Universiteit Leiden

Master Computer Science

Formal Analysis and Improvement of the PKMv3 and 5G AKA Protocols

Name:	Yuanmin Xu
Student ID:	s1929313
Date:	25/05/2019
Specialisation:	Computer Science and Advanced Data Analytics
1st supervisor:	Marcello Bonsangue
2nd supervisor:	Alfons Laarman

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

The mobile communication technology has become one of the most widely used technology and is developing rapidly. Because of the greater demands for data communication and multimedia services, the 3G network has been gradually replaced by the 4G networks, with the advantages of faster spreading and wider covering areas. WiMAX (Worldwide Interoperability for Microwave Access) is one of the standards of the 4G networks. It provides higher confidentiality and security in communication. Privacy Key Management (PKM) Protocol is one of the security protocol of the WiMAX. It uses key-exchange mechanism to encrypt messages in order to ensure the confidentiality of data transmission. So that messages messages cannot easily be eavesdropped, tampered or forged by the intruder. The third version of the PKM (PKMv3) protocol was released because the first two versions cannot effectively resist the attacks.

The fifth generation (5G) of the wireless networks, which enhances the mobile broadband, massive machine type communications and ulter-reliable and low-latency communications, is now under construction and nearly completion to put into use. It is of great importance to guarantee the security of the the 5G networks. The International Standards Organization of the 3rd Generation Partneership Project (3GPP) group has standards and released the new version of the 5G Authentication and Key Agreement (AKA) protocol, which has better privacy performances, to implement the mutual authentication of the substribers and networks.

In this thesis, we first formally analyze the PKMv3 protocol using the Communicating Sequential Processes (CSP) method. We introduce three communication entities, i.e., the server, mobile station (MS) and the base station (BS) and model them as processes in our framework. Moreover, intruders who have capabilities of intercepting, faking and overhearing, are also designed as a process. We employ the Process Analysis Toolkit (PAT), a useful and efficient model checker for CSP, to implement the entire interaction system and obtain the simulation of the protocol. Six non-trivial properties are extracted and verified using LTL formula and assertions. With respect to the verification results, we discuss some cases where intruders may successfully attack the system and propose some approaches to the corresponding threats.

And we also give an overview of the 5G AKA protocol, which includes all parties, i.e., the user equipment (UE), the serving network (SN), the home network (HM) as well as the credential repository resides in the HM. The formal description and verification of the 5G AKA protocol model are also using CSP and implemented in PAT. On the basis of the verification results, some vulnerabilities are found and we proposed useful approaches to improve the performance of the protocol. Consequently, through our framework, a better understanding of the PKMv3 and 5G AKA protocols can be achieved.

Acknowledgements

This master thesis was done at LIACS. First and foremost, I would like to thank my supervisors, Prof. Marcello Bonsangue and Prof. Alfons Laarman, for great knowledge of the formal methods, the patience and encouragement while doing my master project.

I would also thank to the professors giving me lectures at Leiden University, from whom I harvested a lot and broadened my horizons to the computer science technologies.

I would like to thank my friends, who brought me happiness and made me feel at home. Finally, I would like to thank my family for there support and love no matter when and where. Thank you all to help me to accomplish this master thesis.

Contents

Abstract i

Acknowledgements ii

List of Figures vi

List of Tables vii

Abbreviations viii

1 Introduction 1
- 1.1 Related work . . . 3
- 1.2 Contributions . . . 4
- 1.3 Thesis Overview . . . 5

2 Background 6
- 2.1 Introduction to Keys of the PKMv3 Protocol . . . 6
- 2.2 Transmission mechanism of PKMv3 Protocol . . . 7
  - 2.2.1 Certificate reply and response . . . 8
  - 2.2.2 Initialization . . . 8
  - 2.2.3 AK authorization . . . 8
  - 2.2.4 Transmission of security association . . . 9
  - 2.2.5 Exchange of TEK . . . 10
- 2.3 Authentication mechanism of 5G AKA . . . 11
  - 2.3.1 Overall Architecture . . . 11
  - 2.3.2 Normal Description of the 5G-AKA protocol . . . 12
- 2.4 Introduction of CSP . . . 14
- 2.5 Introduction of PAT . . . 15

3 Modeling the PKMv3 Protocol Using CSP 17
- 3.1 Preliminaries . . . 17
  - 3.1.1 Notations . . . 17
  - 3.1.2 Assumptions . . . 18
  - 3.1.3 Sets . . . 18
- 3.2 Channels . . . 19
- 3.3 Model of the PKMv3 Protocol . . . 20
  - 3.3.1 Timer . . . 21
  - 3.3.2 Server . . . 21

3.3.3 Mobile Station . . . 22
3.3.4 Base Station . . . 24
3.3.5 Intruder . . . 25
3.3.6 The whole system . . . 26

4 Implelentation and Simulation of the PKMv3 Protocol 28
4.1 Preliminaries . . . 29
4.1.1 Notations . . . 29
4.1.2 Sets and Functions . . . 30
4.1.3 Channels . . . 31
4.2 Timer . . . 31
4.3 Mobile Station . . . 32
4.3.1 MS with Server . . . 32
4.3.2 MS with BS . . . 32
4.4 Base Station . . . 35
4.4.1 BS with Server . . . 35
4.4.2 BS with MS . . . 35
4.5 Server . . . 37
4.6 Intruder . . . 38
4.7 System . . . 39
4.8 Simulation . . . 39

5 Verification and Analysis of the PKMv3 Protocol 41
5.1 Verification . . . 41
5.1.1 Deadlock Freedom . . . 42
5.1.2 Connectivity . . . 42
5.1.3 Consistency . . . 43
5.1.4 Delay Freedom . . . 43
5.1.5 Safety . . . 44
5.1.6 Secrecy Violation . . . 44
5.1.7 Verification Results . . . 45
5.2 Analysis of Attacks to the PKMv3 Protocol . . . 46
5.2.1 Man-in-the-Middle Attack . . . 46
5.2.2 Replay Attack . . . 50

6 Formal Modeling of the 5G AKA Protocol 51
6.1 Preliminaries . . . 51
6.1.1 Notations . . . 51
6.1.2 Sets and Functions . . . 52
6.1.3 Channels . . . 52
6.1.4 Assumptions . . . 52
6.2 User Equipment . . . 53
6.3 Serving Network . . . 53
6.4 Home Network . . . 54
6.5 APRF . . . 55
6.6 Intruder . . . 55
6.7 System . . . 56

7 Verification and Analysis of the 5G AKA Protocol **57**
7.1 Verification 57
7.1.1 Deadlock Freedom 57
7.1.2 Connectivity 58
7.1.3 Secrecy 58
7.1.4 Verification Results 59
7.2 Analysis of the 5G AKA Protocol 59

8 Conclusion and Future Work **61**

Bibliography **63**

List of Figures

2.1 Communication Mechanism 7
2.2 Overall architecture of 5G AKA obtained from [1] 11
2.3 Authentication mechanism of 5G AKA protocol 12

3.1 Communications using channels among parties 20

4.1 Simulation of MS/BS and Server 40
4.2 Simulation of MS and BS 40

5.1 Deadlock Freedom Result 45
5.2 The verification results of the five properties 45

7.1 Verification Results 60
7.2 Improved Verification Results 60

List of Tables

2.1 Definations of Notations of the 5G AKA protocol 13
2.2 Values of Messages in the 5G AKA protocol 14

4.1 Message Contents 30

Abbreviations

PKM	**P**rivacy **K**ey **M**anagement
MS	**M**obile **S**tation
BS	**B**ase **S**tation
AK	**A**uthorization **K**ey
PMK	**P**air-wise **M**aster **K**ey
TEK	**T**raffic **E**ncrption **K**ey
CMAC	**C**ipher-based **M**essage **A**uthentication **C**ode
SA	**S**ecurity **A**ssociation
LTL	**L**inear **T**emporal **L**ogic
CSP	**C**ommunicating **S**equential **P**rocesses
PAT	**P**rocess **A**nalysis **T**oolkit
DoS	**D**enial-of-**S**ervice
5G AKA	5G **A**uthentication and **K**ey **A**greement
ARPF	**A**uthentication credential **R**epository and **P**rocessing **F**unction
AUSF	**A**uthentication **S**erver **F**unction
UE	**U**ser **E**quipment
SN	**S**erving **N**etwork
HN	**H**ome **N**etwork
SEAF	**S**ecurity **A**rchor **F**unction
SIDF	**S**ubscriber **I**dentity **D**e-concealing **F**unction
SUCI	**S**ubscription **C**oncealed **I**dentifier
SUPI	**S**ubscription **P**ermanent **I**dentifier

Chapter 1

Introduction

With the rapid development of wireless mobile communication technology and multimedia communication services, the fourth generation (4G) mobile communication [2], which can better adapt to the mobile data communication and computing, has emerged. It has super high data transmission speed and larger network coverage. There are many standards for 4G networks, one of which is WiMAX. WiMAX (Worldwide Interoperability for Microwave Access) [3], also known as the IEEE 802.16 Wireless MAN, is a new broadband wireless access technology, which can provide high-speed internet-oriented connection. Although WiMAX has several advantages in the network coverage and transmission speed, its mobility is limited and it is difficult to meet the seamless links in high-speed networks. Therefore, the latest version of 802.16m [4], which can effectively solve the above shortcomings has been put forward.

WiMax has an MAC layer containing a security sublayer, which uses a protocol called Privacy Key Management (PKM) Protocol [5] to be responsible . The security of key distribution scheme, like key parameters synchronization and authentication between the mobile station (MS) and the base station (BS) is provided by the PKM protocol. The PKM procotol has two previous versions named as PKMv1 and PKMv2 protocols. With more users and frequent transmission, some vulnerabilities are exposed which threaten the security of the message [6]. There are many flaws in the PKMv1 protocol, such as man-in-the-middle attack [7]. The PKMv2 protocol also has many defects in authentication and information management. With the attacks upgrading, it is prone

to suffer with Denial-of-Service (DOS) attacks [8] due to its weak encryption mechanism. Therefore, IEEE 802.16m released a new version, named PKMv3 protocol, which could theoretically solve the problems of PKMv1 and PKMv2 protocols in the message management and strengthen the safety. Communication networks not only require the large bandwidth and fast transmission speed, but also need to maintain a very high transmission accuracy and security. Therefore, it is necessary to model and analyze the existing transmission protocols, and to study whether there are vulnerabilities that can pose a threat to the communication or can cause the communication failure.

The newly proposed next generation (5G) wireless networks lie in providing a huge number of connections, 10G bit/s througput per connection, a wider variety of wireless services, more flexibility and exponentially growing cellular data. It will be used not only for the communication between people, but also for the communication between people and things, things and things, to achieve real "Internet of everything". The International Standards Organization of 3GPP [9] group standardize the security mechanisms to protect the security of the mobile network. One key concern of the wireless networks is the security guarantees among the commnication entities, such as the users and carriers. In 5G networks, a trust model consists of 3 elements, the users, services and the network. The authentication between services and users are put forward compared with the 4G network model, to carry out a more secure and efficient management mode.

The 3GPP specified and standarized the 3G, 4G and 5G technologies. Hewly released Technical Specification (TS) version v15.1.0 describes the 5G security architecture and mechanism [10], where 5G Authentication and Key Agreement (AKA) protocol is introduced. It is widely applied for the mutually authentication of the mobile phone with its USIM card with networks to improve the security guarantees. 5G AKA is extended from the 4G AKA [11], defineing new authentication-related services. Four entities are in the 5G AKA protocol, the User Equipment (UE), the Serving Network (SN), the Home Network (HN) and Authentication Credential Repository. The main task is to mutually authenticate the UE and HN and to obain a secret session key, used to protect the messages in the following commnunications. EAP-AKA' [12] authentication is another authentication method of the 5G networks. The choice of using which protocol is determined by the Home Network.

1.1 Related work

The confidentiality and security of the protocol play an important role in the communications. Some efforts and achievements have been made on the analysis of each version of the PKM protocol. Taha et al. [13] presented how the PKMv1 protocol was formalized with backward symbolic state search technique and analyzed security protocols by the model checker Scyther. They also put forward some modifications to improve the protocol. Xu et al. [14] abstracted PKMv1 and PKMv2 protocol as input of CasperFDR and found some attacks like the interleaving attacks, which is solved by attaching signatures of the commnicating entities. They focused more on the flaws than the model and analysis of the protocol itself. Nearly all the formal models of the secure protocols are communication entities with keys and nonces. Attributes omitted make the protocol far from the specification such as [15, 16]. Yang et al. [17] compared Traffic Encryption Key (TEK) exchange period of the PKMv1 protocol with that of the PKMv2 protocol and drew the conclusion that PKMv2 protocol was much more secure than the first version.

Compared with the first two versions of PKM protocol, PKMv3 protocol is more reliable and secure. But it is still necessary to be verified and analyzed whether it is safe enough. Raju et al. have modeled PKMv3 protocol and analyzed it with CSP [18]. They found two possible attacks on authentication specificatoins. In that work, the solutions are left to future work. In [19, 20], the authors performed a formal analysis of the PKMv3 protocol and some properties are specified using DT-Spin. Similarly, some flaws are testified and analysis of the possibility of attacks are discussed, except for the excutable methods. Jiang et al. [21] formalized the protocol as well as proposed improved two approaches to the potential attacks. However, message replay mechanism was not taken into consideration. My previous work [22] on the PKMv3 protocol was implemented in PAT. We extracted some important properties from which vulnerabilities were found. However, seeking solutions was also the future work. The current relevant works maily focused on the subscriber station and the base station, but there is also another important entity called server, playing a necessary role in the communication to distribute the certificate to the stations. Some attacks may also occur during their session and will influence the formalization and verification of the whole comminication mechanism.

Several analysis and improvement of AKA protocols have been done. Borgaonkar et al. [23] analyzed and revealed the new privacy threat on 3G, 4G and 5G networks. And they proposed to use low cost and widely availabel setups to demonstrate the practical feasibility of the attack. And Aiash et al. [24] explored the behaviors of the AKA in the heterogeneous environment and put forward a new authentication and key agreement protocol implemented by the Casper/FDR. In [11], the authors firstly displayed the vulnerabilities the 3GPP AKA protocol is exposed to, and then enhanced the performance by eliminating the need of the synchronization between the MS and HN. Lee et al. [25] put forward a modified signcryption scheme which provides signer anonymity, applying to the AKA protocols. Basin et al. [26] made the formal analysis of the 5G authenticationusing Tamarin tool and explicit recommendations weremade with provably secure fixes to the vulnerabilities they discovered. However, the analysis of the credential repository is lacking which will make the work incomprehensive. The 5G-AKA protocol was studied and modified to prevent the attacks and formally proved the protocol σ-unlikable [27]. The modeling of the protocol was just between the UE and HN, not the four entities according to the protocol's specification.

1.2 Contributions

The main work and contributions of this thesis are listed as follows:

1. We formalize the PKMv3 protocol using the CSP method. We give description and definition of communication entities, such as the channels, base stations, mobile stations, the server as well as the intruders, with CSP syntax. We also introduce the server model in order to give the more specific analysis of the commnication mechanism.

2. We simulate and implement the protocol in PAT. Six properties are extracted and verified, which are the important basis of analyzing the possible attacks. We put forward some approaches according to the verification results to improve the performance of the PKMv3 protocol.

3. Moreover, we describe and formalize the 5G AKA protocol with all parties using CSP and is simulated in PAT. Vulnerabilities are found through the verification

results of the properties. We propose some methods to improve the security of the mutually authentication.

1.3 Thesis Overview

The rest of the thesis is organized in the following. In Chapter 2, we give an overview of the PKMv3 protocol, including the kind of keys and the commnication mechanism. Also, we give the overview architecture and the normal description of the 5G AKA protocol. The CSP method and the tool PAT are introduced in detail. The formal description of the protocol in CSP is shown in Chapter 3. In the next chapter, we simulate and implement the model in PAT as well as verifying six properties extracted from the protocol. For analyzing the results of the verification, we focus on discussing whether some attacks may occur in the protocol and provide some possible traces of the intruders. Moreover, improved approaches are proposed in Chapter 5. The formalization of the 5G AKA protocol is implemented in Chapter 6. In the next chapter, we verify some properties to check whether our mode conforms to the protocol's specification. The verification and analysis make the protocol more secure and efficient. Finally, the conclusion and future expectations are discussed in Chapter 8.

Chapter 2

Background

The PKMv3 protocol is based on PKMv1 and PKMv2 protocols. The protocol includes five steps to accomplish the security key distribution scheme: certificates requirement and reply between servers and base stations(or mobile stations); initialization; AK authorization; transmission of the security association and exchange of TEK.

5G AKA is improved from the 4G AKA. It has four entities, the user equipment (UE), the serving network (SN), the home network (HN) an the credential store used by HN. As for the transition procedure, it contains the initialization, the challenge request and response, the auth info and authentication confirmation.

2.1 Introduction to Keys of the PKMv3 Protocol

In the PKMv3 protocol, many keys are used to encrypt and protect information in the process of communication, such as the Pair-wise Master key (PMK), which is used to generate the Authorization Key (AK). And AK is the shared key used by the base station to authenticate and authorize the mobile station and generate the Ciper-based Message Authentication Code (CMAC). Traffic Encryption Key (TEK), also comes from AK, is applied to ensure the security of information transmission after authorization. CMAC keys are divided into the upstream CMAC_KEY_U key and the downstream CMAC_KEY_D key, in which the upstream CMAC_KEY_U key refers to the CMAC key used when transmitting the message from the mobile station to the base station, while the downstream CMAC_KEY_D key is the opposite.

The base station will use HAMC_KEY_U extracted from an active AK to calculate the CMAC digest in the key request message received from the terminal. The base station can decide which CAMC_KEY_U to use to authenticate the message according to the AK key sequence number attached to each key request message. Then the next information to be sent to the mobile station is encrypted with the CMAC_KEY_D key to generate a new CMAC value attached to it and sent to the mobile station. Similarly, the mobile station uses the received AK to extract the CMAC_KEY_D key, calculating the CMAC digest for the whole message, and then compares it with the CMAC digest calculated by the base station in the message. If the values are the same, it will indicate that the message is correct and has not been tampered with or forged by the intruder. Both entities can continue the transmission process, otherwise, the session is terminated.

2.2 Transmission mechanism of PKMv3 Protocol

The whole transmisson process of the protocol is shown in Fig. 2.1.

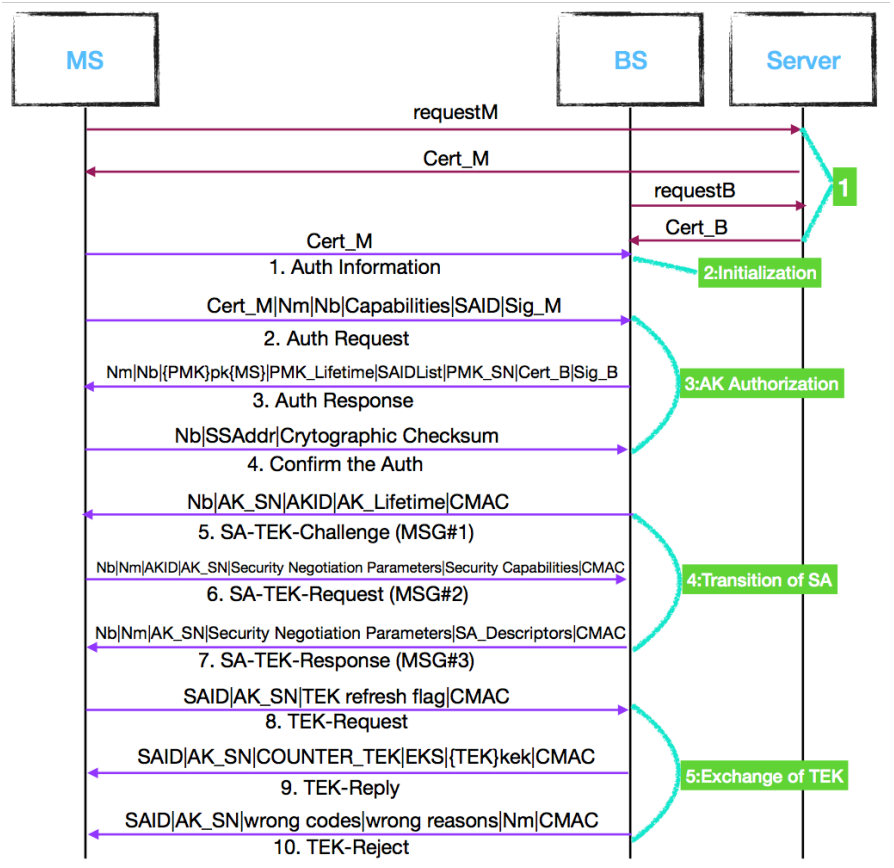

FIGURE 2.1: Communication Mechanism

2.2.1 Certificate reply and response

The MS wants to establish contact with the BS for communication, so there must be a corresponding security mechanism, by using the common transition key to encrypt the message. Firstly, MS need to apply to the Sserver for its own manufacturer certificate (X.509 certificate [28]) to inform the base station of its identity, public key and MAC address. As for the BS, it also need to apply for its own certificate and then pass it to the MS to prove its identity (Part 1 in Fig. 2.1).

The processes are shown as follows:

1. MS asks for certificate: $MS \rightarrow Server :< requestM >$
2. Server gives the certificate to MS: $Server \rightarrow MS :< \{Cert\_M\}sk\{Server\} >$
3. BS asks for certificate: $BS \rightarrow Server :< requestB >$
4. Server gives the certificate to BS: $Server \rightarrow BS :< \{Cert\_B\}sk\{Server\} >$

2.2.2 Initialization

After receiving the certificate itself, MS needs to send an authentication information message to BS which represents the start of the conversation. The message includes the certification of MS (Part 2 in Fig. 2.1):

$Auth\ Information : MS \rightarrow BS :< Cert\_M >$.

2.2.3 AK authorization

AK Authorization is to establish the connections between MS and BS and prevent the theft behavior by using of cloned terminals. The process contains three steps: the authorization request and reply messages between MS and BS, as well as the confirmation of the authorization. (Part 3 in Fig. 2.1)

Immediately after the initialization information is sent by the mobile station, an authorization request message (Auth Request) is sent to BS, which is to request AK from BS and the identification of all static security associations in which the mobile station can be authorized. Messages contain many elements, such as mobile station's certificate $Cert\_M$, including RSA public key, MAC address, manufacturer ID, etc., random

number Nm generated by MS, and encryption function $Capabilities$ to determine the encryption algorithm and protocol of MS and BS, 16-bit initial security association identifier $SAID$, MS digital signature Sig_M.

After receiving the authorization request message from the mobile station, the base station first verifies the identity of the mobile station, that is, whether it is the same certificate. Then it assures the key and encryption algorithms that both sides support through the $Capabilities$ in the authorization request message. BS generates a PMK for the MS, encrypting it with the public key of the mobile station. The authorization reply message (Auth Response) contains the random number Nm of MS; the random number Nb of BS; PMK encrypted with MS public key $\{PMK\}pk\{MS\}$; $PMK\_Lifetime$ of PMK's life cycle; $SAIDList$ of security association identifiers; PMK_SN of PMK's serial number; $Cert\_B$ of BS; and the digital signature Sig_B of BS.

The last message is the authorization confirmation (Confirm the Auth). After receiving the Auth Reponse), MS examines and verifies the identity of BS to see whecher it is legal. The message contains the random number Nb of BS, the MAC address $MSAddr$ of MS, and the password check and $Crytographic\ Checksum$ of the message.

The processes are shown clearly as follows:

- $Auth\ Request : MS \rightarrow BS :< Cert\_M, Ns, Capabilities, SAID, Sig\_M >$;
- $Auth\ Response : BS \rightarrow MS :< Nb, Nm, \{PMK\}pk\{MS\}, PMK\_Lifetime, SAIDList, Sig\_B, PMK\_SN, Cert\_B >$;
- $Confirm\ the\ Auth : MS \rightarrow BS < Nb, MSAddr, Crytographic\ Checksum >$.

2.2.4 Transmission of security association

Security association (SA) is the shared safety information encrypted in the 802.16 network between BS and one or more terminals. The three-way handshake procedure is used to create the dynamic security association (Part 4 in Fig. 2.1).

The first handshake from BS to MS represents the PMK and AK are created and then will be used by the SA (SA-TEK-Challenge, MSG#1). The content of the message includes: the nonce of BS, the sequence number of AK AK_SN, the ID of AK $AKID$, the AK's lifetime $AK\_Lifetime$ as well as the message authentication code $CMAC$.

The second handshake from MS to BS called SA-TEK-Request (MSG#2) is that MS becomes authorized and applies for the SA and its characteristics. It contains a new parameter called $SecurityNegotiationParameters$, representing the protective parameters to confirm the used authentication and information integrity. The third handshake from BS to MS (SA-TEK-Response, MSG#3) describes that BS accepts the request of prime SA and static SA and responds to MS.

The tramsmission processes are:

- $MSG\#1 : BS \rightarrow MS :< Nb, AK\_Lifetime, AKID, AK\_SN, CMAC >$;
- $MSG\#2 : MS \rightarrow BS :< Nb, Nm, AKID, AK\_SN, Security\ Negotiation\ Parameters, Security\ Capabilities, CMAC >$;
- $MSG\#3 : BS \rightarrow MS :< Nb, Nm, AK\_SN, Security\ Negotiation\ Parameters, Security\ Capabilities, SA\_Descriptors, CMAC >$.

2.2.5 Exchange of TEK

After the transmission of SA, before the mobile station sends the real message to the base station, it also needs the key to encrypt the message in the transmission process, that is, the TEK. of the message. TEK is used to encrypt transmitted messages and ensure the consistency and security during communications. The exchange of TEK composes two steps. One is the TEK request (TEK-Request), showing that MS requests for TEK from BS. The other one is the TEK reply (TEK-Reply), which is accomplished by transmitting a message from BS to MS. If MS's CMAC is legal, BS will send the needed TEK to MS. Otherwise, BS will send a reject message(TEK_Reject), which includes the wrong codes and reasons to MS (Part 5 in Figure 1).

Detailed transmission are as follows:

- $TEK\_Request : MS \rightarrow BS :< SAID, AK\_SN, TEK\ refresh\ flag, CMAC >$;
- $TEK\_Reply : BS \rightarrow MS :< SAID, PMK\_SN, COUNTER\_TEK, EKS, \{TEK\}kek, CMAC >$;
- $TEK\_Reject : MS \rightarrow BS :< SAID, AK\_SN, wrong\ codes, wrong\ reasons, CMAC >$.

After finishing all the steps above, BS and MS are connected successfully and can transmit messages to each other.

2.3 Authentication mechanism of 5G AKA

In this section, we will explain the main architecture and how the authentication and key agreement work of the 5G system, which is in accordance with the newly published specification 3GPP TS 33.501 [10].

2.3.1 Overall Architecture

There are four main entities, including the user equipment (UE), the serving network (SN), the home network (HN) as well as the credential store (ARPF, which is resides in the home network), in the 5G cellular network. The vivid architecture is shown in Fig. 2.2.

- **UE**: It is the user equipment, such as the mobile phone, which is identified by the subscription permanent identifier (SUPI).
- **SEAF**: It is the security anchor function, which resides in the serving network. The SN is the network that devices connect to, such as the antenna. Once the UE and SN authenticate with each other, SN will provide services to the UE.
- **AUSF**: It is the authentication server function, which is used in the home network. It is a kind of EAP server, showing which network the phone is signed with, such as the carrier of the mobile phones.
- **ARPF**: It is the authentication credential repository and processing function.

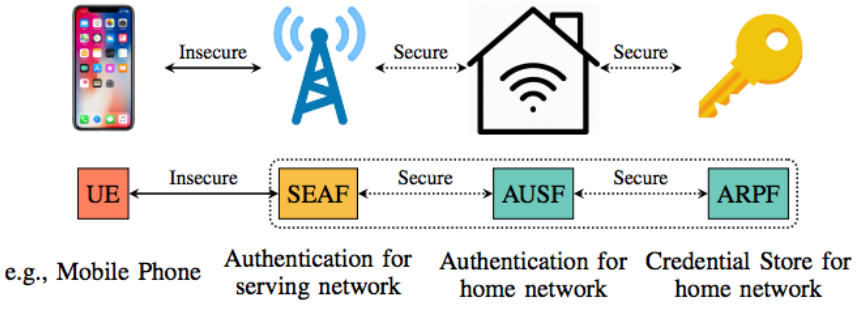

FIGURE 2.2: Overall architecture of 5G AKA obtained from [1]

2.3.2 Normal Description of the 5G-AKA protocol

The HN can choose which authentication method to establish secure channels between the SN and the UE. The detailed authenticaton mechanism is shown in Fig. 2.3.

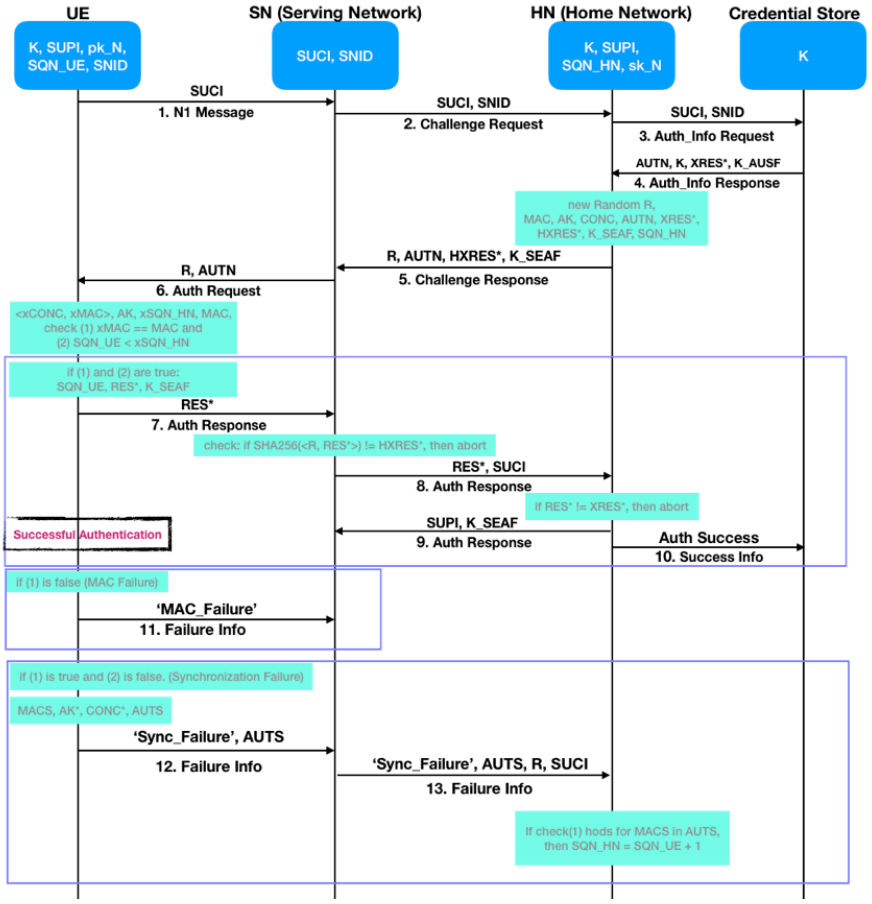

FIGURE 2.3: Authentication mechanism of 5G AKA protocol

The first step is the initialization message (N1 Message), containing the subscription concealed identifiersent (SUCI) from the UE to SN. Here, we use SUCI, the asymmetric encryption of the SUPI, the random nonceand the identifier of HN, to replace SUPI because SUPI must be keep secret. After receives the initialization message, the SN will send the challenge message (Challenge Request), containing the SUCI and the name of SN ($SNID$) to the HN to ask for the authentication materials. Next comes to the Auth_Info Request, delivering from the HN to ARPF in order to obtain the useful parameters. Once obtains the Auth_Info Response message from the ARPF, HN will generate relevant materials and then send the random nonce R, AUTN (to prove the freshness of the message), $HXRES*$ (the SN's 'expected response' value) and K_{SEAF} (the key seed for the secure channel used by UE and SN) to SN.

After that, the SN send an Auth Request messge including R and AUTN to UE. After decodes and calculates the message, UE will compare two parameters. If $xMAC$ equals to MAC and SQN_{UE} equals to $xSQN_{HN}$, UE will send the authentication response message ($RES*$ token) back to SN. Once the hash of received $RES*$ maches $HXRES*$, SN will send Auth Response messge, containing $RES*$ and SUCI to HN, otherwise, abort the authentication. Similarly, HN is going to verify the received $RES*$ with $XRES*$, if they are the same, HN will send SUPI and K_{SEAF} back to SN, and send Success Info to the ARPF to show the success of the authentication. If $RES*$ is different with $XRES*$, the session will be aborted. The successful of the authentication means after that, all entities of the protocol can derive session keys from K_SEAF, by using which UE can call or send messages safely.

If $xMAC$ is different from MAC, UE will send the 'MAC_Failure' information to the SN and terminate the authentication. If $xMAC$ equals to MAC but SQN_{UE} is not the same as $xSQN_{HN}$, there will be the synchronization failure.

The relevant notations' abbreviations and definations are shown in Table 2.1 and the values of used messages are defined in Table 2.2.

TABLE 2.1: Definations of Notations of the 5G AKA protocol

Element	**Meaning**
AKA	Authentication and Key Agreement
AK	Anonymity Key
ARPF	Authentication credential Repository and Processing Function
AUSF	Authentication Server Function
HN	Home Network
SN	Serving Network
UE	User Equipment
K	Long-term secret master key shared between UE and corresponding HNs
K_{AUSF}	Andhor key of AUSF
K_{SEAF}	Anchor key of SEAF
MAC	Message Authentication Code
SEAF	Security Anchor Function
SIDF	Subscriber Identity De-concealing Function
SNID	Identifier of the serving network
SQN	Sequence Number
SUCI	Subscription Concealed Identifier
SUPI	Subscription Permanent Identifier
R	Random number
RES	Response message
XRES	Expected response value

TABLE 2.2: Values of Messages in the 5G AKA protocol

Message Name	Value
CONC	$SQN_{HN} \oplus AK$
$AUTN$	$< CONC, MAC >$
AK	$f5(K, R)$
$XRES*$	$Challenge(K, R, SNID)$
$HXRES*$	$SHA256(< R, XRES* >)$
K_{SEAF}	$KeySeed(K, R, SQN_{HN}, SNID)$
SQN_{HN}	$SQN_{HN} + 1$
$xSQN_{HN}$	$AK \oplus xCONC$
SQN_{UE}	$xSQN_{HN}$
MACS	$f1 * (K, < SQN_{UE}, R >)$
$AK*$	$f5 * (K, R)$
$CONC*$	$SQN_{UE} \oplus AK*$
AUTS	$< CONC*, MAC* >$

There are also some functions used in the protocol. $f1$, $f5$, $f5*$ are different keyed cryptographic one-way functions, which are used for the integrity and confidentiality of the protocol. $SHA256()$ is the hash function. $\oplus$ represents Exclusive-OR. And functions Challenge() and KeySeed() are Key Derivation Functions (KDFs) [26].

2.4 Introduction of CSP

CSP was firstly proposed by Hoare [29] and now it has become a well-rounded formal process algebra. It specialises in describing concurrent and distributed systems. In CSP, a process is abstracted as a mathematical expression, indicating interactions between the environment and the system or between the systems. Every event triggers associated processes to do a series actions (or one action). Below shows the syntax of CSP and more details about CSP syntax and examples can be found in [30].

$$P, Q ::= STOP \mid SKIP \mid a \rightarrow P \mid c!v \rightarrow P(x) \mid c?x \rightarrow P(x) \mid P \Box Q \mid P; Q \mid P||Q \mid P \lhd b \rhd Q \mid P[\|X\|]Q \mid P[[a \leftarrow b]] \mid trace(P)$$

- Here P and Q are processes which have alphabets $\alpha(P)$ and $\alpha(Q)$ to show the sets of actions they may perform respectively.
- $STOP$ means the process is deadlock. $SKIP$ represents that a process does nothing and succeeds in terminating.
- $a \rightarrow P$ shows that a process first acts as an atomic action a and then the subsequent behaviors are like P.

- $c!v \rightarrow P$ means a process uses the channel c to send a message v and then behaves like P.
- $c?x \rightarrow P(x)$ indicates after using the channel c to receive a message and storing it in the variable x, the process then behaves like $P(x)$.
- $P \Box Q$ is a general choice, showing the behavior that is either like P or Q. This choice is determined by the environment.
- Processes $P; Q$ perform sequentially, only if P succeeds in performing will Q go on.
- $P||Q$ describes P and Q execute concurrently.
- $P \lhd b \rhd Q$ is a conditional choice. If the boolean expression b is true, the process acts as P, otherwise Q.
- $P[|X|]Q$ represents that two processes, P and Q, performing concurrent operations on the set of X channels.
- $P[[a \leftarrow b]]$ is a rename operation, in which action a in process P is replaced by b.
- $trace(P)$ is a sequence of actions performed by the process P.

2.5 Introduction of PAT

PAT [31] is an independent framework to simulate and modularise systems such as concurrent systems and real-time systems. Based on the CSP, PAT is widely used in many formalized models and can be applied to verify properties like the reachability, deadlock-freeness and divergence-freeness to cater for different requirements. LTL formulae and assertions can also be devoted to check self-defined properties such as the timeout freedom. The following shows some notations introduced in PAT (more details can be found in [32]).

- *channel* c 0 indicates a channel named c and the capacity is 0;
- $\#define\ N$ 0 defines a global constant N and the initial value is 0;
- $var\ MSG[N]$ means an array variable named MSG whose size is N;

- $c!msg \rightarrow P$ and $c?m \rightarrow P$ are two processes and behaviours are similarly as defined in CSP;
- $if(a)\{A\}\ else\ \{B\}$ refers to the judgement and choice. If a is true, the process will act as A, otherwise B.

Chapter 3

Modeling the PKMv3 Protocol Using CSP

Before modeling the protocol, we briefly introduces the intruders types of the security protocol, which are mainly divided into external intruders and internal intruders. Both of them have similar possible intrusions, such as intercepting, eavesdropping, tampering or reasoning messages. But the difference is that the external intruder knows much less about the message itself than the internal intruder, that is, the internal intruder knows how to calculate the CMAC value and so on, but the external intruder does not understand the information. Therefore, as far as the threat to security protocols is concerned, the harm caused by internal intruders is far greater than that caused by external intruders. In the thesis, we will neglect the modeling of external intruder behavior, but focus on the harm of internal intruder's behaviors. In the following, the pre-defined symbols and protocol hypothesis are introduced first, and then it comes to some additional sets and channels defination as well as the use of CSP formal modeling.

3.1 Preliminaries

3.1.1 Notations

There are some important notations which apply to the modeling of the protocol.

- **M**, **B** and **S**: are the mobile station, base station and the sever respectively;

- $\boldsymbol{M_{M_i} = CMAC(K_{M_i}, M)}$: when the node M sends or receives the $i-th$ message, it will use the key K_{M_i} to compute M's CMAC value, i.e., M_{M_i};
- $\boldsymbol{M_{B_i} = CMAC(K_{B_i}, B)}$: when the node B sends or receives the $i-th$ message, it will use the key K_{B_i} to compute B's CMAC value, i.e., M_{B_i};
- $\boldsymbol{N_b}$, $\boldsymbol{N_m}$: the nonce of B and M;
- **MaxResends**: Maximum time of messages sent by SA-TEK-Request (is the system parameter);
- **MAX**: Maximum time of messages sent by SA-TEK-Challenge (is the system parameter);
- **SAChallengeTime** : The maximum waiting time between the base station sending MSG#1 and receiving MSG#2 from the mobile station;
- **SATEK-Timer**: The maximum waiting time between the mobile station sending MSG#2 and receiving MSG#3 from the base station.

3.1.2 Assumptions

There are two assumptions listed before create the model.

1. The network is bidirectional, that is, if A node can receive information from B node, it can also receive messages sent from A.
2. C node can eavesdrop, tamper and forge messages transmitted between nodes in the whole certificate request, authorized access, SA security association, TEK request processes, and can also communicate with the base station and mobile station as a normal mobile station node.

3.1.3 Sets

Because there are stations, keys and intruders in reality, we abstract them by using some notations, which is easy to represent.

- **Set of MS**: $MobileStation = \{MS\}$;

- **Set of BS**: $BaseStation = \{BS\}$;
- **Set of Server**: $Server = \{S\}$;
- **Set of intruders**: $Intruder = \{Intruder\}$;
- **Set of keys**: Key represents the set of CMAC keys when MS and BS compute the CMAC values. The corresponding CMAC value is calculated by using the CMAC key extracted from AK at a certain time, which is attached to the message. The other party can also use the CMAC key generated from AK to calculate its CMAC compared with the CMAC digest attached to the message. The key is mainly used to authenticate the identity of the node and the integrity of the message. Key is defined as: $Key = \{K_{M_i}, K_{B_i} | i = 5, 6, 7, 8, 9, 10\}$. For example, K_{B_5} is the key extrated from AK which is attached in MSG#1, sent from BS to MS. K_{M_5} is the key generated from MSG#1's AK, to verify whecher M_{B_5} is valid;
- **Set of messages**: MSG is the set of all messages transmitted through the whole communication. There are three types: certificate request (MSG_{req}), certificate reply(MSG_{rep}) and messages between MS and BS (MSG_{MB}). Therefore, the set can be expressed as: $MSG = MSG_{MB} \cup MSG_{rep} \cup MSG_{MB}$. To make it easier, we simplify the communication procedure and indicate the messages between MS (or BS) and the Sever as $M_i, i = 1, 2, 3, 4$. Similarly, the messages between MS and BS are represented as $MSGi, i = 1, ..., 10$.

3.2 Channels

Channes are the places where messages are transmitted. In our work, we define three kinds of channes to model the communication between nodes in the authorization request key process of the PKMv3 protocol.

1. Unicast communication channels $\boldsymbol{ComMS}$, $\boldsymbol{ComMB}$, $\boldsymbol{ComBS}$: These three channels represent the channels used for point-to-point communication between the mobile station and the server, between MS and BS, as well as BS and the server respectively.
2. Intercept channels $\boldsymbol{InterceptM\_ComMS}$, $\boldsymbol{InterceptM\_ComMB}$, $\boldsymbol{InterceptB\_ComMB}$, $\boldsymbol{InterceptB\_ComBS}$, $\boldsymbol{InterceptS\_ComMS}$ and

InterceptS_ComBS: These six channels indicate that intruder C eavesdrops and intercepts messages from the mobile station M, base station B and the server S.

3. Fake channes ***FakeS_ComMS***, ***FakeS_ComBS***, ***FakeB_ComBS***, ***FakeB_ComMB***, ***FakeM_ComMB*** and ***FakeM_ComMS***: They indicate that invading node C sends certificate reply messages as initial node S, or publishes tampered messages in the name of B node or M node.

4. ***Session*** and ***Fake_Session***: ***Session*** indicates that two nodes have successfully completed key requests and other processes. ***Fake_Session*** indicates that the intruder has successfully forged or tampered with the channel used by the transmitted message.

Various channels' connections and nodes' relations are shown in Fig. 3.1.

3.3 Model of the PKMv3 Protocol

In this part, we will use CSP to describe the PKMv3 protocol, which contains the model of four entities: the timer, mobile station, base station and the intruder.

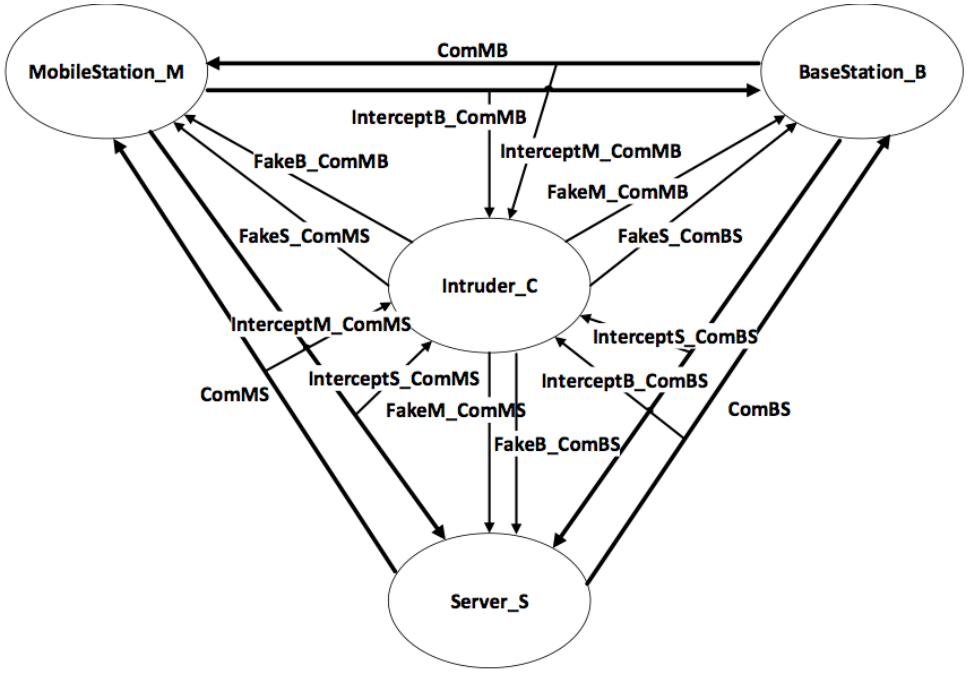

FIGURE 3.1: Communications using channels among parties

3.3.1 Timer

In the PKMv3 protocol, there is a delay in message transmission. If there is not a reply message received within the specified time, the sender will choose to resend the message again. If this situation occurs many times, the communication with be terminated. The timer is a communication entity shared between the base station, server and the mobile station nodes. Therefore, before we model the protocol, we need to model the timer used in the protocol process. The timer is defined as follows:

$$Timer_m =_{def} (tick \rightarrow Timer_{m+1}) \\ \square\ (time?request \rightarrow time!m \rightarrow Timer_m), m \geq 0 \wedge m \in N.$$

Here, *time* represents the communication channel between the timer and each node.

3.3.2 Server

Before the mobile station is ready to apply for the authorization request and related key from the base station, it will communicate with the server to request the certificate of the mobile station. The server will perform the actions of receiving the certificate request message and encrypting the certificate with the public key of the requester and reply.

First, we assume that in an ideal environment without intrusive nodes, the server modeling process is as follows:

$$ServerS =_{def} ComMS?M_1.M.S \rightarrow ComMS!M_2.S.M \rightarrow Server(S) \\ \square\ \ ComBS?M_3.B.S \rightarrow ComBS!M_4.S.B \rightarrow Server(S)$$

After S receives M's certificate grant request, S will encrypt the certificate with the public key of the sender and pass back to M, returning to its original state. Or S receives B's certificate grant request, transmitting the message to B, and then returns to its original state.

However, in practice, this ideal environment without intruders often does not exist. In the process of certificate application, there may be intruders, who will tamper with or forge the original behavior of nodes. Next, we will model the behavior of the initial routing node in the presence of intruders.

$$
\begin{aligned}
ServerS =_{def}\ & Server_1(S) \\
& [[\ ComMS?M_1.M.S \leftarrow ComMS?M_1.M.S \\
& \ \ ComMS?M_1.M.S \leftarrow FakeM\_ComMS?M_1.M.S \\
& \ \ ComMS!M_2.S.M \leftarrow ComMS!M_2.S.M \\
& \ \ ComMS!M_2.S.M \leftarrow InterceptM\_ComMS!M_2.S.M \\
& \ \ ComBS?M_3.B.S \leftarrow ComBS?M_3.B.S \\
& \ \ ComBS?M_3.B.S \leftarrow FakeB\_ComBS?M_3.B.S \\
& \ \ ComBS!M_4.S.B \leftarrow ComBS!M_4.S.B \\
& \ \ ComBS!M_4.S.B \leftarrow InterceptB\_ComBS!M_4.S.B \\
& \ \ Session \leftarrow Session \\
& \ \ Session \leftarrow Fake\_Session\ ]]
\end{aligned}
$$

3.3.3 Mobile Station

The mobile station firstly requests the certificate from the server, then decrypts the encrypted certificate with its private key between S. Later M transmits the certificate to the base station B as the initialization step. Then the authorization request is made, $MSG2$ is sent to B, and the random number generated by itself is attached to the message to ensure the freshness of the message. At the same time, the signature is also sent to B, aiming to tell B that it is the message sent by itself. Then M waits for the authorized reply message $MSG3$ to verify whether the random number sent by the message itself is the same as before. If it is the same, it means that the message has not been changed and the message is complete, otherwise the communication will be terminated. Next step is to reply a confirmation message to B, informing B that it has received the authorized reply. The next three handshake protocol guarantees that B and M can turn on the TEK state machine in order to transfer applications for encryption keys, etc. In each step of the three handshakes and TEK requests and replies, it is useful to extract the AK from the received message and generate the corresponding key to calculate the message authentication code, and to compare the authentication code in the message package. If they are the same, the message will not be tampered with or forged, otherwise, the message will be considered to have been tampered with so as to terminate the communication.

In the situation that there is no intruder node, the mobile station node is modeled using CSP as follows:

$$
\begin{aligned}
&MobileStation(M) =_{def} ComMS!M_1.M.S \\
&\quad \rightarrow ComMS?M_2.S.M \rightarrow ComMB!MSG1.M.B \\
&\quad \rightarrow ComMB!MSG2.M.B \rightarrow ComMB?MSG3.B.M \\
&\quad \rightarrow STOP \triangleleft (M.N_m! = N_m) \triangleright ComMB!MSG4.M.B \\
&\quad \rightarrow ComMB?MSG5.K_{B_5}.B.M \rightarrow STOP \triangleleft (M_{M_5}! = M_{B_5}) \triangleright P \\
&\quad \rightarrow \mu X.((T_2 - T_1 < 0 | T_2 - T_1 > SATEK - Timer) \\
&\quad \rightarrow (COUNT_1 = COUNT_1 + 1; STOP \triangleleft COUNT_1 > MaxResends \triangleright P; X) \\
&\quad \mid ((T_2 - T_1 > 0 | T_2 - T_1 \leq SATEK - Timer) \\
&\quad \rightarrow STOP \triangleleft (M_{M_7}! = M_{B_7}) \triangleright ComMB!MSG8.K_{M_8}.M.B)) \\
&\quad \rightarrow (ComMB?MSG9.K_{B_9}.B.M \\
&\quad \rightarrow STOP \triangleleft (M_{M_9}! = M_{B_9}) \triangleright Session.M.B \rightarrow MobileStation(M) \\
&\quad \Box\ ComMB?MSG10.K_{B_{10}}.B.M \rightarrow STOP) \\
&\text{Here, } P = (ComMB!MSG6.K_{M_6}.M.B \rightarrow time!request \rightarrow time?T_1 \\
&\qquad \rightarrow ComMB?MSG7.K_{B_7}.B.M \rightarrow time!request \rightarrow time?T_2)
\end{aligned}
$$

Similarly, in the presence of intruders, we represent mobile station nodes as $MobileStation_1(M)$ and is shown the behaviors as follows.

$$
\begin{aligned}
&MobileStationM =_{def} MobileStation_1(M) \\
&\qquad [[\ ComMS!M_1.M.S \leftarrow ComMS!M_1.M.S \\
&\qquad\ \ ComMS!M_1.M.S \leftarrow InterceptS\_ComMS!M_1.M.S \\
&\qquad\ \ ComMS?M_2.S.M \leftarrow ComMS?M_2.S.M \\
&\qquad\ \ ComMS?M_2.S.M \leftarrow FakeS\_ComMS?M_2.S.M \\
&\qquad\ \ ComMB!msg.M.B \leftarrow ComMB!msg.M.B \\
&\qquad\ \ ComMB!msg.M.B \leftarrow InterceptB\_ComMB!msg.M.B \\
&\qquad\ \ ComMB?msg.B.M \leftarrow ComMB?msg.B.M \\
&\qquad\ \ ComMB?msg.B.M \leftarrow FakeB\_ComMB?msg.B.M \\
&\qquad\ \ Session \leftarrow Session \\
&\qquad\ \ Session \leftarrow Fake\_Session\ ]] \\
&\text{Here, } msg \in MSG_{MB}.
\end{aligned}
$$

3.3.4 Base Station

In this section, we will use CSP to model the behavior of base station nodes. In PKMv3 security protocol, the base station firstly requests its own certificate from the server. After receiving the initial information of M, BS then sends its own certificate and security information set which can be applied to M's authorization access request to M, indicating the authorization access request of M. Next, the base station will do the 3-shake-hand with the mobile station and the exchange of TEK processes. The specific model of BS is as follows:

$$
\begin{aligned}
&BaseStation(B) =_{def} ComBS!M_3.B.S \\
&\quad \rightarrow ComBS?M_4.S.B \rightarrow ComMB?MSG1.M.B \\
&\quad \rightarrow ComMB?MSG2.M.B \rightarrow ComMB!MSG3.B.M \\
&\quad \rightarrow ComMB?MSG4.M.B \\
&\quad \rightarrow STOP \triangleleft (M.N_b! = N_b) \triangleright Q \\
&\quad \rightarrow \mu X.((T_4 - T_3 < 0 | T_4 - T_3 > SAChallengeTimer) \\
&\quad \rightarrow (COUNT_2 = COUNT_2 + 1; STOP \triangleleft (COUNT_2 > Max) \triangleright Q; X) \\
&\quad \mid ((T_4 - T_3 > 0 | T_4 - T_3 \leq SAChallengeTimer) \\
&\quad \rightarrow STOP \triangleleft (M_{M_6}! = M_{B_6}) \triangleright ComMB!MSG7.K_{B_7}.B.M)) \\
&\quad \rightarrow (ComMB?MSG8.K_{M_8}.M.B \\
&\quad (ComMB!MSG10.K_{B_{10}}.B.M \rightarrow Fake\_Session.B.M \\
&\quad \triangleleft BaseStation(B) \rightarrow (M_{M_8}! = M_{B_8}) \triangleright \\
&\quad (ComMB!MSG9.K_{B_9}.B.M \rightarrow Session.B.M \rightarrow BaseStation(B)) \\
&\text{Here, } Q = (ComMB!MSG5.K_{B_5}.B.M \rightarrow time!request \rightarrow time?T_3 \\
&\qquad \rightarrow ComMB?MSG6.K_{M_6}.M.B \rightarrow time!request \rightarrow time?T_4)
\end{aligned}
$$

When there are intruders in the environment, the behaviors of base station are modeled in the following:

$BaseStationB =_{def} BaseStation_1(B)$

$[[\ ComBS!M_3.B.S \leftarrow ComBS!M_3.B.S$

$ComBS!M_3.B.S \leftarrow InterceptS\_ComBS!M_3.B.S$

$ComMS?M_4.S.B \leftarrow ComMS?M_4.S.B$

$ComMS?M_4.S.B \leftarrow FakeS\_ComBS?M_4.S.B$

$ComMB?msg.M.B \leftarrow ComMB?msg.M.B$

$ComMB?msg.M.B \leftarrow FakeM\_ComMB!msg.M.B$

$ComMB!msg.B.M \leftarrow ComMB!msg.B.M$

$ComMB!msg.B.M \leftarrow InterceptM\_ComMB!msg.B.M$

$Session \leftarrow Session$

$Session \leftarrow Fake\_Session\ ]]$

Here, $msg \in MSG_{MB}$.

3.3.5 Intruder

Throughout the process of modeling, we regard intruders as a process, which can intercept, modify and fake messages transmitted in the whole communication at any time. Intruders can learn to infer more facts. Let's first define the set of facts that an intruder might learn.

$Fact =_{def} Sever \cup BaseStation \cup MobileStation \cup MSG.$

On the basis of the set of facts we have defined, we need to define how an intruder can deduce new facts from the facts it has learned. We use I to denote the set of facts that intruders have learned, and f to denote new facts that can be deduced from the set of facts I. Here we define a set T bigger then I which could be deduced the truth f.

An intruder may obtain facts from existing messages without any reasoning, and then use these facts to forge false messages to attack the routing process. To simplify the model, we assume that the intruder knows the fundermental labels $Info$.

Before the modeling, we have to define an additional deduce channel to represent the reasoning channel used by the intruder to infer new facts from the set of facts, namely: $Channeldeduce : Fact.P(Fact)$. The behavior of the intruder is modeled as follows:

$$
\begin{aligned}
&Intruder(I) =_{def} \\
&\quad \Box_{m \in MSG}\ ComMS.m \rightarrow Intruder(T \cup Info(m)) \\
&\quad \Box_{m \in MSG}\ ComBS.m \rightarrow Intruder(T \cup Info(m)) \\
&\quad \Box_{m \in MSG}\ ComMB.m \rightarrow Intruder(T \cup Info(m)) \\
&\quad \Box_{m \in MSG}\ InterceptB\_ComBS.m \rightarrow Intruder(T \cup Info(m)) \\
&\quad \Box_{m \in MSG}\ InterceptB\_ComMB.m \rightarrow Intruder(T \cup Info(m)) \\
&\quad \Box_{m \in MSG}\ InterceptM\_ComMS.m \rightarrow Intruder(T \cup Info(m)) \\
&\quad \Box_{m \in MSG}\ InterceptM\_ComMB.m \rightarrow Intruder(T \cup Info(m)) \\
&\quad \Box_{m \in MSG}\ InterceptS\_ComMS.m \rightarrow Intruder(T \cup Info(m)) \\
&\quad \Box_{m \in MSG}\ InterceptS\_ComBS.m \rightarrow Intruder(T \cup Info(m)) \\
&\quad \Box_{m \in MSG, Info(m) \subseteq T}\ FakeM\_ComMS.m \rightarrow Intruder(T) \\
&\quad \Box_{m \in MSG, Info(m) \subseteq T}\ FakeB\_ComBS.m \rightarrow Intruder(T) \\
&\quad \Box_{m \in MSG, Info(m) \subseteq T}\ FakeS\_ComBS.m \rightarrow Intruder(T) \\
&\quad \Box_{m \in MSG, Info(m) \subseteq T}\ FakeM\_ComMB.m \rightarrow Intruder(T) \\
&\quad \Box_{m \in MSG, Info(m) \subseteq T}\ FakeS\_ComMS.m \rightarrow Intruder(T) \\
&\quad \Box_{m \in MSG, Info(m) \subseteq T}\ deduce.f.T \rightarrow Intruder(T \cup f)
\end{aligned}
$$

In the above intruder model, because dedection is an intruder's internal action, we hide the channel and get the intruder's action behavior model.

$Intruder(I) =_{def} Intruder(I)\ [|\{deduce\}|]$.

3.3.6 The whole system

First of all, we assume there is no intruder.

$$
\begin{aligned}
&SERVER =_{def} Timer[|\{time\}|]Server(S); \\
&BASESTATION =_{def} Timer[|\{time\}|]BaseStation(B); \\
&MOBILESTATION =_{def} Timer[|\{time\}|]MobileStation(M); \\
&System =_{def} SERVER[|\{ComMS, ComBS\}|]MOBILESTATION \\
&\qquad [|\{ComMB\}|]BASESTATION.
\end{aligned}
$$

When there are intruders, the system is defined as follows:

$$SYSTEM =_{def} System[|\{InterceptM\_ComMS, InterceptM\_ComMB, InterceptB\_ComMB, InterceptB\_ComBS, InterceptS\_ComMS, InterceptS\_ComBS, FakeS\_ComMS, FakeS\_ComBS, FakeB\_ComBS, FakeB\_ComMB, FakeM\_ComMB, FakeM\_ComMS\}|]Intruder(I).$$

Chapter 4

Implelentation and Simulation of the PKMv3 Protocol

We have modeled the PKMv3 Protocol in CSP in Chapter 3. As the model checker PAT is based on CSP, we now apply CSP to model the PKMv3 protocol in PAT. The expressions in CSP models are replaced by the equivalent ones in PAT. We consider the mobile station, the base station and the intruder as a whole. In the following, we will first introduce the definition of variables and channels for the protocol model. Then we give the simulation of the model. We mainly analyze the communication process between the server and the mobile station and the base station, as well as the process of authorization request, security association establishment and TEK exchange between MS and the BS without considering the intruders. Next, the properties of the protocol will be described by assertions and LTL formula, as well as verifying whether the protocol meets the specification. Furthermore, intruders are taken into consideration, we explore whether there are some vulnerabilities, if so, we figure out corresponding solutions to guarantee the security of the communication.

4.1 Preliminaries

4.1.1 Notations

As is shown in Figure 1, there are numerous notations listed in the transition processes. Now we introduce some important notations and labels which are displayed expressively in Table 4.1.

- We use $Server$, MS, BS and I to represent the server, mobile station, base station and the intruder respectively.
- $Msg1$, $Msg2$, ... , $Msg10$ stand for the communication messages transmitted between MS and BS orderly. In order to distinguish different nonces sent by MS and BS, here we add numbers to identify each nonce. e.g., Nm in $Msg2$ is changed as $Nm0$ and Nb in $Msg5$ is substituted by $Nb2$. That is to say, as for BS, each time BS receives a message with the nonce of MS, the nonce is different from the received ones. It acts the same as MS. Each message is defined as an array in PAT so that every element of the message can be achieved by its location in order. For instance, $Msg2[2]$ refers to the third element of $Msg2$ which is $Capabilities$. Messages are defined just below.

  $Msg1 = \{MS\_Cert\}$

  $Msg2 = \{MS\_Cert, Nm0, ..., Sig\_MS\}$

  ...

  $Msg10 = \{SAID, AK\_SN, ..., wrongreasons\}$

- $msg1$, ... , $msg10$ are pre-defined arrays which are in correspondence with the sent messages $Msg1$, ... , $Msg10$, whose sizes are the same. They represent the received messages of the MS and BS.
- We still need some variables to act as time intervals which have real numbers to be in concordance with the protocol standard. Declarations of variables are given as follows:

  $\#define\ MaxResends\ 60;$

  $\#define\ MAX\ 60;$

- Counters should also be defined as variables in PAT to record the times of MS resending $Msg5$ and BS resending $Msg6$. Declarations are shown below:

  $var\ count_1 = 0;$ $\quad$ $var\ count_2 = 0;$

TABLE 4.1: Message Contents

Element	Meaning
X_Cert	The certificate of X (X could be MS, BS or I)
Nm, Nb, Ni	Nonces of MS, BS and intruder I
Capabilities	Cryptography capability
{PMK}pk{MS}	Using the public key of MS to encrypt PMK
PMK_Lifetime	The lifetime of PMK
SAIDList	The list of SAID that MS can use
Sig_X	X's digital signature
MSAddr	The MAC address of MS
AKID, SAID	Identities of AK and SA
{TEK}kek	Using Key Encryption Key (KEK) to encrypt TEK
COUNTER_TEK	Deriving current uplink TEK
EKS	Encryption key sequence number
SATEK_Timer	Time delay between SA-TEK-Request and SA-TEK-Response
MaxResends	Max times for the reputations of sending SA-TEK-Request by MS
SAChallengeTimer	Time delay between SA-TEK-Challenge and SA-TEK-Request
MAX	Max times for reputations of sending SA-TEK-Challenge by BS

4.1.2 Sets and Functions

Being numerous mobile stations, base stations and intruders in the real world, we define the following sets and the combination of messages which are used in the BS and MS's communications. **MobileStation** is the set of mobile stations and the couple of base stations are expressed in the set **BaseStation**. Also, **Intruder** represents the set of intruders as well as **Server** stands for servers. As is mentioned before, $Msg1$ - $Msg10$ are ten possible messages sent between MS and BS. These messages are included in the set: **MSG**$=_{df} Msg1 \cup Msg2 \cup ... \cup Msg9 \cup Msg10$.

Moreover, We define two functions in PAT to implement the encryption and decryption of the message.

- MAC_DValue is used for computing a new CMAC value by BS. The base station will compare the attached CMAC value in the received message with the computed one. Moreover, BS also uses MAC_DValue to calculate the sending message to get a CMAC value and then attaches it with the whole message, transferring to the valid receiver.
- MAC_UValue acts the same as MAC_DValue except that it is used by MS.

For example, the CMAC value which will be sent with $Msg5$ and the new CMAC which is created to test the consistency are displayed in the following form:

$CMAC\_1 = MAC\_DValue(CMAC\_KEY\_D, Msg5);$
$CMAC\_1\_new = MAC\_UValue(CMAC\_KEY\_U, Msg5).$

4.1.3 Channels

We use five channels in the PKMv3 protocol model: *comm*, *intercept*, $fake$, *session* and $fake\_session$.

- *comm*: It is the common communication channel which is used by MS and BS, or between the intruder and BS.
- *intercept*: The intruder uses this channel to intercept messages from the *comm* channel so that the real receiver cannot get the message.
- $fake$: It is used to send forged messages to MS or BS from intruders.
- *session*: It indicates that the communication successfully ends.
- $fake\_session$: This session is interfered by intruders.

4.2 Timer

Time is shared in the whole transition processes. It is used not only to synchronize the MS, BS and the intruder, but also check the time delay. The implementation of timer in PAT is quite similar with the description using CSP.

$Timer(i) =_{def} (tick \rightarrow Timer(i+1))$
$\square(time?request \rightarrow time!i \rightarrow Timer(i));$
where, $i \geq 0 \wedge i \in N$..

4.3 Mobile Station

As a mobile station, there are two nodes to be connected with, the server and the base station. MS can send messages through the common channel *comm* or the messages can be acquired by the intruder through the channel *intercept*. Similarly, messages which are sent from the base station, can be received in the *comm* channel or from the channel $fake$. Due to several transition loops in the communication, we separate the model of MS into four sub-processes while communicating with BS. The first process $M1()$ represents the initialization and AK authorization. As it is essential for MS to check whether the received nonce is fresh, we use $M2()$ to test the following processes. Process $M3()$ is created to examine the time interval between the sent and received messages. And $M4()$ is the description of last period, the exchange of TEK. Each process is executed sequentially.

4.3.1 MS with Server

MS asks for its certificate from the server and once gets it, the session is done.

$M() = ms!requestM \rightarrow ms?b\{b\_in = b\} \rightarrow STOP.$

Here, b is the variable to store the recieved message, i.e., the certificate of MS. ms is the session channel.

4.3.2 MS with BS

Firstly, the mobile station sends its certificate ($Msg1$) to start the communication with the ideal base station through the channel *comm*. Then MS will launch the AK authorization. In $Msg2$, there is a nonce of MS named $Nm0$, it is distinguished from $Nm1$ in $Msg6$ and $Msg7$. After that, MS will receive $msg3$ which contains two nonces. Here $msg3[0]$ represents the first element in the received message $msg3$. If it equals to $Nm0$, MS will go on transmitting the confirmation message, otherwise, MS will end the

process. These four messages' transmissions are included in $M1()$. In order to continue the transmission of SA, we add $M2()$ in the sequential order. This part is designed in PAT as follows:

$$
\begin{aligned}
&M1() =_{df} ((comm!M.B.Msg1 \rightarrow Skip)\\
&\quad \Box(intercept!M.B.Msg1 \rightarrow Skip));\\
&\quad ((comm!M.B.Msg2 \rightarrow Skip)\\
&\quad \Box(intercept!M.B.Msg2 \rightarrow Skip));\\
&\quad ((comm?B.M.msg3 \rightarrow Skip)\\
&\quad \Box(fake?B.M.msg3 \rightarrow Skip));\\
&\quad if(msg3[0]! = Nm0)\ \{STOP\}\\
&\quad else\{((comm!M.B.Msg4 \rightarrow Skip)\\
&\qquad \Box(intercept!M.B.Msg4 \rightarrow Skip)); M2()\};
\end{aligned}
$$

After BS verifies the nonce sent by the sender, MS then will obtain $msg5$ which includes a new nonce of BS and the CMAC value. With the CMAC_KEY_U key derived from AK, MS computes the CMAC value by calling the MAC_UValue function, and then compares it with the received one. If they are the same, both sides will continue the conversation. This method is effective for the following procedures. We use $M2()$ to test and connect with the following processes and detailed information is shown below:

$$
\begin{aligned}
&M2() =_{df} ((comm?B.M.msg5.cmac1 \rightarrow Skip)\\
&\quad \Box(fake?B.M.msg5.cmac1 \rightarrow Skip));\\
&\quad \{CMAC1\_new =\\
&\quad\ call(MAC\_UValue, CMAC\_KEY\_U, msg5)\} \rightarrow\\
&\quad if(cmac1! = CMAC1\_new)\ \{STOP\}\\
&\quad else\ if(msg5[0] == msg3[1])\{M2()\}\\
&\quad\ else\{M3()\};
\end{aligned}
$$

Process $M3()$ is designed to examine the time interval between the sent and received messages. It should be noted that according to the IEEE 802.16 standard, after the mobile station sends $Msg6$, it will ask the $Timer$ for the current time. We model this part as follows:

$M3() =_{df} time!request \rightarrow time?T1;$
$\{CMAC2 =$
$call(MAC\_UValue, CMAC\_KEY\_U, Msg6)\} \rightarrow$
$((comm!M.B.Msg6.CMAC2 \rightarrow Skip)$
$\Box(intercept!M.B.Msg6.CMAC2 \rightarrow Skip)); M4();$

Until receiving $msg7$, MS also needs the time so as to check the time delay. If the interval is more than $SATEK\_Timer$, the counter $counter_1$ will be added and call $M3()$ again. MS will keep on sending $Msg6$ and then receiving $msg7$ until the time delay is less than $SATEK\_Timer$. If $counter_1$ is more than $MaxResends$, MS will not try to communicate with BS anymore. Relevant codes of the last part $M4()$ are represented below:

$M4() =_{df} ((comm?B.M.msg7.cmac3 \rightarrow Skip)$
$\Box(fake?B.M.msg7.cmac3 \rightarrow Skip));$
$time!request \rightarrow time?T2;$
$\{CMAC3\_new =$
$call(MAC\_UValue, CMAC\_KEY\_U, msg7)\} \rightarrow$
$if(msg7[0]! = msg6[0] || CMAC3\_new! = cmac3)\{STOP\}$
$else\ if(msg7[1] == msg5[1] || msg7[1] == msg3[1])\{M4()\}$
$else\{if(T2 - T1 > SATEK\_Timer)\{$
$\{count_1 = count_1 + 1\} \rightarrow$
$if(count_1 > MaxResends)\{STOP\}$
$else\{M3()\}\}$
$else\{\{CMAC4 =$
$call(MAC\_UValue, CMAC\_KEY\_U, Msg8)\} \rightarrow$
$((comm!M.B.Msg8.CMAC4 \rightarrow Skip)$
$\Box(intercept!M.B.Msg8.CMAC4 \rightarrow Skip));$
$((comm?B.M.msg9.cmac5 \rightarrow Skip)$
$\Box(fake?B.M.msg9.cmac5 \rightarrow Skip));$
$\{CMAC5\_new =$
$call(MAC\_UValue, CMAC\_KEY\_U, msg9)\} \rightarrow$
$if(CMAC5\_new! = cmac5)$
$\{fake\_session?B.M.m\}$
$else\{session?B.M.m\}\}\};$

4.4 Base Station

Modeling of the base station is similar to that of the mobile station. Firstly, BS starts a conversation with the server to apply for the certificate. Once BS receives, it can have a session with the MS. We disintegrate the whole model of BS into three parts: $B1(sender)$, $B2(sender)$, $B3(sender)$. We have mentioned that the aim of separating the model is to create and clarify the model more easily and clearly. Owing to the existence of intruders, there is a parameter in the process of BS to indicate the initiator of the communication. Both mobile stations and intruders can initiate sessions with the base station so that BS needs to judge and sends the corresponding messages.

4.4.1 BS with Server

BS asks for its certificate from the server, saveing it to a variable v through channel bs.

$$M() = ms!requestM \rightarrow ms?b\{b\_in = b\} \rightarrow STOP.$$

4.4.2 BS with MS

Process $B1(sender)$ corresponds to the process $M1()$ as we have introduced before and this process contains the first four messages' transmissions.

$$\begin{aligned} B1(sender) =_{df} & ((comm?sender.B.msg1 \rightarrow Skip) \\ & \Box\ (fake?sender.B.msg1 \rightarrow Skip)); \\ & ((comm?sender.B.msg2 \rightarrow Skip) \\ & \Box\ (fake?sender.B.msg2 \rightarrow Skip)); \\ & ((comm!B.sender.Msg3 \rightarrow Skip) \\ & \Box\ (intercept!B.sender.Msg3 \rightarrow Skip)); \\ & ((comm?sender.B.msg4 \rightarrow Skip) \\ & \Box\ (fake?sender.B.msg4 \rightarrow Skip)); \\ & if(msg4[0] \neq Msg3[1])\{Stop\}\ else\{B2(sender)\}; \end{aligned}$$

In the $B2(sender)$ process, firstly the BS needs to ask the current time after receiving the confirmation message. The time, which is used to check the time delay of sending and receiving the sixth message, is stored in $T3$. In the following, BS calculates the

CMAC value and attaches it with $Msg5$ and then sends this message to the sender. $B2(sender)$ is described in PAT as below:

$$
\begin{aligned}
&B2(sender) =_{df} time!request \rightarrow time?T3;\\
&\quad \{CMAC1 =\\
&\quad call(MAC\_DValue, CMAC\_KEY\_D, Msg5)\} \rightarrow\\
&\quad ((comm!B.sender.Msg5.CMAC1 \rightarrow Skip)\\
&\quad \Box(intercept!B.sender.Msg5.CMAC1 \rightarrow Skip));\\
&\quad B3(sender);
\end{aligned}
$$

As for $B3(sender)$, it performs part of the transmission of SA and the whole exchange of TEK procedures. After sending $Msg5$, BS will wait for the next message and check the CMAC values. Having obtained $msg6$, BS immediately asks for the current time and reserves it in $T4$. It is necessary to check whether $T4-T3$ is less than the specified time $SAChallengeTimer$. If the time delay is too much, BS will go back to the $B2(sender)$ and do the examination loop, otherwise, performing the exchange of TEK. Because the communication ways of MS and BS are alike and we have introduced the model of MS detailedly, here we omit the display of $B3(sender)$.

$$
\begin{aligned}
&B3(sender) = ((comm?sender.B.msg6.cmac2 \rightarrow Skip)\\
&\quad \Box(comm?sender.B.msg6.cmac2 \rightarrow Skip))\\
&\quad \rightarrow \{CMAC2\_new = call(MAC\_DValue, CMAC\_KEY\_D, msg6\}\\
&\quad \rightarrow time!request \rightarrow time?T4\\
&\quad \rightarrow if(msg6[1] \neq Msg5[0] || cmac2 \neq CMAC2\_new)\{STOP\}\\
&\quad\ else\{if(msg6[0] == msg2[1])\{B2(sender)\}\\
&\quad\quad else\{if(T4 - T3 > SAChallengeTimer)\{\{count_2 = counts_2 + 1\}\\
&\quad\quad\quad \rightarrow if(count_2 > MAX)\{STOP\}\ else\{B1(sender)\}\}\\
&\quad\quad\quad\ else\{\\
&\quad\quad \{CMAC3 = call(MAC\_DValue, CMAC\_KEY\_D, Msg7)\}\\
&\quad\quad\ \rightarrow ((comm!B.sender.Msg7.CMAC3 \rightarrow Skip)\\
&\quad\quad\quad \Box(intercept!B.sender.Msg7.CMAC3 \rightarrow Skip))\\
&\quad\quad\ \rightarrow ((comm?sender.B.msg8.cmac4 \rightarrow Skip)\\
&\quad\quad\quad \Box(fake?sender.B.msg8.cmac4 \rightarrow Skip))\\
&\quad\quad \rightarrow \{CMAC4\_new = call(MAC\_DValue, CMAC\_KEY\_D, msg8)\}\\
&\quad\quad \rightarrow if(CMAC4\_new == cmac4)\{\\
&\quad\quad\quad \{CMAC5 = call(MAC\_DValue, CMAC\_KEY\_D, Msg9)\}\\
&\quad\quad\quad\ \rightarrow ((comm!B.sender.Msg9.CMAC5\\
&\quad\quad\quad\quad \rightarrow session!sender.B.success)\\
&\quad\quad\quad\quad \Box(intercept!B.sender.Msg9.CMAC5)) \rightarrow \{Skip\}\}\\
&\quad\quad\quad\ else\{CMAC6 = call(MAC\_DValue, CMAC\_KEY\_D, Msg10)\}\\
&\quad\quad\quad\ \rightarrow ((comm!B.sender.Msg10\\
&\quad\quad\quad\quad \rightarrow fake\_session!B.sender.failure)\\
&\quad\quad\quad\quad \Box(intercept!B.sender.Msg10.CMAC6)) \rightarrow \{Stop\}\}\}\}\};
\end{aligned}
$$

4.5 Server

Firstly, the server receives the request of certificates from MS or BS. Then it sends them back, which is encrypted with the sender's public key.

$$
\begin{aligned}
&Server() = ms?m\{m\_in = m\} \rightarrow ms!CertM \rightarrow SKIP\\
&\quad\quad \Box\ bs?n\{n\_in = n\} \rightarrow bs!CertB \rightarrow SKIP;
\end{aligned}
$$

4.6 Intruder

Now, we will introduce another entity, the intruder, which is modeled as a process. The intruder can overhear, intercept and fake messages when there is a session between communication entities. With the ability of learning, the intruder may acquire much more knowledge. Capabilities of intruders are far more than that:

1. Intruders know about each unencrypted element of the intercepted message.
2. Using its private key, the intruder can easily decrypt the messages that are encrypted by its public key and learn new knowledge.
3. Through analyzing the received message and the available knowledge, new messages can be faked by intruders and may be sent to session participants.

Before modeling the intruder, some assumptions and notations will be explained. The intruder can not only be a malicious node who is harmful to communications but also be a normal node who can sponsor a session with the nearby BS. Besides, we use $km1$, $km2$ to track and denote what new things the intruder gets. The values of these two variables are defined as $false$ at first.

First of all, we introduce the behavior of intercepting messages. Once receiving new things from the mobile station or base station, the fresh knowledge is also obtained by the intruder being as $km1 = true$ or $km2 = true$. Intercepting behaviors of the intruder are displayed in the following:

$$Intruder\_Intercept() =_{df} (intercept?M.B.m1; \{km1 = true\} \rightarrow Intruder()) \\ \square (intercept?B.M.m2; \{km2 = true\} \rightarrow Intruder());$$

Additionally, there is a faking case. After checking what knowledge it owns, the intruder can easily fake new messages and send to either side or both sides:

$$Intruder\_Fake() =_{df} (if(km1 == true)\{fake!M.B.msg1 \rightarrow Intruder()\}) \\ \square (if(km2 == true)\{fake!B.M.msg2 \rightarrow Intruder()\});$$

As is concerned in the assumption, the intruder can also be a legal entity to communicate with BS. Because the action of a valid intruder is similar as that of MS, we will not show its specific model here.

$$Intruder\_MS() =_{df} \{comm!I.B.MSG1 \rightarrow Skip\};$$
$$\{comm!I.B.MSG2 \rightarrow Skip\};$$
$$\{comm?B.I.msg3 \rightarrow Skip\};$$
$$\{kNb = true\} \rightarrow ... \rightarrow Intruder();$$

Here, $MSG1$, ... , $MSG10$ are a bit different from $Msg1$ to $Msg10$. The certificate and nonces of the intruder are distinct from MS's. Having represented intruders possible actions above, we know that behaviors of the intruder are random choices in intercepting, faking and natural status. After intercepting one message, the intruder can not only learn new data and transfer it without altering anything, but also fake another message and send it to the receiver. Here the variable kNb is also a tracer which functions like $km1$. The kNb is used to mark whether the corresponding nonce is acquired by the intruder. We now define the whole model of the intruder:

$$Intruder() =_{df} Intruder\_Intercept() \square Intruder\_Fake() \square Intruder\_MS();$$

4.7 System

Mobile station and base station are modeled in parallel in the whole system and both of them share the same entity $Timer$. The synchronized MS and BS start a conversation using common channels such as $comm$. While, with intruders, the channels such as $intercept$ and $fake$ are applied in the whole system. Here we suppose the sender of the BS is a mobile station M. We define two systems. One is the parallel excecution among MS, BS and the server. The other one includes the intruder but except for the server. The model of the systems are displayed as follows:

$$System0() =_{df} M1()\|B1(M)\|Server;$$
$$System() =_{df} M1()\|B1(M)\|Intruder()\|Timer(0);$$

4.8 Simulation

In this section, we firsly perform the simulation of the protocol model, to make sure our design and implementation is according to the protocol specification. Two simulation diagrams are shown below. The simplier one in Fig.4.1 indicates that the base station first requests a certificate to the server and receives a certificate, then a mobile station

requests a certificate to the server, and a certificate is obtained after the termination of the communication.

The second simulation graph is run out with MS and BS. Because we just assume no intruders here, we simplify the channel in the model as *mb*. As is shown in Fig.4.2, first of all MS sends BS its own certificate, followed by the authorization request message to BS. After the verification of the MS, BS sends a confirmation message to build the mutual authentication. Then, the base station establish a secure connection between BS through the three handshake. The challenge message is sent from BS to MS attached with CMAC code. Once MS receives this message, after the calculation and testifies this news has not been tampered with, it will continue the following communication. The simulation result shows that the formalization and of this protocol is consistent with the protocol specification.

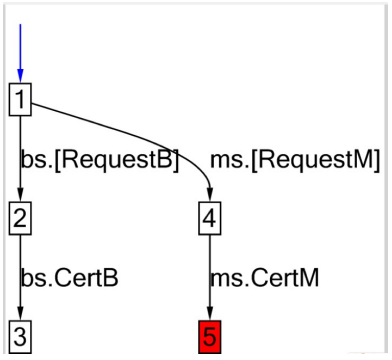

FIGURE 4.1: Simulation of MS/BS and Server

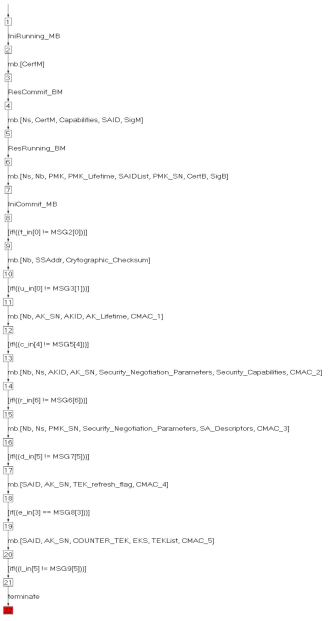

FIGURE 4.2: Simulation of MS and BS

Chapter 5

Verification and Analysis of the PKMv3 Protocol

Having modeled the PKMv3 protocol in the last chapter, we need to check whether the model conforms to our expectations. Moreover, with the assumption that there is an intruder, exploring what kind of attacks the protocol will expose to is also of great importance. For these reasons, we use PAT to extract and implement various properties using LTL formulae and assertions.

Things become complicated with the appearance of more attacks such as man-in-the-middle attacks and replay attacks. With the aim of examining whether the model obeys the protocol's specification and detecting the situations of the existence of intruders, we put forward some properties, through analyzing the traces of the performance, giving some advice to improve the security of message transimssion.

5.1 Verification

Six properties, deadlock freedom, starvation freedom, consistency, timout freedom, safety, secrecy violation verified in assertions and LTL formulae are introduced in detail as follows:

5.1.1 Deadlock Freedom

Deadlock [33] refers to a blocking phenomenon caused by competing resources or communication between two or more processes in the execution process. In the process of communication, deadlock must be avoided, that is to say, the communication will not be interrupted because of competition for resources. We use the PAT tool to verify the deadlock freedom property of the protocol by assertion:

$$\#assert\ System()\ deadlockfree;$$

5.1.2 Connectivity

Once a commnication node build connections with another one, they will send and receive messages through a channel freedomly. In other words, one can receive the message because there is someone else who has sent it. This property indicates that the message transmission works on the basis of the connection of both nodes. Here come the LTL formulae to express this property:

$$\begin{aligned}
&\#assert\ System()| = \Box((\Box!resCommitBI) \\
&\qquad\qquad \|(!resCommitBI\ U\ iniRunningIB)); \\
&\#assert\ System()| = \Box((\Box!resCommitBM) \\
&\qquad\qquad \|(!resCommitBM\ U\ iniRunningMB)); \\
&\#assert\ System()| = \Box((\Box!iniCommitMB) \\
&\qquad\qquad \|(!iniCommitMB\ U\ resRunningBM)); \\
&\#assert\ System()| = \Box((\Box!iniCommitIB) \\
&\qquad\qquad \|(!iniCommitIB\ U\ resRunningBI));
\end{aligned}$$

Here, $System\ |=$ shows the whole system satisfies the following property. The symbol U represents 'until' and $\Box$ stands for 'always'. As for the first assertion, it means there always exists the case that $resCommitBI$ will not be true until $iniRunningIB$ is $true$. In other words, the base station will not receive a message until one intruder acting as a normal MS sends to it. The third assertion represents that only if the base station starts a conversation with the MS, will it receive the message and $iniCommitMB$ turn $true$. The meaning of the rest two assertions are much more similar with the introduced ones. These LTL formulae demonstrate that the message sent and received during a commnication is in a specific order and it is based on the built connection.

5.1.3 Consistency

It is of great importance that during a session, the sent and received messages should be the same, which can be called the message consistency, is also the core value to a security protocol of the mobile network. To check this property, we set a global definition as a condition so as to apply an assertion to explore whether the system conforms to this property. In our model, we suppose each received message is exactly the same as the message to be sent. If the system reaches this property, it means the whole session is successful and no message is faked. Even if there is an intruder, we can also discover whether the message has been tampered with or not. The concrete procedures for the verification of message consistency is shown as follows:

$$
\begin{aligned}
&\#define\ consistency\ ((msg1 == Msg1)\|\\
&\qquad ((msg1 == Msg1)\&\&(msg2 == Msg2))\|\\
&\qquad\qquad ...\\
&\qquad ((msg1 == Msg1)\&\&\ ...\ \&\&(msg9 == Msg9));\\
&\#assert\ System()\ reaches\ consistency;
\end{aligned}
$$

At first, one variable called *consistency* is defined, where $msgx$ $(x = 1, ..., 9)$ represents the message received by the mobile station or the base station. Each received message is compared with the sent message before. In view of the existence of the intruder, we use "or" ("‖") to list all possible equations from $msg1 == Msg1$ to $msg9 == Msg9$. Finally, the predefined keyword in PAT *reaches* is used to test the result.

5.1.4 Delay Freedom

One way to detect the existence of man-in-the-middle attacks is to check the transmission time intervals during a commnication. Our defined timeout freedom property plays an important role in the verification of the PKMv3 protocol. There is a flag set in the model of stations, which could inspect the exact time of receiving a message. As is known, there are several messages transmitted during a session, we set many flags behind each received message. For example, a $flag$ which is $false$ initially, is set after accepting $msg2$ in the process of BS. Once the time of receiving the second message is equal or more than 2 units of time, the $flag$ changes into $true$. On account of the interception behaviour, it will take twice or more time for the receiver to obtain the message. In this way, it is

much easier to find the existence of the man-in-the-middle attack. In the following, we define a variable *timedelay*, representing that there is an intruder posing a threat on the security of the transmission. And we just hypothesize the system is immune against such kind of attacks:

$$\#define\ timedelay\ (flag == true);$$
$$\#assert\ System| = \Box!(timedelay);$$

5.1.5 Safety

This property means either communication entity will continuously receive two same messages within a certain amount of time. The time span should be less than $SATEK\_Timer$ or $SAChallengeTimer$, which is the maximum time interval of the MS sending SA-TEK-Request and receiving SA-TEK-Response messages, or the BS sending SA-TEK-Challenge and receiving SA-TEK-Request messages. If the model conforms to this property, the receiver will have more space to store useful information rather than the same messages.

$$\begin{aligned}
&\#define\ safety(!(msg1 == msg2)\|\| \\
&\quad !((msg1 == msg2)\&\&(msg2 == msg4))\|\| \\
&\quad !((msg1 == msg2)\&\&(msg2 == msg4)\&\& \\
&\quad\ (msg3 == msg5))\|\| \\
&\quad !((msg1 == msg2)\&\&(msg2 == msg4)\&\& \\
&\quad\ (msg3 == msg5)\&\&(msg4 == msg6))\|\| \\
&\qquad\qquad ... \\
&\quad !((msg1 == msg2)\&\&(msg2 == msg4)\&\& \\
&\quad\ (msg3 == msg5)\&\&(msg4n == msg6)\&\& \\
&\quad\ (msg5 == msg7)\&\&(msg6n == msg8)\&\& \\
&\quad\ (msg7 == msg9)); \\
&\#assert\ System()\ reaches\ safety;
\end{aligned}$$

5.1.6 Secrecy Violation

The private information shoubld be secret. If it is intercepted by an intruder, the security of the communication will be threatened. Therefore, it is necessary to check the protocol does not conform to the secrecy violation. As mentioned in Chapter 4, some

secret information like kNm are defined in advance. They are used to test whether the secrecy, like nonces, are leaked. We apply the following assertion to implement the test as an example:

$$\#define\ secrecy\_violation\ (kNm == true);$$
$$\#assert\ System| = \Box!(secrecy\_violation);$$

All the variables are defined as $false$ initially, which means the intruder is blind to Nm at first. After capturing or intercepting the message with this nonce, kNm turns to $true$, which means Nm is transparent to the intruder and there is a probability that it will use this nonce to fake a message and send it to BS. This assertion in PAT suggests that this secret nonce is always safe and beyond the reach of any intruders.

5.1.7 Verification Results

The verification results of the PKMv3 protocol model is shown in Fig.5.1 and Fig.5.2, representing the results of deadlock freedom and the other five properties respectively. Except for the properties secrecy violation and delay freedom, the rest are all valid. The secrecy violation property indicates that there is an intruder starting a conversation with BS and BS sends his nonce in one of the messages. In this way, the intruder obtains the secret information and it may do some malicious actions to other commnications in the future. Furthermore, the possible reason for the invalidation of the delay freedom might bet the intruder firstly intercepting the message from one station and then sending it to the other station. During the message forward, time delay is generated.

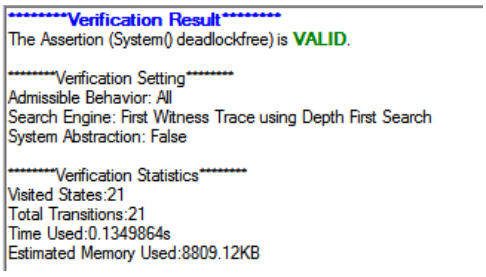

FIGURE 5.1: Deadlock Freedom Result

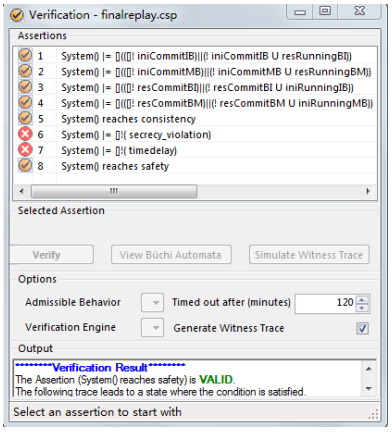

FIGURE 5.2: The verification results of the five properties

5.2 Analysis of Attacks to the PKMv3 Protocol

PKMv1 and PKMv2 protocols are proved to be vulnerable against some attacks [34], so does PKMv3 protocol. Based on the verification results, we discovered that the protocol is susceptible to some attacks. With the study of the excution traces and counterexamples, the model may be exposed to man-in-the-middle attacks because the time delay and the leak of secrecy information. DoS attack is eliminated with the improvement of PKMv2 protocol, so this attack is not taken into consideration. At last, we will also make an analysis on the possibility of the replay attacks.

5.2.1 Man-in-the-Middle Attack

As a kind of indirect attack, man-in-the-middle attacker could be BS or MS, even the server. Therefore, there are two situations that intruders may influence the communication. One is in the process of certificate request and reply among BS, MS and the server, the other one is between the session of MS and BS.

Situation one: One MS wants to apply for the certificate and it sends the request message to the relevant Server. However, the intruder intercepts the message, knowing that this MS will start a session with one BS. Then the intruder pretends to be MS to send the request message to the server. Because the certificate contains some useful information like the public key of MS, the intruder might be eager to get the certificates of the entities during the communication. After the server receives the request and verifies the senders identity, the certificate will be encrypted by the server's secret key. Here we assume the MS and BS nearby the server know the public key of the server, therefore, MS could decrypt the certificate and then build connections with the BS. Here comes the problem. The intruder can also act as a legal MS so as to get the public key of the server, which leads to obtaining the certificate of MS without effort. There are the possible traces of intruders in the following:

Case I:

$$< init \rightarrow intercept.M.S.[requestM] \rightarrow fake.M.S.[requestM]$$
$$\rightarrow intercept.S.M.[cert\_M] \rightarrow ms.S.M.[cert\_M] \quad >$$

To prevent the intruder getting the certificate of stations, we propose an approach by using different keys to encrypt the certificate. The server knows the public key of the

stations, therefore, it could encrypt by using the senders public keys after checking the identification. The certificate cannot be decrypted without the secret key, which is only possessed by the MS. In this way, both MS and BS can receive the certificate safely.

Situation Two: The intruder acts as a mobile station or a base station between two communication entities. Messages transmitted between both sides may be intercepted, tampered with or sniffed by intruders. In the following, we will introduce three possible kinds of man-in-the-middle attacks.

Case I: The intruder personates the BS and intercepts the unencrypted message which contains the certificate of MS. It is a kind of secrecy leak and poses a threat on the security of the stations as well as the communication. Here, we consider two situations:

1. The MS has the public key of BS before. Therefore, the certificate could be encrypted with BS's public key by MS. After receiving the message, BS can use its private key to get and verify the identification of the sender. So our solution is to modify the $Auth\ Information$ message as $\{Cert\_M\}pk\{BS\}$.

2. The public key of BS is not known by MS yet. In this case, it is practical if the server could generate one pair of keys, which is only shared with MS and BS. When MS sends the request to the server, it can also ask for the shared keys. Then the certificate and the keys, encrypted by the public key of MS, are sent to MS by the server. Both methods could guarantee the safety of the stations' certificates and make a good beginning of the following commnication.

In the second case, the intruder just intercepts the message and then transfers it. We offer the Case II below with a communication trace and simplified graph.

Case II

$$
\begin{aligned}
&< init \rightarrow comm.M.B.[MS\_Cert] \\
&\quad \rightarrow intercept.M.B.[Nm0, Capabilities, ..., SAID, Sig\_MS] \\
&\quad \rightarrow fake.M.B.[Nm0, MS\_Cert, Capabilities, SAID, Sig\_MS] \\
&\quad \rightarrow time.request \rightarrow time.4 \\
&\quad \rightarrow comm.B.M.[Nm0, Nb0, \{PMK\}pk\{MS\}, ..., Sig\_BS] \\
&\quad \rightarrow ... \\
&\quad \rightarrow comm.B.M.[SAID, AK\_SN, ..., wrong\ reasons] \\
&\quad \rightarrow fake\_session \quad >
\end{aligned}
$$

This trace containing the attack can be rewritten to be more legible as follows:

$Message\ 1\ MS \rightarrow BS \quad : M.B.Msg1$

$Message\ 2\ MS \rightarrow I_{BS} \quad : M.B.Msg2$

$Message\ 3\ I_{MS} \rightarrow BS \quad : M.B.Msg2$

$Message\ 4\ BS \rightarrow MS \quad : B.M.Msg3$

$Message\ ...$

$Message\ 9\ BS \rightarrow MS \quad : B.M.Msg9$

After MS sends $Msg2$, the intruder pretends to be the BS and then transmits the received message to the real BS. Although it does not bring out some errors to the transmission process, the time consumed in the session becomes longer. So it is more likely to affect the communication efficiency. Besides, it is a waste of time for other mobile stations to establish connections with the same BS. For this case, it is just because the intruders can intercept and send messages from the open network, specifically the common channels. So, we think it is a good method to encrypt the commnication channel by using relevant mechanism.

Case III

As discussed above, another kind of man-in-the-middle attack may occur in our model. If the intruder owns the ability of learning, falsifying and overhearing the message, the situation of attack in the simplified form is shown as follows:

$Message\ \alpha 1\ I\ \rightarrow BS : I.B.[I\_Cert]$

$Message\ \alpha 2\ I\ \rightarrow BS : I.B.[Ni0, I\_Cert, ..., Sig\_I]$

$Message\ \alpha 3\ BS \rightarrow I\ : B.I.[Ni0, Nb0, ..., \{PMK\}pk\{I\}]$

$Message\ \alpha 4\ I\ \rightarrow BS : I.B.[Nb0, ..., IAddr]$

$Message\ \beta 1\ MS \rightarrow I_{BS} : M.B.[MS\_Cert]$

$Message\ \beta 2\ MS \rightarrow I_{BS} : M.B.[Nm0, MS\_Cert, ..., Sig\_I]$

$Message\ \beta 3\ I_{BS} \rightarrow MS : B.M.[Nm0, Nb0, ..., \{PMK\}pk\{MS\}]$

$Message\ \beta 4\ MS \rightarrow I_{BS} : M.B.[Nb0, ..., MSAddr]$

$Message\ ...$

To represent this case, we introduce two separated conversations. One session is between an intruder and the BS, while the other one happens between the intruder and the MS. In the α conversations above, initially the intruder launches a conversation as a normal entity with BS by sending its own certificate, and then it transmits $Message\ \alpha 2$ to offer more details about its abilities. After checking the identity of the sender, BS responds

with $Message\ \alpha 3$, containing two nonces and the PMK, which is encrypted by the sender's public key. Once acquiring this message, the intruder will decrypt it by using its private key and get PMK. Later it returns the confirmation message to the BS. The goal of this part is to gain PMK so as to take advantage of it in the next stage.

While running β, the intruder impersonates itself as the base station, aiming at building connections with the MS. After receiving first two β messages from MS, the intruder uses MS's public key to encrypt PMK, which is obtained from the last section, and then sends $Message\ \beta 3$ to MS. Similarly MS will send the confirmation news to BS, who is disguised by the intruder. Up till now, the MS thinks that itself is communicating with a legal BS and it has owned the private PMK, which is only known by the BS and itself. However, this is not the case. Once MS initiates another session with the same BS, it is likely that they generate the same PMK, with which MS will get the TEK to encrypt communication packages. The intruder, possessing the same TEK, might decrypt the package with little hindrance, posing a treat on the following communicating between the MS and BS.

Timestamp [16, 35] is a string generated according to the current time of the server (or the stations), together with nonce, which can represent the random number generated at a certain point of time. In this way, even if the two random nonce are generated at the same time, they are also valid because they are generated at different time points. To prevent the above safety loophole, we put forward a method which attaches the timestamp from the *AuthRequest* message. If the intruder commnicates with BS earlier than MS, what messages it obtains have a different timestamp with that of MS's apparently. Therefore, MS can easizly realize whether the message is sent by the real BS. Also, it strengthen the ability to prevent the replay attacks.

The property $Timeout\ Freedom$ which does not pass in PAT might result from the existence of the man-in-the-middle attacks. With the limits of time and energy, solutions to guard against this kind of attacks is not figured out but we have the interest to do it in the future.

5.2.2 Replay Attack

Replay attack refers to the attacker who frequently sends the same package to the destination host. Timestamps and nonces [15] have been put forward to prevent such kind of attacks.

As for the PKMv3 protocol, different nonces and digital signatures are attached from $Msg2$ to $Msg7$ which guarantee the freshness of messages. In our model, once the receiver finds that the new message has the same nonce as the received messages, the receiver will discard it and wait for the proper message. As for $Msg1$, it is the certificate of the sender which only represents the initialization of a communication. There is no sense for the intruder to waste resources to replay the message like this. $Msg8$, $Msg9$ and $Msg10$, through testifying the attached CMAC values, the receiver can also know whether the message is replayed or not. Having analyzed the safety property, either of the communication entities will receive two same messages within a certain time in succession. Even so, it cannot be summed up that PKMv3 protocol is invulnerable to the replay attack, it can be claimed that this protocol could resist to any attempt of the duplicated messages.

After we use the public key of the sender to encrypt the certificate, the certificate information will not be obtained by any intruders. And with the improvement secrecy's encryption (also encrypted by the CMAC keys), the Secrecy Violation property passed, which means no secret information, like the keys or nonce are obtained by the intruders.

Chapter 6

Formal Modeling of the 5G AKA Protocol

In this chapter, we will make the formal model of the 5G AKA protocol in CSP and implement it in PAT, which is based on the communication mechanism presented in Chapter 2. Firstly, some preliminaries, including the relevant notations, sets, functions, channels and variables, are introduced. Then we will give the description of the model of each entity, i.e., the UE, SN, HN and APRF respectively, as well as the whole system.

6.1 Preliminaries

6.1.1 Notations

The transmitted messages are defined as follows:

$M1 = \{SUCI\}$; $M2 = \{SUCI, SNID\}$;

$M3 = \{SUCI, SNID\}$; $M4 = \{AUTN, K, XRES*, K_{AUSF}\}$;

$M5 = \{R, AUTN, HXRES*, K_{SEAF}\}$; $M6 = \{R, AUTN\}$;

$M7 = \{RES*\}$; $M8 = \{RES*, SUCI\}$

$M9 = \{SUPI, K_{SEAF}\}$; $M10 = \{Auth - Success\}$;

$M11 = \{MAC - Failure\}$; $M12 = \{Sync - Failure, AUTS\}$

$M13 = \{Sync - Failure, AUTS, R, SUCI\}$;

And We define some pre-defined arrays corresponding to the sent messages $M1$ to $M13$.

$var\ m1[1], m2[2], m3[2], m4[4], m5[4], m6[2], m7[1], m8[2],$
$m9[2], m10[1], m11[1], m12[2], m13[4];$

6.1.2 Sets and Functions

As is known to us, the commnication entities are several, not only one. Therefore, to make it more accurate, we use UE, SN, HN and $APRF$ to represent the sets of UE, SN, HN, APRF respectively. Moreover, we define eight functions, including the $f1()$, $f1*()$, $f5()$, $f5*()$, $SHA256()$, $Challenge()$, $KeySeed()$ and $aenc()$ in PAT to implement the computation of authentication parameters in the model.

6.1.3 Channels

We use fou channels int the 5G AKA protocol. $ch1$, $ch2$ and $ch3$ are common channels used by the UE-SN, SN-HN and HN-APRF respectively. Also, as a concern of the intruder, the *intercept* and $fake$ channels are defined which is similar with that of the PKMv3 protocol.

6.1.4 Assumptions

Before modeling the 5G AKA protocol, we have some assumptions according to the protocol's specification.

- The shared UE's credential information, such as the K and $SUPI$, between the UE and HN should be secret initially.
- The secre key of the HN is also secret at the beginning.
- Functions $f1()$, $f1*()$, $f5()$, $f5*()$ have the integrity and confidencity properties.
- The intruders can impersonate itself as the SN so that it can get access to an authentication channel.

6.2 User Equipment

The 5G AKA protocol is to make the mutual authenticate of the UE, SN and HN, after which UE could establish a secure channel to protect the messages sent in the following. Firstly, we show the model the UE.

$$
\begin{aligned}
UE() =_{df}\ & ((ch1!U.S.M1 \rightarrow Skip) \,\square\, intercept!U.S.M1 \rightarrow Skip));\\
& ((ch1?S.U.m6 \rightarrow Skip) \,\square\, fake?S.U.m6 \rightarrow Skip));\\
\{AK = f5(K, m6[0]); & xSQN_{HN} = AK\ xor\ sk(m6[1][0]); MAC = f1(K, (SQN_{HN}, R))\}\\
& \rightarrow if(sk(m6[1][0]) == MAC\ \&\&\ SQN_{UE} < xSQN_{HN})\\
& \quad \{SQN_{UE} = xSQN_{HN}; RES* = Challenge(K, R, SNID);\\
& \quad\ K_{SEAF} = KeySeed(K, R, SQN_{HN}, SNID); M7 = RES*\}\\
& \rightarrow (ch1!U.S.M7 \rightarrow Skip) \,\square\, (intercept!U.S.M7 \rightarrow SKip));\\
& else\ if(sk(m6[1][0])! = MAC)\\
& \quad \{ch1!U.S.M11 \rightarrow Stop) \,\square\, (intercept!U.S.M11 \rightarrow Stop))\}\\
& else\ if(sk(m6[1][0]) == MAC \&\& SQN_{UE} \geq xSQN_{HN})\\
& \quad \{MACS = f1*(K, (SQN_{UE}, R)); AK* = f5*(K, R);\\
& \quad\ CONC* = SQN_{UE}\ xor\ AK*; AUTS =< CONC*, MAC* >;\}\\
& \rightarrow \{M12[1] = AUTS\}\\
& \rightarrow ((ch1!U.S.M12 \rightarrow Stop) \,\square\, (intercept!U.S.M12 \rightarrow Stop));
\end{aligned}
$$

6.3 Serving Network

The SN will firstly initiated an authentication with the UE, then it sends the Challenge message to HN to obtain relevent authentication materials. One SN receives the Auth Response from the UE, it will the ncheck whether the hashed received $RES*$ equals to the previously obtained $HXRES*$. If they are the same, the response message will be transmitted to the HN. The success of the authentication will be finished when the NS receives the important $SUPI$ and K_{SEAF} from the HN. To make it more clear, we extract the general choice of the three kind of situations (success, MAC failure and synchronization failure) into the submodel $SN1()$. In the following shows the implementation of the serving network model.

$$
\begin{aligned}
SN() =_{df}\ & ((ch1?U.S.m1 \rightarrow Skip) \square fake?U.S.m1 \rightarrow Skip));\\
& ((ch2!S.N.M2 \rightarrow Skip) \square intercept!S.N.M2 \rightarrow Skip));\\
& ((ch2?H.S.m5 \rightarrow Skip) \square fake?H.S.m5 \rightarrow Skip));\\
& ((ch1!S.U.M6 \rightarrow Skip) \square intercept!S.U.M6 \rightarrow Skip)); SN1();\\
SN1() =_{df}\ & (((ch1?U.S.m7 \rightarrow Skip) \square fake?U.S.m7 \rightarrow Skip));\\
& if(SHA256(R, m7)! = m5[2])\{Stop\}\\
& else\{((ch2!S.H.M8 \rightarrow Skip) \square (intercept!S.H.M8 \rightarrow Skip));\\
& \quad ((ch2?H.S.m9 \rightarrow Stop) \square (fake?H.S.m9 \rightarrow Stop)); \});\\
& \square ((ch1?U.S.m11 \rightarrow Stop) \square (fake?U.S.m11 \rightarrow Stop));\\
& \square (((ch1?U.S.m12 \rightarrow Skip) \square (fake?U.S.m12 \rightarrow Skip));\\
& ((ch2!S.H.M13 \rightarrow Stop) \square (intercept!S.H.M13 \rightarrow Stop)));
\end{aligned}
$$

6.4 Home Network

The HN will do the commnication with SN and APRF. After receives the challenge request from the SN, it will start a sesson with APRF to obtian and generate necessary materials. Then the challenge response message is sent to the SN. If the MAC values are the same and the sequence number of UE is less than that of HN, HN will receive the Auth Response from SN. Once checks the received message is the same as itself generated one, the useful key K_{SEAF} will be sent back and HN will also inform the success authentication to its repository, otherwise, the autentication will be terminated. If there is the sychronization failure, the HN will set its sequence number to be one bigger than the UE's. Similarly with the model of SN, HN1() sub-process is created to display the three possible situations. The whole behavior of the HN is shown as follows:

$$
\begin{aligned}
HN() =_{df}\ & ((ch2?S.H.m2 \rightarrow Skip) \ \square\ fake?S.H.m2 \rightarrow Skip));\\
& ((ch3!H.R.M3 \rightarrow Skip) \ \square\ intercept!H.R.M3 \rightarrow Skip));\\
& ((ch3?R.H.m4 \rightarrow Skip) \ \square\ fake?R.H.m4 \rightarrow Skip));\\
& \{MAC = f1(m4[1], (SQN_{HN}, R)); AK = f5(K, R);\\
& CONC = SQN_{HN} xor AK; AUTN = m4[0];\\
& XRES* = Challenge(m4[1], R, SNID);\\
& HXRES* = SHA256(R, XRES*);\\
& K_{SEAF} = KeySeed(K, R, SQN_{HN}, SNID); SQN_{HN} = SQN_{HN} + 1;\}\\
& \rightarrow ((ch2!H.S.M5 \rightarrow Skip) \ \square\ intercept!H.S.M5 \rightarrow Skip)); HN1();\\
HN1() =_{df}\ & (((ch2?S.H.m8 \rightarrow Skip) \ \square\ (fake?S.H.m8 \rightarrow Skip));\\
& if(m8[1]! = m4[2])Stop\\
& else\{((ch2!H.S.M9 \rightarrow Skip) \ \square\ (intercept!H.S.M9 \rightarrow Skip));\\
& ((ch3!H.R.M10 \rightarrow Stop) \ \square\ (intercept!H.R.M10 \rightarrow Stop)); \})\\
& \square\ ((ch3!H.R.M10 \rightarrow Stop) \ \square\ (intercept!H.R.M10 \rightarrow Stop));
\end{aligned}
$$

6.5 APRF

The APRF starts 5G-AKA by sending the authentication response to the HN, with an authentication vector consisting of an $AUTN$ token, an $XRES*$ token, and the key K_{AUSF}. Below shows the model of the APRF.

$$
\begin{aligned}
APRF() =_{df}\ & ((ch3?H.R.m3 \rightarrow Skip) \ \square\ fake?H.R.m3 \rightarrow Skip));\\
& ((ch3!R.H.M4 \rightarrow Skip) \ \square\ intercept!R.H.M4 \rightarrow Skip));\\
& (((ch3?H.R.M10 \rightarrow Stop) \ \square\ fake?H.R.M10 \rightarrow Stop))\\
& \square\ \{Stop\});
\end{aligned}
$$

6.6 Intruder

The detailed behaviors of intruders have been introduced in Chapter 3 and four in detail. In the model of 5G AKA protocol, intruders can also intercept or fake messages from the commnication entities or work as a normal UE. The models of intruders defined as $Intruder\_Intercept()$, $Intruder\_Fake()$, $Intruder\_UE()$ and the whole intruder $Intruder()$ are described in the following.

1. Interception: The intruders can intercept messages from the insecure channels of the entities. Here, we define the received message as $msg1$ and the model is:

$$\begin{aligned} Intruder\_Intercepte() =_{df} & (intercept?U.S.msg1 \rightarrow Intruder()) \\ & \Box (intercept?S.U.msg1 \rightarrow Intruder()) \\ & \Box (intercept?S.H.msg1 \rightarrow Intruder()) \\ & \Box (intercept?H.S.msg1 \rightarrow Intruder()) \\ & \Box (intercept?H.R.msg1 \rightarrow Intruder()) \\ & \Box (intercept?R.H.msg1 \rightarrow Intruder()); \end{aligned}$$

2. Fake: The intruders have the ability to fake message and then transmit to the communication entities. Here we define a message named M representing the message created by the intruder.

$$\begin{aligned} Intruder\_Fake() =_{df} & (fake!U.S.M \rightarrow Intruder()) \\ & \Box (fake!S.U.M \rightarrow Intruder()) \\ & \Box (fake!S.H.M \rightarrow Intruder()) \\ & \Box (fake!H.S.M \rightarrow Intruder()) \\ & \Box (fake!H.R.M \rightarrow Intruder()) \\ & \Box (fake!R.H.M \rightarrow Intruder()); \end{aligned}$$

3. The intruder can also work as an UE to start the authentication with the SN and HN to get the valid K_{SEAF}, keeping the following messages' transmission safe.

$$\begin{aligned} Intruder\_UE() =_{df} & ((ch1!U.S.M1 \rightarrow Skip) \Box intercept!U.S.M1 \rightarrow Skip)); \\ & ((ch1?S.U.m6 \rightarrow Skip) \Box fake?S.U.m6 \rightarrow Skip)); \\ & \ldots \\ & \rightarrow ((ch1!U.S.M12 \rightarrow Stop) \Box (intercept!U.S.M12 \rightarrow Stop)); \end{aligned}$$

4. The whole behaviors of the intruder can be modeled the choice of behaviors as follows:

$$Intruder() =_{df} Intruder\_Intercept() \Box Intruder\_Fake() \Box Intruder\_UE();$$

6.7 System

We consider the situations that intruder may exist, so all entities, including the UE, SN, HN and the APRF as well as intruders are modeled in parallel in the whole system.

$$System() =_{df} UE()\|SN()\|HN\|APRF()\|Intruder();$$

Chapter 7

Verification and Analysis of the 5G AKA Protocol

In this chapter, we will firstly abstract some properties in accordance with the 5G AKA Protocol specification. The properties are explained using LTL formulae and assertions and implemented in PAT. With the results of the verification, we will discuss and analyze whether the protocol is secure enough.

7.1 Verification

In the following, we will introduce three properties, including the Deadlock Freedom, Connectivity and Secrecy.

7.1.1 Deadlock Freedom

In the authentication and authorization procedure, the deadlock of the model should be avoided. Firstly, we will run and check whether our model is deadlock free and use the pre-defined assertion in PAT.

$$\#assert\ System()\ deadlockfree;$$

7.1.2 Connectivity

The aim of the protocol is to assure the UE be authorized by the HN. Then the UE will obtain the session key K_{SEAF}, used to guarantee the security of the following transmitted messages. Therefore, there are some authentications among the UE, SN, HN as well as the ARPF, such as the SN authorized by the HN and the authentication between the UE and SN. Here we define some variables, indicating the initiation of the session or the messages sent or received. Detailed definition and assertions are shown as follows:

$$
\begin{aligned}
&\#assert\ System() \mid = \Box((\Box !CommitSU) \\
&\qquad\qquad \| (!CommitSU\ U\ RunningUS)); \\
&\#assert\ System() \mid = \Box((\Box !CommitHS) \\
&\qquad\qquad \| (!CommitHS\ U\ RunningSH)); \\
&\#assert\ System() \mid = \Box((\Box !CommitRH) \\
&\qquad\qquad \| (!CommitRH\ U\ RunningHR)); \\
&\#assert\ System() \mid = \Box((\Box !CommitSI) \\
&\qquad\qquad \| (!CommitSI\ U\ RunningIS));
\end{aligned}
$$

The relevant symbols in the assertions have introduced in Chapter 5. Here, the first assertion means that once the UE has run the protocol, UE and SN establish the agreement. Similarly, the HN will not commit a session until the SN initiates with it. In this way, we can check the connection or the authentication of each entity.

7.1.3 Secrecy

In the 5G AKA protocol, secrecy of both the session keys (i.e., K_{SEAF} and K_{AUSF}) and the long-term shared secret key K are taken into consideration. The variables like $k1$ and $k2$ are pre-defined as $true$ in the model. If the secrecy leaks, the value will become $false$. Here we consider two keys, the K_{SEAF} and the long-term shared secret key K. In the following, we define the property $secrecy1$ and $secrecy2$. The assertion of the whole system indicates that the secret violation is not satisfied and the secrcy information is safe.

$$
\begin{aligned}
&\#define\ secrecy1\ (k1 == true); \\
&\#define\ secrecy2\ (k2 == true); \\
&\#assert\ System \mid = \Box(secrecy1); \\
&\#assert\ System \mid = \Box(secrecy2);
\end{aligned}
$$

7.1.4 Verification Results

The results of the verification of the 5G AKA protocol is shown in Fig.7.1. We can see that, only the Deadlock Freedom and connection properties passed. However, one secrecy is violated, which is the K_{SEAF}. Therefore, there are entities not authenticated with each other or by others according to the authenticaton specification, which means there will be intruders hinder the procedure so that the authenticated key is obtained by the intruder and poses a threat on the message transmission between the UE and SN.

7.2 Analysis of the 5G AKA Protocol

The results of the secrecy property shows that the mutual authentication of some entities are not valid. And the unsatisfied secrecy property indicates the key like K_{SEAF} is leaked during the authentication. In the following, we will discuss and analysis the vulnerbilities of the protocol and put forward some improvements.

Firstly, we will consider where the intruders occurred and what he did. Suppose the intruder intercepts a message including an $SUCI$ from an authenticated UE, and it can also impersonate itself as a valid UE, so it has its own identity $SUCI\_I$. Next, an authentication request is initiated by the intruder with the SN, sending the intercepted messages. After that, the SN start the challenge request session with the same HN. Nearly the same time, the intruder sends its $SUCI\_I$ to initiate another session with the SN. If the ARPF sends the authenticaton materials at the same time or very close, the HN might mix the two sessions. If the HN gives the valid UE's K_SEAF to the intruder, the SN and HN will consider the valid identifier of the K_SEAF is $SUCI\_I$, having the bad influence on the previous UE. This problem occurs because the HN could not identify the sending message continuously. Therefore, to conquer it, a good method is to add identifiers, like the random nonce. In this way, the message 3 is changed to be $SUCI, SNID, Nh$ and this nonce is also attached with the message 4. Then after checks the nonce, the HN will delievery the challenge response message unique.

Next we will analyze the mutual authentication of the entities in the protocol.

1. SN and HN: From the Fig. 2.3, the challenge request message contains the SN's ID and the identifier of the UE, however, there is no information of the identity of the HN shown in the challenge response message. Therefore, the new random nonce can be added by the SN, attached in the challenge request. After the SN receiveds the response message, it will check the nonce value so as to do the mutual authentication of the SN and HN.

2. UE and SN: Here, after the UE receives the 7-th message, it will consider the authentication finished and it has obtained the useful encryption key K_{SEAF}. However, the K_{SEAF} can be derived by the SN, HN and teh APRS and the initial $SNID$ information in the UE is not the obtained in the authenticated discovery phase. Therefore, we add another message attached with the obtained K_{SEAF} from the HN, transmitted from the SN to UE. It is also the message of authentication success. Then the UE can apply K_{SEAF} to do the following communications.

After adding the fresh nonce to the messages and makint the mutual authentication of the entities of the protocol, the verification results are shown in Fig.7.2. We can see that the secrecy property of the K_{SEAF} now pass the verification.

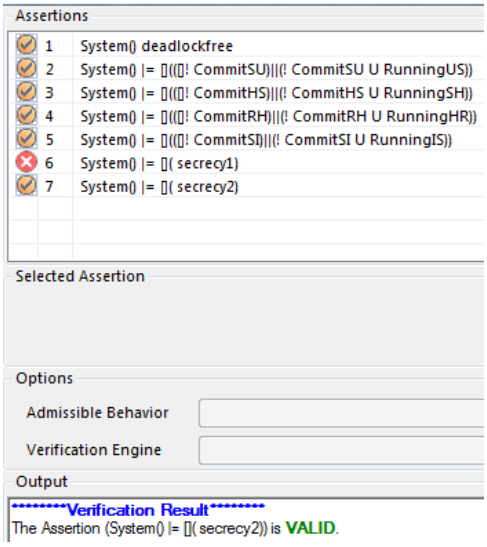

FIGURE 7.1: Verification Results

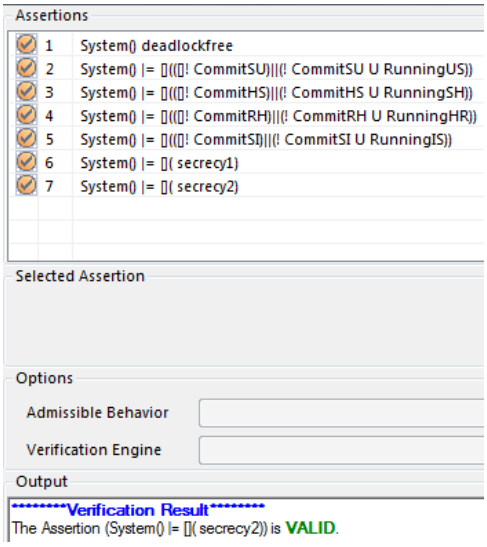

FIGURE 7.2: Improved Verification Results

Chapter 8

Conclusion and Future Work

In this paper, we studied the privacy key management protocol of WiMAX, one of the standards of the 4G network. We built a CSP model of the PKMv3 protocol and implemented in the model checker PAT. The communication entities such as the server, base station and the mobile station, are described as processes respectively. Moreover, the intruder, who could intercept or fake messages through the communication channels was introduced and designed. We simulated the models using PAT to check whether it conformed to the protocol's specification. Six properties written in LTL formulae and assertions were verified automatically. The result demonstrated that there might be some attacks such as the existence of man-in-the-middle attacks, and some attacks such as the replay attack might not be able to happen in the PKMv3 protocol. We analyzed one security leak which would happen during the certificate request/reply procedures between BS/MS and the server and the corresponding solution was put forward. Morevoer, we also proposed approaches to solve the vulnerabilities during the session of MS and BS.

Also, the newly released 5G authentication and key agreement protocol was introduced and implemented using CSP. With the results of the verification, we discovered the secrecy property is not all valid, which resulted from the interception of the intruders and the unsatisfied authentication properties. At last, we put forward relevant solutions to enhance the security of the protocol.

In the future, we would refine our model and inspect whether there are other potential vulnerabilities. The Mesh is another scheme which is widely used in reality, so it

is also meaningful to check the security of the protocol in this mode, where MS can communicate with BS and MS as well. Therefore, modeling and analysing communications among several mobile stations and one base station would be our another interest. Moreover, there is another authentication protocol named EAP-AKA', also in charge of the authentication and key management. It is of great interest to verify that protocol to explore the similarities and differences to the 5G AKA protocol.

Bibliography

[1] Cas Cremers and Martin Dehnel-Wild. Component-based formal analysis of 5g-aka: Channel assumptions and session confusion. 2019.

[2] Mohammed Jaloun and Zouhair Guennoun. Wireless mobile evolution to 4g network. *Wireless sensor network*, 2(04):309, 2010.

[3] Jeffrey G Andrews, Arunabha Ghosh, and Rias Muhamed. *Fundamentals of WiMAX: understanding broadband wireless networking*. Pearson Education, 2007.

[4] Standard for local and metropolitan area networks-Part 16: Air interface for broadband wireless access systems-Amendment 3: Advanced air interface, IEEE Std 802.16m. 2011.

[5] Noudjoud Kahya, Nacira Ghoualmi, and Pascal Lafourcade. Key management protocol in wimax revisited. In *Advances in Computer Science, Engineering & Applications*, pages 853–862. Springer, 2012.

[6] Fuden Tshering and Anjali Sardana. A Review of Privacy and Key Management Protocol in IEEE 802.16e. *International Journal of Computer Applications*, 20(20): 25–31, 2011.

[7] Tao Han, Ning Zhang, Kaiming Liu, Bihua Tang, and Yuanan Liu. Analysis of mobile WiMAX security: Vulnerabilities and solutions. In *IEEE 5th International Conference on Mobile Adhoc and Sensor Systems, MASS 2008, 29 September - 2 October 2008, Atlanta, Georgia, USA*, pages 828–833, 2008.

[8] Christoph L Schuba, Ivan V Krsul, Markus G Kuhn, Eugene H Spafford, Aurobindo Sundaram, and Diego Zamboni. Analysis of a denial of service attack on tcp. In *Proceedings. 1997 IEEE Symposium on Security and Privacy (Cat. No. 97CB36097)*, pages 208–223. IEEE, 1997.

[9] https://www.3gpp.org/.

[10] Security architecture and procedures for 5g system (3gpp ts 33.501 version 15.1.0 release 15). 2018.

[11] Muxiang Zhang and Yuguang Fang. Security analysis and enhancements of 3gpp authentication and key agreement protocol. *IEEE Transactions on wireless communications*, 4(2):734–742, 2005.

[12] Jari Arkko and Henry Haverinen. Eap aka authentication. *Draft-Arkko-Pppext-Eap-Aka-11, IETF*, 2003.

[13] Ahmed M.Taha, Amr T.Abdel-Hamid, and Sofiene Tahar. Formal Verification of IEEE 802.16 Security Sublayer Using Scyther Tool. pages 1 – 5, 2009.

[14] Sen Xu, Chin-Tser Huang, Matthews, and M.M. Modeling and analysis of IEEE 802.16 PKM Protocols using CasperFDR. In *Wireless Communication Systems. 2008. ISWCS '08. IEEE International Symposium on*, pages 653 – 657, 2008.

[15] Jun Kurihara Toshiaki Tanaka Andreas Deininger, Shinsaku Kiyomoto. Security Vulnerabilities and Solutions in Mobile WiMAX. *International Journal of Computer Science & Network Security*, (11):7–15, 2007.

[16] Sen Xu, Manton Matthews, and Chin Tser Huang. Security Issues in Privacy and Key Management Protocols of IEEE 802.16. In *Southeast Regional Conference, 2006, Melbourne, Florida, Usa, March*, pages 113–118, 2006.

[17] Fan Yang. Comparative Analysis on TEK Exchange between PKMv1 and PKMV2 for WiMAX. In *Wireless Communications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference on*, pages 1 – 4, 2011.

[18] K. V. Krishnam Raju, V. Valli Kumari, N. Sandeep Varma, and K. V. S. V. N. Raju. Formal Verification of IEEE802.16m PKMv3 Protocol Using CasperFDR. In *Information and Communication Technologies - International Conference, ICT 2010, Kochi, Kerala, India, September 7-9, 2010. Proceedings*, pages 590–595, 2010.

[19] Xiaoran Zhu, Yuanmin Xu, Jian Guo, Xi Wu, Huibiao Zhu, and Weikai Miao. Formal Verification of PKMv3 Protocol Using DT-Spin. In *International Symposium on Theoretical Aspects of Software Engineering*, pages 71–78, 2015.

[20] Xiaoran Zhu, Yuanmin Xu, Xin Li, Jian Guo, Huibiao Zhu, and Phan Cong Vinh. Formal analysis of the pkmv3 protocol. *Mobile Networks and Applications*, 23(1): 44–56, 2018.

[21] Jinpeng Jiang, Hongyan Mao, Rumeng Shao, and Yuanmin Xu. Formal verification and improvement of the pkmv3 protocol using csp. In *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, volume 2, pages 682–687. IEEE, 2018.

[22] Yuanmin Xu, Huibiao Zhu, Xiaoran Zhu, Xi Wu, Jian Guo, and Gang Lu. Formalization and verification of the pkmv3 protocol using csp. In *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, volume 1, pages 499–504. IEEE, 2017.

[23] Ravishankar Borgaonkar, Lucca Hirschi, Shinjo Park, and Altaf Shaik. New privacy threat on 3g, 4g, and upcoming 5g aka protocols.

[24] Mahdi Aiash, Glenford Mapp, Aboubaker Lasebae, Raphael Phan, and Jonathan Loo. A formally verified aka protocol for vertical handover in heterogeneous environments using casper/fdr. *EURASIP Journal on Wireless Communications and Networking*, 2012(1):57, 2012.

[25] KookHeui Lee and SangJae Moon. Aka protocols for mobile communications. In *Australasian Conference on Information Security and Privacy*, pages 400–411. Springer, 2000.

[26] David Basin, Jannik Dreier, Lucca Hirschi, Saša Radomirovic, Ralf Sasse, and Vincent Stettler. A formal analysis of 5g authentication. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 1383–1396. ACM, 2018.

[27] Adrien Koutsos. The 5g-aka authentication protocol privacy. *arXiv preprint arXiv:1811.06922*, 2018.

[28] David Cooper, Stefan Santesson, Stephen Farrell, Sharon Boeyen, Russell Housley, and William Polk. Internet x. 509 public key infrastructure certificate and certificate revocation list (crl) profile. Technical report, 2008.

[29] C A R Hoare. *Communicating Sequential Processes*. Prentice/Hall International,.

[30] Stephen D. Brookes, C. A. R. Hoare, and A. W. Roscoe. A Theory of Communicating Sequential Processes. *J. ACM*, 31(3):560–599, 1984.

[31] Process Analysis Toolkit. https://sav.sutd.edu.sg/PAT/?page_id=2660.

[32] Jun Sun, Yang Liu, and Jin Song Dong. *Model Checking CSP Revisited: Introducing a Process Analysis Toolkit*. Springer Berlin Heidelberg, 2008.

[33] Patrice Godefroid and Pierre Wolper. Using partial orders for the efficient verification of deadlock freedom and safety properties. In *International Conference on Computer Aided Verification*, pages 332–342. Springer, 1991.

[34] T. S. Supriya. Fast and secure handover of intra-ASN IEEE802.16 network by proposed certificate based pre-authentication. *Proceedings of SPIE - The International Society for Optical Engineering*, 8760:876019–876019–10, 2013.

[35] Lei Fan, Jian-Hua Li, and HongWen Zhu. An enhancement of timestamp-based password authentication scheme. *Computers & Security*, 21(7):665–667, 2002.