Security Issues in Constrained Application Protocol: A Literary Survey

Tanushree Garg [1], Ayush Gupta[1], Prof Snehal Chaudhary [1], Prof Priyanka Paygude[1]
[1]Information Technology, Bharati Vidyapeeth Deemed To Be University College of Engineering, Pune, Maharashtra, India.
[1]itstanushreegarg@gmail.com, ayush.gu96@gmail.com, sdchaudhary@bvucoep.edu.in, pspaygude@bvucoep.edu.in.

Abstract: Internet of things (IOTs) is an emerging technology; there is a wireless network between smart products or smart things connected to the Internet. It not only connects objects and people but also connects billions of gadgets, smart devices .Growth in IOT has increased rapidly in the last few years, and there has been a strong increase in the security vulnerabilities of connected items. As such, 802.15.4, 6LoWPAN, and RPL IoT layers PHY / MAC, adoption and resenting of the network are widely used protocols. While the Constrained Application Protocol (COAP) is, an application layer protocol designed to assist in the development of smaller devices under Classes 1 and 2, designed as a duplication of HTTP. Many implementations of COAP have been efficient enough to indicate this important and upcoming role in the future of all IOT applications .This survey highlights the CoAP, its specification, its implementation and observation of security analysis.

Keywords —6LoWPAN, 802.15.4, COAP, DTLS, IPS, Internet of things, Security.

I. INTRODUCTION

Today the Internet is used everywhere, has touched almost every corner of the world, and is influencing human life in many incredible ways. We are entering an era of IOT .Internet of things refers to a new kind of world where almost all the devices and equipment that we use are connected to a network such as washing machines, televisions, all kitchen appliances, etc. We can use them cooperatively to achieve complex tasks, a high level of intelligence.

IOT technologies allow things, or devices that may or may not be a computer, to work smartly and make valuable decisions applications. Due to the large address space provided in IPv6, IOT optimization has become more clear and realistic and IPv6 deployment has become easier with ease .It enables more machines to take advantage of the Internet feature and, thus, communicate effectively with each other.

IOT security is an important aspect, due to which it is related to sensitive data flowing on the Internet. However, if security is applied then the biggest challenge will be that the performance and speed of the devices are not affected. IOT uses tools that are light, that is, they should keep in mind that they have low processing power and high memory capabilities so that they can be delayed in time with one another and the overall throughput should not be affected.

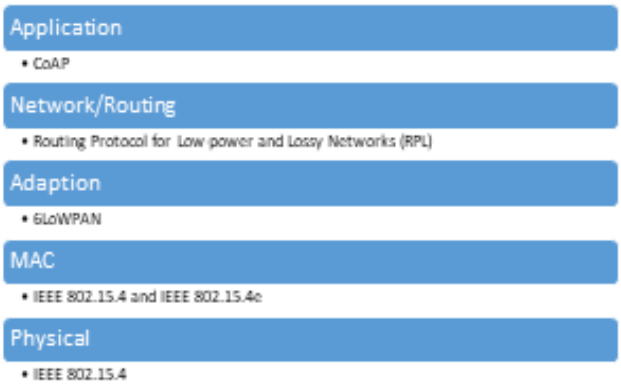

Figure 1. IoT Protocol Stack

Five layers representing IOT: physical, mac, adaption, network, and applications. IOT's protocol mechanism runs in parallel with these layers.

The following are the main features of the given IOT protocol in the following approach:

1) Physical (PHY) and medium access control (MAC) layers require less power communication, which is provided by IEEE 802.15.4, it determines the guidelines on the lower layers of the heap and grounds for the IOT protocol on the higher layers Gives.
2) The low energy communication environment uses IEEE 802.15.4, which is at most 102 bites to move the data into higher layers .This value is less than 1280 bytes, which means the maximum transmission unit (MTU) required for IPv6 [3] .Adaption layer's 6LoWPAN protocol addresses this aspect by enabling IPv6 packet transmission on IEEE 802.15.4. [3] This

 DOI : 10.18231/2454-9150.2018.0128

packet also manages the mechanism of fragmentation and re-assembles between the functions.

3) Routing protocol for RPL that is less power and Lossy network supports more routing than 6LoWPAN. Instead of having a routing protocol, it provides a framework that is flexible for special IOT application domains .Application-specific profiles are already defined to identify related routing requirements and optimization goals.

4) Communication on the application layer is supported by the Constrained Application Protocol (CoAP) .This protocol is still in designing in IETF to provide interoperability to the Web's representative state transfer (REST) architecture.

II. CONSTRAINED APPLICATION PROTOCOL: OVERVIEW

The CoAP - Application Layer Protocol was originally developed for web transmission with constrained nodes and networks. This low power and multicast communication, there is an important version of HTTP to meet the IOT requirement for support .CoAP relies on the REST principle that has been adopted from HTTP and embedded in the UDP for transactions. The main or original reason for developing this protocol is to meet the IOT's high requirements and low rates and light protocols are required.

The main features of the COAP protocol are:

1. It supports machine 2 machine requirements in bound environment,
2. Optional support UDP binding with uni-cast and multicast requests,
3. Unlimited Message Exchange,
4. Less header overhead and parsing complexity,
5. Supports URI (universal resource identifier) and content-type,
6. Has simple proxy and caching capabilities

A. CoAP structural model: Architecture

The interaction model or CoAP's structural model is similar to the HTTP client / server model. However, machine-to-machine (M2M) communications usually work in both client and server because of a CoAP implementation. A CoAP Request Is similar to HTTP and A server is sent by the client to request action using the method code on the processing code (identified by the URI) on the resource .The server then responds with the response code; This response may include resource representation.

It is different from HTTP because CoAP works with these exchanges on a datagram-oriented transport such as UDP with unlimited deals. This is done primarily by using a layer of messages that supports non-essential reliability.

Four types of messages that define CoAP:

1) Confirmable (acknowledgment is required).
2) Non-verifiable (no ACK is required).
3) Acknowledgment (ACK CON message)
4) Reset (messages received but cannot be processed).

Figure 2: HTTP and CoAP protocol heap [4]

B. Message Layer Model

The CoAP messaging model is based on communication between UDP on Endpoints.

CoAP uses a small fixed-length binary header that is 4 bytes, which is compressed Whether binary options and payloads can be followed .The message format is shared by request and feedback messages .One message used for duplicate and alternative reliability in each message ID is. Message ID is compact and its 16-bit size enables approximately 250 messages per second from one endpoint to another with the default protocol parameter.

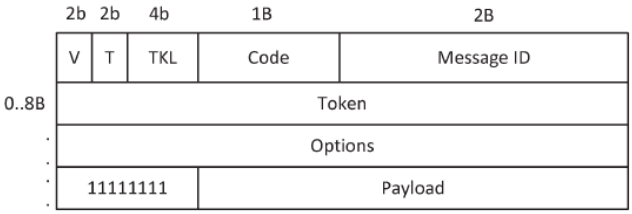

Figure 3: Format of a CoAP Message Header

A message is authenticated by marking it as Confirmable (CON) .A confirmation message is resend until the receiver sends a receipt message (ACK) from the associated end point with the same message ID (in this example, 0x8c56); See Figure 4 and it is done by using default time and counting time faster. When a receiver fails to process a confirmation message i.e. is also not able to provide a suitable error response, it answers with a reset message (RST) instead of a receipt (ACK).

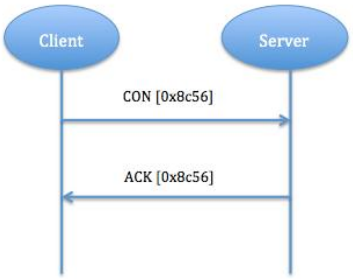

Figure 4: Reliable message transmission [4].

A message for which reliable transmission is not required; it is a non-confirmable message (NON). There is no need to accept it. Still has a message ID for duplicate recognition and is also for retransmission (in this example, 0x8c57); See Figure 5. If the receiver fails to process a non-

 DOI : 10.18231/2454-9150.2018.0128

confirmable message, then it can respond with the reset message (RST).

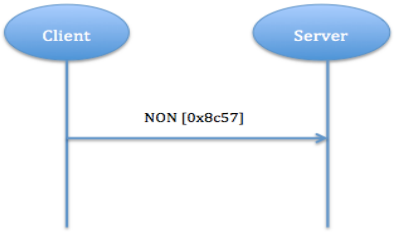

Figure 5: Untrustworthy messaging [4].

C. Request / Response layer model

Response request and response in COAP requests respectively contains the method code or response code. Optional (or default) requests and response information, such as the URI and Payload media type, are taken as AAAP option. A token is used to match the responses to requests independently from the underlying messages. In this case, the concept of tokens is quite different for message ID.

The customer sends a request using a cone type or non-type message and the server responds quickly using ACK with a confirmable message .This piggybacked reaction is known. See Figure 6, a successful and one Example not found.

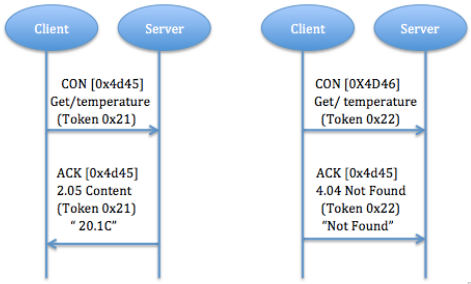

Figure 6: Successful and failure response results of GET method [4].

If the server fails to respond promptly to a request made in a confirmation message, then it only answers with a blank acknowledgment message so that the customer can stop the request again. When answer If it is ready, the server sends it in a new confirmable message (which in turn should be accepted by the client).This is called a "different reaction" See Figure 7

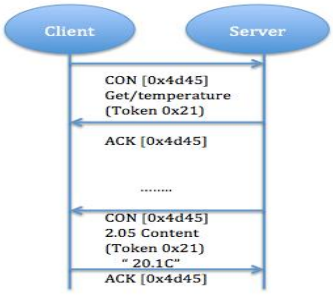

Figure 7: A. Receive the request with a different reaction [4].

If a request is sent in a non-confirmable message, the response is sent using a new non-confirmable message, although the server can send a confirmation message instead. This type of exchange Shown in Figure 8.

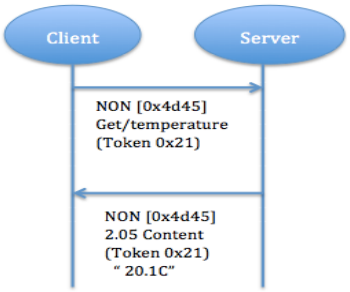

Figure 8: Non-confirmation requests and responses [4].

D. Message Format

COAP is based on the exchange of compact messages, which is broadcast on UDP by default. COAP's messaging format uses a simple binary format.

Message = fixed six 4 byte headers + variable length token + sequence of + CoAP options + payload

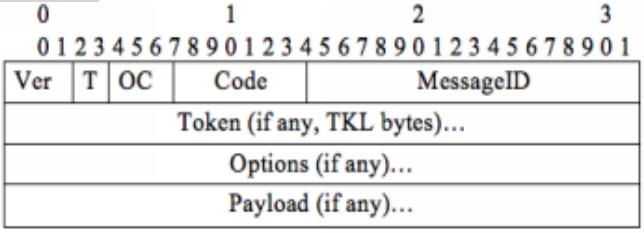

Figure 9: Message Format [4].

The option number is calculated in the option format as follows:

Option number = option delta + previous option number. (See fig 10).

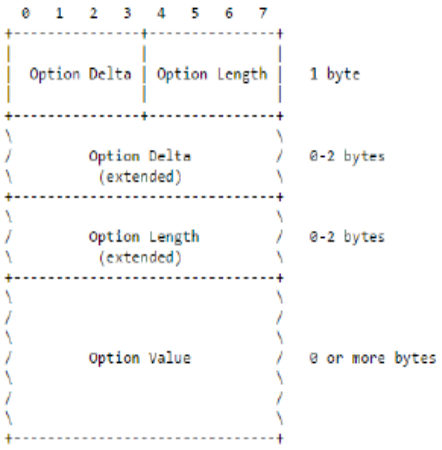

Figure 10: Option Format [4].

III. COAP SECURITY ANALYSIS

CoAP is now the specific protocol for IoT applications is going on .Safety is important because it helps in the protection of communication between devices. There were no safety features in the early design of COAP; recently, researchers looked to examine the security of the COAP

 DOI : 10.18231/2454-9150.2018.0128

implementation .DTLS and IPsec: CoAP Internet Draft offered two security protocols that can be used to secure the COAP network and its traffic.

A. CoAP - DTLS security:

The DTLS protocol is an advanced version of the widely used Transport Layer Security (TLS) protocol. The main difference is that major UDPs like the DTLS 165 Voice over IP / Session Start Protocol (VoIP / SIP) run on the top of the UDP rather than TCP to secure the famous applications. Provides DTLS authentication, data integrity, privacy and automated key management. It also supports a wide range of different cryptographic algorithms, which makes it a potential security protocol candidate.
To achieve security services, CoAP defines four safety modes. These modes are Nosec, PresharedKey, RawPublicKey, and Certificate. [3]

1) **Nosec:** This option assumes that security is not provided in this mode or in the COAP sent message.
2) **PresharedKey:** This mode is enabled by sensing preprogramed devices with symmetric cryptographic keys. This mode is suitable for applications that support devices capable of employing public-key cryptography. In addition, apps can use a key for a key or device group per device.
3) **RawPublicKey:** Mandatory modes for devices that require authentication based on public keys. The device is programmed with a key pre-provision list so that device can start DTLS sessions without a certificate.
4) **Certificate:** Authentication based on public key and application participating in the authentication series supports authentication. The concept of this mode is that the security infrastructure is available. Devices that contain asymmetric keys and can use authenticated X.509 certificate mode and make provision of reliable root keys.

The following two diagrams easily show messages with DTLS and without CoAP message:

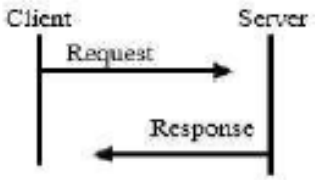

Figure 11: 1 round trip without COAP request / response, DTLS [2].

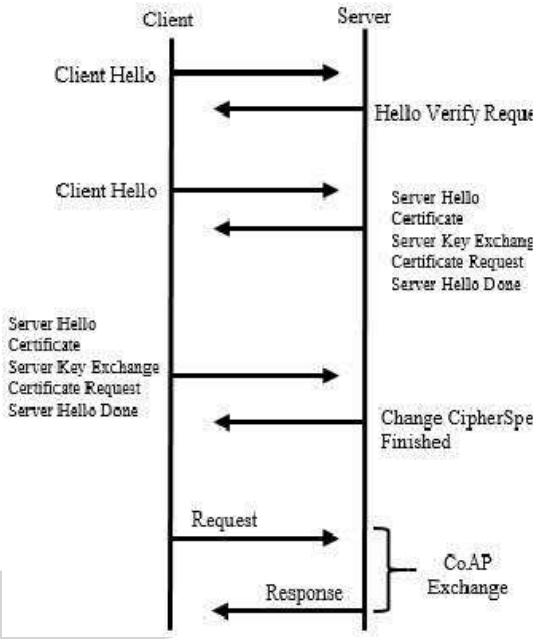

Figure 12: CoAP Request / Response with DTLS, 4 Round Trip [2].

Drawbacks:

1) The DTLS protocol does not support multicast communication, which is an essential part of the COAP protocol and main feature in IOT.
2) DTLS Handshake Protocol With the addition of stateless cookies in case of any attack, the battery operated device can cause the exhaustion of resources. Consequently, nodes can lose their role in the network and can create disruption in the entire communication. [2]
3) Although the DTLS can protect against replay attacks using bitmap windows, nodes first have to pack, process and sometimes they have to get ahead too. Without filtering proxies such as 6LoWPAN boundary router (6LBR), the probability of this attack can present the network flood. The management of such filtering on 6LBR cannot be guaranteed on all scenarios. In addition, the answer pack TS processing is energy consumption.
4) Handshake phase is safely weak because any end-host has not been certified by other end-host. In addition, handshake message fragmentation is still an issue, although a friendly solution was proposed without verification. In addition, to verify handshake messages, a hash function is required on all messages, which means that some nodes require large buffers, and this Not applicable in every case.
5) DTLS security features do not fit well for CoAP. For example, in-flight requires a rematch of all messages that fly to harm a message. On the other hand, if all the flying messages are broadcast simultaneously in the same UDP packet, then more resources are needed to handle large buffers. In addition, if the COAP client requires internet access, which requires the COAP / HTTP mapping process, and then the DTLS handshake

 DOI : 10.18231/2454-9150.2018.0128

process remains a challenge [2]. Specifically, it is not clear whether partial mapping between TLS and DTLS can be done or not. This problem can be even more complex because a CoAP client will not be able to recognize which device has started the request. Finally, COAP messages spent only two transactions (1 round trip) on the network; a message from the client (request) and another from the server (response). If DTLS is used, then 4 round trip is required; Prior to exchanging the actual contents of COAP, 3 Round Trip with DTLS (~ 40-50 Bites) and 1 Round Trip for CoAP [2].

B. CoAP - IPsec Security

In addition to the DTLS security protocol as a security, for COAP, other implementations and applications can use IPSEC. IPSEC is a Layer 3 protocol intended to be used with IPv6, but IPv4 has been modified to be compatible. This is an independent protocol that can secure applications and transport layers applications. IPsec is integrated into the kernel; therefore, it is transparent for applications. Due to its transparency, IPsec can also use other security protocols like TLS and Secure / Multi-purpose Internet Mail Extensions (S / MIME). [2]

The security services that can be provided by IPsec are Connectionless Integrity, Access Control, Data Original Authentication, Privacy, Anti-Replay Mechanism, and Limited Traffic Flow Privacy. IPsec To secure COAP communication with, one method is to use Encapsulating Security Payload Protocol (IPsec-ESP), especially if hardware supports encryption on layer 2, because in the case of some IEEE 802.15.4 radio chips is.[2].

C. Drawback Of CoAP - IPsec And DTLS Security

DTLS and IPsec are not the most optimized solution to protect CoAP due to the following reasons:

1) IPsec and DTLS require additional messages to negotiate safety standards and establish security associations (SAS), this will increase the overhead and the resources of the disrupted equipment will be excluded. This problem will be more problematic when considering the mobile nature of devices in IOT, because the new SA device is required to be installed every time.
2) While considering the scenario of communication between two different networks, the proposed security solution is based on either IPsec or DTLS, which means the presence and support of these protocols in both source and destination networks. This notion cannot be realistic in many situations, especially when considering the fact that there is a compatibility problem with the firewall on the network in the IPsec protocol. [2]
3) Both IPsec and DTLS rely on other protocols such as Internet Key Exchange (IKE) and Extensible Authentication Protocol (EAP) to set up secure protocols; this implies that the vendors of all the binding devices should support these additional protocols (IKE and EAP).
4) Since IPsec and DTLS are designed to secure the connection between two static and remote devices, they are most likely to secure between both ends without regard to QoS, network dependency or any other limitations on the end devices. Try to provide connections. However, while considering providing security in an interrupted environment, more dynamic and sensible measures are needed which, while negotiating safety standards, consider the interrupted nature of end equipment. [2]
5) The IEEE 802.15.4 specification defines the entire payload as 127 bytes. In the case of using DTLS as a security protocol to protect COAP exchanges, 13 bytes (out of 127 bytes of IEEE 802.15.4 frame) will be allotted for DTLS records. 25 bytes are used for link layer addressing information, 10 bytes for 6lowpan addressing, and 4 bytes of CoAP header. As a result, there are 75 bytes remaining for application layer payload, which does not have much room to move actual data. As a result, a large part of the data (greater than 75 bytes) will use more resources than nodes and networks because it will be broken and sent twice. Therefore, whenever possible, some headers have been proposed to use compression mechanism. Due to compression and decompressing requirements, these compression mechanisms can hinder nodes and network resources. [2]
6) In the case of DTLS, some applications may require flexible customization of security services in accordance with application services or scenario requirements. For example, some apps want to keep messages in accordance with their message types. With the DTLS protocol, this process is not possible because after the completion of the DTLS handshake protocol, nodes have already agreed on security policies / cipher suits to protect all security messages, which will be done transparently. However, if the application was applied according to the requirements of the application or scenario, it would contribute to reducing the use of available resources and highly likely to increase network performance.

IV. CONCLUSION

IOT's perspective is not only to simplify our daily lives but also to make security benefits. In IOT, the researcher of products involves reconsidering how they make techniques, safe codes and hardware, for example physical, network, application, compliance, etc. Through this report, we focus on an important aspect of IOTs connected to the Internet

 DOI : 10.18231/2454-9150.2018.0128

Protocol. Although these protocols have been researched, but there is still a deep and comprehensive research needed for further study and analysis for various issues such as security and solutions.

CoAP is one of the major protocols defined as a major layer protocol. To secure COAP in IOT, DTLS and IPsec protocol have been proposed. This paper examines the proposed protocol and analyses such implementation to secure COAP. Given that security can be a competent, there are many factors of many such applications, mechanisms to safeguard communication technologies for IOT. In the survey, with such aspects in mind we do a thorough analysis on the security protocol and available mechanisms for security on communication to secure CoAP. We also address current research proposals and challenges Provide opportunities for future research work in the field.

REFERENCES

[1] Reem Abdul Rahman and Babar Shah, *Security analysis of IoT protocols: A focus in CoAP*, in 3rd MEC International Conference on Big Data and Smart City (ICBDSC), 2016.

[2] T. A. Alghamdi, A. Lasebae, and M. Aiash, *Security analysis of the constrained application protocol in the Internet of Things*, in proceedings of 2nd IEEE International Conference on Future Generation Communication Technology (FGCT), UK, Nov 12-14, 2013.

[3] J. Granjal, E. Monteiro and J. Silva, *Security for the Internet of Things: A Survey of Existing Protocols and Open Research issues*, IEEE Communications Surveys & Tutorials, Vol. 17, no.3, pp. 1294-1312, 2015.

[4] Constrained Application Protocol for Internet of Things, Available at: https://www.cse.wustl.edu/~jain/cse574-14/ftp/coap/index.html

[5] Mario Frustaci, Pasquale Pace, Gianluca Aloi, "Securing the IoT world: Issues and perspectives", *Standards for Communications and Networking (CSCN) 2017 IEEE Conference on*, pp. 246-251, 2017.

[6] M. Brachmann, O. Garcia-Morchon, and M. Kirsche, *Security for practical Coap applications: Issues and solution approaches*, in proceedings of the 10th GI/ITG KuVS Fachgesprch Sensornetze (FGSN), Germany, 2011.

[7] M. Kovatsch, M. Lanter and Z. Shelby, *Californium: Scalable cloud services for the internet of things with CoAP*, in proceedings of the IEEE 4th International Conference on Internet of Things (IOT), USA, Oct 6-8, 2014.

[8] S. Raza, H. Shafagh, K. Hewage, R. Hummen and T. Voigt, *Lithe: Lightweight secure CoAP for the internet of things*, IEEE Sensors Journal, Vol. 13, no. 10, pp. 3711-3720, 2013.

[9] A. Bhattacharyya, T. Bose, S. Bandyopadhyay, A. Ukil, and A. Pal, *LESS: Lightweight Establishment of Secure Session: A Cross-Layer Approach Using CoAP and DTLS-PSK Channel Encryption*, in proceedings of the IEEE 29th International Conference on Advanced Information Networking and Applications Workshops, South Korea, Mar 24-27, 2015.

 DOI : 10.18231/2454-9150.2018.0128