A Survey of Protocols and End-To-End Security Models for Internet of Things

Prateek Thapar
GD Goenka University

Dr. Usha Batra
GD Goenka University

***Abstract*—The Internet of Things (IoT) has brought in an era of ubiquitous computing through which Internet will reach practically everywhere with more number of machines using Internet than humans. This unprecedented increase of physical devices getting online poses both an economic revolution as well as threat to cyber security. The protocol stack for present Internet cannot perform efficiently on resource constrained embedded devices. Hence new protocols have been introduced through optimization and adaption of the classic protocols for enabling IPv6 communication on to the embedded sensors. IEEE 802.15.4 standard define the PHY and MAC layer for the resource constrained devices. 6LoWPAN is the adaptation layer that enables IPv6 communication on IEEE 802.15.4 devices. CoAP is the application layer protocol that interacts seamlessly with Hypertext Transfer Protocol (HTTP) for providing web services to constrained devices. Datagram Transport Layer Security rides over the CoAP to provide security at application layer. The security threats, with Internet of Things, have a deeper reach within out lives and hence end-to-end security of this protocol stack is of paramount importance. To this end, first this survey analyzes the protocol stack for resource constrained embedded devices forming Internet of Things. We then cover the security consideration in each protocol layer. Through this paper we present the analysis of the end-to-end security models proposed by researchers worldwide. This survey, as per best of our knowledge, is the first survey analyzing the end-to-end security models for Internet of Things.**

Index Terms:- 6LoWPAN, IEEE 802.15.4, CoAP, DTLS, end-to-end security, internet of things, security.

I. INTRODUCTION

"Internet of Things" the term was first phrased by Kevin Ashton as title of a presentation to Procter & Gamble (P&G) in year 1999 [1]. IoT is the next big step in evolution of Internet. This is the stage where in addition to humans; the machines (things) start interacting through Internet. Albeit, the number of machines / things connected on Internet have already crossed number of human beings using Internet [2]. According to CISCO Internet Business Solution Group (ISBG), the number of things vis-à-vis people connected to Internet exceeded in year 2008-2009. In addition it is expected that by year 2020 there would be about 50 billion devices connected on the Internet. The devices connected on the Internet include embedded systems, electronics and various types of mechanical devices and sensors. These devices interact with each other; and servers through different network interfaces like RFID, 2.4GHz radios, IEEE 802.11, IEEE 802.15, Bluetooth or proprietary radios. These devices comprise of 'Things' in IoT by bringing the mechanical devices onto the Internet.

Internet of Things is a gradual upgrade of Wireless Sensor Networks of embedded devices, where in the Intra networks can be brought onto the Internet [3]. This transformation not only provides expansion in connectivity and accessibility of the sensor network but also allows one sensor network to interact with other through Internet. The applications of sensor networks have seen an explosion on invent of the technology for Internet of things and this is just a beginning. Smart Cities, Smart Homes, Smart Kitchen Electronics, Smart Security Systems, Health Monitoring Systems, Connected Vehicles, Industrial Automation are few of the applications of Internet of Things being implemented currently. The U.S National Intelligence Council's (NIC) reported *"By 2025 Internet nodes may reside in everyday things—food packages, furniture, paper documents, and more. Today's developments point to future opportunities and risks that will arise when people can remotely control, locate, and monitor even the most mundane devices and articles. Popular demand combined with technology advances could drive widespread diffusion of an Internet of Things (IoT) that could, like the present Internet, contribute invaluably to economic development and military capability"* at conference on Disruptive Civil Technologies in year 2008 [4]. Hence the Internet of Things would lead to ubiquitous computing and would form the face of future economic development.

To this end, Section II introduces the protocol stack in general and Internet of Things in specific. Section III then provides basic details on IEEE 802.15.4 PHY and MAC along with comparison on popular trans-receiver modules implementing IEEE 802.15.4. Section IV explains the 6LoWPAN adaptation layer with focus on frame compression and network discovery. In section V application layer protocol CoAP is discussed followed by security consideration in the protocol stack and existing end-to-end security model in section VI. Section VII elucidates the security of IOT sensor node lifecycle followed by the conclusion in section VII.

II. PROTOCOL STACK

A protocol stack is a set of protocols working together in computer network with intent to provide end-to-end communication. Popular OSI Reference model is a reference model identifying seven protocol layers in a protocol stack. TCP/IP is the protocol stack being used for communication over Internet. However, the implementation of TCP/IP protocol stack is resource intensive because of the transmission reliability achieved through packet acknowledgements. Since Internet of Things primarily constitute Internet enabled embedded devices and sensors; mostly battery based intending very low power consumption,

TCP/IP protocol stack is considered to be inappropriate for Internet of Things. In addition the IPv4 protocol address space has already been depleted [5] and do not have any capability to handle the millions of devices on the Internet. At the application layer, HTTP [6] is based on TCP and has a much higher power consumption making it unsuitable for power constrained IoT devices.

Hence, there was a requirement to define a new protocol stack suitable for Internet of Things. In pursuit of specification of a suitable protocol stack, the research lead to the development of IEEE 802.15.4 [7] standard in 2003 for low power wireless personal area network. IEEE 802.15.4 defines the Physical layer (PHY) and Medium Access Layer (MAC) in the protocol suite for Internet of Things. Internet Engineering Task Force (IETF) introduced the IPv6 Adaption Layer standard through the optimization and adoption of communication protocols to form 6LowPAN [8] – [9] in 2007 and the compression format [10] for IPv6 was improved in 2011. IETF working Group (WG) Routing Over Low Power and Lossy Networks (ROLL) in 2012 introduced RPL - IPv6 Routing Protocol for Low-Power and Lossy Networks [11] to provide efficient routing solution for 6LoWPAN running over IEEE 802.15.4. The Constrained Application Protocol (CoAP) [12] is a web transfer protocol developed for constrained nodes and networks. CoAP has been designed to make constrained nodes running 6LoWPAN utilize Internet and seamlessly interact with HTTP while maintaining simplicity for lossy networks. Hence, the above-mentioned protocols form a stack (as depicted in Fig. 1) for implementation of Internet on constrained nodes and lossy networks. In this survey, ROLL – RPL has not been covered and hence the neighbour discovery mechanism for 6LoWPAN in conjunction with IPv6 caters to the network layer functions.

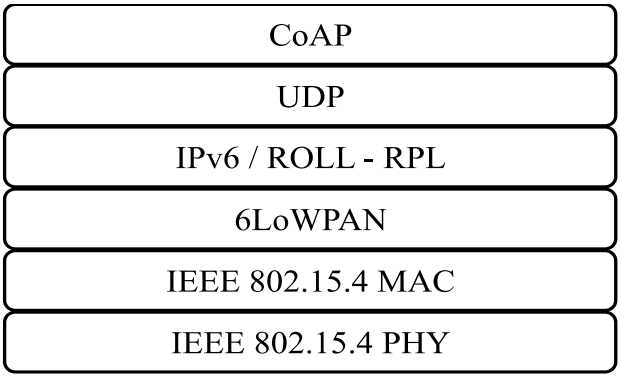

Fig. 1. Protocol Stack for Internet of Things (IoT)

III. IEEE 802.15.4

Low – Rate Wireless Personal Area Network specification under IEEE 802.15.4 standard was released in 2003 with an aim to allow low data rate wireless connectivity for low power and low cost devices. This standard defines Physical Layer (PHY) and Medium Access Control Layer (MAC) for constrained devices. The standard divides the types of constrained devices into two categories:

- Fully Functional Device (FFD): Devices capable of functioning as a coordinator including PAN Coordinator.
- Reduced Functional Device (RFD): Devices not capable of functioning as coordinators in a PAN. RFD can only talk to FFD

The PHY layer of the standard provide following features:

- Star or Peer-to-Peer network topology.
- Each device has two addresses: 64 bit global and 16 bit PAN specific.
- Variety of frequency bands.
- Data rate ranging from 20kbps to 250kbps.
- Activation and Deactivation of Radio trans-receiver.
- Energy Detection (ED).
- Link Quality Indication (LQI).
- Clear Channel Assessment (CCA).
- Precision Ranging in Ultra Wide Band (UWB) PHY.

The MAClayer of the standard provide following features:

- Beacon Management.
- Channel Access by synchronizing to network beacons.
- PAN Association and Disassociation.
- Guaranteed Time Slot (GTS) Management.
- Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA)
- Security Mechanism.
- Contention Access Period (CAP) Maintenance.

IEEE 802.15.4 Trans-Receiver: Trans-Receiver modules in wireless communication are the most power hungry components. Modulator and Power Amplifier (PA) on Transmission end and Demodulator and Low-Noise Amplifier (LNA) at the Receiver end draw maximum current while being operated. However, there is no power consumption while they are OFF. Constrained nodes are designed for low power consumption with battery so as to support device for its lifetime of few years. To support such requirement, various vendors manufacture low power trans-receivers conforming to IEEE 802.15.4. Table I lists characteristics of commercially available 802.15.4 trans-receiver modules (Radio only). A trans-receiver module can be selected out of these options by researchers, developers and system integrators as per the frequency of operation, max data rate and battery life (inversely proportional to current drawn). These requirements may be specific to the embedded devices chosen for the intended applications.

TABLE I

IEEE 802.15.4 TRANS-RECEIVER MODULES (RADIO ONLY)

S. NO.	MANUFACTURER	MODEL	OPERATING FREQUENCY	MAX. DATA RATE	MAX CURRENT TO TRANSMIT	MAX CURRENT TO RECEIVE	ENCRYPTION
1	ATMEL	AT86RF231[13]	2.4 GHz	2Mbps	14.4mA	13.2mA	AES
2		AT86RF232[14]	2.4 GHz	0.25Mbps	13.8mA	11.8mA	AES
3		AT86RF212B[15]	700/800/900MHz	1Mbps	18mA	9.0mA	AES
4		AT86RF233[16]	2.4 GHz	2Mbps	13.8mA	6.0mA	AES
5	TEXAS INSTRUMENTS	CC 2420[17]	2.4 GHz	0.25Mbps	17.4mA	18.8mA	AES
6		CC 2500[18]	2.4 GHz	0.512Mbps	21.2mA	19.6mA	NONE
7	FREESCALE SEMICONDUCTOR	MC13202[19]	2.4 GHz	0.25Mbps	35mA	42mA	AES
8	NORDIC	nRF24L01[20]	2.4 GHz	2Mbps	11.3mA	12.3mA	NONE

As per the data in Table I, AT86RF233 transreceiver module emerges to be the most energy efficient transreceiver providing a max data rate of 2 Mbps on 2.4 GHz. It can be observed that most of the transreceivers support AES for encryption of data.

IV. 6LOWPAN

First 6LoWPAN specification was released in form of RFC 4919 defining the Low Power Wireless Personal Area Networks (LoWPAN) along with the assumptions and goals for transmitting Internet Protocol (IP) packets over IEEE 802.15.4 in year 2007 along with RFC 4944 specifying transmission of IPv6 Packets over IEEE 802.15.4. RFC 4944 was update by two standards namely RFC 6282 specifying the compression format for IPv6 datagram and RFC 6775 which optimized the IPv6 Neighbor discovery to suit 6LoWPAN implementation for constrained devices. In addition RFC 6568 [21] investigates application scenarios and use cases for constrained devices running 6LoWPAN. Also, RFC 6606 [22] defines the routing requirements for implementation of 6LoWPAN.

A. 6LoWPAN Frame Format and Header Compression

IEEE 802.15.4 specifies Maximum Transmission Unit (MTU) of size 127 bytes out of which MAC header can be up to 25 bytes (with nil security) or 46 bytes (with max security of AES-CCM-128). Header of UDP protocol is of 8 bytes and that of IPv6 is 40 bytes. This makes a mandatory maximum of either 54 bytes available for payload in case of nil security and 33 bytes in case of AES-CCM-128 security [23].

The MTU supported by IPv6 is of 1280 bytes and hence to utilize IPv6 over IEEE 802.15.4, Link-Layer fragmentation and reassembly is mandatory. In order to achieve this fragmentation / reassembly operation 6LoWPAN adaptation layer is formed. In addition, to above 6LoWPAN also undertakes compression of IPv6 and UDP/ICMP headers.

Header compression is hence deemed necessary for increasing the payload in MTU of maximum 127 bytes. RFC 6282 defines the compression format of IPv6 datagrams for 6LoWPAN. LOWPAN_IPHC encoding format is used for compressing IPv6 header. LOWPAN_IPHC uses 2 bytes for base encoding and an additional byte if additional context encoding is present. Unicast addresses can be completely omitted or compressed to 64 /16 bits. However, multicast addresses can be compressed to 8 / 32 / 48 bits using LOWPAN_IPHC encoding. In best case, LOWPAN_IPHC can compress IPv6 header to 2 bytes in link-local routing and to 7 bytes in case of routing over multiple IP hops.

LOWPAN_NHC can be used to compress different next headers. Compression format for UDP header using LOWPAN_NHC encoding yields compression of UDP port number to 4 bits and UDP checksum can be elided if authorized by upper layer. Since all the headers encoded using 6LoWPAN-encoding format must be contiguous, the LOWPAN_NHC is preceded by LOWPAN_IPHC or LOWPAN_NHC. In best case, LOWPAN_NHC can compress UDP header to 16 bits where in firs 8 bits are for UDP header encoding, 4 bits for source port and 4 bits for destination port.

B. 6LoWPANNeighbor Discovery

IPv6 Neighbor Discovery mechanism is defined in RFC 4861 [24] that covers router discovery, host address resolution along with Duplicate Address Detection (DAD). Owing to the usage of multicast signaling, the IPv6 neighbor discovery is inefficient for use on LoWPANs where multicast signaling is generally not used. RFC 6775 [25] provides the specification for this optimization covering the host-initiated interactions leading to host address registration using Neighbor Solicitation (NS) and Neighbor Advertisement (NA). The specification also covers the multihop DAD and distribution of prefix, through Authoritative Border Router Option (ABRO), and 6LoWPAN header compression context through 6LoWPAN Context Options (6COs). Multicast signaling is reduces so as to cater to low power nodes.

Optimized neighbor discovery assumes a LoWPAN to be homogeneous enabling host connecting to different 6LoWPAN Routers (6LRs) without changing their 64-bit unique IPv6 address. To achieve this all 6LRs register with all of the 6LoWPAN Border Routers (6LBRs) in the LoWPAN. In case when 16-bit short addresses are used, the uniqueness can be ensured using DHCPv6 [26]. In a multihop scenario, NS and NA messages can be used only if number of hops is limited to 255. Duplicate Address Request (DAR) and Duplicate Address Confirmation (DAC) messages have been introduced in ICMPv6 [27] to cater towards multihop DAD between 6LRs and 6LBRs with capability to hop more than 255 nodes.

As neighbor discovery works with host-initiated communication model, the host sends Router Solicitation (RS) message at the startup or if Neighbor Unreachability Detection (NUD) of one of its default router fails. In case of usage of 64-bit address by host, the same is registered with its one or more default routers using Address registration Option (ARO) in NS message. Router maintains its Neighbor Cache Entries (NCE) using ARO and this is required for power saving and providing network reliability. This eliminates the requirement of sending multicast NSs for address resolution.

V. COAP

The Constrained Application Protocol (CoAP) [12] is the application layer protocol defined for use on constrained networks and nodes. This protocol has been specifically designed to have easy interoperability with Hypertext Transfer Protocol (HTTP) [28] and keep small message overhead so as to avoid fragmentation. CoAP supports asynchronous message exchanges riding on UDP and provides optional reliability feature. This lightweight reliability mechanism is implemented through simple stop-and-wait retransmission with exponential back-off. The sender retransmit the confirmable message at exponentially increasing intervals until it receives an acknowledgement or it runs out of maximum number of attempts. In addition to stop-and-wait retransmission, this feature also supports the duplicate detection of messages. For every duplicate confirmable message, the recipient must send acknowledgement / reset however, it should only once process the request contained in the message. CoAP also provides security through Datagram Transport Layer Security (DTLS) [29].

CoAP can be considered to have two abstract layers namely Messages and Request / Response interaction. The Messaging layer handles the asynchronous interaction with UDP while the Request / Response layer carry service requests or responses through methods and response codes. The messaging layer exchanges two types of messages over UDP namely Confirmable (CON) with reliability feature and Non-Confirmable (NON) which are unreliable. Hence for every CON message there has to be an acknowledgement (ACK) and if recipient is not able to process the request, it sends a Reset Message (RST). ACK is not compulsory for NON message however, if recipient is unable to process a NON message, it may send a RST.

The Request / Response layer the request is carried in a CON or NON message while the response can be carried in a CON or NON message or even in an ACK message (ACK to a CON). This results in piggybacking of response. When the recipient is unable to respond immediately, it sends an empty ACK and response in separate CON or NON message.

VI. SECURITY CONSIDERATIONS

The security goals [30]-[31] in communication security can be divided broadly into following categories:

- Confidentiality: Confidentiality relates to data privacy such that the unintended user cannot read the data that is transmitted. Confidentiality can be maintained for data at rest and while transit.
- Integrity: Data integrity means that attacker cannot change the data while in transit. Data integrity provides assurance that the data that is received is exactly the data that was transmitted.
- Authentication: Authentication guarantees that the entity that is communicating is the same that it is claiming to be. Authentication comes in play at two main points; one is Peer Entity Authentication where in while an association is made the peer is authenticated to be non-malignant. The second instance of authentication is about the origin of the data, that is, sender of data is as is claimed.
- Non-Repudiation: Non-Repudiation provides a method through which none of the two parties having made the communication can deny of this event of communication. This method maintains the proof that the sender has successfully transmitted data and the receiver has successfully received it.
- Availability: This security goal provides an assurance that the system provides intended services at intended time and is not subject to Denial of Service Attack.

Security in IEEE 802.15.4:IEEE 802.15.4 standard provides security at MAC layer through symmetric cryptography using the Advanced Encryption Standard (AES) [32]. This is provided through various security modes at MAC layer. This security mode is set in the Security Control field in Auxiliary Security Head that is read if Security Enabled Bit is set in Frame Control being send at the beginning of the header. AES-CTR mode provides data confidentiality by encryption of data. To provide data integrity and data authentication AES-CBC-MAC modes can be used. CTR (Counter) and CBC-MAC (Cyber Block Chaining – Message Authentication Code) modes can also be combined to for AES-CCM (Counter with CBC-MAC) mode that provides all the three data confidentiality, data integrity and authentication. The standard also support Access Control List by allowing 802.15.4 radio to store ACL with maximum 255 entries. In addition usage of Absolute Slot Number (ASN) as Frame Counter value allows developers to instill replay protection in the network.

Security in 6LoWPAN: 6LoWPAN specification does not define any security mechanism rather offloads the responsibility of end-to-end security to the higher layers.

Datagram Transport Layer Security (DTLS) [29][33] protocol provides security to UDP. TCP is not suitable for use on constrained nodes in Internet of Things hence UDP is used at transport layer of protocol stack. To provide the End-to-End security to the application layer, in this scenario, DTLS is used as for normal Internet TLS is used over TCP. DTLS is based on the TLS and provides similar features and security assurance. DTLS handles packet loss using simple retransmission timer and optionally supports replay detection. In addition, DTLS uses a stateless cookie exchange for preventing Denial of Service (DoS) attacks.

Security in CoAP: CoAP can be secured using DTLS as HTTP is secured through TLS. This secured CoAP is called CoAPs similar to https. There are four security modes defined in the standard of which NoSec and RawPublicKey modes have been made mandatory for implementation during the manufacturing of CoAP devices. The four security modes are explained briefly as under:

- NoSec: This mode indicates that DTLS is disabled. Hence there is no CoAP protocol level security implemented if this mode is selected.
- PreSharedKey: This indicates that DTLS is enabled and security is achieved through pre-shared keys saved in the device during manufacturing. Each key associates the CoAP device with one or more nodes for communication. The implementation must support TLS_PSK_WITH_AES_128_CCM_8 [34].
- RawPublicKey: DTLS is enabled in this mode and the device has a asymmetric key pair without a certificate. The type of keys and their length is dependent on the cipher suite being used. This key pair is validated using an out-of-band mechanism as specified in RFC 7250 [35]. The implementation must support TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8 [36].
- Certificate: This mode indicates the presence of an X.509 certificate along with a pair of asymmetric keys. DTLS is enabled and the implementation must support TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8.

Researchers worldwide are carrying out research for providing End-to-End security to Internet of Things riding on IEEE 802.15.4 using 6LoWPAN for connecting these constrained embedded systems to Internet. The published research proposals on implementation of cipher suite for IoT have been studied in depth and a comparative table is presented as Table II.

Table II indicates that End-to-End security cipher suite can be provisioned using TLS / DTLS in conjunction with MAC security in IEEE 802.15.4. It can be deduced that the energy consumption and latency in case of security model implementing DTLS is significantly lower than that of models implementing TLS. In addition, the figures of memory footprint show that RSA requires much more memory than PSK and RC4. It emerges from the table that most of the

TABLE II

COMPARISON OF RESULTS OF RESEARCH PAPERS IMPLEMENTING END-TO-END SECURITY

S.No.	REFERENCE	SECURITY MODEL	SOFTWARE	HARDWARE	MEMORY FOOTPRINT		ENERGY CONSUMPTION	LATENCY
					RAM	ROM		
1	BlinkToSCoAP: An End-to-End Security Framework for IoT [37]	Uses DTLS_PSK_WITH_AES_128_CBC_SHA -256 composed of PSK involving 128-bit AES in CBC_CCM mode with implementation of SHA-256	TinyOS	Zolertia Z1 Board	6832 bytes	37360 bytes	774μJ	7.84ms (overall)
2	A DTLS Based End-to-End Security Architecture for the Internet of Things with Two-Way Authentication [38]	Uses TLS_RSA_WITH_AES_128_CBC_SHA1. TPM is used for accelerating RSA	TinyOS	ATMEL SAM3U micro-controller and ATMEL AT97SC3203S TPM	17839 bytes	63379 bytes	579mJ	Single-Hop 2048-bit fully authenticated DTLS handshake = 4000ms
3	SSL-based Lightweight Security of IP-based Wireless Sensor Networks [39]	Uses ECDH_ECDSA based ECC for key exchange and authentication, RC4 for data encryption, light weight SSL supporting MD5 and SHA1. ECC_RC4_MD5_SHA1	Snail Sensor node (running ANTS EOS) and Snail Gateway	TI MSP430 for node and embedded Intel Pentium for gateway	6936 bytes with TCP/IP buffer size as 512 bytes	47642 bytes with TCP/IP buffer size as 512 bytes	Not Carried Out	2000ms for SSL full handshake
4	DTLS & CoAP based Security for Internet of Things Enabled Devices [40]	Uses TLS_PSK_WITH_AES_128_CCM_8 and introduces DTLS header compression scheme	ContikiOS with TinyDTLS	WiSMOTES with MSP430 microcontroller and CC2520 transreceiver	Not Carried Out		DTLS header compression achieves 4-26% energy saving	DTLS header compression achieves 50% shorter Round Trip Time
5	Management of Resource Constrained Devices in the Internet of Things [41]	Comparison of SNMP vs NETCONF using TLS/DTLS implementation with PSK_AES_128_CCM_8	ContikiOS	ATMEL AVR RAVEN and RZUSB Stick	TLS-1861, DTLS-1961 bytes	TLS-37440, DTLS-37232 bytes	Not Carried Out	DTLS handshake-2624ms, TLS handshake-1496ms
6	Proxy-based End-to-end key establishment Protocol for the Internet of Things [42]	Compared Proxy based End-to-End security with basic DH protocol using AES_128_CBC_SHA2	Not Specified	TelosB	Not Carried Out		Proxy based approach – 3.338mJ. Basic approach- 3.137mJ	Not Carried Out

implementations use either TinyOS or ContikiOS. The wise selection of hardware and software is an important for implementation of a cipher suite as the consumption of memory and energy would pose restriction on the intended use of the device and its battery life.

VII. SECURING THE IOT SENSOR LIFECYCLE

The IoT Sensors / constrained nodes / motes have a definitive lifecycle[43]. This lifecycle starts with Design and Development in which various manufacturers select the hardware components and corresponding system software for integration / manufacturing process. These hardware are then integrated through selection and further implementation of various I/O interface technologies / protocols. This is further tested for compatibilities and the resultant prototype is built. A long process of trials, testing, analysis and re-work is carried through the Design and Development stage of IoT sensor Lifecycle.

Manufacturing / Production of the sensor nodes / motes in bulk initiates post completion of Design and Development. At this stage the commercial aspects govern the fit to use methodology as adopted by most of system integrators for project execution and completion. The system integrator selects the products based on use / compatibilities with other components from vast variety of commercially available sensors. The cost factor mostly governs the selection of a product having similar functionality. These sensors from varied origins but compatible with each other are commissioned to for the desired system of operation by the system integrator. The user then tests the system as a whole for acceptance and thereafter the system is made operational. During the use of system regular maintenance in terms of repairs and updates of various software is carried out. This may involve restarts / re-configurations / re- bootstrapping of the sensor nodes.

The end of life of a sensor node as part of system is governed by various reasons as the dependencies of one sensor node is on various other components of the system. Upgrade in technology is one of the most expected reason for end of life for a sensor node as the IoT communication technology is growing at a very fast pace. Other reasons for obsolesce may be increase in incompatibilities injected due to system software / interface software changes in various sensor nodes by their own manufacturers without backward compatibility. Such issues are likely to happen by the manufacturing companies producing low cost sensor nodes having very limited research and development and using core technologies of other companies. So selection of products from manufacturers having limited / no research background are likely to live less and may also cause collapse of the whole operational system.

Having discussed the lifecycle of a sensor node, it becomes evident that there are numerous interactions of product in terms of hardware / software by one manufacturer / designer / developer with another. In this complex infrastructure of sensor nodes it is nearly impossible to have all the products and software from a single source. Hence the need for implementation of security at each step / phase of product lifecycle is mandatory to achieve a wholesome security of complete infrastructure. The security model therefore has to identify and fill the security gaps in system hardware, system software and corresponding implementation of I/O interface protocols. The system integrator must select the compatible products and use software for checking the compatibility and vulnerabilities arising from this integration. The bootstrapping has to be secured with multi-layer security process. There has to be a balance between system security and unit security. Rather, unit security implementation must ensure that the larger system does not breed vulnerabilities due to this unit.

In addition to the introduction of vulnerabilities during initial development of system software / protocol implementation there is always a possibility of generation of bugs in the upgrades / patches of the software if the strict process of coding is not followed. The security / functionality updates of sensor nodes should always be checked and tested in a safe environment before executing them in the live operational system. There is always a possibility of introduction of incompatibilities when one of the components is upgraded without testing the system as a whole. Therefore, the job of system integrator in the case of implementation of IOT Sensor infrastructure is very important and challenging from the point of view of security of each sensor node and system in totality.

VIII. CONCLUSION

This paper outlines the prominent features of IEEE 802.15.4 and 6LoWPAN under the ambit of Internet of Things. The comparison of features of various transreceivers can help reader in their decision making for selection of suitable radio for embedded constrained nodes. This paper discusses security considerations in general and specific to protocol stack for IoT which provides an insight into available security features for implementation. The comparison of various End-to-End security proposals provides the reader with understanding of implications on memory, latency and battery life posed by security models implementation. This paper marks an indicator that only a few security models have been tried and tested on IoT. Hence, there is a scope to explore other cipher techniques for implementation to provide more efficient End-to-End security to Internet of things.

REFERENCES

[1] Kevin Ashton. That ‘Internet of Things’ Thing. RFID JOURNAL, 2009.
[2] D. Evans, “The Internet of Things – How the Next Evolution of the Internet is Changing Everything,” CISCO white paper 2011.
[3] M. Zorzi et al., “From Today’s INTRAnet of things to a Future INTERnet of Things: A Wireless- and Mobilit- Related View”, IEEE Wireless Commun., vol. 17, no. 6, p. 44-51, December 2010.
[4] U.S NIC Conference Report on Disruptive Civil Technologies, April 2008. [Online] Available: http://fas.org/irp/nic/disruptive.pdf, November 2015.
[5] IPv4 Depletion by American Registry for Internet Numbers (ARIN). [Online] Available as on November 2015: https://www.arin.net/resources/request/ipv4_countdown.html.

[6] Walter Colitti et al. Integrating Wireless Sensor Networks with the Web. [Online] Available as on November 2015: http://web.cs.wpi.edu/~rek/IoT/Papers/Colitti_CoAP_paper.pdf
[7] Low – Rate Wireless Personal Area Networks (LR-WPANs). IEEE 802.15.4, September 2011.
[8] N. Kushalnagar et al., "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)," IETF RFC 4919, August 2007.
[9] G. Montenegro et al., "Transmission of IPv6 Packets over IEEE 802.15.4 Networks," IETF RFC 4944, September 2007.
[10] J. Hui, P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4 – Based Networks," IETF RFC 6282, September 2011.
[11] T. Winter et al., "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," IETF RFC 6550, March 2012.
[12] Z. Shelby et al., "The Constrained Application Protocol (CoAP)," IETF RFC 7252, June 2014.
[13] ATMEL AT86RF231. [Online] Available: http://www.atmel.com/devices/AT86RF231.aspx?tab=parameters.
[14] ATMEL AT86RF232. [Online] Available: http://www.atmel.com/devices/AT86RF232.aspx?tab=parameters.
[15] ATMEL AT86RF212B. [Online] Available: http://www.atmel.com/devices/AT86RF212B.aspx?tab=parameters.
[16] ATMEL AT86RF233. [Online] Available: http://www.atmel.com/devices/AT86RF233.aspx?tab=parameters.
[17] TEXAS INSTRUMENTS CC2420. [Online] Available: http://www.ti.com/product/cc2420.
[18] TEXAS INSTRUMENTS CC2500. [Online] Available: http://www.ti.com/product/cc2500.
[19] FREESCALE SEMICONDUCTOR MC13202. [Online] Available: http://www.freescale.com/files/rf_if/doc/data_sheet/MC13202.pdf.
[20] NORDIC SEMICONDUCTORS nRF24L01. [Online] Available: http://www.nordeicsemi.com/eng/Products/2.4GHz-RF/nRF24L01.
[21] E. Kim et al., "Design and Application Spaces for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN)," IETF RFC 6568, April 2012.
[22] E. Kim et al., "Problem Statement and Requirements for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) Routing," IETF RFC 6606, May 2012.
[23] Asoke K. Talukder et al., "Convergence Through All-IP Network". Chapter 8: Internet of Things. CRC Press. 2013.
[24] T. Narten et al., "Neighbor Discovery for IP Version 6 (IPv6)," IETF RFC 4861, September 2007.
[25] Z. Shelby et al., "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (LoWPANs)," IETF RFC 6775, November 2012.
[26] R. Droms et al., "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) ," IETF RFC 3315, July 2003.
[27] A. Conta et al., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification," IETF RFC 4443, March 2006.
[28] R. Fielding et al., "Hypertext Transfer Protocol – HTTP/1.1," IETF RFC 2616, June 1999.
[29] E. Rescorla, N Modadugu, "Datagram Transport Layer Security," IETF RFC 4347, April 2006.
[30] Zach Shelby, Carsten Bormann 6LoWPAN: The Wireless Embedded Internet. Chapter 3. pp 83-90. Wiley Publication. Novemeber 2009.
[31] E. Rescorla, B. Korver, "Guidelines for Writing RFC Text on Security Considerations," IETF RFC 3552, July 2003.
[32] F. Miller et al., Advanced Encryption Standard, 2009.
[33] E. Rescorla, N Modadugu, "Datagram Transport Layer Security Version 1.2," IETF RFC 6347, January 2012.
[34] D. McGrew, D. Bailey, "AES-CCM Cipher Suites for Transport Layer Security (TLS)," IETF RFC 6655, July 2012.
[35] P. Wouters et al., "Using Raw Public Keys in TLS/DTLS," IETF RFC 7250, June 2014.
[36] D McGrew et al., "AES-CCM Elliptic Curve Cryptography (ECC) Cipher Suites for TLS," IETF RFC 7251, June 2014.
[37] Peretti, G. et al., "BlinkToSCoAP: An end-to-end security framework for the Internet of Things," in *Communication Systems and Networks (COMSNETS), 2015 7th International Conference on* , vol., no., pp.1-6, 6-10 Jan. 2015.
[38] T. Kothmayr et al., "A DTLS based end-to-end security architecture for the Internet of Things with two-way authentication," in *Local Computer Networks Workshops (LCN Workshops), 2012 IEEE 37th Conference on*, vol., no., pp.956-963, 22-25 Oct. 2012.
[39] Jung Wooyoung et al., "SSL-Based Lightweight Security of IP-Based Wireless Sensor Networks," in *Advanced Information Networking and Applications Workshops, 2009. WAINA '09. International Conference on*, vol., no., pp.1112-1117, 26-29 May 2009.
[40] D. UthayaSinthan, M.S. Balamurugan, "DTLS & COAP Based Security For Internet of Things Enabled Devices," [Online] Available: http://www.ijesrt.com/issues%20pdf%20file/Archives%202013/dec-2013/86.pdf.
[41] A. Sehgal et al., "Management of resource constrained devices in the internet of things," in *Communications Magazine, IEEE* , vol.50, no.12, pp.144-149, December 2012.
[42] P. Porambage et al., "Proxy-based end-to-end key establishment protocol for the Internet of Things," in *Communication Workshop (ICCW), 2015 IEEE International Conference on*, vol., no., pp.2677-2682, 8-12 June 2015.
[43] O. Garcia-Morchon et al., "Internet of Things(IoT) Security: State of the Art and Challenges," IRTF RFC 8576, April 2019.