Available online at www.worldscientificnews.com

World Scientific News

An International Scientific Journal

WSN 41 (2016) 27-33 EISSN 2392-2192

An Approach to Secure Communication in IoT (Internet of Things)

G. F. Ali Ahammed[1], Reshma Banu[2], Nasreen Fathima[3]

[1]Department of CSE, VTU Post Graduate Centre, Mysore, India

[2]Department of ISE, GSSSIETW, Mysore, India

[3]Department of CSE, ATME College of Engineering, Mysore, India

[1-3]E-mail address: aliahammed78@gmail.com , reshma127banu@gmail.com , nasreenfathima16@gmail.com

ABSTRACT

The term Internet of Things (IoT) refers to the use of standard internet protocols for interaction between human to things or things to things in an embedded network. Although the security needs are well-recognized, it is still not fully clear how existing IP-based security protocols can be applied to this new setting. In this paper we discuss the various security challenges in an IoT system. The paper also provides a standard IP-based security protocols and its implementation model which can be used as a security system for IoT.

***Keywords*:** Bootstrapping, 6LoWPAN, IETF, threats, IoT

1. INTRODUCTION

The Internet of Things (IoT) denotes the interconnection of highly heterogeneous networked entities and networks following a number of communication patterns such as: human-to-human (H2H), human-to-thing (H2T), thing-to-thing (T2T), or thing-to-things (T2Ts). The term IoT was first coined by the Auto-ID [4] center in 1999. Since then, the

(Received 01 February 2016; Accepted 10 February 2016; Date of Publication 11 February 2016)

development of the underlying concepts has ever increased its pace. Nowadays, the IoT presents a strong focus of research with various initiatives working on the (re)design, application, and usage of standard Internet technology in the IoT. The introduction of IPv6 and web services as fundamental building blocks for IoT applications promises to bring a number of basic advantages including [3]:

- A homogeneous protocol ecosystem that allows simple integration with Internet hosts
- A unified interface for applications, removing the need for application-level proxies. Such features greatly simplify the deployment of the envisioned scenarios such as building automation in production environments.

1. 1. Threat Analysis

Security threats have been analyzed in related IP protocols including HTTPS, 6LoWPAN, ANCP, DNS security threats, SIP, IPv6, ND, and PANA. Nonetheless, the challenge is about their impacts on scenarios of the IoTs. In this section, we specifically discuss the threats that could compromise an individual thing, or network as a whole, with regard to different phases in the thing's lifecycle:

1. Cloning of things: During the manufacturing process of a thing, an untrusted manufacturer can easily clone the physical characteristics, firmware/software, or security configuration of the thing. Subsequently, such a cloned thing may be sold at a cheaper price in the market, and yet be still able to function normally, as a genuine thing. In the worst case scenario, a cloned device can be used to control a genuine device. One should note here, that an untrusted manufacturer may also change functionality of the cloned thing, resulting in degraded functionality with respect to the genuine thing. Moreover, it can implement additional functionality with the cloned thing, such as a backdoor.

2. Malicious substitution of things: During the installation of a thing, a genuine thing may be substituted with a similar variant of lower quality without being detected. The main motivation may be cost savings, where the installation of lower-quality things (e.g., non-certified products) may significantly reduce the installation and operational costs.

3. Eavesdropping attack: During the commissioning of a thing into a network, it may be susceptible to eavesdropping, especially if operational keying materials, security parameters, or configuration settings, are exchanged in clear using a wireless medium. After obtaining the keying material, the attacker might be able to recover the secret keys established between the communicating entities (e.g., H2T, T2Ts, or Thing to the backend management system), thereby compromising the authenticity and confidentiality of the communication channel.

4. Man-in-the-middle attack: The commissioning phase may also be vulnerable to man-in-the-middle attacks, e.g., when keying material between communicating entities is exchanged. The security of the key establishment protocol depends on the tacit assumption that no third party is able to eavesdrop on or sit in between the two communicating entities during the execution of this protocol. Additionally, device authentication or device authorization may be nontrivial, or may need support of a human decision process, since things usually do not have

a priori knowledge about each other and can, therefore, not always be able to differentiate friends and foes via completely automated mechanisms.

5. Firmware Replacement attack: When a thing is in operation or maintenance phase, its firmware or software may be updated to allow for new functionality or new features. An attacker may be able to exploit such a firmware upgrade by replacing the thing's with malicious software, thereby influencing the operational behavior of the thing.

6. Extraction of security parameters: A thing deployed in the ambient environment (such as sensors, actuators, etc.) is usually physically unprotected and could easily be captured by an attacker. Such an attacker may then attempt to extract security information such as keys (e.g., device's key, private-key, group key) from this thing or try and re-program it to serve his needs. If a group key is used and compromised this way, the whole network may be compromised as well.

7. Routing attack: Routing information in IoT can be spoofed, altered, or replayed, in order to create routing loops, attacks etc.

8. Privacy threat: The tracking of a thing's location and usage may pose a privacy risk to its users. An attacker can infer information based on the information gathered about individual things, thus deducing behavioral patterns of the user of interest to him. Such information can subsequently be sold to interested parties for marketing purposes and targeted advertising.

9. Denial-of-Service attack: Typically, things have tight memory and limited computation; they are thus vulnerable to resource exhaustion attack. Attackers can continuously send requests to be processed by specific things so as to deplete their resources. This is especially dangerous in the IoTs since an attacker might be located in the backend and target resource-constrained devices in an LLN. Additionally, DoS attack can be launched by physically jamming the communication channel, thus breaking down the T2T communication channel. Network availability can also be disrupted by flooding the network with a large number of packets.

2. SECURITY SYSTEM

The term security subsumes a wide range of different concepts. In the first place, it refers to the basic provision of security services including confidentiality, authentication, integrity, authorization, non-repudiation, and availability. These security services can be implemented by means of different cryptographic mechanisms, such as block ciphers, hash functions, or signature algorithms. For each of these mechanisms, a solid key management infrastructure is fundamental to handling the required cryptographic keys.

In the context of the IoT, however, security must not only focus on the required security services, but also on how these are realized in the overall system and how the security functionalities are executed. This section provides a security approach to an IP based network. We use the following terminology to analyze and classify security aspects in the IoT:

- *The security architecture* refers to the system elements involved in the management of the security relationships between things and the way these security interactions are handled (e.g., centralized or distributed) during the lifecycle of a thing.
- *The security model* of a node describes how the security parameters, processes, and applications are managed in a thing. This includes aspects such as process separation, secure storage of keying materials, etc.
- *Security bootstrapping* denotes the process by which a thing securely joins the IoT at a given location and point in time. Bootstrapping includes the authentication and authorization of a device as well as the transfer of security parameters allowing for trusted operation.
- *Network security* describes the mechanisms applied within a network to ensure trusted operation of the IoT. Specifically, it prevents attackers from endangering or modifying the expected operation of networked things. Network security can include a number of mechanisms ranging from secure routing to data link layer and network layer security.
- *Application security* guarantees that only trusted instances of an application running in the IoT can communicate with each other, while illegitimate instances cannot interfere.

We now discuss an exemplary and traditional security architecture relying on a configuration entity for the management of the system with regard to the introduced security aspects (see Figure 1). Assume a centralized architecture in which a configuration entity stores and manages the identities of the things associated with the system along with their cryptographic keys. During the bootstrapping phase, each thing executes the bootstrapping protocol with the configuration entity, thus, obtaining the required device identities and the keying material. The security service on a thing in turn stores the received keying material for the network layer and application security mechanisms to resort for secure communication.

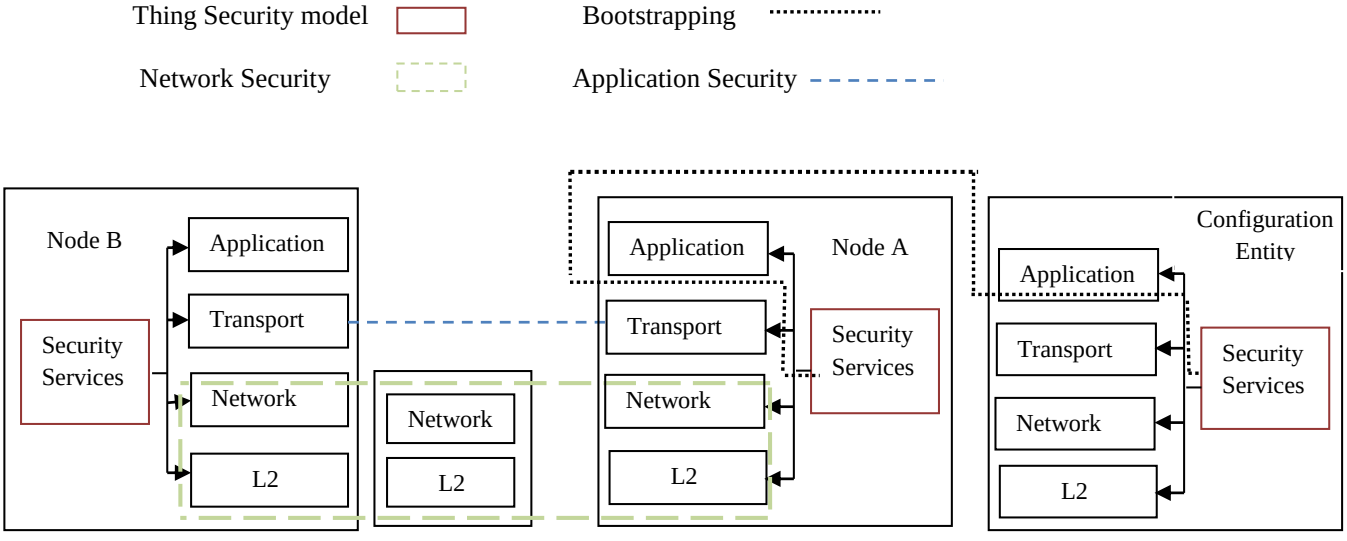

Fig. 1. Overview of Security Mechanisms.

Things can then securely communicate with each other during their operational phase by means of the deployed network and application security mechanisms.

2. 1. IP-based security solutions for IoT

In the current scenario, there exists a multitude of control protocols for the IoT. Recent trends, however, focus on an IP approach for system control. Currently, a number of IETF working groups are designing new protocols for resource constrained networks of smart things. The 6LoWPAN working group [4] focuses on the definition of methods and protocols for the efficient transmission and adaptation of IPv6 packets over IEEE 802.15.4 networks [5]. The CoRE working group [8] provides a framework for resource-oriented applications intended to run on constrained IP network (6LoWPAN). One of its main tasks is the definition of a lightweight version of the HTTP protocol, the Constrained Application Protocol (CoAP) [9] that runs over UDP and enables efficient application-level communication for things.

In the context of the IP-based IoT solutions, consideration of TCP/IP security protocols is important as these protocols are designed to fit the IP network ideology and technology. While a wide range of specialized as well as general-purpose key exchange and security solutions exist for the Internet domain, such as IKEv2/IPsec [10], TLS/SSL [11], DTLS [12], HIP [13, 14], PANA [15], and EAP [16] etc. Many of these protocols are currently discussed as candidate solutions in the 6LoWPAN and CoRE IETF working groups. Application layer solutions such as SSH [17] also exist, however, these are currently not considered. Figure 2 depicts the relationships between the discussed protocols and how they can be used in the security system of an IoT [18].

The Internet Key Exchange (IKEv2)/IPsec and the Host Identity Protocol (HIP) reside at or above the network layer in the OSI model. Both protocols are able to perform an authenticated key exchange and set up the IPsec transforms for secure payload delivery.

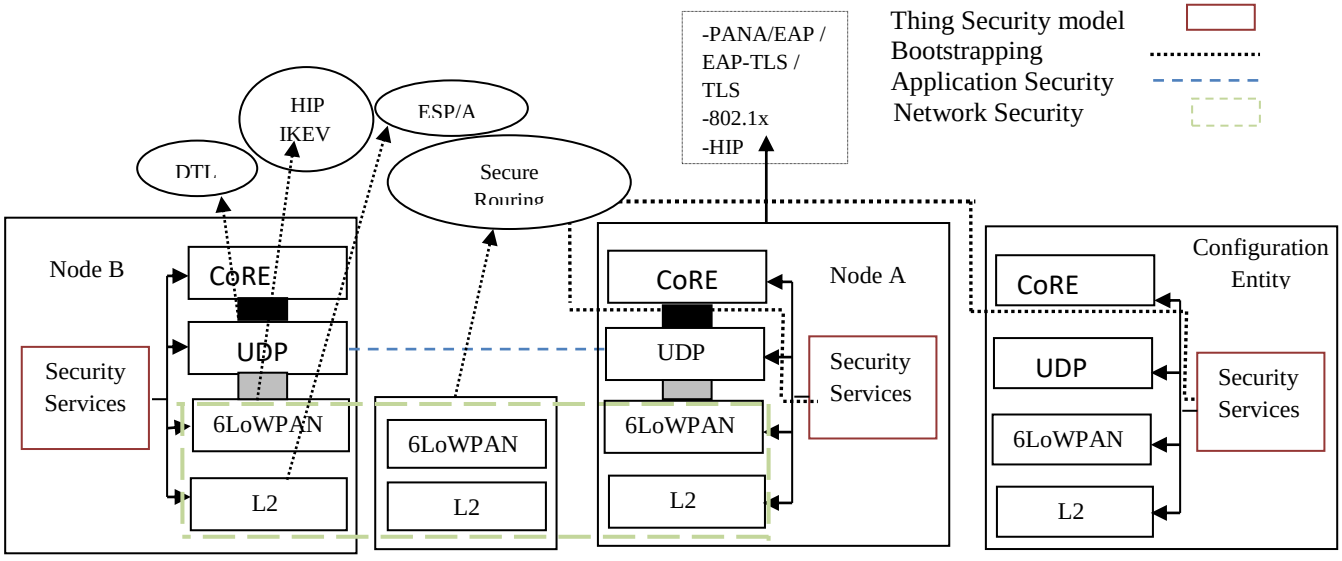

Fig. 2. Relationships between IP-based security protocols.

Currently, there are also ongoing efforts to create a HIP variant coined Diet HIP [13] that takes lossy low-power networks into account at the authentication and key exchange level. Transport Layer Security (TLS) and its datagram-oriented variant DTLS secure transport-layer connections. TLS provides security for TCP and requires a reliable transport, while DTLS secures and uses datagram-oriented protocols such as UDP. Both protocols are intentionally kept similar and share the same ideology and cipher suites. The Extensible Authentication Protocol (EAP) is an authentication frame-work supporting multiple authentication methods. EAP runs directly over the data link layer and, thus, does not require the deployment of IP. It supports duplicate detection and retransmission, but does not allow for packet fragmentation. The Protocol for Carrying Authentication for Network Access (PANA) is a network-layer transport for EAP that enables network access authentication between clients and the network infrastructure. In EAP terms, PANA is a UDP-based EAP lower layer that runs between the EAP peer and the EAP authenticator [1-8]

Even though 6LoWPAN and CoAP progress towards reducing the gap between Internet protocols and the IoT, they do not target protocol specifications that are identical to their Internet pendants due to performance reasons. Hence, more or less subtle differences between IoT protocols and Internet protocols will remain. While these differences can easily be bridged with protocol translators at gateways, they become major obstacles if end-to-end security measures between IoT devices and Internet hosts are used.

3. CONCLUSION

Starting from discussing the security challenges in an IoT network, this paper reviewed the architectural design for a secure IP-based Internet of Things and its challenges with special focus on standard IP security protocols. A first conclusion refers to the fact that the security architecture should fit the lifecycle of a thing and its capabilities. This includes aspects such as the way security domain is created, the need for a trusted-third party in this process, or the type of protocols applied. Security protocols should further take into account the resource-constrained nature of things and heterogeneous communication models. The link layer, the network layer, as well as the application layer has distinct security requirements and communication patterns. As future work, we aim at a deeper feasibility analysis of the discussed protocols in different settings and for different trust models.

References

[1] IETF 6LoWPAN Working Group. http://tools.ietf.org/wg/6lowpan/. online, last visited 30. June 2011.

[2] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler. Transmission of IPv6 Packets over IEEE 802.15.4 Networks, RFC 4944, September 2007.

[3] Heer, T., Garcia-Morchon, O., Hummen, R. *et al.* Security Challenges in the IP-based Internet of Things. *Wireless Pers Commun* 61, 527–542 (2011). https://doi.org/10.1007/s11277-011-0385-5

[4] AUTO-ID LABS. http://www.autoidlabs.org/. online, last visited 30. June 2011.

[5] E. Kim, D. Kaspar, N. Chevrollier, and JP. Vasseur. Design and Application Spaces for 6LoWPANs draft-ietf-6lowpan-usecases-09, January 2011.

[6] S. Blake-Wilson, N. Bolyard, V. Gupta, C. Hawk, and B. Moeller. "Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)" RFC 4492 (Informational), May 2006. Updated by RFC 5246.

[7] D. Fu and J. Solinas. Elliptic Curve Groups modulo a Prime (ECP Groups) for IKE and IKEv2, RFC 5903 (Informational), June 2010.

[8] IETF Constrained REST ful Environment (CoRE) Working Group. https://datatracker.ietf.org /wg /core/charter /. online, last visited 30. June 2011.

[9] Z. Shelby, K. Hartke, C. Bormann, and B. Frank. " Constrained Application Protocol (CoAP). draft-ietf-core- coap-04 (Internet Draft)", January 2011.

[10] C. Kaufman. Internet Key Exchange (IKEv2) Protocol, RFC 4306, December 2005. Updated by RFC 5282.

[11] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2, RFC 5246, August 2008. Updated by RFCs 5746, 5878.

[12] T. Phelan. Datagram Transport Layer Security (DTLS) over the Datagram Congestion Control Protocol (DCCP), RFC 5238, May 2008.

[13] R. Moskowitz, P. Nikander, P. Jokela, and T. Henderson. Host Identity Protocol. RFC 5201 (Experimental), April 2008.

[14] R. Moskowitz, P. Jokela, T. Henderson, and T. Heer. Host Identity Protocol Version 2. draft-ietf-hip-rfc5201- bis-03 (Work in progress), October 2011.

[15] D. Forsberg, Y. Ohba, B. Patil, H. Tschofenig, and A. Yegin. Protocol for Carrying Authentication for Network Access (PANA). RFC 5191, May 2008.

[16] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz. Extensible Authentication Protocol (EAP), RFC 3748, June 2004.

[17] T. Ylonen and C. Lonvick. The Secure Shell (SSH) Protocol Architecture. RFC4251, January 2006.

[18] Perrig, A., Szewczyk, R., Tygar, J. *et al.* SPINS: Security Protocols for Sensor Networks. *Wireless Networks* 8, 521–534 (2002). https://doi.org/10.1023/A:1016598314198