International Journal of Control and Automation
Vol. 11, No. 5 (2018), pp.37-48
http//dx.doi.org/10.14257/ijca.2018.11.5.04

Entity Authentication and Secure Registration for Lightweight Devices in Internet of Things

Namhi Kang and Jeongin Kim

Duksung Women's University, Seoul, Korea
kang@duksung.ac.kr, jeonginkim3@gmail.com

Abstract

The number of devices connected to the Internet is increasing as the Internet of Things (IoT) technologies are evolving. The increasing number of the Internet connected IoT devices result in highly increasing amount of exposing data, thereby increasing privacy invasion and threat of survival as well. Therefore, security in the IoT service must be provided. Recently, there have been several hacking attacks mainly targeting small devices that do not change an initially configured default password. To respond to these security threats, devices must be securely configured. However, most of the small devices that are deployed for IoT service have restricted user interface (UI). Moreover, general users have difficulties in performing initial configuration safely because of their lack of security knowledge. Also, in order to provide secure IoT services, a new IoT device and an authentication server which register the device are to be mutually authenticated each other. In this paper, we propose a method to perform mutual authentication between the new device and the authentication server by using NFC as an additional communication channel and to securely configure the registration and initialization key of the IoT device.

Keywords: *Internet of thing, NFC, Mutual Authentication, Secure bootstrapping, Secure Registration*

1. Introduction

IoT (Internet of Things) technology enables information sharing between users and objects and between objects and objects by connecting them with each other. IoT devices are increasing, as IoT technology is evolving. By 2020, 26 billion devices will be connected to the network, and most of them are expected to be wearable devices and IoT devices [1]. As IoT devices connected to the network increase, the amount of data in danger of exposure, such as medical information, lifestyle, and user location, is also increased. The security in the IoT environment must be guaranteed; otherwise, it may lead to life-threatening problems as well as violation of privacy. However, most IoT devices are vulnerable to attack because security issues are not considered [2, 3].

Recently, Dyn, a US Domain Name System (DNS) provider has received Distributed Denial of Service (DDoS) attacks. It is found that Mirai, a malicious code targeted at IoT devices, caused the attack. Mirai caused the DDoS attack by using as tools the IoT devices that are vulnerable in security and that have a default password unchanged [4]. The risk of attacks can be reduced by resetting the access password of devices securely, but many IoT devices are using the default settings (*e.g.*, 1234) without modification. In addition, most of IoT lightweight devices often have limited user interface (UI) or none, and it is difficult for general users lacking of security knowledge to reset the default password securely.

Received (January 3, 2018), Review Result (March 5, 2018), Accepted (March 12, 2018)

ISSN: 2005-4297 IJCA

Session key must be securely shared to provide encryption or integrity in a communication session. The Pre-Shared Key (PSK) exchange method is suitable for the IoT environment with limited resources, because public key operation is not required in the method [5]. Before using PSK, the secret key must be securely shared in advance by using a secure channel between the communication subjects [6]. Most studies assume that PSK is registered securely in advance. However, in the IoT environment consisting of various heterogeneous devices, it is difficult to assume that all devices register PSKs securely.

In order to provide secure IoT services, mutual authentication between a device and an authentication server is necessary [20]. Without mutual authentication process, a device or an authentication server would share keys with a malicious attacker and be exposed to threats of attack. Before registering the device, the user may not believe that the newly purchased IoT device is an appropriate one, and the IoT device may not be sure if the server to be registered is appropriate as well.

This article suggests a technology to register devices and reset their initial key (default password) through mutual authentication when there exists no trust between a user and a purchased IoT.

The use of WiFi, ZigBee, and BlueTooth to send and receive messages for mutual authentication is exposed to attacks such as eavesdropping and counterfeit attacks because of its wide communication range. IoT devices have limited resources and lack an UI. Therefore, in this paper, Near Field Communication (NFC) having a small communication radius was used as an additional transmission channel for transmitting safe initial setup data. NFC provides a data transmission speed of 106 Kbps to 424 Kbps, and less than 0.1 second recognition speed in the local area communication within 10 cm in the 13.56 MHz frequency band. The communication range is so narrow that it may cope with physical (looking around) attacks. However, since NFC also has received an attack before, this article suggests an initial password resetting method that does not expose important information even if NFC is used to send and receive necessary messages to reset the initial password.

The content of this paper is as follows. Chapter 2 explains the related studies, Chapter 3 explains the operation steps of the proposed system, and Chapter 4 verifies the proposed system using security analysis and security verification tool, Scyther. Finally, Chapter 5 concludes this paper.

2. Related Work

As mentioned in the introduction, it is difficult to apply the present initialization method to the devices used in the IoT environment because they do not have UI and are dependent on batteries.

NFC has a communication radius of approximately 10 cm, which is smaller than that of WiFi, ZigBee and BlueTooth. Because NFC has a small communication range, it is relatively secure against physical attacks in comparison with WiFi, ZigBee and BlueTooth, if the user carefully looks around. In the case of the IoT security setting method using NFC as the Out-Of-Band (OOB) channel, this NFC communication section opens an initial key for encryption and decryption, and transmits it. This is because the communication radius is narrow and messages are transmitted by using the relatively secure NFC [7]. However, while NFC is relatively secure in comparison with the other wireless communications, NFC is also under the risk of attacks. Among the NFC operation modes, the active operating mode of NFC has the 10 m eavesdropping range, but as for the passive mode, the range is significantly reduced to about 1 m. Data transmitted in the passive mode may be more secure than the one in the active mode. However, when transmitting sensitive data, an attacker may use an antenna to intercept messages sent over the

NFC range, and thus the data transmission may not be secure even in the passive mode [8, 9]. Consequently, if messages of an NFC communication section are eavesdropped in the IoT security setting protocol using NFC as the OOB channel, the messages to generate a PSK may be exposed, and the PSK may not be considered as secure.

Similar to NFC, tags and readers in Radio Frequency Identification (RFID) systems are vulnerable to message eavesdropping and replay attacks because they use radio frequencies. An authentication protocol for RFID systems was proposed to solve this problem. Readers and tags only perform the symmetric key encryption and the XOR operation to generate a random value for mutual authentication, and thus the protocol is energy efficient. However, the protocol may easily be exposed to an attacker because all the readers and tags have the encryption and decryption key that are not updated [10]. To solve this problem, another system has been proposed, wherein a dynamic ID changed in each communication session is used and only the symmetric key encryption and the XOR operation are used by considering the limited resources of tags. The proposed system is also designed to prevent attacks on the communication of other tags even if the network general key is exposed [11].

An unchanged default password of devices may result in attacks. As IoT devices are increasing, search engines for IoT devices are also increasing. One search engine named Shodan lets the user easily search for devices that do not have patched service programs or that still use the default password. These devices may be accessed without the ID and the password, and an attacker who obtained the root privilege is able to control the system [12]. These devices are vulnerable to attacks.

OpenID is a lightweight ID system, which is proposed to be used as a method for user authentication in the IoT environment [13]. A registered OpenID service provider or a user domain appoints Home Registration Authority (HRA). When a device requests access, Registration Authority (RA), acting as an access point or gateway, requests to HRA to present the ID for user authentication, and a session key based on an Elliptic Curve Cryptosystem (ECC) is generated after user authentication.

3. Proposed System

This chapter suggests a method to register a new device by using NFC and to securely reset the default password. As shown in Figure 1, the proposed system consists of four subjects: a new device, a controller, an authentication server and a manufacturer server. While the controller, the authentication server and the manufacturer server have abundant resources for computing and communication, the new device has limited resources for initial setup and registration. The new device and the authentication server cannot transmit data to each other directly because NFC has a short communication range. Therefore, the controller that shares a secret key with the authentication server is used as a mediator, and the new device and the controller exchange data by using NFC as a communication channel safe to physical attacks. The controller, the authentication server and the manufacturer server having abundant resources use Internet communications such as WiFi, Zigbee, and BlueTooth.

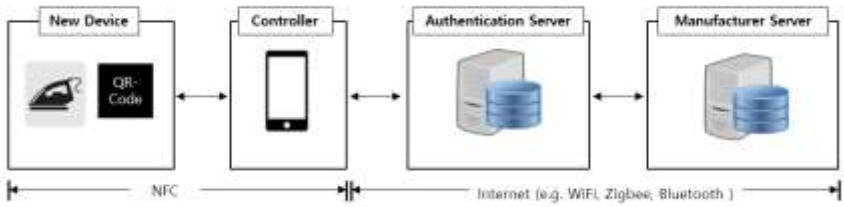

Figure1. System Configuration Diagram

The following Table 1 is the parameter used in the proposed system. The information to be set in advance before operating the proposed system and the trust relationship between individual devices at the first running time are based on the following assumptions:

- The new device and the controller do not trust each other until the secure setting and registration are completed.
- The authentication server and the manufacturer server do not trust each other until the verification of the certificate of each subject is completed.

Table.1. Abbreviation Notation and Parameters

Notation	Description
ID_D	Identifier of a Device 'D' (32bit identifier)
IK_D	Initial key for a Device 'D' (128bit Symmetric key)
FW_E	Firmware of a device
PSK_E	128bit Pre-Shared key between device and authentication server
RN_x	Random Nonce of an entity 'x' (128bit integer)
SK_{cs}	128bit Secret key between Controller and authentication servers
TID_D	Transaction Identifier of a Device 'D'
$Cert_x$	Certification of an entity 'x'
$Sock_x$	IP, PORT of an entity 'x'
TS_i	ith time stamp
$PubKey_x$	Public Key of an entity 'x'
VT_E	Valid Time of PSK_E
H(·)	Secure hash Function

As shown in the following Figure 2, the controller and the authentication server are assumed to have a trust relationship in advance, are the environment that a user belongs to, and share a mutual secret key SK_{cs}.

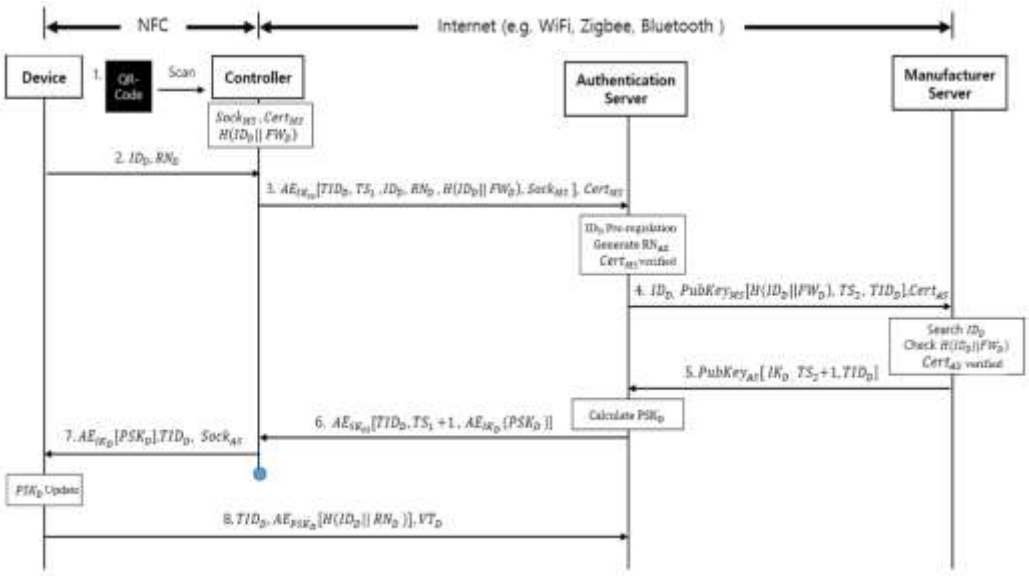

Figure 2. System Operation Steps for the Proposed Scheme

The manufacturer server and the new device have an initialization key IK_D. The manufacturer gives a new device QR code that contains $Sock_{MS}$ having IP and PORT information of the manufacturer server, $H(ID|FW_D)$ for the firmware verification, and the certificate $Cert_{AS}$ of the manufacturer server.

ID_D and IK_D of devices and the firmware value FW_E of devices are securely stored in the database of the manufacturer server. In the communication between the controller and the authentication server, and between the authentication server and the manufacturer, messages are exchanged through a secure channel.

Figure 2 is a schematic diagram of the operation steps of the proposed system, and the operation procedures are described below.

1) The new device and the controller do not trust each other until the new device is registered in the authentication server. The controller uses a QR-Code to obtain the information to verify the new device. The proposed system securely verifies a value received in the setting process based on the premise that the QR-Code can easily be maliciously modified. The QR-Code contains IP address and port number of the manufacturer server $Sock_{MS}$ for the internet communication between the authentication server and the manufacturer server, a hashed value $H(ID|FW_D)$ by the identifier ID_D and the firmware value FW_E of the device to verify the device firmware, and the certificate of the manufacturer server $Cert_{AS}$. The controller scans the QR-Code to obtain $Sock_{MS}$, $Cert_{AS}$ and $H(ID|FW_D)$ before the step 2).

2) The new device transmits a random nonce RN_E to the controller using NFC to generate ID_D and PSK_E.

3) The controller generates a TID_D to distinguish transactions of new devices, as the controller may set multiple devices at the same time. The controller also encrypts TS_1 to cope with replay attack between the controller and the authentication server, ID_D, RN_E, $H(ID|FW_D)$, $Sock_{MS}$ and TID_D with the secret key SK_{CS}, shared with the authentication server in advance, and then send those values to the authentication server with the certificate $Cert_{AS}$.

4) The authentication server verifies $Cert_{AS}$ received from the controller. Various present methods may be applied to the verification of the certificate [14, 15]. The authentication server temporarily registers ID_D of the new device and generates a random nonce RN_{AS} to create PSK_E. The authentication server uses the $Sock_{MS}$ received

from the previous step and encrypts $H(ID|FW_D)$, TS_3 that copes with replay attack between the authentication server and the manufacturer server, and TID_D with the public key of the manufacturer server. Then, the authentication server transmits those values to the manufacturer server with ID_D and $Cert_{AS}$.

5) The manufacturer server verifies the received $Cert_{AS}$ and identifies the authentication server, and retrieves the FW_L of the device using the received ID_D. The manufacturer server compares the received ID_D and the retrieved FW_L with the $H(ID|FW_D)$, which is received from the authentication server by hashing an adjacent value and encrypting it. If matched, IK_D corresponding to ID_D and TID_D, and TS_2+1 of TS_3 received in step 4) are encrypted with the public key of the authentication server and those values are transmitted to the authentication server. Since public key encryption may be performed by anyone if the certificate is disclosed, the public key has to be encrypted with TS_2+1 of TS_3 received from step 4). The operation to obtain TS_2+1 is performed only by the subject that can decrypt step 4) and obtain TS_3.

6) After confirming which new device has IK_D using TID_D, the authentication server generates PSK_L by using equation (1) and then encrypts it with IK_D. The authentication server encrypts $AE_{IK_D}(PSK_D)$, TID_D, TS_2+1 into the secret key SK_{C1} and transmits to the controller.

$$PSK_D = E_{IK_D}(RN_D \oplus RN_{AS}) \tag{1}$$

7) The controller finds the matching new device by using TID_D and transmits $AE_{IK_D}(PSK_D)$ with $Sock_{AS}$ and TID_D for the communication between the new device and the authentication server.

8) The new device decrypts the received value with IK_D and updates PSK_L, then sends TID_D and $AE_{PSK_D}(H(ID_D||RN_D))$ to the authentication server. The authentication server hashes ID_D and RN_L, and verifies the device by comparing with $H(ID_D||RN_D)$ received from the new device. If matched, the authentication server registers the device.

Even if there is not a trust relationship between the new device and the controller, the new device and the controller may verify each other, and the authentication server may register an appropriate new device. In addition, the default password can be reset securely, and the controller may verify the integrity of the firmware of the new device.

4. Security Analysis

4.1. Safety Analysis

4.1.1 Replay Attack

This system is able to cope with replay attack even if an attacker eavesdrops messages between a controller and an authentication server and causes replay attack after a certain period of time. This is because the messages include a timestamp TS_1 and a random value RN_L newly generated for each session, which make the attacker unable to reuse messages received from the previous session.

4.1.2. Man in the Middle Attack

Since NFC is used for the communication between a new device and a controller and the user carefully looks around before exchanging a message, it may be hard for an attacker to intercept the message. Even if an attacker intercepts ID_D in step 2 of the proposed system and obtains the message of step 7, the attacker is unable to obtain PSK_L because of not knowing the initial key IK_D. In addition, the authentication

server is able to recognize that there is an attacker in the middle if the response does not arrive within a certain time, because the attacker is unable to carry out step 8. Even when an attacker intercepts a message between the controller and the authentication server, the attacker is unable to obtain PSK_I of the device because the attacker does not know the secret key SK_{ci} shared by the controller and the authentication server in advance.

4.1.3. Impersonation Attack

The proposed system is able to cope with impersonation attacks from the following unauthorized devices:

- Unauthorized IoT lightweight devices

  Even though an unauthorized device transmits the ID value and the random nonce value to the controller, the manufacturer server may not find the firmware value matching the ID value of the unauthorized device; therefore, the device may not be registered.

- Unauthorized controllers

  An unauthorized controller is unable obtain the secret key SK_{ci} which is shared with the authentication server in advance, and thus the unauthorized controller is unable to obtain the message to verify the relationship between the new device and the controller.

- Unauthorized authentication servers

  Since an unauthorized authentication server may not obtain the secret key SK_{ci} shared with the controller in advance, the controller may recognize that the server is an unauthorized one.

4.1.4. Mutual Entity Authentication

If the new device obtains PSK_I by decrypting with IK_D in step 7, it means the manufacturer server has authenticated the authentication server. The authentication server may also authenticate the new device by comparing the hashed value of ID_D and RN_I received from step 8 with the hashed value of ID_D and RN_I that the authentication server already has.

4.2. Security Verification using the Scyther Tool

This paper used the Scyther tool to check the security protocol of the proposed system. The Scyther is a tool for security protocol verification and counterfeit analysis. The Scyrher tool has been used in other previous studies for protocol verification [16, 17]. The security attribute claims for verification are Alive, Weakagree, Niagree, Nisynch, and others. Each claim verifies whether a protocol is safe from man-in-the-middle attack, replay attack, and reflection attack [18]. The Scyther tool provides three methods for protocol verification [19].

- Verification of claims: The Scyther tool can directly set the security attributes of the protocol using claims, and write them in the protocol decryption for the verification.
- Automatic claims: If claims are not set, the Scyther automatically generates claims to verify the protocol.
- Characterization: The Scyther tool analyzes a protocol and shows a trace containing the execution of the protocol role.


```
protocol NFC (D, C, AS, MS)
{
	role D // new Device
	{
		fresh IDd, RNd: Nonce;
		var TID, RNas : Nonce;

		send_1 (D,C, IDd, RNd);
		recv_6 (C, D,  {{XOR(RNd,RNas)}IKd}IKd ,TID);

	};

	role C // Controller
	{

		var IDd, RNd: Nonce;
		fresh TID, IDd, FW: Nonce;

		var RNas:Nonce;
		fresh TS1: Timestamp;

		recv_1 (D,C, IDd, RNd);
		send_2 (C,AS, {TID, TS1, IDd, RNd, H(FW,IDd)}SKcs);
		recv_5 (AS, C, {TID, TS1, {{XOR(RNd,RNas)}IKd}IKd}SKcs);
		send_6 (C, D,  {{XOR(RNd,RNas)}IKd}IKd ,TID);

	};

	role AS // Authentication Server
	{

		var TID, IDd, RNd, FW : Nonce;
		fresh RNas : Nonce;
		var TS1 : Timestamp;
		fresh TS2 : Timestamp;

		recv_2 (C,AS, {TID, TS1, IDd, RNd, H(FW,IDd)}SKcs);
		send_3 (AS, MS, IDd, {H(FW,IDd), TS2, TID}pk(MS));
		recv_4 (MS, AS, {IKd, TS2, TID}pk(AS));
		send_5 (AS, C, {TID, TS1, {{XOR(RNd,RNas)}IKd}IKd}SKcs);

	};

	role MS // Manufacturer Server
	{

		fresh FW: Nonce;
		var IDd, TID, FW : Nonce;
		var TS2 : Timestamp;

		recv_3 (AS, MS, IDd, {H(FW,IDd), TS2, TID}pk(MS));
		send_4 (MS, AS, {IKd, TS2, TID}pk(AS));

	};
}
```

Figure 3. Proposed Protocol Specified in SPDL

In this paper, the security of the proposed protocol was verified by directly setting security attributes (Verification of claims). The protocol operation procedure and the messages sent and received are described in Security Protocol Description Language (SPDL). D, C, AS, and MS in Figure 3 and Figure 4 represent Device, Controller, Authentication Server, and Manufacturer Server of the proposed system. As shown in Figure 3, a subject sending and receiving messages is defined as a role that sends and receives messages using 'send' and 'recv.' This enables the detection of several possible attacks and generates graphs for each attack detected by claims.

Figure 4 shows the verification result of the proposed system using the Scyther tool. If each security attribute is satisfied, the status displays OK; if not satisfied, it displays Fail. As an example, the security attribute statuses of Alive, Weakagree, Niagree, and Nisynch of D are OK, which confirmed the proposed system is safe from the man-in-the-middle attack, replay attack, and reflection attack. In addition, it showed that the reset password (initial key) of a new device by using the Secret Key (PSK) is secure.

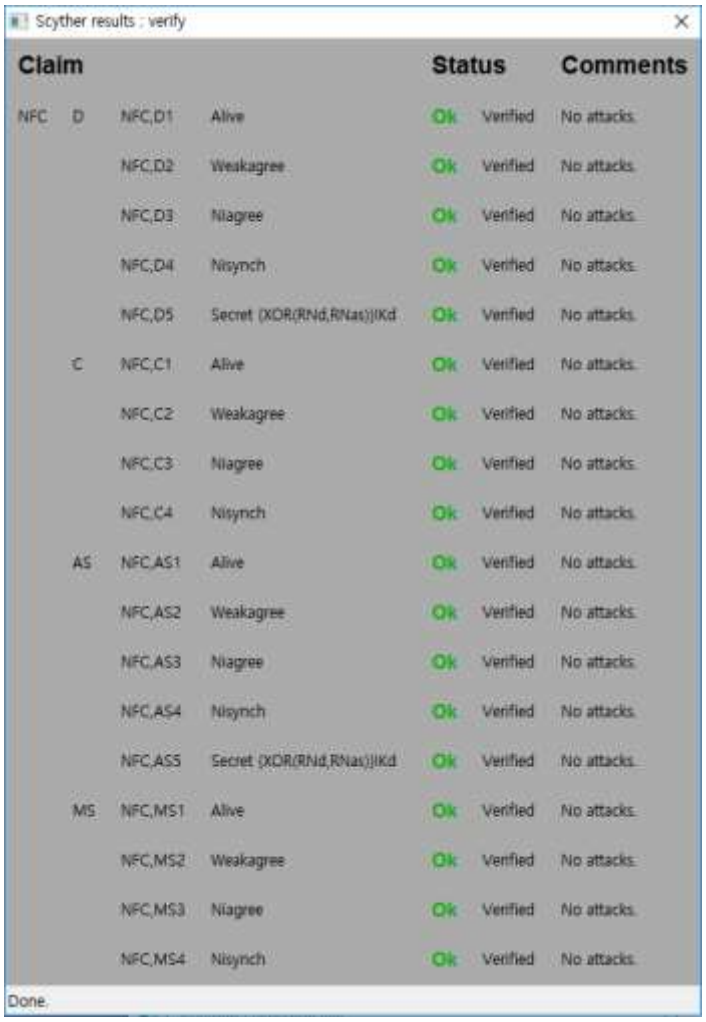

Figure 4. Verification Results using the Syther Tool

5. Conclusion

For general users lacking security knowledge, it is difficult to initialize IoT devices without user interface and to register devices securely. In order to solve this issue, this article suggests a technology for secure registration of IoT devices with limited resources to be used in application services. In particular, securely resetting the initial secret value (connection password, preset secret key, *etc*.,) set in IoT devices is key to this technology. Accordingly, this technology may be used when an additional resetting of the secret value is required by IoT devices that have been applied to services. In addition, to prevent malicious devices from being installed in the network, this technology provides mutual device authentication between IoT devices and registration devices by using the registration server operated by vendors as well as the authentication server operated for services.

The proposed technology is mainly applied to IoT devices that have a limited user interface or none to input information from users or to display the progress. The IoT devices to which this technology may be applied are the devices supporting NFC communication module that supplements the limited user input and output interface. In other words, NFC communication in the proposed technology is used as an additional data channel to register IoT devices. Data transmission for receiving services after the registration process may be performed by using various wireless access technologies (*e.g.*, WiFi, Zigbee, BlueTooth, *etc*.).

This paper showed through analysis that the proposed technology may cope with man-in-the-middle attack, eavesdropping attack, and others. In addition, we verified by using the Scyther tool that the proposed protocol is safe from various attacks.

References

[1] P. Middleton, P. Kjeldsen and J. Tully, "Forecast: The internet of things", Gartner worldwide, **(2013)**.
[2] J. Wurm, K. Hoang, O. Arias, A. R. Sadeghi and Y. Jin, "Security analysis on consumer and industrial iot devices", IEEE 21st Asia and South Pacific Design Automation Conference (ASP-DAC), Macao, China, **(2016)** January 25-28.
[3] S.-s. Kim, M.-s. Jun and D.-h. Choi, "Chameleon Hash-Based Mutual Authentication Protocol for Secure Communications in OneM2M Environments", Journal of KICS, vol. 40, no. 10, **(2015)**, pp. 1958-1968.
[4] K. Angrishi, "Turning Internet of Things (IoT) into Internet of Vulnerabilities (IoV): IoT Botnets", arXiv preprint arXiv:1702.03681, **(2017)**.
[5] H. Kwon and N. Kang, "Analysis on energy consumption required for building DTLS session between lightweight devices in internet of things", Journal of KICS, vol. 40, no. 8, **(2015)**, pp. 1588-1596.
[6] P. Eronen and H. Tschofenig, "Pre-shared key ciphersuites for transport layer security (TLS)", IETF Standard, RFC 4279, **(2005)**.
[7] J. Kim and N. Kang, "Secure Configuration Scheme for Internet of Things using NFC as OOB Channel", Journal of IIBC, vol. 16, no. 3, **(2016)**, pp. 13-19.
[8] E. Haselsteiner and K. Breitfuß, "Security in near field communication (NFC)", In Workshop on RFID security, **(2016)**, pp. 12-14.
[9] H. Kortvedt and S. Mjolsnes, "Eavesdropping near field communication", The Norwegian Information Security Conference (NISK), vol. 27, **(2009)**.
[10] S. Oh, C. Lee, T. Yun, K. Chung and K. Ahn, "Improved authentication protocol for privacy protection in RFID systems", Journal of KICS, vol. 38C, no. 1, **(2013)**, pp. 12-18.
[11] J. Kim and D. Won, "Security Analysis and Improvements of Authentication Protocol for Privacy Protection in RFID Systems", Journal of KICS, vol. 41, no. 5, **(2016)**, pp. 581-591.
[12] K.-H. Han and S.-H. Lee, "A Study on the Security Threats of IoT Devices Exposed in Search Engine", Journal of KIEE, vol. 65, no. 1, **(2016)**, pp. 128-134.
[13] J. Liu, Y. Xiao and C. L. Philip Chen, "Authentication and access control in the internet of things", IEEE Distributed Computing Systems Workshops (ICDCSW), Macau, China, **(2012)** June 18-21.
[14] A. Bates, J. Pletcher, T. Nichols, B. Hollembaek, D. Tian, K. R. Butler and A. Alkhelaifi, "Securing SSL certificate verification through dynamic linking", In Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (ACM), Scottsdale, Arizona, **(2014)**, November 3-7.
[15] Wasef, A., and Shen, X. "ASIC: Aggregate signatures and certificates verification scheme for vehicular networks", In Global Telecommunications Conference (GLOBECOM), Honolulu, HI, USA, **(2009)**, November 30-December 4.
[16] N. El Madhoun, F. Guenane and G. Pujolle, "An online security protocol for nfc payment: Formally analyzed by the scyther tool", IEEE Mobile and Secure Services (MobiSecServ), Gainesville, Florida, **(2016)**, February 26-27.
[17] H. A. Elbaz, "Analysis and Verification of a Key Agreement Protocol over Cloud Computing Using Scyther Tool", International Journal of Cloud Computing and Services Science (IJ-CLOSER), vol. 3, no. 6, **(2014)**.
[18] G. Lowe, "A hierarchy of authentication specifications", IEEE Computer security foundations workshop, Rockport, Massachusetts, **(1997)** June 10-12.
[19] C. J. F. Cremers, "The scyther tool: Verification, falsification, and analysis of security protocols", Springer Berlin Heidelberg, **(2008)**.
[20] J. Kim and N. Kang, "Secure Key Configuration and Mutual Entity Authentication Scheme for Lightweight IoT Devices", The 1st Int. Conf. on Convergent Research Theory and Technology, **(2017)** August 20-22.

Authors

Namhi Kang received B.E. and M.S. degrees in Electrical and Communications Engineering from Soongsil University, Korea in 1999 and 2001, respectively. He received a Ph.D. degree in Information and Communications Engineering from Siegen University, Germany, in December 2004. During studies he concentrated on system security and secure multimedia communications. In 2007 he started to teach at the Department of Computer Engineering, Catholic University. Since 2009, he has been a professor in the Department of IT Media Engineering, Duksung Women's University, where he leads a research team working in computer networks and security Laboratory. His research interests include QoS and security in wired and wireless networks, and the development of secure applications for smart media. He has recently become interested in the secure communication protocols of constrained devices in Internet of Things (IoT).

Jeongin Kim received her B.E. and M.Sc. in Computer Science and Information Technology from Duksung women's university, Seoul, Korea in 2015 and 2017 respectively. She was a research member of computer networks and security Laboratory of Duksung women's university. Her research interests include security in wired and wireless networks, and the development of secure applications for smart media. She has recently become interested in the secure communication protocols of constrained devices in Internet of Things (IoT).