Design of efficient and secure authentication schemes in IEEE 802.11ah networks

Zhang, Lyuye

2018

Zhang, L. (2018). Design of efficient and secure authentication schemes in IEEE 802.11ah networks. Master's thesis, Nanyang Technological University, Singapore.

<http://hdl.handle.net/10356/75846>

<https://doi.org/10.32657/10356/75846>

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

**DESIGN OF EFFICIENT AND SECURE
AUTHENTICATION SCHEMES IN IEEE 802.11ah
NETWORKS**

ZHANG LYUYE

**SCHOOL OF ELECTRICAL AND ELECTRONIC
ENGINEERING**

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Master of Engineering

Acknowledgements

Thanks to the help and support from several people, I got the chance to finish this thesis.

First of all, I would like to thank Professor Ma Maode for being an extraordinary supervisor and giving valuable advice and guidance during the thesis process.

Also, I would like to thank my colleague and friend, Miss Qiu, who helped me learn background and offer precious advice on my works.

There are other colleagues from EEE and friends from China as well, who gave me a lot of help and assistance.

Finally, I do appreciate the support and encouragement from my parents and other family members. Without their support, I would never finish the works.

Table of Contents

Abstract	5
List of Figures.....	6
List of Tables.....	6
Chapter 1. Introduction	8
1.1 Background	8
1.2 Motivations	10
1.3 Objectives	11
1.4 Major Contribution	12
1.5 Organization	12
Chapter 2. Literature Review	13
2.1 IEEE 802.11ah	13
2.2 IEEE 802.11i and 802.11ai	15
2.3 IEEE 802.11ai	19
2.4 Key Reinstallation Attack on IEEE 802.11 Standard Family	26
2.4.1 KRA Review	26
2.4.2 KRA on FILS	28
2.5 Security Evaluation Tool Backgrounds	30
2.5.1 BAN Logic	30
2.5.2 Scyther Tool	31
Chapter 3. FKR Mechanism.....	33
3.1 Motivations	33
3.2 The Proposed FKR Scheme	34
3.3 Security Evaluation	36
3.4 Performance Evaluation	44
3.5 Summary	48
Chapter 4. SEF Mechanism.....	49
4.1 Motivations	49

4.2	SEF Overview	49
4.3	Details of SEF	50
4.3	Security Evaluation	53
4.4	Performance Evaluation	60
4.5	Summary	63
Chapter 5.	Conclusions	64
5.1	Conclusions	64
	Recommendations for Further Research	64
	Author's Publication	66
	Bibliography	67

Abstract

The increasing deployment of smart devices has proved that Internet of Things (IoT) is a new trend for the next era. The IEEE 802.11ah, a specification belonging to the 802.11 Wireless Local Area Network (WLAN) protocol family, has been recently released to support the long-range, low-power and low-rate wireless communications among smart devices used in IoT systems. However, security requirements of the energy-constrained devices have plenty of features different from the traditional line-powered devices. Due to the relatively high overhead and latency, link setup protocol, such as IEEE 802.11i, does not meet the requirements of IoT networks. It requires that the lightweight protocol has to support low-power and low-latency features of IoT devices. To meet these demands, an upcoming solution proposed by IEEE 802.11 group, which was specified in IEEE 802.11ai. It is called the Fast Initial Link Setup (FILS), a brand-new method aiming at establishing fast and secure links among devices to be applied to 802.11ah networks. Based on the FILS, this thesis has proposed two enhancement proposals tailored for the 802.11ah networks regarding the authentication process in the link setup procedures to improve security and efficiency. The first proposal is to improve the efficiency of link setup procedure, and the second proposal is designed to enhance the security of FILS without degrading its performance.

Keywords—802.11ah; 802.11ai; Fast Initial Link Setup

List of Figures

Figure 1 Coverage of IEEE Standard.....	9
Figure 2 Transmission Range	8
Figure 3 Message Flow of 802.11i Authentication Process in 5 Stages.....	16
Figure 4 Message Flow and Operations of FLAP.....	17
Figure 5 FILS Typical Shared Key Authentication Operations	22
Figure 6 802.11ai Key Hierarchy	21
Figure 7 FILS Public Key Authentication Operations	23
Figure 8 ERP Message Flow and Operations	28
Figure 9 Typical KRA Attacking Procedure.....	29
Figure 10 FKR Message Flow and Operations by STA and AP	34
Figure 11 FKR Element Format	35
Figure 12 Scyther Tool Verification Result of FILS	40
Figure 13 The Retransmission Scheme If First Message Is Lost	42
Figure 14 The Retransmission Scheme If Second Message Is Lost.....	42
Figure 15 Flowchart of Simulations	47
Figure 16 Average Time Delay and Energy Consumption against Unknown Attacks	48
Figure 17 SEF Message Flow and operations of STA, AP, and AS	50
Figure 18 STA State Machine	51
Figure 19 AP State Machine.....	52
Figure 20 Scyther Tool Verification Result of SEF.....	58
Figure 21 Average Time Delay and Average Energy Consumption against Unknown Attacks	63

List of Tables

Table 1 Detailed Information of Encryption Keys.....	16
Table 2 802.11 Standards Comparison	17
Table 3 Message Comparison.....	44
Table 4 Length of Parameters	45
Table 5 Time Delay and Energy Consumption of Algorithms	45

Table 6 Time Delay and Energy Consumption Comparison	46
Table 7 Operations Comparison	61
Table 8 Length of Parameters	62
Table 9 Performance Comparison	62

Chapter 1. Introduction

1.1 Background

This thesis discusses the performance and security concerns about the IEEE 802.11ah specification based on the premise that IEEE 802.11ai specification is applied. Two possible solutions are proposed to address problems in 802.11ah networks based on the Fast Initial Link Setup (FILS) scheme. Besides, the evaluations of these two schemes are given to ensure that both of them are secure and efficiency.

1.1.1 IEEE 802.11ah

This decade has witnessed the rapid progress of the Internet of Things (IoT) industry. For a typical IoT application, while a large number of devices are deployed in an area, Wireless Sensor Network (WSN) is built to gather information collected by nodes. After the collection, data gathered is sent to servers via the internet. Thus there are two types of communications roughly distinguished by the target. The first is inter-device communications, it includes the communications among all end devices and border router who connect the outer world. The second is the uplink communications. It is simply in charge of transferring data to servers. Although both of them are susceptible to attacks, uplink communications are out of the scope of this thesis, and thus not discussed here. Only inter-devices communications are included. In this network, communications can be operated with WLAN, Bluetooth, ZigBee, or Sigfox and so forth. Although they have different topologies and protocols, they mostly work on the lowest two layers of the internet structure, namely, Physics layer and Media Access Control (MAC)

Figure 1 Coverage of IEEE Standard

layer. For example, IEEE 802.11, the standard family of WLAN works on Physics and MAC layer. Since they only work on two layers, they are not as susceptible to attacks as the internet. But it is also extremely dangerous if the network is compromised.

The attackers are able to use fake routers, access points, or nodes to launch DoS attack or intercept packages, or launch Man in the Middle (MitM) attack. In consequence, the data maybe disclosed. The system maybe paralyzed. The user may receive fake data. Hence, the security of

inter-devices network is paramount important. However, IoT scenario based protocol needs its own security mechanism, because the legacy security defining protocol 802.11i was different regarding the resources, data rate, and so forth. For example, the IEEE 802.11ah implemented for IoT scenario was released in last year, which did not have a well-defined security mechanism. It operates on sub-1-GHz with relatively low data rate as other IoT standards. Comparing to previous standards, supporting long-range communication is its main advantage because the sub-1-GHz has lower obstruction losses than those of the systems with 2.4-5GHz of the common wireless frequency band. It is claimed to be over 1km. To demonstrate the difference clearly, there are comparisons among several standard coverages in [1]. To make it more vivid, Figure 2 gives the coverages of several IEEE standards. It clearly illustrates that the previous standards whose frequency is higher than 1GHz are applicable to high speed indoor scenarios. But the IEEE 802.11ah with its long range and low obstruction losses is more applicable to outdoor scenarios, such as sensor networks, home automation. Normally in this kind of applications, the number of devices associated is way beyond the daily applications used nowadays. Thus the new version of security mechanism is imperative.

There are several new features introduced by the 802.11ah to IoT scenarios, such as

Figure 2 Transmission Structure

Restricted Access

Window (RAW), Associated ID (AID) hierarchy, Target Wake Time (TWT). These mechanisms are introduced to reduce contentions among thousands of nodes, to optimize the management, to extend the battery life and so forth. Thus this new standard is basically for scenarios different from the traditional WLAN applications. Within the new scenarios, authentication requirements also change, which lead to the optimization for authentication procedures. There is another new standard, IEEE 802.11ai can be applied.

It is important that TWT has to be introduced before the following content. According to [2], TWT is that AP wake up stations (STA) at some predefined amount of time to make sure these STA's wake-up timings are temporarily spread out. This move is to limit the access time of each STA and decrease the contentions at peak hour. Therefore, STA has to wake up from time to time for several times after first association with AP to transmit data intermittently. The process is illustrated in Figure 1. After each beacon broadcasted by AP, STA has their chances

to transmit if they are scheduled. And between each transmission interval, STAs sleep to save energy.

1.1.2 IEEE 802.11ai

IEEE 802.11ai [3], released in the end of 2016, was an amendment to the legacy 802.11 standard. Its main contribution is to specify FILS and achieve less than 100 ms link setup. As mentioned in [4], the legacy standard are facing some urgent problems, such as too much signaling overhead, slow handoff mechanism between APs, and too many probes used resulting in exhaustion of resources and so forth. To solve these problems, IEEE 802.11ai improves the link setup process by defining steps to complete authentication and association with a four-way handshake and IP address assignment in just two round-trip messages. And more efficient and frequent beacon transmissions allow the increasingly rapid network detection. By defining Neighbor AP advertisement and FILS parameter advertisement in beacons and probe responses, the faster AP handoff is achieved. The information in Neighbor AP advertisement including basic service set identification (BSSID), service set identification (SSID), channel information, and target beacon transmit time (TBTT) offset of neighboring APs significantly reduce the device scanning time. In practice, Qualcomm Incorporated has prototyped elements of 802.11ai on handset devices. It was observed that the time of handset connection to a new AP with different SSID and subnet ID is reduced from 8s to 0.5s.

1.2 Motivations

Now that the IEEE 802.11ah does not specify the authentication and association step, and IEEE 802.11ai does provide a new thought. In other words, because the IEEE 802.11ah needs a time-saving and energy-saving way to perform authentication and association, the newly-proposed mechanism, FILS authentication, can be applied to it. The FILS scheme can be properly applied to the IEEE 802.11ah specified WLANs to decrease the latency and simplify the process of the link setup.

Then the next problem solved in this thesis is how to apply the FILS to the 802.11ah standard. Although FILS is designed to be definitely superior to the predecessors, it is not developed for one specific protocol, instead, all new WLAN standards may utilize it, which include the standard with high-speed, resource-abundant devices. Thus for resource-constrained networks like 802.11ah, it is to be improved. FILS includes several steps where authentication and association is one of them. In this thesis, only the authentication and association part is discussed.

In 802.11ah, there are some aspects may be different from other standards. As mentioned before, in [4], it is said that the AP and STA are able to complete the authentication and association with a four-way handshake and IP address assignment in two round trip messages. However, after the link setup and data transferred, there will be more data for nodes to transfer after the sleeping time. Thus STAs normally have to reestablish connections with the AP over

and over again. And in each connection restoration, nodes, or STAs, may contend for the transmission interval every time because of the random sleeping time. But with TWT, the STAs wake up at the scheduled moment, there will not be much contention. The transmission process is supposed to be completed directly without the full procedure as mentioned before due to the redundancy of the original authentication and association operations. Therefore, there may be some possibilities that the connection restoration step can be simplified. To simplify the connection restoration procedure, the authentication and association part does not need to be executed for several times.

Another thought is derived from the Key Reinstallation Attack (KRA), which is a new type of attack proposed in [5]. The KRA is able to jeopardize the security of 802.11 authentication protocols. Many authentication protocols belonging to 802.11 standard family are susceptible to the KRA, including the FILS authentication. Thus this thesis also proposes the second proposal to defend against KRA.

1.3 Objectives

With the thoughts in previous section, new authentication and association mechanisms are designed to optimize the performance for 802.11ah low power scenario and solve the security problem concerning KRA. Besides that, it is necessary to compare them two in terms of performance. In this authentication and association operations, there must be several requirements listed below,

- Mutual Authentication

Once the authentication process begins to take place, they firstly have to confirm each other's identities and the possession of the previous session key, PTK, and long term session key, PMK.

- Exchange input keying materials and confirm them

Fresh keying materials have to be generated every time the connection establishes. Both parties must believe their freshness and validity.

- Derivation and confirmation of new session key

With input keying materials received, both parties are able to derive PTK, and use part of it to encrypt the data in this session. Also, the confirmation of each other's possessions is necessary.

- Be more efficient and economical

This new mechanism needs to be more efficient and economical to gain an edge over the original one.

- Against malicious attacks

New mechanism is not supposed to be vulnerable and prone to attacks. It has ability to defend against several typical malicious attacks as analyzed in following contents.

1.4 Major Contribution

In this thesis, improvements have been made to FILS authentication protocols to be adapted to 802.11ah networks

- A new scheme based on FILS authentication protocol is proposed to improve the performance of all subsequent rounds of authentication. With this scheme, energy and time consumption can be decreased for resource-constrained devices networks. In the meantime, the security of this new scheme is not degraded.
- To avoid the KRA, some changes have been made to FILS authentication protocol. After these necessary changes, the enhanced version of FILS is still able to be against other typical attacks as the original one, and the performance of it will not be degraded.

1.5 Organization

The rest of this thesis is organized as follows: In Chapter 2, Literature review presents previous papers concerning the related topics and the comments and simple evaluation of those papers. In Chapter 3, an enhanced mechanism of FILS is proposed for 802.11ah. In Chapter 4, in order to deal with the KRA, a new scheme is proposed and evaluated. In the last chapter, the conclusions and recommendations for future works of related topics are discussed.

Chapter 2. Literature Review

This chapter gives a full view of security contents in the 802.11ah specification as well as FILS. Unfortunately, the detailed authentication process is not much mentioned in the 802.11ah. Hitherto, the authentication schemes defined in 802.11 standard family are presented, especially the new 802.11ai. Additionally, the advantages and problems in each standard are also discussed and analyzed with the possible solutions given by researchers. Finally, the KRA scheme is introduced and the typical attack mechanism on FILS follows.

2.1 IEEE 802.11ah

Since the IEEE 802.11ah was a specification released at the end of 2016, it was designed to be backward-compatible to the legacy IEEE 802.11 system. Although several newly proposed mechanisms were introduced, such as RAW, AID hierarchy, and TWT, it has to follow the same access procedure as previous specification to maintain backward compatibility. The classic access procedure is roughly divided into five parts according to [7].

- Discovery

AP uses messages called beacon to advertise the presence of it to STA or STA uses probe message to find AP and waits for probe response. With messages like beacon and probe response, STA is able to identify the AP with Wireless Local Area Network (WLAN) it wishes to communicate. In the messages exchanged, STA decides which cipher suite and authentication mechanism it intends to use in the beacon or probe response

- Authentication and association

During this phase, STA and AP have to prove their identities to each other. AP may verify STA's identity by forwarding the message from STA to Authentication Server (AS) and block all non-authentication traffic between the STA and AS.

- Key generation and distribution

AP and STA perform several operations to calculate key sets with input keying materials they just received in authentication phase, and install them. Frames are only exchanged between STA and AP.

- Protected data transfer

Data begins to be transmitted in frame between the STA and the end station, such as a server through the AP. And data is protected by the cipher suite they negotiated in the first phase, discovery. However, the security does not provide end-to-end, but STA to AP only.

- Connection termination

After some frames exchanged, AP and STA terminate the connection and quit the association state to the original state.

Nodes intending to join the network have to go through these steps to establish connections. Upon successful association, the nodes are connected to the network. Afterwards, according to the specification [6], all messages exchanged between AP and this node are protected by Counter Mode Cipher Block Chaining Message Authentication Code Protocol (CCMP) with Advanced Encryption Standard (AES) defined in 802.11i [7]. But this only specifies the fourth step, protected data transfer. The security chapter only provides options to encrypt data. When it comes to authentication process, the 802.11ah specification only specifies the Authentication Control (AC). For STAs accessing the network with APs, they have to first establish associations with AP then transmit data. AC does entail authentication and association, but it is only in charge of the control of which STA is able to perform authentication and association, while the RAW is in charge of the control of which STA is able to transmit data. Both AC and RAW belong to Access Control. However, the exact procedure of how to perform authentication and association is not the focus of AC. Thus for the main topic in this thesis, how to perform authentication and association, is not much mentioned in the specification of 802.11ah.

To avoid too much contentions, there are two ways to perform AC according to [6], the Centralized Authentication Control (CAC) [8-9] and Distributed Authentication Control (DAC). Those two mechanisms aim at allowing STAs to perform authentication with as little contentions as possible. CAC allows the AP to regulate the number of STAs who are trying to establish links. To achieve it, the AP uses beacons, special management frames sent with periodical Beacon Interval (BI), to broadcast the network parameters and claim the presence of the AP to STAs within the coverage. In the frame, AP includes a parameter, called Authentication Control Threshold (ACT), which is picked from interval (1-1023) according to implementation-dependent algorithms. If the STAs wish to associate with this AP, they will generate random numbers from the interval (1-1022) and compare with ACT. If this number is less than ACT, the STA is then allowed to transmit the authentication request during the current BI. Otherwise, the STA has to wait for the next BI and proceed this procedure again. Therefore, ACT is the key to control the number or percentage of STAs who wish to connect. The algorithm of deciding ACT adaptively is a widely discussed topic. The author of [9] designed a new contention-free channel access method, which claims to have superior performance than traditional methods, such as Polling-based Point Coordination Function (PCF) and Hybrid Controlled Channel Access (HCCA). In [10], the practical situation of 100 STAs is presented and analyzed. The author of [11] proposed a mathematical model of the joining process and suggest to use the model to minimize the time needed for a group of STAs to join the network in a new channel. The author of [12] studied the handover process of Wi-Fi networks and proposed the authentication process and data transmission to the new AP can be accelerated through a temporary tunnel with the previous AP.

Although in 802.11ah specification, Access Control and data encryption are well-defined, how AP and STA perform the second step, authentication and association, is not well covered, which means there is more to be explored and improved. Thus the following contents will demonstrate the unreliability of legacy standard and the necessity of introducing the new amendment, the 802.11ai.

2.2 IEEE 802.11i and 802.11ai

The IEEE 802.11i [7] released in 2004 gave a complete definition of security mechanism for wireless network. It is also referred as WPA2 or Robust Security Network (RSN), which takes advantage of AES block cipher. The IEEE 802.11i authentication process as illustrated in Figure 4 is briefly introduced as follows. Better authentication and confidentiality is provided in the

Figure 1 Message Flow of 802.11i Authentication Process in 5 Stages

802.11 networks by Robust Security Network Association (RSNA) than WEP. RSNA is formed after the 4-Way Handshake is performed. Three entities are included, called the authentication

Figure 2 Message Flow and Operations of FLAP

server (AS), the authenticator, and the supplicant. For example, the stations are supplicants, and the APs are authenticators. An authentication server can normally be a Remote Authentication Dial-In User Service (RADIUS) [13] server. Basically, the supplicant and the authenticator complete the authentication process by verifying the identity of each other. After the verification, some information exchanged during authentication process is used to generate common keys for data transmissions. So long as the authenticator shares a physically secure link with AS, AS could be implemented either as separate server or inside AP. No matter which way is used, the security of the wire-based link to AS is out of the scope of this thesis, which does not mean that the AS is not involved. As introduced in [14], after PMK is derived from previous conversations, 4-Way Handshake is performed to establish a RSNA. Fresh PTK is generated for subsequent session. And in case of the multicast scenarios, the Group Temporal Key (GTK) [15] is transferred from the authenticator to the supplicants. After 4-Way Handshake, a secure and stable communication channel is established between the supplicant and the authenticator. The shared PTK and GTK are going to be used to encrypt information exchanged between the supplicant and the authenticator. Besides that, 4-Way Handshake maybe used for several times using the same PMK. The 4-Way Handshake message exchange can be found in Figure 3 from [24]. The details of the protocol are not going to be discussed here, because they are not the

focus in the thesis. Figure 4 is to show the complexity and cumbersomeness of the original protocol.

As a legacy standard, 802.11i has some vulnerabilities. The typical possible attack is DoS attack. It is mentioned and attempted to be solved in [16-22]. Unlike the 2, 3, 4 messages, the first message is not protected by MIC. The attacker is able to flood the STA's memory with new message 1 including different nonce. For stations, receiving and processing every messages is obligate. Thus the attackers pretend to be the AP and send a large number of messages to make stations believe that the messages are retransmissions. Having received too many messages, the stations' memory is exhausted and the legitimate message maybe ignored so that the authentication is blocked. [14] proposed several solutions to solve it: additional Temporary PTK (TPTK), random drop queue, PMKID and so on. [19] proposed the PMK encrypted version. And [23] put forward the dynamical adjusting strategy. Furthermore, Zha in [17] presented two plausible solutions: TPTK calculated with single nonce to protect first message and supplicant's active protection.

Although there are dozens of proposals to solve problems as mentioned above, none of them is able to change the essence or nature of the standard. The increasing amount of devices and demand for mobility have challenged the legacy WLAN protocols. With WLAN protocols applied to various application scenarios, how fast the connection setup can be becomes a new criterion for highly mobile devices scenario. To achieve high-speed and seamless handover among plenty of APs, the legacy 802.11i authentication mechanism has to evolve to a new fast authentication mechanism, such as FLAP in [24] and FILS in [3], to adapt to highly mobile scenario. The trend of increasing link setup speed is obviously imperative. Therefore, the fast link setup mechanism was proposed years ago within the same framework of 802.11i to maintain the backward compatibility. [24] has proposed an authentication protocol called FLAP claimed to be an efficient initial access authentication protocol. It was designed to simplify the whole authentication process including the Extensible Authentication Protocol (EAP). It reduced the legacy authentication steps from the practical 13 round trip messages to 3 round trip messages, and drastically reduce the time consumption. All the conclusion was simulated and verified with ns-3. It is claimed that the EAP-TLS authentication takes 206.253ms, whereas FLAP takes only 13.884ms with 94.7 percent reduction. The figures below can clearly illustrate the enormous progress made by FLAP. Having eliminating those flaws remained in 802.11i, the new FLAP was proposed in 2014 illustrated in Figure 3. Since the first message is for AP discovery, it is omitted in following discussion. Starting from the Authentication Request sent by STA, FLAP retains the basic part of EAP authentication and send the SNonce and its own identity proof, 'F' in Figure 4, to AP in the meantime. With the AP relaying, AS verify the identity of STA and calculate the PMK with the counter value, User-ID, and AS-ID. After AP receives the confirmation and PMK from AS, it generates its own ANonce and calculates the PTK following the same way as 802.11i, and send ANonce with acknowledgement to STA. These four steps are much similar to the classic 4-Way Handshake with the only difference is that first two, not one,

messages are open and unprotected in FLAP. Besides the drastic simplification of 802.11i, the author also mentioned that the higher layer operations, such as IP distribution, could be embedded in the last two confirmation messages instead of using DHCP exclusively. From my angle, FLAP did make huge progress in authentication simplification. However, the security analysis is insufficient. The only security problem mentioned in the paper is the replay attack, which can be countered by the counter value, 't' used in first four messages. The first two unprotected messages may face the threats from DoS attacks as the legacy 4-Way Handshake.

FLAP [24] was an ingenious and advancing proposal in 2014, because when 802.11ai [3] was released in the end of 2016, FILS, proposed in 802.11ai, shared a lot in common with FLAP as illustrated in Figure 5. The first is the message framework, including the message sequence, parallel operations. The second is the embedded higher layer operations. However, FILS made additional improvements. Firstly, it has another authentication method, called public key authentication using Diffie-Hellman public key exchange mechanism [25] to support Perfect Forward Secrecy (PFS). Secondly, AS does not help calculate PMK, instead AP generates it by itself. Thirdly, the acknowledgement messages use different attributes and algorithm, such as digital signature.

The security of 4-Way Handshake in 802.11i was widely discussed, but FLAP or FILS evolving from it have not caught enough attentions so far. Due to the similar framework they share with 4-Way Handshake, they are also prone to malicious attacks. The next research topic could be focused on the security of FILS in various scenarios.

2.3 IEEE 802.11ai

2.3.1 Review of FILS Authentication

According to the specification of the IEEE 802.11ai standard, the authentication and association phase in the FILS is called FILS Authentication, which has two ways to perform the authentication, shared key and public key. A STA can use one of the two options to associate with a FILS-capable AP depending on the system status. If a STA shares either a valid re-authentication Root Key (rRK) with a Trusted Third Party (TTP) or a PMK cached from the previous authentication connection, it will perform the shared key authentication. If a STA does not share anything with the AP, it will perform the public key authentication. By either way, after exchanging two frames of the authentication information, the STA and the AP can derive a shared secret key which is used to generate PTK, which will be confirmed in next two frames. Thus, the whole process includes four frames of control information exchanged, which is explained as follows.

It is also necessary to give the key hierarchy of mains key discussed about in this thesis in Figure 5 and Table 1. From the root key rRK, STA and AS are able to derive re-authentication Master Session Key (rMSK) for further PMK derivation and rIK for integrity protection. With

rMSK, AP and STA are able to derive PMK as long-term session key. Furthermore, PMK is used to derive PTK as short-term session key and GTK as short-term broadcast key. All three IEEE 802.11 standards are compared in Table 2.

Table 1 Detailed Information of Encryption Keys

Properties	rRK	rMSK	rIK	PMK	PTK	GTK
Full name	re-authentication Root Key	re-authentication Master Session Key	re-authentication Integrity Key	Pairwise Master Key	Pairwise Transient Key	Group Temporal Key
Derivation	Pre-shared by STA and AS	Generated based on rRK, cryptographically separated from rIK by STA and AS	Generated based on rRK, cryptographically separated from rMSK by STA and AS	Generated based on rMSK by STA and AP	Generated based on PMK by STA and AP	Generated based on PMK by AP
Purpose	Serve as parent key	Generated independently to ensure the secrecy and serve as parent key of session keys	Ensure the integrity of the first two messages, used as STA identifier.	Long term session key, used to derive short term session keys	Short term session key, used to be truncated into several multi-functional encryption keys	Short term session key, used to encrypt broadcast messages

2.3.1.1 Shared Key Authentication

The shared key authentication is fully specified in 802.11ai standard [3], where there are three participants in the authentication process, which are a STA, an AP and a third party authenticator to support the shared key authentication. In this thesis, our discussion focuses on the operations of the AP and the STA in the shared key authentication process, which is illustrated in Figure 6.

When a STA chooses an AP to initiate the authentication, it will generate an initiation message with a nonce and decide to use Pairwise Master Key Security Association (PMKSA) or not. If the PFS is desired, the ephemeral private and public key pair will be generated. And if the STA attempts to initiate the Extensible Authentication Protocol-Re-authentication Protocol (EAP-RP) as described in [26], it will construct an EAP-Initiate/Re-auth message. After the reception of authentication message by the AP, it will check the corresponding fields to see whether the features mentioned above are enabled and will perform corresponding operations. If the STA attempts to use the EAP-RP, the AP shall forward the message to the authenticator. In the case of successful authentication by the authenticator, it returns the associated EAP-RP rMSK with the EAP-Finish/Re-auth message. The AP then can reply the authentication message

to the STA. Once the STA has validated it, it shall perform key derivation and key confirmation. If not, it may attempt to use the alternative authentication approach, which is the public key authentication as described above for another attempt of the authentication.

Table 2 802.11 Standards Comparison

Properties	802.11i	802.11ai	802.11ah
Published year	2004	2016	2017
Application scenario	Used for security issues specification including authentication in most WLAN standards	Used for link setup procedure specification including authentication in future WLAN standards	Used for sub-1-GHz communications WLAN based IoT applications.
Security level	Legacy security specification	Newly proposed security solutions	Security solutions not included

Figure 5 802.11ai Key Hierarchy

2.3.1.2 Public Key Authentication

The public key authentication works based on the Diffie-Hellman exchange as illustrated in Figure 7. The AP and the STA exchange the ephemeral public keys and calculate the common secret with ephemeral private keys. Moreover, they are capable of verifying each other's public key during the process. Due to ephemeral key pairs generated, the PFS for the long term key is innate. The other difference lies on the derivation of keys. For shared key authentication, PFS is optional, and PFS is compulsory for public key authentication. Thus the Diffie-Hellman shared secret (DHss) is included to the calculation of PMK for the public key authentication depending

Figure 6 FILS Typical Shared Key Authentication Operations

on whether PFS is desired or not.

2.3.1.3 Key Derivation and Confirmation

The last two steps are the key derivation and confirmation. Before the key confirmation, the AP and the STA should derive the common key set respectively. In key derivation step, both the AP and the STA use the keying input materials received from the previous steps, including AP nonce, STA nonce and rMSK, to derive PMK as follows:

$$\text{PMK} = \text{HMAC-Hash}(\text{SNonce} \parallel \text{ANonce}, \text{rMSK} [\parallel \text{DHss}])$$

PTK is derived as follows:

Figure 3 Public Key Authentication

$\text{FILS-Key-Data} = \text{PRF-X}(\text{PMK}, \text{"FILS PTK Derivation"}, \text{SPA} \parallel \text{AA} \parallel \text{SNonce} \parallel \text{ANonce})$

Once the keys are derived, key confirmation is performed. For the shared key authentication, the STA constructs a (Re)Association Request message with Key-auth field as the important element by using the HMAC mode of the negotiated hash function with a key of Integrity Check Key (ICK) on a concatenation of the STA's nonce, the AP's nonce, the STA's MAC address, the AP's BSSID, and conditionally both STA's and AP's public Diffie-Hellman value as follows:

$\text{Key-Auth} = \text{HMAC-Hash}(\text{ICK}, \text{SNonce} \parallel \text{ANonce} \parallel \text{STA-MAC} \parallel \text{AP-BSSID} [\parallel \text{gSTA} \parallel \text{gAP}])$

For the public key authentication, the Key-Auth field of the FILS Key Confirmation component is a digital signature using the similar input keying materials as the shared key authentication, except the ICK as follows:

$\text{Key-Auth} = \text{Sig-STA}(\text{gSTA} \parallel \text{gAP} \parallel \text{SNonce} \parallel \text{ANonce} \parallel \text{STA-MAC} \parallel \text{AP-BSSID})$

The (Re)Association Request message is encrypted by the Authenticated Encryption with the Associated Data (AEAD) algorithm introduced in the specification.

The AP calculates the same value with its own input keying materials. After generating its own Key-Auth value, the AP decrypts the message with the same AEAD to verify the value. Once it is successfully verified, the AP constructs the similar (Re)Association Response message with the Key Delivery element indicating the current GTK and the Key Receive Sequence Counter(RSC) and sends to the STA. Once the STA verifies the value successfully. They both

will trust the common keys derived and authenticate each other. The STA can install GTK and RSC keys.

2.3.2 FILS Discussion and Solutions

Since IEEE 802.11ai was released in late 2016, it was rarely discussed or analyzed systematically before. In most cases, it was mentioned or briefly discussed in research survey papers. As [6] mentioned before briefly introduced several new expected features of FILS in 2015, [27-28] also pointed out that FILS in IEEE 802.11ai is meant to solve some upcoming challenges.

In [27], it is indicated that WLAN faces two main challenges. The first is increasing deployment of access points to cover more new areas and provide higher transmission rates results in AP dense scenario. And the second is users' need for higher throughput is significantly increasing because of the evolution of multimedia requirements, for example, real-time high-definition audio and video contents. As for these two challenges, FILS does provide a fine solution to them. Having dense deployment of APs, the faster AP discovery mechanism, frequent beacons, smart and faster handoff, and neighbor AP advertisement will help to decrease the communication traffic, provide seamless AP handoff, and solve probe storm.

The FILS is described as the key to address the scenario of massive amount of users in [28]. In this scenario, the users could be a large crowd of people flock in urban area, trying to connect to the internet at approximately the same time. For example, the stadium, arena typically need plenty of APs to cover the area and provide enough capacity. The users could also be a large number of sensors or actuators scattered in the field or forest. To monitor the sensing status timely, thousands or even more sensors need to feedback in a short period of time. A fast and secure access mechanism is important to ensure the low latency and high sensitivity. Thus here comes FILS providing an efficient and well-scalable mechanism guarantees that the network capacity is not going to be overwhelmed. To achieve this, each step, consisting of AP discovery, network discovery, authentication and association, and higher layer configurations has to be optimized to reduce the total access time. Apart from other steps, it is specifically mentioned that the optimization has been made in both active and passive scanning. In active scanning, AP's probe response can be broadcasted instead of unicasted to save time. The passive scanning performance is boosted by transmitting FILS Discovery (FD) frame including information of beacon frame more frequently.

[29] claimed to be the first paper attempting to introduce 802.11ai amendment, and analyze the situation where multiple APs in an extended service set (ESS) environment with a large number of STA in context of the Task Group (TGai) reference use case in 2012. It firstly presented the TGai reference use case that a quantity of people entering and leaving the coverage of ESS and could be generalized as Tokyo central station case. Although the legacy IEEE 802.11 standard works well when the number of STA in a given time period is small, it needs to be reformed when the number of users accessing the system simultaneously scales to another

magnitude. To solve this, [29] proposes an enhancement of active scanning scheme to minimize the amount of time spent on discovery by STA by amending the scanning rules and including more context information in the discovery phase. For example, additional context information could be provided by external sources, such as offline Wi-Fi database, location information in mobile devices. Another way to reduce the probability of medium congestion is to reduce the amount of probe request and response used during the active scanning, which is to use notion of broadcast instead of unicast addressing.

According to [29], the FILS procedure includes four steps, namely, access point discovery, network discovery, authentication and association, and higher layer operation. In [5], only the network discovery algorithm has been discussed, while other steps including the authentication and association have not been much mentioned. The purpose of the step of authentication and association is to provide a security mechanism to protect the system from malicious attackers. The authentication and association scheme allows the network devices to authenticate each other and negotiate to generate session keys to protect the information in the delivery.

There is also a paper discussed about the security concerns about FILS [30]. The author Lee expressed the concerns about the susceptibility to DoS attack. According to [30], the attackers are able to forge any message in during the exchange at any time. In this assumption, the attackers are able to launch three kinds of DoS attacks: ERP Failure Attack, FILS Handshake Corruption Attack.

The first attack can be launched after Msg 1, authentication request and before Msg 2, authentication response by sending fake Msg 2 to the STA with setting the R-bit as 1. R-bit as 1 indicates that the identity confirmation fails. Since the Msg 1 has to be forwarded to AS to check the validity, there is a time delay between the Msg 1 and the Msg 2. Thus the attacker may make use of this delay to convince the STA that the authentication already fails. But the first two messages are inserted in ERP. According to Extensible Re-authentication Protocol (ERP) specification, rfc6696 [31], two messages are transmitted to accomplish the re-authentication functions. And these two messages are both protected by re-authentication Integrity Key (rIK). The details of ERP will be fully introduced in Chapter 4. The protection of rIK ensures the integrity of these messages so that the attackers cannot forge the Msg 2 unless the attackers have rRK, which is the pre-shared key from 802.1x authentication. And replaying the message is also infeasible because the replay counter is introduced in ERP. Thus, this attack is unlikely to succeed.

The second attack intends to interfere the derivation of PMK of both parties. To achieve this goal, the author put forward that the attacker could sabotage the delivery of FILS Key Confirmation data by forging the Msg3 or Msg4, which are the association messages. FILS Key Confirmation data here, according to [30], is used to generate PMK. However, according to [3], the FILS Key Confirmation element is designed to confirm the consistency of PMK, which means that PMK plays a part in FILS Key Confirmation data. The typical example is Key-auth

field in FILS Key Confirmation data, which will be introduced in detail later. And based on this fact, PMK has already been derived before association messages, which contains the FILS Key Confirmation data. Apart from this, all the input keying materials are delivered to each other by Msg 1 and Msg 2. Therefore, there is by no means any way for attackers to interfere the generation of PMK by forging association messages.

2.4 Key Reinstallation Attack on IEEE 802.11 Standard Family

2.4.1 KRA Review

According to [5], 4-Way Handshake provides the mutual authentication and session key agreement, which with Advanced Encryption Standard - Counter Mode Cipher Block Chaining Message Authentication Code Protocol (AES-CCMP), a data confidentiality and integrity protocol, forms the crucial foundation of 802.11i. 802.11i amendment has been widely used for 14 years, some weaknesses have been revealed. The well-known weakness is Wi-Fi Protected Access - Temporal Key Integrity Protocol (WPA-TKIP) [32-33]. And some attacks exploit flaws in Wi-Fi Protected Setup in [34], predictable pre-shared key in [35], and flawed random number generators in [36]. But these attacks are not exclusively against 4-Way Handshake. Although with some design flaws found in early release mentioned in previous sections, 4-Way Handshake is generally secure against a variety of attacks. However, unfortunately, KRA seems a new challenge for 14-years-old 802.11i.

The KRA can be possibly launched on devices installing 802.11 authentication protocols. As described in [5], several authentication protocols are analyzed in different situations, including 4-Way Handshake, Group Key Handshake, Peerkey Handshake, and FT Handshake. But the core thought of those attacks is managing to make either STA or AP to reinstall the common keys and meanwhile reset the nonce so that the nonce is reused. Whereas nonce is supposed to be an arbitrary number that can only be used once. Once the nonce is reused, it will cause damage to different message confidentiality protocols depending on the roles nonce plays. For example, if Temporal Key Integrity Protocol (TKIP) is used, the abuse of reuse of nonce can not only decrypt the message, but forge frames in one specific direction after attacking the weak Michael algorithm [33]. But only messages can be decrypted if CCMP is used. Although [37] provides some thoughts about forging messages with repeated nonce, it is not feasible if used on any message. As mentioned in the paper, if Galois/Counter Mode Protocol (GCMP) is used, apart from replaying and decrypting packets, recovering authentication key is also possible in [38]. With the recovered authentication key, the adversary is able to forge message from both side of the channel, which makes the communication more fragile.

As described above, the messages can always be decrypted and replayed against legitimate party no matter which confidentiality protocol is used. In general, the reuse of nonce can severely jeopardize legal communications. But attacks on different handshake protocol depend on which communication is vulnerable. For example, 4-Way Handshake is executed for a general key

exchange purpose, such as data transfer, so that unicast or broadcast frames entailing data transfer are susceptible to various attacks. And for FT Handshake, target is different and the messages in FT Handshake and subsequent communications are unsecure.

KRA is indeed devastating if launched successfully. But during the 14 years when 4-Way Handshake is widely used, it has not been revealed to be as a threat as other typical widely discussed attacks. This is because it is different from other typical attacks, such as the attack against the first unprotected message in 4-Way Handshake. It depends on whether the retransmission is attacked. In other words, the previous attacks mostly aim at the first-time message, but KRA aims at the retransmission of a specific message, which would trigger the reinstallation of keys. In this perspective, the logic correctness and security properties of the first-time messages are not affected by the KRA. Thus in [5], it is mentioned that KRA does not violate the security properties proven in formal analysis of the 4-Way and group key handshake. Even in [39], the authors proved that the 4-Way Handshake provide the key secrecy and session authentication. For key secrecy, the authors proved that only supplicants and AP possess the PTK, but KRA does not recover PTK for the third party. For session authentication, it is said that a protocol is secure if the only way that an adversary can get a party to complete the protocol is by faithfully relaying messages [40], but KRA just relays messages and does not change any bit of them. Consequently, KRA does not theoretically violate formal security properties.

Another aspect makes KRA different is the platform running the protocol. For industrial applications, the program engineers or software engineers will code the protocol in devices according to the same framework within 802.11 to allow various devices to communicate. However, apart from the framework, there are still some parts specified by the engineers to make the practical protocol fully functional. Therefore, KRA would have different strategies when confronting different platforms, such as Android, Windows, and iOS. On the contrary, the previous typical attacks mostly do not depend on the platforms.

Although the author in [5] admitted that the reasons why KRA is possible also include the flaws in practical applications, the discussion part at the end of the paper revealed that specification of the protocol should be sufficiently precise and explicit. In other words, if the specification or standard is unambiguous enough, it would not leave too much room for application designers to design various versions, some of which involve flaws. As for 802.11ai, the paper did not entail whether there are flaws within. I suppose it partially results from the fact that no commercial products running 802.11ai. But according to the conclusion drawn in the paper, it is still possible to avoid the problem if an unambiguous protocol is designed. Hence, SEF is proposed in Chapter 4.

always to be encrypted so that it has to use encrypted Msg 3 retransmission strategy to attack FILS. But for encrypted Msg 3 retransmission, it also classifies them into two consecutive unencrypted Msg 3 are acceptable and strictly encrypted Msg 3 is acceptable. Finally, according to FILS specification, strictly encrypted Msg 3 is applicable. Hitherto, how KRA is going to attack 4-Way Handshake with strictly encrypted Msg 3 retransmission falls into one category and is introduced as follows.

For strictly encrypted Msg 3 retransmission against 4-Way Handshake, KRA first has to obtain a MitM position between STA and AP for adversary. This could be infeasible if a rouge AP is employed with different MAC address because the MAC address is involved in session key derivation. Therefore, a channel-based MitM attack [41] is used. After a MitM position is established, the adversary will faithfully relay the message to both parties till the last message Msg 4. Since Msg 4 is intercepted by the adversary, AP cannot receive the acknowledgement from STA. AP would retransmit the Msg 3 again, which would make STA reinstall the already-in-use key and reuse nonce if STA accepts the retransmitted Msg 3. The problem here is how to make STA accepts the two encrypted Msg3. So the attack is designed as Figure 9 illustrates from [5]. To avoid the situation that the victim does not accept two encrypted Msg 3s, the race

Figure 5 Typical KRA Attacking Procedure

between main CPU and wireless NIC is taken advantage of. Two consecutive Msg 3s would make main CPU process the decrypted messages twice so that the keys would be installed twice and reuse nonce. How the race can be achieved is described in detail in [5].

2.5 Security Evaluation Tools

Security evaluation tools are indispensable in protocols design. Thus in this section, the introduction on two security evaluation tools, Burrows, Abadi and Needham (BAN) and Scyther Tool, used in the subsequent chapters are provided.

2.5.1 BAN Logic

To verify the security functionality of the designed protocols, BAN Logic can ensure the logic correctness stands. BAN Logic is usually used to analyze and prove the security of the authentication protocols. Based on message origin, messages freshness and origins trustworthiness, it uses logical postulates and basic definition to verify the belief of each agent, which leads to the trust among each other [42]. The agents are the participants of the communications for information exchanges. BAN Logic uses symbols to denote the operations and principles. The meaning of a symbol is listed as follows.

$P| \equiv X$: P believes X

$P \triangleleft X$: P sees X

$P/\sim X$: P once sent X

$P/\Rightarrow X$: P has jurisdiction (control) over X

$\#(X)$: The message X is fresh

$P \stackrel{K}{\leftrightarrow} Q$: Secret key K is shared between P and Q for communication

$\{X\}_K$: Message X is encrypted using the key K

To complete the verification, four crucial elements are needed, idealized form, goals, assumptions, and formal verification.

BAN Logic also provides useful rules or postulates for the completion of the verification in [42]. There are several common postulates as follows.

Seeing rule: When P believes in the shared key K between P and Q , P will see the message content if the message is encrypted by the shared key.

$$\frac{P| \equiv P \stackrel{K}{\leftrightarrow} Q, P \triangleleft \{X\}_K}{P \triangleleft X}$$

Message-meaning rule: When P believes in the shared key K between P and Q , P will believe that Q once sent X if P receives the message encrypted by the shared key K

$$\frac{P| \equiv P \stackrel{K}{\leftrightarrow} Q, P \triangleleft \{X\}_K}{P| \equiv Q| \sim X}$$

Nonce-verification rule: If P believes that the nonce X is fresh and P believes that Q once sent X , P will believe that Q also believes X .

$$\frac{P| \equiv \#(X), P| \equiv Q| \sim X}{P| \equiv Q| \equiv X}$$

Jurisdiction rule: If P believes that Q has jurisdiction over X and Q also believes X , P will believe X by itself. This postulate can be used to acquire a goal.

$$\frac{P| \equiv Q| \Rightarrow X, P| \equiv Q| \equiv X}{P| \equiv X}$$

Believe properties: If P respectively believes X and Y , P also believes the message containing X and Y . If P believes the message containing X and Y , P also believes X and Y , separately. If P believes that Q believes the message containing X and Y , P believes Q also believes X and Y , separately

$$\frac{P| \equiv X, P| \equiv Y}{P| \equiv (X, Y)}$$

$$\frac{P| \equiv (X, Y)}{P| \equiv X}$$

$$\frac{P| \equiv Q| \equiv (X, Y)}{P| \equiv Q| \equiv X}$$

Sent property: If P believes that Q once sent the message containing X and Y , P will believe that Q once sent X and Y , separately.

$$\frac{P| \equiv Q| \sim (X, Y)}{P| \equiv Q| \sim X}$$

Fresh property: If P believes the freshness of nonce X , P will believe the freshness of message containing X .

$$\frac{P| \equiv \#(X)}{P| \equiv \#(X, Y)}$$

2.5.2 Scyther Tool

With large amount of protocols have been developed, the thorough security analysis of each turns out to be difficult for humans. For example, the Wired Equivalent Privacy (WEP) was proved to have security vulnerabilities after its deployment. To avoid releasing flawed protocols, using protocol formal verification tool to guarantee the security invulnerabilities is critical to design practical protocols. There are some popular tools, such as Scyther Tool [42],

ProVerif [43] and Automated Validation of Internet Security Protocols and Application (AVISPA) [44]. However, Scyther tool has increasingly wide use nowadays, thus it is utilized to perform formal verification of the design protocols in this thesis. Scyther tool has several distinguished features, such as free-to-use, open source, verification, falsification, and analysis of security protocol. [45]. Besides providing the cutting-edge performance as other tools, some novel features have been held such as the possibility of unbounded verification with guaranteed termination, analysis of infinite sets of traces in terms of patterns, and support for multi-protocol analysis [46].

Since Scyther Tool is a role-based analysis tool, those participants of the communications for information exchanges should be defined respectively in a verification model, where they are named agents. The behavior, namely, calculations, sending and receiving messages are described by Security Protocol Description Language (SPDL) based on the operational semantics found in [47]. To clearly and unambiguously express the meanings of each message, the events, such as sending and receiving, have to be carefully specified, because the information in messages, such as sender, and receiver, forwarded content, nonce, encryption keys, confirmations, is complex, and the claims cannot be verified correctly if the information has a tiny flaw. Another feature, the verification of claims indicates whether there are vulnerabilities for attackers to make use of or not. Fortunately, Scyther Tool will give a clear flow chart of the possible attacks if there is any flaw in the protocol design to help users refine the details and avoid unexpected vulnerabilities. The claims are also very important because they are directly related to the quality of the protocol. After all, the tool can automatically analyze the protocol based on the claims which are those properties that a user intends to acquire. Usually, it includes the secrecy and integrity of each variables delivered in the protocol and the aliveness, weak agreement, non-injective agreement, and non-injective synchronization of each agent. Another convenient feature offered by Scyther Tool is that it can generally analyze the protocol with automatically generated commonly used claims to help the user have a quick overview of the protocol.

Chapter 3. FKR Mechanism

3.1 Motivations

Based on those discussions in the previous chapter, the legacy scheme proposed in 802.11i was not entirely safe or efficient. In the following decade, many proposals attempted to make positive changes to it until a finalized version of IEEE 802.11ai's release, which integrated some advantages of ingenious proposals and eliminated some unnecessary steps. With superior new features, 802.11ai is much more adaptive than its predecessors. Thus, it is reasonable to be applied to IEEE 802.11ah, who needs a simplified, lightweight and specifically adapted authentication mechanism. Moreover, in the IoT scenarios, there could be hundreds of nodes serving as STAs, which cannot keep the connection with an AP all the time because of excessive energy consumption. Those nodes mostly stay dormant unless there are data to transfer or they are interrupted. Once they intend to establish a connection, new session keys are required to be generated. Therefore, multiple session key generations are possible. They may need to renegotiate for the short term session key named PTK. It needs to re-perform the entire FILS authentication process over and over again within the PMK lifetime, which would take more time and energy. Here PMK lifetime is indicated by AS from $\text{Session-Timeout} + \text{dot1xAuthTxPeriod}$ or from $\text{dot11RSNA-ConfigPMKLifetime}$. And PTK and GTK lifetime cannot exceed PMK lifetime. Based on FILS, the core mechanism of 802.11ai, we propose a new mechanism called Fast Key Reestablishment (FKR) to simplify the authentication and key generation process while keeping the same security level of the FILS. It provides a faster way to accomplish authentication and session keys generation after first round of communication without degrading the security functionality. In the followings, details of FILS and FKR are presented and compared in terms of simulated and calculated performance.

3.2 The Proposed FKR Scheme

3.2.1 Details of Proposal

With the aims to simplify the subsequent FILS authentication, we design the FKR scheme in Figure 10, which works as follows:

After the first round of the FILS authentication has been executed, the PMK is settled and can be used for a certain period of time, while the PTK could be changed and renegotiated for the rest of the session. The FKR scheme provides a 2-way method to reestablish a new PTK set with the same PMK serving as an input keying material. During the 2-way handshake, the STA and the AP can renegotiate a new PTK session and confirm each other again without exposing the 4-way handshake. Thus, it is assumed that the STA and the AP have already shared a PMK and PTK from the previous authentication. By the FKR, they have agreed on a same hash function.

First frame: $S \rightarrow A: \{N_S, \{N'\}_{PTK'}\}_{PTK}$

The STA and the AP first use hash function to calculate N as follows:

$$N = Hash(SNonce || ANonce)$$

N is then used in the same formula with N_S to derive N' who serves as an input keying material. The STA then generates N_S as a new nonce for this key derivation. And the new PTK denoted as PTK' is calculated as follows:

$$PTK' = PRF - X(PMK, SPA || AA || N' || N_S)$$

After they are derived, the STA sends the second message.

Second frame: $A \rightarrow S: \{N', GTK\}_{PTK'}$

Upon receipt, the AP uses PTK derived in previous session to decrypt the frame from the STA and derive N_S . The AP then uses the same way to calculate PTK' . Once PTK' is calculated, the AP is able to verify the N_S encrypted with PTK' in the first message. If it is verified to be

Figure 6 FKR Message Flow and operations by STA and AP

successful, the AP can construct the second message with the GTK and the current Key RSC included. Through a successful verification, the STA has the knowledge of the PMK, the previous used PTK, and the freshly generated nonce for this PTK' , the AP can trust the newly calculated PTK' as the new PTK for the next session. The second message sent by the AP proves that the AP accepts the new PTK' and has the knowledge of PMK and previous used PTK.

The keying material N' will be saved by both the AP and the STA for further usage. In the subsequent session, they simply derive N'' serving as the same function in the current session with N_S and newly generated N_S' .

Note that although the AS role is not presented in the FKR scheme, it does not affect the authentication. Since the FKR is implemented in subsequent rounds of the authentication, and AS has already been involved in the first round of authentication so that the AS is not necessary to be shown in the operation of the FKR scheme.

FILS and FKR can be possibly implemented if the advertising message, such as a beacon or probe response, broadcast by AP includes available FILS information. FILS indication element contains the information related to FILS authentication and higher layer setup capabilities of AP. And FILS information field occupies two bytes in the element. Within the FILS information field, AP is able to indicate its capability to perform several authentication methods, shared key authentication with or without PFS, public key authentication. These options are denoted with one bit each. While there can also be one more bit indicating the availability for FKR, because reserved bits exist. If STA intends to begin a FILS authentication with AP with the FKR option enabled, it may include this information about the expected number of rounds in Msg 1. And

Figure 7 FKR Element Format

when AP receives the lifetime of PMK, the maximum FKR rounds will be calculated by AP and STA will be informed with Msg 2. The element format of FKR information is as follows in Figure 11:

Because the subsequent authentication rounds are usually not many, one octet is enough for the indication. Therefore, FKR information only takes one more bit in advertising element and one more element if it is enabled. If either STA or AP is not able to support this scheme, only the one bit indication is at most included and element field can be ignored.

3.3 Security Evaluation

To verify the security functionality of the proposed FKR scheme, we take advantage of three components to give a comprehensive evaluation. The logical correctness of FKR is to ensure that there is no logical loophole in logic can be made use of by attackers. In this and next chapter, BAN Logic is employed, because it is a widely used and recognized logic verification tool. Secondly, the formal verification is accomplished by the Scyther Tool, which has increasingly wide use in recent years. Some of its new features are designed to be used efficiently. And the security provided by its verification is also widely recognized. The third component is the qualitative analysis of the protocol. By giving the reasons why those well-known typical attacks cannot be launched successfully, the overview of its security is provided.

3.3.1 Proof of Logic Correctness

As introduced in previous chapter, BAN Logic needs four elements to complete the verification. In this section, the four elements are presented as follows.

First, the two messages have been exchanged as follows:

First message: $S \rightarrow A: \{N_S, \{N'\}_{PTK'}\}_{PTK}$

Second message: $A \rightarrow S: \{N', GTK\}_{PTK'}$

- Idealized Messages

Idealized form is an important step that messages have to be idealized to derive the appropriate basis for formal analysis. Firstly, the information like plaintext must be omitted because it can be forged. Secondly, the encrypted messages have to be transformed into assertions to make them clearer and more complete [42]. The idealized form is

$S \rightarrow A: \{N_S, A \xleftrightarrow{PTK} S\}_{PTK}$

$A \rightarrow S: \{N', GTK, A \xleftrightarrow{PTK'} S\}_{PTK'}$

- Assumptions

Assumptions are those conditions made to initialize the verification. Typically, the assumptions state what keys are initially shared between the principals, which principals have generated fresh nonces, and which principals are trusted in certain ways. The freshness refers to the message that has not been transmitted before the current run of the protocol. The freshness of nonce is often to be true because nonces commonly include a timestamp or a number that is used only once. The assumptions of this verification are:

Both the STA and the AP believe in the previous used PTK.

$$S| \equiv A \xleftrightarrow{PTK} S$$

$$A| \equiv A \xleftrightarrow{PTK} S$$

The AP believes the freshness of the nonce generated by the STA because of the timestamp embedded in it.

$$A| \equiv \#(N_S)$$

The STA calculates the next PTK by itself so that it naturally believes in PTK' ,

$$S| \equiv A \xleftrightarrow{PTK'} S$$

The AP generates and distributes the GTK.

$$S| \equiv A| \Rightarrow GTK$$

- Goals

Goals are the destination of verifications, which are usually the conclusions to be drawn. It has to prove the derivation of the new session key PTK' for both parties so that they are able to use the new PTK' they both believe to transmit data. And they both believe that each other believes it. The goals of this verification is listed as follows:

Both sides have to believe the other side and believe in the common key set derived.

$$A| \equiv S| \equiv A \xleftrightarrow{PTK'} S$$

$$S| \equiv A| \equiv A \xleftrightarrow{PTK'} S$$

Both sides have to believe the other side believe in the newly calculated nonce.

$$A| \equiv S| \equiv N'$$

$$S| \equiv A| \equiv N'$$

The STA has to believe in the GTK distributed by the AP.

$$S| \equiv GTK$$

- Verification

With these postulates, the verification can be proceeded with concrete theoretical basis as follows:

According to the message-meaning rule,

$$\frac{A| \equiv A \xleftrightarrow{PTK} S, A \triangleleft \{N_S, A \xleftrightarrow{PTK} S\}}{A| \equiv S| \sim \{N_S, A \xleftrightarrow{PTK} S\}}$$

According to the properties,

$$\frac{A| \equiv \#(N_S),}{A| \equiv \#(N_S, A \xleftrightarrow{PTK} S)}$$

According to the nonce-verification rule,

$$\frac{A| \equiv \#(N_S, A \xleftrightarrow{PTK} S), A| \equiv S| \sim \{N_S, A \xleftrightarrow{PTK} S\}}{A| \equiv S| \equiv \{N_S, A \xleftrightarrow{PTK} S\}}$$

According to the properties,

$$\frac{A| \equiv S| \equiv \{N_S, A \xleftrightarrow{PTK} S\}}{A| \equiv S| \equiv \{N_S\}}$$

Once the AP believes that the STA holds the belief on N_S , it will calculate N' as well as PTK' . Thus it is reasonable to believe that

$$A| \equiv \#(N')$$

$$A| \equiv \#(N', A \xleftrightarrow{PTK'} S)$$

Since the STA mentioned N' and PTK' in the first message, the AP can compare them with the values calculated by itself. Hence the idealized form can be modified as:

$$S \rightarrow A: \{N_S, A \xleftrightarrow{PTK} S, N', A \xleftrightarrow{PTK'} S\}_{PTK}$$

Then, according to the message-meaning rule,

$$\frac{A| \equiv A \xleftrightarrow{PTK'} S, A \triangleleft \{N_S, A \xleftrightarrow{PTK} S, N', A \xleftrightarrow{PTK'} S\}}{A| \equiv S| \sim \{N_S, A \xleftrightarrow{PTK} S, N', A \xleftrightarrow{PTK'} S\}}$$

Again according to the nonce-verification rule,

$$\frac{A| \equiv \#(N_S), A| \equiv S| \sim \{N_S, A \xleftrightarrow{PTK} S, N', A \xleftrightarrow{PTK'} S\}}{A| \equiv S| \equiv \{N_S, A \xleftrightarrow{PTK} S, N', A \xleftrightarrow{PTK'} S\}}$$

According to the properties,

$$\frac{A| \equiv S| \equiv \{N_S, A \xleftrightarrow{PTK} S, N', A \xleftrightarrow{PTK'} S\}}{A| \equiv S| \equiv \{N', A \xleftrightarrow{PTK'} S\}}$$

$$\frac{A| \equiv S| \equiv \{N', A \xleftrightarrow{PTK'} S\}}{A| \equiv S| \equiv \{N'\}}$$

$$\frac{A| \equiv S| \equiv \{N', A \xleftrightarrow{PTK'} S\}}{A| \equiv S| \equiv \{A \xleftrightarrow{PTK'} S\}}$$

For message 2, according to the message-meaning rule,

$$\frac{S| \equiv A \xleftrightarrow{PTK'} S, S \triangleleft \{N', GTK, A \xleftrightarrow{PTK'} S\}}{S| \equiv A| \sim \{N', GTK, A \xleftrightarrow{PTK'} S\}}$$

According to the properties,

$$\frac{S| \equiv \#(N'),}{S| \equiv \#(N', GTK, A \xleftrightarrow{PTK'} S)}$$

According to the nonce-verification rule,

$$\frac{S| \equiv \#(N', GTK, A \xleftrightarrow{PTK'} S), S \sim \{N', GTK, A \xleftrightarrow{PTK'} S\}}{S| \equiv A| \equiv \{N', GTK, A \xleftrightarrow{PTK'} S\}}$$

According to the properties,

$$\frac{S| \equiv A| \equiv \{N', GTK, A \xleftrightarrow{PTK'} S\}}{S| \equiv A| \equiv \{N'\}}$$

$$\frac{S| \equiv A| \equiv \{N', GTK, A \xleftrightarrow{PTK'} S\}}{S| \equiv A| \equiv \{A \xleftrightarrow{PTK'} S\}}$$

$$\frac{S| \equiv A| \equiv \{N', GTK, A \xleftrightarrow{PTK'} S\}}{S| \equiv A| \equiv \{GTK\}}$$

According to the jurisdiction rule,

$$\frac{S| \equiv A| \Rightarrow GTK, S| \equiv A| \equiv \{GTK\}}{S| \equiv GTK}$$

Hitherto, we have proved that both sides believe that the other side believes in the common secret keys derived, and they both believe the nonces generated and GTK.

In conclusion, the BAN Logic proof of FKR completes the goals as follows:

- Mutual authentication between STA and AP based on SNonce and previous PTK.
- Derivation of the new session key PTK for both parties.
- Successful delivery of group temporary key from AP to STA.

With conclusions drawn above, the protocol is now logically accurate and no logic flaws may emerge in the future to be taken advantage of by the attackers.

3.3.2 Formal Verification

As introduced in previous chapter, the protocol operations have to be described in SPDL to complete the formal verification with Scyther Tool. The followings are the SPDL descriptions of the participants. The description of the FKR scheme and the derived result are presented as follows and Figure 12.

hashfunction hash;

// User type declaration

usertype Key;

//constant defining


```

const nonceold: Nonce;
const prf: Function;
// Protocol description
protocol protocol2(I,R)
{
  role I
  {
    fresh ni: Nonce;
    var nr: Nonce;
    var ptk2: Key;
    match(ptk2, prf(hash(nonceold, ni),k));
    send_1 (I,R, { ni,I,{nonceold}ptk2 }k );
    recv_2 (R,I, { nr,R} ptk2);
    claim_i1 (I, Secret,ptk2 );
    claim_i2 (I, Nisynch );
  }
}
role R
{
  var ni: Nonce;
  fresh nr: Nonce;
  var ptk2: Key;
  recv_1 (I,R, { ni,I,{nonceold}ptk2 }k );
  match(ptk2, prf(hash(nonceold, ni),k));
  send_2 (R,I, { nr,R} ptk2);
  claim_r1 (R, Secret, ptk2 );
  claim_r2 (R, Nisynch );
}
//s: (nounce, (nonce_old)PTK2 ) PTK
//a: (nounce2, )PTK2

```

The Figure 12 indicates that the claims either required in the script or generated automatically are verified. Note that the protocol name is referred as protocol2 with I as the initiator and R as the responder. Initiator this experiment is STA and the responder in this experiment is AP. Since the re-authentication process does not have to include AS to authenticate again, only two parties are engaged in the experiment.

In this description, the claims I intend to verify are Secrecy and non-injective synchronization (Nisynch). The

Claim				Status		Comments
protocol2	I	protocol2,I1	Secret _Hidden_1	Ok	Verified	No attacks.
		protocol2,I2	Secret ni	Ok	Verified	No attacks.
		protocol2,I3	Secret ptk2	Ok	Verified	No attacks.
		protocol2,I4	Secret nr	Ok	Verified	No attacks.
		protocol2,I5	Alive	Ok	Verified	No attacks.
		protocol2,I6	Weakagree	Ok	Verified	No attacks.
		protocol2,I7	Niagree	Ok	Verified	No attacks.
		protocol2,I8	Nisynch	Ok	Verified	No attacks.
	R	protocol2,R1	Secret _Hidden_3	Ok	Verified	No attacks.
		protocol2,R2	Secret _Hidden_2	Ok	Verified	No attacks.
		protocol2,R3	Secret ptk2	Ok	Verified	No attacks.
		protocol2,R4	Secret nr	Ok	Verified	No attacks.
		protocol2,R5	Secret ni	Ok	Verified	No attacks.
		protocol2,R6	Alive	Ok	Verified	No attacks.
		protocol2,R7	Weakagree	Ok	Verified	No attacks.
		protocol2,R8	Niagree	Ok	Verified	No attacks.
		protocol2,R9	Nisynch	Ok	Verified	No attacks.
	Done.					

Figure 8 Scyther Tool Verification Result of FILS

definitions of Secrecy and Nisynch are given in [48]. Secrecy expresses that certain information is not revealed to an adversary, even though this data is communicated over an untrusted network. And Nisynch states that everything we intended to happen in the protocol description also happens in the trace. These are the basic security property required for the executions of this protocol. And the execution result shows that these basic properties are satisfied. Another mode is to automatically generate proper claims as illustrated in Figure 12.

From the result, the Aliveness, Weak agreement, Non-injective agreement (Niagree) are additional security properties generated by scyther tool. Aliveness defined in [49] is a property of one agent to another agent in the protocol. For example, Aliveness of agent B is ensured to agent A, if A (acting as initiator) completes a run of the protocol, apparently with responder B, then B has previously been running the protocol. Note that B may not necessarily have believed that he was running the protocol with A. Also, B may not have been running the protocol recently. Many protocols fail to achieve this property, such as a mirror attack, simply reflecting an agent's messages back at himself in [50]. The Weak agreement [49] of agent B is ensured agent A, if A (acting as initiator) completes a run of the protocol, apparently with responder B, then B has previously been running the protocol, apparently with A. Note that B may not necessarily have been acting as responder. The typical attack in [49] that the Weak agreement is failed is described that an intruder imitates B to communicate with A by using B as an oracle in a parallel run where the intruder uses his own identity. This means that A thinks he is talking to B, but B thinks he is talking to the intruder. The example is this typical attack on Needham-Schroeder Public Key protocol [51] demonstrated in [52]. The last one added automatically is the Niagree [48], which is similar as the Nisynch, with the only difference is is that synchronization additionally requires that the communication order is respected. For example, send events occur before their corresponding read event. The Secrecy properties of other variables are also verified.

3.3.3 Security Analysis

The proposed FKR serves as the FILS's supplement to achieve the FILS functions. Firstly, it confirms the PMK on each side. Secondly, the STA and the AP can mutually authenticate each other. Thirdly, both sides derive a fresh session key PTK. Finally, it distributes the GTK to the STA. The FKR requires the AP to calculate the new PTK with the PMK and the new nonce received. It proves that the AP has the knowledge of the PMK. From the information encrypted with new PTK in the first message, it is apparent that the STA has the knowledge of the PMK. And the knowledge of the old PTK as well as the knowledge of the PMK makes them authenticate each other. After 2 messages exchanged, both sides demonstrate that both of them accept the newly calculated PTK. The second message contains the GTK distributed to STA.

Figure 9 The Retransmission Scheme If First Message Is Lost
The FKR scheme is resistant to several malicious attacks.

- Message Loss

Attackers may intercept messages and use them to interfere the operation of protocol or even impersonate the

Figure 10 The Retransmission Scheme If Second Message Is Lost
other side. Two messages can be lost or intercepted.

If the first message, $S \rightarrow A: \{N_S, \{N'\}_{PTK'}\}_{PTK}$ is lost, the STA has to regenerate N_S and resend the first message. Since the STA has no idea on which message lost, it simply does not receive any reply. Thus, it generates a new N_S with the same PTK used in the first message. Before it reaches to reasonable times of attempts, the STA would always save all the newly calculated PTK' derived from different N_S . The AP will simply ignore them to use another N_S for its work.

If the second message, $A \rightarrow S: \{N', GTK\}_{PTK'}$ is lost, the STA still resends the same message with a different N_S , while the AP simply discards all the messages and resend the lost second message. When the STA finally

receives the lost second message, it begins to use the saved PTK' to decrypt the second message. Upon success, the right PTK' is the negotiated one. By the FKR scheme, the AP and the STA can locate the right messages to proceed with next operations.

The reason to implement this retransmission scheme is to prevent replay attacks. However, adding a monotonically increasing sequence number in the first message can achieve better efficiency. Because encrypted sequence number is not vulnerable and can indicate the latest first message. And when STA receives the second message from AP, the sequence number can also indicate which SNonce is used to generate the new PTK' if the sequence number is encrypted by the old PTK.

- DoS Attack

The FKR scheme has its ability against the denial of service (DoS) attacks. Since the two messages are protected by the previous used PTK, it is unlikely for the DoS attacks to succeed, because the integrity of those two messages is ensured by the PTK. The legacy standard has the vulnerability to DoS attacks because of its unprotected first message, which could be flooded by the attackers to induce consistent calculations or deadlock. In those proposed schemes aiming at resolving this problem, making use of temporary PTK or PMK, or even other types of temporary keys to ensure the integrity of the first message is common.

- Replay Attack

The FKR scheme has its ability against replay attacks. The STA may generate more than one first message with different N_S , $S \rightarrow A: \{N_S, \{N'\}_{PTK'}\}_{PTK}$ due to the message loss. Thus, these old messages transmitted before could be used to perform replay attacks by an attacker. Since the AP only chooses the first one it has received as the valid message to calculate the new PTK, the replayed messages will be silently discarded by the AP as well as other redundant messages sent by the STA. Since the AP chooses only one N_S to calculate one PTK' , if the STA receives more than one second message, $A \rightarrow S: \{N', GTK\}_{PTK'}$ from the AP, it only picks one to use and discard the others. It is pointless to replay these two messages. Moreover, the sequence number mentioned in the message loss scheme is commonly used to counter the replay attacks.

- Man-in-the-middle Attack

Man-in-the-middle (MITM) attacks could be launched successfully by an attacker if the attacker can relay the deciphered messages exchanged between the two parties of the communication to make them believe that they are talking to each other. By the FKR scheme, since both messages are protected by the ICK contained in the PTK to ensure the integrity of each message. Even if the PTK is compromised in the previous rounds, in the next round, the PTK is still not able to be derived by the attackers, due to the ignorance of PMK. Although the data transmitted in a whole session would be exposed if the PTK used in this session is compromised, there is no easy to compromise PTK because PMK is always a secret to the attackers unless the long-term key is compromised, which is out scope of this thesis. Thus it is not possible for the Man-in-the-middle attack to be successful.

- Perfect Forward Secrecy

The PFS for FKR depends on if PFS is desired in first round. For the subsequent rounds, PMK does not lead to previous PTK. If an attacker has the knowledge of the PMK, the long-term session key, the randomness of nonce

and the one-way nature of a hash function make it computationally impossible to get the knowledge of previous nonce and the previous session key, which can ensure that the old PTK cannot be obtained.

3.4 Performance Evaluation

3.4.1 Efficiency Analysis

In this section, we present the comparison of time and energy consumption between the proposed FKR and the FILS specified in the standard in order to demonstrate that the proposed scheme FKR is more resource-efficient. Since the reestablishment process does not include derivation of the PMK and the long term session key, they are not involved in the comparison.

Table 3 Message Comparison

Messages	FILS	FKR
S→A	Nonce generation, Public key pair generation	PTK derivation, Nonce generation, Nonce hash,
A→S	Nonce generation, Public key pair generation, PTK derivation, GTK derivation	PTK derivation, GTK derivation Nonce hash,
S→A	PTK derivation, Key-auth hash,	N.A
A→S	Key-auth hash,	N.A

The messages exchanged by both schemes are listed in Table 3. The lengths of various parameters are listed in Table 4. Some of these lengths are defined in 802.11ai specification. They may have different values according to the negotiated AKM. Thus, the minimum typical values are used in the efficiency analysis. According to [53-54], the time delay and energy consumption of various security functions are listed in Table 5. Since no products running 802.11ai has been commercialized, the calculations based on data tested acquired from experiments in [28] gives a qualitative analysis. The author conducted the experiments with PC and Compaq iPAQ H3670, which contains an Intel SA-1110 StrongARM processor clocked at 206MHz. Note that there are surely not only one source of power drain, transmission, receiving, verification, retransmission and so forth also consume time and energy, but they depend on various factors, such as devices, environments, negotiated AKM and so on. Therefore, only performances of operations are demonstrated to highlight the improvements.

In the case of N rounds of authentication required, since the first round of authentication is supposed to be completed by the FILS authentication, the resource consumption of the rest of $N-1$ rounds of authentication is worth analyzing. Based on the unit consumption of each round, Table 6 presents the time and energy

consumption of $N-1$ rounds of FILS authentication and $N-1$ rounds of FKR authentication. It is demonstrated that with increasing rounds of authentication, the FKR authentication is able to achieve significant performance improvements over the FILS authentication.

Table 4 Length of Parameters

Parameters	Length(bits)
PMK	256
PTK	512
Nonce	256
GTK	128
MAC address	48
Sequence Number	128
Diffie-Hellman Key	512
BSSID	48
ICK	256

Table 5 Time Delay and Energy Consumption of Algorithms

Algorithm	Time delay	Energy consumption
Hashing using MD5	25.57MB/s	24.06MB/J
Hashing using SHA-256	8.31MB/s	6.86MB/J
Hashing using SHA-512	5.27MB/s	4.52MB/J
AES-256	6.13MB/s	3.59MB/J

Message	Time Delay		Energy Consumption	
	FILS	FKR	FILS	FKR
S→A	17.52	76.098	52.61	59.02
A→S	87.433	27.024	73.376(275.936)	68.333

Table
Time
and

S→A	54.46	N.A	52.156	N.A
A→S	20.46	N.A	38.012	N.A
Total	179.873	103.122	216.154(418.714)	127.353
10 rounds	1978.603	1211.093	2377.694	1489.684
N rounds	$N \times 179.873$	$179.873 + (N - 1) \times 103.122$	$N \times 216.154$	$216.154 + (N - 1) \times 127.353$

6
Delay

Energy Consumption Comparison

Figure 11 Flowchart of Simulations

3.4.2 Simulation Evaluation

I further carry out simulation experiments to demonstrate the efficiency of the proposed FKR scheme when confronting the unknown attacks. The FKR scheme is able to be resistant to the known attacks which have been discussed in the security analysis. There are still unknown attacks which may interrupt the authentication process by interfering each step of the protocol. The FILS scheme and the FKR scheme have seven and six steps in the authentication process, respectively. Once any of these steps is interrupted, the entire process has to be reinitialized as illustrated in Figure 15. In the simulation, we analyze the performance of both schemes with increasing proportion of the unknown attacks in terms of the average time delay and average energy consumption by two schemes. It is calculated by dividing the sum of accumulated time required or energy consumed over total number of the authentications by the times of successful runs:

$$\text{Average consumption} = \frac{\text{Sum of time or energy}}{\text{successful runs}}$$

The simulation environment is Matlab R2014b, and the applied cryptographic algorithms are hash function, Secure Hash Algorithm 256 (SHA-256), and symmetric encryption, AES-128. So the model built in Matlab is basically the protocol behaves against the randomly emerging unknown attacks for over ten thousand runs. Once the unknown attack comes, the protocol might be interrupted by the attack because the unknown attack not always succeeds. If the attack succeeds, the simulation will count this run as a failure reinitiate the protocol. If

the attack does not succeed, the simulation will count this run as a success and add one to the counter. With the proceeding of the protocol, all time and energy consumption are recorded for calculations.

First, we have the average time delay and average energy consumption with the increasing of the percentage of the unknown attacks as illustrated in Figure 16. It is clear to see that with the increasing probability of unknown attacks, the time and energy consumption increase as well. Because more unknown attacks emerges during the simulation mean higher probability to interrupt he protocol within the period. And if the unknown attack probability is zero, there is no attack launched so that the time and energy value are the same as those in the Table 6. If the attack probability is 100%, the protocol can barely finish the authentication process so that the consumption of both reaches the highest point on average. Anyway, the comparison of the average time delay and average energy consumption by the FILS scheme and FKR scheme has explicitly demonstrated that the proposed FKR scheme has outstanding advantages over the FILS scheme.

Figure 12 Average Time Delay and Energy Consumption against Unknown Attacks

3.5 Summary

In this chapter, we have proposed the FKR scheme to enhance the efficiency of the link setup process in the 802.11ah wireless networks. To achieve this, the FKR has introduced a new message exchange process in the subsequent communications after the first round of the authentication. The logic correctness has been proved by using BAN logic and security analysis shows the security functionality of the FKR scheme with the ability against several malicious attacks. The performance evaluation has proved that the proposal is more efficient in terms of time and energy consumption.

Chapter 4. SEF Mechanism

4.1 Motivations

In this chapter, Secure and Efficient FILS (SEF) is proposed according to FILS as an authentication protocol. Basically, it has two new features: it is secure enough to be against KRA and other typical attacks, and it needs fewer messages to accomplish the process and reallocate the task to increase the efficiency.

Due to the immaturity of a new released standard, the practical commercialization still needs a large amount of time to complete its various unfinished features and solve all potential problems caused by legacy standards. Because these unresolved problems could still exist in the new standard, caused by the similar schemes. For legacy standards, especially the contents specifying the authentication protocol, flaws such as designing loopholes, vulnerable schemes, and unattended unnecessary overhead, to a great extent, have been eliminated, but there are still problems that could bring damage to the system. In [5], the KRA is described. This attack claims to be able to threaten various authentication protocols belonging to 802.11 standard family, such as 4-Way Handshake, Group Key Handshake, Fast Transition (FT) Basic Service Set (BSS), and Peer-Key Handshake and would cause severe damage to the secure operations. Although the reasons why this attack can be launched against various protocols are not exactly the same, they still share some common places. FILS authentication, proposed in 802.11ai, serving as an authentication mechanism belonging to 802.11 standard family, which means it inherits some features from its predecessors. Although industrial products operating 802.11ai have not entered the market so that the real performance of 802.11ai devices against KRA cannot be practically tested, it is still useful that this attack can be designed to be prevented from the protocol perspective. A better version of FILS authentication is needed to prevent KRA. Besides the enhanced security property, a further change to the protocol is made to optimize its performance to be better adapted to 802.11ah networks

4.2 SEF Overview

SEF scheme is proposed based on FILS to overcome the susceptibility of KRA and keep the performance still acceptable to 802.11ah networks. As mentioned in Chapter 2, the direct reason why KRA is able to launch successfully is that the attackers can acquire a MitM position and trick the victim to install the key set twice. Also, FILS authentication has the always encrypted Msg 2 and Msg 3. To avoid being attacked with two consecutive messages, including the retransmitted message, the state machines should be defined to make sure the victims strictly followed secure work flow. And the flag must be introduced to avoid reinstallation of key set. Meanwhile, the original valid operations must not be affected. And by redistributing the calculation tasks to AS, the burden of AP can be decreased. Moreover, the authentication process can be finished in three ways without affecting the original security. With less message, transmission and receiving overhead is less than before.

4.3 Details of SEF

The details of SEF include the message details and state machines of STA and AP as well as backward compatibility descriptions so that the operations can be strictly defined without causing ambiguity. To give a clear view of SEF, the messages exchange flow is presented first as follows in Figure 17:

Figure 13 SEF Message Flow and Operations of STA, AP, and AS

The message exchange flow is presented as in Figure 2. The messages exchanged between the AP and the AS, denoted as Msg1', Msg2', etc. are presented downward according to the message order. Note that PTK used in the message means the message is protected by authenticated encryption with associated data (AEAD) algorithm with Key Encryption Key (KEK) as specified in 802.11ai. Auth-field is used to confirm the possession of session keys, whose name in 802.11ai is Key-auth field. The calculation of Key-auth field is mentioned in previous chapter.

SEF needs three messages exchanged between STA and AP to accomplish the authentication. The session keys calculation task is redistributed to AS, because AP is not necessarily power-line device in 802.11ah networks and is possibly connected to thousands of STAs. Calculations of each session key set can be a huge amount of works, especially when AP is simply an ordinary router. But as for AS, server could be highly competent in calculations. Thus allocating these tasks to AS sounds more reasonable. Another change is containing the confirmation of the possession of session keys in the second message so that one less message is

needed after the third message. One less message means less time for both parties to send message and monitor the channel so that less energy is spent.

The authors in [5] pointed out that the reason why supplicants or STAs in this thesis always reinstall session key sets after the first time installation by demonstrating the state machine of the supplicants. Since STA would eventually return to install the keys if it receives a valid Msg 3 and fulfills several requests. Moreover, the 802.11ai specification does not specify several state machines to define the states to guarantee the stable operations for FILS authentication. Thus in order to avoid the attack and save as many resources as possible, state machines describing operations of STA, AP and AS respectively are proposed. In the state machines, messages sent from AP to AS are called Msg 1', Msg 2' respectively.

In Figure 18, Apart from the PRE-SEF state, STA has 4 states, and UCT means unconditional transition. From this state machine, it is clear that STA use information in Msg 2 to confirm the possession of rRK of AS and calculate PMK and PTK meanwhile. If STA verifies the Key-auth field successfully in Msg 2, it would install key set and send Acknowledgement (ACK). And if the Msg 2 is abused like KRA mechanism, STA would not return to install keys, instead it accepts the first ANonce and key set, and simply resend the Msg 3 as ACK again so that KRA would not succeed. To achieve this, STA can use Boolean value as flag to indicate that key set is installed. And both authentication failure from AS and Key-auth verification failure make STA abort this attempt and return to initial state. With this retransmission mechanism, STA does not accept more than one Msg 2 from AP, instead, it only accepts the first valid Msg 2. This is to reduce the overhead for calculating new keys and installing new key set. In other words, it transfers the memory burden from STA to AP to be friendlier

Figure 14 STA State Machine

to resource-constrained devices.

Figure 19 is AP's state machine. Since the Msg 1 is the initialization message, AP does not have to do anything before receiving Msg 1. When the Msg 1 is received, AP verifies the freshness of SNonce and appends ANonce to the Msg 1. After receiving the Msg 2', AP temporarily assign the PMK and PTK from AS to Temporary PMK (TPMK) and Temporary PTK (TPTK). If AP does not receive Msg 3 after sending Msg2, it would regenerate the new ANonce to ensure the freshness of new Msg 2 and send again. Because for most cases, communications are not always affected by lossy channels, calculation tasks transferred to AS for the first time can save more resources for AP, and the subsequent recalculations are not necessary to be transferred again if the Msg 3 is not lost. And if any of the received Msg 3 indicates the failed Key-auth element, AP would return the initial state and wait for Msg 1. The last thing is since AP does not return to any previous state if it receives more than one valid Msg 3, AP is not going to reinstall the key set, and is not prone to KRA.

For backward compatibility, the SEF scheme is designed to serve as a substitute to the FILS shared key authentication. In the authentication frame, there is an algorithm number field to indicate which algorithm is used. The original FILS is attempted to use by setting the algorithm number as 4 or 5 to indicate whether Perfect Forward Secrecy (PFS) is desired or not. After integrating the SEF, there are also two similar options: SEF with and SEF without PFS. The additional options are supposed to be broadcasted in the AP's beacon or probe response. If the STA would like to attempt the SEF while the AP is able to perform the SEF, the STA can indicate this option in the Msg1, with algorithm number set as 7 for SEF without PFS and as 8 for SEF with PFS.

Figure 15 AP State Machine

Since no additional parameter or element is required, the frame size is not affected.

4.3 Security Evaluation

To make sure the security functionality of the proposed SEF scheme are evaluated objectively, three parts of verification are given for a comprehensive evaluation. BAN Logic is employed to ensure that there is no logical loophole in logic can be made use of by attackers. And also the Scyther Tool is utilized to provide the formal verification of the protocol. The third part is the qualitative analysis of the SEF. By giving the reasons why those well-known typical attacks cannot be launched successfully, the overview of its security is provided. This section has the similar evaluation procedure as FKR. But since the protocols are different, how these tools are applied to SEF is quite different from the previous one.

4.3.1 Proof of Logic Correctness

BAN Logic is applied to verify the logical correctness of SEF to ensure the inexistence of logical loophole. The idealized forms of messages keep the encryption key and nonces included. The first two messages are encrypted by rIK as specified in ERP. The rIK is used here, instead of the rRK , because rIK is cryptographically derived from rRK and the rRK is already shared, but never used to encrypt messages. Thus the message form only includes rIK and rRK can be used in assumptions. Besides, the last message is encrypted by newly generated PTK. Nonces included in the messages are used to calculate keys and provide the freshness of messages. Also, AS is an indispensable role in this proof and serve as the bridge to connect STA and AP, because rRK is shared between STA and AS, not STA and AP, and AP shares absolutely secure link with AS. The postulates and symbol meanings were introduced in Chapter 3.

- Idealized Messages

$$S \rightarrow A: \{SNonce, A \xleftrightarrow{rIK} S\}_{rIK}$$

$$A \rightarrow S: \{ANonce, \{GTK, A \xleftrightarrow{PTK} S\}_{PTK}, A \xleftrightarrow{rIK} S\}_{rIK}$$

$$S \rightarrow A: \{Nonce, A \xleftrightarrow{PTK} S\}_{PTK}$$

- Goals

To confirm the possession of rRK , both AP and STA has to believe that each other also believe rRK shared between Authentication Server and STA. These goals allow AP to use $rMSK$ to derive PMK. They are denoted as follows:

$$A| \equiv S| \equiv AS \xleftrightarrow{rRK} S$$

$$S| \equiv A| \equiv AS \xleftrightarrow{rRK} S$$

To use the newly derived PTK in various encryptions, the both parties not only have to believe the possession of PTK and its parent key PMK, but believe that each other also believes in them.

$$A| \equiv A \xleftrightarrow{PMK} S, \quad A| \equiv S| \equiv A \xleftrightarrow{PMK} S$$

$$\begin{aligned}
S| &\equiv A \xleftrightarrow{PMK} S, & S| &\equiv A| \equiv A \xleftrightarrow{PMK} S \\
A| &\equiv A \xleftrightarrow{PTK} S, & A| &\equiv S| \equiv A \xleftrightarrow{PTK} S \\
S| &\equiv A \xleftrightarrow{PTK} S, & S| &\equiv A| \equiv A \xleftrightarrow{PTK} S
\end{aligned}$$

The Group Temporal Key has also to be delivered to STA and confirmed.

$$\begin{aligned}
A| &\equiv S| \equiv GTK \\
S| &\equiv GTK
\end{aligned}$$

- Assumptions

Before the verification begins, assumptions are made to initialize it. They first have to believe in the previously shared key, rRK. Note that these two parties here are STA and Authentication Server. Because the rRK is derived via 802.1x authentication between STA and Authentication Server.

$$\begin{aligned}
AS| &\equiv AS \xleftrightarrow{rRK} S \\
S| &\equiv AS \xleftrightarrow{rRK} S
\end{aligned}$$

Similarly, both parties, STA and AP, have to believe in the freshness of nonces distributed by each other.

$$\begin{aligned}
A| &\equiv \# \{SNonce\} \\
A| &\equiv \# \{Nonce\} \\
S| &\equiv \# \{ANonce\}
\end{aligned}$$

Because the GTK needs to be delivered, STA has to believe that AP has control over the GTK.

$$S| \equiv A| \Rightarrow GTK$$

- Verification

When AP receives the first message from STA, and relay it to AS because it believes the freshness of SNonce. And since $AS| \equiv AS \xleftrightarrow{rRK} S$, and rIK is derived from rRK as mentioned before. We can derive

$$AS| \equiv AS \xleftrightarrow{rIK} S$$

According to the message-meaning rule,

$$\frac{AS| \equiv AS \xleftrightarrow{rIK} S, AS \triangleleft \{SNonce, AS \xleftrightarrow{rIK} S\}}{AS| \equiv S| \sim \{SNonce, AS \xleftrightarrow{rIK} S\}}$$

According to belief properties,

$$\frac{AS| \equiv \# \{SNonce\},}{AS| \equiv \# \{SNonce, AS \xleftrightarrow{rIK} S\}}$$

According to the nonce-verification rule,

$$\frac{AS| \equiv \#\{SNonce, AS \xleftrightarrow{rIK} S\}, AS| \equiv S| \sim \{SNonce, AS \xleftrightarrow{rIK} S\}}{AS| \equiv S| \equiv \{SNonce, AS \xleftrightarrow{rIK} S\}}$$

According to belief properties,

$$\frac{AS| \equiv S| \equiv \{SNonce, AS \xleftrightarrow{rIK} S\}}{AS| \equiv S| \equiv AS \xleftrightarrow{rIK} S}$$

Theoretically, since rIK is uniquely derived from rRK, belief on rIK can be regarded as belief on rRK.

$$AS| \equiv S| \equiv AS \xleftrightarrow{rRK} S$$

Since AP trust what AS sends to it, it believes what AS believes.

$$A| \equiv S| \equiv AS \xleftrightarrow{rRK} S$$

After receive rMSK from AS, AP generates its ANonce and use it to calculate PMK and PTK. Thus

$$A| \equiv A \xleftrightarrow{PMK} S$$

$$A| \equiv A \xleftrightarrow{PTK} S$$

And for the second message sent by AP,

$$S \triangleleft \{ANonce, GTK, A \xleftrightarrow{PTK} S, A \xleftrightarrow{rIK} S\}$$

Since rIK is derived from rRK,

$$S| \equiv AS \xleftrightarrow{rIK} S$$

According to the message-meaning rule,

$$\frac{S| \equiv AS \xleftrightarrow{rIK} S, S \triangleleft \{ANonce, GTK, A \xleftrightarrow{PTK} S, AS \xleftrightarrow{rIK} S\}}{S| \equiv AS| \sim \{ANonce, GTK, A \xleftrightarrow{PTK} S, AS \xleftrightarrow{rIK} S\}}$$

According to belief properties,

$$\frac{S| \equiv \#\{ANonce\},}{S| \equiv \#\{ANonce, GTK, A \xleftrightarrow{PTK} S, AS \xleftrightarrow{rIK} S\}}$$

According to the nonce-verification rule,

$$\frac{S| \equiv \#\{ANonce, GTK, A \xleftrightarrow{PTK} S, AS \xleftrightarrow{rIK} S\}, S| \equiv AS| \sim \{ANonce, GTK, A \xleftrightarrow{PTK} S, AS \xleftrightarrow{rIK} S\}}{S| \equiv AS| \equiv \{ANonce, GTK, A \xleftrightarrow{PTK} S, AS \xleftrightarrow{rIK} S\}}$$

According to belief properties,

$$\frac{S| \equiv AS| \equiv \{ANonce, GTK, A \xleftrightarrow{PTK} S, AS \xleftrightarrow{rIK} S\}}{S| \equiv AS| \equiv AS \xleftrightarrow{rIK} S}$$

Similarly,

$$S| \equiv A| \equiv AS \xleftrightarrow{rRK} S$$

With ANonce sent from AP, STA is able derive PMK and PTK so that

$$S| \equiv A \xleftrightarrow{PMK} S$$

$$S| \equiv A \xleftrightarrow{PTK} S$$

With the similar rules we can derive

$$S| \equiv A| \equiv GTK$$

$$S| \equiv A| \equiv A \xleftrightarrow{PTK} S$$

According to jurisdiction rule,

$$\frac{S| \equiv A| \Rightarrow GTK, S| \equiv A| \equiv GTK}{S| \equiv GTK}$$

Similarly, PTK is also uniquely derived from PMK so that

$$S| \equiv A| \equiv A \xleftrightarrow{PMK} S$$

After processing the second message, STA constructs the third message and send it to AP.

With the same rules applied to Msg 3: $S \rightarrow A: \{Nonce, A \xleftrightarrow{PTK} S, \}_{PTK}$, it can be derived as follows:

$$A| \equiv S| \equiv A \xleftrightarrow{PMK} S$$

$$A| \equiv S| \equiv A \xleftrightarrow{PTK} S$$

$$A| \equiv S| \equiv GTK$$

After verifications described as above, the logical correctness of SEF is verified. It not only confirm the possession of parent keys in key hierarchy and prove that child session keys, such as PTK and GTK are successfully derived. The confirmation of possession of parent keys allows STA and AS authenticate each other's identity. And the possession of child session keys allows STA and AP encrypt data flow in the data transfer step.

4.3.2 Formal Verification

The verification of Scyther Tool is indispensable on verifying the security of SEF. The mechanism of Scyther Tool has been introduced in previous chapter. In this section, how the tool is applied to SEF is elaborated.

According to the Scyther Tool Manual [55], the scripts are written as follows:

```
//This is Secure Efficient FILS(SEF) Protocol
usertype SessionKey;
usertype GroupKey;
protocol sef(STA,AP,AS)
```

<pre> { role STA { fresh SN: Nonce; fresh N: Nonce; var AN: Nonce; var GTK: GroupKey; fresh PTK: SessionKey; send_1(STA,AP, STA, {STA,AP,AS,SN}k(STA,AS)); recv_!4(AP,STA,{ AP, GTK}PTK, {AN,AS}k(STA,AS)); send_5(STA,AP, {N, AN}PTK); claim(STA, Secret,N); claim(STA, Secret,AN); claim(STA, Secret,SN); claim(STA, Secret,GTK); claim(STA, Secret,PTK); claim(STA, Alive); claim(STA, Weakagree); claim(STA, Niagree); claim(STA, Nisynch); } } </pre>	<pre> var T1: Ticket; var T2: Ticket; var PTK: SessionKey; var N: Nonce; recv_1(STA,AP, STA,T1); send_2(AP,AS, { AP, STA,AN,T1 } k(AP,AS)); recv_3(AS,AP, {AS,PTK,SN,T2}k(AP,AS)); send_!4(AP,STA,{ AP, GTK}PTK, T2); recv_5(STA,AP, {N, AN}PTK); claim(AP, Secret,N); claim(AP, Secret,AN); claim(AP, Secret,SN); claim(AP, Secret,GTK); claim(AP, Secret,PTK); claim(AP, Alive); claim(AP, Weakagree); claim(AP, Niagree); claim(AP, Nisynch); } </pre>
<pre> role AP { fresh AN: Nonce; fresh GTK: GroupKey; var SN: Nonce; </pre>	<pre> role AS { fresh PTK: SessionKey; var SN,AN: Nonce; recv_2(AP,AS, { AP, STA, AN,{STA,AP, AS,SN}k(STA,AS)} k(AP,AS)); </pre>

```

send_3(AS,AP, {AS,PTK,
SN,{AN,AS}k(STA,AS)}k(AP,AS) );

```

```

claim(AS, Secret,AN);

```

```

claim(AS, Secret,SN);

```

```

claim(AS, Secret,PTK);

```

```

claim(AS, Alive);

```

```

claim(AS, Weakagree);

```

```

claim(AS, Niagree);

```

```

claim(AS, Nisynch);

```

```

}

```

```

}

```

Scyther results : autoverify					
Claim				Status	Comments
sef	STA	sef,STA1	Secret PTK	Ok	No attacks within bounds.
		sef,STA2	Secret N	Ok	No attacks within bounds.
		sef,STA3	Secret SN	Ok	No attacks within bounds.
		sef,STA4	Secret GTK	Ok	No attacks within bounds.
		sef,STA5	Secret AN	Ok	No attacks within bounds.
		sef,STA6	Alive	Ok	No attacks within bounds.
		sef,STA7	Weakagree	Ok	No attacks within bounds.
		sef,STA8	Niagree	Ok	No attacks within bounds.
		sef,STA9	Nisynch	Ok	No attacks within bounds.
AP	sef,AP1	sef,AP1	Secret GTK	Ok	No attacks within bounds.
		sef,AP2	Secret AN	Ok	No attacks within bounds.
		sef,AP3	Secret N	Ok	No attacks within bounds.
		sef,AP4	Secret PTK	Ok	No attacks within bounds.
		sef,AP5	Secret T2	Ok	No attacks within bounds.
		sef,AP6	Secret T1	Ok	No attacks within bounds.
		sef,AP7	Secret SN	Ok	No attacks within bounds.
		sef,AP8	Alive	Ok	No attacks within bounds.
		sef,AP9	Weakagree	Ok	No attacks within bounds.
		sef,AP10	Niagree	Ok	No attacks within bounds.
		sef,AP11	Nisynch	Ok	No attacks within bounds.
AS	sef,AS1	sef,AS1	Secret PTK	Ok	No attacks within bounds.
		sef,AS2	Secret AN	Ok	No attacks within bounds.
		sef,AS3	Secret SN	Ok	No attacks within bounds.
		sef,AS4	Alive	Ok	No attacks within bounds.
		sef,AS5	Weakagree	Ok	No attacks within bounds.
		sef,AS6	Niagree	Ok	No attacks within bounds.
		sef,AS7	Nisynch	Ok	No attacks within bounds.

Done.

Figure 16 Scyther Tool Verification Result of SEF

After running the auto-verification and correcting errors, the report is generated as in Figure 20. All claims regarding the properties of three parties have been verified as successful. Note that PMK is not mentioned in the

script because it is not used to encrypt any message, and the child key, PTK, is used to denote the PMK&PTK hierarchy. And since rIK is derived from rRK which is the shared common key between STA and AS, the rIK&rRK hierarchy is denoted as $k(\text{Party name}, \text{Party name})$ (e.g. $k(\text{STA}, \text{AS})$) which is the common reference of shared common key. The meaning of each claims are introduced in previous chapter.

4.3.3 Security Analysis

- Key Reinstallation Attack

SEF is designed to defend against KRA. It is necessary to analyze how its mechanism works to succeed. There are two possible way to attack. One is on STA and one is on AP. By defining the state machines of STA and AP, ambiguity of standard can be avoided to ensure the stable operations. For example, the core request of the KRA is to make STA believe that the second received Msg 2 is also a legitimate message, which would lead to the next key installation. However, with the state machine, STA is able to distinguish the retransmitted the valid, but not forcing key reinstallation requests and strictly follow the secure procedure without installing keys twice. Because STA would have Boolean value as flag to indicate the installing state of keys and STA only accepts the first valid Msg 2 from AP. This move not only ensures that STA would not be susceptible to reinstallation request, but also decrease the overhead for calculating and installing new keys. The state machine for AP has the similar functions which have been presented before.

Also, SEF scheme is analyzed to be immune to other attacks. The scheme is generally secure, because all three messages are protected by adding MIC with shared keys and both these keys and newly derived keys are never transferred on the air. For example, the first MIC is calculated with rIK which parent key on the key hierarchy is rRK so that rRK remains unrevealed. In the followings, a series of typical attacks are analyzed, and the reasons why they cannot be launched successfully are given.

- DoS Attack

DoS Attack on any of these three messages are not possible. For the first message, even if the attackers flood it, the AP would not be bothered by it because AP simply relays Msg 1, and AS would not accept them because the MIC cannot be correct. The second and third messages are both protected by KEK so that the attackers also cannot fool the victim to believe them.

- Replay Attack

All messages contain the sequence number. So long as the sequence numbers are higher than before, the parties accept them. Thus these messages are all protected by MIC so that the sequence numbers cannot be changed after being sent onto the air. That is why KRA needs to take the MitM position to control the retransmission and imitate replay attack with the legitimate retransmitted message. Replaying the same message can be easily detected and discarded.

- Man in the Middle Attack

That KRA needs to take the MitM position does not mean that KRA can be categorized as a MitM attack. Because if the attackers can do nothing but faithfully relay the messages, instead of altering messages or injecting any information into messages. Although KRA does intercept the replying message to lure the retransmitted message, it eventually relays both the original and retransmitted messages to the victim and does not change any single bit of them. Thus, technically, KRA is not one of the MitM attacks. Usually, simply de-authenticating each other and terminating the connection without disclosing any useful information do not count as an effective MitM attack, because if the victims are not able to establish connection via the MitM, they can simply bypass this unreliable proxy and establish the connection by themselves so long as the broadcast beacon is not vulnerable to flooded invalid messages. Hence, in SEF, MitM cannot change or alter messages without being detected so that effective MitM attacks cannot prevail.

- Impersonation Attack

Impersonation attack is an attack in which an adversary successfully assumes the identity of one of the legitimate parties in the system or in a communication protocol, thus receiving messages directed to the node it fakes[55]. The reason why impersonation attacks succeed depends on that the attackers can obtain the trust from the network. In SEF, STA and AS already share a common rRK because they complete 802.1X authentication before or the STA has already registered in AS. Thus any attackers wish to impersonate the STA will fail, because they do not have a valid rRK. As for impersonating AP, the communications between AP and AS are assumed to be absolutely secure, and the protocol used is out of scope, which results in that impersonating AP is out of scope.

- Compromised Attack

Although obtaining key is a quite difficult and resource-intensive process for attackers, it is still possible for attackers compromise the STA and have the common key rRK. However, according to rRK has its life time, and all child keys derived from it must have no-longer-than-rRK's life time. Even if all keys generated from rRK are known to attackers, this key set can only decrypt the messages sent to the STA itself, which means this common key set is only used for this STA and AP within the limited life time as specified. Compromised attack does cause damage to the network, but it can be limited to the STA.

4.4 Performance Evaluation

4.4.1 Time and Energy Consumption Comparison

SEF aims at being adapted to 802.11ah networks. Thus the performance of this protocol is always the paramount important aspect. To prove that changed version does not increase the overhead, calculations of time and energy are provided. Since it is not a technical calculation comparison and besides these operations, other operations are mostly the same, only costs of several typical cyptographical operations are included, regardless of the time and energy for constructing packets, transmission, receiving, installing key sets, or running verification algorithms and so forth. Table 7 lists the operations comparison of two schemes. The operations listed are what both parties have to finish before sending the corresponding messages. In other words, they are what both parties have to prepare before sending. And in Table 8, additional parameters are given compared

with Table 2. Making use of values in Table 2, the final results can be calculated as in Table 9. The calculation is also based on the same conditions as in Chapter 3. The result shows that the performance is a little bit better than before (17.2% for time and 22.8 for energy), mainly due to the reallocated calculation tasks. Note that the result is only for reference and it is not for strictly quantitative comparison. Since the FILS scheme focuses on the subsequent rounds of authentication, which does not use PMK, PMK is not calculated in the FILS scheme. However, the SEF scheme focuses on all the rounds of authentication, especially the first one, which definitely needs to use PMK. Thus it takes PMK and other parent keys into account unlike the calculations in Chapter 3. This is also the reason why these two proposals are not compared in the same table or graph. Simply because they are not designed to serve on the same position, and they are not comparable.

Table 7 Operations Comparison

Messages	FILS	SEF
S→A	SNonce generation, rIK derivation	SNonce generation, rIK derivation
A→S	ANonce generation, PMK derivation, PTK derivation, GTK derivation	ANonce generation, GTK derivation
S→A	rMSK derivation PMK derivation, PTK derivation, Key-auth hash,	PMK derivation, PTK derivation, Key-auth hash
A→S	Key-auth hash,	N.A

Table 8 Length of Parameters

Parameters	Length(bits)
PMK	256
PTK	512
Nonce	256
GTK	128
MAC address	48
Sequence Number	128
rRK	512
rIK	512
rMSK	512

Table 9 Performance Comparison

Message	Time Delay(us)		Energy Consumption(mJ)	
	FILS	SEF	FILS	SEF
S→A	127.95	127.95	150.59	150.59
A→S	183.03-30	55.08+80	216.23-80	65.64+40
S→A	313.67-80	216.52+20	371.15-60	257.88+30
A→S	88.57	N.A	107.29	N.A
Total	603.22	499.55	705.26	544.11

4.4.2 Performance Evaluation against Unknown Attacks

The last evaluation is for unknown attack model. This simulation experiment reveals the ability of both schemes against unknown attacks based on same platform as Chapter 3, Matlab 2014b. There are four experiments carried out on Matlab for FILS and SEF regarding time and energy. The theory behind it is the same as FKR. But the results are different because of the different operations or steps. The results are illustrated in Figure 21. The tendency is clear to see, which is with the increasing probability of unknown attacks, the time and energy consumption increase as well. Because more unknown attacks emerges during the simulation mean higher probability to interrupt he protocol within the period. The both ends of the curve also have the similar meanings. If the unknown attack probability is zero, there is no attack launched so that the time and energy value are the same as those in the Table 7. If the attack probability is 100%, the protocol can barely finish the authentication process so that the consumption of both reaches the highest point on average. It is clear that the differences between two schemes are not very distinct. However, even the time and energy savings of transmission and receiving have not been taken into account, it still demonstrates that the improvement on security at least does not degrade the performance.

Figure 17 Average Time Delay and Average Energy Consumption against Unknown Attacks

4.5 Summary

SEF scheme is introduced in this chapter. Firstly, the how SEF was designed is introduced, especially about the ability against KRA and other attacks. And the followings give the proof of security by proving the logical correctness and security properties with Scyther Tool. Thirdly, SEF has been proven to be resistant to other typical attacks. The last part is about the rough performance evaluation, which clearly demonstrates that SEF with higher security does not degrade the original performance and could be even more efficient than FILS.

Chapter 5. Conclusions

5.1 Conclusions

The research work that supports this thesis has been done based on two IEEE standards, 802.11ah and 802.11ai. Although the FILS in the IEEE 802.11ai is designed for all subsequent IEEE 802.11 standards, no researcher so far has committed the work on the exclusive enhancement of the implementation of FILS in 802.11ah networks. After exploring all the papers mentioned in Chapter 2, I had an overview on this research area and chose to work on the combination of these two standards. While avoiding all the shortcomings of other researchers' works analyzed in Chapter 2, I have proposed my own two schemes and polished them till they are secure and efficient enough.

Chapter 3 then presents the contribution of the FKR scheme, which aims at achieving faster re-authentications than simply using the original FILS. With two messages exchanged, the keying materials are exchanged and the common key is derived. Besides the function, the logic correctness is guaranteed by the proof BAN Logic. Security properties are verified by the simulation tool, scyther. And the most paramount important comparison between FKR and FILS is calculated. The result demonstrates that FKR needs less time and energy to complete when the rounds increase. Finally, the unknown attacks simulation indicates that FKR is less prone to unknown attacks than FILS.

Last but not the least, in order to prevent the recent revealed attack, KRA, the new and more secure scheme, SEF, has been proposed. It is designed to be against KRA and other typical attacks. The security of it is guaranteed by BAN Logic, Scyther Tool and the new retransmission mechanism. Meanwhile, the performance of it does not degrade with the increasing security, which is presented as performance evaluation. Also, the ability against the unknown attacks also outperforms the original FILS.

Although these two proposals are both inherited from the FILS scheme, they are not intended to serve as the same functions. After security evaluations and performance evaluation, it is clear that they are proved to be efficient, secure, and potential to be implemented.

Recommendations for Further Research

Although the IEEE 802.11ah did not define the detailed authentication and association procedure and use the legacy authentication protocol by default [56], the specification [6] did define AC mechanism as mentioned before. Since the AC step is performed before the formal authentication and association, the re-authentication and re-association may take advantage of the previous parameters. For example, STAs would be allocated with AIDs after the first round authentication and association. This AID can be bounded with some parameters to accelerate the next authentication process, such as the PMKID if the next PMK is going to be derived, and current PTK if the next PTK is going to be derived. For STAs whose AID is bounded with additional information, the authentication control process may give more priority to them, because the additional information brings faster authentication and association. With this mechanism, the AC process can be accelerated due to the less

contentions resulted from hierarchy of STAs. Apart from that, the formal authentication and association part is also accelerated because of the cached additional identity information.

The typical application scenario of 802.11ai is that thousands of mobile users enter and leave the coverage of multi-AP area, such as a railway station, and a stadium. Since mobile devices would not stay with one AP all the time, the seamless handoff is critical to user experience, handoff is a significant topic in 802.11ai. [57] summarizes features, open issues, and research topics of some contemporary standards. It is mentioned that for the initial functionality of achieving quick, efficient and robust handoff, the amendment IEEE 802.11ai known as Fast Initial Link Setup is aimed for a complete handoff within 46 ms that includes discovery of new AP, and to authenticate and configure the user. [58] introduces 802.11ai handover process and reveals the reason that the faster handoff is achieved. The reason is similar as the initial link setup. [29] and [30] have introduced the faster handoff scheme proposed by 802.11ai, while none of them has investigated the details or try to make improvement because they have their own focuses. The handoff process shares some common places with initial link setup process, whereas the difference is that the previous AP has the information that may be used for the authentication with new AP to achieve faster link setup process. Thus the handoff process can explored to make improvements.

Author's Publication

Lyuye Zhang and Maode Ma, "Performance and Security Enhancements to Fast Initial Link Setup in IEEE 802.11ah Wireless Networks," *IEEE International Conference on Communications (ICC) 2018*, accepted.

Bibliography

- [1] T. Adame, A. Bel, B. Bellalta, J. Barcelo, and M. Oliver, "IEEE 802.11 ah: the WiFi approach for M2M communications," *IEEE Wireless Communications*, vol. 21, no. 6, pp. 144-152, 2014.
- [2] W. Sun, M. Choi, and S. Choi, "IEEE 802.11 ah: A long range 802.11 WLAN at sub 1 GHz," *Journal of ICT Standardization*, vol. 1, no. 1, pp. 83-108, 2013.
- [3] IEEE Standard for Information technology--Telecommunications and information exchange between systems - Local and metropolitan area networks--Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 1: Fast Initial Link Setup," in *IEEE Std 802.11ai-2016 (Amendment to IEEE Std 802.11-2016)* , vol., no., pp.1-164, Dec. 30 2016
- [4] V. Jones and H. Sampath, "Emerging technologies for WLAN," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 141-149, 2015.
- [5] M. Vanhoef and F. Piessens, "Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2," in *Proceedings of the ACM Conference on Computer and Communications Security*, Dallas, TX, USA, 2017, vol. 30.
- [6] "IEEE Standard for Information technology--Telecommunications and information exchange between systems - Local and metropolitan area networks--Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: Sub 1 GHz License Exempt Operation," *IEEE Std 802.11ah-2016 (Amendment to IEEE Std 802.11-2016, as amended by IEEE Std 802.11ai-2016)*, pp. 1-594, 2017.
- [7] "IEEE Standard for information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 6: Medium Access Control (MAC) Security Enhancements," *IEEE Std 802.11i-2004*, pp. 1-190, 2004.
- [8] S. Aust, R. V. Prasad, and I. G. Niemegeers, "IEEE 802.11 ah: Advantages in standards and further challenges for sub 1 GHz Wi-Fi," in *Communications (ICC), 2012 IEEE International Conference on*, 2012, pp. 6885-6889: IEEE.
- [9] D. Bankov, E. Khorov, and A. Lyakhov, "The Study of the Centralized Control Method to Hasten Link Set-up in IEEE 802.11 ah Networks," in *European Wireless 2015; 21th European Wireless Conference; Proceedings of*, 2015, pp. 1-6: VDE.
- [10] H. Wang, "Supporting Authentication/Association for Large Number of Stations," ed, 2012.
- [11] A. Safonov, A. Lyakhov, and E. Khorov, "Channel switch time distribution in ECMA-368 networks," in *Personal, Indoor and Mobile Radio Communications, 2008. PIMRC 2008. IEEE 19th International Symposium on*, 2008, pp. 1-5: IEEE.

- [12] Z. Zhang, R. W. Pazzi, and A. Boukerche, "Design and evaluation of a fast authentication scheme for WiFi-based wireless networks," in World of Wireless Mobile and Multimedia Networks (WoWMoM), 2010 IEEE International Symposium on a, 2010, pp. 1-6: IEEE.
- [13] C. Rigney, S. Willens, A. Rubens, and W. Simpson, "Remote authentication dial in user service (RADIUS)," 2070-1721, 2000.
- [14] C. He and J. C. Mitchell, "Analysis of the 802.11 i 4-Way Handshake," in Proceedings of the 3rd ACM workshop on Wireless security, 2004, pp. 43-50: ACM.
- [15] Y.-S. Kang, K. Oh, B.-H. Chung, and K.-I. Chung, "Apparatus for dynamically managing group transient key in wireless local area network system and method thereof," ed: Google Patents, 2009.
- [16] W. A. Arbaugh, "An inductive chosen plaintext attack against WEP/WEP2," [http://www. cs. umd. edu/~ waa/attack/frame. htm](http://www.cs.umd.edu/~waa/attack/frame.htm), 2001.
- [17] X. Zha and M. Ma, "Security improvements of IEEE 802.11 i 4-way handshake scheme," in Communication Systems (ICCS), 2010 IEEE International Conference on, 2010, pp. 667-671: IEEE.
- [18] S. Zhao, C. A. Shoniregun, and C. Imafidon, "Addressing the vulnerability of the 4-way handshake of 802.11 i," in Digital Information Management, 2008. ICDIM 2008. Third International Conference on, 2008, pp. 351-356: IEEE.
- [19] J. Liu, X. Ye, J. Zhang, and J. Li, "Security verification of 802.11 i 4-way handshake protocol," in Communications, 2008. ICC'08. IEEE International Conference on, 2008, pp. 1642-1647: IEEE.
- [20] D. Yadav and A. Sardana, "Enhanced 3-way handshake protocol for key exchange in IEEE 802. 11i," in Electronics Computer Technology (ICECT), 2011 3rd International Conference on, 2011, vol. 6, pp. 132-135: IEEE.
- [21] C.-S. Park, "Two-way Handshake protocol for improved security in IEEE 802.11 wireless LANs," Computer Communications, vol. 33, no. 9, pp. 1133-1140, 2010.
- [22] R. Singh and T. P. Sharma, "A key refreshing technique to reduce 4-way handshake latency in 802.11 i based networks," in Computer and Communication Technology (ICCCT), 2013 4th International Conference on, 2013, pp. 157-162: IEEE.
- [23] F. De Rango, D. C. Lentini, and S. Marano, "Static and dynamic 4-way handshake solutions to avoid denial of service attack in Wi-Fi protected access and IEEE 802.11 i," EURASIP Journal on Wireless Communications and Networking, vol. 2006, no. 1, p. 047453, 2006.
- [24] X. Li, F. Bao, S. Li, and J. Ma, "FLAP: An efficient WLAN initial access authentication protocol," IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 2, pp. 488-497, 2014.
- [25] W. Diffie and M. Hellman, "New directions in cryptography," IEEE transactions on Information Theory, vol. 22, no. 6, pp. 644-654, 1976.
- [26] J.-C. Chen and Y.-P. Wang, "Extensible authentication protocol (EAP) and IEEE 802.1 x: tutorial and empirical experience," IEEE communications magazine, vol. 43, no. 12, pp. suppl.

- 26-supl. 32, 2005.
- [27] B. Bellalta, "IEEE 802.11 ax: High-efficiency WLANs," *IEEE Wireless Communications*, vol. 23, no. 1, pp. 38-46, 2016.
 - [28] W. Sun et al., "Wi-Fi could be much more," *IEEE Communications Magazine*, vol. 52, no. 11, pp. 22-29, 2014.
 - [29] E. H. Ong, "Performance analysis of fast initial link setup for IEEE 802.11 ai WLANs," in *Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium on*, 2012, pp. 1279-1284: IEEE.
 - [30] S. Lee, B. H. Chung, "Denial of Service Attack against IEEE 802.11 WLAN Fast Initial Link Setup Technology," *International Journal of Computer Science and Electronics Engineering (IJCSEE)*, vol. 3, no. 4, pp. 335-337, 2015.
 - [31] Z. Cao, B. He, Y. Shi, Q. Wu, and G. Zorn, "EAP Extensions for the EAP Re-authentication Protocol (ERP)," 2070-1721, 2012.
 - [32] K. G. Paterson, B. Poettering, and J. C. Schuldt, "Plaintext recovery attacks against WPA/TKIP," in *International Workshop on Fast Software Encryption*, 2014, pp. 325-349: Springer.
 - [33] E. Tews and M. Beck, "Practical attacks against WEP and WPA," in *Proceedings of the second ACM conference on Wireless network security*, 2009, pp. 79-86: ACM.
 - [34] S. Viehböck, "Brute forcing wi-fi protected setup," *Wi-Fi Protected Setup*, vol. 9, 2011.
 - [35] E. N. Lorente, C. Meijer, and R. Verdult, "Scrutinizing WPA2 Password Generating Algorithms in Wireless Routers," in *WOOT*, 2015.
 - [36] M. Vanhoef and F. Piessens, "Predicting, Decrypting, and Abusing WPA2/802.11 Group Keys," in *USENIX Security Symposium*, 2016, pp. 673-688.
 - [37] P.-A. Fouque, G. Martinet, F. Valette, and S. Zimmer, "On the Security of the CCM Encryption Mode and of a Slight Variant," in *Applied Cryptography and Network Security*, 2008, pp. 411-428: Springer.
 - [38] A. Joux, "Authentication failures in NIST version of GCM," *NIST Comment*, p. 3, 2006.
 - [39] C. He, M. Sundararajan, A. Datta, A. Derek, and J. C. Mitchell, "A modular correctness proof of IEEE 802.11 i and TLS," in *Proceedings of the 12th ACM conference on Computer and communications security*, 2005, pp. 2-15: ACM.
 - [40] M. Bellare and P. Rogaway, "Entity Authentication and Key Distribution," in *Crypto*, 1993, vol. 93, pp. 232-249: Springer.
 - [41] M. Vanhoef and F. Piessens, "Advanced Wi-Fi attacks using commodity hardware," in *Proceedings of the 30th Annual Computer Security Applications Conference*, 2014, pp. 256-265: ACM.
 - [42] M. Burrows, M. Abadi, and R. M. Needham, "A logic of authentication," in *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 1989, vol. 426, no. 1871, pp. 233-271: The Royal Society.
 - [43] B. Blanchet, "An Efficient Cryptographic Protocol Verifier Based on Prolog Rules," in *csfw*,

2001, vol. 1, pp. 82-96.

- [44] A. Armando et al., "The AVISPA tool for the automated validation of internet security protocols and applications," in International conference on computer aided verification, 2005, pp. 281-285: Springer.
- [45] C. J. Cremers, "The scyther tool: Verification, falsification, and analysis of security protocols," in CAV, 2008, vol. 8, pp. 414-418: Springer.
- [46] C. Cremers, "Feasibility of multi-protocol attacks," in Availability, Reliability and Security, 2006. ARES 2006. The First International Conference on, 2006, pp. 8 pp.-294: IEEE.
- [47] C. Cremers and S. Mauw, "Operational semantics of security protocols," in Scenarios: Models, Transformations and Tools: Springer, 2005, pp. 66-89.
- [48] C. Cremers and S. Mauw, Operational semantics and verification of security protocols. Springer Science & Business Media, 2012.
- [49] G. Lowe, "A hierarchy of authentication specifications," in Computer security foundations workshop, 1997. Proceedings., 10th, 1997, pp. 31-43: IEEE.
- [50] R. Bird et al., "Systematic design of a family of attack-resistant authentication protocols," IEEE Journal on Selected Areas in Communications, vol. 11, no. 5, pp. 679-693, 1993.
- [51] R. M. Needham and M. D. Schroeder, "Using encryption for authentication in large networks of computers," Communications of the ACM, vol. 21, no. 12, pp. 993-999, 1978.
- [52] G. Lowe, "An attack on the Needham-Schroeder public-key authentication protocol," Information processing letters, vol. 56, no. 3, pp. 131-133, 1995.
- [53] E. Khorov, A. Lyakhov, A. Krotov, and A. Guschin, "A survey on IEEE 802.11 ah: An enabling networking technology for smart cities," Computer Communications, vol. 58, pp. 53-69, 2015.
- [54] H. Rifa-Pous and J. Herrera-Joancomartí, "Computational and energy costs of cryptographic algorithms on handheld devices," Future internet, vol. 3, no. 1, pp. 31-48, 2011.
- [55] L. Tamilselvan and D. V. Sankaranarayanan, "Prevention of impersonation attack in wireless mobile ad hoc networks," International Journal of Computer Science and Network Security (IJCSNS), vol. 7, no. 3, pp. 118-123, 2007.
- [56] M. Park, "IEEE 802.11 ah: sub-1-GHz license-exempt operation for the internet of things," Communications Magazine, IEEE, vol. 53, no. 9, pp. 145-151, 2015.
- [57] D. Macha Sarada and A. Damodaram, "Review of 802.11 based Wireless Local Area Networks and Contemporary Standards: Features, Issues and Research Objectives," Global Journal of Computer Science and Technology, vol. 16, no. 4, 2016.
- [58] S. Fujikami, T. Sumi, R. Yagiu, and Y. Nagai, "Fast Device Discovery for Vehicle-to-Pedestrian communication using wireless LAN," in Consumer Communications and Networking Conference (CCNC), 2015 12th Annual IEEE, 2015, pp. 35-40: IEEE.