International Journal on Future Revolution in Computer Science & Communication Engineering
Volume: 3 Issue: 11
ISSN: 2454-4248
355 – 359

IoT Protocols : Z-Wave and Thread

Ishaq Unwala, *Assistant Professor*
Jiang Lu, *Assistant Professor*
Computer Engineering Dept.
University of Houston Clear Lake
Houston, TX, USA
Unwala@uhcl.edu, LuJ@uhcl.edu

***Abstract*—** Today there are a multitude of IoT protocols available in the marketplace. Two protocols from different time periods are selected to be compared. The two protocols selected are: *Z-Wave* which is one of the oldest and the most commercially successful protocol, and *Thread* which is the latest protocol released for commercialization. This paper discusses both the protocols PAN, PHY, MAC, Routing, encryption, etc. *Z-Wave* is based on propriety standards, most of which is not publicly available, although some have been reverse engineered by researchers. *Thread* on other hand is based completely on open standards. All *Z-Wave* modules are made by a single company, while *Thread* modules are expected to be available from multiple vendors. *Z-Wave* has a large installed base and has proven to be a commercial success. *Thread* is new and has a open protocol, but *Thread* based devices are not yet readily available in the market for users.

Keywords*-*Z-Wave, Thread, Threadgroup,IoT,Internet-of-things

*****

I. INTRODUCTION

Internet-of-Things (IoT) as a field has been growing ever since Kevin Ashton introduced the term in 1999 [1]. According to Gartner's recent press release [2], by the end of 2017, total IoT installed devices is expected to reach 8.4 billion units and the IoT related spending in 2017 is expected to be $1.69 trillion. Gartner, in the same press release, also projects that by 2020 the IoT installed devices will reach 20.4 billion units and spending will reach $2.9 trillion. It is expected that IoT will play an important part in everyday life. The IoT systems will be involved in smart cities, smart workplaces, smart industries, smart cars and smart homes. This paper discusses two very different IoT protocols, *Z-Wave* and *Thread*. *Z-Wave* is one of the first IoT protocols to be widely commercialized and *Thread* is the latest IoT protocol to be released. It is important to understand the differences and similarities between the two protocols.

II. INTERNET OF THINGS

What is an IoT system? An IoT system is a concept of connecting sensors and actuators into a coherent network with Cloud computing services that can provide services to the user. An IoT system is depicted in Fig. 1. In an IoT system, the IoT devices can be localized in a home or spread across the city. The IoT devices are connected in a Private Area Network (PAN) using wireless communication protocols. The PAN is then connected through the router(s) to Internet allowing it to accessing the Cloud computing services. The data processed in the Cloud can result in necessity for physical action. The action may be as simple as sending a text message or as complex as activating a set of actuators to accomplish some task. The action is communicated from the Cloud computer back to the IoT actuator device(s) through the router and the IoT PAN.

An IoT system, Fig. 1, has following features [5]:

- Device Identification. An individual device in an IoT PAN has a unique address.
- Sensing. The IoT PAN has a number of sensing devices that collect data from a monitored event or environment. Examples of sensors include detectors for motion, light, vibration, pressure, temperature, acceleration, magnetic field, infrared, proximity, distance, biometric parameters and many others.
- Response. The IoT PAN may also have a number of devices that respond, i.e. perform actions. Examples include relays, actuators and switches.
- Communication. The IoT devices in the PAN communicate wirelessly with each other. The sensor data is communicated, by the router, to a central location, usually using Cloud computing, for processing and evaluation. Results of data evaluation are then communicated back to the IoT PAN for any required physical action.
- Computation. Cloud computing services are usually used to process the IoT sensor data. Cloud computing formulates a response and sends to the IoT router, if needed. The IoT router then uses the PAN wireless communication links, to deliver the message.
- Services. The IoT PAN is ubiquitous and provides a number of services: item identification, collecting and communicating sensor data, event monitoring, performing physical action(s), and device collaboration.
- Semantics. Cloud processing software analyzes the data and provides a context-aware response for the IoT system.

All these capabilities in the IoT PAN are provided under practical constraints. Here are few of the major constraints on the IoT PAN:

- Low power. Many of IoT devices are battery powered; replacing batteries is expensive, inconvenient, and unreliable. Even if the IoT devices are attached to the power grid, low power is required for a couple of reasons. First, the IoT devices are continuously operating and therefore even small amount of power consumption can quickly add up. Second, a typical IoT installation consists of hundreds of devices. The aggregate power consumption of these devices operating continuously can be quite significant.
- Low cost. Cost is always a major factor in the consumer adoption of any product. Low cost promotes a wider use of the smart technologies.

 ISSN: 2454-4248
355 – 359

- Security. Beside physical security of the IoT devices, authenticity, privacy, confidentiality, and integrity of data are important consideration in an IoT system.
- Communication. The IoT faces communication constraints on multiple fronts. The IoT PAN radio operates on low power, short range transmission, in a restricted public frequency spectrum. The operating environment is noisy with other equipment generating radio interference, and there are many physical obstacles. The IoT communication also faces the challenge of frequent network route changes as IoT devices are moved, some of the IoT devices are mobile, plus the users may move obstacles in its communication paths. Additionally, the IoT data packets are small, making it harder to include redundancy for fault tolerance.
- User interface. Most IoT devices either have a primitive or no user interface. The lack of graphical user interface (GUI) constraints the communication between the user and the device. GUI-based components, like smart-phone, smart-tablet, a computer or even a voice controlled device is normally required to set up and communicate with the IoT device.

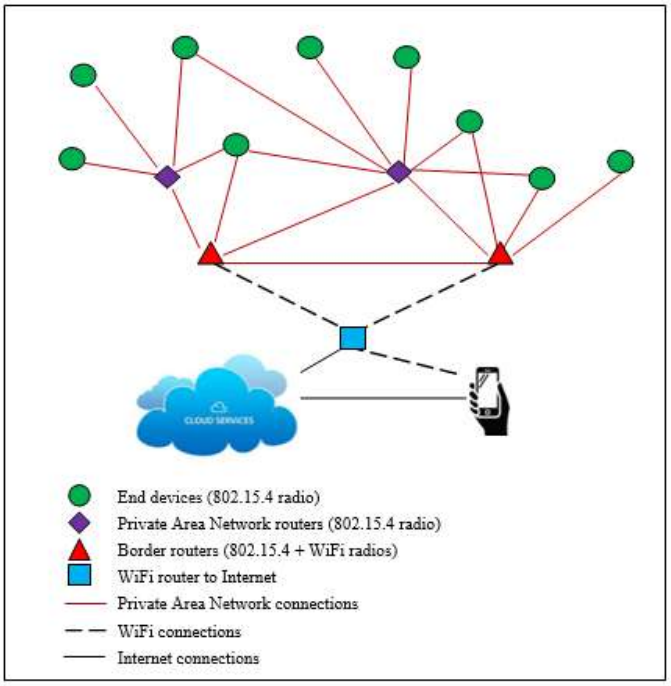

Figure 1. IoT system

III. IoT Private Area Network

There are a number of IoT protocols currently available, *Z-Wave*, *Zigbee*, *Thread*, *IoTivity*, *AllJoyn*, and among others. Although particulars for each of these protocols are different, they share many similarities. All these protocols establish a wireless Private Area Network (PAN). PAN network topology is built using connectivity through wireless radios. Topologies commonly used for PAN are point-to-point, star and mesh. The mesh topology is the most popular due to its robustness, scalability and fault resilience. A typical PAN with mesh topology can be seen in Fig. 1. Generally, an IoT network have three types of nodes: End devices, PAN routers, Border routers. Data from IoT devices can be transmitted across multiple hops, as determined by a routing protocol, before arriving at gateway Border router(s). The End devices normally consist of sensors or actuators. The PAN routers serve multiple purposes, besides being sensors or actuators, PAN routers also act as communication router for the End devices. The PAN routers have additional capability to connect with other PAN routers and/or Border routers. The Border router(s) have an additional capability to act as gateway to Internet/Cloud by connecting to Internet router. The End devices can communicate with either the PAN network router or directly with Border routers. An example of a Border router is Nest thermostat, which has connectivity to both PAN and WiFi router, plus it is also an IoT sensor and actuator.

IV. Z-Wave protocol

Z-Wave (originally *ZenSys*) is a proprietary IoT protocol, owned by Sigma Designs, so only limited information is publicly available. Sigma Designs is the only supplier of *Z-Wave* hardware, this was done to maintain interoperability between different IoT device producers. *Z-Wave* has been available for more than 10 years. A large variety of *Z-Wave* devices are available in the market with millions of devices installed.

Z-Wave wireless network uses a proprietary wireless standard. *Z-Wave* uses a wireless MESH network topology. *Z-Wave* operating frequency is 908.42 MHz (USA) or 868.42 MHz (Europe), with a data rate of 9.6 Kbps, and uses FSK (Freq. Shift Keying) modulation. *Z-Wave* has a maximum range of 30m per hop, with maximum of 4 hops. Maximum number of devices in one *Z-Wave* PAN is limited to 232. *Z-Wave* network is connected to the Internet/Cloud through a Smart Hub (Z/IP). Smart Hub, is the Border router for *Z-Wave* protocol providing connection between PAN and Internet gateway. Smart Hub is a single point of failure for the *Z-Wave* network as all data traffic in and out of PAN passes through it. Each *Z-Wave* Smart Hub has a unique 32-bit Home-ID. For security purposes, the Home-ID is permanently embedded in Smart Hub during manufacturing and cannot be reprogrammed. All IoT devices in *Z-Wave* PAN receive a common Home-ID and a unique 8-bit Node-ID after the device has been authenticated. Multiple *Z-Wave* PANs in close proximity is not an issue, since each IoT device has a unique Home-ID and a Node-ID tag [8][9][10][23]. *Z-Wave* Smart Hub to Internet connection is encrypted with AES-based pre-shared key (TLS 1.1, PSK) [12]. Two *Z-Wave* PANs can also be connected through two Smart Hubs and DTLS [12]. *Z-Wave* accomplishes secure key exchange using Elliptic Curve Diffie-Hellman (ECDH) [12][18].

V. Thread protocol

Thread was announced by Threadgroup Inc. in 2015, so it is a relatively new IoT protocol [7]. Hardware boards with wireless radios, plus software libraries for *Thread* are now available for research and development. Consumer devices are not yet widely available, although Nest thermostat and some *Zigbee* devices are *Thread* compatible. *Thread* is an open standard. *Thread* specification is available from the Threadgroup. *Thread* was founded by seven companies (ARM (acquired by SoftBank Group), Big Ass Fans, Freescale (acquired by NXP Semiconductors), Nest (acquired by Alphabet/Google), Samsung, Silicon Labs, and Yale). Today, Threadgroup has more than 200 member companies. *Thread* intends to consolidate IoT protocols by working with other IoT

 ISSN: 2454-4248
355 – 359

alliances. *Thread* is specially designed for Home Automation and supports a wide variety of home use applications: appliances, access control, climate control, energy management, lighting, safety, and security. This paper discusses the latest *Thread* specification v1.1.1 2017.

Thread is based on open standards such as IEEE 802.15.4 (2450 MHz), IPv6, 6LoWPAN, wireless PAN MESH network topology with no single point of failure. The wireless PAN operates at 2.4 GHz with a data rate up to 250 Kbps and using O-QPSK modulation. *Thread* has a maximum range of 30m per hop, with a default hop limit of 36 hops. *Thread* has the ability to connect 250+ devices in a single PAN. *Thread* can have multiple Border routers, which connect to Internet, this eliminates Border router failure as a single point of failure. Each IoT device in *Thread* has an IPv6 address. The IPv6 address is compressed using 6LoWPAN, which is a standard for low power devices. Readers might notice similarities between *Thread* and the *Zigbee* protocol. As a matter of fact, *Zigbee* libraries can run as an application on *Thread* [24].

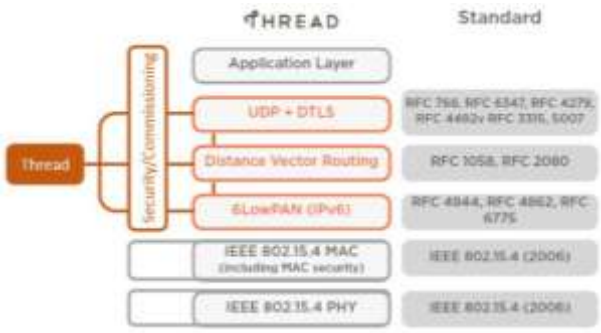

Figure 2. *Thread* protocol stack IoT system [7]

VI. NETWORK STACK

The IoT protocol layers are shown in Fig. 3 and Fig. 4. The main security for IoT protocols are in Perception Layer, Data Link Layer and Transport Layer. Although data in Cloud is an important part of the IoT system, Cloud is not part of IoT protocols and is addressed separately in [28].

A. Perception Layer Security

Perception layer includes WSN (Wireless Sensor Network), IMD (Implantable Medical Devices), RFID (Radio Frequency Identification), NFC (Near Field Communication), temperature sensors, pressure sensors, luminance sensors, vibration sensors, accelerometer magnetometer, gyroscope, etc.

Z-Wave and *Thread* protocol have devices with similar functionality at perception layer, but are incompatible with each other. The communication standard for each protocol is very different.

Perception layer requires physical security of devices. Physically manipulated faulty nodes may have unauthorized attachments, or reprogrammed boot to change their behavior. This can be secured by running fault detection algorithms [14][15][16]. Encryption related data must protected by using encryption before storing it in memory. To add new devices in PAN, strong passwords, biometric recognition (fingerprints, iris, voice, face, etc) or a synchronized key generator must be used. Currently, none of the major IoT protocols address the issue of protecting data stored in the device registers and memory.

B. Physical and Data Link Layer (PHY and MAC) Security

In *Z-Wave* PHY and MAC layers are proprietary. In *Thread* PHY and MAC layers are defined by IEEE 802.15.4. Although IEEE 802.15.4 allows other frequencies, *Thread* only uses the 2450 MHz related portion of the standard. This is the lowest layer for which there are protocol-based security specifications. Support for this encryption is built into the radio hardware.

Almost all IoT protocols offer AES (Advanced Encryption Standard) 128-bit [13] [17]. However, AES does not scale well with hundreds of devices in the network and requires very large keys. Therefore, IoT protocols implement, ECC (Elliptic Curve Cryptography, RFC 4492) [18]. ECC is an asymmetric, public-key method that scales well and provides a higher level of security for the same number of bits. Other low power public key encryption algorithm Rabin's Scheme and NtruEncrypt are also good candidates for IoT [19].

Z-Wave security has an interesting history when it comes to data encryption. Initial *Z-Wave*, series 100, had TDES (Triple Data Encryption Standard) [11]. Later in series 200 silicon, encryption was dropped according to Knight [4]. Encryption was again added back and now it uses AES (Advanced Encryption Standard) [12][13]. Since then *Z-Wave* has taken steps to improve security features and recently has been evaluated by UL for security applications according to press release by Sigma Design, "*Z-Wave* modules models ZM5101, ZM5202, and ZM5304 with protocol SDK version 6.60 have been evaluated to UL's standards for home security" [3]. This paper considers *Z-Wave* protocol, with AES-128, as it exists today. *Z-Wave* Smart Hub to Internet connection is encrypted with AES-based pre-shared key (TLS 1.1, PSK) [12]. *Z-Wave* accomplishes secure key exchange using Elliptic Curve Diffie-Hellman (ECDH) [12][18].

Thread uses AES-128 and ECC for data link encryption [25]. AES does not scale well with hundreds of devices, as expected in the *Thread* network and requires very large keys. Therefore, *Thread* [25] uses ECC (Elliptic Curve Cryptography, RFC 4492) [12] [18]. ECC is an asymmetric, public-key method that scales well and provides a higher level of security for the same number of bits.

Network Layer Comparison

	Z-Wave	Thread	
Application	Device & Cmd	Application	
Transport	Routing Layer	UDP	RIPng
Network	MESH	MESH,IPv6 – 6LoWPAN	
Data Link	Proprietary MAC	IEEE 802.15.4 MAC	
Physical	Proprietary PHY	IEEE 802.15.4 PHY	
Perception	Sensors/Switches	Sensors/Switches	

Figure 3. Network Layer Comparison

C. Transport Layer Security

Transport layer mainly consists of routing protocol and data transport protocol.

 ISSN: 2454-4248
355 – 359

Z-Wave routing protocol is propriety, however researcher have reverse engineered most of it [6]. *Z-Wave* routing has 4 possible hops plus one final hop to the destination node.

Thread uses standard routing RIPng [RFC 2080]. RIPng is a distance vector routing protocol [RFC 1058] for IPv6. RIPng has maximum of 15 hops.

There are two main data protocols currently being used at transport layer in IoT PANs, CoAP (Constrained Application Protocol) [20] [27] and MQTT (Message Queuing Telemetry Transport) [21]. Both of these protocols were specifically designed for low power IoT type devices.

Actual transport protocol used by CoAP is a UDP (User Datagram Protocol), which enforces use of DTLS (Datagram Transport Layer Security, IETF RFC 6347). CoAP has a set of security modes and mandatory-to-implement ciphers. CoAP borrows some concepts from REST (Representational State Transfer) protocol [22].

The transport protocol used by MQTT is TCP, which enforces TLS (Transport Layer Security, IETF RFC 5246). MQTT is a publisher/subscriber style protocol, requiring a server-broker. A typical MQTT session consists of establishing a connection, authentication, communication, termination. MQTT provides authentication and authorization scheme, but does not have any security implementation requirements. MQTT is designed to operate in a secure network, and thus has no defined security mechanism. MQTT sends username and password in clear text and thus relies on Transport layer encryption, like TLS, to provide security. MQTT should not be used for global network as it does not scale well. TLS is expensive protocol and not well suited for very low power devices [29] [30]. MQTT has TCP port 1883 reserved for non-encrypted and TCP port 8883 reserved for encrypted communication using TLS.

Although the DTLS and TLS have similar concepts, CoAP with DTLS is preferred in IoT protocol due to low power IoT devices having limited memory and processing capabilities. Both CoAP and MQTT can be operated in "no security" mode, "pre-shared key" mode and "certificate" mode. CoAPs (CoAP secured) can also be configured in "raw public key" [IETF RFC 7250], but MQTT implementation is not yet available. In pre-shared key mode CoAPs hashes pre-shared keys with a list of corresponding communication nodes. Although use of certificate mode is well established, its use in IoT is discouraged due to resource constraints. However, there is a one big advantage in use of certificates; the certificate can be revoked if the IoT device is compromised. In raw public key mode the IoT device holds an asymmetric key pair but without a certificate. Normally, the asymmetric key pair is generated and embedded in the device by the manufacturer. This asymmetric key pair needs to be validated in out-of-band (OOB) mechanism using public key. A device can have multiple raw public keys.

Z-Wave and *Thread* both utilize DTLS encryption for the transport layer. If the radio hardware does not support data encryption than Transport layer encryption must be used. Using "no security" mode is acceptable if the network is running on a VPN, or if Data layer encryption is being used. Encrypting the data twice, at Data layer and Transport layer can decrease the data compression ratio, which leads to more package transmissions and higher power consumption.

Thread Mesh is self-configuring and each link is individually encrypted. Every device in *Thread* holds the credentials which allow it to be part of the network. However, before the device can receive the credentials it has to be authenticated by the Commissioner (an authentication server). Commissioner itself must be authenticated. Commissioner authentication requires a onetime Commissioning Session, a secured client/server socket connection, between Commissioner and the Border Router via DTLS (RFC 6347) or TLS (RFC 5246). Using the advertized UDP port, during discovery, Commissioner provides PSKc (Pre-Shared Key for Commissioner) credential. The Border router with human interaction than authenticates the Commissioner.

A new device wishing to join the *Thread* network must transmit an unsecured Discovery Request message. A router responds with a Discovery Response message including the joining UDP port. The device will perform DTLS handshake to establish a secure session with router. The router will relay UDP messages to the Border router, which in turn will relay them to the Commissioner. The device and Commissioner then exchange token to establish trust. Commissioner inspects the device IID (Interface Identifier) and credentials. If the Commissioner is satisfied with the responses from the device, it will be provisioned with the appropriate data and services, and also provided with KEK (Key Encryption Key). Once it is authenticated by the Commissioner, router will provide the device network credentials.

Each *Thread* node also receives a master key when joining. Two different 16-bit keys, one for MAC and other for DTLS, are generated using Hashed Message Authentication Mode with SHA-256 algorithm (HMAC-SHA256) produces 32-bit output [RFC 6234] and the master key. The key set are rotated based on key index changes, or the key rotation timer expiry, or incoming messages matches the next key.

Network Security Comparison

	Z-Wave	Thread
Application	Device & Cmd	Application
Transport	DTLS 1.0, PSK	DTLS 1.2, PSK
Network	Home Network ID	IPv6-6LoWPAN
Data Link	AES-128	AES-128, ECC
Physical	None or facility door	None or facility door
Perception	Password	Password

Figure 4. Network Security Comparison

VII. CONCLUSION

Both the protocols. *Z-Wave* and *Thread*, fulfill the requirements, features and capabilities of an IoT protocol. Both protocols have a reliable MESH configuration for PAN. *Z-Wave* has tried to protect the PAN from intruders by hiding the information in the propriety standards. These same propriety standards also make it hard to judge the quality of *Z-Wave* standards, but researchers have reverse engineered some aspects of the PAN [26][6]. *Thread* is based on open standards, which allows everyone to study and detect any security flaws. Fig. 3 shows the network layer comparison, which clearly highlights the differences in PHY, MAC, Routing between *Z-Wave* and *Thread*. Network security comparison is shown in Fig. 4, where the differences are less discernible. The reason

for similarity in security is that *Z-Wave* seems to have been upgrading its devices. *Z-Wave* has been commercially successful with its Z-Alliance. *Thread*, with its Thread Group Inc., still has to prove itself in the marketplace.

REFERENCES

[1] Aston, Kevin. "That 'Internet of Things' Thing", Available: http://www.rfidjournal.com/articles/view?4986, accessed on June 2, 2017.

[2] Gartner, Inc. "Gartner Says 8.4 Billion Connected 'Things' Will Be in Use in 2017, Up 31 Percent From 2016", http://www.gartner.com/newsroom/id/3598917, accessed on June 2, 2017.

[3] Sigma Design press release, "Sigma Designs Announces Break-Through Z-Wave(R) UL Component Recognition", http://www.sigmadesigns.com/news/sigma-designs-announces-break-through-z-waver-ul-component-recognition , accessed on June 2, 2017.

[4] M. Knight, "Wireless security - How safe is Z-wave?," in Computing & Control Engineering Journal, vol. 17, no. 6, pp. 18-23, Dec.-Jan. 2006. doi: 10.1049/cce:20060601

[5] M. Al-Zyoud, T. Atkison and J. Carver, "An Overview of Emerging Privacy Issues in the Internet of Things," 2016 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, 2016, pp. 212-217. doi: 10.1109/CSCI.2016.0047

[6] C.W.Badenhop, S.R.Graham, B.W.Ramsey, B.E.Mullins, L.O.Mailloux, "The Z-Wave routing protocol and its security implications" https://doi.org/10.1016/j.cose.2017.04.004

[7] Threadgroup Inc. website: http://Threadgroup.org/What-is-Thread/Connected-Home

[8] H. Sharma and S. Sharma, "A review of sensor networks: Technologies and applications," 2014 Recent Advances in Engineering and Computational Sciences (RAECS), Chandigarh, 2014, pp. 1-4. doi: 10.1109/RAECS.2014.6799579

[9] M. B. Yassein, W. Mardini and A. Khalil, "Smart homes automation using Z-wave protocol," 2016 International Conference on Engineering & MIS (ICEMIS), Agadir, 2016, pp. 1-6. doi: 10.1109/ICEMIS.2016.7745306

[10] Galeev, M.T., S. Engineer, and M. Inc, "Catching the Z-Wave". Embedded Systems Design, 2006. 19(10): p. 28.

[11] M. Zareei, A. Zarei, R. Budiarto and M. A. Omar, "A comparative study of short range wireless sensor network on high density networks," The 17th Asia Pacific Conference on Communications, Sabah, 2011, pp. 247-252. doi: 10.1109/APCC.2011.6152813

[12] "Introduction to the Z-Wave Security Ecosystem", available from Sigma Design, Inc., on the company website: http://z-wave.sigmadesigns.com/wp-content/uploads/2016/08/Z-Wave-Security-White-Paper.pdf

[13] J. Daemen and V. Rijmen, AES Proposal: Rijndael, version 2, 1999. Available from URL: http://www.esat. kuleuven.ac.be/~rijmen/rijndael

[14] J. Chen, S. Kher, and A. Somani, "Distributed fault detection of wireless sensor networks," in Proceedings of the 2006 Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks, ser. DIWANS '06, 2006, pp. 65–72

[15] C. Lo, J. P. Lynch and M. Liu, "Distributed Reference-Free Fault Detection Method for Autonomous Wireless Sensor Networks," in IEEE Sensors Journal, vol. 13, no. 5, pp. 2009-2019, May 2013. doi: 10.1109/JSEN.2013.2244881

[16] I. Chatzigiannakis and A. Strikos, "A decentralized intrusion detection system for increasing security of wireless sensor networks," 2007 IEEE Conference on Emerging Technologies and Factory Automation (EFTA 2007), Patras, 2007, pp. 1408-1411. doi: 10.1109/EFTA.2007.4416949

[17] Federal Information Processing Standards , "Announcing the ADVANCED ENCRYPTION STANDARD(AES)", Federal Information Processing Standards Publication 197 November 2001 [online] Available: http://www.csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[18] Z. Liu, X. Huang, Z. Hu, M. K. Khan, H. Seo and L. Zhou, "On Emerging Family of Elliptic Curves to Secure Internet of Things: ECC Comes of Age," in IEEE Transactions on Dependable and Secure Computing, vol. 14, no. 3, pp. 237-248, May-June 1 2017. doi: 10.1109/TDSC.2016.2577022

[19] G. Gaubatz, J. P. Kaps, E. Ozturk and B. Sunar, "State of the art in ultra-low power public key cryptography for wireless sensor networks," Third IEEE International Conference on Pervasive Computing and Communications Workshops, Kauai Island, HI, 2005, pp. 146-150. doi: 10.1109/PERCOMW.2005.76

[20] C. Bormann, A. P. Castellani and Z. Shelby, "CoAP: An Application Protocol for Billions of Tiny Internet Nodes," in IEEE Internet Computing, vol. 16, no. 2, pp. 62-67, March-April 2012. doi: 10.1109/MIC.2012.29

[21] "Oasis Message Queue Telemetry Transport", OASIS MQTT version 3.1.1, 2014

[22] S. Zamfir, T. Balan, I. Iliescu and F. Sandu, "A security analysis on standard IoT protocols," 2016 International Conference on Applied and Theoretical Electricity (ICATE), Craiova, 2016, pp. 1-6. doi: 10.1109/ICATE.2016.7754665

[23] M. B. Yassein, W. Mardini and A. Khalil, "Smart homes automation using Z-wave protocol," 2016 International Conference on Engineering & MIS (ICEMIS), Agadir, 2016, pp. 1-6. doi: 10.1109/ICEMIS.2016.7745306

[24] "The Thread Group Expands Influence through Partnership with ZigBee Alliance, TCLA and Innovation Enabler Award", (Accessed June 7, 2017) http://mysmahome.com/news/5905/the-Thread-group-expands-influence-through-partnership-with-zigbee-alliance-tcla-and-innovation-enabler-award-2/

[25] S. Rajesh, "Ubiquitous energy efficient aquaculture management system," 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Jaipur, 2016, pp. 1124-1128. doi: 10.1109/ICACCI.2016.7732195

[26] HackersOnBoard. "Black Hat 2013 – Honey, I'm Home!! – Hacking Z-Wave Home Automation Systems." Online video clip. YouTube. YouTube, 19 Nov 2013. Web 1 Nov. 2016. URL: https://www.youtube.com/watch?v=KYaEQhvodc8

[27] M. B. Tamboli and D. Dambawade, "Secure and efficient CoAP based authentication and access control for Internet of Things (IoT)," 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, 2016, pp. 1245-1250. doi: 10.1109/RTEICT.2016.7808031

[28] J. Singh, T. Pasquier, J. Bacon, H. Ko and D. Eyers, "Twenty Security Considerations for Cloud-Supported Internet of Things," in IEEE Internet of Things Journal, vol. 3, no. 3, pp. 269-284, June 2016. doi: 10.1109/JIOT.2015.2460333

[29] L. Nastase, "Security in the Internet of Things: A Survey on Application Layer Protocols," 2017 21st International Conference on Control Systems and Computer Science (CSCS), Bucharest, Romania, 2017, pp. 659-666. doi: 10.1109/CSCS.2017.101

[30] M. B. Yassein, M. Q. Shatnawi and D. Al-zoubi, "Application layer protocols for the Internet of Things: A survey," 2016 International Conference on Engineering & MIS (ICEMIS), Agadir, 2016, pp. 1-4. doi: 10.1109/ICEMIS.2016.7745303