The Seybold REPORT

ISSN 1533-9211

DOI 10.5281/zenodo.7152683

MODIFIED COAP TO ENSURE AUTHENTICATION IN IOT NETWORK

NIDHI DANDOTIYA[1] and Dr. PALLAVI KHATRI[2]

[1, 2] CSA Deptt, ITM University, Gwalior, M.P., India.

Abstract

The IoT environment must be capable of inter connecting large number of heterogeneous devices. There have been numerous attempts by researchers to build countermeasures unique to IoT layers and devices in order to solve the security concerns raised. With a focus on standardized communication protocols, investigate security concerns of the internet of things business. An attack on a device's web or mobile application can allow attackers to gain access to its credentials, making IoT security a major concern when the technology is implemented. This work proposes a safe authentication strategy based on COAP Protocol for use with an IoT security system that aims to improve authentication.

Keywords: IoT, COAP, Hash Function, MD5

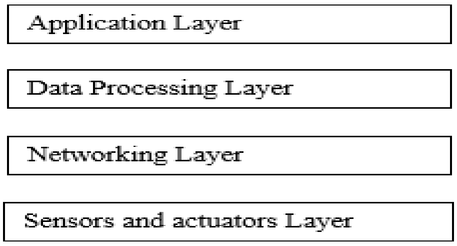
1. INTRODUCTION

The IoT is a collection of devices that include actuators and sensors as well as embedded and distributed computer capabilities to make jobs more efficient and precise. It has a significant influence on our everyday life. Its applications include smart city and industrial deployments. Smart home, automation, healthcare, emergency response and transportation [1, 2].

1.1 Architecture of IOT

The Internet of Things ecosystem should be capable of networking a large number of diverse items. As a result, layered architecture (shown in Fig 1) must be flexible and adaptable. Every layer in the IoT is characterized by the functions and applications that it employs. The bottom layer is made up of smart objects that have sensors built in. The se nsors allow thephysical and digital worlds to be connected, permitting realtime data to be composed and managed [3] This layer uses different type sensors for various application [4].A strong and high-performance wired or wireless network infrastructure is needed as a transport medium for the massive volume of data that tiny sensors will generate at the network layer [5]. The Data Processing Layer enables information processing, device management etc [6, 7]. IoT application layer work covers various "smart" environments for example smart car, smart city , IoT Agriculture, Supply chain, Healthcare, Tourism and Safety. [8].

Figure 1: Internet of things Layers



1.2 Authentication in IOT

Today, IoT devices are more vulnerable to security breaches because most of them lack adequate countermeasures. IoT device authentication must be distinct and significantly lighter than existing user or personal authentication methods that are not directly applicable to IoT devices with limited resources [9, 10]. As a result, selecting the proper authentication method is critical to ensuring robust security for IoT devices. The most basic form of IoT device authentication is single-factor authentication, in which devices or users present something they know to verify their identity. Two-factor authentication, on the other hand, extends one-factor authentication of usernames/passwords by adding another layer in which users or devices must verify something they own. At the time of IoT device manufacturing manufacturers generally don’t include security features so which authentication technique use to protect data and unauthorized user information must know by the IT administrators [11].IT administrators can choose different authentication method from token based ,two factor, one factor authentication etc.[12,13,14]

1.3 Threats in IOT Network

IoT security used to take critical data from others such as social engineering, DOS, DDOS etc.

Organization and people need to aware of following IoT security threats-

A network of systems known as a "botnet" is used to remotely manipulate a victim's computer and disseminate malware. [14].By sending numerous requests, a denial-of-service (DoS) attack aims to intentionally overload the target system. [15]. A hacker compromises the communication connection between two separate systems to intercept messages inside in a Man-in-the-Middle (MITM) attack. [16]. Identity and Data Theft – In 2018, there were numerous data breaches that exposed the personal information of millions of people. In these data breaches, private information including email addresses, credit and debit card numbers, and personal details were taken. [17] Hackers use social engineering to trick people into disclosing their private information, like passwords and banking information. For many organizations, advanced persistent threats (APTs) are a top security concern. A targeted cyber-attack where a hacker gains unauthorized access to a network and remains undetected for a long time is known as an advanced persistent threat

1.4 Routing Protocol in IOT

The standard and unconventional protocols used for routing in Internet of Things applications are covered in this section. The routing layer, which manages the packet transfer from source to destination, and an encapsulation layer, which creates the packets, these two sub-layers partitioned the network layer. The most popular one is RPL [18]. A specific web transfer protocol called Constrained Application Protocol (CoAP) is used with constrained nodes and networks in the Internet of Things. This protocol is optimized for low bandwidth and high congestion reliability. COAP uses UDP as the underlying network protocol. In the client-server IoT protocol known as COAP, a request is made by the client, and the server responds as in HTTP. COAP and HTTP both employ similar techniques.

Rest of paper is organized in seven sections where: The literature review conducted for the pitch of the routing protocol in the Internet of Things and its risks is defined in Section II. Section III research gap. Section IV defines the proposed methodology in this area. Section V Experiment Analysis. Section VI defines analysis of various security measures applied in our proposed approach are defined. Sections VII define conclusion.

2. ALREADY DONE WORK REVIEW IN THIS FIELD

In this segment, important information regarding methodological issuesof the study is presented:

K. Mohapatra et al. [2021] look at some of the more traditional hierarchy-based routing protocols in IoT and some of the more advanced hierarchy- based routing protocols that have been developed in recent years. A comprehensive comparison is carried out on different routing protocols based on specific benchmarks like network delay, the network’s lifetime, scalability, and fault tolerance. The comparison table shows the boon and bane of each routing protocol. Finally, the paper ends with a quick summary followed by a conclusion and future direction [19]. On the other hand, Z. Magubane et al. [2021] To route data in IoT devices, a routing protocol for low-power and lossy networks (RPL) has been developed. The pathways from source nodes to destination nodes are built using objective functions in RPL. Based on a single routing metric, each objective function conducts routing. However, because they do not take into account balancing the load distribution, heterogeneous networks are unable to transport data effectively. The researchers were noticed the poor performance of RPL in IoT networks [20]. To improve the RPL's functionality in the network, they suggested a load-balancing routing mechanism. I. Kassem and A. Sleit [2020] investigated how well when the number of connected sensors is increased, CoAP performs better. The study's findings help to determine which IoT application protocols should be used for e-health, IoT devices in general, and ECG devices specifically [21]. This is relevant. MQTT-CoAP Interconnector (MCI), which is developed for compatibility of MQTT and CoAP protocols at the application layer of the Internet of Things, was explored by Dave et al. in [2020][22]. It works as a bride between the local MQTT message and the remote CoAP message with the ability to parse data. The IoT ecosystem's devices are resource-constrained, hence MCI is a lightweight interoperability solution. MCI performs better when compared to open-source alternatives when communication metrics like transmission time, throughput, latency, and packet loss are taken into account. Therefore, solving one of the key problems in the IoT ecosystem will be beneficial for the MCI interoperability solution. A rate-based congestion control method for COAP, known as BDP-COAP, was designed by E. Ancillotti and R. Bruno [2019] in a different manner and was adapted from the TCP BBR protocol. More specifically, by building a gateway for each protocol and then establishing a connection with a broker, communication between the device's multi-protocol domains of CoAP, MQTT, and Web socket can be accomplished. Each sensor's data will be processed by the ESP32 microcontroller. Making a gateway on one sensor that uses the MQTT Protocol and another sensor that uses the CoAP Protocol can help to solve the issue of differences in multi-protocol domains. The Raspberry Pi functions as a multi-protocol gateway to process the sensor data [23]. As explained by A. Zain din et al. [2019], the data are then multiplexed and delivered to the database server via the Web socket Protocol. For the purpose of discovering smart things in the Internet of Things (IoT), S. Kajwadkar and V. K. Jain [2018] suggest ECTX- CoAP, which is an improvement of CoAP-CTX. Context-Aware discovery of smart objects is possible with CoAP-CTX, but at the expense of slower discovery response times. In order to improve CoAP-CTX, it must first have a discovery response time that is comparable to CoAP and must also include capabilities for group communication in addition to multicasting [24]. This study offers insight into the numerous IoT protocols used at different tiers of the IoT protocol suite and evaluates their effectiveness and dependability based on their lightweight, secure, and energy-efficient nature.

3. RESEARCH GAP

According to the literature review, using COAP poses a difficulty if non-confirmable messages are used; reliability is not attained because the protocol runs across UDP. Additionally, using Confirmable messages merely confirms message arrival and not considering any potential errors. Non-confirmable communications are not subject to congestion control, potentially overloading the network. While growing in acceptance, the COAP protocol is still in development. Different protocol implementations may not be compatible with one another as a result of the open-source release. Due to overheads and connection creation, DTLS is excessively complex. Error management and flow controls for streaming transport are not taken into consideration by the traditional CoAP scheme. In wireless sensor networks in particular, the throughput performance often degrades. For instance, in CoAP over UDP, a message that is lost will be retransmitted following a timeout event. As a result, the error recovery technique can have a tendency to increase a major transmission delay. However, the TCP technique may add some overhead to the IoT environment. CoAP over TCP can quickly recover the lost packet by making use of TCP's fast retransmission. The head-of-line (HOL) blocking issue further demonstrates that CoAP over TCP inherits the complexity of TCP methods, making them unsuitable for real-time streaming services in the IoT environment. Many IoT devices only have a small amount of storage, memory, and processing power, therefore they frequently need to be able to run on a reduced amount of power, for instance while using a power backup. Because these restricted devices are unable to complete complex encryption and decryption swiftly enough to provide for secure real-time data transmission, security strategies that heavily rely on encryption are not suitable for them. The use of side-channel attacks, such as power analysis attacks, can frequently exploit these devices.

4. PROPOSED METHODOLOGY

An application layer protocol is COAP. COAP is a request-response protocol. Instead of merely reusing HTTP, a new protocol is created for restricted IP networks to significantly minimize the implementation complexity and bandwidth requirement. A CoAP-based system for IoT authentication and access control is proposed by Tamboli et al. For the main server, a low-power security framework is suggested, and service-based fine-grain access control is implemented. CoAP is used in the IoT environment to communicate via low-overhead packets. The system uses Kerberos with CoAP for authentication and access control, and optimized ECDSA is issued for encryption and privacy. For authentication and service access, a ticket generation-based solution is offered. Upon registration, the client receives a valid ticket for authentication, which is utilized to obtain access control when making a request for a specific service from the main server. In contrast to secrecy, integrity and authenticity are thought to be the most crucial security concerns in smart home applications. For some purposes, data integrity is more important than data confidentiality. The room inside the house for some requirement is an illustration of such an application. It is not necessary; if someone is aware of the temperature in a particular room, they can use that information to decide whether to turn on the air conditioning. However, the suggested scheme equal importance on secrecy. For example, maintaining anonymity is crucial so that no one may even view the images taken by a camera that is installed in a home room. Additionally, there are numerous other home applications that call for confidentiality.

Proposed Scheme: COAP Authentication

The COAP protocol is recommended to be improved in this study in order to provide authentication and integrity. The wireless sensor and controller will be the interface for the solution. The server and controller will concur on the following values, which are prerequisites for improving the COAP protocol:

1. A symmetric key that the sensor and controller have already agreed upon.
2. A reliable hashing algorithm for message authentication

To provide authentication, the sensor (User) and controller (server) will follow some procedure to authenticate the user. Here COAP protocol is used to send data between them. COAP protocol, by default, sends data as plain text. But here in the proposed scheme, COAP protocol sends encrypted data betweensensor and controller that enhance the security of COAP protocol. First, the user must be registered on the server in order to complete user authentication. Calculate a hashed value for that user, which is then added to the original payload value. The final step is to send the output message via COAP as the payload portion. When users attempt to log in to the server after completing the registration

The Seybold REPORT
ISSN 1533-9211
DOI 10.5281/zenodo.7152683

process, that time server will authenticate the user using the credential saved on the server with the COAP protocol's help. Below is a description of this authentication process:

Phase 1: User Registration - This is the first phase where the user must register himself with the server with personal credentials. Registration credentials considered by the proposed system are used as identity and password (of user choice per password policy). The user makes a registration request to the server with these credentials, and registration process starts as shown in Fig2

Algorithm Use registration (User End):

Step 1: Input user identity IDi, Password PWDi

Step 2: Input a random number Ri

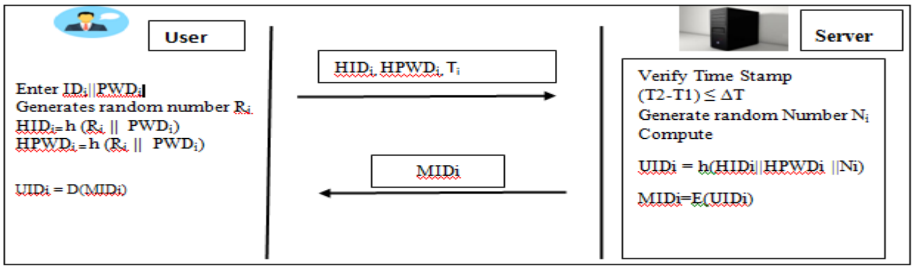
Step 3: Computer 2 hash values using MD5 hashing algorithm with Ri and each input from user.

Step 4: Compute Masked identies HIDi and HPWDi as given in equation 1 and 2.

$$\mathrm{HIDi} = h\,(\mathrm{Ri}\|\mathrm{IDi}) -------------(1)$$

$$\mathrm{HPWDi} = h\,(\mathrm{Ri}\|\mathrm{PWDi}) ------------(2)$$

Figure 2: User Registration


User
Server
Enter $ID_i||PWD_i$
Generates random number R_i
$HID_i = h\ (R_i\ ||\ PWD_i)$
$HPWD_i = h\ (R_i\ ||\ PWD_i)$
$UIDi = D(MIDi)$
$HID_i, HPWD_i, T_i$
MIDi
Verify Time Stamp
$(T2-T1) \leq \Delta T$
Generate random Number N_i
Compute
$UIDi = h(HIDi||HPWDi\ ||Ni)$
$MIDi=E(UIDi)$

A COAP protocol is used to transfer the message HIDi and HPWDi and Ti

Algorithm User Registration (Server End):

When the message reach the server the server computers.

Step 1: The server verify the time stamp using (T2-T1) ≤ ΔT where ΔT is mutually agreed upper bound between the trusted entities to prevent the various threat. Than server generate a random number Ni. Then compute User identity Compute random number Ni at server.

Step 2: Compute the identity in equation 3 MIDi of the user using HIDi and HPWDi

Step 3: Store user id UIDi in server database and encrypt UIDi with AES algorithm using preshared secret key and generate MIDi in equation no 4 and share with the user via COAP protocol

$$\mathrm{UIDi} = h\,(\mathrm{HIDi}\|\mathrm{HPWDi}\|\mathrm{Ni}) -------------(3)$$

The Seybold Report
ISSN 1533-9211
DOI 10.5281/zenodo.7152683

$$MIDi = E\ (UIDi) \qquad -------------(4)$$

From MIDi user computer generate UIDi as shown in equation no 5and computed UID is save in user data base.

$$UIDi=D\ (MIDi) ---------------------------\ (5)$$

Phase 2: Login Authentication

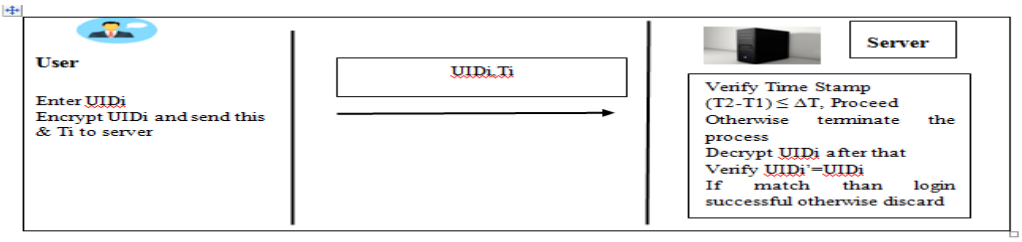
Once the registration of user is done user needs to login to access home network on the server. User uses UIDi which is unique for every user. The login procedure is shown in Fig 3. In this phase user submit the login request to server. To initiate a session following step are used for user login

Step 1: The user inserts UIDi

Step 2: Than application encrypt UIDi and send to server. Step 3: Server decrypt UIDi

UIDi'=d (UIDi)

Figure 3: Login Authentication



Enter UIDi
Encrypt UIDi and send this & Ti to server

Verify Time Stamp (T2-T1) ≤ ΔT, Proceed Otherwise terminate the process
Decrypt UIDi after that
Verify UIDi'=UIDi
If match than login successful otherwise discard

Step 4: Server match UIDi'=UIDi if both match than login successful otherwise discard login process.

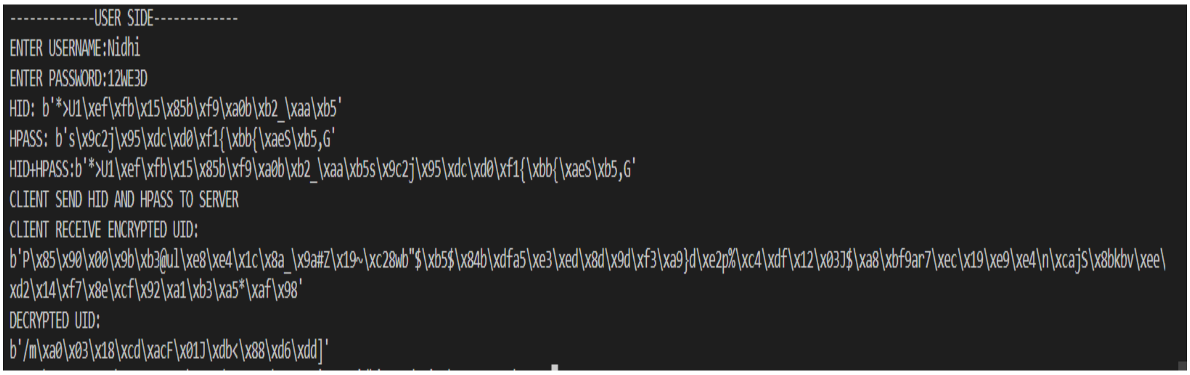
5. EXPERIMENTAL ANALYSIS

This proposed work is enhance COAP protocol security. Proposed COAP protocol is tested on client server model using python 3.0. User login into the client using id ,Password and random number using which client machine compute 2 hash value HID,HPWD. These credentials are sent to the server using COAP protocol with a timestamp Ti shown in Fig 4

ISSN 1533-9211

DOI 10.5281/zenodo.7152683

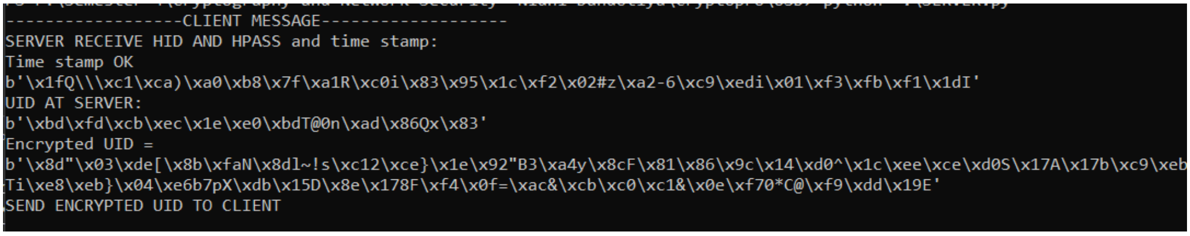
Figure 4: User Registration



```
------------USER SIDE------------
ENTER USERNAME:Nidhi
ENTER PASSWORD:12WE3D
HID: b'*>U1\xef\xfb\x15\x85b\xf9\xa0b\xb2_\xaa\xb5'
HPASS: b's\x9c2j\x95\xdc\xd0\xf1{\xbb{\xaeS\xb5,G'
HID+HPASS:b'*>U1\xef\xfb\x15\x85b\xf9\xa0b\xb2_\xaa\xb5s\x9c2j\x95\xdc\xd0\xf1{\xbb{\xaeS\xb5,G'
CLIENT SEND HID AND HPASS TO SERVER
CLIENT RECEIVE ENCRYPTED UID:
b'P\x85\x90\x00\x9b\xb3@ul\xe8\xe4\x1c\x8a_\x9a#Z\x19~\xc28wb"$\xb5$\x84b\xdfa5\xe3\xed\x8d\x9d\xf3\xa9}d\xe2p%\xc4\xdf\x12\x03J$\xa8\xbf9ar7\xec\x19\xe9\xe4\n\xcajS\x8bkbv\xee\
xd2\x14\xf7\x8e\xcf\x92\xa1\xb3\xa5*\xaf\x98'
DECRYPTED UID:
b'/m\xa0\x03\x18\xcd\xacF\x01J\xdb<\x88\xd6\xdd]'
```

When the message reaches the server, server first verifies time stamp and compute UID from HID, HPWD and random number using equation no 3. Computed UID is send back to the client as shown in Fig 5.

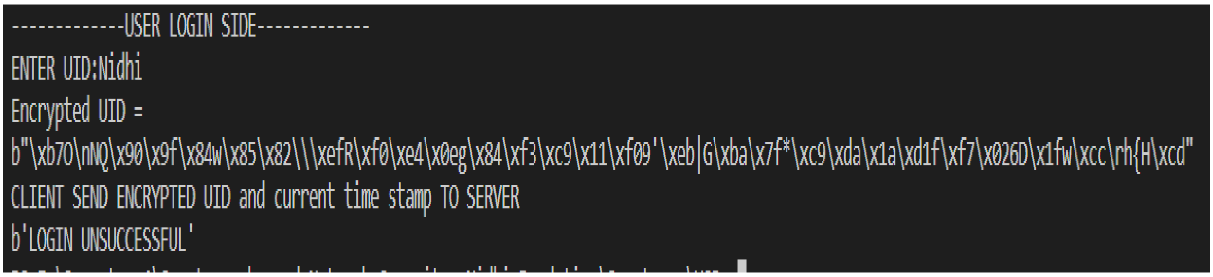
Figure 5: Server Phase



```
------------------CLIENT MESSAGE------------------
SERVER RECEIVE HID AND HPASS and time stamp:
Time stamp OK
b'\x1fQ\\\xc1\xca)\xa0\xb8\x7f\xa1R\xc0i\x83\x95\x1c\xf2\x02#z\xa2-6\xc9\xedi\x01\xf3\xfb\xf1\x1dI'
UID AT SERVER:
b'\xbd\xfd\xcb\xec\x1e\xe0\xbdT@0n\xad\x86Qx\x83'
Encrypted UID =
b'\x8d"\x03\xde[\x8b\xfaN\x8dl~!s\xc12\xce}\x1e\x92"B3\xa4y\x8cF\x81\x86\x9c\x14\xd0^\x1c\xee\xce\xd0S\x17A\x17b\xc9\xeb
Ti\xe8\xeb}\x04\xe6b7pX\xdb\x15D\x8e\x178F\xf4\x0f=\xac&\xcb\xc0\xc1&\x0e\xf70*C@\xf9\xdd\x19E'
SEND ENCRYPTED UID TO CLIENT
```

Registered user can login to the system using encrypted UID received to the server with current timestamp as shown if Fig 6.

Figure 6: User Login



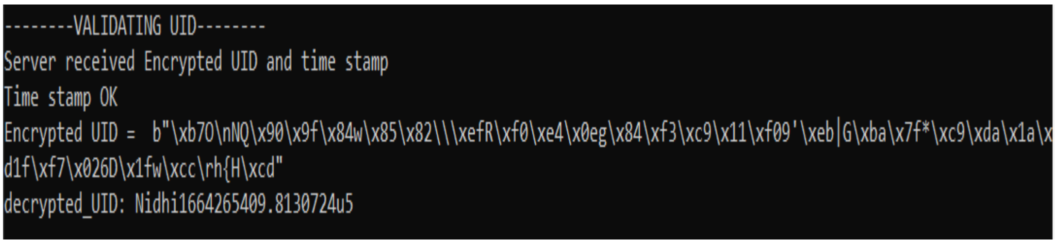
```
------------USER LOGIN SIDE------------
ENTER UID:Nidhi
Encrypted UID =
b"\xb7O\nNQ\x90\x9f\x84w\x85\x82\\\xefR\xf0\xe4\x0eg\x84\xf3\xc9\x11\xf09'\xeb|G\xba\x7f*\xc9\xda\x1a\xd1f\xf7\x026D\x1fw\xcc\rh{H\xcd"
CLIENT SEND ENCRYPTED UID and current time stamp TO SERVER
b'LOGIN UNSUCCESSFUL'
```

Server authenticate the user after verifying the timestamp followed by decrypting the UID and verifying it from its database as shown in Fig 7

The Seybold Report
ISSN 1533-9211

DOI 10.5281/zenodo.7152683

Figure 7: Server User Validation



```
--------VALIDATING UID--------
Server received Encrypted UID and time stamp
Time stamp OK
Encrypted UID =  b"\xb7O\nNQ\x90\x9f\x84w\x85\x82\\\xefR\xf0\xe4\x0eg\x84\xf3\xc9\x11\xf09'\xeb|G\xba\x7f*\xc9\xda\x1a\x
d1f\xf7\x026D\x1fw\xcc\rh{H\xcd"
decrypted_UID: Nidhi1664265409.8130724u5
```

6. ANALYSIS OF SECURITY MEASURES

The proposed COAP protocol proves to be more efficient than the base protocol as tabulated in Table 1. Authentication, access control, data integrity, content protection and other essential factors are analyzed under this study. This authentication scheme provides authentication and password protection in COAP Protocol and is resistant to several attacks. As other researcher provide security enhancement to the COAP by adding the only integrity to the COAP packet and also DTLS enhanced the security of COAP Protocol but this research provide security and integrity. The analysis confirms that the proposed scheme is resistant to following attacks-

Table 1: Comparison between propose and base approach

Security concept	Base Approach	Propose Approach
Authentication	Passwords, Biometrics	Passwords, Tokens.
Authorization Access	DRM-based Access Restriction, Biometric	Control Lists
Confidentiality	Cryptography, Access Controls, Multimedia	Cryptography, Access Controls, Database Views
Integrity	Hashing (Revised MD5)with RSA /DSA And Digital Certificate, Digital Watermarking	Hashing(SHA-l,MD5), Authentication Codes
Accountability	Secure Electronic Transaction, Firewall, Cryptography	Logging & Audit Trails
Availability	Continuous monitoring of the different modules and keeping a constant view of network connections speed,IP Tracking	Increase redundancy to eliminate a singlepoint of failure, and place" restrictionson what legal may do.
Non-repudiation	Digital Certificate, Digital Signature	Generate evidence /receipts(digitally signed)

This authentication scheme provides authentication and password protection in COAP Protocol and is resistant to several attacks. As other researcher provide security enhancement to the COAP by adding the only integrity to the CoAP packet and also DTLS enhanced the security of COAP Protocol but this research provide security and integrity both. The analysis confirmed that the proposed scheme is resistant to following attack-

6.1 User Impersonation attack to become a trusted user, consider attacker A to send an illegal login request to the home server. The opponent computes identity ID attacker, PWD and present time stamp. Because the opponent does not have a user id UID, that is why this request fails.

6.2 Man In the Middle attack is also not possible in this enhanced COAP protocol because this research send data in encrypted format.

6.3 DoS attack the user is secure against DoS attack in this proposed scheme. This is possible because the user receives an acknowledgment or denial message from the node that lets them know the response message was genuine. The proposed scheme is resistant to DoS attacks.

6.4 Replay Attack - Accept that the attacker interrupted the conveyed message < UIDj, Tj> either during login phase and with the login request message attacker starts a new session < UIDj’, Tj’>. Then the process will terminate because the proposed scheme verify the timestamp during every transmission

7. CONCLUSION

This work enhances the COAP protocol by adding to its security by ensuring integrity and security of the COAP packet. The message integrity is reached using
the MD5 hash algorithm, and AES Algorithm achieves security. The results show how se curity and integrity work with COAP protocol. This enhances COAP protocol resistant to several attacks.

References

1) Z. Shelby, K. Hartke, C. Bormann, B. Frank, Constrained application protocol (coap). Draft-ietfcore-coap-12, Expires December 27 (2012).

2) Z. Shelby, K. Hartke, C. Bormann, B. Frank, Constrained application protocol (coap). Draft-ietfcore-coap-12, Expires December 27 (2012).

3) S. R. Guha, B. Khulna, Constrained application protocol for internet of things.

4) X. Chen, Constrained application protocol for internet of things, URL: https://www. cse. Wustl. edu/˜ jain/cse574-14/ftp/coap (2014).

5) Z. Shelby, Core link format, draft-ietf-core-link-format-06, IETF workin progress (2011).

6) A. P. Castellani, T. Fossati, S. Loreto, Http-coap cross protocol proxy:an implementation viewpoint, in: 2012 IEEE 9th International Confer- ence on Mobile Ad-Hoc and Sensor Systems (MASS 2012), IEEE, 2012, pp. 1–6.

7) Z. Shelby, K. Hartke, C. Bormann, The constrained application protocol (coap), Tech. rep. (2014).

8) M. R. Abdmeziem, D. Tandjaoui, I. Romdhani, Architecting the internet of things: state of the art, Robots and Sensor Clouds (2016) 55–75.

9) A. P. Castellani, M. Gheda, N. Bui, M. Rossi, M. Zorzi, Web services for the internet of things through coap and exi, in: 2011 IEEE International Conference on Communications Workshops (ICC), IEEE, 2011, pp. 1–6.

10) F. Van den Abeele, I. Moerman, P. Demeester, J. Hoebeke, Secure ser- vice proxy: A coap (s) intermediary for a securer and smarter web of things, Sensors 17 (7) (2017) 1609.

11) C. Bormann, Z. Shelby, Block-wise transfers in the constrained application protocol (coap), Tech. rep. (2016).

12) G. Choi, D. Kim, I. Yeom, Efficient streaming over coap, in: 2016 in- ternational conference on information networking (ICOIN), IEEE, 2016, pp. 476–478.

13) C. Bormann, S. Lemay, H. Tschofenig, K. Hartke, B. Silverajan, B. Ray-mor, Coap (constrained application protocol) over tcp, tls, and websock- ets, Tech. rep. (2018).

14) S. Feizi, D. E. Lucani, C. W. Sørensen, A. Makhdoumi, M. Médard, Tunable sparse network coding for multicast networks, in: 2014 Inter- national Symposium on Network Coding (NetCod), IEEE, 2014, pp. 1–6.

15) J. Zhang, F. Ren, L. Tang, C. Lin, Modeling and solving tcp incast problem in data center networks, IEEE Transactions on Parallel and Distributed systems 26 (2) (2014) 478–491.

16) M. Zhang, M. Mezzavilla, R. Ford, S. Rangan, S. Panwar, E. Mel- lios, D. Kong, A. Nix, M. Zorzi, Transport layer performance in 5g mmwave cellular, in: 2016 IEEE Conference on Computer Communica- tions Workshops (INFOCOM WKSHPS), IEEE, 2016, pp. 730–735.

17) M. Zhang, M. Polese, M. Mezzavilla, J. Zhu, S. Rangan, S. Panwar, M. Zorzi, Will tcp work in mmwave 5g cellular networks?, IEEE Com-munications Magazine 57 (1) (2019) 65–71.

18) D. Gligoroski, K. Kralevska, H. Øverby, Minimal header overhead for random linear network coding, in: 2015 IEEE international conference on communication workshop (ICCW), IEEE, 2015, pp. 680–685.

19) K. Mohapatra, R. K. Lenka and S. Sharma, "A Survey on Classical and Optimized Hierarchical Routing Protocols for IoT and WSN," 2021 6th International Conference on Signal Processing, Computing and Control (ISPCC), 2021, pp. 620-624

20) A. K. Mishra, O. Singh, A. Kumar and D. Puthal, "Hybrid Mode of Operations for RPL in IoT: A Systematic Survey," in IEEE Transactions on Network and Service Management

21) Kassem and A. Sleit, "Elapsed Time of IoT Application Protocol for ECG: A Comparative Study between CoAP and MQTT," 2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), 2020

22) M. Dave, J. Doshi and H. Arolkar, "MQTT- CoAP Interconnector: IoT Interoperability Solution for Application Layer Protocols," 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2020, pp. 122-127

23) E. Ancillotti and R. Bruno, "BDP-CoAP: Leveraging Bandwidth-Delay Product for Congestion Control in CoAP," 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), 2019, pp. 656-661

24) S. Kajwadkar and V. K. Jain, "ECTX-CoAP: An Enhanced Context-Aware CoAP Extension for Smart Objects Discovery in Internet of Things," 2018 Conference on Information and Communication Technology (CICT), 2018, pp. 1-6