenges that need immediate attention from the research community are presented, along with some potential solutions that could work well with the IoT cloud paradigm. Lastly, they hope that our study will be a useful contribution to allowing the secure integration of IoT and cloud computing. Junlong Zhou et al. [2] presumed that security services based on cryptographic algorithms are provided in real-time MPSoC systems. Conversely, in some real-world settings, this presumption is not always correct. Yang Lu et al. [3] examined critical elements of IoT cyber security in this article, including the state-of-the-art of current role and prospective future research directions, massive defenses against IoT threats, and industry applications. They also presented and discussed a proposed four-layered IoT cyber security architecture as well as taxonomy of IoT cyber security attacks. IoT becoming a safe connection for users, software/hardware, processes, and things thanks to cyber security. If this is the case, the Internet of Things will provide the world with more transparency, authenticity, reliability, sustainability, secrecy, and interoperability. Simultaneously, one of the key objectives of IoT in the near future will be cyber security. Shanto Roy et al. [4] discussed that because of the distributed structure, the IoT ecosystem

provides a broader possibility for leveraging BC for security purposes. Further security concerns can be addressed by installing a lightweight BC system in addition to the autonomous authenticating and verification procedure in a decentralized system. Furthermore, the development of a standard integration of many ecosystems necessitates additional research. Francesca Nizzi et al. [5] introduced Address Shuffling Algorithm with HMAC (AShA), a revolutionary method for fast, safe, and collision-free addressing renewal in any (IPv6) network, in this study. They examined its characteristics and limitations using an analytical formulation and simulations, and then demonstrated that our simulation and experimental results are consistent. The mathematical study can be used to anticipate AShA performance, providing a direction for cyber security planning. In comparison to previous work, AShA offers the advantage of preventing any conceivable information leakage, including the network's number of nodes. Park et al. [6] concentrated their efforts on developing lightweight mutual authentication for medical IoT applications. A 10ms processing time was recorded in this study. The secure paradigm for mutual validation and key agreements was also presented by Xu et al. [7] and Chen et al. [8]. The processing time was found to be around 5 milliseconds. Strong identification systems developed by Lee et al. [9] and Maitra et al. [10] reached 13ms and 6ms, correspondingly.

3. Methodology

IASS allows users to start a dialogue well with cloud side across unsecured public networks. Every IoT client may connect together through remote server as just a feature of IASS, i.e., a user passes her/his information to the cloud, and another IoT client gathers and uploads her/his information to the cloud. Moreover, because communications are conducted through an insecure communication, it is critical to understand how to obtain additional assistance via IoT services via an unsafe communication route.

The major design challenges in the prior frameworks are :

- During the upload phase, several protocols fail to authenticate messages.
- In the upload phase, certain protocols don't really allow session keys.
- During the user's data upload phase, some schemes are vulnerable to impersonation attacks.
- User anonymity is not guaranteed in certain protocols.

In order to handle these issues, this section proposed an application layer security model for IASS. In this section a secure mutual authentication protocol is proposed using low complexity elliptical curve cryptography. The preliminaries of elliptical curve cryptography is discussed in below section.

3.1 IASS Authentication Protocol

3.1.1 Architecture of IASS Authentication Protocol

Proposed Protocol architecture is described as below, figure 2:

- IoT User (U_A) register itself to the server, C_s.
- IoT Gateways (G) register itself to the server, C_s.
- If another IoT user (U_B) wants to access the file of U_A, then authentication is performed among them to access data by C_s.
- IoT devices collects data (F_A) and forward it securely to the G and G collectively transfer data to C_s.
- Meanwhile, network administrator (N_A) checks the arriving data for malicious activities as discussed in above section.
- Network administrator (N_A) sends the report to C_s.
- C_s upload the file to the server.

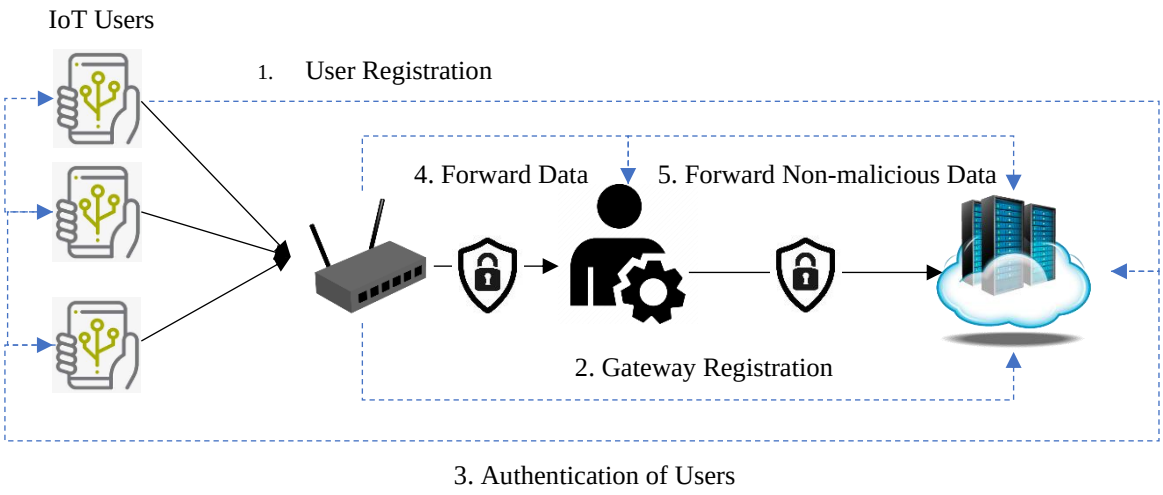

Figure 2: Architecture of IASS Application Layer Security Protocol

3.1.2 Phases of IASS Authentication Protocol

While deploying above mentioned algorithm, for IoT protection, a lightweight and secure authentication mechanism is required. as devices or nodes in IoT network are resource limited. Therefore, the IASS application layer security model proposed a secure

Step 2: Gateway, G then generates a secret number, P_{uid} and perform xor operation with UI_d such that $H_{id} \rightarrow U_{id} \oplus P_{Uid}$. Then P_{uid} and H_{id} are transmitted to server. This id is termed as identification for both user and gateway at server. Server then save both identities for further authentication.

Step 3: Server, C_s, then generate a smart identity, $S_{cd} = \{H_{id}, P_{Uid}, S_{hcd}\}$, for G and G generates $S_{cd1} = \{U_{id}, P_{Uid}, S_{hcd2}\}$ and send it to U through secure channel.

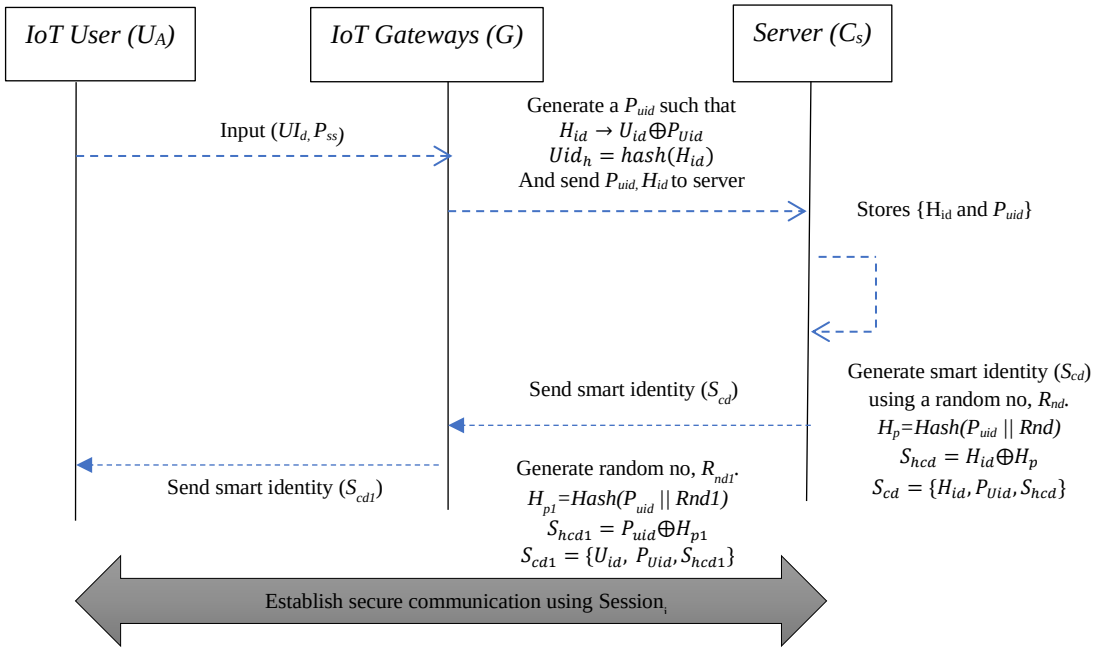

Figure 3: User Registration in IASS Application Layer Security Protocol

Login and Authentication Phase

For login and transmission of data or accessing data, user (U) sends UI_d to gateway G and its authentication is represented as in following steps (Figure 4):

Step 1: For login or authentication of user, U enters S_{cd} and enters H_{id} and send these to server.

Step 2: At G, server checks the validity of S_{cd1} as H_{p1}=Hash(P_{uid} || Rnd1) and $H_{id}^{*} = S_{hcd1} \oplus H_{p1}$. Match the H_p and H_{id}^{*}. If they match then the S_{hcd} is validated otherwise not. If verification is positive then it a $V_{f1,}$ is send to C_s.

Step 3: At $C_{s,}$ server checks the validity of S_{cd} as H_p=Hash(P_{uid} || Rnd) and $H_{id}^{*} = S_{hcd} \oplus H_p$. Match the H_p and H_{id}^{*}. If they match then the S_{hcd} is validated otherwise not. If verification is positive then it a $V_{f,}$ is send to U that login successful otherwise failed $V_{f,}$ is send to gateway to show that authentication is failed. If the V_f fails, then C_s will reject the authentication request. Such situation occurs when their desynchronization among G and U. In such situation, authentication reattempt is performed. After successful reattempt of authentication, G and U are resynchronized for future secure connection.

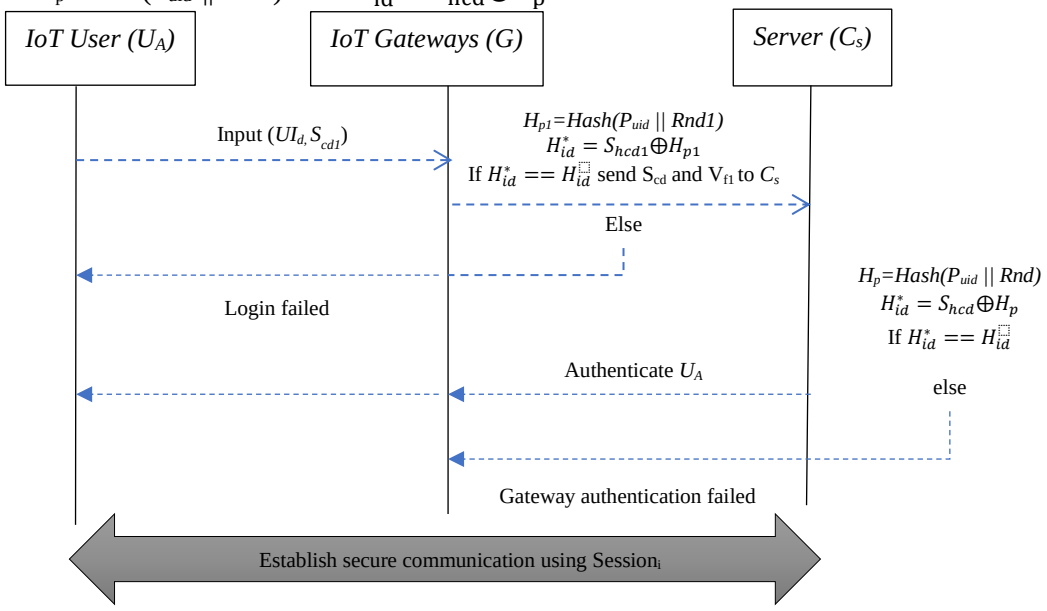

Figure 4: Login and authentication in IASS Application Layer Security Protocol

Credentials Update

In this section, if any user wants to update the credentials such as their password, then legal IoT user U whose steps are discussed as below, (as illustrated in figure 5):

- The user U first logs into the system as described in above section.
- Then user enters his new password, and repeat authentication phase to generate smart identity.
- The author regenerates $S_{cd1}$and S_{cd} and update at U, G and Cs.

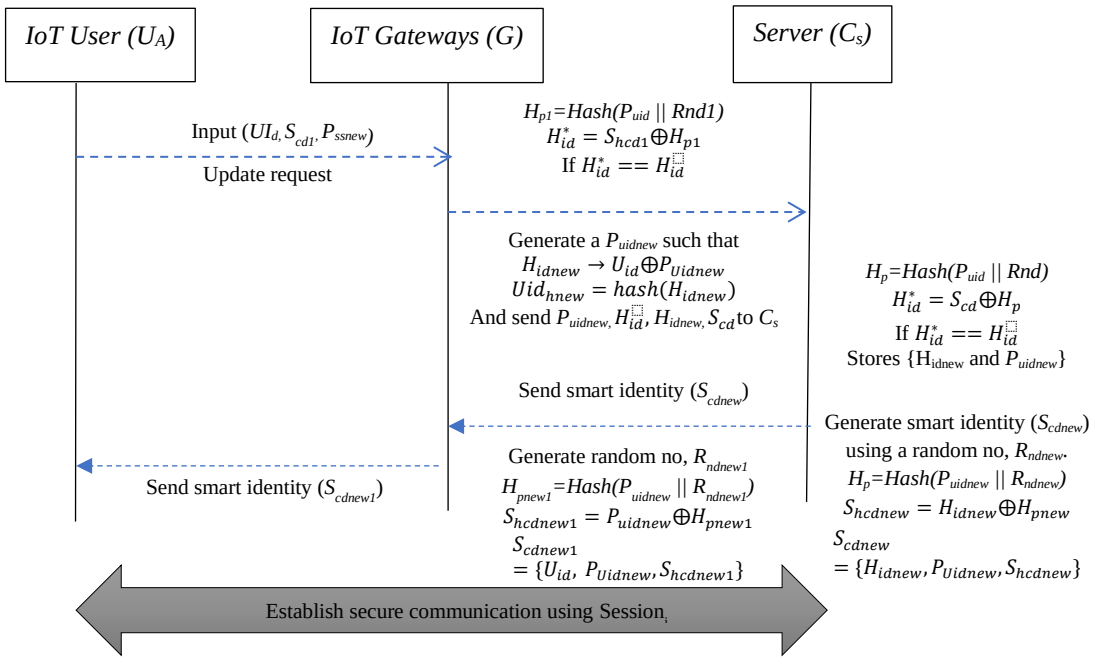

Figure 5: Credential Update in IASS Application Layer Security Protocol

3.2 IASS Data Confidentiality Protocol

3.2.1 Basics of Elliptical Curve Cryptography

Suppose E denote an elliptic curve over through the prime finite field, and q be the large prime. Fq is an elliptic curve expression across a prime finite set given by $x3 + ax + b = y^+$ and mod q with $a, b \in Fq$ and $4a3 + 27b2$ mod q not equal to zero. This is an elliptic curve that is not singular. And then there is the G an additive elliptic curve group as given by:

$$G == \{(x,y) : x,y \in Fq;\ (x,y) \in E\}\ S\ \{\Theta\}, \quad (6.8)$$

The following are a few elliptic curve functions:

1. Suppose $P = (x,y) \in G$, then $-P = (x,-y)$ such that $P + (-P) = \Theta$.
2. Whenever $P = (x1,y1), Q = (x2,y2) \in G$, then $P + Q = (x3,y3)$, where $x3 = \lambda 2 - x1 - x2$ mod q and $y3 = \lambda(x1 - x2) - y1$ mod q, where

$$\begin{cases} \dfrac{y2-y1}{x2-x1} \bmod q \text{ if } P, Q \\ \dfrac{3x\,2\,1 + a}{2y1} \bmod q \text{ if } P = Q \end{cases} \quad (6.9)$$

3. Suppose $P = (x,y) \in G$ then, scalar multiplication in G defined as: $tP = P + P \ldots + P\ (t - times)$.
4. Whenever g is the base point of G with order n, then $ng = \Theta$.

The ECC has the benefit of having shorter keys, encryption keys, and signatures [11]. Furthermore, considerably faster encryption algorithm, signatures, encryption/decryption, and elliptic curves in hardware are also possible. The following Table 1 compares the key size security of ECC and RSA in terms of number of bits [12].

Table 1: Comparison of ECC with RSA with Security Concerns

ECC key size(Bits)	163	256	384	512
RSA key size(Bits)	1024	3072	7680	15360
Key size ratio	1:6	1:12	1:20	1:3

3.2.2 Architecture of IASS Data Confidentiality Protocol

The architecture of IASS data confidentiality protocol is illustrated in figure 6 and details are described as further:

- Firstly, IoT users, U, takes registration to gateway G as well as at server C_s.
- G authenticates U at server.
- U collects the data, D and forward it to G securely with digital signature.

- C_s upload the D to server with its File identification number F_{Id}.

In this phase, U collects data and wants to transfer to C_s. U makes the request to C_s using its identity and send to G. Then G on behalf of U send identification Id of its own and U to C_s. Then C_s issues smart identity to ensure authentication of U and G for further communication. After mutual authentication, following steps are used for secure communication over insecure channel as discussed below:

Step 1. G got data from U (U_{id}, Data) from IoT devices. Then G generate its identity H_{id} and encrypt the data ($Data_{enc}$=Enc(Data)). G generates a random number s (such that sϵZ), sequence number, $R = session_{rnd} \oplus h(Pu_{id}||U_{id})$. Then G generates the H_s=hash($Data_{enc}$). Then G transmit $Data_{enc}$ to Cs M(H_{id}, $Data_{enc}$, R, H_s) via doubtful transmission network.

Step 2. On getting msg, M, from G, C_s verifies the message. Then generate $session^*_{rnd} = R \oplus h(Pu_{id}||U_{id})$ and verifies $session^*_{rnd}$== evaluate $session_{rnd}$ and if it is true then C_s generates H_s^*=hash($Data_{enc}$). If H_s^*==H_s then decrypt the data and transmit the acknowledgement to U through insecure communication channel.

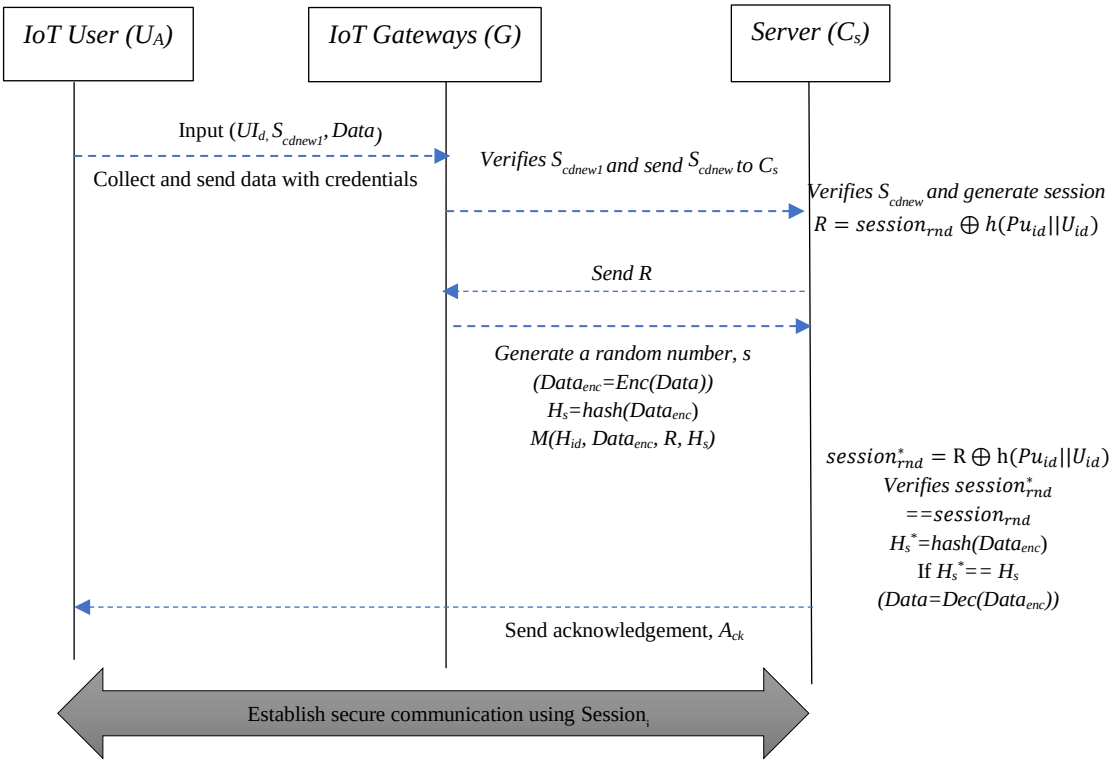

Figure 6: Secure Communication using IASS Application Layer Security Protocol

4. Result Analysis

IASS enables IoT application and its security system at network and application layer. The cloud-based technology enables the delivery of data security for safe storage and retrieval. Because the cloud computing technology isn't entirely safe, a secured and authenticated structure is necessary to prevent security breaches. Therefore, in this sub-section, a mutual authentication mechanism between IoT users as well as the cloud is presented. The sub-section seeks to set up and maintain IASS's authenticating users reliability. In addition, the provided architecture is resistant to Man in the middle attacks, user anonymity, confidentiality of information, data non-repudiation, message integrity security, and session key security.

4.1 Analysis of Security

4.1.1 Formal Security Analysis

In this part, IASS used the widely-accepted AVISPA method to perform formal security verification on the developed framework. AVISPA is a one-press tool for automatic security protocol testing. The AVISPA is used to determine if a security protocol is safe or unsafe regarding reply attack threat. The security mechanism that will be evaluated in AVISPA must be written in the participation terminology referred as High-Level Protocol Speciation Language (HLPSL). HLPSL2IF is a built-in interpreter that translates HLPSL coding to the Intermediate Format (IF). The IF is then sent to one amongst AVISPA's four accessible database servers to generate the Output Format (OF). AVISPA's four backend are as regards:

- The 1st backend is OFMC, which uses multiple symbolism approaches to investigate the subspace on request.
- The 2nd backend is the CL-AtSe, that renders any security protocol specification expressed as a transitional connection in intermediary format into a set of conditions that may be used to determine whether there were any threats on protocols.
- The 3rd backend is the SATMC, which generates a propositional equation that can then be put into a cutting-edge SAT resolver, as well as any modeling discovered is translated back into an attack".
- The 4th backend is the TA4SP which uses normal tree linguistics to estimate attacker information.

AVISPA is a commonly used modeling system in comprehensive security validation, and it determines whether an authentication process in security is Secure or not. HL-PSL is supported by AVISPA. Figure 7 depicts an AVISPA tool's construction.

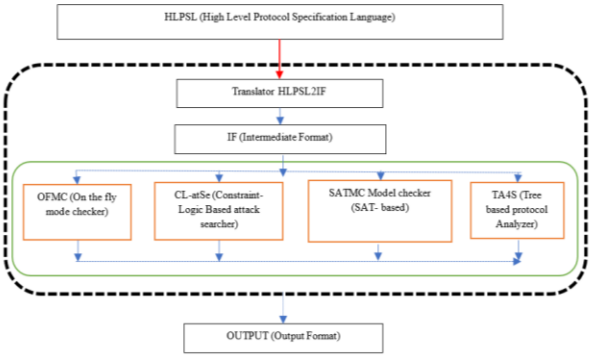

Figure 7: Architecture of HLPSL

Figure 8 and 9 shows the simulation result of avispa tool to verify the IASS protocol for application security.

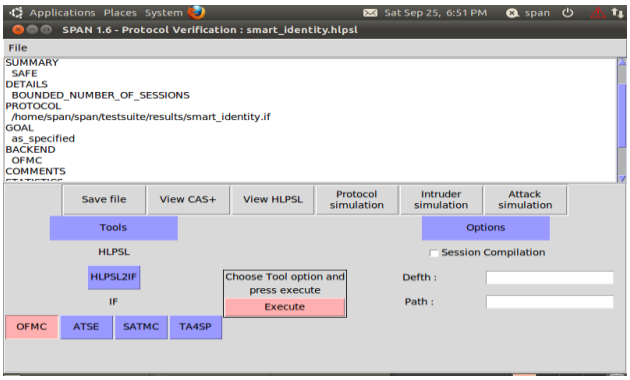

Figure 8: Authentication Simulations in the OFMC Backend

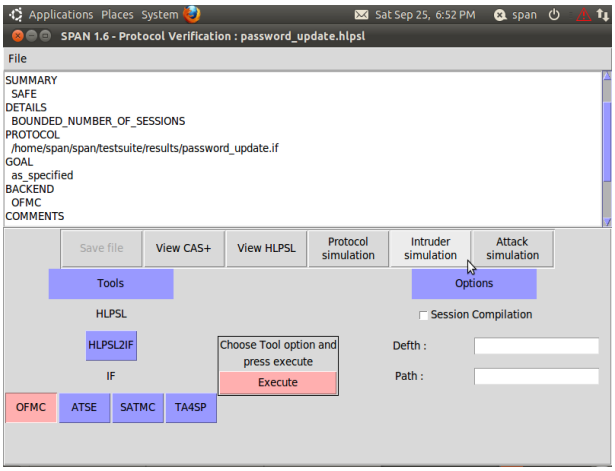

Figure 9: OFMC Backend Credential Update Simulation Result

4.1.2 Informal Security Analysis

While the authentication stage, estimations are carried out among the involved entities, which including IoT clients U and gateways units G, as well as the path linking it. IASS provides safe transfer of confidential parameters via an unstable link. Furthermore, IASS shows that the protocol and communication truthfulness within the transmitted entities are efficient verification techniques.

Assuming a shared secret code: Our approach produces a diversified intriguing function from sample data of both operations, in addition to the protection of the U and session keys. The IASS incorporates secret parameters like

$PU_{k_i}, PR_{k_i}, Sec_{k_i}$, sessions series, and Hash algorithm strengths to safeguard the protocol's core characteristics. Furthermore, while our model employs a cryptographic keys elimination mechanism, it is extremely impossible for a hacker to guess the addresses by reading the KAS database.

Anonymous Endpoints: The identity of user U is masked by a set of arbitrarily determined attributes, that should be preserved private by either one-way decoding technique. It is exceedingly challenging for an attacker to estimate the sent parameters on an unsecured network. After the U's identities are discovered within the startup phase, the gateways node G renew the U's identity documents for every single session at the C_s.

Brute Force Attack: We're utilising the SHA series of algorithms in our project, that are 512-bits in length. As a consequence, the attacker will be unable to conduct a polynomial-time estimation attacks on our gadget hash algorithms key or any variables. The device's hash functions key lengths is adequate for the access control mechanism. This could, although, be enhanced as in longer run to meet safety standards if required [14].

Replay Attack: Once the C_s, get a response they generate a sensible id S_{cd1} which equals $S_{cd1} = \{U_{id}, P_{Uid}, S_{hcd2}\}$. Although the difficulties of the identity anonymization procedure, the attacker is incapable to identify all of those factors in order to create a fraudulent S_{cd1}. Because an attacker never exploit an old sessions id for a present operation, the offender should also knows present session id. As a result, the replaying strike cannot be used against this system. [15].

Integrity: The module's protection procedures are protected by the one-way hash algorithm's capability. It also saves the identity manufacturing, mainly within the data transfer authentication procedure [16]. As a result, in addition to confidentiality and secrecy, integrity is essential.

Session Hijacking Attack: Adversary A might acquire any data supplied through an unstable data transmission medium. The communicating entities U, G and C_s also have dispersed borders. The enemy may get all requirements conveyed on all these ends through U to G, G to C_s, and return. The intruder might not be unable to determine the particular transmission route or the disguised identifier because the values are encoded. Moreover, even if the intruders has a legitimate previous sessions id, he or she would still not be capable of attacking the G or U for the present session[17]. As a result, the method is secure against data breaches.

Collision Attack: Adversary A tries a variety of methods to defeat the technique and obtain criteria of the models. As an outcome, the sophisticated encoding or hashing should avoid conflicting with adjacent functions [18]. A key lengths above 256 is higher than enough to withstand a collision assault [19].

Forward/Backward Secrecy: Forward confidentiality refers to the suspect's ability to predict the real key combinations. When an attacker collects as many past session keys as feasible in order to uncover the prior session id, backwards secrecy is established. A randomized value, a fresh sensor identification, a secret value, and the intelligent id are among the factors that protect the framing regulations in the IASS methodology. Even if the attacker properly anticipated the keypair, he or she might not be capable to estimate intelligent id or breach prior intelligent id due to the complex modelling technique. As a consequence, total forwards and backwards secrecy is achieved using this technique [22]. Table 2 shows an informal features evaluation that is matched to previous studies.

Table 2: Evaluation of Informal Safety Aspects in Comparison

Algorithm	Park et al. [6]	Xu et al. [7]	Lee et al. [8]	Maitra et al. [9]	Chen et al. [10]	Proposed
MA	√	√	x	√	√	√
SKA	√	√	√	√	√	√

SS	x	x	x	x	x	√
F/BS	√	√	√	√	√	√
SH	√	√	√	√	√	√
SA_1	NM	NM	√	√	NM	√
SA_2	√	√	√	√	√	√
SA_3	√	√	√	√	√	√
SA_4	NM	NM	x	x	NM	√
SA_5	NM	NM	x	x	NM	√
SAC	x	x	x	x	x	√

Where,

MA= Mutual Authentication, SKA=Secure Key Agreement, SS=Secure Session, F/BS= Forward/Backward Secrecy, SH= Secure Hashing, SA_1= Security Against Brute Force Attack, SA_2=Security Against Reply Attack, SA_3= Security Against Man-in-Middle Attack, SA_4= Security Against Session Hijack Attack, SA_5= Security Against Collision Attack, SAC= Secure Access Control. √=security feature is included, x= security feature is not included, NA= security feature not mentioned.

SAC = Secure Access Control. √ = safety mechanism is available, x = safety mechanism is missing, NM = safety mechanism is not mentioned.

4.1.3 Performance Analysis

The simulated performance testing of our safe authenticating & integrity verifying technique in aspects of processing complexity and expense is discussed in this part. This portion additionally includes a comparison of the proposed model to some other trustworthy alternatives. We tested the suggested approach on a private laptop containing intel i5 microprocessor running at 3.71GHz, 6GB of RAM, and Windows 10 installed. Python is a programming language that is widely used. The processing time, also known as running time, is a rough estimate of how long it will take to complete a task. The overall calculated time for executing the application is shown in tables 3 and 4. The developed algorithm's processing time, complexities, and information exchange expense are also assessed. The processing duration is estimated to be 3 milli-second. Authentication duration, encryption/decryption time, and hash checking time are all used to assess the complexities.

Table 3: Analysis of the Proposed Model's Execution Time

	Execution Time (in ms)
User Authentication	~3
Gateway authentication	~4
Verification time	~7

File Execution time	~3

Table 4: Analysis of the Proposed Model's Execution Time

Schemes	Execution Time (in ms)	Complexity
Park et al.,	~10	$20T_h$
Xu et al.,	~5	$10T_h$
Lee et al.,	~13	$12T_h+5T_{Enc/Dec}$
Maitra et al.,	~6	$9T_h+8T_{Enc/Dec}$
Chen et al.,	~5	$11T_h$
IASS	~3	$7T_A+7T_h+3T_{Enc/Dec}$

T_A=Authentication time, T_h= Hash time, $T_{Enc/Dec}$= Encryption/decryption time

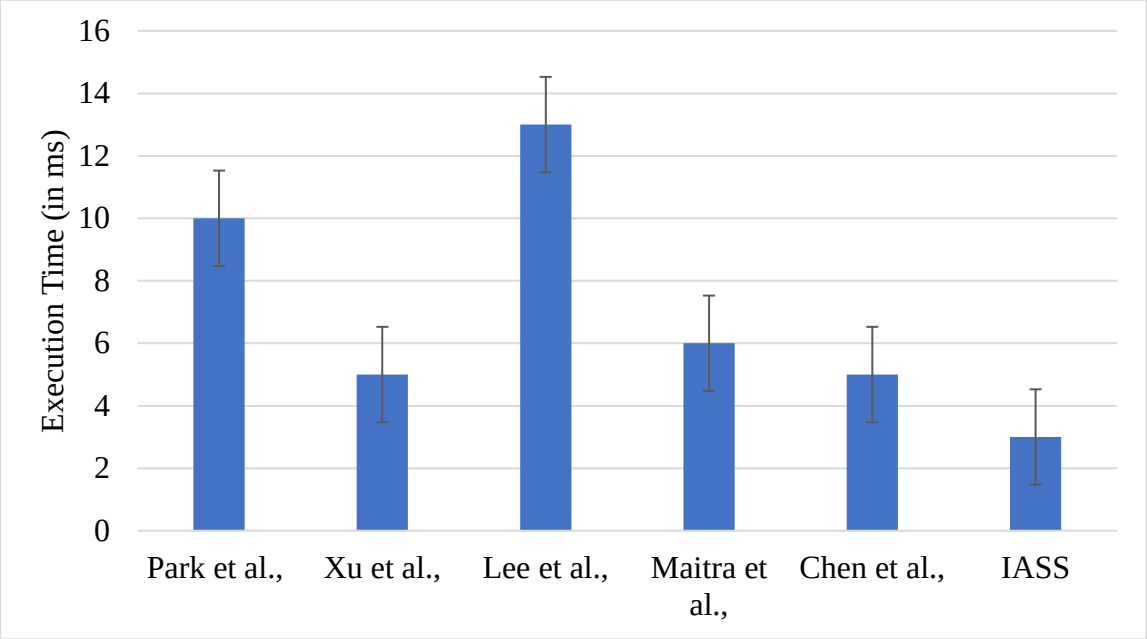

Figure 10: Comparative Execution Time Analysis

5. Conclusion

While providing cyber-security in IoT device application's security is a big concern, but there are still a couple of major hurdles to solve. This field of invention is in its early stages, but researchers anticipate it will generate a lot of attention in the future years. They hope that this work inspires new and realistic implementations of safe, reliable, and privacy-enhancing IoTs. In this paper, a secure mutual authentication, cryptographic framework is proposed to overcome these issues. In mutual authentication algorithm is verified on AVISPA tool and further the secure communication is further simulated to provide IoT device application security. This model also ensures integrity, scalability and low computation cost. Additional data analysis methodologies as well as cryptosystem, including such homomorphic or searchable encryption, are required to protect the confidentiality of customers and/or enterprises while processing large amounts of data.

References

[1] Almolhis, N., Alashjaee, A. M., Duraibi, S., Alqahtani, F., & Moussa, A. N. (2020). The Security Issues in IoT-Cloud: A Review. Proceedings - 2020 16th IEEE International Colloquium on Signal Processing and Its Applications, CSPA 2020, Cspa, 191–196.

https://doi.org/10.1109/CSPA48992.2020.9068693

[2] Zhou, J., Sun, J., Cong, P., Liu, Z., Zhou, X., Wei, T., & Hu, S. (2020). Security-Critical Energy-Aware Task Scheduling for Heterogeneous Real-Time MPSoCs in IoT. IEEE Transactions on Services Computing, 13(4), 745–758. https://doi.org/10.1109/TSC.2019.2963301

[3] Lu, Y., & Xu, L. Da. (2019). Internet of things (IoT) cybersecurity research: A review of current research topics. IEEE Internet of Things Journal, 6(2), 2103–2115. https://doi.org/10.1109/JIOT.2018.2869847

[4] Roy, S., Ashaduzzaman, M., Hassan, M., & Chowdhury, A. R. (2019). BlockChain for IoT security and management: Current prospects, challenges and future directions. Proceedings of 2018 5th International Conference on Networking, Systems and Security, NSysS 2018, 1–9. https://doi.org/10.1109/NSysS.2018.8631365

[5] Nizzi, F., Pecorella, T., Esposito, F., Pierucci, L., &Fantacci, R. (2019). IoT security via address shuffling: The easy way. IEEE Internet of Things Journal, 6(2), 3764–3774. https://doi.org/10.1109/JIOT.2019.2892003

[6] K. Park et al. (2020). LAKS-NVT: Provably Secure and Lightweight Authentication and Key Agreement Scheme Without Verification Table in Medical Internet of Things. IEEE Access, 8,119387-119404. https://doi.org/10.1109/ACCESS.2020.3005592

[7] Xu, Z., Xu, C., Liang, W., Xu, J., & Chen, H. (2019). A lightweight mutual authentication and key agreement scheme for medical internet of things. IEEE Access, 7, 53922–53931. https://doi.org/10.1109/ACCESS.2019.2912870

[8] Xu, Z., Xu, C., Chen, H., & Yang, F. (2019). A lightweight anonymous mutual authentication and key agreement scheme for WBAN. Concurrency and Computation: Practice and Experience, 31(14), e5295. https://doi.org/10.1002/cpe.5295.

[9] Lee, Y. C., Hsieh, Y. C., Lee, P. J., & You, P. S. (2014). Improvement of the ElGamal based remote authentication scheme using smart cards. Journal of Applied Research and Technology, 12(6), 1063–1072. https://doi.org/10.1016/S1665-6423(14)71666-9.

[10] Maitra, T., Obaidat, M. S., Giri, D., Dutta, S., & Dahal, K. (2019). ElGamal cryptosystem-based secure authentication system for cloud-based IoT applications. IET Networks, 8(5), 289–298. https://doi.org/10.1049/iet-net.2019.0004

[11] V. Kumar, S. Jangirala, M. Ahmad. (2018). An efficient mutual authentication framework for healthcare system in cloud computing. Journal of medical systems 42 (8) 142.

[12] N. Kumar, K. Kaur, S. C. Misra, R. Iqbal. (2016). An intelligent rfid-enabled authentication scheme for healthcare applications in vehicular mobile cloud, Peer-to-Peer Networking and Applications 9 (5) 824–840.

[13] H. Darrel, M. Alfred, V. Scott. (2004). Guide to elliptic curve cryptography. Hankerson Darrel, Menezes Alfred J., Vanstone Scott Springer-Verlag Professional Computing Series.311.

[14] Vinod Kumar, Musheer Ahmad, Adesh Kumari,Saru Kumari, M. K. Khan. "SEBAP: A secure and efficient biometric-assisted authentication protocol using ECC for vehicular cloud computing", International Journal of Communication Systems, 2019.

[15] Adesh Kumari, Srinivas Jangirala, M. Yahya Abbasi, Vinod Kumar, Mansaf Alam. "ESEAP: ECC based secure and efficient mutual authentication protocol using smart card", Journal of Information Security and Applications, 2020.

[16] Pawan Kumar, Ashutosh Kumar Bhatt, "Enhancing multi-tenancy security in the cloud computing using hybrid ECC-based data encryption approach", IET Communications, 2020.

[17] Junlong Zhou, Jin Sun, Peijin Cong, Zhe Liu, Xiumin Zhou, Tongquan Wei, Shiyan Hu, "Security-Critical Energy-Aware Task Scheduling for Heterogeneous Real-Time MPSoCs in

IoT", IEEE Transactions on Services Computing, 2020.

[18] Kisung Park, Sungkee Noh, Hyunjin Lee, Ashok Kumar Das, Myeonghyun Kim, Youngho Park, Mohammad Wazid. "LAKS-NVT: Provably Secure and Lightweight Authentication and Key Agreement Scheme Without Verification Table in Medical Internet of Things", IEEE Access, 2020.

[19] Mohammed Alshahrani, Issa Traore. "Secure mutual authentication and automated access control for IoT smart home using cumulative Keyed-hash chain", Journal of Information Security and Applications, 2019.

[20] Peng Liu, Kun Liu, Tingting Fu, Yifan Zhang, Jia Hu. "A privacy-preserving resource trading scheme for Cloud Manufacturing with edge-PLCs in IIoT", Journal of Systems Architecture, 2021.

[21] Shantanu Pal, Michael Hitchens, Vijay Varadharajan, Tahiry Rabehaja. "Policy-based access control for constrained healthcare resources in the context of the Internet of Things", Journal of Network and Computer Applications, 2019.

[22] Sahil Garg, Kuljeet Kaur, Georges Kaddoum, Joel J. P. C. Rodrigues, Mohsen Guizani, "Secure and Lightweight Authentication Scheme for Smart Metering Infrastructure in Smart Grid", IEEE Transactions on Industrial Informatics, 2020.