Automatic and Secure Wi-Fi Connection Mechanisms for IoT End-Devices and Gateways

Fu-Chiung Cheng(✉)

Tatung University, Taipei, Taiwan 104, Republic of China
fccheng@ttu.edu.tw

Abstract. Internet of Things (IoT) are developed rapidly in recent years and more than 50 billion of IoT devices are expected to be deployed worldwide in 2020. How to automatically and securely connect the tremendous number of IoT end devices to Internet is one of critical problems to be addressed. This paper proposes secure and automatic Wi-Fi (Wireless Fidelity) connection mechanisms for connecting IoT end devices and IoT gateways. Our design has the following advantages. First, IoT end devices, once powered on, can automatically connect to an IoT gateway without human intervention. Secondly, the SSID and password for high security strength WPA2 connection are randomly generated to enhance IoT security. Finally, the randomly generated password and SSID are automatically changed every day or when network attacking is detected.

Keywords: Automatic connection · IoT security
Wi-Fi wireless communication · IoT applications

1 Introduction

According to Gartner's forecast report [1] on the Internet of Things, the number of connected IoT devices in global use in 2017 will reach 8.4 billion, which is a 31% increase from 2016, and it will increase to 20.4 billion by 2020. The amount of expenditures related to service and endpoints will also reach $2 trillion in 2017. Internet of Things has become the most important research area in the industry as well as in academia. In addition, a newer Gartner's report [2] estimates worldwide spending on IoT security will reach $1.5 billion in 2018, a 28 percent increase from 2017. A more widespread and optimistic report from Cisco [3] estimates that the number of connected devices on the Internet will exceed 50 billion by 2020. It is no double that IoT has huge market values in industries and becomes very important research area in academia.

The current Wi-Fi technology provides high secure wireless communication mechanism such as WPA3 and WPA2 [4], but it may be inconvenience for people since setting passwords and SSID [12] is needed to connect to Access Points (AP). For machine to machine IoT applications, it is formidable and undesirable to set passwords by hand due to the large number of IoT clients. This paper proposes original algorithms

M. H. Miraz et al. (Eds.): iCETiC 2018, LNICST 200, pp. 98–106, 2018.
https://doi.org/10.1007/978-3-319-95450-9_8

for automatically and securely connecting IoT clients to IoT gateways (APs) based on widely-used Wi-Fi technology. The contributions are as follows. First, Wi-Fi IoT clients can automatically connect to an IoT gateway without human intervention. Secondly, for better IoT security, the SSID and password for Wi-Fi connection are randomly generated. Finally, the randomly generated passwords and SSIDs are automatically changed every day or when cracking or attacking is detected.

2 Background Knowledge

This section introduces existing Wi-Fi connection technologies and Wi-Fi security and presents the problems and difficulties in the development of the IoT system platform as a reference for the theoretical basis and system design.

2.1 Wi-Fi Network

Wi-Fi network [5, 12] refers to the two basic service combinations defined in the IEEE 802.11 standard:

- Basic Service Set (BSS): BBS is mainly responsible for all message transmissions in the AP (Access Point) and wireless clients such as laptops, smartphones and IoT end devices in a local area. In IEEE 802.11 standards, IEEE 802.11b, IEEE 802.11a and IEEE 802.11 g are in the 2.4 GHz band while IEEE 802.11 h is in the 5 GHz band.
- Independent BSS (IBSS): IBSS supports peer to pear ad hoc network. Wireless clients directly communicate with each other without assistance of APs.

It is obvious that BSS-enabled Wi-Fi network is suitable for IoT applications. However, to connect to an AP, a Wi-Fi client needs to know the SSID and the password of the AP. It is formidable to connect billions of IoT devices to APs (gateways) by hand.

2.2 Wi-Fi Protected Access

Wi-Fi Protected Access [4], consisting of WPA, WPA2 and WAP3, is the security protocol developed by the Wi-Fi Alliance to secure wireless computer network. WPA and WPA2 are not secure and need to be improved [8]. A new protocol, WPA3, is released in 2018 to address the weak password problem with a 4-way handshake [6], as shown in Fig. 1 and 192-bit encryption.

2.3 Wi-Fi Security Attacks

Once a Wi-Fi network connection is established, it is vulnerable to attack. There are two common kinds of attacks:

- DoS (Denial of service) attacks: In DoS attacks, hackers (the attackers) try to make the system unavailable to normal users by flooding the targeted system with false requests. For example, there are three users and one attacker labeled as hacker in

Fig. 2. The hacker first monitors authentication (Auth) request packets. Once found, he can sends a great deal number of false Auth packets to disrupt network services, leading to no services available among the connected users and no new connection for the unconnected ones.

- Password cracking attacks: In password cracking attacks, hackers try to recover passwords of APs by repeatedly guess the passwords to gain unauthorized access. For example, the hacker in Fig. 2 may obtain the EAP and EAPOL packets in WPA2 and then uses a dictionary attack method to crack the passwords. It works for easy guessed meaningful password or simple password [7]. Currently the security of WEP, WPA and WPA2 are all problematic in Wi-Fi communication.

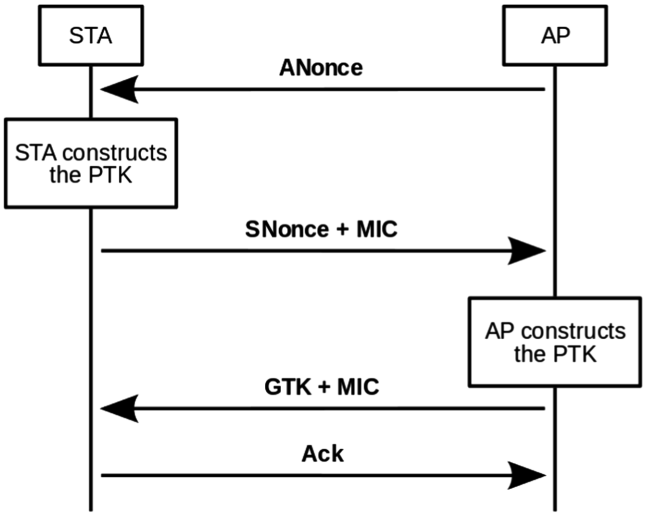

Fig. 1. 4-way handshake

Fig. 2. Denial of service attacks

In [11], Zhang et al. proposed an intrusion prevention method for Wi-Fi clients in DoS attack. A Wi-Fi client is able to differentiate between legitimate and forged frames by using Medium Access Control filtering.

2.4 Wi-Fi Protected Setup

Wi-Fi Protected Setup (WPS) [9] is designed to ease the setup of secure Wi-Fi networks in home and small office environments by Wi-Fi Alliance. There are three supported methods:

- PIN method (PIN-WPS): Users read the PIN of the AP and enter the PIN number to the client device to connect to the AP.
- Push button method (PBC-WPS): Users push the setup buttons of the client device and the AP simultaneously to establish a Wi-Fi network.
- Near-field communication method (NFC-WPS): When a NFC-enabled client is closed to a NFC-enabled AP, PIN code is received and then a secure Wi-Fi network is established.

Neither PIN-WPS nor PBC-WPS are secure [9, 10]. PIN-WPS can be easily cracked in [9] and PBS-WPS can be cracked when the PBC buttons are pressed [10]. A related work in [14] also uses NFC to facilitate Wi-Fi setup. The AP is set in open mode and clients receive the password and AES key of the AP with NFC.

3 Automatic and Secure Connection

There are two cases to be considered in automatic and secure connection in a Wi-Fi network: (A) Pre-shared key auto-connection and (B) Keyless auto-connection. Both methods can be applied in WPA2 or WPA3 for secure connection. We define the following terms to be used in our algorithms:

- P1 is a randomly generated password of the IoT gateway.
- KEY1 is the pre-shared key stored in the IoT gateway.
- EP1 is the encrypted key by some encryption algorithm such as AES [13] and EP1 = AES(P1, KEY1)
- KEY2 is the pre-shared key stored in the IoT client devices and KEY1 = KEY2.
- $\text{Prefix}_{\text{ssid}}$ is the predefined prefix label as part of SSID.

3.1 Pre-shared Key Auto-Connection

In the pre-shared key auto-connection, the IoT gateway (GW) and the IoT End-devices (ED) have the preloaded shared keys, KEY1 and KEY2, where KEY1 = KEY2, respectively. The algorithm is shown in Fig. 3.

It works as follows: (1) GW loads the pre-shared key, KEY1, (S301) and then generates the random password, P1 (S302). (2) In step S303, GW generates the encrypted password EP1 by applying some encryption algorithm (e.g. AES) on P1. (3) GW generates a SSID by concatenating $\text{Prefix}_{\text{ssid}}$, EP1 and date in Step S304. (4) During steps

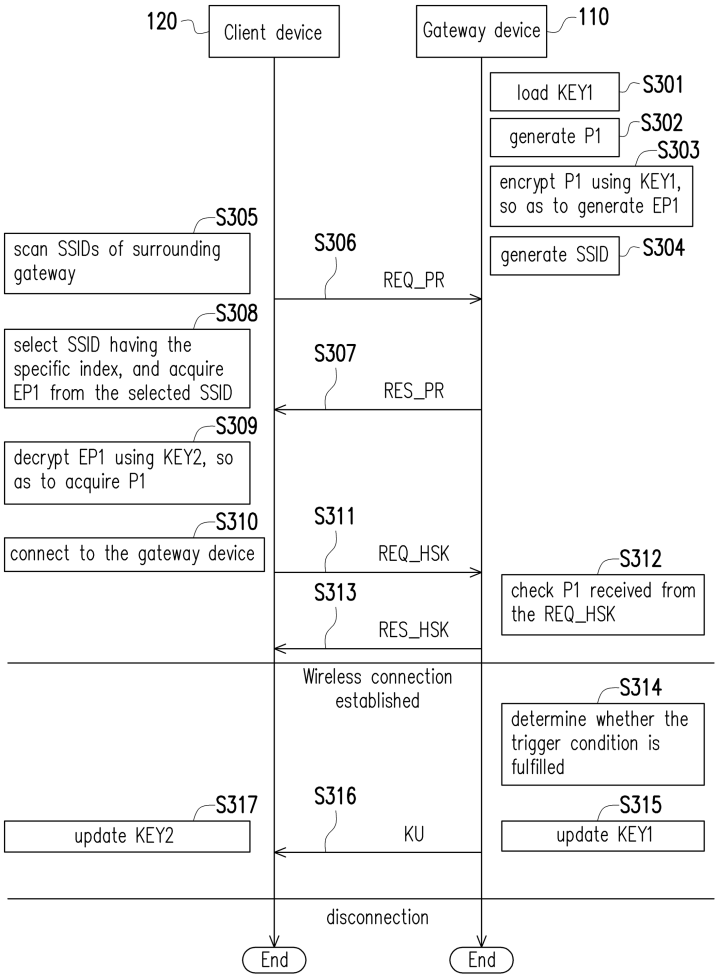

Fig. 3. Pre-shared key secure and automatic connection

305–307, ED sends a probe request (REQ_PR) packet to the surrounding GWs and receives probe response (RES_PR) packets from the responding GWs. (5) In step 308, ED selects the correct GW with identifying label, $\text{Prefix}_{\text{ssid}}$ and retrieves EP1 from SSID. (6) In step S309, ED decrypts EP1 with KEY2 and recovers P1. (7) During steps S310–S313, GW and ED perform 4-way handshake as shown in Fig. 1 to establish Wi-Fi connection. (8) Once connected, GW prepares a new shared key (or backup key), $\text{KEY1}_{\text{backup}}$ and sends it to connected EDs as a backup key, $\text{KEY2}_{\text{backup}}$. (9) Entire pre-shared key auto-connection process is repeated with the new keys (i.e. $\text{KEY1}_{\text{backup}}$ and $\text{KEY2}_{\text{backup}}$) at midnight or when some password cracking attack is detected.

3.2 Keyless Auto-connection

In the keyless auto-connection, GW and EDs do not have the preloaded shared key. Instead, some symmetric encryption/decryption algorithms (called keyless encryption/decryption algorithm) are preloaded in both GWs and EDs. The algorithm is shown in Fig. 4. It works as follows: (1) GW randomly generates the password, P1 and then encrypt P1 into the encrypted password, EP3, by applying keyless encryption algorithm, as shown in step S501 and S502. (2) In step S503, GW generates a SSID by concatenating $\text{Prefix}_{\text{ssid}}$, EP3 and date. (3) ED sends a probe request (REQ_PR) packet to the surrounding GWs and receives probe response (RES_PR) packets from the responding GWs during steps S504–S506. (4) In step S507, ED selects the correct GW

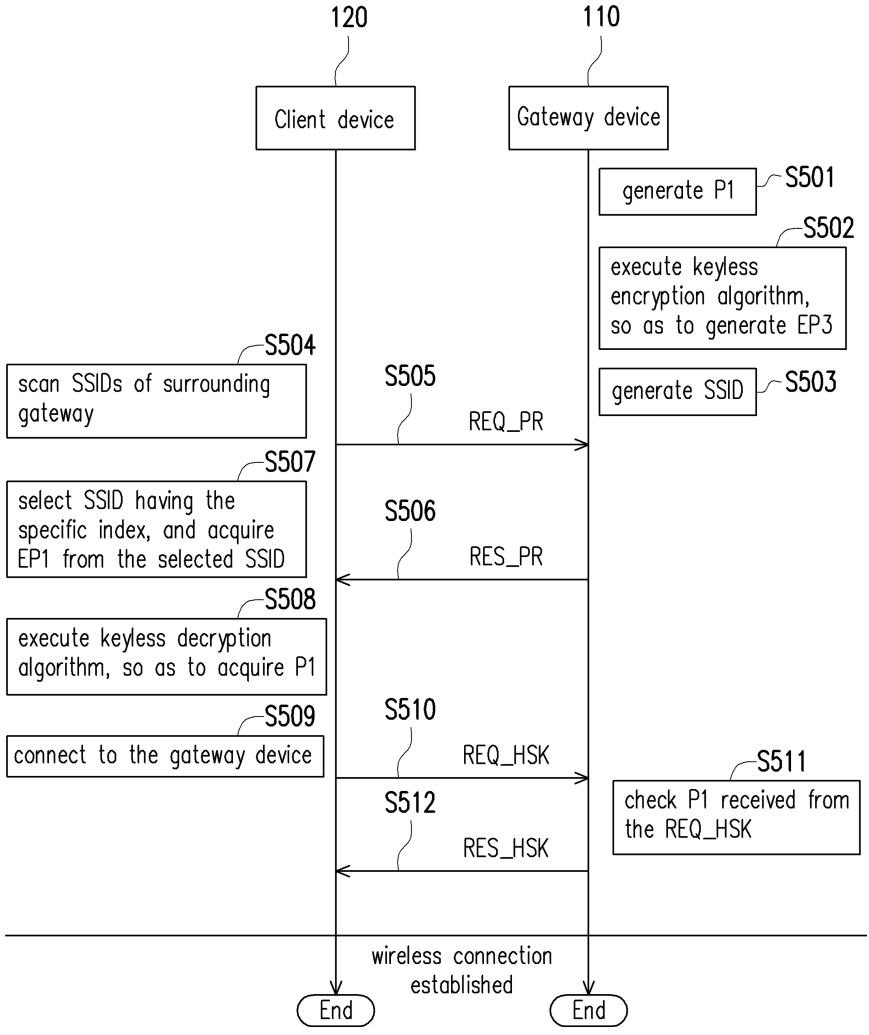

Fig. 4. Keyless secure and automatic connection

with identifying label, $Prefix_{ssid}$ and retrieves EP3 from SSID. (5) ED decrypts EP3 with KEY2 and recovers P1 in step S508. (6) GW and ED perform 4-way handshake to establish Wi-Fi connection. These steps are shown in steps S509– S512. (7) Once connected, GW prepares new shared key and sends it to connected EDs. (8) Entire pre-shared key auto-connection process with the new key is repeated at midnight or when the password cracking attack is detected.

3.3 Wi-Fi Validation Algorithm

The auto-connection algorithms presented in previous subsections can be further secure by applying the validation algorithm, as shown in Fig. 5. The idea is a valid ED can send a challenge request to validate the connected GW and vice versa. Assume all the valid EDs have a backup KEY2, $KEY2_{backup}$, and are connected to the valid GW having a backup KEY1, $KEY1_{backup}$, where $KEY1_{backup} = KEY2_{backup}$, and an invalid

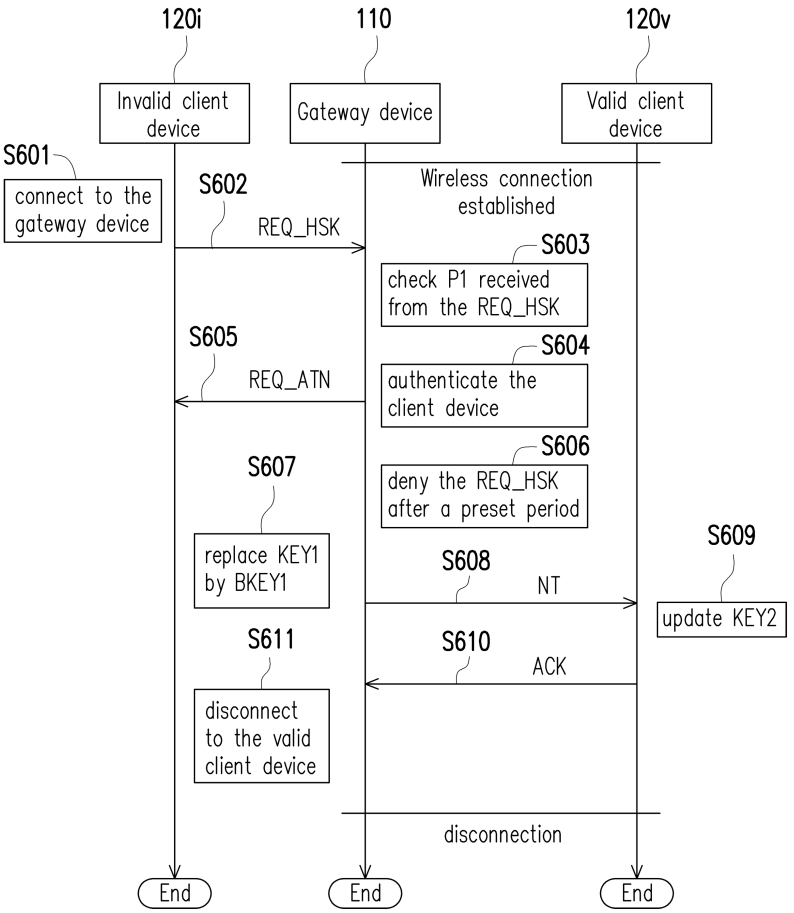

Fig. 5. Validation algorithm

ED cracks the password of the valid GW. The validation algorithm can prevent the invalid client from accessing network. It works as follows: (1) Since the invalid ED cracks the password P1, a 4-way handshake can be applied to connect to the valid GW as denoted in steps S601–S603 and an IP address may be granted from GW. (2) GW may further apply the validation algorithm to challenge the invalid ED in steps S604–S605. If the invalid ED fails to reply with a valid response, GW will disconnect the connection and record the MAC address of the invalid ED and thus no further connection is possible for the invalid ED in step S606. (3) In step S608, GW notifies all valid EDs that the network will be rebooted with the backup KEY for the incoming connection. (4) Once all the valid EDs are notified, GW reboots itself and starts a new auto-connection flow as shown in Fig. 3.

4 Discussions

The experimental results are conducted with a Raspberry Pi 2 as the AP (i.e. GW) and NodeMCUs [15] as the Wi-Fi clients. Four related works (i.e. PIN-WPS, NFC-WPS, PBC-WPS and [14]) are compared with our works with respect to automatic connection. The factors to be evaluated of the comparison table are "conn time" (the connection time required for an ED connecting to a GW), "conn mode" (the connection mode is classified into Easy (i.e. with human intervention) and Auto (without human intervention) connection) and "extra cost". The result is shown in Table 1. Only our methods provide secure and automatic Wi-Fi network connection with no extra hardware cost. Note that NFC-WPS and [14] may provide automatic Wi-Fi connection, however, both approaches require expensive NFC on both GWs and EDs. In addition, it may be not suitable for using NFC if the devices are too heavy to be moved or hard to reach (e.g. hang in high place).

Table 1. Wi-Fi auto or easy connection comparison table

	PIN-WPS	NFC-WPS	PBC-WPS	[14]	Proposed
conn time	60 s	30 s	30 dec	30 s	60 s
conn mode	Easy	Auto	Easy	Auto	Auto
extra cost	No	Yes	Yes	Yes	No

Four related works (i.e. PIN-WPS, NFC-WPS, PBC-WPS and [11]) are compared with our works with respect to Wi-Fi security. The factors to be evaluated of the comparison table are "DoS_{GW} secu" (DoS security on GW), "DoS_{ED} secu" (DoS security on ED) and "WPA2 secu" (better WPA2 security). The result is shown in Table 2. Only our methods provide DoS security for gateways and enhance WPA2 security.

Table 2. Wi-Fi security comparison table

	PIN-WPS	NFC-WPS	PBC-WPS	[11]	Proposed
DoS_{GW} secu	No	No	No	No	Yes
DoS_{ED} secu	No	No	No	Yes	No
WPA2 secu	No	No	No	No	Yes

5 Conclusion

We propose automatic and secure Wi-Fi connection mechanisms for IoT applications in which IoT gateways and end devices can automatically establish secure Wi-Fi connection without human intervention and with almost zero configuration cost. In addition, our works enhance WPA2 security by introducing randomly generated SSIDs and passwords. We also propose a validation algorithm to prevent password cracking and DoS attacks. To the best of our knowledge, our design is the only solution achieving secure and automatic Wi-Fi connection with no extra hardware cost.

References

1. Gartner IoT market report, February 2017. https://www.gartner.com/newsroom/id/3598917. Accessed 3 Apr 2018
2. Gartner IoT security report, March 2018. https://www.gartner.com/newsroom/id/3869181. Accessed 3 Apr 2018
3. Evans, D.: The Internet of Things: how the next evolution of the internet is changing everything? (2011). https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf
4. Wi-Fi Protected Access (WPA). https://en.wikipedia.org/wiki/Wi-Fi_Protected_Access. Accessed 3 Apr 2018
5. Wi-Fi. https://en.wikipedia.org/wiki/Wi-Fi. Accessed 3 Apr 2018
6. 4-way handshake. https://en.wikipedia.org/wiki/IEEE_802.11i-2004. Accessed 3 Apr 2018
7. KRACK. https://en.wikipedia.org/wiki/KRACK. Accessed 3 Apr 2018
8. Liu, X.: The security analysis on Wi-Fi communication, Master thesis, Shanghai Jiaotong University (2009)
9. Wi-Fi Protected Setup. https://en.wikipedia.org/wiki/Wi-Fi_Protected_Setup. Accessed 3 Apr 2018
10. Chang, J.: Secure and automatic connection for IoT end-devices and gateways based on WiFi Technology, Master thesis, Tatung University (2015)
11. Zhang, Y., Sampalli, S.: Client-based intrusion prevention system for 802.11 Wireless LANs. In: IEEE Wireless and Mobile Computing, Networking and Communications, pp. 100–105 (2010)
12. Service set (802.11 network). https://en.wikipedia.org/wiki/Service_set_(802.11_network). Accessed 3 Apr 2018
13. Advanced encryption standard. https://en.wikipedia.org/wiki/Advanced_Encryption_Standard. Accessed 3 Apr 2018
14. Jie, M.A., Jin-long, E.: WiFi transmission connection scheme based on near field communication. Comput. Eng. **39**(6), 1–6 (2013)
15. NodeMCU. https://en.wikipedia.org/wiki/NodeMCU. Accessed 3 Apr 2018