Computing and Informatics, Vol. 24, 2005, 149–165

DESIGN QUALITY OF SECURITY SERVICE NEGOTIATION PROTOCOL

ZhengYou Xia, Jian Wang

Departments of Computer Science
Nanjing University of Aeronautics and Astronautics
Nanjing 210016, P. R. China
e-mail: zhengyou_xia@yahoo.com

YiChuan Jiang

Departments of Computing and Information Technology
Fudan University, Shanghai 20043, China

Manuscript received 16 August 2004; revised 14 February 2005
Communicated by Ladislav Hluchý

Abstract. With future network equipment the security service becomes a critical and serious problem. Especially in the network, users do not want to expose their message to others or to be forged by others. They make extensive use of cryptography and integrity algorithms to achieve security. The sender can achieve the high quality of security service (high security level), only if the receivers and routers along path to receivers can support or satisfy the quality of security service requested by the sender. Therefore, this paper proposes a protocol to provide the needed mechanism for quality of security service, to dynamically negotiate the quality of security service among the senders and receivers of multicasts in the network. It provides different quality of security service resolutions to different receiver nodes with different security service needs and includes six different negotiation styles.

Keywords: Quality of security service, SSRSVP, security service negotiation, negotiation style, multicast security

1 INTRODUCTION

The term "quality of security service" is first presented by Cynthia Irvine [4]. The original definition is: "quality of security service refers to the use of security as a quality of service dimension and has the potential to provide administrators and users with more flexibility and potentially better service, without compromise of network and system security policies." The original definition is focused on the quality of security service from the point of view of system administrators and users. We refine and define the term "quality of security service" in relation to security service negotiation among senders and receivers in a network, i.e. we focus on the quality of security service in the network.

Definition 1 (Quality of security service)**.** refers to security service multi dimension spaces that are composed of strength of cryptographic algorithms, length of cryptographic key and Robustness of authentication mechanisms, etc., and is negotiated among the senders and receivers in the network.

Security is very important for an Internet application. The users don't want to expose their message to others or be forged by others. They make extensive use of cryptography and integrity algorithms to achieve security. Although lots of cryptography and integrity algorithms have been suggested for Internet, if the users want to use a different security configuration for their application, they need to use dynamic mechanisms to negotiate quality of security service with the receivers. The sender can achieve the high quality of security service (high security level), only if the receivers and routers along path to receivers can support or satisfy the security service level requested by the sender. At this time the networking community has yet to develop a generic mechanism to solve the negotiation process for quality of security service. The traditional session mechanism [9, 10, 11, 12] between the sender and receiver is only suited to the Point-to-Point case, because one obvious security service negotiation paradigm would have the sender transmit a negotiation request towards the receiver. However, the point-to-multipoint and multipoint-to-multipoint case [13, 14] is very difficult to solve using the session mechanism. In particular, one must not assume that all the receivers of a multicast group possess the same security capacity for processing incoming data, nor even necessarily desire or require the same quality of security service from the Internet. At the same time, the membership in a multicast group can be dynamic. To solve the above problems, we propose an extended RSVP [1, 2, 3] protocol called SSRSVP (security service RSVP) to provide the needed mechanism for quality of security service, to dynamically negotiate the quality of security service among the senders and receivers of multicast on the Internet. It provides different quality of security service resolutions to different receiver nodes in a multicast group with different security service needs. However, the SSRSVP is different to [15, 16, 17], which describe the format and use of RSVP's INTEGRITY object to provide hop-by-hop integrity and authentication of RSVP messages, and can support the IPSEC protocols [17]. SSRSVP is not to enhance or support RSVP security function but to provide different security service negotiation among the senders and receivers.

The rest of the paper is organized as follows. In the following section, we propose the SSRSVP model with its characteristics. In Section 3, SSRSVP for quality of security service is described. We present SSRSVP mechanism in Section 4. Sections 5 and 6 present the negotiation styles of SSRSVP and one complete example to illustrate the setup process of SSRSVP, respectively. Section 7 discusses other problems of SSRSVP. The Section 8 outlines a simple implementation, and the final section concludes the paper.

2 SSRSVP MODEL

The SSRSVP model similar to RSVP [1, 3] is illustrated in Figure 1. In the model, a quality of security service negotiation request of SSRSVP is passed to two decision modules – admission control and policy control, where admission control determines whether this router has sufficient available security processing capability to support the negotiation request, and policy control determines whether the user has sufficient privilege to make this request. If both decisions are positive, the corresponding parameters of SSRSVP are set. The classifier module determines the quality of security service (QOSS) level for each packet in the host or router. SSRSVP defines a session as a packet flow, and each session is treated independently by SSRSVP. A SSRSVP session is defined by its IP destination address, protocol ID, and Destination port. The security process module in the model implements the security processing per packet, including the confidentiality and integrity function, etc. Since the quality of bandwidth service is not important in SSRSVP, the packet scheduler is less important than the one in RSVP.

When a packet arrives a router without SSRSVP module, it checks route table information and forwards the packet.When a packet arrives a router with the SSRSVP module and the correct configuration, the router would do the following:

1. The classifier module determines the quality of security service (QOSS) level for each packet according to IP destination address, protocol Id and destination port.
2. The security process module extracts information from the QOSS database of SSRSVP according to the QOSS of the packet arrived. The database information includes: keys, cryptography algorithms, and quality of security service information negotiated with next hop, etc.
3. The security process module of the router will re-encapsulate the packet according to quality of security service information negotiated with next hop.
4. The router forwards the packet to next hop according to route table information.

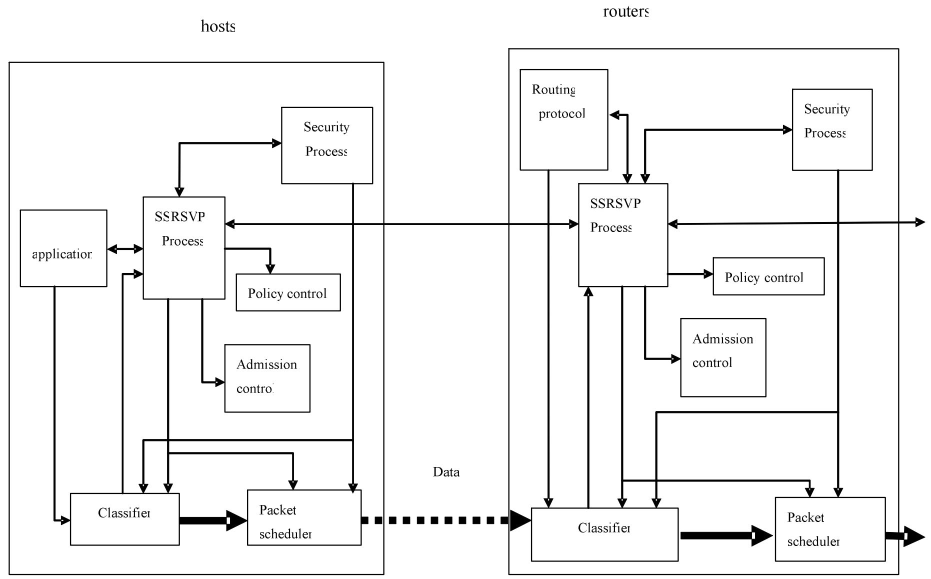

Fig. 1. SSRSVP model in hosts and routers

3 EXTENDING RSVP FOR QUALITY OF SECURITY SERVICE (SSRSVP)

A RSVP message consists of a common header, followed by a body consisting of a variable number of variable-length, typed objects.

The fields in the common header of RSVP [2] are shown in the Figure 2.

0		1	2 3
Vers	Flags	Msg Type	RSVP Checksum
Send_TTL		(Reserved)	RSVP Checksum

Fig. 2. Common header of RSVP

No flag bits are defined yet to extend RSVP for quality of security service. Here we propose to define 1001(0x09) in Flags field (4 bits) to denote the RSVP for the quality of security service. In the RSVP common header, the Msg type (8 bits) includes the Path, Resv, PathErr, ResvErr, PathTear, and ResvTear and ResvConf message types.

When the Flags are equal to 0x09, we propose that the Msg type field will become the SSRSVP message type, which is as follows:
Msg Type: 8 bits

- 1 = Spath,
- 2 = Snego,

- 3 = SpathErr,
- 4 = SnegoErr,
- 5 = SPathTear,
- 6 = SnegoTear,
- 7 = SnegoConf,
- 8 = KeyMess,
- 9 = Keyfail,
- 10 = KeySucc.

4 SSRSVP MECHANISMS

SSRSVP, similarly to RSVP [1], adopts receiver-initiated design principles: Receivers choose the quality of security service negotiation request and are responsible for initiating and keeping the negotiation active as long as they want to receive the packet. It is the receiver who knows its own limitation of the security processing capacity; furthermore, the receiver is the only one who experiences, and thus who is directly concerned with, the quality of security service experienced by the incoming data. Additionally, if network charging were deployed in the future, the receiver would likely be the party paying for the requested security service. Thus, it should be the receiver who decides what security service should be negotiated.

There are three fundamental SSRSVP message types: "Snego", "Spath" and "KeyMess" illustrated in Figure 3.

Each SSRSVP sender host transmits SSRSVP "Spath" messages downstream along the uni-/multicast routes provided by the routing protocols, following the paths of the packet. These "Spath" messages store "path state" in each router along the way. This path state includes at least the unicast IP address of the previous hop router, which is used to route the "Snego" messages hop-by-hop in the reverse direction.

Each SSRSVP receiver host sends SSRSVP negotiation request ("Snego") messages upstream towards the senders. These messages must follow exactly the reverse of the paths the packets will use on their way upstream to all the sender hosts included in the sender selection.

When the SSRSVP negotiation request is finished, the Sender host transmits SSRSVP "KeyMess" messages downstream along the "Path" to the next hop, which includes negotiation acknowledgement information and the key information of quality of security service negotiated with next hop. The process is shown in Figure 3 and described as follows:

1. The sender hosts first send the "KeyMess" message to the $router_1$, which includes negotiation acknowledgement information and the key information of quality of security service that $router_1$ uses to decrypt data packets encrypted by sender. If the processing of the "KeyMess" transmission results in failure, the

"KeyFail" message is returned to the sender, otherwise the following operations are performed.

2. The $router_1$ will send a "KeyMess" message that includes negotiation acknowledgement information and key information of quality of security service that $router_2$ uses to decrypt data packets encrypted by $router_1$. The "KeyMess" that $router_1$ sends is different to the "KeyMess" that $router_1$ received from the sender. If this process fails, $router_2$ returns "KeyFail" to $router_1$, and then $router_1$ returns "KeyFail" to the sender. If the above process succeeds, the following operations are performed.
3. If, at last, $router_3$ succeeds in transferring "KeyMess" to the receivers, the receiver returns the "KeySucc" message to $router_3$, and $router_3$ will return "KeySucc" message to $router_2$, and so on, until finally the sender receives the "KeySucc" message. The Sender then begins to transfer the data along the "path" to the receivers.

We omit to detail the process of "KeyMess" transmission in this paper, because there are many mechanisms to implement it, e.g. the sender host first generates the session key "K" and encrypts the "KeyMess" (i.e. $E_K(\text{KeyMess})$), then the sender host uses the public key of $router_1$ to encrypt the session key "K" and encrypted KeyMess (i.e. $E_{router_1}(K, E_K(\text{KeyMess})$. At last, the sender uses the private key to encrypt a timestamp, encrypted "KeyMess" and session key (i.e. $E_{sender}(timestamp, E_{router_1}(K, E_K(\text{KeyMess}))$, and forwards it to $router_1$. The above operations assume the sender and $router_1$ can get each other's public key.

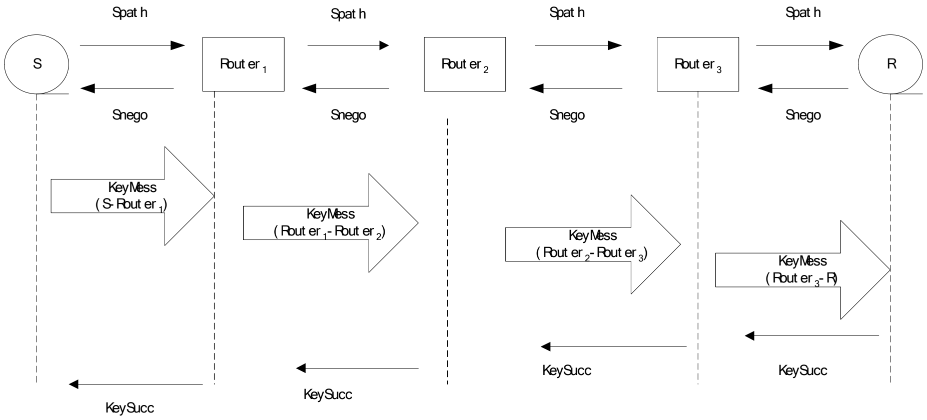

Fig. 3. SSRSVP mechanism

At each intermediate router, a "Spath" message triggers three general actions as follows (Note, the "Spath" message must pass the authentication and policy control of router):

1. The intermediate router adds the previous hop address in the SSRSVP_Hop field of the "Spath" message.
2. The intermediate router stores the information it extracts from the "Spath" message. The information includes the Sender Template, which describes the format of packet that the sender will originate. The Sender Template specifies the IP destination address, protocol ID and destination port. The security properties of the Sender specify the scope of quality of security service that is used to indicate the maximal and minimal security processing capability of the sender and claimed quality of security service level limit for receivers.
3. The intermediate router forwards the "Spath" message downstream towards the receivers along with uni/multicast routes provided by the routing protocols.

At each intermediate router, a "Snego" message triggers two general actions, as follows:

1. The SSRSVP module of the router passes the "Snego" request to the admission control engine and policy control. If either test fails, the negotiation request is rejected and the SSRSVP process returns a "SnegoErr" message to the receiver(s). If both succeed, the router will return the confirmation message (quality of security service) to the receiver(s), and the confirmation message indicates the quality of security service installed in the network. If the quality of security service level limit is claimed in the "Spath" message of sender, the router will check the quality of security service level of negotiation request of the receivers. If the negotiation request level is below the limit claimed by the sender, the router will reject the negotiation request of the receiver. When the receiver receives "SnegoErr" message from the router, the receiver will resend the negotiation request at a higher of high quality of security service level if the receiver has sufficient security processing capability. If the receiver has not the necessary security processing capability, it can't receive the packets from the sender.
2. The intermediate router makes negotiation mergence according to the negotiation style claimed by the receiver, and then a new negotiation request message is propagated upstream towards senders by the router.

The "SpathErr", "SnegoErr", "SpathTear", and "SnegoTear" messages are similar to the message types of RSVP [1]. We omit to further describe them in the article due to constraints on the article length.

5 NEGOTIATION STYLE OF SSRSVP

From the point of view of RSVP, a sender can always transmit data, whether or not an adequate bandwidth service exists in the network to deliver the data. However, since the quality of security service is different to the quality of (bandwidth) service, senders don't want to expose their message to others or be forged by others during transmission in the network. The sender sometimes can only hope when transferring packets that the receivers' quality of security service level is not below the limit claimed by the sender. Therefore, with SSRSVP, the sender can force the receivers to keep quality of security service level of negotiation request above the limit claimed by sender. Of course, the sender will always use the maximum security processing level to encapsulate data packets and forward them to next hop (router with SSRSVP) regardless of the negotiation request level from the next hop. To solve the problem, we hope to use a negotiation style to guarantee it. The negotiation style of SSRSVP is different to the reservation style RSVP [1]. We classify the style of SSRSVP into two categories: limit (i.e. LWF, LFF and LSE negotiation style) and non-limit (i.e., FF and SE negotiation style).

We discuss the negotiation style of SSRSVP in the following terms. Let a negotiation request be (SQ_i). SQ_i denotes the quality of security service level of negotiation request from the th receiver. $Inf_{s_j}()$means the quality of security service level negotiated from the receivers is not below the limit claimed by the th sender host. Since the non-limit negotiation style of SSRSVP is the same as the reservation style of RSVP [1], we omit to further describe it. We defined the limit negotiation styles as follows.

LWF style: $(*\max[Inf_{s_i}(SQ_i)])$ and WF style: $(*\max(SQ_i))\ i \in (1,\ldots,n)$

During the LWF style negotiating process, separate negotiation requests are merged into one negotiation request for each upstream sender.

LFF style: $(S1\max[Inf_{s_1}(SQ_i)],\ S2\max[Inf_{s_2}(SQ_j)])$ and FF style: $(S1 \max(SQ_i),\ S2\max(SQ_j))\ (i,j \in (1,\ldots,n))$

Assume that S1 and S2 denote the $sender_1$ and $sender_2$ .The LFF style negotiation request creates a distinct negotiation for data packets from a particular sender, not sharing them with packets of other senders for the same session.

LSE style: $((S1,S2)\{\max[Inf_{s_1}(Inf_{s_2}(SQ_i))]\})$ and SE style: $((S1,S2)\{(SQ_i)\})$ $(i \in (1,\ldots,n))$

The LSE style negotiation request creates a single negotiation shared by selected upstream senders. Unlike the LWF style, the LSE style allows a receiver to explicitly specify the set of senders to be included.

6 SSRSVP EXAMPLE

Users hope that their data packets are transferred with confidentiality and integrity in the network. To conveniently explain the SSRSVP example, suppose the quality of security service level is defined as the following Figure 4.

Figure 5 and Figure 6 show one example of negotiation request from one source node to the four receiver nodes by using FF and LFF style, respectively.

Because SSRSVP is a receiver-oriented request protocol similar to RSVP [1], receivers R_1, R_2, R_3, and R_4 will initiate a LFF style SSRSVP request. Suppose the $router_1$, $router_2$, $router_3$, $router_4$ and $router_5$ have available security processing capability to support the SSRSVP negotiation request. Assume that the quality of

cryptography and integrity algorithms	DES and MD5	3DES and MD5	1024 bits RSA and MD5	1024 bits RSA and SHA	2048 bits RSA and SHA
Quality of security service level	1S	2S	3S	4S	5S

Fig. 4. Quality of security service level

security service level of negotiation request from receivers R_1, R_2, R_3, and R_4 is 4S, 2S, 1S and 5S, respectively.

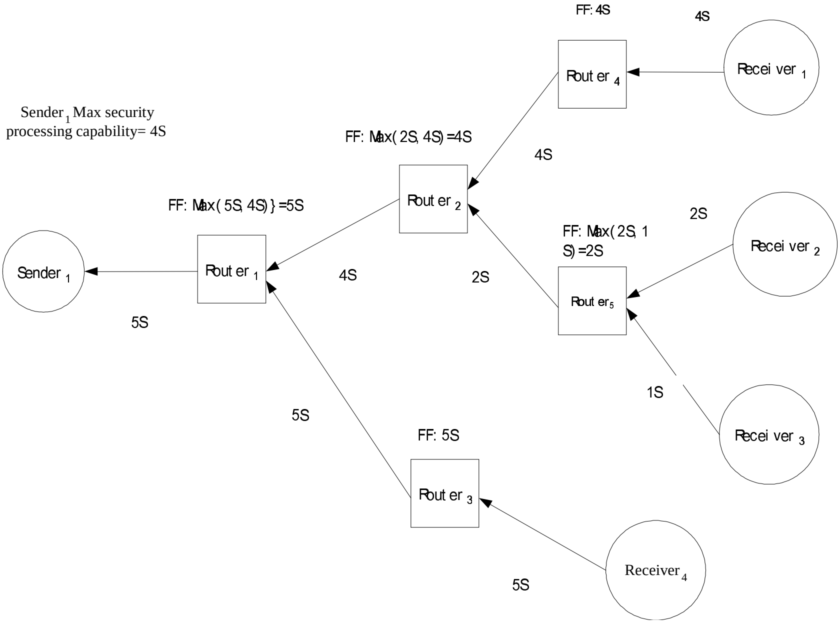

Fig. 5. One complete SSRSVP negotiation request example from one source node to the four different receiver nodes using FF style

In the Figure 6, since the negotiation request level of the $receiver_3$ is below the limit claimed by $sender_1$, the $Router_5$ rejects the negotiation request of $sender_1$. If the max quality of security service level of the $receiver_3$ is 1S, the $receiver_3$ will never receive any data packets from the sender1. Of course, if 1S is not the max security process capability of $receiver_3$, the $receiver_3$ will try to sends the negotiation request of high quality of security service level to the $Router_5$ when the $receiver_3$ receives the "SnegoErr" message.

When the negotiation request is finished, the sender node S1 sends the "Key-Mess" that includes negotiation acknowledgement information and the Key informa-

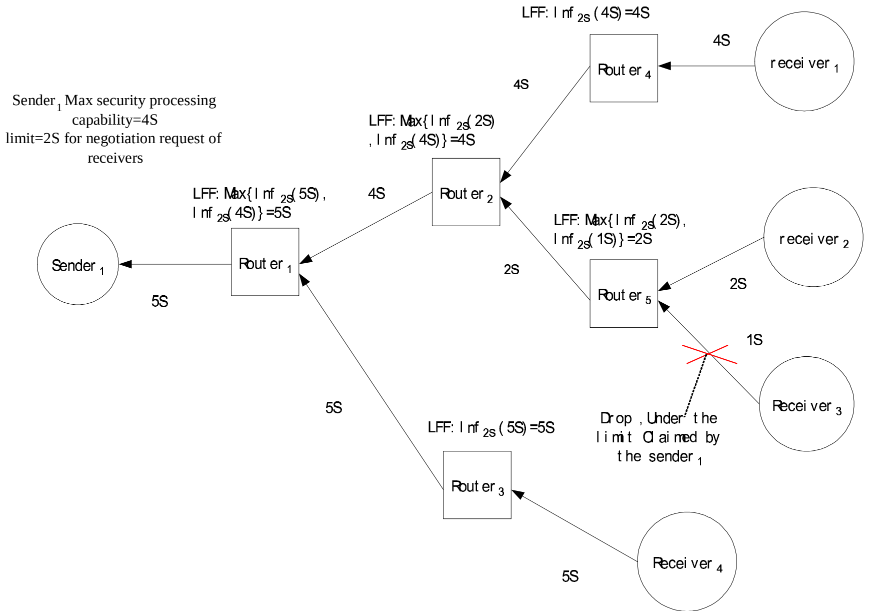
LFF: Inf 2S (4S) =4S
4S
Router 4
receiver 1
4S
Sender 1 Max security processing capability=4S limit=2S for negotiation request of receivers
LFF: Max{Inf 2S(2S), Inf 2S(4S)} =4S
LFF: Max{Inf 2S(5S), Inf 2S(4S)} =5S
4S
Router 2
LFF: Max{Inf 2S(2S), Inf 2S(1S)} =2S
receiver 2
Sender 1
Router 1
2S
Router 5
2S
5S
1S
Receiver 3
5S
LFF: Inf 2S (5S) =5S
Drop , Under the limit Claimed by the sender 1
Router 3
Receiver 4
5S

Fig. 6. One complete SSRSVP negotiation request example from one source node to the four different receiver nodes using LFF style

tion of quality of security service that $router_1$ uses to decrypt data packets encrypted by the sender. When the Router1 receives the "KeyMess" from the $sender_1$, the $router_1$ sends negotiation acknowledge information and key information of quality of security service that the $router_2$ and $router_3$ will use to decrypt data packets encrypted by the $router_1$, respectively. When all of the receiver nodes receive the upstream "KeyMess" message, the SSRSVP setup is successful. The detailed process is illustrated in the Figure 7 and Figure 8 by Using FF and LFF style, respectively.

7 DISCUSSION ABOUT SSRSVP

7.1 Tear, Error Message and Policy Control

SSRSVP "teardown" messages remove path or negotiation state immediately. There are two types of SSRSVP teardown message: SpathTear and SnegoTear. A SpathTear message travels towards all receivers downstream from its point of initiation and deletes path state, as well as all dependent negotiation state along the way. A SnegoTear message deletes negotiation state and travels towards all senders upstream from its point of initiation. Similarly, there are two SSRSVP error message: SpathErr and SnegoErr. SpathErr message sent upstream to the sender that created the error, and they do not change path state in the routers though which they pass.

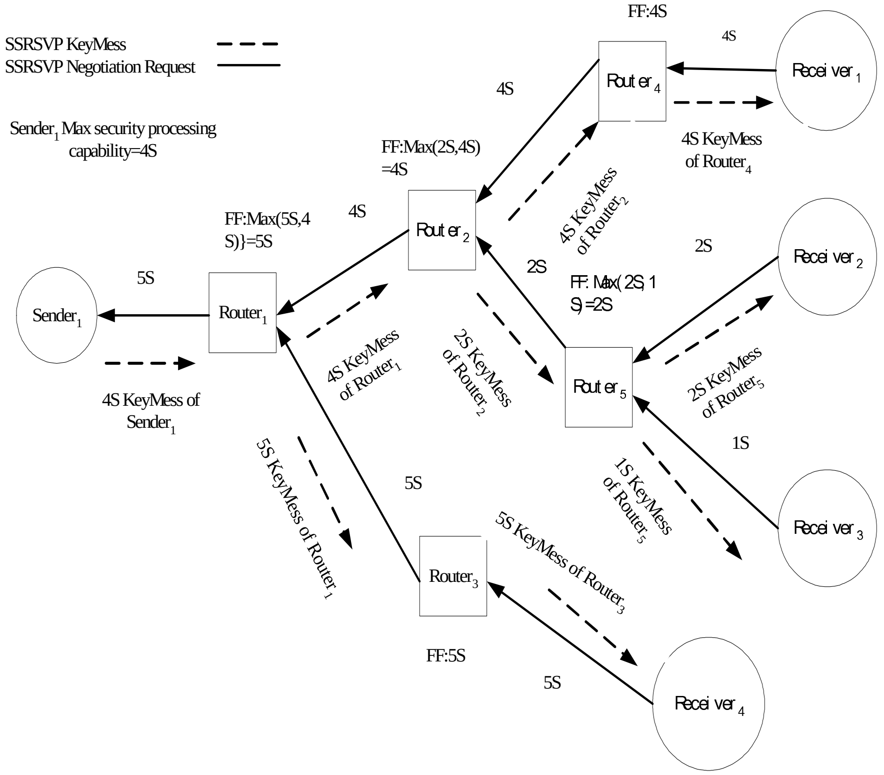

Fig. 7. One complete SSRSVP negotiation request and KeyMess example from one source node to the four different receiver nodes by using FF style

A negotiation request of quality of security service that fails Admission Control creates SnegoErr.

The router will spend plenty of CPU and memory cost when it implements the cryptography algorithms for quality of security service of receivers. To prevent CPU and memory resource abuse by users, reliable user identification and selective admission will generally be needed when negotiation for quality of security service is requested. “Policy control” is used for the mechanisms required to support access policies and back pressure for SSRSVP negotiation for quality of security service. SSRSVP carries POLICY_DATA objects. Policy data may include credentials identifying users or user levels, account numbers, limits, etc.

7.2 SSRSVP Refresh and Non- SSRSVP Clouds

In the context of a SSRSVP-enabled network, a soft state refers to a state in router and end router that can be updated by certain SSRSVP messages. The soft state

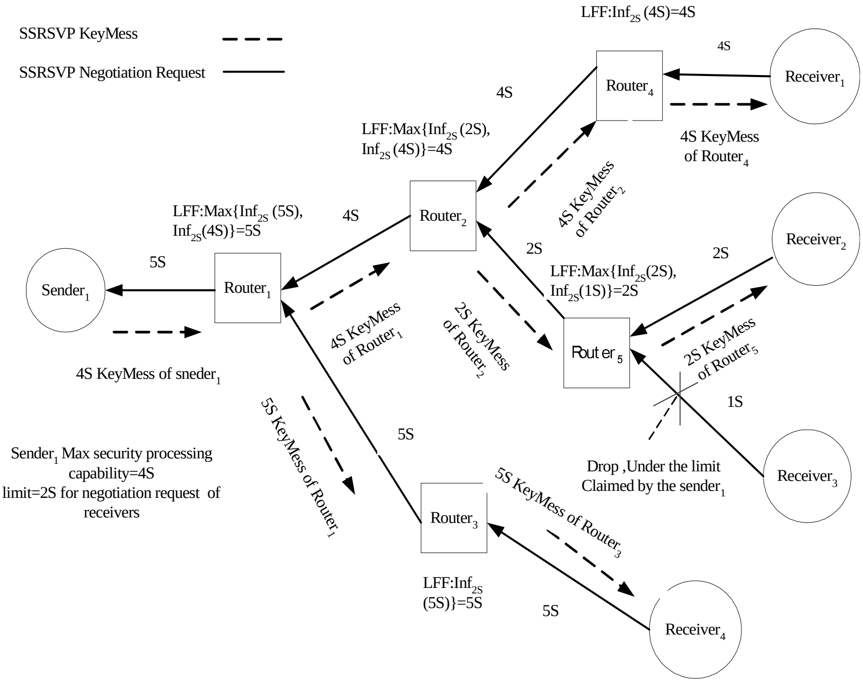

Fig. 8. One complete SSRSVP negotiation request and KeyMess example from one source node to the four different receiver nodes by using LFF style

characteristic permits a SSRSVP network to support dynamic group membership changes and adapt to changes in routing. In general, the soft state is maintained by a SSRSVP-based network to enable the network to change states without consultation with ends. To maintain a security negotiation state, SSRSVP tracks a soft state in the router and host. The SSRSVP soft states are created and must be periodically refreshed by Spath and Snego messages. SSRSVP periodically scans the soft state to build and forward Spath and Snego refresh messages to succeeding hops. When a route changes, the next Spath message initializes the path state on the new route. When state changes occur, SSRSVP immediately propagates those changes from end to end within a SSRSVP network. If the received state differs from the stored negotiation state, the stored state is updated. If the result modifies the refresh messages to be generated, refresh messages are generated and forwarded immediately. A host sends IGMP messages to join a multicast group and SSRSVP Messages to negotiate quality of security service along the delivery path(s) from that group. Each router that is capable of participating in negotiating quality of security service passes incoming active packets to a packet classifier. The SSRSVP packet classifier determines the route and quality of security service level for each packet.

It is impossible to deploy SSRSVP at the same time throughout the entire Internet. Therefore, SSRSVP must provide correct protocol operation even when two SSRSVP-capable routers are interconnected via an arbitrary cloud of network routers. An intermediate cloud that does not support SSRSVP is incapable of performing security negotiation, so quality of security service guarantees cannot be made. To support connection of SSRSVP networks through traditional networks, SSRSVP supports tunneling, which occurs automatically through non-SSRSVP networks. Tunneling requires SSRSVP and non-SSRSVP routers to forward Spath messages toward the destination address by using a local routing table. When a Spath message traverses a non-SSRSVP cloud, the Spath message-copies carry the IP address of the last SSRSVP-capable router. Snego messages are forwarded to the next upstream SSRSVP-capable router.

8 SIMPLE IMPLEMENTATION

Different negotiation styles and quality of security service will bring different impacts on the performance of a router. Consider an experiment consisting of one sender, three receivers and two routers, which are PCs, as illustrated in Figure 9.

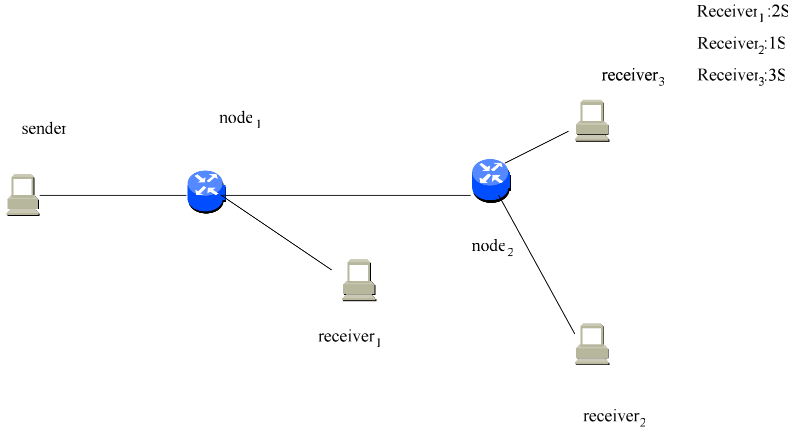

Fig. 9. Experiment paradigm

In the experiment, let the 64 bits DES [5] and MD5 [6] be the quality of security service first level (i.e. 1S). Let be 3DES[7] and MD5 be the quality of security service second level (i.e. 2S). Let be 1024 bits RSA [8] andMD5 be the quality of security service third level (i.e. 3S). The DES, 3DES and RSA are used to ensure the confidentiality of packets. The MD5 is used to ensure the integrity of packets. We test the relation of processing throughput of packet with the utilization of CPU and RAM of router1 and router2 when the two different negotiation styles of SSRSVP are used in our experiment. The simulation experiment results are shown in Figure 10 and Figure 11.

The simple experiment shows that the CPU capability to process algorithms for confidentiality and integrity is a bottleneck to improved network throughput. Although it is feasible to implement algorithms for confidentiality and integrity by software in the router, we can see that the throughput of network processing is improved only if the algorithms for confidentiality and integrity are implemented by hardware modules in a router.

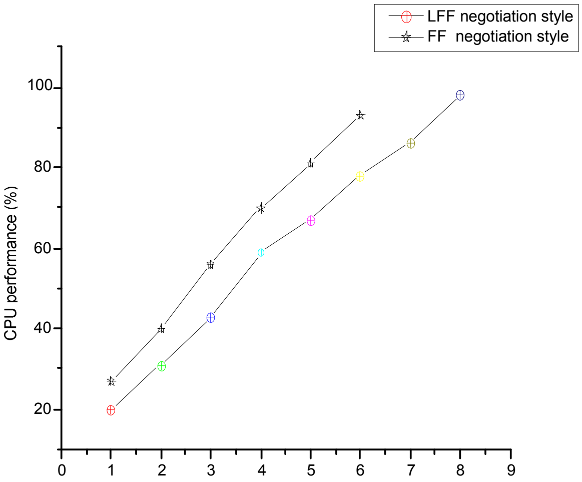

Fig. 10. Throughput out $node_2$ (M bytes/s)

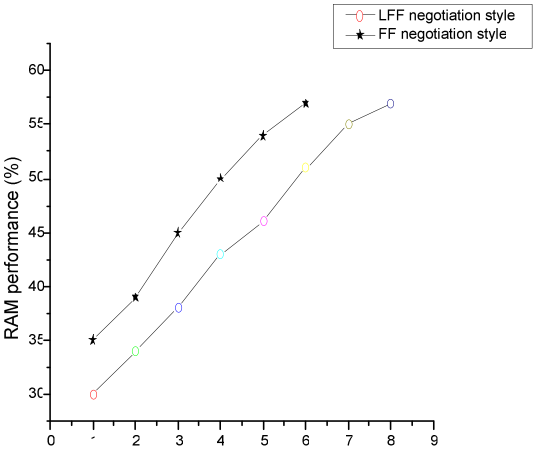

Fig. 11. Throughput out $node_2$ (M bytes/s)

9 SUMMARY

With future network equipment the security service becomes a critical and serious problems. Especially in the network, users don't want to expose their message to others or to be forged by others. They make extensive use of cryptography and integrity algorithms to achieve security. The sender can achieve the high quality of security service (high security level), only if the receivers and routers along path to receivers can support or satisfy the quality of security service requsted by the sender.

To solve above the problems, we propose an extension of RSVP called SSRSVP (security service RSVP) to provide the needed mechanism for quality of security service, to dynamically negotiate the quality of security service among the senders and receivers of multicasts on the Internet. It provides different quality of security service resolutions to different receiver nodes with different security service needs.

In this paper, SSRSVP is proposed, which differs from RSVP [1] in the following three aspects.

RSVP [1] is used to guarantee the quality of (bandwidth) service negotiation. SSRSVP is designed to provide a flexible mechanism to guarantee the quality of security service negotiation among senders and receivers in the network.

There are two fundamental message types in the RSVP [1] (e.g. Path and Resv message). In SSRSVP, we define three fundamental messages: "Spath", "Snego" and "KeyMess". The "KeyMess" message includes the negotiation acknowledgement and key information of quality of security service.

Since quality of security service is different from the quality of (bandwidth) service, the sender is given the right to force receivers to keep quality of security service level of negotiation request above limit claimed by the sender and to ensure confidentiality and integrity of the sender's data packets during transmission in the network. Therefore, we make small change in the style of RSVP [1]. We extend the reservation styles of RSVP [1] to negotiation styles of SSRSVP, which includes WF, FF, SE, LWF, LFF and LSE.

Detailed discussion of security consideration for SSRSVP is omitted in the paper. We only propose a mechanism of SSRSVP for negotiating quality of security service, which could provide different quality of security service resolutions to different receiver nodes with different security service needs.

Acknowledgments

ZhengYou Xia's research was supported by grants from the JiangSu High Technology Research Plan Project No. BG2004005. We gratefully acknowledge the assistance of reviewers of this paper from our hearts. Their detailed comments helped us to substantially improve the accuracy and readability of the paper.

REFERENCES

[1] Zhang, L. et al.: RSVP: A New Resource Reservation Protocol. IEEE Network, Sept. 1993, pp. 8–18.

[2] Braden, R. et al.: Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification. RFC 2205, Sept. 1997.

[3] Braden, R.—Clark, D.–Shenker, S.: Integrated Services in the Internet Architecture: An Overview. RFC 1633, June 1994.

[4] Irvine, C.—Levin, T.: Quality of Security Service. Proc. of New security Paradigms workshop 2000, Cork, Irelanbd, September 2000.

[5] ANSI X3.106, American National Standard for Information Systems – Data Link Encryption. American National Standard Institute, 1983.

[6] Rivest, R. L.: The MD5 Message Digest Algorithm, RFC1321, Apr. 1992.

[7] Kenneth Castelino 3DES and Encryption, `http://kingkong.me.berkeley.edu/~kenneth/courses/sims250/des.html`.

[8] Rivest, R. L. et al.: Amethod for Obtaining Digital Signatures and Public-Key Cryptosystems. Communication of the ACM, v21, n.2, Feb. 1978, pp. 120–126.

[9] Peterson, J.: Neustar, RFC 3323 – A Privacy Mechanism for the Session Initiation Protocol.

[10] Handley, M.—Jacobson, V.: SDP: Session Description Protocol. RFC 2327, April 1998.

[11] Cherkasova, L.—Phaal, P.: Session Based Admission Control: A Mechanism for Improving Performance of Commercial Web Service, `http://citeseer.ist.psu.edu/cherkasova98session.html`.

[12] Arkko J.—Torvinen, V.—Niemi, A.—Haukka, T.: Security Mechanism Agreement for the Session Initiation Protocol. RFC 3329.

[13] RFC 2408, ISAKMP.

[14] Mittra, S.: A Framework for Scalable Secure Multicasting. ACM SIGCOMM '97, pp. 227–288, 1997.

[15] RFC 2747 – RSVP Cryptographic Authentication.

[16] RFC3097 RSVP Cryptographic Authentication – Updated Message Type Value.

[17] RFC 2207 – RSVP Extensions for IPSEC Data Flows.

ZhengYou Xia (born in 1974) is Associate Professor at Nanjing University of Aeronautics and Astronautics. He received Ph. D. degree in computer science of the Department of Computing and Information Technology, Fudan University, China. His research interests include mobile agent system, and network security.

Yichuan Jiang was born in 1975. He received his M.S. degree in computer science from Northern Jiaotong University, China in 2002. He is currently a Ph. D. candidate in computer science of the Department of Computing and Information Technology, Fudan University, China. His research interests include mobile agent system, artificial intelligence and network security.

Wang Jian, Associate Profesor, Ph. D., received the Ph. D. Eng. degree in October 1998 from the Department of Computer Science and Technology, Nanjing University. His present research interests include multicast security, key management, broadcast encryption, and sensor network.