Security Analysis of Multicast/Unicast Router Key Management Protocols

Yiqi Huang

A Thesis

in

The Department

of

Computer Science & Software Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Computer Science at

Concordia University

Montréal, Québec, Canada

December 2017

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Yiqi Huang**

Entitled: **Security Analysis of Multicast/Unicast Router Key Management Protocols**

and submitted in partial fulfillment of the requirements for the degree of

Master of Computer Science

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

_____ Chair
Dr. Volker Haarslev

_____ Examiner
Dr. Lata Narayanan

_____ Examiner
Dr. Jeremy Clark

_____ Supervisor
Dr. J. William Atwood

Approved by _____
Volker Haarslev, Graduate Program (Research) Director
Department of Computer Science & Software Engineering

_____ 2017

_____
Amir Asif, Dean
Faculty of Engineering and Computer Science

Abstract

Security Analysis of Multicast/Unicast Router Key Management Protocols

Yiqi Huang

Key Management Protocols (KMPs) are intended to manage cryptographic keys in a cryptosystem. KMPs have been standardized for Internet Protocol Security (IPsec), and these KMPs have been formally validated for their security properties. In the Internet, routing protocols have different requirements on their KMPs, which are not met by the existing IPsec KMPs, such as IKE, IKEv2, and GDOI. Protocol modeling has been used to analyze the security of the IPsec KMPs. For routing protocols, there are new KMPs proposed by the Keying and Authentication for Routing Protocols (KARP) working group of the Internet Engineering Task Force: RKMP, MRKM, and MaRK. These KMPs are designed to have better applicability for general routing protocols. However, the security of these protocols has not been validated. In this thesis, we have summarized the necessary conditions for security of routing protocols. We have analyzed the security aspects of RKMP, MRKM, and MaRK, by formally validating those protocols using the AVISPA modeling tool. This has shown that these KMPs meet the necessary security requirements.

Keywords: Security, Routing, Key management.

Acknowledgments

I would first like to thank my supervisor, who gave me unconditional support whenever I ran into trouble or had a question about my reserch. He consistently allowed this paper to be my own work, but steered me in the right direction whenever he thought I needed it.

I would also like to thank all my friends. They bring me joy in life and help me release the pressure so that I can better focus on the work.

Finally, I must express my very profound gratitude to my parents, for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Routers are used for managing the flow of data through the Internet. They use a *routing protocol* to determine the best route; this involves the exchange of messages with peer routers. The data that are flowing may contain personal information or business secrets with significant value. Although the data themselves may be encrypted to protect them, it is also important to ensure that the routers that are handling these data are legitimate. Therefore we must ensure that the peer exchanges are secure, and that the peer routers are authenticated.

We can consider the security of the routing exchanges from three aspects: keys and cryptographic procedures, key maintenance, and control of physical access. Keys and cryptographic procedures means the use of strong keys and a reliable cryptographic algorithm to generate these keys. With the rapid development of computer performance, most current cryptographic algorithms will be less secure in five years and insecure in ten years, which makes it very important to keep cryptographic algorithms up-to-date. Key maintenance includes procedures to ensure that keys remain effective, such as key hygiene, key replacement and key rollover. It is never safe to configure some keys and leave them in place for five years. Control of physical access ensures that the cryptographic parameters cannot be obtained by non-electronic means. For example, simply walking into an office and seeing the keys written on the whiteboard, or accessing the console of a router. Making sure all the keys and networking equipment are under good control of physical access is the baseline of routing security.

Each routing protocol will have one or more mechanisms to ensure the security of its message

exchanges. Different routing protocols will, in general, have different approaches. It is also possible that a specific routing protocol will have more than one possible mechanism that has been standardized. Thus, a general view of routing protocol security needs to take this diversity into account. Given a particular routing protocol, and a choice of a specific mechanism to be used between a pair or routers, two things are necessary: procedures to determine the cryptographic parameters to be used in a specific pairing, and procedures to decide when and how to update these parameters.

In today's world, installation of the cryptographic parameters is typically done manually. Two approaches can be used for the allocation of parameters. One is to install the actual keys that will be used for the secure exchange, and the other is to install a *credential*, which will be used to authenticate the peer, and then generate the parameters automatically, using a *Key Management Protocol* (KMP).

Several KMPs exist for specific types of relationships among the participants. To date, none has been standardized for the specific area of secure routing. However, recent work in the Keying and Authentication for Routing Protocols (KARP) Working Group of the Internet Engineering Task Force (IETF) has resulted in proposals for KMPs for this area (KARP Project, 2017). Validations of the security of many standardized KMPs have been done, but to our knowledge no validation of the security of the KARP proposals has been attempted.

A proposal for managing the updating of keys has been made by Prajapati (Prajapati & Atwood, 2016a). Suggestions for dealing with the diversity of mechanisms for ensuring security are included in the KARP proposals for KMPs.

In order to have a better understanding of the proposed KMPs, we have modeled their operation using a security-oriented protocol validation tool, and then designed several experiments to analyze their security, to see if these protocols meet the general requirements for routing security.

This thesis is structured as follows: Chapter 2 outlines previous work in routing protocols and their security, and presents a previously-developed ideal framework for routing protocol management. This is followed by a brief introduction of key management protocols (KMP) and some KMPs that are in use today. Finally, the experimental environment is discussed, and the tools we used to model the protocols in order to validate their security. Chapter 3 presents the specific problem that we are going to solve. Chapter 4 gives the details of several scenarios we designed to better analyze the problem. Chapter 5 demonstrates the validation of our proposed model and gives a summary analysis.

Finally, we give the conclusion in Chapter 6.

Chapter 2

Previous work

From this chapter, we will get a better understanding of the background pertaining to the security problems we are going to analyse, the relevant work that has been done, and where improvement is needed. Also we will introduce the concepts and terms we are going to use in the following sections.

2.1 Routing Protocol Security

The wide use of the Internet nowadays has created a great challenge to information security. Routing protocols are used to exchange the topological information among routers within a single network or between different networks. According to the topological information, routers are able to select the best paths to forward packets, i.e., these associated with the least cost. Without the correct routing information, the forwarding of packets could be inefficient, or cause unreachable destinations. This makes the security of routing protocols into a crucial problem that we have to deal with.

For routing security, a routing protocol needs to make sure of the authenticity of each of the peer routers and also the integrity of the packets that they exchange. In order to achieve that, most routing protocols use keys (or a shared secret) to authenticate the peers and use different kinds of security mechanisms to protect the packets from being modified.

There are many different security mechanisms used by routing protocols. Some examples are:

(1) An *authentication trailer*, appended to the routing protocol messages, i.e., as part of the routing protocol payload;

(2) Using a security service that is part of the transport subsystem, i.e., the TCP-MD5 (Heffernan, 1998) or TCP-AO (Touch, Bonica, & Mankin, 2010) extensions to TCP;

(3) Using the services of IPsec (Internet Protocol Security, 2017), which is a generic security service at the network (IP) level.

The Border Gateway Protocol (BGP) (Rekhter, Li, & Hares, 2005) relies on the transport subsystem to provide security services. Protocol Independent Multicast - Sparse Mode (Atwood, Islam, & Siami, 2010) specifies IPsec to provide its security. Open Shortest Path First (OSPF) (Gupta & Melam, 2006) specifies either an Authentication Trailer, or IPsec.

2.2 Routing Protocol Framework

Just as we discussed, routing security is not just one particular area. In order to achieve a higher level of routing security, different kinds of components should work together as a system. If we can get a bigger picture of this system, it would be clearer to know how to make it more secure.

A Four-layer security management framework for routing protocols was introduced by Prajapati and Atwood (Prajapati & Atwood, 2016b). Table 2.1 shows the basic structure of these four layers.

Layer 1 is the routing protocol layer, all routing protocols that are used in practice, exist in this layer. Routers running routing protocols work to disseminate network information and maintain the routing table. Layer 2 has different kinds of security mechanisms, which depend on what routing protocol the network is running. All the associated keys/SA configurations, and all different kinds of security parameters are managed on Layer 3. Although there are some KMPs to help manage the configuration automatically, currently, most configurations are managed manually. We will talk about all the KMPs later. Layer 4 represents the security configuration of all the routing and management information, which automatically distributes them to every individual router.

2.3 Key Management

As noted in Section 2.1, IPsec is used in a number of cases to provide the security mechanisms. For a particular instance of IPsec, the parameters of the interaction form a Security Association (SA).

Routing Protocol (Layer 1)
Security Mechanism (Layer 2)
Key Management (Layer 3)
Configuration and Distribution (Layer 4)

Table 2.1: Four-layer Routing Protocol Framework

Key/SA management not only manages the keys, it also includes various security parameters, such as the cryptographic algorithms, the lifetime of the keys and which shared materials are used to derive the key. Those parameters should not be generated manually, otherwise the network operator will have to go to each device and configure the parameters on each of them. This is very expensive, inefficient, and difficult to maintain. In order to make key management more flexible and scalable, it is a good solution to use the Four-layer secure management framework with an appropriate Key Management Protocol (KMP).

There is large amount of routing protocols that are used in different areas. However, the KMPs are relatively fewer. We will consider two classes of KMPs: those intended to work with IPsec, and those intended to satisfy the needs of routing protocols.

2.3.1 IPsec KMPs

- IKE

  IKE (either IKE or IKEv2) is the protocol used to set up an SA in the IPsec protocol suite. It is widely used to set up a shared session secret and manage the cryptographic keys for unicast communications. There are two phases for IKE to set up the security association:

  - In phase 1, two peers will exchange the key materials, then negotiate the encryption type so that they can establish a secure and authenticated channel. The shared secret key is generated using the Diffie-Hellman key exchange algorithm, and the authentication can either use a digital signature or a Message Authentication Code (MAC). More specifically, in phase one, the messages consist of request/response pairs. The first pair of

messages (IKE_SA_INIT) exchange nonces, negotiate the cryptographic algorithms, and do a Diffie-Hellman exchange. The second pair of messages (IKE_AUTH) authenticate the previous messages using either digital signature or MAC, exchange identities and certificates, and establish the first CHILD_SA. Parts of these messages are encrypted and integrity is protected with keys established through the IKE_SA_INIT exchange, so the identities are hidden from eavesdroppers and all the fields in all the messages are authenticated (Kaufman, 2005).

- In phase 2, after the secure channel has been established, the IKE peers are able to negotiate the security association for further use. During phase 2, there will be only one pair of request/response messages, and it could be initiated by either end of the IKE_SA after the phase 1 exchange has completed. The initiator will start by sending a CREATE_CHILD_SA request along with SA offer(s) in the SA payload, and a nonce in the Ni payload (optionally with KE payload for an additional Diffie-Hellman exchange to enable stronger guarantees of forward secrecy for the CHILD_SA). The responder replies with the accepted offer in a SA payload (and the Diffie-Hellman value if the initiator sent the KE payload) and the selected cryptographic suite (Kaufman, 2005). All the messages exchanged in phase 2 are protected by the key established from phase 1, therefore, the authenticity and some integrity are achieved.

Compared to the first version, IKEv2 has a number of improvements. For more information, the need and intent of an overhaul was described in RFC 4306 (Kaufman, 2005).

- GDOI

While IKE is running between two peers to establish a "pair-wise security association", the GDOI protocol is runing between a group member and a "group controller/key server" (controller) and establishes a security association among two or more group members. As with IKE, GDOI also has two phases of message exchanges. In the first phase, the GDOI members will be authenticated to the controller, and during the second phase, the valid members can send a "pull" request to the controller for the group state. A key-encrypting key (KEK) is the

most important element in group state, which can be used to encrypt keys that decrypt the application data. The controller can establish the KEK using multicast and will send ("push") unsolicited updates of the group security association to members. In fact, the phase 1 of GDOI can be any protocol that provides the following protections:

- Peer Authentication
- Confidentiality
- Message Integrity

GDOI chooses to use ISAKMP (part of IKE) to provide those properties because ISAKMP phase 1 meets those requirements perfectly. There is a GDOI document that describes how the ISAKMP Phase 1 protocols meet the requirements of a GDOI Phase 1 protocol (Baugher, Weis, Hardjono, & Harney, 2003), so we are not going to discuss it deeply.

The phase 2 exchange of GDOI can be divided into two parts of exchanges: GROUPKEY-PULL and GROUPKEY-PUSH.

GROUPKEY-PULL exchange is protected by the key established by phase 1, and allows group members to request security associations and keys. At the end of a GROUPKEY-PULL exchange, the member or members have been authorized and have a set of SAs installed that represent group policy, and they are ready to participate in the group communications.

GROUPKEY-PUSH rekeys the protocol exchange. The rekey protocol is a datagram initiated ("pushed") by the GCKS, usually delivered to group members using an IP multicast address. The rekey protocol is an ISAKMP protocol, where cryptographic policy and keying material ("Rekey SA") are included in the group policy distributed by the GCKS in the GROUPKEY-PULL exchange. At the culmination of a GROUPKEY-PUSH exchange, the key server has sent (group) policy to all authorized group members, allowing receiving group members to participate in secure group communications. If a group management method is included in

group policy, at the conclusion of the GROUPKEY-PUSH exchange, some members of the group may have been de-authorized and no longer able to participate in the secure group communications.

IKEv1 has been made obsolete by IKEv2 because IKEv2 can achieve the same secure level as IKEv1 but with a better design. Due to the fact GDOI uses the IKEv1 Phase 1 to authenticate a group member to the group controller, although it is still secure to use GDOI, in order to get the better performance, there is a way to replace IKEv1 with IKEv2 in GDOI to make it stronger and it is basically how G-IKEv2 comes up. Strictly speaking, G-IKEv2 is still a work in progress, but the technology it used is very straightforward, therefore we are not going to further discuss it, check the document (Rowles, Yeung, Tran, & Nir, 2013) for more details if necessary.

2.3.2 Routing Security KMPs

The Keying and Authentication for Routing Protocols (KARP) (KARP Project, 2017) working group of the IETF is working on ways to improve security for key management of routing protocols. The existing KMPs such as IKE and GDOI are used by IPsec, which means they are not suitable for the routing protocols that use other mechanisms (e.g., Authentication Trailer or TCP-AO). In order to make key management more general, they have developed several proposals for KMPs that manage SA for routing protocol exchange. These are still "work in progress".

- RKMP

  It is based on modifying IKEv2, which we mentioned above. It defines a mechanism for securing the routing protocols that uses the unicast pairwise communication model. It allows network devices to automatically exchange keying material-related information between the network devices (Jethanandani et al., 2013). Unlike IKE, RKMP carries different payloads in order to adapt itself to different routing protocols.

Because IKEv2 is an existing mature protocol, there is no need to design new protocol exchanges and methods. RKMP makes use of IKEv2 protocol exchanges, policy definitions, and the state machine to define a key management protocol that can support TCP-AO, BFD, and RSVP-TE.

The overview of RKMP states is shown as Figure 2.1. It uses the state machine of IKEv2. When a network device wants to communicate with other peers, they are in State 1 at first. Before sending any routing protocol packets, two peers need to perform an IKE_SA_INIT exchange, which basically carries the secure policy and keying material for generating security associations. If IKE_SA_INIT succeeds, both network devices are transferred to State 2. At this point, the two peers have not authenticated each other yet, so they perform an IKE_AUTH exchange to do the authentication and use the keying material to generate the security association for the routing protocol they intend to support.

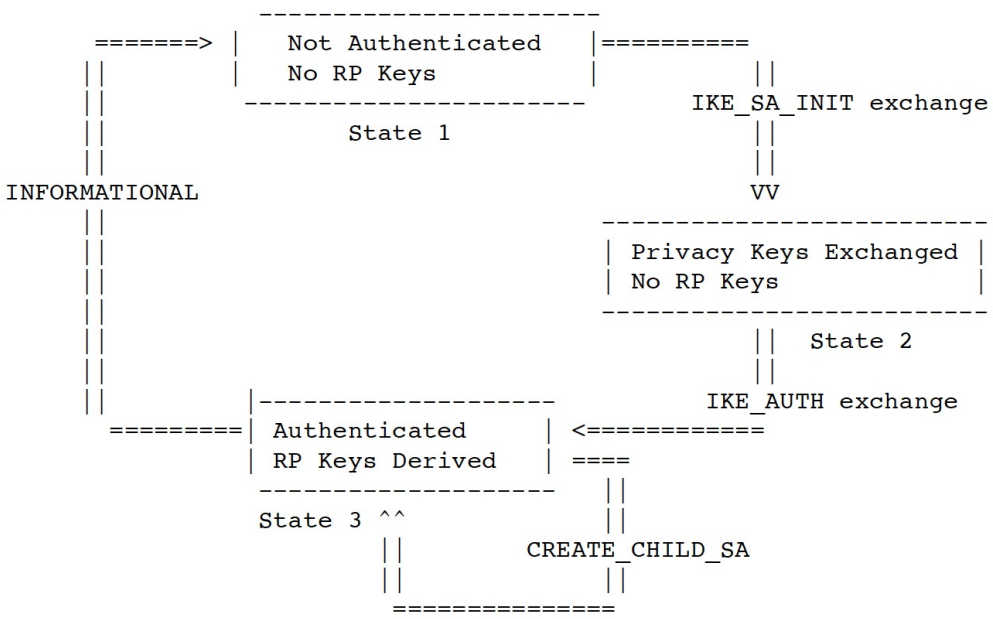

Figure 2.1: RKMP State Machine (Jethanandani et al., 2013)

Since RKMP supports TCP-AO, BFD, and RSVP-TE, it should generate specific payloads for

those three protocols.

Payload for TCP-AO: The TCP Authentication Option (TCP-AO) (Touch et al., 2010) is mainly designed for BGP and other TCP-based routing protocols. To negotiate TCP-AO policy for IKEv2, a new Security Protocol Identifier should be defined in the IANA registry for "IKEv2 Security Protocol Identifiers" Magic Numbers for ISAKMP Protocol [IKEV2-PROTOCOL-IDS](Maughan & Schneider, 1998). RKMP proposes adding a new Protocol Identifier to the table. The Protocol Name is "TCP_AO" and the value is 6.

Payload for BFD: To negotiate BFD authentication policy for IKEv2, a new Security Protocol Identifier should be defined in the IANA registry for "IKEv2 Security Protocol Identifiers" Magic Numbers for ISAKMP Protocol [IKEV2-PROTOCOL-IDS](Maughan & Schneider, 1998). RKMP proposes adding a new Protocol Identifier to the table. The Protocol Name is "BFD" and the value is 7.

Payload for RSVP-TE: To negotiate RSVP-TE authentication policy for IKEv2, a new Security Protocol Identifier should be defined in the IANA registry for "IKEv2 Security Protocol Identifiers" Magic Numbers for ISAKMP Protocol [IKEV2-PROTOCOL-IDS](Maughan & Schneider, 1998). RKMP proposes adding a new Protocol Identifier to the table. The Protocol Name is "RSVP-TE" and the value is 8.

- G-IKEv2-MRKM

  The G-IKEv2 key management protocol protects group traffic, usually in the form of IP multicast communications among the members of a set of network devices. MRKM is an extension to G-IKEv2 allowing it to protect routing protocols between a group of network devices.

  The exchange of private keying material between two network devices using a dedicated key management protocol is a common requirement. There is no need to define an entirely new protocol for routing protocols having this requirement when existing mature protocol

exchanges and methods have been vetted. MRKM extends the G-IKEv2 protocol exchanges, policy definitions, and state machine.

The G-IKEv2 GKM protocol provides for a group member (GM) receiving security associations from a GCKS in two IKEv2 exchanges: the IKE_SA_INIT exchange [RFC5996] (Kivinen, 2012) to set up the encrypted session, and the GSA_AUTH exchange (Rowles et al., 2013) (similar in construction to the IKEv2 IKE_AUTH protocol) to authenticate, authorize, and distribute group policy.

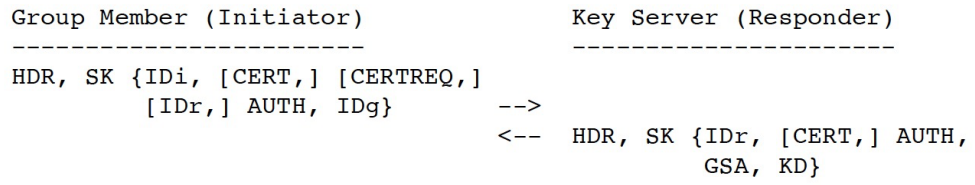

Figure 2.2: G-IKEv2 GSA_AUTH Exchange

In the GSA_AUTH exchange, the group member (GM) sends the identification of the group to which it wants to join or register. The key server (KS) authenticates and authorizes the group member and pushes the policy, traffic selector in GSA payload, and the key in the KD payload to the group member. At the successful conclusion of the GSA_AUTH exchange, the group member has the policy and the keying material to securely communicate with other group members that also registered with the key server. With this IKEv2 SA established between GM and KS, the GM can request for policy and keys of an additional group using the GSA_CLIENT_SERVER exchange. In the GSA_CLIENT_SERVER exchange, the GM will send the group ID that it wants to join, where the key server response will include the policy (GSA) and the key material (KD).

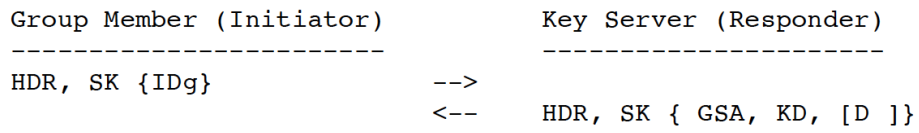

Figure 2.3: G-IKEv2 GSA_CLIENT_SERVER Exchange

Once a GSA_AUTH has completed, the group member and key server may destroy the G-IKEv2 SA. However, when the number of group members is small, as is usually the case for routing protocol participants, it is recommended for them to maintain the G-IKEv2 association SA for the key server to notify group members that they should re-register in order to obtain a new group policy. This notify exchange replaces a separate rekey mechanism optimized for large groups.

One of the GSA types is the Traffic Encryption Key (TEK) policy. The TEK describes the Traffic Encryption Policy defined by a supported security protocol. Some routing protocol definitions (i.e., OSPFv3 (Gupta & Melam, 2006), LMP (Lang et al., 2005), and PIM (Atwood et al., 2010)) describe the use of ESP and AH, which are supported by existing G-IKEv2 TEK policy definitions. However, a number of routing protocol specific security transforms exist and these require new TEK definitions.

Section 4.5 of [I-D.yeung-g-ikev2] (Hartman et al., 2012) defines the TEK payload as a Protocol-ID followed by a TEK Protocol-Specific Payload, replicated in Figure 2.4 for reference.

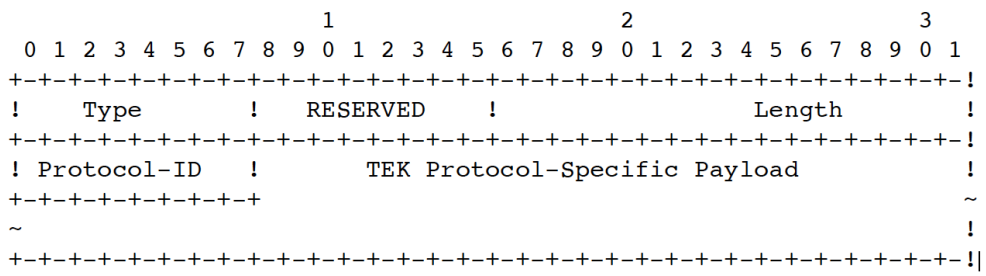

Figure 2.4: GSA TEK Payload

MRKM extends the list of Protocol-ID values to include a set of routing protocols that use group keys, shown in Table 2.2.

Protocol-ID	Value
RESERVED	0
GSA_PROTP_IPsec_ESP	1
GSA_PROTP_IPsec_AH	2
GSA_PROTP_OSPFv2	TBD1
GSA_PROTP_OSPFv3	TBD2
GSA_PROTP_IS_IS	TBD3
GSA_PROTP_LDP_HELLO	TBD4
GSA_PROTP_RSVP_TE	TBD5
GSA_PROTP_BFD	TBD6

Table 2.2: Routing Protocol-ID Values

- MaRK

  MaRK protocol is very similar to G-IKEv2, as we discussed before, GDOI establishes a security association among group members and a "group controller/key server" (short for GC/KS), and the GC/KS is assigned manually. In this case, if the current GC/KS crashes and the network administrator could not assign another new GC/KC, then the whole network will have huge problems. To solve the problem, instead of assigning a GC/KS, MaRK defines an election procedure to allow the network members to have the ability to elect the GC/KS by themselves; which gives the network more fault tolerance and self-healing ability.

  After a successful GCKS election procedure, a single router is selected to act as the GCKS for

the group. Similar with other popular announcer electing mechanisms (e.g., VRRP, HSRP), in MaRK, only GCKSes use multicast to periodically send advertisement messages. Such advertisements can be used as heart beat packets to indicate the aliveness of GCKSes. In addition, a state machine with six states (Initial, Validate, GCKS, GCKS2, Follower, and Member) shown in Figure 2.5 is specified for GCKS election. When a router is first connected to a multicast network, its state is set as Initial, then the router sends a multicast advertisement. If a GCKS is working on the network, it will reply to the router with an advertisement. After receiving the broadcasting from the GCKS, the router will try to register with the GCKS using the initial exchange. Typically this registration will succeed and the state of the router is transferred to Member. After a certain period, if the router still does not receive any advertisement from a GCKS or other group members, the router then believes there are no other group members on the network and sets its state as GCKS. If during the period the router does not receive any advertisement from a GCKS but receives advertisements from other more preferred routers on the network, the router believes that the group is involved in a GCKS election process. The router then adds these more preferred routers into its candidate list. When the time of the initial state expires, the router tries to authenticate the most preferred one in the candidate list and validates whether it can be a GCKS. If the validation result is positive, the router then transfers its state to Member and the validated one transfers its state to GCKS.

Apart from the initialization of a multicast group, the fail-over of a GCKS can also trigger an election process. For instance, if a router does not receive the heart beat advertisement for a certain period, then it will transfer its state to Initial and try to elect a new one. In a GCKS electing process, a router has to stay in the Initial state until a new GCKS is allocated. Particularly, the router first sends its initial advertisement with its priority and waits for a certain period. During the period, if a router receives an initial advertisement, which consists of a lower priority, the router then sends the advertisement again with a limited rate. After this period, if the router does not find any router with a higher priority, it announces itself as the GCKS. If two routers have the same priority, the one with the lowest IP source address used for messages on the link will be the GCKS. After a router transfers its state to GCKS,

it will reply to the initial advertisements from other routers with GCKS advertisements. It occurs even when the initial advertisements consist of higher priorities than its priority. This approach guarantees that a GCKS will not be changed frequently after it has been elected. After receiving the GCKS advertisement of the newly elected GCKS, other routers transfer their states to Member.

If a node in state member fails to perform an initial exchange with the router it believes to be GCKS, it resets its state to initial but ignores advertisements from that router. This way an attacker cannot disrupt communications indefinitely by masquerading as a GCKS.

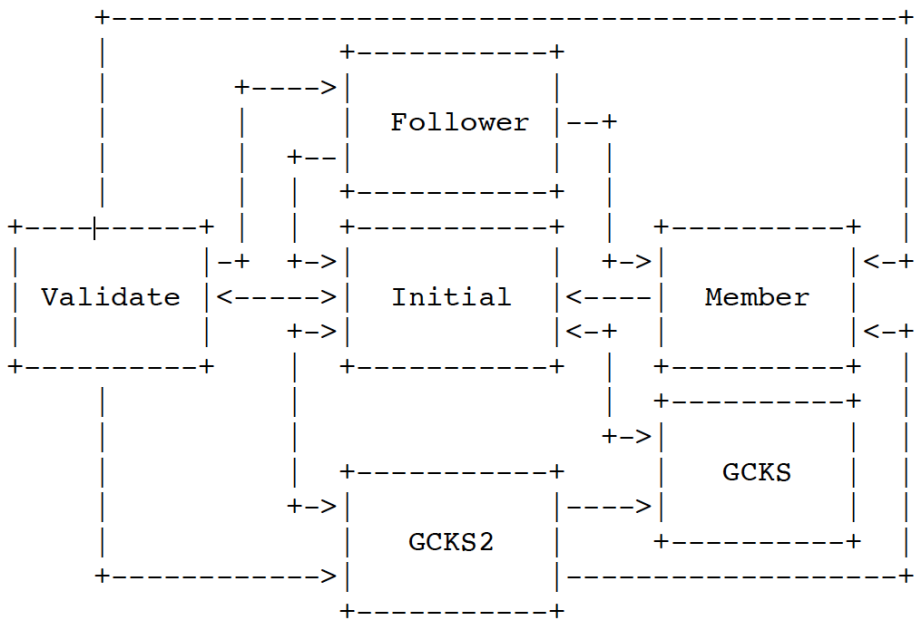

Figure 2.5: State Transitions of GCKS Election (Hartman et al., 2012)

2.3.3 KMPs Validation

Among all the KMPs we mentioned above, IKE (either version 1 or version 2) is just a protocol in the IPsec protocol suite, which has a great limit when we use it to do the key management, and

it has already been modelled and analyzed on AVISPA website (AVISPA Project, 2017). Plenty of work has been done to test its security. Therefore, we are not going to further analyze it. GDOI uses IKE and other technologies to achieve group key management. Similar to IKE, related work of GDOI has been done including the modelling and analysis (Islam & Atwood, 2010), so we will keep it out of our scope as well.

On the other hand, comparing to IKE and GDOI, the KMPs proposed by KARP working group (RKMP, MRKM and MaRK) have some significant differences. As we discussed in the earlier section (2.3.1 and 2.3.2), RKMP and IKE are both for key management in unicast communication. RKMP carries more payloads to adapt different network environments while IKE can only be deployed over IPsec. For GDOI and MaRK, in addition to the payloads, MaRK also adds the election mechanism which allows the GC/KS to be self-elected by group members within the same group. This election mechanism MaRK added provides the self-healing ability to the system when certain accidents happen (e.g., GC/KS crash, group member be compromised) and increase the tolerance of errors.

All the detailed design and explanations of RKMP (Jethanandani et al., 2013), MRKM (Tran & Weis, 2012) and MaRK (Hartman et al., 2012) could be found in related KARP documents. However, there is no any work that has been done to analyze their security. This work is achieved in this thesis.

2.4 Protocol Modelling using AVISPA

2.4.1 AVISPA

AVISPA stands for 'Automated Validation of Internet Security Protocols and Applications' (Armando et al., 2005). AVISPA provides a push-button tool as well as an industrial-strength technology for the analysis of large-scale Internet security-sensitive protocols and applications. AVISPA provides a language called the High Level Protocol Specification Language (HLPSL) for describing security protocols and specifying their intended security properties. HLPSL allows users to specify different environments and roles played in the protocol as well as their security goals for the protocol to be validated. According to the abstracted roles and environments, the AVISPA tool then verifies whether the goals have been met or not.

The AVISPA tool incorporates four back-ends that help it to perform the function of determining

if a protocol meets the specified goals. The architecture of the tool is shown in Figure 2.6.

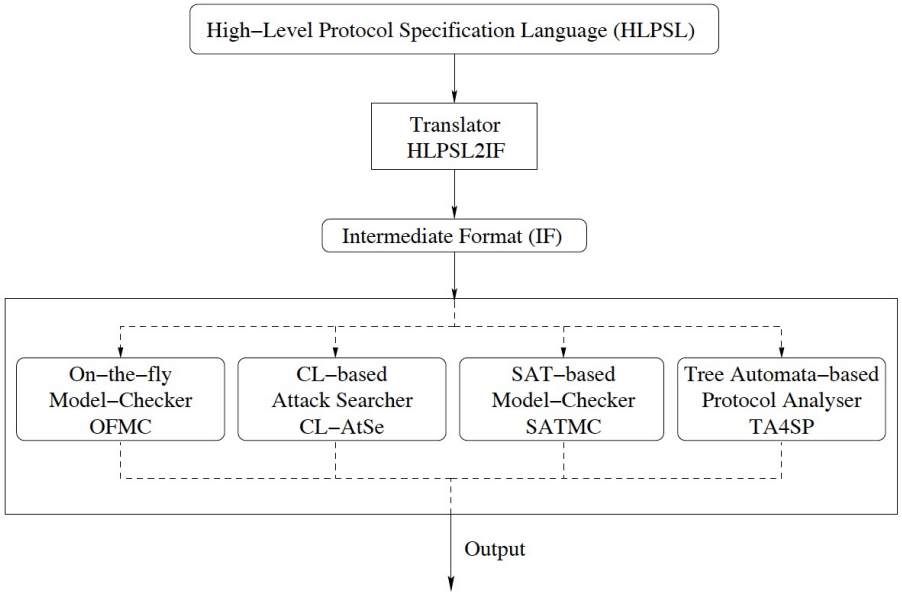

Figure 2.6: Architecture of the AVISPA Tool

From Figure 2.6, the input to the AVISPA tool is a protocol model written in HLPSL. The HLPSL specification is translated into a lower level language called Intermediate Format (IF) by a translator called hlpsl2if. This step is completely transparent to the user. The IF specification is read by any of the four back-ends of the tool as selected. These back-ends check the protocol and generate an output either confirming that the protocol is safe or showing that it is vulnerable to attacks thereby printing out an attack trace. The details of the four back-ends, namely, OFMC, CL-AtSe, SATMC and TA4SP can be found in the AVISPA user manual (Team, TA and others, 2006).

Let us now see some details of the protocol specification language provided by AVISPA, that is, HLPSL.

2.4.2 HLPSL

HLPSL stands for 'High Level Protocol Specification Language'. The HLPSL is an expressive, modular, role-based, and formal language. It is used to specify control-flow patterns, data-structures, alternative intruder models and complex security properties, as well as different cryptographic

primitives and their algebraic properties (Glouche, Genet, & Houssay, 2006). HLPSL also supports the Intermediate Format (IF), which is a lower-level language than HLPSL. A translator called hlpsl2if can translate HLPSL into IF and the IF specification of a protocol is then inputed to the back-ends of the AVISPA Tool to analyse if the security goals are satisfied or violated (AVISPA Team and others, 2006).

The three main parts of the HLPSL specification are described below:

(1) Definition of roles HLPSL is a role-based language. There are two kinds of roles. Firstly, basic roles, which are represented by agents or participants performing some actions. Secondly, composed roles, which specify how a collection of participants interact with each other.

(2) Declaration of goals This is the section where the user specifies the security goals required by the protocol being validated. Internally, these goals are represented as Linear Temporal Logic (LTL) formulae but externally useful macros are provided for the commonly used security goals, namely, authentication and secrecy.

(3) Instantiation of Roles This is similar to calling a function in a high level language such as 'C', where the appropriate type and number of arguments have to be passed into the function being called. A role can be instantiated from a higher level role by passing the correct arguments to it. Usually a composed role has a 'composition' section in which the basic roles are instantiated.

2.4.3 SPAN

SPAN is short for 'Security Protocol ANimator for AVISPA' . It comes with a Local Graphical User Interface for AVISPA. It helps in interactively producing Message Sequence Charts (MSC for short) which can be seen as an "Alice & Bob" trace from an HLPSL specification. SPAN can represent one or more sessions of the protocol in parallel with each other according to the informations given in the role environment. Then, MSCs are produced interactively with the user. SPAN also includes the possibility to check the values at every moment of the variables of each principals: the user chooses the variables of each roles he wants to monitor.

The three modes of SPAN are:

- **Protocol Simulation** for simulating the protocol and building a particular MSC corresponding to the HLPSL specification.
- **Intruder Simulation** for simulating the protocol with an active/passive intruder.
- **Attack Simulation** for automatic building of MSC attacks from the output of either OFMC or CL-ATSE tools (Glouche et al., 2006).

2.4.4 System Environment

As we are using the AVISPA tool to analyse the protocol, and the AVISPA tool can only be run under LINUX and Mac operating system, we decided to run AVISPA under Ubuntu Linux, which is a Debian-based, and very popular Linux distribution. The AVISPA tool can also be run on other Linux distributions and the method is very similar.

2.5 AVISPA Models for IKEv2 and GDOI

2.5.1 IKEv2 Model

IKEv2 has already been modelled by Sebastian Mdersheim and Paul Hankes Drielsma using AVISPA, and the code could be found in the AVISPA library (AVISPA Project, 2017).

The message exchange in the model could be explained in A-B notation as follows:

```
IKE_SA_INIT:
1. A -> B: SAa1, KEa, Na
2. B -> A: SAb1, KEb, Nb

IKE_AUTH:
3. A -> B: {A, AUTHa, SAa2}K
where K = H(Na.Nb.SAa1.g^KEa^KEb) and
AUTHa = {SAa1.g^KEa.Na.Nb}inv(Ka)
```

```
4. B -> A: {B, AUTHb, SAb2}K
where
        AUTHb = {SAb1.g^KEb.Na.Nb}inv(Kb)
```

The AVISPA goals of the model:

- secrecy of sec_a_SK, sec_b_SK
- strong authentication on sk1
- strong authentication on sk2

The authors did not list the security goals of their modelling, however, from the AVISPA goals we can speculate the model proves the following level of security.

- Confidentiality of shared key
- Peer authentication

2.5.2 GDOI Model

GDOI has also been modelled and published by Islam and Atwood (Islam & Atwood, 2010).

GDOI is composed of two sub-protocols: GROUPKEY_PULL and GROUPKEY_PUSH. The message fields that are related to security analysis of these two sub-protocols are shown in A-B notation below:

GROUPKEY_PULL

1. R - S: M.{h(SKEY.Nr.G).Nr}_Krs
2. S - R: M.{h(SKEY.Nr.Ns.SA).Ns.SA}_Krs
3. R - S: M.{h(SKEY.Nr.{Nr.Ns}_inv(Kr)).{Nr.Ns}_inv(Kr)}_Krs
4. S - R: M.{h(SKEY.Nr.Ns.Q.DEK.{Nr.Ns}_inv(Ks)).Q.DEK.{Nr.Ns}_inv(Ks)}_Krs

GROUPKEY_PUSH

1. S - R: M.{Q.SA.DEK'.M.Q.SA.DEK'_inv(Ks)}_DEK

The two communicating peers S and R represent the GCKS and the receivers of a multicast group, respectively. According to the paper, the security goals that GDOI should achieve are listed below:

(1) Replay protection

(2) Secrecy of group keys

(3) Message authentication

(4) Peer authentication

(5) Preventing Man-in-the-Middle attack

By developing GDOI models in AVISPA, the authors detected two attacks, thus the security goals were not achieved. Then the authors proposed a modified version of GDOI so that it could be attack-resistant.

Chapter 3

Problem statement

As noted in Section 1, the routing protocol is just a very small component when it comes to routing security. Much work has been done to improve the security level of routing protocols but actually, these works have very limited effects on preventing routers from being compromised.

When a router receives messages from its peer, for security purposes, the routing protocol has to ensure that the peer is legitimate, and the received messages have not been modified in transit. Nowadays, routing protocols use keys to achieve the authenticity and authority of peers, and use different kinds of security mechanisms to ensure the integrity when messages are on the way. Since key management protocols negotiate the key used in the routing protocol, the key management protocols become particularly important. If the key management protocol has been compromised, the whole routing infrastructure is fragile.

Many Key Management Protocols exist for negotiating keys and other Security Association parameters for IPsec. For instance, Internet Key Exchange (IKE, sometimes IKEv1 or IKEv2, depending on version) is the protocol used to set up a security association (SA) between two peers in the IPsec protocol suite (Internet Protocol Security, 2017). Also, Group Domain of Interpretation (GDOI) is a cryptographic protocol that takes advantage of IKE and establishes a security association among group members to do the group key management. Since IKEv2 has already been formally modeled and proved to be secure, and GDOI uses the technology of IKE to establish the security association, we have enough reasons to believe GDOI has reached the same secure level as well.

However, for the routing protocols that use other mechanisms (e.g., AT, TCP-AO), the IPsec

KMPs are clearly inappropriate to use. So the problem is that the use of IPsec KMPs have their limitations which require the routers within a network to run IPsec protocol suite. The Keying and Authentication for Routing Protocols (KARP) working group of the IETF was chartered (KARP Project, 2017) to explore ways to improve security for routing protocols. In order to solve the problem we just mentioned, they are developing KMPs in such a way that they could carry different kinds of payload so that whatever mechanisms the routing protocols are using, the KMP could provide the appropriate solution. Table 3.1 shows the work on developing KMPs for routing protocols that occurred within KARP which are not only designed for IPsec.

WORK IN PROGRESS KEY MANAGEMENT PROTOCOLS

KMP	Type	Description	Reference
RKMP	Unicast	Based on modifying IKEv2	Negotiation for Keying Pairwise Routing Protocols in IKEv2 (Jethanandani et al., 2013)
G-IKEv2	Group	Based on using IKEv2 in GDOI	Group key management using IKEv2 (Rowles et al., 2013)
G-IKEv2-MRKM	Group	Based on extending G-IKEv2	The use of G-IKEv2 for multicast router key management (Tran & Weis, 2012)
MaRK	Group	Similar with G-IKEv2 but has election procedure	Multicast Router Key Management Protocol (MaRK) (Hartman et al., 2012)

Table 3.1: Work in Progress Key Management Protocols

The problem that we are addressing is to formally model and validate these protocols since they have not been analysed so that we can make a good evaluation on how secure they are or what problems they have, then how to fix these problems.

3.1 Security Definition and Properties

As we noted in section 2.2, the Four-layer possible security management framework for routing protocols will work based on two factors:

(1) Security mechanisms in routing protocols are strong, so that the routers inside an autonomous system will not be compromised in the first place.

(2) Key management protocol is strong enough. Since key management protocol manages all the shared keying materials and the parameters of security association that established among the group members, it will be a disaster if the KMP itself is not secure enough.

For the first factor, KARP and other working groups have put many efforts to make the routing protocol security mechanisms more reliable. Also, many reports and papers have been published aiming to analyze the routing protocol security aspect, which gives us enough reasons to believe the security mechanisms in most routing protocols can be trusted.

As the second factor, we will discuss and analyse those KARP proposed KMPs in the later sections.

3.1.1 KMP security

In order to better analyse the protocols and support the conclusion, the security definition of key management protocol should be clarified and the security properties of those KMPs should be discussed as well.

Due to the different usages of a protocol, there could be many definitions of security. In order to find out a way to verify the security of the key management protocol, we need to understand what is the main purpose of a KMP. In the documentation of ISAKMP (part of IKE) (Piper, 1998), it is noted that the key management protocol should be able to manage the Security Association (including negotiating, establishing, modifying, and deleting Security Associations along with their attributes) and handle the establishment of cryptographic key for the Internet.

As the above procedures are very important, the definition of the KMP security is to ensure the above procedures are achieved securely.

The term “key management” refers to the establishment of the keys which are generated by cryptographic keying materials and cryptographic algorithms to provide protocol security services, especially integrity, authentication, and confidentiality (Bellovin & Housley, 2005). As our main targets, RKMP, MRKM and MaRK are all automated key management protocols, they derive one

or more short-term session keys by making use of long-term keys (pre-shared symmetric secret value, RSA public key, DSA public key, and others). Apart from that, they also provide several other features such as protection against replay attack, authentication of each peer, and confirmation that short-term keys are generated.

3.1.2 Type of Keys

The following symmetric keys are manipulated by the KMPs we discussed in this paper:

- PSK (Pre-Shared Key) : a PSK is a pair-wise unique key, which can be used for securing the routing protocol exchanges or be used for authenticating a network device by a KMP. These keys are configured by some mechanism such as manual configuration or a management application outside of the scope of KMP.
- Protocol master key: A protocol master key is a key exported by a KMP for use by a routing protocol. This is the key that is shared in the Key Management Data Base (KMDB) between the routing protocol and KMP. A routing protocol may use a protocol master key directly or derive traffic keys from it.
- Traffic/Transport key: A traffic/transport key is the key actually used to protect the integrity of the routing messages exchanged in a routing protocol. In existing cryptographic authentication mechanisms for routing protocols, the traffic key can be the same as or derived from the protocol master key. If there is no KMP provided, a traffic key can be the same as or derived from a pre-shared key.
- KEK (Key Encryption Key): A KEK is a key used to encrypt group key management messages to the current members of a group. A KEK is learned as the product of establishing a security association or through a group key management message encrypted in a previous KEK. A KEK has an explicit expiration but may also be retired by a message encrypted in the KEK sent by the GCKS.(Jethanandani et al., 2013)

A simple comparison between the keys described in this section is provided in the following table.

Keys	KMP vs. RP	Group vs. Pair-Wise	Usage
Pre-Shared Keys	KMP	Pair-Wise	Distributed in an out-of-band way
Key Encryption Keys	KMP	Group	For GCKS to distribute protocol master keys
Protocol Master Key	KMP or Both	Group	Used by group members to secure routng packets or generate traffic keys
Transport Key	RP usage	Group	Used by group members to secure routng packets

Table 3.2: Comparison of Keys

3.1.3 Security Properties

(1) RKMP: As we discussed in section 2, RKMP is a modified version of IKEv2, that is to say, the properties are very similar to IKE, such as automatically exchanging keying material and setting up a shared session secret from which cryptographic keys are derived. In all, the properties could be simplified as two things: first, authenticate the other peer during the exchange, and second, build a secure channel between the two peers for secure communication. Unlike IKE, which only supports devices that deploying IPsec, RKMP could be used integrated with TCP-AO, BFD or even RSVP-TE. Although the payload will certainly be different depending on the different authentication mechanism, the properties will be the same.

(2) MRKM: Since MRKM is very similar to G-IKEv2, we could say MaRK shares most of the security properties with G-IKEv2. Compared to GDOI, G-IKEv2 fixed the cryptographic weakness with authentication HASH and improved performance and network latency, as well as some reliabilities. So it should have the following properties as G-IKEv2 and GDOI:

- Authenticating the group members to a group controller.
- Establishing a security association among group members.
- Using the security association to encrypt the messages during further communications.

(3) MaRK: Since MaRK is very similar to G-IKEv2 but with election procedure in it, we could say MaRK shares most of the security properties with G-IKEv2. MaRK also defines an

election procedure for a valid group controller, which should be able to finish the election in a reasonable time and the potential new group controller should be authorized before it becomes the official group controller.

3.1.4 Performance Properties

To make those key management protocols fully functioning, only security properties are not enough, there should be certain performance properties as well, such as the algorithm election, different kinds of payload to make protocol flexible, etc.

However, the KMP performance is not relevant to its security. It is important especially when it is been deployed, but it is a little out of our scope.

3.2 Security Goals

To analyse the security of the KMPs we mentioned above, a set of security goals should be made. Since RKMP, MRKM, and MaRK are proposed by KARP working group, the general routing protocol threats document (Barbir, Murphy, & Yang, 2006) and KARP threats document (Lebovitz, Bhatia, & Weis, 2013) could be used as a starting point for establishing our goals.

Since the protocol should resist the attack of adversaries, the following security goals should be met:

- RKMP

  R-1 Peer authentication, to make sure each peer in the network is valid, mutual authentication must be performed.

  R-2 Message authentication, which includes origin authentication and integrity authentication.

  R-3 Confidentiality of RP keys. The RP keys are derived to protect the routing protocols, so they must be kept secret.

  R-4 Perfect forward security (PFS) and perfect backward security (PBS). If necessary, the network devices may destroy the state associated with the IKEv2 SA then rekey an IKEv2 SA and establish a new equivalent IKEv2 SA.

R-5 Protection against replay attacks. If an adversary replays an old message, the system must be able to ignore it.

R-6 Resistance to man-in-the-middle attacks.

R-7 Usage of strong keys.

- MRKM

When MRKM is performing exchanges to distribute keys, there may be two kinds of situation:

(1) The router that wants to join the group has no accepted credential. Then GCKS needs to perform GSA_AUTH exchange to ensure that router is valid in order to further communicate.

(2) The router that wants to join the group is authenticated. The GCKS does not need to perform the authentication procedure.

MRKM has two sets of security goals, depending on which case is true.

Case 1:

M-1 Group member authentication, to make sure all members in the network are valid, GCKS and the router must perform mutual authentication before the router joins the group.

M-2 Message authentication, which includes origin authentication and integrity authentication.

M-3 Confidentiality of RP keys. The RP keys are derived to protect the routing protocols, so they must be kept secret.

M-4 Perfect forward security (PFS) and perfect backward security (PBS). If necessary, a GCKS may need to change the group policy and/or rekey before current keys expire.

M-5 Protection against replay attacks. If an adversary replays an old message, the system must be able to ignore it.

M-6 Resistance to man-in-the-middle attacks.

M-7 Usage of strong keys.

Case 2:

M*-1 Confidentiality of RP keys. The RP keys are derived to protect the routing protocols, so they must be kept secret.

M*-2 Perfect forward security (PFS) and perfect backward security (PBS). If necessary, the network devices may destroy the state associated with the IKEv2 SA then rekey an IKEv2 SA and establish a new equivalent IKEv2 SA.

M*-3 Protection against replay attacks. If an adversary replays an old message, the system must be able to ignore it.

M*-4 Resistance to man-in-the-middle attacks.

M*-5 Usage of strong keys.

- MaRK

Ma-1 Authenticity of the GCKS. If an adversary participates into the election procedure, it should not pass the authentication.

Ma-2 Authenticity of the initiating routers. The initiating routers need to authenticate to GCKS.

Ma-3 Message authentication of the group key management messages, which includes origin authentication and integrity authentication.

Ma-4 Confidentiality of RP keys. While routing security does not typically require confidentiality, the key management protocol does because keys are exchanged and these must be protected.

Ma-5 Perfect forward security (PFS) and perfect backward security (PBS). The GCKS MUST change the protocol master key if a router was part of the group under the current protocol master key and reboots.

Ma-6 Protection against replay attacks. If an adversary replays an old message, the system must be able to ignore it.

Ma-7 Resistance to man-in-the-middle attacks.

Ma-8 Usage of strong keys.

Peer authentication and origin authentication are two different concepts. Peer authentication focuses on the authenticity of the peer member, while origin authentication tries to find out if the message comes from a valid member. Therefore, we divided them into two security goals.

The security goals listed above can be divided into two categories: those that can be formally proved (i.e., where the analysis performed by AVISPA can be guaranteed) and those that can only be improved by the practice of certain design techniques, but formal proof is impossible.

The things we can prove with AVISPA are properties that depend only on the sequence of messages, i.e., time is not relevant. The other properties are those where time has an effect. For example, the vulnerability of a key depends on the available CPU power, which increases with time. We can formally model the first category but it is not possible to use formal models for the rest.

For RKMP security goals (R-1,R-2,R-3), MRKM security goals (M-1,M-2,M-3,M*-1) and for MaRK security goals (Ma-1,Ma-2,Ma-3,Ma-4), which can be proven formally, we will give the mapping in Section 4 to show more detailed modelling, and in Section 5.2 we make a table 5.2 to explain the factors to achieve goals.

Chapter 4

Modelling of RKMP, MRKM and MaRK using AVISPA

4.1 Protocol Modeling

In order to model and analyse the RKMP and MaRK protocols, we designed several scenarios to all kinds of different situations that may occur. Each scenario contains a brief description, a list of roles, and a list of goals.

It is easier to translate a protocol into HLPSL if it is first written in Alice-Bob (A-B) notation; A-B notation is a high level modeling language that only shows the message exchanges of the protocol. If the protocol specifies the interaction between only two participants, we usually name them Alice and Bob, i.e., A-B. The A-B notation is convenient, as it gives us a clear illustration of the messages exchanged in a normal run of a given protocol. Several protocol specification languages, including an older version of HLPSL, are based on the A-B notation. In practise, however, A-B notation is not expressive enough to capture the sequence of events that need to be specified when considering large-scale Internet protocols. For instance, such protocols often call for control-flow constructs such as if-then-else branches, looping and other features. A-B notation, which shows only message exchanges, is too high level to capture such constructs that talk about the execution of actions by a single participant of a protocol run. Thats why we need a more expressive language, such as HLPSL.

HLPSL is a role-based language, meaning that we specify the actions of each kind of participant in a module, known as a basic role. Later, we could make these resulting participants interact with one another by "gluing" multiple basic roles together into a composed role.

4.1.1 RKMP

- RKMP model using HLPSL

  As mentioned before, we must have the basic roles and composed roles during the modeling in AVISPA. Since we need to use different scenarios to test the protocol to cover as many situations as possible, the composed roles, which will be session and environment, in our case should be slightly different. Therefore, the details of composed roles will be discussed in the corresponding scenario.

  RKMP is used to secure the unicast pairwise communication model. It is much simpler than multicast as the only thing that we need to secure is the authentication procedure. Once the two peers have authenticated each other, a secure tunnel is settled. Also, the GCKS can distribute the policy, routing messages and keys using the tunnel directly.

  The authentication procedure can be divided into two parts. The first part is an IKE_SA_INIT exchange, which is a two-message exchange that allows the network devices to negotiate cryptographic algorithms, exchange nonce information, and perform a Diffie-Hellman (DH) (Boyko, MacKenzie, & Patel, 2000) exchange for their routing protocols. Afterward, protocols on these network devices can communicate privately. Note that at the end of an IKE_SA_INIT exchange the endpoints on the both sides have not authenticated each other yet. For the details on this exchange, see IKE_SA_INIT in IKEv2 (Kivinen, 2012).

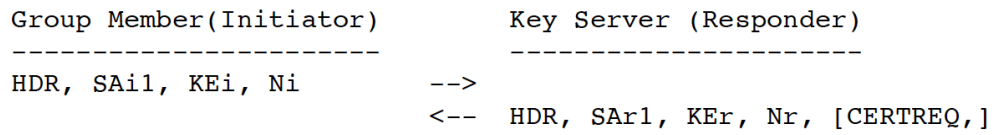

Figure 4.1: IKE_SA_INIT Exchange

Next, the network devices perform an IKE_AUTH exchange defined in IKEv2 (Kivinen, 2012). The SA payloads contain the security policies for a key and the associated parameters, and the TS payloads contains traffic selectors as defined in IKEv2 (Kivinen, 2012). For the details on the exchange, see IKE_AUTH in IKEv2 (Kivinen, 2012).

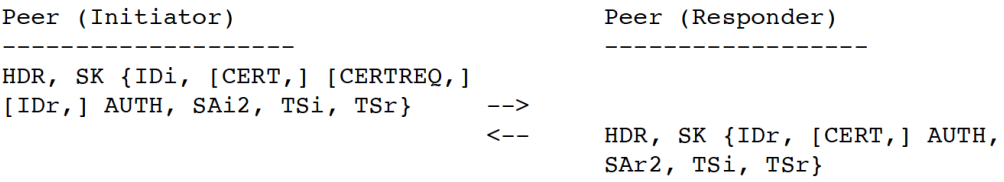

Figure 4.2: IKE_AUTH Exchange

In the IKE_AUTH exchange, the Initiator proposes one or more sets of policies for the key used to secure a routing protocol in the SAi2. The SA payload indicates that the supported policies associated with the key are being proposed.

The Responder returns the one policy contained in SAr2 that it accepts. Based on this policy, appropriate keying material is derived from the existing shared keying material. At the successful conclusion of the IKE_AUTH exchange, the initiator and responder have agreed upon a single set of policy and keying material for a particular routing protocol.

If we describe IKE_SA_INIT and IKE_AUTH exchange in A-B notation, it is easier to understand the message flow.

```
IKE_SA_INIT:
A -> B: SAa1, KEa, Na
B -> A: SAb1, KEb, Nb

IKE_AUTH:
A -> B: {A, AUTHa, SAa2}K
where K = H(Na.Nb.SAa1.g^KEa^KEb) and
AUTHa = {SAa1.g^KEa.Na.Nb}inv(Ka)
B -> A: {B, AUTHb, SAb2}K
where
       AUTHb = {SAb1.g^KEb.Na.Nb}inv(Kb)
```

Parameters. In addition to passing the role name, parameters model the keys, the functions, and some payloads in the above two exchanges. They mainly include:

(1) F. It models hash function to generate AUTH payload.

(2) Ka, Kb. They model the public key used to authenticate and encrypt messages.

Variables. Variables model most of the payloads in the above two exchanges that have fresh values generated at runtime. In our model, their values are generated using the new() operation in HLSPL. These variables mainly include:

(1) SA1, SA2. They contain each peer's preference for establishing SA, in our model, we abstracted away from the negotiation of cryptographic parameters.

(2) Ni, Nr. They model the nonces.

(3) KEi, KEr. They model the Diffie-Hellman half key used to negotiate the session key.

(4) DHX, DHY. They are the Diffie-Hellman variables.

Although the exchange is almost the same as IKE, we still design the following scenario according to the IKEv2 model for learning purposes in order to analyse the authentication exchange, and all the details we will cover in the next section.

- Scenario 1
  - Description

    This scenario describes the most basic situation about authentication in the unicast case for the RKMP protocol.

  - Roles

    In this scenario we simulated two roles, Alice and Bob. They are trying to communicate to each other and before they started the conversation, it is necessary to authenticate each other and generate a session key for later use. During this procedure, which party starts is not important, so we assume Alice to be the initiator and Bob the responder.

  - Goals

    Goal description:

    (1) Secrecy of session key

    (2) Authentication on each party

    HLPSL Code:

```
%secrecy_of SK
secrecy_of sec_a_SK, sec_b_SK

%Alice authenticates Bob on sk1
authentication_on sk1
%Bob authenticates Alice on sk2
```

```
authentication_on sk2
```

- Limitation

  Issues abstracted from:

  (1) The parties, Alice and Bob, should negotiate mutually acceptable cryptographic algorithms. This we abstract by modelling that Alice sends only a single offer for a crypto-suite, and Bob must accept this offer.

  (2) There are goals of IKEv2 that we do not yet consider. For instance, identity hiding.

  (3) IKEv2-DS includes provisions for the optional exchange of public-key certificates. This is not included in our model.

  (4) We do not model the exchange of traffic selectors, which are specific to the IP network model and would be meaningless in our abstract communication model.

- Environment

```
role environment()
def=
const sk1,sk2 : protocol_id,
        r,x : agent,
 kr,kx,ki : public_key,
 g,pr1,pr2,pi : text,
        f : hash_func

intruder_knowledge = {g,f,r,x,kr,kx,i,ki,pr1,pr2,pi,inv(ki)}


composition

session(r,x,kr,kx,g,pr1,pr2,f)
```

```
/\session(r,i,kr,ki,g,pr1,pi,f)
/\session(i,x,ki,kx,g,pi,pr2,f)

end role
```

- Intruder

  The intruder is modeled by the channel(s) over which the communication takes places. In our case, we use Dolev-Yao intruder model (DolevYao model, 2017) to describe the intruder's behavior.

  intruder_knowledge = {g,f,a,b,ka,kb,i,ki,inv(ki),zero,one}

g:	key material
f:	hash function
a,b,i:	parties
ka,kb,ki:	public keys of Alice, Bob and intruder
inv(ki):	private key of intruder
zero,one:	constant in Extension to provide key confirmation

- Goals Mapping

  Security goal R-1 (peer authentication) has two components, although each one acts equally in the communication (either of them can start the authentication). The authentication should be performed mutually. As such, AVISPA has two goals authentication_on sk1 and authentication_on sk2.

  Security goal R-2 (message authentication) has two components: authentication of the data origination point and verification of data integrity. Data origin is verified by encrypting the secret keys shared by the entities participating in the communication. The secrecy of these keys is ensured by the security goal R-3.

  Security goal R-3 (RP key secrecy) is validated by AVISPA goal secrecy_of sec_a_SK, sec_b_SK.

It is achieved due to the fact that strong authentication is chosen when the two peers perform the IKE_INIT exchange.

4.1.2 MaRK

- MaRK model using HLPSL

  The work of MaRK is divided into two tasks:

  (1) Modeling the election of new GCKS

  (2) Modeling the authentication procedure

  We designed two scenarios to test the election section, scenario 2 describes the election between two routers while scenario 3 starts the election among three routers. If the routers are more than three, the message exchanges will still be very similar to three routers.

  Since the modeling of authentication section is very similar to the modeling of RKMP, we shared several parameters and variables as we mentioned in RKMP modeling. Thus, we will only list the differences. They mainly include:

  (1) Pr. It models the priority of each router. The routers could elect their new GCKS based on that priority.

  (2) GK. It models the group key and group materials that GCKS will share with the router after it is authenticated.

  We describe two cases that will trigger the election procedure in scenario 2 and scenario 3. Later in scenario 4 and scenario 5, we explain how the GCKS distributes group keys.

- Scenario 2

  - Description

    This scenario describes the simplest situation that triggers an election procedure. We assume there are three routers, two members(Alice and Bob) and one GCKS. In this

scenario, GCKS stops working. That triggers the election procedure between Alice and Bob.

The election procedure could be divided into three parts:

(1) The two members announce their own priority by sending it to the other members.

(2) The one with lower priority starts the IKE_SA_INIT exchange as we discussed before.

(3) After the IKE_SA_INIT, the IKE_AUTH exchange will be performed to finish the authentication and generate the session key.

- Roles

  There are three roles: Alice, Bob and GCKS. Since the GCKS stoped working in the first place, it actually does not take part in the entire exchange. So we only have Alice and Bob as the two roles to perform the election procedure.

- Goals

  Goal description:

  (1) Secrecy of session key

  (2) Authentication on each party

  HLPSL Code:

```
%secrecy of SK between member and GCKS
secrecy_of sec_mem_SK, sec_gcks_SK
%member authenticates GCKS on sk1
authentication_on sk1
%GCKS authenticates member on sk
authentication_on sk2
```

- Limitation

(1) The point of this scenario is to trigger an election procedure. Since the election is based on each member's priority, which does not have a specific definition, we simply generate a random number for each member.

(2) After all the routers exchange their priorities, there comes the authentication procedure. Since the authentication method is not indicated in the paper (Hartman et al., 2012), we use IKEv2-DS in the first scenario to do the job. This means that all the limitations in the first scenario can apply to this scenario.

◦ Environment

```
role environment()
def=
const sk1,sk2 : protocol_id,
        r,x : agent,
    kr,kx,ki : public_key,
 g,pr1,pr2,pi : text,
           f : hash_func

intruder_knowledge = {g,f,r,x,kr,kx,i,ki,pr1,pr2,pi,inv(ki)}

composition

    session(r,x,kr,kx,g,pr1,pr2,f)
  /\session(r,i,kr,ki,g,pr1,pi,f)
  /\session(i,x,ki,kx,g,pi,pr2,f)

end role
```

◦ Intruder

The intruder is modeled by the channel(s) over which the communication takes place. In our case, we use the Dolev-Yao intruder model (DolevYao model, 2017) to describe intruders behavior.

intruder_knowledge = {g,f,r,x,kr,kx,i,ki,pr1,pr2,pi,inv(ki)}

g:	key material
f:	hash function
r,x,i:	parties
kr,kx,ki:	public keys of two routers and intruder
inv(ki):	private key of inturder
pr1,pr2,pi:	priorities of two routers and intruder

- Scenario 3
  - Description

    This scenario has three routers that communicate with each other. The steps of message exchanges are described below.

    (1) Three routers exchange their own priority.

    (2) The router with highest priority becomes the GCKS, others become members.

    (3) Members try to authenticate GCKS using their public keys.

  - Roles

    In order to make the election modelling more general, we have three routers communicating in this scenario, R, X and Y. By assigning them different priorities, the three parties will decide their sequences by themselves.

  - Goals

    Goals description:

    (1) Secrecy of session key

    (2) Authentication on each party

HLPSL Code:

```
%shared secrecy among GCKS and other two members
secrecy_of sec_memx_SK, sec_memy_SK, sec_gcks_SK
%mutual authentication on sk1 and sk2
authentication_on sk1
authentication_on sk2
```

- Limitation

  This scenario is basically an extension of the second one. Compared to the election between two routers, we extend it to three. In this case, we could push our conclusion to a more general case (for the situation with more than three routers, it is very similar to three).

  The difference between the previous scenario and this one is that every router has to compare more priorities than just one. Once the comparison is done, the rest of them follow the same steps as before, so we can assume the limitation of this scenario is the same as the second scenario.

- Environment

```
role environment()
  def=
    const sk1,sk2 : protocol_id,
          r,x,y : agent,
      kr,kx,ky,ki : public_key,
 g,pr1,pr2,pr3,pi : text,
                f : hash_func
```

```
intruder_knowledge = {g,f,r,x,y,kr,kx,ky,i,ki,pr1,pr2,
                      pr3,pi,inv(ki)}


    composition

        session(r,x,y,kr,kx,ky,g,pr1,pr2,pr3,f)
      /\session(r,i,y,kr,ki,ky,g,pr1,pi,pr3,f)
      /\session(r,x,i,kr,kx,ki,g,pr1,pr2,pi,f)
      /\session(i,x,y,ki,kx,ky,g,pi,pr2,pr3,f)

  end role
```

- Intruder

  The intruder is modeled by the channel(s) over which the communication takes place. In our case, we use Dolev-Yao intruder model (DolevYao model, 2017) to describe intruder's behavior.

  intruder_knowledge = {g,f,r,x,y,kr,kx,ky,i,ki,pr1,pr2,pr3,pi,inv(ki)}

g:	key material
f:	hash function
r,x,y,i:	parties
kr,kx,ky,ki:	public keys of three routers and intruder
inv(ki):	private key of inturder
pr1,pr2,pr3,pi:	priorities of three routers and intruder

- Goal Mapping

  Security goals Ma-1 and Ma-2 have two components, GCKS and Group Member (GM). When

the election succeeds, GCKS should start the mutual authentication. Since the roles are different, it is better to keep the goals separated. The two authentication goals reflect to AVISPA goals authentication_on sk1 and authentication_on sk2 in both scenario 2 and scenario 3.

Security goal Ma-3 (message authentication) has two components, authentication of the data origination point and verification of data integrity. Data origin is verified by encrypting the secret keys shared by the entities participating in the communication. The secrecy of these keys is ensured by the security goal Ma-4.

Security goal Ma-4 (RP key secrecy) is validated by AVISPA goal secrecy_of sec_mem_SK, sec_gcks_SK in scenario 2. AVISPA goals secrecy_of sec_memx_SK, sec_memy_SK, sec_gcks_SK are validated in scenario 3. It is achieved by the fact that strong authentication is chosen when the two peers perform the IKE_INIT exchange.

4.1.3 MRKM

- MRKM model using HLPSL

  MRKM assumes the GCKS already exists in the network either assigned by the administrator or elected through MaRK election procedure. The group keys that the GCKS managed will be distributed to two kinds of routers, authenticated routers and unauthenticated routers.

  With unauthenticated routers, MRKM first performs IKE_SA_INIT exchange, which we discussed in section 4.1.1 to establish a security tunnel, then followed G-IKEv2 GSA_AUTH exchange, which is substantially the same as the IKE_AUTH exchange defined in RFC 5996 (Kaufman, 2005) except that the SA, TSi, TSr payloads in IKE_AUTH are not used. Policy and traffic selectors are pushed from the key server to group members using new payloads GSA and KD. For the details of the rest of the exchange please refer to Section 4 of (Hartman et al., 2012).

  In the GSA_AUTH exchange, the group member sends the identification of the group to which

it wants to join or register. The key server authenticates and authorizes the group member and pushes the policy, traffic selector in GSA payload, and the key in the KD payload to the group member. At the successful conclusion of the GSA_AUTH exchange, the group member has policy and keying material to securely communicate with other group members that also registered with the key server.

With authenticated routers, MRKM would perform GSA_CLIENT_SERVER exchange. The GSA_CLIENT_SERVER exchange is very simple, a group member sends the group ID that it wants to join, the GCKS then responds with the group policy and keys. All messages they exchanged will be encrypted with the shared session key that was generated previously.

The A-B notation of both exchanges is listed below:

```
GSA_AUTH:
A -> B: SAa1, KEa, Na
B -> A: SAb1, KEb, Nb

IKE_AUTH:
A -> B: {A, AUTHa, G}K
where K = H(Na.Nb.SAa1.g^KEa^KEb) and
AUTHa = {SAa1.g^KEa.Na.Nb}inv(Ka)
B -> A: {G, AUTHb, GK}K
 where
       AUTHb = {SAb1.g^KEb.Na.Nb}inv(Kb)

GSA_CLIENT_SERVER:
A -> B: {G}SK
B -> A: {GK}SK
```

Scenario 2 and scenario 3 described two cases that will trigger the election procedure, and scenario 4 and scenario 5 show how GCKS distributes keys to two different group members (authorized and unauthorized).

- Scenario 4
  - Description

    This scenario models the exchange that enables the group member to register with the key server in order to get the policy, traffic selector, and keys used to communicate with other group members. We divided the exchange into two parts as before.

    The first part is IKE_SA_INIT, which has been used in many protocols as the first step to allow two parties to negotiate cryptographic algorithms, exchange nonces, and perform a Diffie-Hellman exchange (Boyko et al., 2000).

    The second part is G-IKEv2 GSA_AUTH exchange, which we already mentioned above.

  - Roles

    In this scenario we have one router (Alice) and GCKS, in which Alice has a credential but did not authenticate itself to the GCKS yet. GCKS knows the group materials and will distribute the keys and policies to Alice once Alice has passed the authentication.

  - Goals

    Goals description:

    (1) Secrecy of group key

    (2) Authentication on each party

    HLPSL Code:

```
%secrecy of session key and group key
secrecy_of sec_a_SK, sec_g_SK, sec_GK
%Alice authenticates GCKS on sk
authentication_on sk1
%GCKS authenticates on Alice sk2
authentication_on sk2
```

- Limitation

  (1) Just the same as limitations in the first scenario.

  (2) We do not model the exchange of GSA, which includes group policy for the later use in the real protocol and it has limited usage in our abstract communication model.

- Environment

```
role environment()
    def=
      const sk1, sk2  : protocol_id,
                a, g : agent,
            ka, kg, ki  : public_key,
                    gk  : message,
                     z  : text,
                     f : hash_func

      intruder_knowledge = {z,a,g,i,f,ka,kg,ki,inv(ki)}


      composition

          session(a,g,ka,kg,z,gk,f)
        /\session(a,g,ka,kg,z,gk,f)
```

```
        /\session(i,g,ki,kg,z,gk,f)


    end role
```

- Intruder

  The intruder is modeled by the channel(s) over which the communication takes place. In our case, we use the Dolev-Yao intruder model (DolevYao model, 2017) to describe the intruder's behavior.

  intruder_knowledge = {z,a,g,i,f,ka,kg,ki,inv(ki)}

z:	key material
f:	hash function
a,g,i:	parties
ka,kg,ki:	public keys of Alice, GCKS and intruder
inv(ki):	private key of inturder

- Scenario 5

  - Description

    Once an IKEv2 SA has been established between a group member and the GCKS, the group member could ask GCKS for the group policy and keys of an additional group using the GSA_CLIENT_SERVER exchange.

  - Roles

    Since this scenario is relatively simple, we only need Alice, which stands for a group member, and GCKS, which represents a key server in this model.

  - Goals

    Goal description:

(1) Secrecy of group key

(2) Authentication on group member

HLPSL Code:

```
%secrecy of group key
secrecy_of sec_gk
%GCKS authenticates client on skey
authentication_on skey
```

- Limitation

  We do not model the GSA (as in the previous scenario), because it will not affect the security.

- Environment

```
role environment()
    def=
      const skey, sec_gk  : protocol_id,
                      sk  : symmetric_key,
                      gk  : message

      intruder_knowledge = {a,g,i}

      composition

          session(a,g,sk,gk)
        /\session(a,i,sk,gk)
        /\session(i,g,sk,gk)
```

```
end role
```

- Intruder

  The intruder is modeled by the channel(s) over which the communication takes place. In our case, we use the Dolev-Yao intruder model (DolevYao model, 2017) to describe the intruder's behavior.

  intruder_knowledge = {a,g,i}

  a,g,i: | parties

- Goal Mapping

  The security goal M-1 (GM Authentication) has two components, the group member and GCKS. They perform the mutual authentication that was validated by AVISPA goals authentication_on sk1 and authentication_on sk2 in scenario 4.

  Security goal M-2 (message authentication) has two components, authentication of the data origination point and verification of data integrity. Data origin is verified by encrypting the secret keys shared by the entities participating in the communication. The secrecy of these keys is ensured by the security goal M-3.

  Security goal M-3 (RP key secrecy) was validated by AVISPA goals secrecy_of sec_a_SK, sec_g_SK, sec_GK in scenario 4. It is achieved by the fact that strong authentication is chosen when the two peers perform the IKE_INIT exchange.

  Security goal M*-1 (RP keys secrecy) is validated by AVISPA goals secrecy_of and sec_gk in scenario 5. It is achieved by the fact that both parties have reliable credentials to confirm each

peer's authenticity.

Chapter 5

Results and Analysis

5.1 Validation Result

AVISPA is used to validate RKMP, MaRK and MRKM based on the scenarios created in Section 4. Although AVISPA has four back-ends, only two of them are adopted in our experiments, namely OFMC and CL-AtSe. CL-AtSe is the back-end that translates a protocol specification into constraints and finds possible attacks on the protocol, while OFMC is the back-end that has the highest speed in finding attacks on a protocol. The other two back-ends, SATMC and TA4SP, are not designed to validate our security goals.

5.1.1 Scenario 1

The validation results of both OFMC and CL-AtSe are "SAFE". The scenario simulation is shown in Figure 5.1 and Figure 5.2.

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /opt/avispa-1.1/testsuite/results/IKEv2-DSx.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 21910 states
  Reachable  : 16491 states
  Translation: 0.02 seconds
  Computation: 9.94 seconds
```

Figure 5.1: CL-AtSe output for Scenario 1

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /opt/avispa-1.1/testsuite/results/IKEv2-DSx.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 14.87s
  visitedNodes: 5819 nodes
  depth: 14 plies
```

Figure 5.2: OFMC output for Scenario 1

5.1.2 Scenario 2

The validation results of both OFMC and CL-AtSe are “SAFE”. The scenario simulation is shown in Figure 5.4 and Figure 5.3.

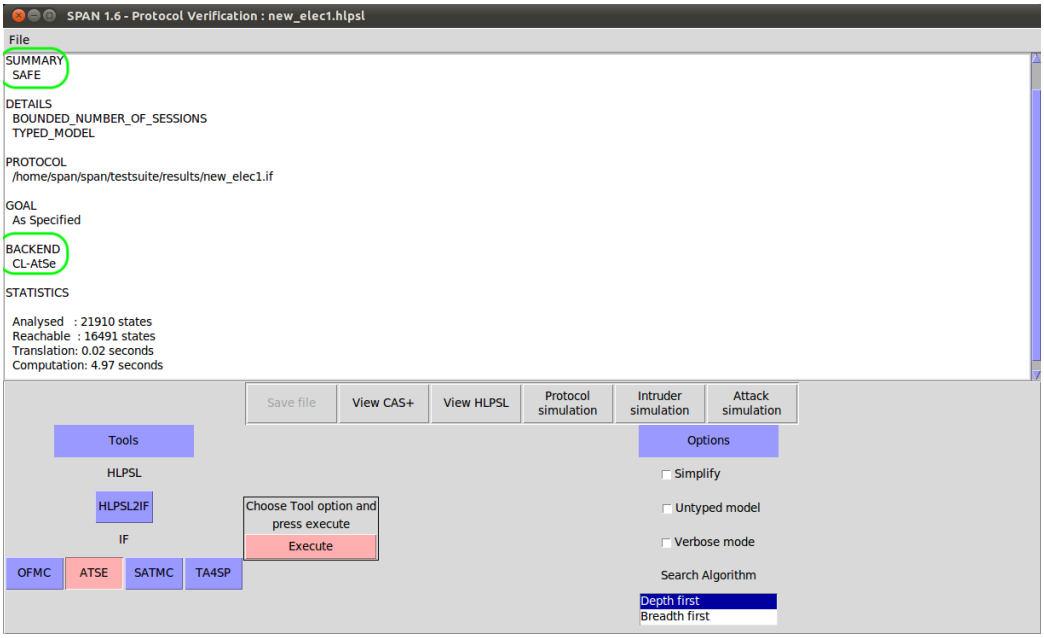

Figure 5.3: CL-AtSe output for Scenario 2

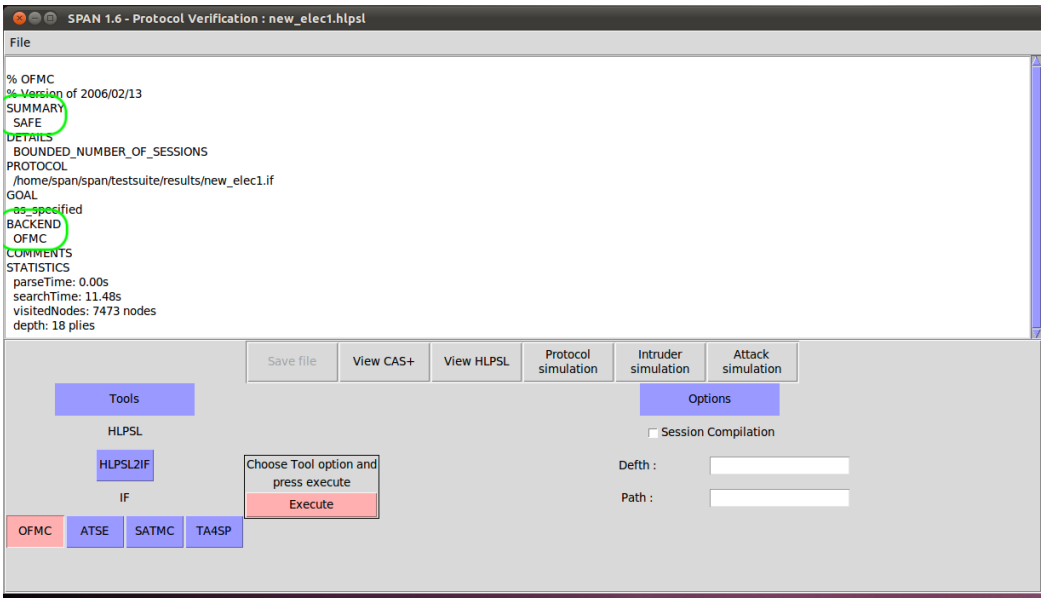

Figure 5.4: OFMC output for Scenario 2

5.1.3 Scenario 3

As shown in Figure 5.5, the validation result of CL-AtSe back-end is “SAFE”. There is no result for OFMC.

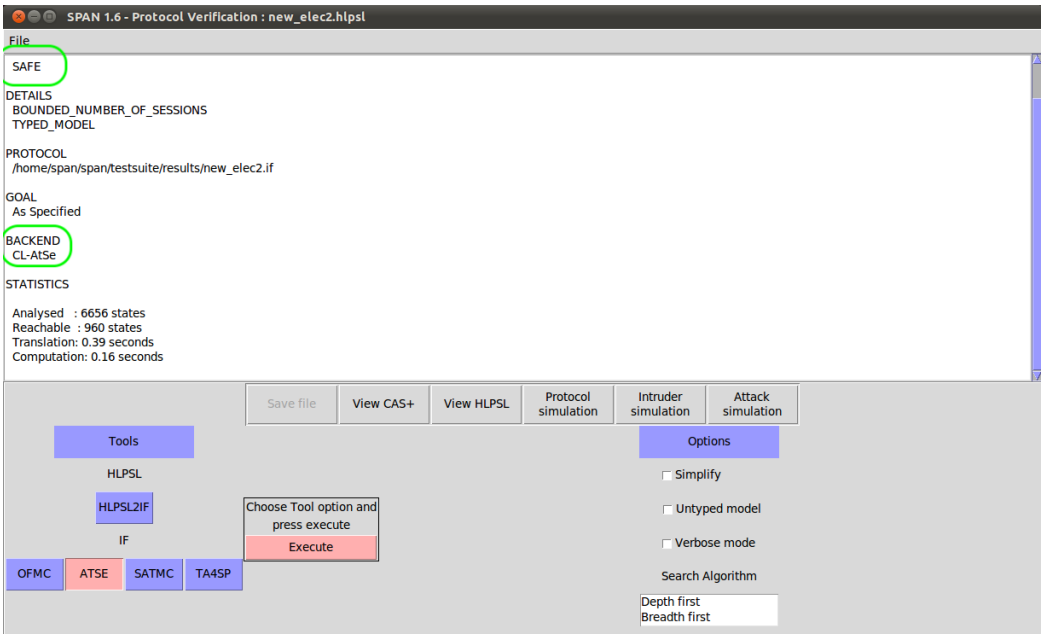

Figure 5.5: CL-AtSe output for Scenario 3

5.1.4 Scenario 4

As shown in Figure 5.6, the validation result of both OFMC and CL-AtSe are “SAFE”.

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /opt/avispa-1.1/testsuite/results/GSA_AUTH.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 647 states
  Reachable  : 430 states
  Translation: 0.00 seconds
  Computation: 0.24 seconds
```

Figure 5.6: CL-AtSe output for Scenario 4

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /opt/avispa-1.1/testsuite/results/GSA_AUTH.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 3.14s
  visitedNodes: 1455 nodes
  depth: 10 plies
```

Figure 5.7: OFMC output for Scenario 4

5.1.5 Scenario 5

The validation result of both OFMC and CL-AtSe are “SAFE”. The scenario simulation is shown in Figure 5.8 and Figure 5.9.

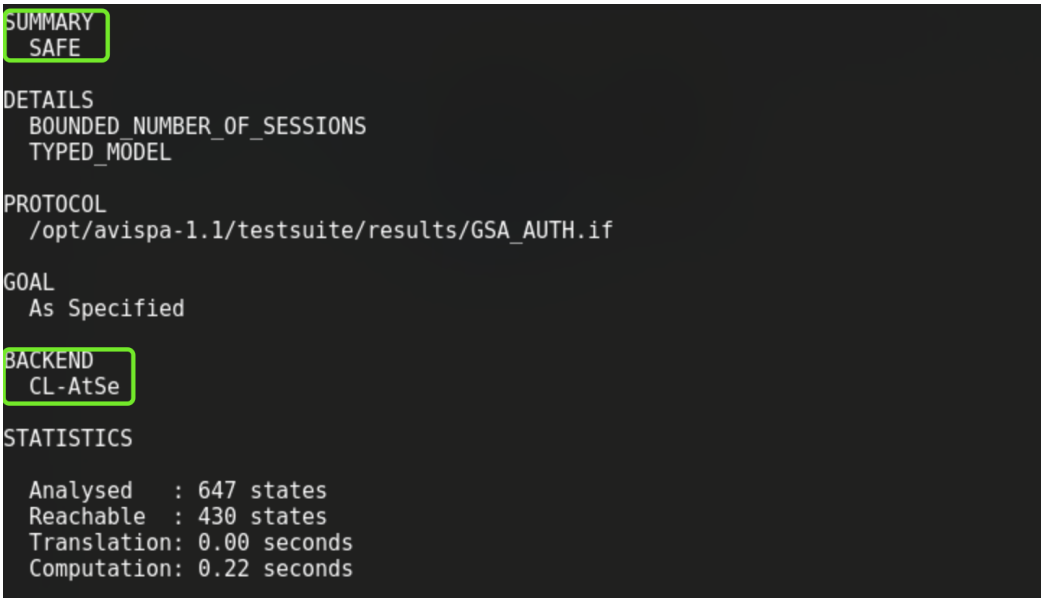

Figure 5.8: CL-AtSe output for Scenario 5

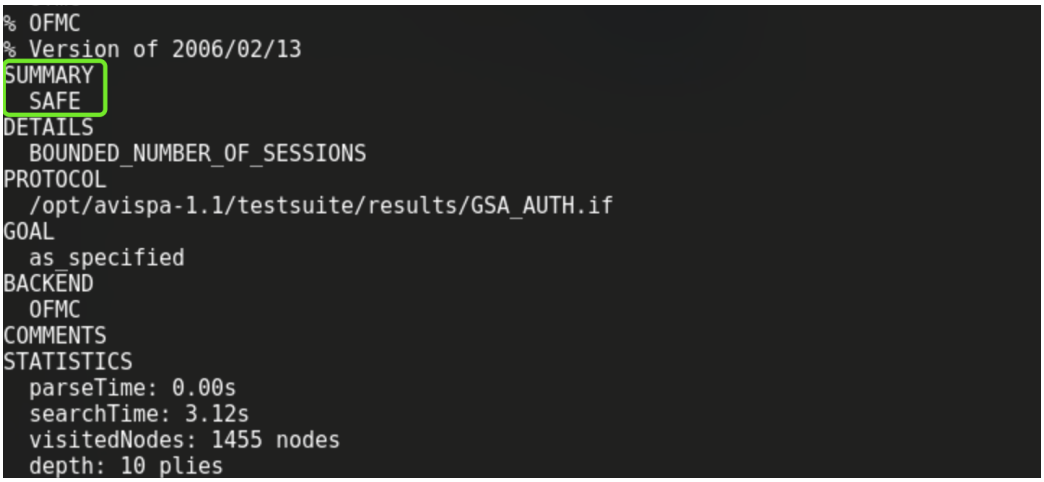

Figure 5.9: OFMC output for Scenario 5

5.2 Analysis

In all the above cases, we have seen that the OFMC and the CL-AtSe back-ends of AVISPA have reported the results as “SAFE”. This means that the key management protocols RKMP, MaRK and MRKM successfully meet the security goals, authentication and secrecy of required parameters,

which are specified in the HLPSL code. The results also show that the other goals that we have modeled are satisfied. As explained in Section 3.2, these include protection from replay attacks and message integrity.

Since RKMP is a key management protocol for peers, the distribution of keys is relatively simple, which is why we focus our analysis on the authentication procedure. For MaRK, we modeled the election procedure between two routers and among three routers, and we also modeled the key distribution of MRKM. Finally we show a summary of how RKMP, MaRK and MRKM meet all the requirements. To further explain it, the analysis is listed in Table 5.2.

Goal Requirements and Factors

Goal#	Requirement	Factors to achieve goal
R-1	Peer authentication, to make sure each peer in the network is valid, mutual authentication must be performed	Achieved through multiple factors: KE is generated by Diffie-Hellman based on peer's key material, and AUTH payload, which is encrypted using peer's private key to prevent man-in-the-middle attack
R-2	Message authentication, which includes origin authentication and integrity authentication	AUTH payload
R-3	Confidentiality of RP keys. The RP keys are derived to protect the routing protocols, so it must be kept secret	RP keys are generated by Diffie-Hellman algorithm, with AUTH payload, we can assure the confidentiality of the keys will be kept between two peers

Goal#	Requirement	Factors to achieve goal
R-4	Perfect forward security (PFS) and perfect backward security (PBS). If necessary, the network devices may destroy the state associated with the IKEv2 SA then rekey an IKEv2 SA and establish a new equivalent IKEv2 SA	In peers, if both network devices choose to retain the RP policy and keying material, the use of CREATE_CHILD_SA is required to do the rekey
M-1	Group member authentication, to make sure all members in the network are valid, GCKS must authenticate the router before it joins the group	MRKM performs GSA_AUTH exchange to authenticat both group member and GCKS, AUTHa and AUTHb payloads contain the authentication information for GCKS and group member respectively
M-2	Message authentication, which includes origin authentication and integrity authentication	AUTH payload
M-3	Confidentiality of RP keys. The RP keys are derived to protect the routing protocols, so it must be kept secret	RP keys are generated by Diffie-Hellman algorithm, with AUTH payload, we can assure the confidentiality of the keys will be kept between two peers
M-4	Perfect forward security (PFS) and perfect backward security (PBS). If necessary, a GCKS may need to change the group policy and/or rekey before current keys expire	GCKS can send an INFORMATIONAL exchange with a Notify payload directing the group member to re-register. Alternatively, GCKS can distribute a GSA_REKEY exchange if GCKS policy support the G-IKEv2 group maintenance channel

Goal#	Requirement	Factors to achieve goal
Ma-1	Authenticity of the GCKS. If an adversary participates in the election procedure, it should not pass the authentication	MaRK uses IKE to handle the peer's key management, AUTH payload in IKE_AUTH exchange could authenticate peers mutually
Ma-2	Authenticity of the initiating routers. The initiating routers need to authenticate to GCKS	AUTH payload
Ma-3	Message authentication of the group key management messages, which includes origin authentication and integrity authentication	AUTH payload
Ma-4	Confidentiality of RP keys. While routing security does not typically require confidentiality, the key management protocol does because keys are exchanged and these must be protected	When GCKS distributes group key using GSA_AUTH exchange, AUTH payload is required, which is encrypted by peer's private key. After the GCKS authenticates a group member, group member could get the group key encrypted by the session key to keep the integrity
Ma-5	Perfect forward security (PFS) and perfect backward security (PBS). The GCKS MUST change the protocol master key if a router was part of the group under the current protocol master key and reboots	In peers, if both of the network devices choose to retain the RP policy and keying material, the use of CREATE_CHILD_SA is required to do the rekeying

Goal#	Requirement	Factors to achieve goal
R-5, M-5, Ma-6	Protection against replay attacks. If an adversary replays an old message, the system must be able to ignore it	Nonce such as Na, Nb
R-6, M-6, Ma-7	Resistance to man-in-the-middle attacks	AUTH payload
R-7, M-7, Ma-8	Usage of strong keys	The key KE should be of sufficient length

Chapter 6

Conclusion

The operation of routing protocols should be secure. Unfortunately, this is not often true in real deployments. If security is enabled at all, the security keys are installed once and forgotten. In order to improve this situation, the first step is to ensure that key assignment can be done automatically. Key management protocols (KMP) exist for IPsec; these include IKE and IKEv2 (for unicast) and GDOI (for multicast). These KMP have been formally validated for their security properties.

However, while some routing protocols use IPsec to ensure their security, other routing protocols have different security approaches, such as using an authentication trailer within the routing protocol or using TCP-Authentication Option to communicate with peers.

The Keying and Authentication for Routing Protocols (KARP) working group of the IETF has proposed several KMPs for routing protocols. To our knowledge, there are no formal validations of the security of these proposals. In this thesis, we focus on three KMPs: RKMP, MRKM, and MaRK. We validate them with the AVISPA tool to ensure that they meet the necessary security requirements.

We first enumerate the desirable security properties for routing protocols, and separate them into those that can be formally validated, and those that cannot.

We then design a scenario for RKMP to model its peer authentication. Although the message exchanges of RKMP have the same structure as IKEv2, they have different payloads. This modeling process allows us to confirm the validity of our approach.

MaRK, on the other hand, defines a new election system that gives MaRK the ability to self-heal and to tolerate errors when the Group Controller and Key Server (GCKS) fails, which makes the

routing group “Autonomous”. Since this election procedure has never been analyzed, we design two scenarios to model the election procedure when it is between two routers and when it is among three routers. We assign the priorities to these routers by randomly generating the numbers and use the same technique to mutually authenticate the GCKS and the members.

After the election, there are two kinds of members to which the GCKS could distribute keys: authenticated members and unauthenticated members. One of the aspects of security is that the GCKS must not distribute keys to the unauthenticated members. Based on those two kinds of members, we designed two scenarios for MRKM to test its security. MRKM is an extension to G-IKEv2, which in turn is an updating of GDOI from IKE exchanges to IKEv2 exchanges. MRKM is able to provide authenticity, integrity and authority when it is doing the key management for multicast situations.

We then analyzed the five scenarios above using the AVISPA modelling tool. The result formally proved that the key management protocols RKMP, MaRK, and MRKM have the necessary security properties, including authentication, confidentiality, integrity, and replay protection. They meet our security requirements.

References

Armando, A., Basin, D., Boichut, Y., Chevalier, Y., Compagna, L., Cuéllar, J., ... others (2005). The avispa tool for the automated validation of internet security protocols and applications. In *International conference on computer aided verification* (pp. 281–285).

Atwood, W., Islam, S., & Siami, M. (2010). *Authentication and confidentiality in protocol independent multicast sparse mode (pim-sm) link-local messages* (Tech. Rep.).

AVISPA Project. (2017). *Avispa project — automated validation of internet security protocols and applications.* Retrieved from http://www.avispa-project.org/ ([Online; accessed 2017])

AVISPA Team and others. (2006). *Hlpsl tutorial the beginners guide to modelling and analysing internet security protocols.*

Barbir, A., Murphy, S., & Yang, Y. (2006). Generic threats to routing protocols.

Baugher, M., Weis, B., Hardjono, T., & Harney, H. (2003). *The group domain of interpretation* (Tech. Rep.).

Bellovin, S. M., & Housley, R. (2005). Guidelines for cryptographic key management. In *Symposium on research in security and privacy.*

Boyko, V., MacKenzie, P., & Patel, S. (2000). Provably secure password-authenticated key exchange using diffie-hellman. In *Advances in cryptologyeurocrypt 2000* (pp. 156–171).

DolevYao model. (2017). *Dolevyao model — Wikipedia, the free encyclopedia.* Retrieved from https://en.wikipedia.org/wiki/Dolev%E2%80%93Yao_model ([Online; accessed 2017])

Glouche, Y., Genet, T., & Houssay, E. (2006). Span–a security protocol animator for avispa–user

manual. *IRISA/Université de Rennes*, *1*, 20.

Gupta, M., & Melam, N. (2006). *Authentication/confidentiality for ospfv3* (Tech. Rep.).

Hartman, S., Lebovitz, G., & Zhang, D. (2012). Multicast router key management protocol (mark).

Heffernan, A. (1998). Protection of bgp sessions via the tcp md5 signature option.

Internet Protocol Security. (2017). *Ipsec — Wikipedia, the free encyclopedia.* Retrieved from https://en.wikipedia.org/wiki/IPsec ([Online; accessed 2017])

Islam, S., & Atwood, J. W. (2010). Sender access and data distribution control for inter-domain multicast groups. *Computer Networks*, *54*(10), 1646–1671.

Jethanandani, M., Weis, B., Patel, K., Zhang, D., Hartman, S., Chunduri, U., & Touch, J. (2013). Negotiation for keying pairwise routing protocols in ikev2. *Internet Engineering Task Force, Internet-Draft (Work in Progress)*.

KARP Project. (2017). *Charter, secure inter-domain routing (sidr) working group.* Retrieved from http://datatracker.ietf.org/wg/sidr/charter/ ([Online; accessed 2017])

Kaufman, C. (2005). Internet key exchange (ikev2) protocol.

Kivinen, T. (2012). Minimal ikev2.

Lang, J., et al. (2005). Rfc 4204: Link management protocol (lmp). *Internet Engineering Task Force (IETF)*.

Lebovitz, G., Bhatia, M., & Weis, B. (2013). *Keying and authentication for routing protocols (karp) overview, threats, and requirements* (Tech. Rep.).

Maughan, D., & Schneider, M. (1998). Internet security association and key management protocol (isakmp).

Piper, D. (1998). The internet ip security domain of interpretation for isakmp.

Prajapati, N., & Atwood, J. W. (2016a). Rpsec: Managing routing protocol security. In *Electrical and computer engineering (ccece), 2016 ieee canadian conference on* (pp. 1–6).

Prajapati, N., & Atwood, J. W. (2016b). Rpsec: Managing routing protocol security. In *Electrical and computer engineering (ccece), 2016 ieee canadian conference on* (pp. 1–6).

Rekhter, Y., Li, T., & Hares, S. (2005). *A border gateway protocol 4 (bgp-4)* (Tech. Rep.).

Rowles, S., Yeung, A., Tran, P., & Nir, Y. (2013). Group key management using ikev2. *Internet*

Engineering Task Force, Internet-Draft (Work in Progress).

Team, TA and others. (2006). Avispa v1. 1 user manual. *Information Society Technologies Programme (June 2006), http://avispa-project. org*.

Touch, J., Bonica, R. P., & Mankin, A. (2010). The tcp authentication option.

Tran, P., & Weis, B. (2012). The use of g-ikev2 for multicast router key management. *Internet Engineering Task Force, Internet-Draft (Work in Progress)*.