SURVEY ON SECURITY PROVISIONS FOR IoT LAYERS

Foram Suthar¹, Gaurang Raval², Sharada Valiveti²

¹Indrashil Institute of Science and Technology, Gujarat, India
²Institute of Technology, Nirma University, Gujarat, India

ABSTRACT:

IoT can be defined as new technology for new generation which improves our daily life using some tiny devices, sensor nodes which are connected to each other through Internet. We can easily operate our home appliance using IoT at anytime and anywhere. But the main issue behind IoT based communication is security. Secure communication between all devices is highly important and a mandate in IoT. The purpose of this paper is to explore various security provisions available in different layers of IoT stack. This paper examines various application layer protocol, transport layer protocols and other layer protocols from security point of view. The paper also discusses other related work done in the security domain of IoT.

Keywords: IoT, CoAP, DTLS, 6LOWPAN

[1] INTRODUCTION

Internet of Things is growing at a rapid pace aiming at connecting every possible electronic device using Internet. It works as a standard of Communication Bridge between computer and human, computer and computer as well as machine to machine. Among the many application layer protocols in IoT the most popular and preferable protocol is CoAP (Constrained Application Protocol). CoAP basically serves the purpose of providing almost realtime message exchange service between devices or nodes complying with IoT standard.

IoT enables a new type of communication mode that is machine to machine (M2M). The communication can be between any two devices interconnected, may perform certain fixed set of tasks also without human intervention. The medium of communication during such events may be wired or wireless. Interesting example to explain this concept is the synchronization of email account in a mobile device for which the necessary credentials are already stored in the mobile device. As and when any new mail arrives for the specific account the servers push the message to the device and accordingly the notification also pops up on the device about the new mail arrival. The mobile device is tightly synchronized with such sophisticated mail services. Messages are also delivered in a reliable fashion, no messages are dropped. All these activities take place without any human intervention.

IoT standards and protocols provide the framework to design interesting applications involving tiny devices and communication networks. An example here can be, on the basis of
preferences priory given by the user, room temperature and illumination conditions can be set in advance when the user is about to reach the place. IoT provides such highly customized service features which not only involves devices but humans are also part of this network as well as machines. Hence IoT applications are generically categorized in to M2H (machine to human), H2M (human to machine) or M2M (machine to machine) type applications targeting users either for personal use or business related applications.

IoT can be of much help in improving the living standards of people in upcoming decade. The difficult problem being faced by IoT users is that the security standards are not yet matured in IoT. The IoT security is must and highly needed for establishing secure communication between two devices. The purpose of this paper is to explore security provisions for IoT layers. In this paper, we discuss and examine various security related provisions in different IoT layers and assess other relevant work done in this domain by researchers.

[2] COAP (CONSTRAINED APPLICATION PROTOCOL)

The popular application layer protocol for IoT is CoAP which can be applied to devices with constrained resources. Such devices have small footprint in terms of size, Operating system, memory, bandwidth and most importantly energy source [12]. These type of devices when interconnected can form a network which may have higher rate of transmission errors whereas throughput can be low ranging in Kbits. These networks due to the typical characteristics of its own cannot provide advance services like IP multicast, example 6LowPower Personal Area Network [11]. Such networks aims at integrating IPv6 features into Personal Area Network comprising of lightweight devices having limited computability and restricted communication facility to transmit information in wireless mode applying IP. In [8] authors have proposed implementation of reliability layer on top of CoAP layer of IoT using Reed-Solomon code applying the technique of erasure coding. Authors have also applied LZ77 and LZ78 compression techniques for the packets being exchanged between IoT devices.

CoAP works as per the conventional request-reply protocol works, where client requests for some service and server provides the same. CoAP applied transport layer protocol UDP which hash simple header format of 4 bytes. Traditional data link layer mechanism of stop and wait is used by CoAP also interchangeably called “Block wise Transfer” [2]. CoAP employs four types of messages for its functioning, Confirmable, Non-Confirmable, ACK and finally Reset. A Request may be of type confirmable or Non-Confirmable. The CoAP protocol also supports Publisher / Subscriber mechanism where publisher publishes message without having any information about subscribers. These messages are categorized into different channels as per the subscriptions for such messages or services. Subscribers register their interest for certain messages or a specific publishing service based on their interest. Layered architecture of CoAP is shown in Figure 1.
[3] DTLS

DTLS is the lightweight transport layer protocol for especially designed for secure UDP based communication in the IoT domain. DTLS employs TLS for its functioning however TLS is fundamentally used for connection oriented service like secure web traffic or e-mail. Two issues in TLS are as follows.

1. Individual records cannot be independently decrypted. As the integrity check relies on the sequence number, if record N is not received, then the integrity check on record N+1 will be based on the wrong sequence number and thus it will fail [10].
2. The assumption of handshake messages being reliably delivered can break the TLS functioning if those messages are lost in worst case [10].

Because of these issues TLS protocol cannot be directly applied to secure CoAP based communication. Hence new protocol DTLS was proposed by researchers based on datagram protocol for supporting services like media streaming, internet telephony, online gaming application etc. The functioning of DTLS is described as follows.

1. Packet loss
   - A retransmission timer is employed by DTLS to deal with packet loss event
   - After sending hello message to server, client starts timer and waits for the reply. In case of timer expiring before any message arrival client accordingly concludes that either the hello message was lost or verify request is lost. Here client initiates message retransmission.
2. Reordering
   - During every handshake session DTLS assigns specific sequence number to each and every handshake message. Upon arrival of the handshake message a peer can determine that the message is the expected one in sequence or not. If the peer finds it suitably in sequence it immediately processes it otherwise queues it up for processing. The queued up messages are processed when all pending messages of lower sequence numbers are received [10].
3. Denial-of-Service Countermeasures
   - Two types of attacks occur in Datagram security protocols
1. An Attacker transmits a series of handshake initiation message and consumes excessive resources on server.
2. An Attacker send connection initiation message with forged source of the victim and use server as amplifier to send the message.

DTLS handles the attacks by employing stateless cookie mechanism which applies Photuris \[10\] and IKE \[10\] techniques. Here a hello message is first sent by Client to the Server. The Server then replies with HelloVerify message. This verification message is having a stateless cookie which is generated by Photuris technique. After this client again transmits Hello message with the cookie added. The cookie validity is verified by the Server first before further processing related to the handshake process.

4. Record Layer
   - DTLS Record Layer exactly matches with the TLS Record Layer except two fields. Two new fields which are added in DTLS Record layer are epoch and sequence number.

   Type: - The higher level protocol same as TLS 1.2 Record Layer.
   Version: - This field specified the version value. Here DTLS 1.2 version used.
   Epoch: - Epoch is counter when the cipher state change counter value is incremented.
   Sequence Number: - Record sequence number.
   Length: - This field is same as length field of TSL 1.2 record layer which shouldn’t exceed power \((2^{14})\).
   Fragment: - The application data. This data is transparent and treated as an independent block to be dealt with by the higher-level protocol specified by the type field.

   ```
   struct {
       ContentType type;
       ProtocolVersion version;
       uint16 epoch;          // New field
       uint48 sequence_number; // New field
       uint16 length;
       opaque fragment[DTLSPlaintext.length];
     } DTLSPlaintext;
   ```

   ![FIGURE: 2. RECORD LAYER OF DTLS](image)

[4] RELATED WORK

Researchers practicing in this domain have contributed significantly in the recent past in the area of IoT security. In [6] they introduce work emphasizing on low power usage in Contiki. The Internet of wireless things is strongly in need of power-efficient protocols. Radio transceiver is the most power consuming component in wireless network so we have to keep its radio-off as much as possible. There are many radio duty cycle algorithms available which allows nodes to keep transceivers off for more than 99 % of the time [3][5]. Here they are using ContikiMAC protocol at radio duty cycle layer. Their efficient wake-up mechanisms have 8 Hz wake-up frequency and 0.6% radio duty cycle [3].
Node periodically wake-up and check the channel. If any transmission is going on than it keeps on the radio to listen for the data arrival. If data is received, receiver send ACK frame to sender repeatedly send data until it received an ACK or entire wake-up period. Figure 3 shows the operation of ContikiMAC.

![ContikiMAC Operation Diagram](image1)

**FIGURE: 3. A CONTIKIMAC SENDERS WAKES ITS NEIGHBORS UP BY SENDING A STROBE OF DATA FRAMES UNTIL IT GETS AN ACKNOWLEDGEMENT [6]**

In [1] authors have introduced a new concept that provides security during Bootstrapping process. In Bootstrapping [1] process nodes are vulnerable. They do not have knowledge about network and communication partners. New nodes are accidentally added to a neighbor’s network. Thus, the authentication mechanism is required so that only legitimated device participates in the network and related applications. Florian Junge implement the security using shutter control mechanism as shown in figure 4. This mechanism is having four main components switch, shutter, light sensor and micro controller.

Micro controller is the main component in this mechanism. Shutter can be operated automatically based on sensor value or manually. To keep the room cool shutter will go down if outside light value is too high. Switch is having four status values, up, down, auto and off. To close and open the shutter real time clock of the controller is used. All other components are managed by micro controller. Micro controller collects the state information from switch, shutter, and sensor and sends command to the shutter controller. So all the messages are passed by micro controller only and no one make their own decision and security during communication is ensured. Figure 4 shows the architecture of shutter control approach.

![Shutter Control Architecture Diagram](image2)

**FIGURE: 4. ARCHITECTURE OF SHUTTER CONTROL [1]**
This mechanism works in three phases
(1) first phase is to discover server using CoAP Server Discovery Server
(2) second phase is to provide secure channel using DTLS
(3) Third phase is the implementation phase.

In [7] authors have presented security framework for IoT called BlinkToSCoAP. It is combination of mainly three IoT protocols DTLS, CoAP and SiGLowPAN (here 6LoWPAN [11] is SiGLowPAN). The architecture of BlinkToSCoAP [7] is shown in Figure 4. BTSCTest is top most components in the BlinkToSCoap stack. BTSCTest performs the role of CoAPClient or CoAPServer using CoAP functionalities which are provided by CoAPClient and CoAPServer components. BlinkToSCoAP client send CoAP Blink request to the Blink- ToSCoAP server and wait for reply. Server receives the proper request from client and counter is incremented by 1 at server side. SSLP work as gateway between the DTLS, CoAP and SiGLowPAN.

Authors proved that secure communication is established between the communicating entities. If communication is taking place between the two layers that is, CoAP and SigLowPAN then data first pass to DTLS for control information to be added for secure communication through SSLP. DTLS prepares the packet accordingly by adding necessary control bits and encrypting the data and again pass it to SSLP. SSLP pass that data to SigLowPAN layer. Architecture of BlinkToSCop [7] is shown in Figure 5.

![Architecture of BlinkToSCoAP](image)

In [9] authors have introduced new concept that is compression of DTLS for COAP. They propose a mechanism to compress the size of upper layer headers and security header of DTLS. The header compression mechanism provided by 6LOWPAN is so processes such that it reduces header size by 62 percentages. The reason behind using this compression mechanism is that DTLS is having a too long header which is unable to fit in single LR-WPANs. IPv6 over Low-power Wireless Personal Area Network (6LoWPAN)[4] enables the use of IP in IEEE 802.15.4 low power and lossy wireless networks such as wireless sensor network (WSN)[9].

This type of networks are IP-connected Internet of things networks using UDP and CoAP protocol for running the required set of application in the tiny devices. For security provision DTLS is applied along with UDP. As shown in Figure 6 Border Router works as...
bridge between conventional Internet and 6LOWPAN network. CoAP constrained node can communicate with devices like phone, laptop using secure DTLS channel and DTLS header is compressed by the proposed compression mechanism.

![Diagram of IP-connected IoT](image)

**FIGURE: 6. ARCHITECTURE OF IP-CONNECTED IOT [9]**

For IP header 6LowPAN [4] define IP Header Compression and for IP extension and UDP header 6LowPAN define Next Header Compression. DTLS is part of UDP payload which is compressed by 6LOWPAN-GHC [9] because UDP payload has been compressed by that. Authors here propose 6LowPAN to compress four header formats which comprises of Record, Handshake, Client Hello and Server Hello messages. Two cases are possible while compressing record and Handshake header. First case is using single encoding byte to compress both Record and Handshake header and another is using 6LowPAN- GHC-R[9] only for Record Layer. Figure 7 shows the Record and Handshake Header compression and only Record Header compression.

<table>
<thead>
<tr>
<th>BIT</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>V</td>
<td>EC</td>
<td>SN</td>
<td>F</td>
</tr>
<tr>
<td>Record+Handshake (LOWPAN_GHC_RHS)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BIT</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>V</td>
<td>EC</td>
<td>SN</td>
</tr>
<tr>
<td>Record only (LOWPAN_GHC_R)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

V: Version
EC: Epoch
SN: Sequence Number
F: Fragment

**FIGURE: 7. HEADER FORMAT FOR RECORD AND HANDSHAKE HEADER AND ONLY FOR RECORD HEADER [9]**

In Figure 7 first bits are representing ID bits. The value 1001 used for 6LOWPAN-GHC-R and 1000 used to represent the scheme 6LOWPAN-GHC-RHS. 6LOWPAN-GHC-CH used for client Hello message as shown in Figure 8. At the time of handshake process client hello message is sent two times, once with cookie and other is without cookie. The value 1010 ID is used for 6LOWPAN-GHC-CH.
6LOWPAN-GHC-SH[9] is used for serverHello message as shown in Figure 9. The value 1011 ID is used for representing 6LOWPAN-GHC-CH.

[6] CONCLUSION

This paper explores various protocols used by different applications of IoT and discusses why the security plays importance role in IoT based networks. IoT security is most critical issue during the communication between two devices following or applying IoT standards and protocols. The constrained application layer protocol play important role in Internet of thing for simple data exchange based applications. This paper also studied few frame works reported in the literature, security algorithm like BlinkToSCoAP. Protocol also exists when applied at radio duty cycle layer; it provides security to tiny components used in IoT. Paper also shown the difference between DTLS and TLS and and also how DTLS plays lead role for data-gram or unreliable communication based application scenarios. Due to large Header size of DTLS, 6LowPAN uses various compression mechanisms to compress the DTLS header format and also compresses security header.
REFERENCES

Author[s] brief Introduction

Foram Suthar is working as Assistant Professor in Computer Engg department with Indrashil Institute of Science and Technology, Gujarat. She has done her PG studies At Institute of Technology, Nirma University. Gaurang Raval is working as Associate Professor in IT department at Institute of Technology, Nirma University. His area of interest are sensor networks and distributed Computing. Sharada Valiveti is working as Associate Professor in CE department at IT department at Institute of Technology, Nirma University. Her area of interest is adhoc network security.