Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 31 No. 3s (2024)

Secure and Swift: A Proactive Approach to Defend Lightweight Key Exchange Mechanisms Against Replay, Masquerade, and Message Forgery Attacks in Cloud-IoT Communication

Gaikwad Vidya Shrimant[1], K. Ravindranath[2]

[1] Research Scholar, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India.

[2]Supervisor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India.

Mail id: gaikwad.vidya30@gmail.com
ravindra_ist@kluniversity.in

Article History:

Received: *13-04-2024*

Revised: *30-05-2024*

Accepted: *16-06-2024*

Abstract:

The Internet of Things (IoT) has rapidly advanced, making it feasible to connect numerous embedded devices to the internet for data sharing. Since the embedded devices have limited power, storage, and processing capacity, it is necessary to integrate embedded devices with a big resource pool, like the cloud. This technological integration is anticipated to bring about a remarkable expansion in the present and future exciting uses of IoT. Security concerns such as device data privacy and authentication are important ones in this regard. The goal of the current study is to develop a lightweight key establishment for safe mutual authentication system for cloud servers and the Internet of Things. Lightweight key establishment are used in our approach for authentication, while advanced encryption standard is used for confidentiality and integrity. The suggested plan achieves efficiency in performance analysis as well as security aspects including confidentiality, integrity, and authentication in security analysis. Proposed mechanism additionally, does an informal analysis and comparison of the security attributes, including replay attack, masquerade attack, message forgery attack.

Keywords: IoT, Message Forgery Attacks, Masquerade etc.

1. Introduction:

Cloud-IoT security is crucial for protecting the privacy and security of data and devices in environments that utilize both cloud computing and Internet of Things (IoT) technologies. As IoT continues to expand, connecting physical devices, sensors, and actuators to the internet, it becomes increasingly important to implement robust and effective security measures. In order to address the unique challenges and risks inherent in this interconnected ecosystem, various components and considerations must be taken into account.

Key components and considerations in Cloud-IoT security include:

a) Device Security:

- **Authentication and Authorization:** Implement secure methods for authenticating and authorizing devices before allowing them to connect to the cloud.

- **Device Identity Management:** Manage and secure unique identities for each IoT device to prevent unauthorized access.

b) Data Security:

- **Encryption:** Use strong encryption protocols to protect data both in transit and at rest, ensuring confidentiality and integrity.
- **Secure Data Storage:** Implement secure storage practices for sensitive IoT data within cloud services.

To fulfil these security requirements, numerous authentication protocols have been put forth for IoT and cloud servers. Nevertheless, existing protocols exhibit certain limitations that warrant further attention. Particularly in environments characterized by limited memory and power resources, where achieving heightened security with minimal key length is imperative, the Advanced Encryption Standard (AES) stands out as the preferred private key cryptography scheme.

2. Related Work:

The author presents an overview of various types of data breaches, including insider attacks, outsider attacks, and attacks on cloud providers. The authors also discuss the impact of data breaches on organizations, such as loss of reputation, legal issues, and financial losses [1].

The authors claimed that their proposed protocol is secure against various types of attacks such as replay attacks, man-in-the-middle attacks, and impersonation attacks. They also claimed that their protocol is efficient in terms of computational overhead and communication overhead [2].

The authors identified several security and privacy concerns associated with cloud-based IP cameras, including the lack of encryption, inadequate authentication mechanisms, and vulnerabilities in the cloud infrastructure. They also discussed the risks associated with the use of third-party cloud services, which may not provide adequate security measures [3].

The authors also discuss the role of cloud service providers in ensuring data security. They argue that cloud service providers should be transparent about their security practices and should provide their customers with tools and resources to help them manage their data security [6].

The paper highlights the need for organizations to adopt a comprehensive approach to cloud security, including regular security audits, risk assessments, and the implementation of robust security controls [7].

The study by Rohan H. Shah and Prof. D. P. Salapurkar proposes a multifactor authentication system using secret splitting in the context of the Cloud of Things (CoT). The authors aim to address the security challenges associated with CoT, where a large number of devices are connected to the cloud and require secure authentication mechanisms [9].

The proposed architecture uses a combination of a public cloud and a private cloud, with the private cloud acting as a gateway for IoT devices to communicate with the public cloud. The authors also use secure communication protocols, such as Transport Layer Security (TLS), to ensure the confidentiality and integrity of data transmitted between IoT devices and the cloud [10].

The proposed protocol uses a combination of symmetric key encryption and hash functions to ensure mutual authentication between the IoT device and the server. The protocol also uses a lightweight message exchange mechanism to reduce the computational overhead and energy consumption of the IoT device [11].

The study identifies several common security failures in cloud computing, including data breaches, denial-of-service attacks, insider threats, and misconfiguration of cloud resources. The author also discusses the potential impact of these security failures, such as financial loss, reputation damage, and legal and regulatory consequences [12].

The study emphasizes the need for a comprehensive and integrated approach to cloud security, where different security components work together to provide a robust security framework. The author recommends that organizations adopt a security-first mindset and prioritize security measures when designing and deploying cloud-based solutions [13].

The authors propose the use of cryptographic techniques to further enhance the security of cloud servers. They propose the use of symmetric key encryption and hashing algorithms to protect sensitive data stored on cloud servers. These techniques can provide an additional layer of security, making it much more difficult for attackers to access or tamper with sensitive data [14].

The study identifies the limitations of traditional authentication methods, such as passwords and PINs, which can be easily compromised or stolen. The author proposes a new authentication scheme that combines biometric authentication with cryptographic techniques to enhance the security of cloud-based systems [15].

The study finds that while traditional authentication techniques such as passwords are commonly used in IoT systems, they are susceptible to various security threats such as phishing attacks and password guessing. The author suggests that biometric authentication can provide a more secure and reliable authentication mechanism for IoT systems, as it relies on unique physiological or behavioural characteristics that are much harder to forge or steal [16].

The authors survey several secure data sharing protocols used in edge-cloud IoT systems, such as key management protocols, secure communication protocols, and access control protocols. The study evaluates the effectiveness of these protocols in addressing the security challenges of IoT data sharing and identifies their strengths and limitations [17].

The proposed scheme involves three phases: registration, authentication, and communication. During the registration phase, the IoT device and the cloud server exchange their identity information and establish a shared secret key. In the authentication phase, the IoT device sends a challenge message to the cloud server, which responds with a valid response message. The IoT device then verifies the response message using the shared secret key. Finally, in the communication phase, the IoT device and the cloud server exchange data using the shared secret key and hash functions [18].

The proposed scheme utilizes a combination of Public Key Infrastructure (PKI) and elliptic curve cryptography (ECC) to establish a secure communication channel between IoT devices and the cloud server. The scheme involves two phases: registration and authentication [19].

he paper categorizes IoT authentication schemes into five groups: symmetric key-based, asymmetric key-based, hybrid authentication, physically unclonable functions (PUFs), and other techniques [20].

The paper categorizes these threats into a taxonomy spanning authentication, access control, data protection, infrastructure security, and accountability. Proposed solutions include multi-factor authentication, risk-based adaptive access control, data encryption, key management, network segmentation, anomaly detection, and logging/auditing [21].

The paper proposes a secure authentication scheme for IoT devices to connect to cloud servers. It uses a combination of symmetric and asymmetric cryptography to balance security and efficiency. Devices are registered with the server using an asymmetric key pair. Shared keys are generated for future authentication. The scheme uses nonces and random numbers to prevent replay attacks. Devices authenticate to servers using a challenge-response protocol with the shared key. Session keys are generated for secure communication.

The scheme provides mutual authentication between devices and servers [22].

The taxonomy provides a framework to understand the security-performance trade-offs between different categories like password-based, challenge-response, PUFs, etc. And the analysis of factors like computational overhead, storage needs, scalability, and susceptibility to various attacks offers insights into selecting the right schemes for specific IoT use cases and constraints [23].

The paper provides a comprehensive security checklist covering core technical, operational, and organizational considerations critical for secure cloud-backed IoT deployments [24].

The paper examines the primary risks to confidentiality in the cloud and both preventative and detective controls to mitigate them based on encryption, access controls, secure enclaves, auditing, and standards/certifications [25].

The paper provides a taxonomy of data leakage threats in the cloud and reviews techniques for prevention, detection, and response to safeguard confidential data [26].

3. Attacks in Cloud IoT:

1) Replay attack:

- IoT devices often communicate with cloud services over the internet using protocols like MQTT. This communication is not inherently secure or encrypted.
- An attacker can eavesdrop on the network traffic between an IoT device and the cloud service. They can then capture legitimate sensor data or control messages being sent between the devices.
- Later on, the attacker can resend the same captured data/messages again to the cloud service or IoT device. As far as the receiver is concerned, it looks like valid data coming from a legitimate source.
- For example, an attacker could capture a temperature sensor reading of 22 degrees being transmitted from a smart thermostat to the cloud. The attacker can then replay this same temperature message later on, essentially impersonating the thermostat(Figure 1).
- This could impact analytics in the cloud by providing false data. Or could trigger incorrect control actions on the IoT devices in response to the replayed commands.
- To prevent replay attacks, communications between IoT devices and cloud services should be encrypted and include nonce or timestamps to detect replayed messages. Authentication should also be used to verify the identity of the sender.

- In summary, lack of encryption and authentication in IoT-cloud communications enables network capture and replay of messages which can be used to spoof identities or corrupt data analytics. Proper security protocols need to be implemented to detect and prevent such replay attacks.

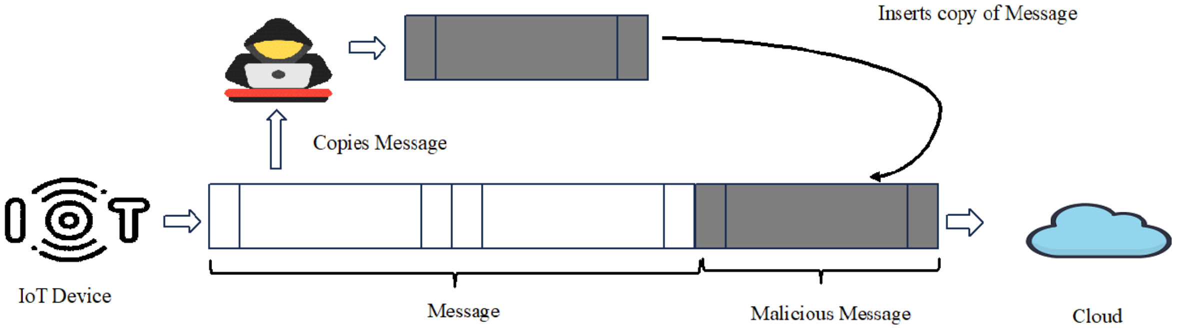


Figure 1. Replay Attack in Cloud - IoT Communication

2) Masquerade Attack

- In a masquerade attack, an unauthorized entity impersonates the identity of an authorized device or user to gain access to the system and evade detection.
- In cloud IoT, an attacker can spoof the identity of a legitimate IoT sensor and send falsified data to the cloud platform acting as that spoofed device.
- For example, an attacker can pretend to be a smart utility meter by using the meter's valid credentials and transmit incorrect readings to the utility company's cloud analytics platform(Figure 2).
- The cloud platform or IoT gateway is unable to distinguish between the fake and real device as the credentials are valid. So incorrect data gets ingested into the system.
- This can lead to wrong analytics and decisions taken based on the falsified data. The masquerading device identity also makes detection difficult.
- Strong authentication using digital certificates or tokens should be implemented to validate device identities. Encrypted communications are also necessary to prevent snooping of credentials.
- Access control policies should be refined to only allow trusted device identities to connect to the cloud services. Monitoring for abnormal traffic patterns can also help detect potential masquerade attacks.
- In summary, the lack of strong identity verification in IoT-cloud communications enables masquerade attacks which can inject false data to evade detection. Proper authentication, authorization and encryption mechanisms are required to thwart such threats.

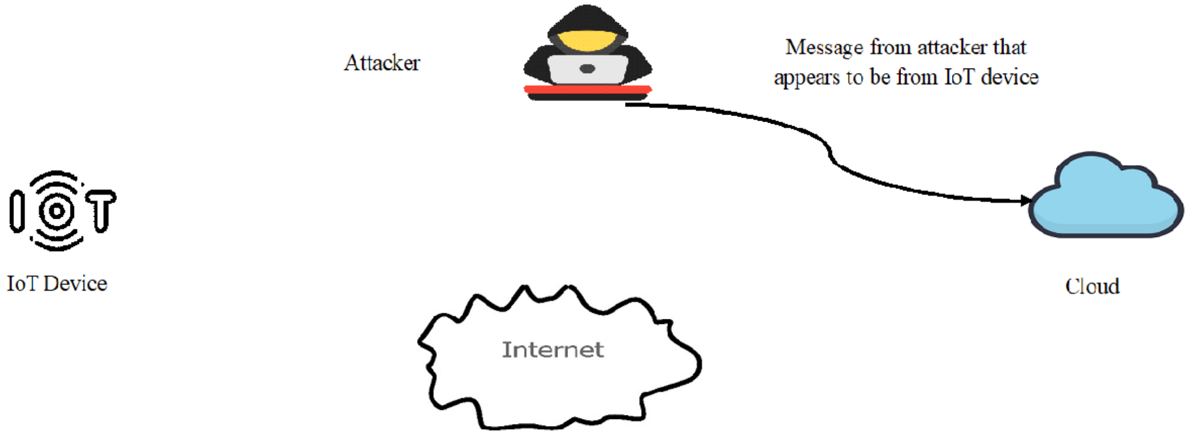


Figure 2. Masquerade Attack in Cloud - IoT Communication

3) Message Forgery Attack:

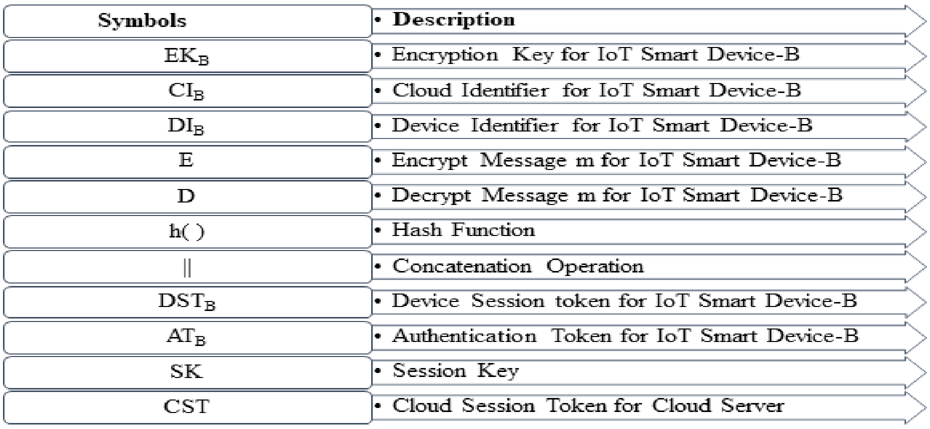
Cloud IoT or Internet of Things message forgery is when bad guys twist or make up messages to trick the system. These attacks are a big problem because they can let someone sneak in without permission, mess up data, or do things they aren't supposed to on an IoT network.

Security measures can be used to mitigate messaging network attacks in cloud IoT networks, e.g.

- **Message Authentication Codes (MAC):** The use of MACs helps ensure the integrity and authenticity of a message by adding a cryptographic tag to each message, allowing the receiver to verify its origin.
- **Device compliance:** Making sure devices are authenticated before engaging in communication helps prevent unauthorized companies from sending spoofed messages.
- **Access Control:** Implementing robust access control mechanisms helps restrict unauthorized access to devices and communication channels, preventing malicious entities from tampering with messages.

4.Proposed Mechanism

Figure 3 represents the symbols and description of our proposed mechanism and Figure 4 represents the proposed lightweight key establishment.



Symbols	• Description
EK_B	• Encryption Key for IoT Smart Device-B
CI_B	• Cloud Identifier for IoT Smart Device-B
DI_B	• Device Identifier for IoT Smart Device-B
E	• Encrypt Message m for IoT Smart Device-B
D	• Decrypt Message m for IoT Smart Device-B
h( )	• Hash Function
||	• Concatenation Operation
DST_B	• Device Session token for IoT Smart Device-B
AT_B	• Authentication Token for IoT Smart Device-B
SK	• Session Key
CST	• Cloud Session Token for Cloud Server

Figure 3. Notations and Description

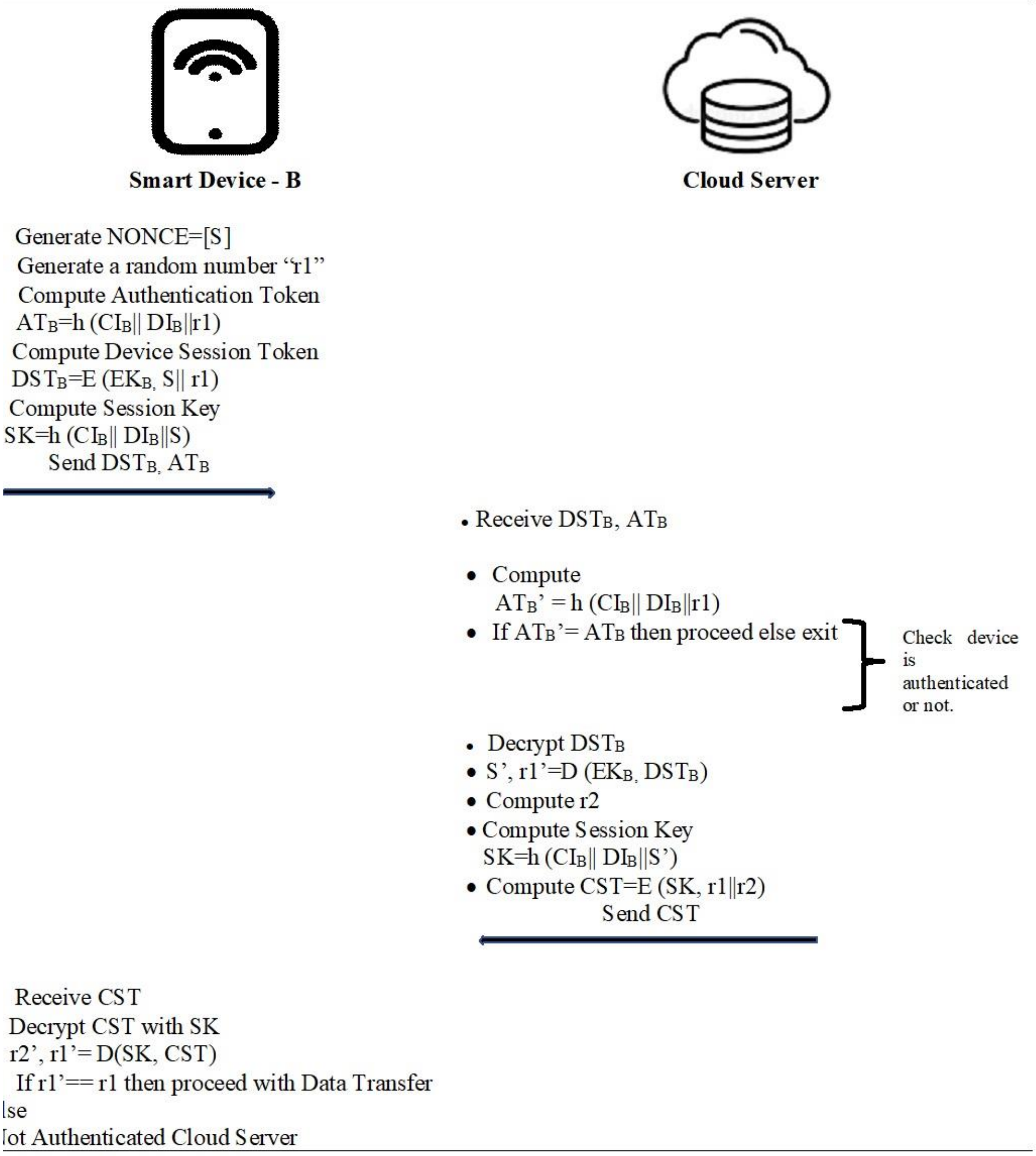

Smart Device - B
Cloud Server
Generate NONCE=[S]
Generate a random number "r1"
Compute Authentication Token
AT_B=h (CI_B|| DI_B||r1)
Compute Device Session Token
DST_B=E (EK_B, S|| r1)
Compute Session Key
SK=h (CI_B|| DI_B||S)
Send DST_B, AT_B
• Receive DST_B, AT_B
• Compute
AT_B' = h (CI_B|| DI_B||r1)
• If AT_B'= AT_B then proceed else exit
Check device is authenticated or not.
• Decrypt DST_B
• S', r1'=D (EK_B, DST_B)
• Compute r2
• Compute Session Key
SK=h (CI_B|| DI_B||S')
• Compute CST=E (SK, r1||r2)
Send CST
Receive CST
Decrypt CST with SK
r2', r1'= D(SK, CST)
If r1'== r1 then proceed with Data Transfer
lse
Iot Authenticated Cloud Server

Figure 4. Proposed lightweight key establishment mechanism to detect various attacks

The goal of the suggested key setup method is to offer a simple and safe way to exchange keys between cloud servers and smart Internet of things devices. By employing AES (advanced encryption standard) for data encryption and decryption and by authenticating the cloud server and IoT devices prior to communication, the suggested approach eliminates security risks.

To guarantee the protection of the send, the suggested technique generates a new current session key for each new current session between the Smart IoT Device and cloud server. Furthermore, each smart IoT device and cloud server is given a unique identification that is used to authenticate them before any connection occurs, hence increasing the difficulty of unauthorized entities gaining access to the network.

In order to add randomness to the data and make it more difficult for an attacker to predict or guess the authentication token, the technique also makes use of random nonce and random numbers. The

Smart IoT device transfers the necessary tokens for authentication and device sessions to the cloud server. The cloud server then uses the encryption key, nonce, and random number to decrypt the Device Session Token and confirms the authenticity of the tokens.

The cloud server receives the token from the device session, decrypts it, and compares it to the authentication token that was computed there. The Internet of Things device is regarded as securely authorized if the tokens match.

The session key is computed at the cloud server and sent to the smart IoT device once authentication is complete. The device decrypts the session key, which is then used to verify the authenticity of the cloud server. This implies that secure communication is only permitted for authenticated entities.

- The system can be made more secure by creating a NONCE[S]—a unique value used for authentication—at the smart IoT device (SD-B) and adding a "r1"—a random number—into the message authentication token computation.

where AT_B=Authentication Token of Smart IoT Device B and h (CI_B|| DI_B || r1).

- The system can guarantee that the token is unique for each message and difficult for an attacker to guess or predict by incorporating the cloud identifier (CI_B) of the cloud server to which the IoT device is connected, the smart device identifier (DI_B) of the smart IoT device, and the "r1" random number in the data being hashed.
- Adding "r1" with the hash function creates randomness and guards against attacks.
- The smart IoT device (SD-B) can create a secure session between itself and the cloud server by computing a DST_B (Device Session Token) using an encryption key (EK_B) with a NONCE=S and a random number=r1.
- The information is kept private and can only be decrypted by authorized entities with the encryption key (EK_B) by encrypting the data (S) and the "r1" (random number).

E(EK_B, S||r1) = DST_B where Device Session Token Calculation (DST_B).

- The Device Session (DST_B) and Authentication (AT_B) tokens produced by the Smart IoT device (SD-B) are sent to the Cloud server.
- Using the encryption key, the Cloud server decrypts the DST_B to provide the random integer (r1') and the nonce (S').

S', r1' = D (DST_B, EK_B)

- Using the Cloud Identifier (CI_B), Device Identifier (DI_B), and random number (r1), the Cloud server computes the Authentication Token (AT_B') and compares it with the received Authentication Token (AT_B). The smart IoT device is authenticated if AT_B' = AT_S; if the values differ, the smart IoT device is not reliable or authentic.

(CI_B || DI_B || r1) = AT_B' = h

- The Cloud server uses a hash function with the Cloud Identifier (CI_B), Device Identifier (DI_B), and nonce (S') to calculate the Session Key (SK) and generates a new random number (r2).

(CI_B || DI_B || S') = h for SK

- Using the generated session key (SK), the random numbers (r1) and (r2) as input, the cloud server calculates the Cloud Server Token (CST) and encrypts this data.

E (SK, r1||r2) = CST

- The CST is sent to the IoT device (SD-B) via the cloud server.
- The Smart IoT device can be authenticated by the Cloud server using these methods, and a secure session key can be established for communication between the Smart IoT device and Cloud Server. The Cloud Server Token (CST) serves as a means of confirming the authenticity of the Cloud Server.
- The Internet of Things device will use the Session Key (SK) to decrypt the Cloud Session Token (CST) after receiving it from the Cloud Server.
  D(SK, CST) = r1', r2'.
- In order to prevent any potential security issues or malicious access, the IoT device will detect that the Cloud Server is not authenticated if the random numbers do not match and will not proceed with the data transfer.
- **Assessment of Security**

The suggested approach additionally ensures resilience against potential attacks such as replay attack, masquerade attack, message forgery attack. We elaborate on the scheme's capacity to thwart these attacks in the subsequent discussion:

1) **Replay Attack:**

- **Scenario:**

  Let's say that tom records the request message {DST_B,AT_B} that SD-B sends to CS and replays it at predetermined intervals.
- **Resilience:**
- The random nonce S is generated and stored by the SD.
- The CS receives the generated random nonce S in an encrypted manner.
- The copy of the received nonce is retained by CS.
- Because the CS already knows the received nonce, it rejects the replayed message in a replay packet.

2) **Masquerade attack:**

The authentication process on the cloud server involves generating an Authentication Token (AT_B') by combining the Cloud Identifier (CI_B), Device Identifier (DI_B), and a random number (r1) using a hashing function (h). The cloud server then checks if the calculated AT_B' matches the received Authentication Token (AT_B). If the two values are equal, the Smart IoT Device is considered authenticated and trustworthy. Conversely, if the values do not match, the Smart IoT device is deemed untrusted or not authentic.

- AT_B=h (CI_B|| DI_B||r1)
- **Scenario:**
- Let's say that tom send the authentication token as AT_{tom}

- AT_{tom}=h (CI_{tom}|| DI_{tom}||r1)
- **Resilience:**
- Since we have to check AT_B=AT'_B if tom is sending the authentication token AT_{tom} then it will not be equal
- AT'_B! = AT_{tom}

3) **Message Forgery Attack**

- **Scenario:**
- Message forgery is the sending of a message to deceive the recipient as to whom the real sender is.
- Tom does not have CI and DI and puts some random CI and DI values

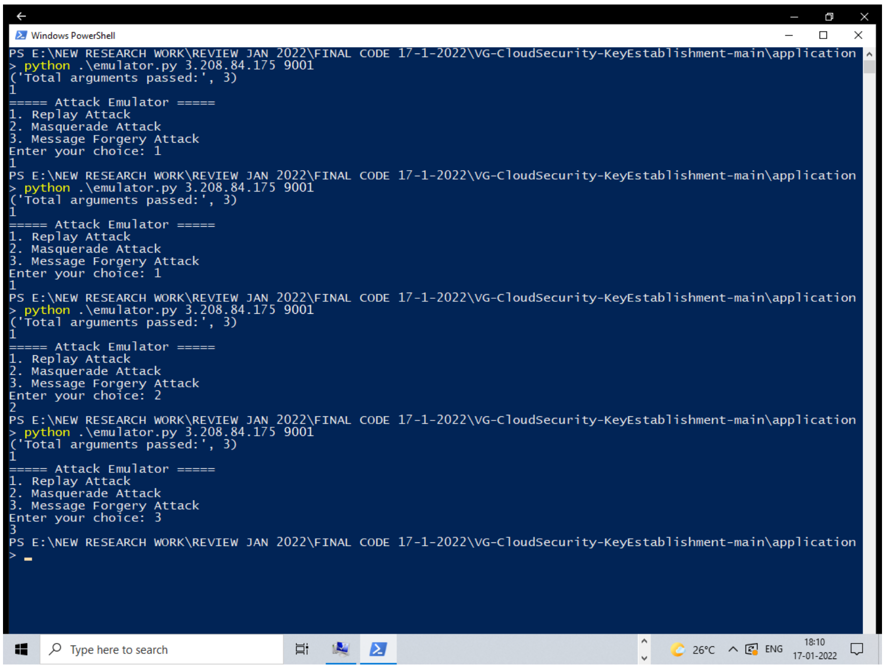
  AT_{tom}=h (CI_{tom}|| DI_{tom}||$r1_{tom}$)

  DST_{tom}=E ($EK_{tom,}$ S|| $r1_{tom}$)
- Since we have to check AT_B=AT'_B if tom is sending the authentication token AT_{tom} then it will not be equal
- AT'_B! = AT_{tom}

5.Results and Discussion

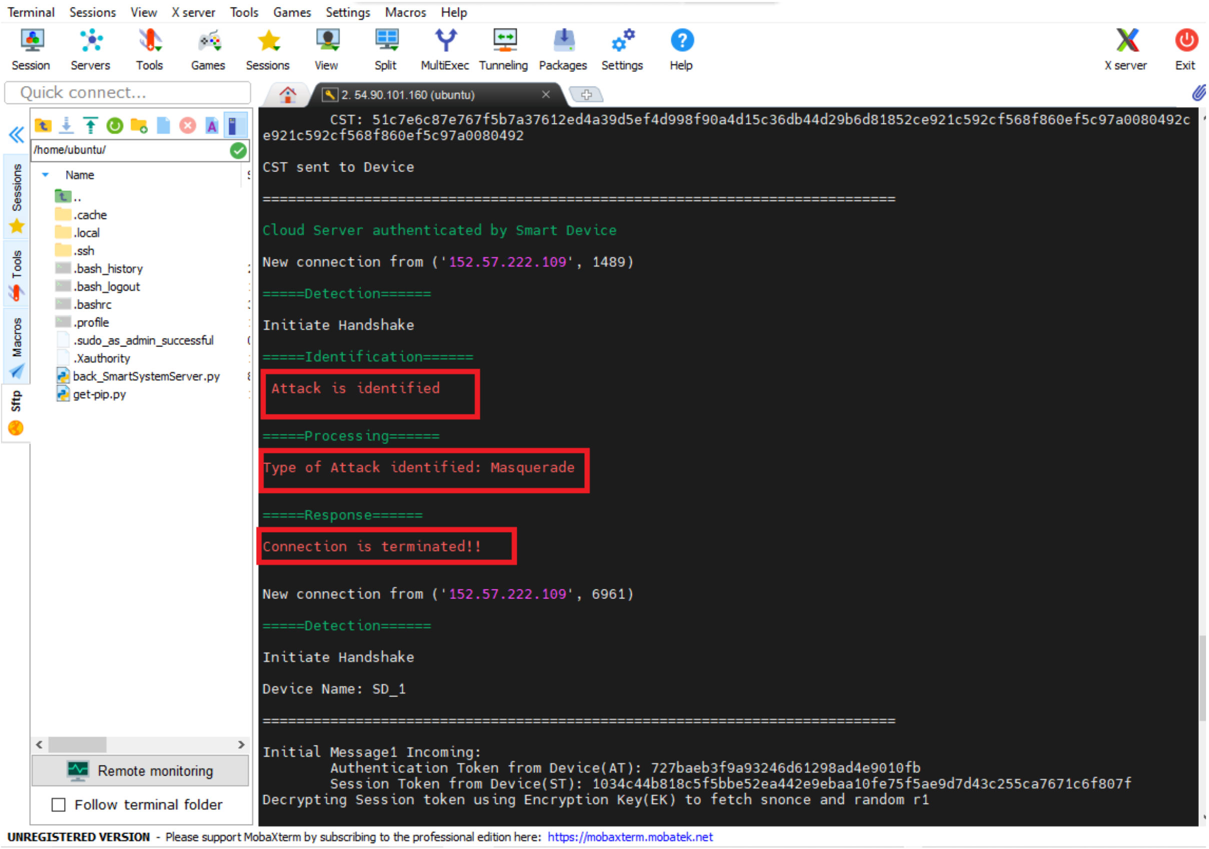
In the PowerShell window the menu of Attack Emulator like Replay Attack, Masquerade Attack and Message Forgery Attack gets displayed(Figure 5).

If we choose any one of the options, then the message “Attack is identified”, and displays “Type of Attack Identified: ______”, will get displayed and the “connection gets terminated”(Figure 6).



```
PS E:\NEW RESEARCH WORK\REVIEW JAN 2022\FINAL CODE 17-1-2022\VG-CloudSecurity-KeyEstablishment-main\application
> python .\emulator.py 3.208.84.175 9001
('Total arguments passed:', 3)
1
===== Attack Emulator =====
1. Replay Attack
2. Masquerade Attack
3. Message Forgery Attack
Enter your choice: 1
1
PS E:\NEW RESEARCH WORK\REVIEW JAN 2022\FINAL CODE 17-1-2022\VG-CloudSecurity-KeyEstablishment-main\application
> python .\emulator.py 3.208.84.175 9001
('Total arguments passed:', 3)
1
===== Attack Emulator =====
1. Replay Attack
2. Masquerade Attack
3. Message Forgery Attack
Enter your choice: 1
1
PS E:\NEW RESEARCH WORK\REVIEW JAN 2022\FINAL CODE 17-1-2022\VG-CloudSecurity-KeyEstablishment-main\application
> python .\emulator.py 3.208.84.175 9001
('Total arguments passed:', 3)
1
===== Attack Emulator =====
1. Replay Attack
2. Masquerade Attack
3. Message Forgery Attack
Enter your choice: 2
2
PS E:\NEW RESEARCH WORK\REVIEW JAN 2022\FINAL CODE 17-1-2022\VG-CloudSecurity-KeyEstablishment-main\application
> python .\emulator.py 3.208.84.175 9001
('Total arguments passed:', 3)
1
===== Attack Emulator =====
1. Replay Attack
2. Masquerade Attack
3. Message Forgery Attack
Enter your choice: 3
3
PS E:\NEW RESEARCH WORK\REVIEW JAN 2022\FINAL CODE 17-1-2022\VG-CloudSecurity-KeyEstablishment-main\application
> _
```

Figure 5. Windows PowerShell displaying menu of replay, masquerade, message forgery attack.


Terminal Sessions View X server Tools Games Settings Macros Help
Session
Servers
Tools
Games
Sessions
View
Split
MultiExec
Tunneling
Packages
Settings
Help
X server
Exit
Quick connect...
2. 54.90.101.160 (ubuntu)
/home/ubuntu/
Name
..
.cache
.local
.ssh
.bash_history
.bash_logout
.bashrc
.profile
.sudo_as_admin_successful
.Xauthority
back_SmartSystemServer.py
get-pip.py
Sessions
Tools
Macros
Sftp
Remote monitoring
Follow terminal folder
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

```
        CST: 51c7e6c87e767f5b7a37612ed4a39d5ef4d998f90a4d15c36db44d29b6d81852ce921c592cf568f860ef5c97a0080492c
e921c592cf568f860ef5c97a0080492

CST sent to Device

===========================================================================

Cloud Server authenticated by Smart Device

New connection from ('152.57.222.109', 1489)

=====Detection======

Initiate Handshake

=====Identification======

 Attack is identified

=====Processing======

Type of Attack identified: Masquerade

=====Response======

Connection is terminated!!

New connection from ('152.57.222.109', 6961)

=====Detection======

Initiate Handshake

Device Name: SD_1

===========================================================================

Initial Message1 Incoming:
        Authentication Token from Device(AT): 727baeb3f9a93246d61298ad4e9010fb
        Session Token from Device(ST): 1034c44b818c5f5bbe52ea442e9ebaa10fe75f5ae9d7d43c255ca7671c6f807f
Decrypting Session token using Encryption Key(EK) to fetch snonce and random r1
```

Figure 6.Displays “Attack is identified”, “Type of Attack Identified” ,and terminates the connection .

Table 1 Security comparison of the proposed protocol with other existing protocols

Reference Mechanism	**Security Features**				
	Mutual Authentication	**Replay Attack**	**Masquerade Attack**	**Message Forgery Attack**	**Session Key Agreement**
Panda et al.[28]	Yes	Yes	No	No	Yes
Roy et al.[29]	Yes	Yes	No	No	Yes
Biswas et al [30]	Yes	Yes	No	No	Yes
Hsiao et al[31]	No	Yes	No	No	Yes
Sood et al[32]	No	Yes	No	No	No
WuH-L et al[33]	Yes	Yes	No	No	No
Chen C-M et al[34]	Yes	Yes	No	No	No
Karuppiah et al[35]	Yes	Yes	No	No	No
Ananth et al[36]	Yes	Yes	No	No	No
Proposed Mechanism	Yes	Yes	Yes	Yes	Yes

Yes: stops the attack or backs a certain characteristic; No: cannot stop the attack or doesn't support a protocol feature

6.Discussion

6.1 Cost of Computation

Table 2 shows how the computation costs of our proposed mechanism compare to those of other methods. Two hashing functions, one message authentication code, one encryption, and one decryption are required for cryptographic operations in our suggested mechanism.

Table 2: Comparison of Computation cost with other protocol/mechanism

Reference Mechanism	**Hash Function**	**Message Authentication Code**	**Cryptosystem (Number of Encryption and Decryption)**
Y.Li et al[37]	1H	1 MAC	1E+1D
B.Vaidya et al[38]	4H	-	-
K. Han et al[39]	5H	7 MAC	4E+ED
Jebri et al[40]	4H	-	1E+1D
Proposed Mechanism	2H	1 MAC	1E+1D

Below is a summary of the general conclusions drawn from the aforementioned analysis:

1. In contrast to protocol [30], the suggested protocol accomplishes mutual authentication.

2. Compared to previous protocols [30, 31, 32–36], the suggested protocol achieves higher security.

3. In terms of computing overhead, the suggested protocol performs better than the protocols [30, 31, 32].

4. The proposed protocol achieves better security than the protocols [28-36] and achieves forward secrecy through the session key agreement between SD-B and CS, which is not feasible in the protocols [32- 36].

5.As a result, the computational overhead of the proposed protocol is slightly different than the protocols [37-40].

7.Conclusion

This work proposes a lightweight key establishment mechanism for cloud servers and the Internet of Things. Previous associated authentication mechanisms that were in place for cloud servers and the Internet of Things did not meet the requisite security standards.

Unlike protocol [30], mutual authentication is achieved by the proposed protocol. The proposed protocol achieves stronger security than earlier protocols [30, 31, 32–36]. The proposed protocol outperforms the methods [30, 31, 32] in terms of computing overhead.

The suggested protocol outperforms the protocols [28–36] in terms of security and accomplishes forward secrecy by means of the session key agreement between SD-B and CS, something that the protocols [32–36] are unable to accomplish.

As a result, the suggested protocol's computational overhead differs slightly from the protocols [37–40].

Thus, it can be said that our suggested protocol has the ability to offer a more advanced safe mutual authentication paradigm for cloud server and IoT smart devices.

Our work can be expanded in the future to significantly reduce the computational cost and total computing time of the suggested protocol without compromising security. In order to provide users with prior knowledge about the system's behaviors and reliabilities before utilizing the model, we would also like to create the behavior and reliability model for the suggested protocol. The suggested protocol can be used in any Internet of Things industry where connecting embedded devices and cloud servers requires data security and authentication as a top priority.

REFERENCES

[1] R. Barona and E. A. M. Anita, "A survey on data breach challenges in cloud computing security: Issues and threats," 2017 International Conference on Circuit ,Power and Computing Technologies (ICCPCT), Kollam, India, 2017, pp. 1-8, doi: 10.1109/ICCPCT.2017.8074287.

[2] Panda, P.K., Chattopadhyay, S. A secure mutual authentication protocol for IoT environment. J Reliable Intell Environ 6, 79–94 (2020). https://doi.org/10.1007/s40860-020-00098-y

[3] Panda, P.K., Chattopadhyay, S. A secure mutual authentication protocol for IoT environment. J Reliable Intell Environ 6, 79–94 (2020). https://doi.org/10.1007/s40860-020-00098-y

[4] J. Liranzo and T. Hayajneh, "Security and privacy issues affecting cloud-based IP camera," 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON), New York, NY, USA, 2017, pp. 458-465, doi: 10.1109/UEMCON.2017.8249043.

[6] D. Zhe, W. Qinghong, S. Naizheng and Z. Yuhan, "Study on Data Security Policy Based on Cloud Storage," *2017 ieee 3rd international conference on big data security on cloud (bigdatasecurity), ieee international conference on high performance and smart computing (hpsc), and ieee international conference on intelligent data and security (ids)*, Beijing, China, 2017, pp. 145-149, doi: 10.1109/BigDataSecurity.2017.12.

[7] Elkafrawy, Passent & Abdo, A.A. & Shawish, Amr. (2015). Security Issues Over Some Cloud Models. Procedia Computer Science. 65. 853-858. 10.1016/j.procs.2015.09.041.

[9] R. H. Shah and D. P. Salapurkar, "A multifactor authentication system using secret splitting in the perspective of Cloud of Things," *2017 International Conference on Emerging Trends & Innovation in ICT (ICEI)*, Pune, India, 2017, pp. 1-4, doi: 10.1109/ETIICT.2017.7977000.

[10] A. Sharma, T. Goyal, E. S. Pilli, A. P. Mazumdar, M. C. Govil and R. C. Joshi, "A Secure Hybrid Cloud Enabled architecture for Internet of Things," 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), Milan, Italy, 2015, pp. 274-279, doi: 10.1109/WF-IoT.2015.7389065.

[11] M. N. Aman, K. C. Chua and B. Sikdar, "A Light-Weight Mutual Authentication Protocol for IoT Systems," *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Singapore, 2017, pp. 1-6, doi: 10.1109/GLOCOM.2017.8253991.

[12] Z. P. Gariba and J. A. Van Der Poll, "Security Failure Trends of Cloud Computing," 2017 IEEE 3rd International Conference on Collaboration and Internet Computing (CIC), San Jose, CA, USA, 2017, pp. 247-256, doi: 10.1109/CIC.2017.00041.

[13] Khanna, Abhirup & Arora, Akshay & Rastogi, Anmol & Agarwal, Amit. (2017). Cloud Security Ecosystem for Data Security and Privacy. 10.1109/CONFLUENCE.2017.7943164.

[14] Khalid, Zarnab & Rizwan, Muhammad & Shabbir, Aysha & Shabbir, Maryam & Ahamd, Fahad & Manzoor, Jaweria. (2019). Cloud Server Security using Bio-Cryptography. International Journal of Advanced Computer Science and Applications. 10. 10.14569/IJACSA.2019.0100321.

[15] T. -F. Cheng, "Comments on the authentication scheme for IoT and cloud servers," 2016 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), Nantou, Taiwan, 2016, pp. 1-2, doi: 10.1109/ICCE-TW.2016.7520711.

[16] M. El-hajj, M. Chamoun, A. Fadlallah and A. Serhrouchni, "Analysis of authentication techniques in Internet of Things (IoT)," 2017 1st Cyber Security in Networking Conference (CSNet), Rio de Janeiro, Brazil, 2017, pp. 1-3, doi: 10.1109/CSNET.2017.8242006.

[17] Prasanthi, K & Sagar, K v. (2018). Survey on secure protocols for data sharing through edge of cloud assisted internet of things. International Journal of Engineering & Technology. 7. 92. 10.14419/ijet.v7i2.7.10267.

[18] Chandra Babu, Gokulnath & Gandhi, Usha & Balan, E. & M K, Priyan. (2015). Mutual Authentication Scheme for IoT Application. Indian Journal of Science and Technology. 26. 10.17485/ijst/2015/v8i26/80996.

[19] Park, N.; Kang, N. Mutual Authentication Scheme in Secure Internet of Things Technology for Comfortable Lifestyle. *Sensors* **2016**, *16*, 20. https://doi.org/10.3390/s16010020

[20] K. S. Roy and H. K. Kalita, "A Survey on Authentication Schemes in IoT," *2017 International Conference on Information Technology (ICIT)*, Bhubaneswar, India, 2017, pp. 202-207, doi: 10.1109/ICIT.2017.56.

[21] Habiba, U., Masood, R., Shibli, M.A. *et al.* Cloud identity management security issues & solutions: a taxonomy. *Complex Adapt Syst Model* **2**, 5 (2014). https://doi.org/10.1186/s40294-014-0005-9

[22] Sheetal Kalra, Sandeep K. Sood,Secure authentication scheme for IoT and cloud servers, Pervasive and Mobile Computing,Volume 24, 015,Pages 210-223, ISSN 1574-1192, https://doi.org/10.1016/j.pmcj.2015.08.001.(https://www.sciencedirect.com/science/article/pii/S1574119215001510).

[23] Sarika, K., Hiremath, P.S. A Survey of Internet of Things (IoT) Authentication Schemes. In: Hiremath P., Kakkasageri M., Manvi S., Mahapurush A. (eds) Emerging Research in Computing, Information, Communication and Applications. ERCICA 2020. Communications in Computer and Information Science, vol 1325. Springer, Singapore. https://doi.org/10.1007/978-981-15-8893-6_58.

[24] M. Shahzad and M. P. Singh, "Continuous Authentication and Authorization for the Internet of Things," in *IEEE Internet Computing*, vol. 21, no. 2, pp. 86-90, Mar.-Apr. 2017, doi: 10.1109/MIC.2017.33.

[25] J. Singh, T. Pasquier, J. Bacon, H. Ko and D. Eyers, "Twenty Security Considerations for Cloud-Supported Internet of Things," in *IEEE Internet of Things Journal*, vol. 3, no. 3, pp. 269-284, June 2016, doi: 10.1109/JIOT.2015.2460333.

[26] R. Padilha and F. Pedone, "Confidentiality in the Cloud," in *IEEE Security & Privacy*, vol. 13, no. 1, pp. 57-60, Jan.-Feb. 2015, doi: 10.1109/MSP.2015.4.

[27] Allaf, Zirak & Adda, Mo. (2017). Review of data leakage attack techniques in cloud systems.

[28] Panda, P.K., Chattopadhyay, S. A secure mutual authentication protocol for IoT environment. *J Reliable Intell Environ* **6**, 79–94 (2020). https://doi.org/10.1007/s40860-020-00098-y

[29] K. S. Roy and H. K. Kalita, "A Survey on Authentication Schemes in IoT," *2017 International Conference on Information Technology (ICIT)*, Bhubaneswar, India, 2017, pp. 202-207, doi: 10.1109/ICIT.2017.56.

[30] Hafizul SK, Biswas GP (2011) A more efficient and secure IDbased remote mutual authentication with key agreement scheme for mobile devices on elliptic curve crypto systems. J Syst Softw 84(11):1892–1898

[31] Liao YP, Hsiao CM (2014) A secure ECC-based RFID authentication scheme integrated with ID-verifier transfer protocol. Ad Hoc Netw 18:133–146

[32] Kalra S, Sood SK (2015) Secure authentication scheme for IOT and cloud servers. Pervasive Mob Comput 24:210–223

[33] Chang C-C, Wu H-L, Sun C-Y (2017) Notes on secure authentication scheme for IOT and cloud servers. Pervasive Mob Compute 38:275–278

[34]Wang K-H, Chen C-M, Fang W, Wu T-Y (2017) A secure authentication scheme for internet of things. Pervasive Mob Comput 42:15–26

[35] Kumari S, Karuppiah M, Das AK (2018) A secure authentication scheme based on elliptic curve cryptography for IoT and cloud servers. J Supercomput 74:6428–6453

[36] Bhubaneswari S, Ananth NV (2018) Enhanced mutual authentication scheme for cloud of things. Int J Pure Appl Math 119(15):1571–1583

[37] Y. Li, “Design of a key establishment protocol for smart home energy management system,” in Proc. 5th Int. Conf. Comput. Intell., Commun. Syst. Netw. (CICSyN), Jun. 2013, pp. 88–93

[38] B. Vaidya, D. Makrakis, and H. T. Mouftah, “Device authentication mechanism for smart energy home area networks,” in Proc. IEEE Int. Conf. Consum. Electron. (ICCE), Jan. 2011, pp. 787–788.

[39] K. Han, J. Kim, T. Shon, and D. Ko, “A novel secure key paring protocol for RF4CE ubiquitous smart home systems,” Pers. Ubiquitous Comput., vol. 17, no. 5, pp. 945–949, Jun. 2013

[40] Jebri, S., Ben Amor, A., Abid, M. et al. Enhanced Lightweight Algorithm to Secure Data Transmission in IoT Systems. Wireless Pers Commun 116, 2321–2344 (2021). https://doi.org/10.1007/s11277-020-07792-3