ASTES

Advances in Science, Technology and Engineering Systems Journal Vol. 4, No. 1, 317-328 (2019)

www.astesj.com

ASTESJ

ISSN: 2415-6698

Application Layer Security Authentication Protocols for the Internet of Things: A Survey

Shruthi Narayanaswamy, Anitha Vijaya Kumar*

Department of ECE, Dayananda Sagar College of Engineering, Bangalore 560078, Karnataka, India

ARTICLE INFO

Article history:
Received: 15 December, 2018
Accepted: 04 February, 2019
Online : 28 February, 2019

Keywords:
IoT
Wireless Sensor Networks (WSNs)
Application Layer
Security Authentication Protocol
Vulnerabilities

ABSTRACT

Network security challenges due to nearly limitless internet connectivity, platform limitations, ubiquitous nodal mobility and huge data transactions is burgeoning by the day and the need for transcend Internet of Things (IoT) based cloud security authentication protocols is on an exponential rise. Even though many secure classic layered security mechanisms are available for implementation, they cannot be applied on IoT devices because of the huge energy that they consume. The essence of the paper is an attempt to revisit the existing IoT based security authentication protocols operating in the Application Layer (AL) , AL being the end user's actual service provider. This gateway to the outside world definitely demands stringent and safe data handling and processing. The main objective of the paper is to highlight the positives of the AL protocols and also take a note of the drawbacks in terms of security and defensive measures. The author intends to support the users with information sufficient enough to decide on the type of protocol based on the application. The paper helps the future researchers to have a comparative analysis of each AL protocol's performance and further work on effective improvised defensive measures to tackle the threat-prone IoT environment even better. The paper discusses the architecture implementations, security provisions as well the pros and cons of certain avowed AL protocols currently being used in an IoT environment. Furthermore, the vulnerabilities and possible open issues currently encountered in the AL contribute valuably to the paper since they unravel the path to future research opportunities for secure interconnection of communicating devices.

1. Introduction

IoT envisages millions of communicating nodes with sensing, actuating and processing capabilities actively connected to the Internet and the number of physical objects eyeing to get connected to the Internet is booming to an unprecedented rate. IoT environments require their sensory nodes to sense continuously and communicate with the environment; needless to say the polling method of data collection fails.

Almost all layers of the protocol stack are vulnerable to security threats and attacks. Layered security protections have to be introduced to combat unique physical security concerns of IoT [1]. It is vital to bring in the security mechanisms of existing IoT based protocols, analyze existing open research security issues, and evolve with better security mechanisms for existing protocols and a step ahead to innovation of many more ingenious IoT based protocols. Significant obstacles in IoT security involve Application, System, Communication Network and Infrastructure security [2]. Also, IoT still does not have global policies and standards to standardize application development, interaction and implementation. Hence, best security practices and standards requirements must evolve to enhance data integrity.

The Application Layer on the top of the protocol stack is the most open ended of all of the layers providing the widest attacker surface to hackers and hence is more vulnerable to network threats when compared to the other layers of the stack. All application dependent high level functions operate from this layer. The primary focus of this paper is on Application Layer security, prominent security authentication protocols of the layer and their security implementations. Even though breakthrough researches have made their way into the world of IoT security, each day the network threats and vulnerabilities are not failing to create network troubles. The ever growing jargon of vulnerabilities motivate the author to discuss the existing defensive measures offered by the prominent AL protocols and further provoke

*Anitha Vijaya Kumar, Email: anithavijaya@gmail.com

researchers to evolve with techniques and measures superior to the existing ones. In fact, each drawback tabulated under vulnerabilities encourages us to work on solutions for the same.

The paper is organized as follows. Section II describes few IoT architectures encompassing the essentials of the IoT system namely heterogeneous physical objects, sensors and actuators, data storage and handling and smart network technologies. The section also projects a pictorial representation of an IoT protocol stack. Section III recalls the existing IoT based security authentication protocols based on three classification criteria. Section IV deals with vulnerabilities and security issues, specifically in the Application layer and highlights few contributions attempting for better security mechanisms. Section V focuses on research challenges and required enhancements for the IoT based security authentication protocols followed by the conclusion of the survey in Section VI.

2. Architectural Support

The authors of [3] provide references to four proposed architectures, one of them being the five-layered generic architecture which many IoT implementations relate to. Lack of standardization and common IoT designs encourage researchers to dwell more into generic architectures whereas an efficient standardization would probably drive researchers to fix common security issues much more effectively. The architectures mentioned are: (a) Three layered architecture (b) Middleware based architecture (c) Service oriented architecture (SOA) (d) Five layered architecture.

There is no single consensus over the choice of IoT architectures. The Five layered architecture in which the AL provides an interface to the Business Layer for high level analysis of data. Data accession control mechanisms are mainly handled in this layer. These reasons are quite a reason for network engineers and designers to settle down for the Five- layered architecture comprising of the following layers [4].

(a) Business Layer at the top constitutes the financial and service benefits yielded from the Application layer provided data.
(b) Application Layer defines the various applications in which IoT can be deployed.
(c) Processing Layer is the middleware layer which stores, analyzes and processes to accomplish Service Management.
(d) Transport Layer is responsible for mutual data transfer between Processing and Perception layers using different communication networks.
(e) Perception Layer is responsible for sensing and information gathering from the IoT environment.

The content of this paper is a primitive contribution to the implementation of an IoT based security authentication protocol in the Application layer as shown in the stack diagram above in Figure 1. The protocol flow would definitely prove to be more performance oriented than the regular IP flow in terms of security features and protocol efficiency.

IoT protocols like Constrained Application Protocol (CoAP), Datagram Transport Layer Security (DTLS), User Datagram Protocol (UDP) and IPv6 over Low Power Wireless Personal Area Networks (6LoWPAN) are designed for optimized IP access and smaller data overhead of few tens of bytes in a network of constrained devices.

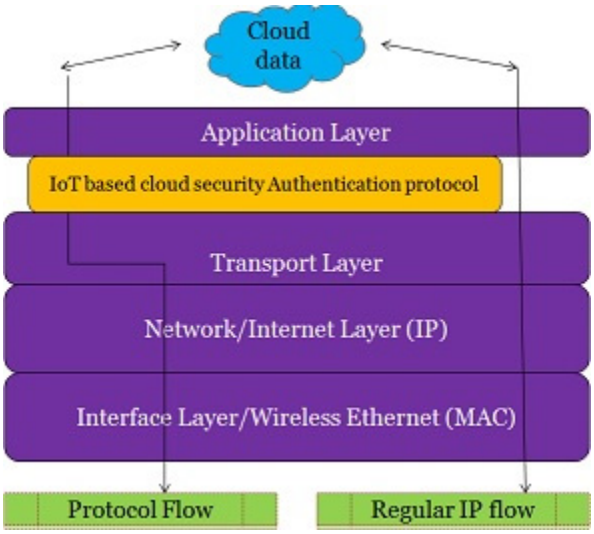

Figure 1. Proposed Architecture Diagram of Protocol Stack

3. Existing IoT based Protocols

The following three approaches define a fine way to categorize the IoT based protocols, the details of which are put down in the table below.

(a) Based on the layer to which the protocol belongs - Application Layer protocols, Network Layer protocols, Data Link Layer protocols
(b) Based on key distribution schemes - Symmetric Key, Asymmetric Key
(c) Based on the nature of IoT application - Application protocols, Service Discovery, Infrastructure protocols, Influential protocols.

3.1. Based on the Layer

Messaging among various subsystems of the IoT environment is enabled by the session layer or transport layer protocols [5] like Message Queue Telemetry transport (MQTT), Secure MQTT (SMQTT), Data Distribution Service (DDS), Advanced Message Queuing Protocol (AMQP), Extensible Messaging and Presence Protocol (XMPP), CoAP, HTTP, Embedded Binary HTTP (EBHTTP), Lean Transport Protocol (LTP), Simple Network Management Protocol (SNMP), IPfix, DNS, Network Time Protocol(NTP), Secure Shell Protocol (SSH), Device Language Message Specification (DLMS/COSEM), Distributed Network Protocol (DNP), MODBUS. All these protocols are built on either TCP or UDP. However, the protocol stack standardized by Institute of Electrical and Electronics Engineers (IEEE) and Internet Engineering Task Force (IETF) shows the Application layer as the topmost in the stack [6]. The above list of protocols may belong to Session layer, Transport layer or Application layers.

3.1.1. MQTT and versions

MQTT, led by Dr Andy Stanford-Clark has been standardized by OASIS in 2013 [7] and now is an open standard. MQTT is an IBM's event-driven, lightweight many-to-many communication, publish-subscribe based protocol developed on TCP. MQTT is message-oriented, every message is published to an address, called "topic". Every client subscribed to a topic receives every message published to the topic. An intermediate broker distributes messages from publishers to the respectively demanding client machines. Some of the brokers used are Mosquitto, Really Small Message Broker (RSMB), MQTT.js, HiveMQ, RabbitMQ and VerneMQ. Jose Luis Espinosa-Aranda et.al has proposed a tiny open-source MQTT broker for flexible and secure IoT deployments in [8].

The message format shown in Figure is usually expressed as 2-byte fixed header, a variable length header and payload, out of which the fixed header is mandatory whereas the other two are optional.

<table>
<tr><td colspan="4">Fixed Header field (minimum 2 bytes)</td><td rowspan="3">Variable Length Header (size depends on the type of message)</td><td rowspan="3">Variable Length Payload (Payload refers to the data sent)</td></tr>
<tr><td colspan="3">Control Header (1 byte)</td><td rowspan="2">Packet Length (1 to 4 bytes)</td></tr>
<tr><td>Message Type</td><td>DUP flag</td><td>QoS Level</td><td>Retain</td></tr>
<tr><td>4bits</td><td>1 bit</td><td>2 bits</td><td>1 bit</td><td></td><td></td></tr>
</table>

Figure 2. MQTT Message Format

Message types include CONNECT, CONNACK, PUBLISH, PUBACK, PUBCOMP, SUBSCRIBE, SUBACK, UNSUBSCRIBE and many more. The DUP flag when set conveys to the receiver of already having received the data and indicates duplication. The QoS field indicates the delivery assurance assisted by three modes/profiles namely (a) Fire and forget/ At most once/ QoS0 (b) Acknowledged delivery/ At least once/ QoS1 (c) Assured delivery/ Exactly once/ QoS2.

MQTT is TCP/IP based and designed for constrained devices and low-bandwidth, high-latency networks, best suited as communications bus for live data. MQTT is therefore, an ideal messaging protocol for IoT and M2M communications. MQTT ensures reliability by providing three QoS levels. Semantic data extraction is supported by MQTT protocol and is one of the best suited paradigms for IoT [9], especially on battery-run devices. In fact, MQTT outperforms CoAP in managing higher traffic, lower latency, higher throughput, optimal memory, low power operation and CPU usage [10].

The MQTT design is suitable to operate in secure networks and has no security mechanisms imposed. Security in MQTT is based on SSL/TLS encryption, a relative standard for authentication in an IoT environment. A matter of concern is that SSL/TLS is quite expensive to be used for a constrained IoT environment. SMQTT is secure MQTT in which a message is encrypted and delivered to multiple nodes which is suitable for IoT applications. This broadcast encryption feature dependent algorithm of SMQTT has 4 stages of operation namely setup, encryption, publish and decryption. MQTT- SN v1.2, formerly known as MQTT-S is a dedicated MQTT version for Sensor Networks handling embedded devices on non-TCP/IP networks, such as Zigbee. MQTT-SN too is a publish/subscribe messaging protocol operating beyond the reach of TCP/IP infrastructure i.e. UDP based for Sensor and Actuator solutions. MQTT-SN envisages power constraint oriented communication with a UDP platform and adds broker support to index topic names unlike MQTT. Secure versions SMQTT and SMQTT-SN have been augmented to MQTT and MQTT-SN respectively based on an attribute-based Key/Cipher Text Policy using Elliptic Curve Cryptography (ECC). The authors of [11] have explained the possible solutions in MQTT systems to implement different protection levels from varied network threats; however, ECC always has played a good choice for MQTT implementations. MQTT is finding its way into many domains [12] like Healthcare, Energy and Utilities, Industry and Irrigation systems, Social Networking and many IoT based applications.

The protocol has a low overhead in spite of operating on TCP when compared to other TCP based Application layer protocols [13]. MQTT can carry only a maximum of 256 MB of data and is hence suitable for expensive, unreliable networks. MQTT also experiences lower delays; uses bandwidth and battery moderately and hence well preferred in lower delay message delivery applications. The limitations of MQTT include limited security, broker overloading and hence message expiry, message ordering challenge and no message priority principle. The authors of [14] have experimentally compared the protocol efficiencies of CoAP, MQTT and WebSocket and revealed the mediocre performance of MQTT with QoS1 in terms of protocol efficiency. Results show CoAP to be the best, followed by WebSocket and MQTT with QoS0.

3.1.2. CoAP

CoAP is the brainchild of CoRE (Constrained Resource Environments) IETF group and enables web applications on smart objects [15]. CoAP is a one-to-one protocol best suited for a partially event based state transfer model and is built on UDP to provide a reliable low weight mechanism. CoAP provides a request and response communications model and supports end-to-end communication at the application layer between constrained IoT devices and other Internet devices. It works similar to HTTP in order to benefit from existing web-based technologies using the same methods (GET, PUT, POST, and DELETE) as HTTP, but with an additional ability for resource discovery and observation [16].

A standard interface called Representational State Transfer (REST) is the standard interface used between client and the servers in CoAP. CoAP operates with a 2-layer convention of Request/Response and Transaction/Messaging. The Request/Response layer manages the REST operation and the Messaging layer ensures reliable UDP communication with the help of exponential backoff.

The four messages of CoAP are Confirmable (CON), Non-Confirmable (NON), Acknowledge (ACK) and Reset (RESET).

A typical CoAP message is 10 to 20 bytes, the first fixed part is a 4-byte header, a token, Options and Payload fields which are not mandatory. The details of the CoAP message format are depicted in Figure 2. "Ver" is 2 bit unsigned integer for the version number of CoAP, followed by another 2 bit unsigned integer "T" to indicate the type of messages (CON, NON, ACK or RESET). The 4 bit "TKL" field represents the token. The 8 bit "Code" splits as a 3 bit class and 5 bit detail for MSBs and LSBs respectively. "Message ID" is used for matching responses and message duplication indication.

Ver	T	TKL	Code	Message ID
2 bits	2 bits	4 bits	8 bits	16 bits
Token (optional) – 0 to 8 bytes				
Options (optional)				
Payload (optional)				

Figure 3. CoAP Message Format

CoAP promises to fulfil low overhead, asynchronous message exchanges, URI and content-type support, simple parsing, enhanced reliability due to data reduction, reduced latency in low-power lossy wireless networks and multicast communication support in IoT based resource constrained environments. Other important features include Resource Observation, Block-wise resource Transport, Resource Discovery and easy interaction with HTTP [17,18].

Comparing MQTT with CoAP in terms of overhead, CoAP allures with its appreciable low overhead. However, due to the dearth of TCP retransmission mechanisms, packet losses tend to be on the higher side. CoAP races over MQTT with lesser traffic generation in the case of small-sized messages. CoAP outruns MQTT with lower delays but only when packet loss rate is high. On the contrary, for lower packet loss rates, MQTT delivery rates are comparatively quicker.

The authors of [19] propose an adaptive (Retransmission Time Out) RTO method rather than a fixed RTO, which consists of a Smooth Round-trip Time getting multiplied by a constant parameter (K) to reduce energy consumption of nodes by nearly 8% and improve reliability enhanced packet delivery ratio (PDR) of MQTT-SN and CoAP protocols. A comparative study by the authors of [20] for a Smartphone application between CoAP and MQTT showed that CoAP's bandwidth usage is lesser than that consumed by MQTT. The authors of [21] have provided recorded results demonstrating the better side of CoAP in terms of energy usage and transmission time. Other metrics have also been analysed such as discarded publication message ratio, retransmitted message ratio and duplicated message ratio in support of the adaptive RTO method. The authors here state that a maximum achieved PDR is better with CoAP than what is achieved with MQTT-SN. CoAP does offer a basic congestion control mechanism for unicast messages, better congestion control mechanisms are required to only handle the gradually increasing traffic of multicast communications.

CoAP secure communication was earlier based upon IPSec [22]. Since CoAP is built on UDP, SSL/TLS cannot provide security but can be achieved with DTLS. CoAP backed up by DTLS is unitedly termed as Secured CoAP (CoAPs). DTLS with enhanced features of TLS to deal with UDP communications of CoAP succeeds in targeting Confidentiality, Integrity, Non-Repudiation and Data Protection against Replay Attacks. CoAP provides inbuilt support for content negotiation and discovery thereby allowing devices to probe each other to find ways of exchanging data. The introduction of raw public keys with compressed DTLS in CoAP reduces message size and hence energy savings, avoidance of 6LoWPAN fragmentation at the link layer for larger datagram sizes and reduced burden on constrained devices during DTLS handshake. The suitability of DTLS and IPSec for CoAP security implementations is questionable in spite of their usage in many IoT based applications. Secured CoAP or CoAPs by implementing the three modes of DTLS namely:

(a) RawPublicKey mode – An asymmetric raw public key pair is generated by the manufacturer and installed on the device. However, devices may have one or more raw public keys.
(b) PreSharedKey – This mode is based on a list of pre-shared keys. Each key in turn includes many communicating nodes. The communication process to a new node includes a DTLS session start using the pre-shared key, the system selecting an appropriate key based on the destination nodes.
(c) Certificate – The devices operating in this mode use a root trust anchor - validated X.509 certificate with an asymmetric key pair.

Although security implementations like DTLS for CoAP is a necessity, one should also be aware of the fact that quite a significant overhead will be added in constrained environments, thereby challenging the limitations on memory and/or bandwidth. A sensible solution would be to dislodge from unused nodes and make the protocol lighter and re-introduce them only when required. The suitability of DTLS and IPSec for CoAP security implementations is questionable in spite of their usage in many IoT based applications. However, DTLS does not support multicast communications since it lacks group key management. IPSec faces Network Address Translation (NAT), Port Address Translation (PAT) and multicast communication issues. Both IPSec and DTLS have an incompetent QoS, Access Control and network trust and rely upon out-of-the-box extra protocols like Extensible Authentication Protocol (EAP) and Internet Key Exchange (IKE).

On the contrary, CoAP is also known for its high latency, poor packet delivery and inability to be used for complex data types [23]. MQTT and MQTT-SN are quite prevalent than CoAP and find applications in the area of social networks, Vehicle to Vehicle communication (V2V) and sensor networks.

In spite of the constantly evolving upgrades of CoAP, cost, power efficiency, supreme data security, network robustness and application deployment, gullibility of a CoAP based system still

remains a challenge. In fact, the generic CoAP can no longer be used with increased number of transmissions and network congestion. The authors of [24] mention and discuss about an advanced CoAP termed as CoAP Congestion Control Mechanism (CoCoA). Comparative study is made based upon parameters like latency, throughput and re-transmission. The authors in [25] have carried out a CoCoA analysis and have implemented a 4-state-Strong CoCoA adaptation that uses a 4-state estimator for variable backoffs. Results signify an improvement in throughput and goodput even in highly lossy networks. CoCoA+ is add on to the CoAP and CoCoA and the drafters of this mechanism in [26] prove the upper hand performance of CoCoA+ with many use cases in a variety of network topologies. On the contrary, one of the authors in their works uplifts the degradations of CoCoA+ when compared to generic CoAP [27]. The results indicate that CoCoA+ can perform significantly worse than default CoAP, especially with burst traffic and in networks with few clients as a result of an improper selection of the retransmission timeouts (RTOs).

3.1.3. XMPP

XMPP originally coined as "Jabber" is a well demonstrated IETF protocol which provides both asynchronous (publish/subscribe) and synchronous (request/response) messaging supports. This TCP based, instant messaging standard protocol supports a variety of authentication patterns via the Simple Authentication and Security Layer (SASL – RFC4422). XMPP was designed for near real-time communications and therefore it supports small message imprint and low latency message exchanges [28] and is used in multi-party chatting, voice and video calling. XMPP was extended to IoT applications because of its eXtensible Markup Language (XML) feature, addressing, security and scalability features.

In terms of security, SASL provides a set of authentication methods from which the client can choose the best fit. SASL uses Base64 coding to hide recognizable information. While SASL is responsible for authentication, TLS looks after channel encryption for XMPP. XMPP is fulfilling the needs of IoT cloud providers in terms of message management and security. However, XMPP lacks native advanced security features to address security requirements of emerging federation-enabled IoT cloud scenarios [29]. The authors of [30] provide a security mechanism for XMPP based communication in sensor networks as well, but at the cost of extra overhead.

The overhead of XMPP too remains a concern to be used in IoT sometimes and requires a makeover in preferably the architecture. The cons are additional overhead due to gratuitous tags, increased power consumption due to complex computation and not many QoS options. In an attempt to unify XMPP with IoT, the authors of [31] have proposed a solution to unify sensors and actuators with Internet by omitting application protocol gateways and protocol translators. XMPP has been evolving from a simple Instant Messaging (IM) system to Cloud Computing.

3.1.4. DDS

DDS is a data-centric, PKI based certificate authentication protocol based on a brokerless, publish/subscribe architecture and hence more reliable with impressive QoS and suited for M2M as well as IoT. Object Management Group (OMG)'s DDS uses multicasting and also supports token mechanism catalysed by RSA and DSA algorithms. DDS uses a device-to-device relational data model to transfer data directly to the device using bus communication. DDS architecture is 2 layered as Data-Subscribe Publish-Subscribe (DCPS) and Data-Local Reconstruction Layer (DLRL). DCPS delivers data to subscribers. DLRL is an optional interface to DCPS. DDS is a standards-based QoS-enabled data centric middleware platform that enables applications to communicate by publishing information they have and subscribing to information they need in a timely manner [32]. DDS offers detailed QoS control, multicast, configurable reliability and pervasive redundancy [33] and resolves data distribution and management challenges [34].

3.1.5. AMQP

AMQP is a message-centric standard which is based on the publish/subscribe architecture similar to MQTT and runs on TCP. AMQP is an open standard used to send large number of messages per second [35] when compared to other RESTful services.

Exchanges and message queues constitute the AMQP broker and exchange information between each other according to pre-defined protocols. The exchanges route messages to appropriate queues. Queues store the received information and deliver to appropriate subscribers when required.

The key capabilities of AMQP are its ability to connect across technologies, organizations and time domains and hence, AMQP finds applications based on control plane or server-based analysis functions. AMQP is not very suitable for constrained environments and real-time applications. It does not support automation discovery too. However, AMQP is well interoperable in multiple environments. AMQP 1.0 is now approved as an International standard and has become an OASIS standard too.

3.1.6. EBHTTP

EBHTTP is a space-efficient, binary formatted, stateless encoding of the standard HTTP/1.1 protocol. It is used to transfer smaller messages in a constrained environment [37].

3.1.7. LTP

LTP allows constrained nodes/devices to exchange web service messages. The authors of [38] penned this versatile, light-weight Web service transport protocol in 2010. LTP allows the transparent exchange of Web Service messages between all kinds of resource-constrained devices and server or PC class systems.

3.2. Based on Key Distribution Schemes

IoT security solutions may either rely upon asymmetric key schemes or pre distributed symmetric keys. Each of the categories

Table 1: Summary of the Application Layer Protocols

Protocol	Archi-tecture	Transp ort	QoS	Secu-rity	Areas of Application	Advantages	Limitations
MQTT	Asynch-ronous	TCP	Yes	SSL	Healthcare, Energy & Utilities, Industry & Irrigation, Social networking, IoT based applications	Low overhead, delays and power consumption, high latency, better than CoAP in traffic management, higher throughput, optimal memory and CPU usage	Moderate bandwidth and battery usage compared to CoAP
CoAP	Synch-ronous	UDP	Yes	DTLS	Live data communication, sensor networks, IoT based applications	URI & Content-type support, enhanced reliability, reduced latency, single parsing, multicasting, reduced bandwidth usage, good PDR	Packet losses due to TCP retransmissions, high cost, network robustness, application deployment gullibility
DDS	Asynch-ronous	TCP/U DP	Yes	SSL DTSL	M2M and IoT based applications, air traffic and vehicle control systems, industrial automation systems	Excellent QoS control, configurable reliability, pervasive redundancy, multicasting	Limited scalability, resiliency in data delivery, network heterogeneity
EBHTTP	Asynch-ronous	UDP	No	SSL	Applications involving transfer of smaller messages in constrained, hypermedia information systems	Resource discovery due to RESTful design, simplicity in design, extensibility of HTTP to suit highly constrained networks	No support for fragmentation, must follow HTTP caching behaviour
LTP	Synch-ronous	TCP/U DP	Yes	SSL	Web service message exchanges	Standard -compliant to Web services, combines with microfiber to give SOAP messages, header compression, message fragmentation	High implementation and maintenance cost
XMPP	Synch/Asy nc	TCP	No	SSL	Voice & Video calls, chatting & message exchange applications	Good to use if application is already built and running with XML	High power consumption due to complex computations, additional overhead, no QoS and not suitable for M2M
AMQP	Asynch-ronous	TCP	Yes	SSL	Applications based on control plane & server-based analysis functions	Can connect across technologies, organizations and time domains, store-and-forward strategy for good reliability	Not suitable for constrained real-time applications, no support for automation discovery

has its own pros and cons. However, researchers use both of these and optimizing the existing solutions of Asymmetric and Symmetric Key schemes continues to be the area of prominence.

3.2.1. Asymmetric Key Schemes (AKS)

Asymmetric algorithms are commonly deployed in conventional internet. But, AKSs quote comparatively higher computation cost and consume higher energy for operation. Such schemes are widely implemented for IoT based applications since they offer high resilience against node capture attacks, have low memory requirements for keying materials, few message exchanges and high scalability for large networks. Asymmetric approaches can be Public key encryption key based transport wherein a public key (secret code) is the authentication link to information sharing parties. This approach is more vulnerable to man-in-the-middle attacks. The key establishment techniques

may range from simple traditional mechanisms like raw public key encryption, certificate based encryption and identity based encryption to higher levels of complicated X.509 based implementations. Security certificates are expensive and hence require few hardware and software improvements in design. The need for optimization and cryptographic hardware accelerators arises. Identity Based Schemes (IBSs) provides a well-known identity which acts as the public key. A trusted party called the Public Key Generator (PKG) generates the private key of each entity. Even though certificates are eliminated here, IBSs are prone to key-escrow attacks. IBSs based implementations like RSA or ElGamal type IBE, IBAKA, TinyIBE are already being used in many applications.

3.2.2. Symmetric Key Schemes (SKS)

There is a demand for high memory space for keying materials, lower scalability for wider networks and vulnerability against node capture attacks. Therefore, SKS cannot be considered as the default protocol for IoT. Diffie-Hellman (DH) protocol is expensive and not suitable for constrained environments, a variant of DH called the Elliptic Curve DH (ECDH) protocol helps. ECDH is based on Elliptic Curve Cryptography (ECC) and has a relatively smaller key size than with the RSA algorithm. A Digital Signature Algorithm ECDH-EDSA algorithm [39] is an effective key agreement protocol too. The HIP-DEX algorithm generates an ECDH encrypted session key between two entities after 4 messages and uses the least number of cryptographic primitives. Many IoT based works rely upon HIP-DEX. The authors of [40] have reused ECDH boosted with the session resumption mechanism. The authors of [41] came up with a combination of ECDH-IBE, IBE uses an ECC primitive. However, it still demands 2 bilinear pairings and 3 scalar point multiplications each time a session key is bootstrapped. As an attempt to eliminate pairing, TinyIBE [42] was proposed in which the session key between two nodes is retrieved after just two messages.

3.3. Based on nature of IoT Application

The authors of [43] have attempted to categorize IoT protocols based on the nature of IoT applications and their core functionalities. The categories are the Application protocols, Service Discovery protocols, Infrastructure protocols and Influential protocols. The Application protocols include DDS, CoAP, AMQP, MQTT, MQTT-SN, XMPP and HTTP REST.

The Service Discovery protocols are listed as Multicast DNS (mDNS) and DNS Service Discovery (DNS-SD). mDNS is well suited for Internet based embedded devices since its working is unaffected by infrastructure failure. mDNS can execute the unicast DNS server operation. The working operation of mDNS can be analyzed as follows. mDNS sends IP multicast messages at once to all nodes in its local domain inquiring by NAME for the preferred client node. When the target node receives its NAME, it will send a response to the calling mDNS along with its IP address. Devices which receive the response message will update their NAME and IP address in their respective local cache.

DNS-SD is similar to mDNS with respect to the fact that it too does not require additional manual configuration or administration. In fact, the client machines use mDNS and pair required services to constitute the DNS-SD. mDNS finds the required services by host name and pairs their IP addresses with them. Since mDNS are DNS-SD are configuration independent protocols, they are suited for IoT based implementations wherein smart devices can join or quit the platform without affecting the entire system operation. On the contrary, these protocols demand caching DNS entries for constrained devices and cache handling and timing operations can be challenging enough to consider other protocols instead.

The Infrastructure protocols are actually the whole sum set of Network, Link and Physical layers which are Long Term Evolution – Advanced (LTE-A), EPCglobal, IEEE 802.15.4, Z-Wave, 6LoWPAN, IPv4, IPv6 and Routing Protocol for Low Power and Lossy Networks (RPL). LTE-A promises reasonable service costs and scalability as far as cellular solutions matter. Architectural essentials include the Core Network (CN) dealing with packet flows and device control and Radio Access Network (RAN) for radio access. Base stations typically called evolved nodes and represented as eNBs connect each other through the X2 interface. RAN and CN connect through S1 interface. And finally, other mobile devices connect through the gateway. LTE-A uses the Orthogonal Frequency Division Multiple Access (OFDMA) to partition bandwidth into smaller bands called Physical Resource Blocks (PRBs). Problems of QoS compromise and network congestion come along with LTE-A protocol, solutions however exist to lower contention in network.

EPCglobal manages Electronic Product Code (EPC) and RFID technologies and standards. Its architecture supports good interoperability, reliability and scalability. The entire RFID based tag system works on two components – the tag and tag reader. A chip in the tag is the storage element which has an object's unique identity. This chip communicates with the tag reader with an antenna using radio waves. The tag reader passes over the unique identity/tag number to a computer application named Object Naming Service (ONS) which further interacts with the IoT applications.

IEEE 802.15.4 finds a reasonable place in the choices for IoT, M2M and WSNs due to its reliability in communication, low cost, power consumption and data rate, high throughput and interoperability but at the cost of poor QoS. The protocol uses three Direct Sequence Spread Spectrum (DSSS) modulation technique.

Z-Wave is a low power protocol preferred for low distance data transmissions typically of few meters and hence finds applications in home appliances, light control, access control, wearable technology etc. The architecture comprises of the

controller and slave nodes, controller maintains a table for updating and hence monitoring routing strategies of the topology.

6LoWPAN supports IPv6 with mapping services, provides fragmentation and header compression (headers typically compressed to two bytes [44]). Link layer forwarding for multi hop delivery and IPv6 overhead reduction are added features of 6LoWPAN.

RPL supports multipoint-to-point, point-to-multipoint and point-to-point communications. The essence of RPL is a directed acyclic graph with a single root node called Destination Oriented Directed Acyclic Graph (DODAG) responsible for routing. The RPL routers work in either Storing mode or Non- Storing mode. In the former, destination IPv6 addresses direct the downward routing whereas, in the latter, IP source routing come into picture.

There is the Influential protocol category that includes IEEE 1888.3, IPSec and IEEE 1905.1. It is evident that IoT environments have many underlying technologies and interoperability is essential and this category of protocols aims for the same. In fact, IEEE 1905.1 standard was designed for heterogeneous technologies and convergent digital networks.

4. AL Security – Vulnerabilities and Issues

IoT Security Systems Engineering is constantly evolving with state-of-the-art security approaches to counter the exponentially growing "headless" security threats. Defining and designing a protective architecture is definitely a security requirement at the system or architecture level. However, we restrict our discussion to protocol based security authentication, especially at the Application layer. Achieving end-to-end security triggers network challenges due to the discrepancy between the high demand for security standards and the available envisioned constrained hardware. Unprotected protocols (without security based implementations) are often vulnerable to various network attacks, eavesdropping, spoofing etc. Having SSL/TLS, IPSec, DTLS or any other security mechanism still does not assure the protocol of flawless security. In fact, IPSec faces Network Address Translation (NAT), Port Address Translation (PAT) and multicast communication issues. DTLS does not support multicast communications since it lacks group key management. Both IPSec and DTLS have an incompetent QoS, Access Control and network trust and rely upon out-of-the-box extra protocols like Extensible Authentication Protocol (EAP) and Internet Key Exchange (IKE). SSL/TLS is expensive to be used in constrained devices.

Vulnerabilities are the weaknesses of a system due to poor design which allow the network to be hacked illegally. An attacker may bank upon improperly maintained network access and permissions, buffer overflow, cross site scripting, error configurations, data tampering and poor data authentication mechanisms. The authors of [45] provide a classification for security threats in the Application layer. They are Privacy Leak, DoS Attack, Malicious Code and Social Engineering.

Another major setback to AL security has ever since been the distribution of keys among devices [46]. Few solutions like vendor based access control and virtual networks have helped, but not been a major breakthrough to handle key distribution issues very effectively. General security measures and counter attacks can be put up as Data Security, Authentication, Trust Management, Risk Assessment and Intrusion Detection. However, below is a tabulation of the possible vulnerabilities and challenges threatening the Application Layer.

5. Research Challenges and Proposals

Research still prevails to minimize potential threats and probable network attacks. The authors of [68] had proposed a DTLS improvement to send multiple CoAP messages in a multicast group using a common group key. Large buffers are required at the receiver end to hold data for retransmission due to inadequate timers in DTLS and code size required to support DTLS. Stateless compression of DTLS headers help to reduce overhead [69]. There are few DTLS header compression implementations, one of them being the usage of LOWPAN_IPHC 6LoWPAN [70]. The authors of [71] proposed the RESTful DTLS handshake to confront the fragmentation limitation. Larger messages are transferred in blocks. To mitigate costs of DTLS operations common security gateways are mapped between TLS and DTLS as well as between CoAP and HTTP. Mutual authentication using DTLS not using ECC too was a proposal of an end-to-end architecture that used specialized trusted-platform modules (TPM) that supports RSA cryptography. Public-key and Digital Certificates support involve computational complexity.

Works attempting to optimize this complexity [72] have come up with certification pre-validation and session resumption to eliminate the need for additional handshake. The authors of [73] have proposed an optimized DTLS integration within CoAP with minimum ROM usage and ECC technique. The proposal highlights block wise message transfer and message reordering. Newer assembly routines which use registers more effectively have been added to minimize the number of memory operations and reduce RAM and ROM occupancy. The authors have used an ECC library of their own which are based on TinyECC and Relic libraries further reducing complex operation execution time.

The authors of [74] provide another breakthrough CoAP based Communication Architecture for sensor and actuator networks (CASAN) to reduce device constraints, reduce intelligence (software complexity, code execution) at the minimum needed in constrained nodes and transfer it to a more competent device which acts as gateway between the sensors and Internet. The basic idea is to have a "REST" level communication by providing a RESTful interface for all sensors and simplify application programming tasks. CoAP based communication proves to be in the best list if we are able to minimize the number of intermediate servers and yet provide secure data delivery over large distances. The authors of [75] have proposed a scalable, flexible embedded CoAP solution for Web applications or the

Table 2: Application Layer Threats and Vulnerabilities

Vulnerability/ Challenges		Problem Description	Solutions Proposed
Attacks	DoS	Deceiving node to breach defensive system	➢ Dynamic threat anticipation ASTM [47, 48] – Adaptive learning technique with changing internal parameters ➢ Risk transfer mechanism based security systems [49] ➢ Support for Software Defined Networks (SDNs) architectures [50]
	Sphear Phishing	Luring emails for adversary gains	
	Sniffing	Introduction of a sniffer application into the system	
	Overwhelm	Undue consumption of energy by nodes and bandwidth	
Insecure web interface & Data Privacy		Log and keys leakage at IoT end-node, illegitimate malicious nodes feeding contaminating data and/or accessing critical information (Malicious Code Injection due to end user hacking techniques)	➢ Preference Based Privacy ➢ Protection Method - Third party evaluation, report to service provider and appropriate security level based sensed preferences [51]
Insecure mobile interface & Cloud Interface		Unsecured apps, no Device Lockout, In-Cloud data leakage, Cross site scripting, poorly configured SSL/TSL	➢ Stronger passwords ➢ Testing the interface against the vulnerabilities of software tools (SQLi and XSS) ➢ Using https along with firewalls [52]
Insecure Remote Security Configuration		Fails to implement security measures @ interfaces, IoT end -node, end-device, end-gateway, no security logging, lack of granular permission model, lack of add-on password security options, lack of comprehensive security management	➢ Remote safe configuration ➢ Scalable security enhancement system of the SMC model for distributed resources – SMSC [53] ➢ Simplified security management of network security teams
Insecure Software/Firmware		Threats to system from pirated softwares, malware installations, unencrypted update files, inability to receive timely security patches	➢ Encryption with validation ➢ Anti-virus, anti-adware, firewalls, Real Time Intrusion Detection Systems (IDS) [54] ➢ Security patches ➢ Code with languages such as JSON, XML, SQL and XSS needs to be tested carefully
Insufficient Authentication/Authorization		Lack of multifactor authentication, unsecure password recovery mechanism, Account Enumeration, lack of Role based access, No account Lockout	➢ Cross-layer authentication and authorization ➢ Sensitive information isolation/Data leakage protection ➢ Administrator/Identity Manager authentication ➢ Effective Key coordinate sharing, frequent key coordinate updates [55, 56] ➢ Identity Authentication and Capability based Access Control (IACAC) [57] ➢ Strong Encryption schemes ➢ Cryptographic Hash functions & Feature Extraction – [58] ➢ Decentralized control of authentication using user-dependent security context [59]

Risk Assessment/Trust Management	Lack of convenient tools for real time risk expectancy, threat detection and security reporting, absence of global and standard trust policies	➢ Security quantified in terms of incident and asset loss – CCM [60] ➢ Mutual trust for inter-system security [61] ➢ Agent-based and weight-based trust models
Lack of Protocol Standardization &	Lack of global standards and policies guiding development of security protocols, failure of existing policies to provide 100% protection from threats	➢ Smart Object Lifecycle Architecture for Constrained Environments (SOLACE) [62]
Existing protocols coping with newer & stronger threats	Network bottlenecks are still prevalent in existing security protocols which are only relatively successful (like CoAP) [63]	➢ TLS/DTLS and HTTP/CoAP mapping ➢ Mirror Proxy (MP) and Resource Directory ➢ TLS-DTLS tunnel and message filtration using 6LBR 64-67]

Web of Things (WoT) in general, integrating the browsers and Web clients without intermediate gateways and proxies. Authors of [76] discuss upcoming CoAP options to enhance security in CoAP by highlighting a granular per message based security scheme.

There are a few proposed approaches to few CoAP research challenges discussed below: (i) Group key management mechanisms may be applied externally to CoAP or integrated within the DTLS handshake. (ii) Security gateways can offer intrusion detection and attack tolerance mechanisms [77, 78, and 79]. (iii) Online certification validation can be improved with a foundational idea like the one discussed in [80] about Online Certification Status Protocol (OCSP). Another related work is based on OCSP stapling using TLS Certificate Status Request extension defined in RFC 6066 [81]. (iv) Optimized hardware design to handle computational complexity and cost imposed due to ECC implementations. (v) Support for varying heterogeneous Convergent Networks considering the possible compatibility and performance issues. (vi) Combination of communication paradigms such as open cloud resource access and single hop long range rather than the former multi hop short range communications.

6. Conclusion

IEEE, IETF and International Telecommunication Union (ITU) have provided several standards and security mechanisms in order to cater to the demands of the uprising IoT. However, a designer is free to rise up with an entirely new authentication protocol or bring about modifications in the existing chain of protocols.

Working in an IoT environment involves IoT devices operating in a wireless environment which are constrained in terms of battery life, processing power, and memory which invites a number of networking challenges. Each of the graded and regulated protocols complement each other and work in coordination for the very cause of IoT security in spite of the fact that they behave differently at different layers in their individual operation . Object security is of primary concern rather than the layer security, be it transport or the application layer. Security mechanisms have to be incorporated or embedded within the protocol itself.

However, the primary objective of this study was to lay the foundation to a state-of-the-art security authentication protocol in the Application layer for IoT applications. The paper can help understand the concerns, issues and progress of research ideas to secure the IoT protocols of the Application layer. The review of the most widely used protocols in terms of operation and security indicates that none of them actually are the best. Each protocol has its own pros and cons, but a wise trade-off between protocol parameters has to be made which is purely application dependent. As far as Application Layer security is concerned, it must be rated the topmost priority parameter which needs to be taken care of since it is the most vulnerable layer to the user world of cyber attacks. The end point of this paper leads to the beginning of research ideas to defend the AL, be it improvisations in Trust Management, Key Management Strategies, Intrusion Detection systems, Encryption schemes and many more to follow. Further research intends to provide a broader contribution to improvised key management strategies for AL security generic to any domain. However, choice of protocol would be application and domain dependent and performance may be evaluated experimentally for each of the predominant AL protocols.

Conflict of Interest

The authors declare no conflict of interest.

References

[1] Cloud Security Alliance (CSA), April 2015, 'Security Guidance for Early Adopters of the Internet of Things'.

[2] Keoh, S., Kumar, S., and Tschofenig, H. (2014), 'Securing the Internet of Things: A Standardization Perspective', Internet of Things Journal, IEEE, Vol. 1, No. 3, pp. 265-275.

[3] Lavinia, Nastase 2017, 'Security in the internet of Things: A Survey on Application Layer Protocols', 2017 21st International Conference on Control Systems and Computer Science, IEEE.

[4] Pallavi, Sethi & Smruti, Sarangi 2017, 'Internet of Things: Architectures, Protocols, and Applications', Journal of Electrical and Computer Engineering Volume, Article ID 9324035, 25 pages,https://doi.org/10.1155/2017/9324035.

[5] Tara, Salman & Prof. Raj, Jain 2015,'Networking Protocols and Standards for Internet of Things', Washington University in St.Louis, Recent Advances in Networking.

[6] J, Granjal, E, Monteiro & J, Silva 2010, '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.

[7] D, Locke 2010,'MQ telemetry transport (MQTT) v3. 1 protocol specification', IBM Developer Works Technical Library.

[8] Jose, Espinosa 2015, 'A tiny open-source MQTT broker for flexible and secure IoT deployments', Communications and Network Security (CNS), IEEE.

[9] Satyavrat, Wagle 2016,'Semantic Data Extraction over MQTT for IoTcentric Wireless Sensor Networks', International Conference on Internet of Things and Applications (IOTA) Maharashtra Institute of Technology, Pune, India 22 Jan - 24 Jan, 2016.

[10] Muneer, Yassein, Mohammed, Shatnawi, & Du'a, Al-Zoubi 2016, 'Application Layer Protocols for the Internet of Thing's, IEEE International Conference on Internet of Things and Pervasive Systems, At 22-24 September 2016, Agadir, Morocco.

[11] Giovanni, Perrone, Massimo, Vecchio, Riccardo, Pecori & Raffaele, Giaffreda 2017,' The Day After Mirai: A Survey on MQTT Security Solutions After the Largest Cyber-attack Carried Out through an Army of IoT Devices', Proceedings of the 2nd International Conference on Internet of Things, Big Data and Security (IoTBDS 2017), pages 246-253.

[12] Dipa, Soni & Ashwin, Makwana 2017,' A Survey on MQTT: A Protocol of Internet of Things (IoT)', International Conference on telecommunication, Power Analysis and Computing Techniques (ICTPACT-2017).

[13] Dinesh, Thangavel, Xiaoping, Ma, Alvin, Valera, Hwee-Xian, Tan, Colin, Keng-Yan Tan 2014, 'Performance Evaluation of MQTT and CoAP via a Common Middleware', IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 21-24 April 2014, pp. 1-6.

[14] Stefan, Mijovic, Erion, Shehu & Chiara, Buratti 2016,' Comparing Application Layer Protocols for the Internet of Things via Experimentation', 2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI).

[15] Z, Shelby, K, Hartke & C, Bormann 2013, 'Constrained application protocol (CoAP), draft-ietf-core-coap-18', IETF, 2013.

[16] Angelo, Castellani, Mattia, Gheda, Nicola, Bui, Michele, Rossi & Michele, Zorzi, 'Web Services for the Internet of Things through CoAP and EXI', IEEE International Conference on Communications Workshops (ICC), 5-9 June 2011, pp. 1-6.

[17] C. Bormann, A. P. Castellani, Z. Shelby, "CoAP: An application protocol for billions of tiny Internet nodes", *IEEE Internet Comput.*, vol. 16, no. 2, pp. 62-67, Mar./Apr. 2012.

[18] C. Lerche, K. Hartke, M. Kovatsch, "Industry adoption of the Internet of Things: A constrained application protocol survey", *Proc. IEEE 17th Conf. ETFA*, pp. 1-6, 2012.

[19] Davis, Ernesto, Calveras, Anna & Demirkol, Ilker 2013, 'Improving packet delivery performance of publish/subscribe protocols in wireless sensor networks', Vol. 13, No. 1. Multidisciplinary Digital Publishing Institute, 2013, pp. 648–680.

[20] De Caro, N, Colitti, W, Steenhaut, K, Mangino, G & Reali, G 2013, 'Comparison of two lightweight protocols for smartphone-based sensing', Communications and Vehicular Technology in the Benelux (SCVT), 2013 IEEE 20th Symposium On, pp. 1-6.

[21] Colitti, W, Steenhaut, K, De Caro, N, Buta, B and Dobrota, V 2011,'Evaluation of constrained application protocol for wireless sensor networks,' Local & Metropolitan Area Networks (LANMAN), 2011 18th IEEE Workshop On, 2011, pp. 1-6.

[22] Thamer, Alghamdi, Aboubaker, Lasebae & Mahdi, Aiash 2013,' Security Analysis of the Constrained Application Protocol in the Internet of Things', Second International Conference on Future Generation Communication Technologies (FGCT), London, UK.

[23] Talaminos-Barroso, A, Estudillo-Valderrama, M. A., Roa, L. M.rrrr, Reina-Tosina, J., & Ortega-Ruiz, F 2016, 'A Machine-to-Machine protocol benchmark for eHealth applications–Use case: Respiratory rehabilitation', Computer Methods and Programs in Biomedicine, 129, 1-11, 2016.

[24] Ananya, Pramanik, Ashish, Luhach, Isha, Batra & Upasana, Singh 2017,' A Systematic Survey on Congestion Mechanisms of CoAP Based Internet of Things', Advanced Informatics for Computing Research, July 2017, pp 306-317.

[25] Rahul, Bhalerao, Sridhar, Subramanian & Joseph, Pasquale 2016, 'An analysis and improvement of congestion control in the CoAP Internet-of-Things protocol, Consumer Communications & Networking Conference (CCNC), Jan 2016.

[26] August, Betzler, Carles, Gomez, Ilker, Demirkol, Josep, Paradells 2015, 'CoCoA+: An advanced congestion control mechanism for CoAP', Elsevier, October 2015.

[27] Emilio, Ancillotti & Raffaele, Bruno 2017,'Comparison of CoAP and CoCoA+ Congestion Control Mechanisms for Different IoT Application Scenarios', IEEE Symposium on Computers and Communications (ISCC), Heraklion, Greece.

[28] Sven, Bendel, Thomas, Pringer, Daniel, Schuster, Alexander, Schill, Ralf, Ackermann & Michael, Ameling 2013, 'A Service Infrastructure for the Internet of Things based on XMPP, IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), 18-22 March 2013, pp. 385-388.

[29] Antonio Celesti, Maria Fazio, MassimoVillari,' Enabling Secure XMPP Communications in Federated IoT Clouds Through XEP 0027 and SAML/SASL SSO', Sensors (Basel), 2017 February 17(2): 301, doi: 10.3390/s17020301.

[30] Longhua Guo, Jun Wu, Zhengmin Xia, Jianhua Li,' Proposed Security Mechanism for XMPP-Based Communications of ISO/IEC/IEEE 21451 Sensor Networks', IEEE SENSORS JOURNAL, VOL. 15, NO. 5, MAY 2015.

[31] Michael, Kirsche, Ronny, Klauck 2012,'Unify to bridge gaps: Bringing XMPP into the Internet of Things', Pervasive Computing and Communications Workshops (PERCOM Workshops), March 2012.

[32] Douglas, Schmidt , Angelo, Corsaro & Hans, Hag 2008, 'Addressing the Challenges of Tactical Information Management in Net-Centric Systems with DDS, http://citeseerx.ist.psu.edu.

[33] Gururaj, Kulkarni, Manoor, S, Mitragotri, P V 2017,'Enabling Technologies, Protocols, and Applications: A Detailed Survey on IOT', International Journal of Advance Research, Ideas and Innovations in Technology, Vol 3, Issue 2.

[34] Douglas, Schmidt & Hans, Hag 2008, 'Addressing the challenges of mission-critical information management in next-generation net-centric pub/sub systems with OpenSplice DDS' , Parallel and Distributed Processing, IEEE International Symposium, April 2008.

[35] Joel, Fernandes, Ivo, Lopes, Joel, J P, Rodrigues, C & Sana, Ullah 2013, 'Performance Evaluation of RESTful Web Services and AMQP Protocol', Fifth International Conference on Ubiquitous and Future Networks (ICUFN), 2-5 July 2013, pp. 810-815.

[36] Xi Chen, Constrained Application Protocol for Internet of Things, http://www.cse.wustl.edu.

[37] Nils, Glombitza, Dennis, Pfisterer & Stefan, Fischer 2010, 'LTP: An Efficient Web Service Transport Protocol for Resource Constrained Devices Sensor Mesh and Ad Hoc Communications and Networks (SECON)', 2010 7th Annual IEEE Communications Society Conference, June 2010.

[38] Kim Thuat, Nguyen, Maryline, Laurent & Nouha, Oualha 2015, 'Survey on secure communication protocols for the Internet of Things', Elsevier, ScienceDirect, February 2015, Pages 17-31.

[39] Moskowitz, R & Jokela, P 2013,'Host Identity Protocol version 2 (HIPv2), Draft-Internet, 2013.

[40] De Meulenaer, G et al. 2008,' On the energy cost of communication and cryptography in wireless sensor network', IEEE International Conference on Wireless and Mobile Computing,Network & Communication, 2008.

[41] Yang, L, Ding, C& Wu, M 2013,'Establishing Authenticated Pairwise Key using Pairing-based Cryptography for Sensor Network', 8th Chinacom, 2013.

[42] Szczechowiak, P & Collier, M 2009,'TinyIBE: identity-based encryption for heterogeneous sensor networks', 5th International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2009.

[43] Al-Fuqaha, Guizani, Mohammadi, M, Aledhari, M & Ayyash, M 2015, 'Internet of things: A survey on enabling technologies, protocols and applications, IEEE Communications Surveys Tutorials, vol. PP, no. 99, 2015.

[44] J. W. Hui, D. E. Culler, "Extending IP to low-power wireless personal area networks", *IEEE Internet Comput.*, vol. 12, no. 4, pp. 37-45, Jul./Aug. 2008.

[45] Weizhe, Zhang, Baosheng, Qu 2013, 'Security Architecture of the Internet of Things Oriented to Perceptual Layer', International Journal on Computer, Consumer and Control (IJ3C), Vol. 2, No.2 (2013).

[46] I. Ishaq et al., "IETF standardization in the field of the Internet of Things (IoT): A survey", *J. Sens. Actuator Netw.*, vol. 2, pp. 235-287, 2013.

[47] Abie H., and Balasingham I., "Adaptive security and trust management for autonomic message- oriented middleware", IEEE 6th Int. Conference on Mobile Ad hoc and Sensor Systems (MASS'09), pp. 810-817, 2009.

[48] Sathish Alampalayam Kumar, Tyler Vealey, Harshit Srivastava, "Security in Internet of Things: Challenges, Solutions and Future Directions", 2016 49th Hawaii International Conference on System Sciences, IEEE.
[49] Vipindev Adat ; Amrita Dahiya ; B. B. Gupta, "Economic incentive based solution against distributed denial of service attacks for IoT customers", 2018 IEEE International Conference on Consumer Electronics (ICCE), March 2018.
[50] Akhunzada, A., Gani, A., Anuar, N. B., Abdelaziz, A., Khan, M. K., Hayat, A., and Khan, S. U. (2016). Secure and dependable software defined networks. Journal of Network and Computer Applications, 61, 199–221.
[51] Tao and Peiran, "Preference-based Privacy Protection Mechanism for the Internet of Things", International Symposium on Information Science and En g i n e e r i n g, (ISISE), pp. 531 - 534 2010
[52] OWASP, Top IoT Vulnerabilities, 2016. URL https://www.owasp.org/index.php/Top_IoT_Vulnerabilities
[53] Pierre de Leusse., Panos Periorellis., Theo Dimitrakos.,and Srijith K., Nair., "Self-Managed Security Cell, a security model for the Internet of Things and Services",First International Conference on Advances in Future Internet, pp. 47 – 52, 2009.
[54] S. Raza, L. Wallgren, and T. Voigt, "SVELTE: Real-time intrusion detection in the Internet of Things," Ad Hoc Netw., vol. 11, no. 8, pp. 2661–2674, Nov. 2013.
[55] Q. Wen, X. Dong, and R. Zhang, "Application of dynamic variable cipher security certificate in internet of things," in Int'l Conference on Cloud Computing and Intelligent Systems (CCIS), 1062-1066, 2012.
[56] Tasneem Yousuf, Rwan Mahmoud, Fadi Aloul, Imran Zualkernan, "Internet of Things (IoT) Security: Current Status, Challenges and Countermeasures", International Journal for Information Security Research (IJISR), Volume 5, Issue 4, December 2015.
[57] P. N. Mahalle, B. Anggorojati, N. R. Prasad, and R. Prasad, "Identity authentication and capability based access control (iacac) for the internet of things," J. of Cyber Security and Mobility, vol. 1, 309-348, 2013.
[58] G. Zhao, X. Si, J. Wang, X. Long, and T. Hu, "A novel mutual authentication scheme for Internet of Things," in Int'l Conference on Modelling, Identification and Control (ICMIC), 563-566, 2011.
[59] X. Duan and X. Wang, "Authentication handover and privacy protection in 5G hetnets using software-defined networking," Communications Magazine, vol. 53, 28-35, 2015.
[60] Weiß, S., Weissmann, O., and Dressler, F., "A comprehensive and comparative metric for information security", In Proceedings of IFIP International Conference on Telecommunication Systems, Modeling and Analysis (ICTSM'05), pp. 1-10, 2005.
[61] Y. Xie and D. Wang, "An Item-Level Access Control Framework for Inter-System Security in the Internet of Things," in Applied Mechanics and Materials, 1430-1432, 2014.
[62] IETF SOLACE Info Page. Available online: https://www.ietf.org/mailman/listinfo/solace (accessed on 27 December 2012).
[63] Minhaj Ahmad Khan , Khaled Salah, "IoT security: Review, blockchain solutions, and open challenges", Future Generation Computer Systems, 2017.
[64] M. Brachmann, S.L. Keoh, O.G. Morchon, S.S. Kumar, End-to-end transport security in the IP-based Internet of Things, in: 2012 21st International Conference on Computer Communications and Networks, ICCCN, 2012, pp. 1–5. http://dx.doi.org/10.1109/ICCCN.2012.6289292.
[65] J. Granjal, E. Monteiro, J.S. Silva, Application-layer security for the WoT: extending CoAP to support end-to-end message security for internet-integrated sensing applications, in: International Conference on Wired/Wireless Internet Communication, Springer Berlin Heidelberg, 2013, pp. 140–153.
[66] M. Sethi, J. Arkko, A. Kernen, End-to-end security for sleepy smart object networks, in: 37th Annual IEEE Conference on Local Computer Networks - Workshops, 2012, pp. 964–972. http://dx.doi.org/10.1109/LCNW.2012.6424089.
[67] M. Brachmann, O. Garcia-Morchon, S.-L. Keoh, S.S. Kumar, Security considerationsaround end-to-end security in the IP-based Internet of Things, in: 2012 Workshop on Smart Object Security, in Conjunction with IETF83, 2012, pp. 1–3.
[68] Keoh, S, Kumar, S, Garcia-Morchon, O & Dijk, E 2014,'DTLS-Based Multicast Security for Low-Power and Lossy Networks (LLNs)', 2014.
[69] Hartke, K 2014,' Practical Issues With Datagram Transport Layer Security in Constrained Environments', Issues-01, 2014.
[70] Shahid, R, Daniele, T & Voigt, T 2012,'6LoWPAN compressed DTLS for COAP', Proc. 8th IEEE Int. Conf. DCOSS, 2012, pp. 287–289.
[71] Keoh, S, Kumar, S & Shelby, Z 2013,'Profiling of DTLS for CoAP-Based IoT Applications', draft-keoh-dice-dtls-profile-iot-00, 2013.
[72] Hummen, R, Ziegeldorf, J, Shafagh, H, Raza, S & Wehrle, K 2013,'towards viable certificate-based authentication for the Internet of things', Proc. 2nd ACM Workshop Hot Topics Wireless Netw. Security Privacy, 2013, pp. 37–42.
[73] Angelo, Capossele, Valerio, Cervo, Gianluca, Cicco & Chiara, Petrioli 2015,' Security as a CoAP resource: An optimized DTLS implementation for the IoT', IEEE ICC 2015 SAC – Internet of Things.
[74] Pierre, David & Thomas, No¨el 2015, 'CASAN: A New Communication Architecture for Sensors Based on CoAP', Proceedings of 2015 IEEE 12th International Conference on Networking, Sensing and Control, Howard Civil Service International House, Taipei, Taiwan, April 9-11, 2015.
[75] Miguel, Castro, Antonio, Jara & Antonio, Skarmeta 2014,'Enabling end-to-end CoAP-based communications for the Web of Things', Elsevier, October 2014.
[76] Granjal, J, Monteiro, E & Sá Silva, J 2013,'Application-layer security for the WoT: Extending CoAP to support end-to-end message security for Internet-integrated sensing applications, Wired/Wireless Internet Communication. Berlin, Germany: Springer-Verlag, 2013, pp. 140–153.
[77] Butun, E, Morgera, S D & Sankar, R 2014,'A survey of intrusion detection systems in wireless sensor networks', IEEE Commun. Surveys Tutorials vol. 16, no. 1, pp. 266–282,
[78] Young, M & Boutaba, E 2011,' Overcoming adversaries in sensor networks: A survey of theoretical models and algorithmic approaches for tolerating malicious interference', IEEE Commun. Surveys Tuts., vol. 13, no. 4, pp. 617–641, 2011.
[79] Abduvaliyev, A, Pathan, A, Jianying, Z, Roman, R & Wong, W C 2013,'On the vital areas of intrusion detection systems in wireless sensor networks', IEEE Commun. Surveys Tuts., vol. 15, no. 3, pp. 1223–1237, 2013.
[80] Myers, M, Ankney, R, Malpani, A, Galperin, S & Adams, C 1999,'X. 509 Internet Public Key Infrastructure Online Certificate Status Protocol-OCSP, RFC 2560, 1999.
[81] Eastlake, D 2011,'Transport Layer Security (TLS) Extensions: Extension Definitions, RFC 6066.