ISCP: Design and Implementation of An Inter-Domain Security Management Agent (SMA) Coordination Protocol [*]

Z. Fu, H. Huang, T. Wu, S. F. Wu
Computer Science Department
North Carolina State University
Raleigh, NC 27695
{zfu, hhuang2, twu2,wu}@eos.ncsu.edu

F. Gong, C. Xu, I. Baldine
Advanced Networking Research
MCNC
RTP, NC 27709-2889
{gong, chong, ibaldin}@anr.mcnc.org

Abstract

Many security mechanisms and protocols have been developed to handle security problems in various circumstances. This trend has created a heterogeneous security environment for today's global Internet. Although most of security functions and modules can be managed “individually” through SNMP, very little has been researched in coordinating a set of distributed security modules to provide and manage an "End-to-End" security service. In order to support service management for network security, in this paper, we developed the ISCP protocol (Inter-Domain Security Management Agent Coordination Protocol) to communicate security capability and policy information among the security management agents in each policy domain. ISCP is designed with good scalability, interoperability, extensibility and security. Performance evaluation using our ISCP prototype implementation are also presented.

Keywords

Security Management Agent, Quality of Protection, SNMP, RSVP, Security Association, IPSec, ATM CellCase, Firewall Traversal, Security Policy

1. Introduction

The explosive growth in worldwide communication via the Internet has increased the reliance of organizations and individuals on the electronically transmitted information, which consequently created rising demands to protect data from information leakage, corruption or alteration during transmission. Various security service requirements are demanded among different applications and customers

[*] This research is supported by the U.S. Department of Defense Advanced Research Projects Agency under contract DABT63-97-C-0045.

with consideration of respective data sensitivity level, performance requirement and monetary investment. For example, some applications care about the message integrity while some others are more concerned with message confidentiality or both. Some customers adopt ordinary security service with cheap price while others might be willing to pay a higher price to get stronger security service for their highly sensitive data. It becomes important to provide end-to-end security service commitment to satisfy the diverse customers needs. We expect the Quality of Protection (QoP) to fulfill end-to-end security service commitment is to be included within the integrated network service model [1] to support secure QoS Internet service.

Recently, many security mechanisms and protocols have been developed to handle security problems in various circumstances. This trend has created a heterogeneous security environment for today's global Internet. For instance, the IPSec framework [2,3,4] has been standardized for IP layer security, while ATM cell encryption [7,8] has been implemented in hardware to support Gigabit network traffic. On the other hand, firewall venders such as Checkpoint has deployed solutions for not only blocking unauthorized traffic in different protocol layers but also interacting, in a limited sense, with other security management products such as intrusion detection (ISS's RealSecure) through the IPSec protocol. Most of the existing research efforts focus on development of individual security mechanisms and protocols to solve a particular security problem. Although many "individual" security functions and modules have been deployed and most of them can be managed "individually" through SNMP, very little has been researched in coordinating a set of distributed security modules to provide and manage an “End-to-End” security service.

What we need, in order to systematically support security services for an “End-to-End” application, is a service management framework for network security. The Celestial project [9] at MCNC/NCSU is targeting at this goal in developing a security management architecture, which enables an end-to-end application to reliably utilize multiple distributed security mechanisms in a heterogeneous networking environment. In this paper, we will present the ISCP protocol (Inter-Domain Security Management Agent Coordination Protocol), the core protocol under the Celestial architecture, for exchanging security capability and policy information among the security management agents in each policy domain. ISCP is designed with good scalability, interoperability, extensibility and security. Performance evaluation using our ISCP prototype implementation will also be discussed.

2. Challenges in End-to-End Security Service Provisioning

Various Security Capabilities: Diversified security development efforts have created a heterogeneous networking security environment. Firstly, cryptographic algorithms are varied from fast shared secret algorithm to expensive public key agorithm. Also, diverse security mechanisms are provisioned in multiple protocol layers, for example, SSL/TLS[5,6] in transport layer, IPSec in IP layer, ATM encryption system in ATM layer etc. Furthermore, various implementation

0-7803-5930-5 (c) 2000 IEEE

technologies are from all hardware to all software. By security capability, we mean a particular implementation of a security mechanism, for example, SSL3.0 implemented on a workstation, IPSec on a router etc. Different devices (host, router, switch etc.) might equip with different security capabilities.

While each of these security mechanisms may solve a particular problem well, there is significant overlap among the security services they provide. Moreover, there are important tradeoff (e.g. degree of security and performance) associated with these solutions. One important question is how to efficiently utilize and manage the capabilities to provide satisfying end-to-end security service.

Security service provisioning relies on establishment of Security Associations (SA) in which two party agree upon suite of keys and protocols to do encryption/decryption, signature/verification. However, two devices with different security capabilities can not communicate and set up SA. For example, if one end has done correct signature but the other end does not have correct algorithm to verify it, then the authenticity requirement clearly can not be accomplished. Even among those devices which have compatible security capabilities, the tasks of locating existing SAs, establish necessary new ones, terminating unused ones could still be difficult to achieve without a management system.

Diverse security policies: Firewall is widely used to restrict access and selectively enforce IPSec operations. The selection is accomplished through policies manually configured by administrators. The diversified regional security policy enforcement created significant management problems for end-to-end communication. The draft [11] described some of the pitfalls applications may encounter as a result of firewall policy enforcement. One is that if the two peering security gateways do not have matching policies, then packets could be tunneled by the local gateway device and dropped by the peering device. The other one is that even if peering gateways have matching policies and the associated SAs, there could be security breaches when the policies have overlap and the order is different on the peering gateways. In addition, firewalls could also seriously block legitimate communication due to lack of mandatory SAs. For instance, host A and host B establish an IPSec ESP SA between them to protect their sensitive data exchange without awareness that a firewall on the path will not allow any encrypted packets to pass. As a result, all their packets are dropped in the middle by the offended firewall.

Efficiency: Some cryptographic algorithms are very compute intensive. There is tradeoff between high-degree security and high speed communication itself. One challenging problem is to manage the diverse security capabilities such that an end-to-end security service can be provided with highest performance possible.

Survivability: One other important problem is how we can manage security capabilities so that they can be reconfigured dynamically upon route changes, policy update, detection of intrusion or security service degradation etc., in order to maintain adequate levels of end-to-end security service.

3. Overview of Celestial system

3.1 Celestial two phase security context establishment and ISCP

Celestial system aims to provide reliable and scalable end-to-end security services using multiple distributed security mechanisms and create an integrated framework for security management, and intrusion detection and response coordination. In Celestial system, Security Management Agent (SMA) is to sit in management plane of any SMA-enabled node (switch, router, security gateway etc.) and is authorized to configure or reconfigure various local security mechanisms at all protocol layers. The SMAs are responsible for coordinating all security-related activities on a network system.

Inter-Domain SMA Coordination Protocol (ISCP) provides the transport function for security service negotiation and reservation in order for the Celestial system to implement Quality of Protection (QoP). The messages to be transported in ISCP include the security service request, security capabilities and policies of SMAs, security configuration/assignment for provisioning the requested services, and maintenance messages, etc.[9].

In Celestial, security context establishment is done in two phases: the discovery phase where the application's service requirements are distributed along the communication path and the service capabilities/policies of the nodes along the path are collected; and the reservation phase which distributes assignments to the nodes selected for providing the security services. ISCP is designed to support this two-phase operation.

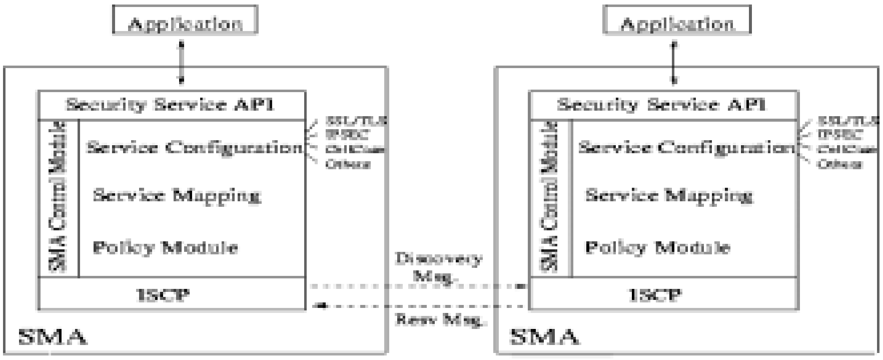


Figure 1: Two Phase Security Context Establishment

In the discovery phase, the sender (SMA for the sending application) will issue a discovery message downstream to the receiver. Every SMA node on the path will append its security capability/policy information to the message if its capability and policy support the request. The message will eventually come to the receiver who will analyze all the intermediate nodes' security capabilities/policies and determine the optimal set of SAs to set up to satisfy the security requirement. Once the decision is made, the receiver will invoke the reservation phase by issuing a security reservation message which contains the assignment information for each node. The reservation message will be sent back along the reverse path of the discovery message. Each SMA node will pick its assignment up upon receiving the reservation message and take appropriate actions (e.g. trigger the SA setup process).

Confirmation messages are sent from those assigned nodes to the sender when their corresponding part of the security service is established. Upon successful confirmation of all the selected security associations, the sender will inform the receiver that the end-to-end security context is complete. The final acknowledgment from the receiver to the sender signals the sender that the receiver is ready for receiving and the sender can start the data flow on this security context. The context thus provides the security services required by the application. The refreshing discovery and security reservation messages will be sent periodically to dynamically adapt to route changes and intrusion event etc. Finally a teardown message will be transmitted from the sender along the same path of security context at the end of the transmission to delete the context and the state information maintained for the session.

The ISCP is used to provide the transport service to support the two-phase process. In Celestial system, for example, the Security Mapping & Configuration Modules [9] will be responsible for determining the optimal configuration plan and push the decision down to ISCP for transmission. Policy module determines whether the requested service is permitted by the policy. The messages transported by ISCP is opaque to ISCP, which simply hands them to upper layer modules for interpretation and processing.

3.2 Example Scenarios

We present several example scenarios in figure 2 to illustrate how SMAs help to establish a security context to fulfill the end-to-end security service requirement in a heterogeneous environment.

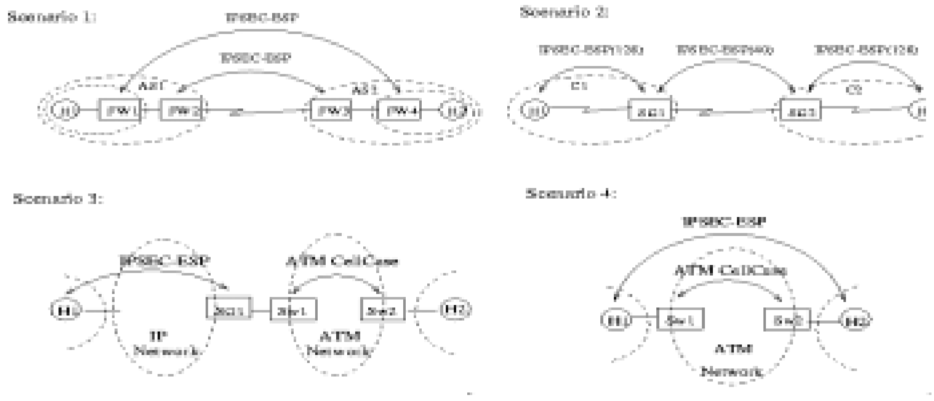


Figure 2: End-to-end encryption over heterogeneous environment

Assume there are two end hosts H1 and H2 that need to exchange large amount of sensitive information between them and the sending application in H1 requests SMA for confidentiality service.

In the scenario 1, H1 and H2 are not IPSec capable. There are four firewalls encountered en-route. The policies regarding the application flow in the four firewalls are similar: one IPSec-ESP SA must be set up between itself and a security gateway in the domain of the other end. Upon having collected the security capability/policy information along the path, the receiver SMA makes decision that the security requirement and en-route policies can be satisfied by establishing two SAs as drawn in the figure: one from the first firewall (FW1) to the fourth firewall (FW4), the other from FW2 to FW3, as long as the link between H1 and FW1, and the link between FW4 and H2 are safe. However, if the policy for FW3 is that no encrypted flow can be allowed unless it is one party of the SA, which conflicts with the policy of FW4 that one SA must be set up from one security gateway in AS1 (Administrative System) to FW4, then SMA will send error message to notify the policy conflict. This example is where Celestial system can be used for conflict detection and policy management.

In the scenario 2, H1 and H2 are in different countries. The export law says that the key length within C1 (country 1) can be as large as 128 bits but the key length outside C1 can only be no larger than 40 bits. The same policy is applied in C2. To ensure the reliable confidentiality service, the receiver might decide to arrange to set up three SAs as drawn in the figure, instead of establishing a single SA using 40 bit key length which is less secure. There is tradeoff to consider between performance and security.

In the scenario 3, H1 is IPSec capable but H2 is not. On the transit path, the SG1 (security gateway 1) enables IPSec while Sw1 and Sw2 (ATM switch) only support CellCase encryption. With all these capability information, the receiver SMA finds that the confidentiality protection can only be arranged by using IPSec ESP between H1 and SG1 and then using ATM cell encryption between Sw1 and Sw2; furthermore, the link between SG1 and Sw1, as well as the one between Sw2 and H2 are trustworthy.

The scenario 4 illustrates how SMA can reconfigure security context to maintain a good security service throughout communication. In the scenario, both H1 and H2 are IPSec capable. The Sw1 and Sw2 are CellCase compliant. At the initial security context establishment, SMAs have made the decision to bypass the IPSec mechanisms at the end hosts and rely only on the ATM cell encryption for fast operation. However, later in the communication some intrusion alarm go off indicating that some components in the ATM network may have been compromised. This event can be detected at any of the SMAs along the communication path between H1 and H2, from their interfaces with some intrusion detection conmponents. In this case, it is determined that IPSec-ESP should be applied on top of the ATM cell encryption in order to provide the additional protection. Therefore, the new configuration plan is distributed to all the SMAs involved and the security context is reconfigured accordingly.

4. ISCP protocol design

4.1 Design objectives

The design of ISCP protocol has the following objectives:

- We design several types of message to help efficiently accomplish the two-phase process for end-to-end security context establishment.
- We adapt some existing mechanisms of RSVP[13] (e.g. Two phase, Soft state) to achieve ISCP goals. By doing so, we benefit not only from the mature experience in RSVP but also in future integration of QoP with QoS provisioning. More importantly, new mechanisms are developed to best suit ISCP needs.
- The protocol needs to provide the security assurance for the delivery of ISCP messages. There are few research on RSVP message security. As identified in [15], the INTEGRITY object proposed by Baker in [14] does not help in preventing insider attacks, if we define the attacks from RSVP-enabled nodes on the communication path to be insider attacks and others to be outsider attacks. We'd like to propose ISCP security schemes for both insider and outsider attacks.
- Security protection is very important to applications. However, the value of security protection diminishes if the additional overhead prevents effective communication from happening. Optimal operational efficiency and scalability is the goal in protocol design and software development.

4.2 ISCP protocol process overview

We illustrate the ISCP protocol process of transmitting and securing messages with an example shown in Figure 3. In the example, there are four ISCP-capable nodes (including the sender and receiver) on the transit path between the sender S and the receiver R. To ensure ISCP security, the information provided to the ISCP messages need to be encrypted and signed. The public key algorithm can be used to perform ISCP message security to prevent insider attack (definition in section 4.4) but it is computing intensive. To improve efficiency, ISCP protocol is also utilized to distribute the shared secret keys which are generated by each node on the path and shared between the node and the sender/receiver to facilitate later communications. We will focus on overall process of the protocol while message content and security will be elaborated in next subsections.

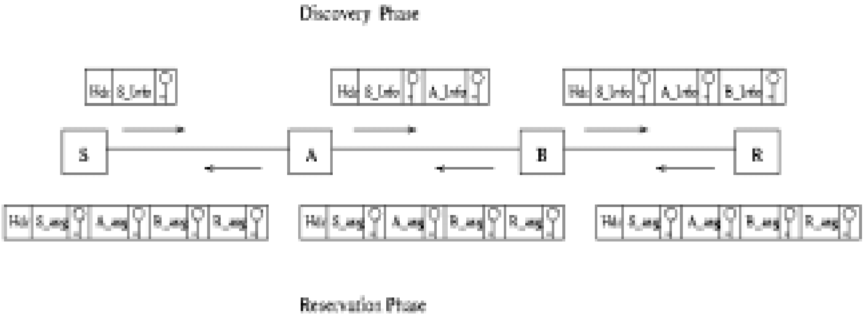


Figure 3: ISCP message transmission

First, being signaled by the application, the sender issues a Discovery message which is composed of an ISCP header (contains session ID, source address, destination address etc.) and the request information. Then it encrypts its capability and policy information with a locally generated secret key and encrypt the local key with receiver's public key (such that only receiver can decrypt), then sign the whole message before append to the Discovery message to deliver. In the figure, the S_Info box indicates the encrypted capability/policy information of the node S and the small key figure indicates the encrypted secret key of the node. The intermediate nodes A and B will intercept the message and append their secured capability/policy and secret key information with the same procedure as S did. Eventually the Discovery message arrives at the receiver R which will verify all the signature and decrypt the secret keys and capability/policy information. Then the receiver will decide an optimal security context utilizing the existing security capabilities on the path without violating relevant policies. Now it starts the reservation phase to distribute the assignment to each node and transmit the secret keys to the sender. The receiver will encrypt each assignment using the corresponding node's secret key (such that only the right node can decrypt) and encrypt all secret keys using the sender's secret key (such that only the sender can decrypt). Then it concatenates the pieces together and sign the whole message before transmits it upstream to the sender. Each intermediate node will find its assignment and act upon it for its part of security context establishment. At last, the sender will receive the message and obtain all the secret keys and assignment information (decrypt using those secret keys). Later communications (refreshing, confirming, error reporting etc.) will be greatly expedited using shared secret keys which are set up and distributed in the first round.

Those participants will start required SA establishment and will signal the sender the completion of SA's setup by a Confirmation message. Upon successful confirmation of all the selected security associations, the sender will inform the receiver that the end to end security context is complete by an ContextReady message. The final acknowledgment ContextReadyAct from the receiver to the sender signals the sender that the receiver is ready for receiving and the sender can start the data flow on this security context. The context thus provides the security services required by the application. The refreshing discovery and security reservation messages will be sent periodically to update the state and dynamically configure/reconfigure to adapt to route changes and maintain adequate security service. During the transmission, any error condition will be reported to sender and receiver by an Error message. Finally a teardown message will be transmitted from the sender along the same path of security context at the end of the transmission to delete the context and the state information maintained for the session.

4.3 ISCP messages

In Celestial system, the soft state of sessions are created and maintained in Session Control Table (SCT) in SMA state controller of SMA control module. ISCP will simply hand the collected information to the state controller to store and maintain the state information. We implemented SCT in our ISCP prototype as well to make ISCP a relatively independent software module. The SCT contains: (1) a unique session ID, (2) original service request, (3) assignment/role of the node, (4) pointers to control blocks of local capabilities servicing the request, (5) last path time and last reservation time which is used for time-out check and refreshing triggering, and (6) previous hop addresses.

The previous hop addresses in the table is for installation of a reverse path for reservation message. As we know, the reservation message must travel the same route of opposite direction as the discovery message. In discovery phase, each node will record the previous hop address. Then in reservation phase, every node will forward the reservation message directly to the previous hop so as to ensure the reservation message travel through the same SMA nodes as discovery message did.

ISCP daemon (or a separate state controller) needs to establish and maintain the entries in SCT for the application sessions dynamically, i.e. periodically bring the state up-to-date, and trigger the reconfiguration when the old ones is no longer satisfactory.

All the messages consist of common header and message body part. There are message type, source address, destination address, security context handle, sequence number, checksum fields in the common header. The message type is to distinguish among the ISCP messages; The security context handle is to identify the security context for a particular flow; Sequence number is used for preventing replay attack; Use of Checksum field will be explained in section 4.4. Different types of message have different message body. We will discuss them respectively in the following.

Discovery and Reservation messages: The two fundamental message types are Discovery and Security reservation messages. The discovery message is initiated by the sender and transmitted all along the path to the receiver. It contains the security service request and collects capability/policy information of all SMA nodes on the transit path. Capability information includes security mechanisms which the node supports like IPSec AH, IPSec ESP, CellCase ATM encryption, SSL etc. The policy information includes transfer policy and IPSec policy (see [17]) . The reservation message contains the assignment for each SMA node on the path about how they may participate in the security context for the request.

Soft state and refreshing messages: ISCP takes a soft state approach like RSVP to dynamically adapt to changes. The refreshing messages will be sent out by the sender and the receiver periodically. Refreshing discovery message and reservation message have the same message body as that of the discovery and reservation message. Message type is used to tell it is original or refreshing one.

Error reporting: Once running into error condition, an intermediate node will report to the sender or the receiver for fast repair. The errors identified include message processing error (wrong message format or content), policy control error, SA setup error, etc. Error message body contains error code and error value to specify the error.

Confirmation: Every participating node will send the confirmation message to the sender once its part of security context is successfully established. The confirmation message does not have message body. The message type and source address in the header indicate it is a confirmation from a particular node.

Sender and receiver synchronization: The sender will send ContextReady message to the receiver to notify the complete status of the security context upon receiving all the expected confirmations. The receiver sends a ContextReadyAck message back to the sender for acknowledgment of receiving ContextReady message and notification that its local setup is ready to receive the application's packets. The ContextReady and ContextReadyAck has no message body.

Teardown: Teardown messages are used to delete the session entry and free the unneeded context. Sender will issue a Teardown message to go along the same path as the Discovery message did.

The detail message format and content can be found in protocol specification.

4.4 ISCP message security

It is critical for the inter-SMA exchanges to be secure in order to support end-to-end service negotiation and configuration. Mechanisms are proposed to ensure authentication and confidentiality of ISCP messages hop-by-hop, to protect the messages against corruption, spoof, forgery, message modification, message replay, repudiation of sender etc. We classify the attacks from ISCP-enabled nodes on the communication path to be insider attacks and the attacks from non-ISCP nodes on the path and all other nodes not on the path to be outsider attacks. The proposed security schemes prevent the ISCP messages from both insider and outsider attacks.

To defend against insider attacks, public key system (PKS) is utilized to do authentication and encryption. The security capability/policy information is encrypted for confidentiality of security capability and privacy of locally established policies. Every piece of information provided to ISCP will be signed by the provider for authenticity and integrity.

The security of ISCP messages is accomplished at the cost of expensive PKS usage. Schemes to greatly reduce the usage of PKS without sacrificing the security are employed. The idea is to distribute shared secret keys using ISCP messages. We will talk about the security mechanisms employed in ISCP in detail for the two phases respectively next.

Discovery phase:

In discovery phase, the sender will first create a discovery message which consists of ISCP common header and the request. The discovery message will look like the following (we will define the appendList later):

$$IP_{dis\,\mathrm{cov}\,ery} = IP_{hdr} + IP_{payload}$$

$$IP_{payload} = ISCP_{hdr} + \mathrm{Re}\,quest + Sig_{sender}(HMAC(ISCP_{hdr} + \mathrm{Re}\,quest)) + appendList$$

Every SMA node on the path (including the sender) will need to append its capability/policy information if it supports the request. It will randomly generate a secret key and use it to encrypt the capability/policy information and then use the receiver's public key to encrypt the secret key itself.

$$ENC[LocCapInfo] = ENC_{RcvrPubKey}[LocKey] + ENC_{LocKey}[LocCapInfo]$$

The secret key (shared by the local node, the receiver and the sender) can be reused for a period of time. This implies that for the same receiver, we can cache and reuse the $ENC_{RcvrPubKey}[LocKey]$ part without the public key encryption operations.

When the receiver receives this part, it will also compare the bits to decide whether PKS decryption is necessary. In some cases, the receiver can reuse the LocKey that has been decrypted in earlier request.

As we know, every intermediate ISCP node will append its security capability and relevant policy information onto the discovery message one by one. One important attack to this train-like appending list is what we called "cut" attack in which attackers can simply cut off several sections from the tail of the appending train unnoticeably. Authentication and signature which can maintain the integrity and authenticity of message segments simply can not deal with the "cut" attack. A checksum algorithm is developed to smartly prevent the "cut" attack from happening as the following:

$$Checksum_i = MD5(Checksum_{i-1}, LocKey_i, IP_{payload})$$

in which $i >= 0, Checksum_0 = 0, Checksum_i$ means the checksum value for the ith node on the path, suppose the sender is the 1st node. $IP_{payload}$ is the non mutable part of current ISCP message.

The checksum is one of the fields in the ISCP common header. Upon receiving the discovery message, each node will calculate its checksum and overwrite the checksum field with the new checksum value. Having all nodes' local keys, the receiver will make the message digest exactly the same way as each node did when it receives the discovery message and compare to verify if the content has been modified. Any attacker in the middle can cut off sections of the messages but fail to recover the old checksum such that it is not able to create correct checksum to pass the receiver's verification.

Furthermore, a message digest will be generated for the encrypted part plus the original packet and signed by the appender. Hence, the appended part looks like:

$$Append = ENC[LocCapInfo] + Sig[MD5[IP_{oldpayload} + ENC[LocCapInfo]]]$$

The appendList is simply the list of Append collected along the way.

All the intermediate routers only append without verification of previous part for performance concern. If we verify every hop during the discovery phase, a potential denial of service threat is enabled by flooding large number of fake request packets such that the verification process takes away a significant amount of the SMA's CPU cycles.

The receiver will finally verify and decrypt each piece of capability information collected on the path. If the whole process is successful, then the receiver will invoke the reservation phase.

Reservation phase: After the receiver decides the optimal pairs of SAs to satisfy the request based on the information collected, it will send a reservation message along the reverse communication path. Each of the intermediate nodes on the path can find an assignment in the reservation message as the following:

$$Asg = ENC_{ownerLocKey}[Assignment] + ENC_{senderKey}[LocKey]$$

Then finally the receiver would sign the message digest of the whole message.

The assignment is encrypted using the owner's local key such that only the right node can decrypt. The sender also needs to know all the intermediate routers' secret keys to ease the communicate with them. The mechanisms to let sender obtain all the key information is to encrypt each node's secret key using the sender's secret key and send it along with each assignment section as above. All intermediate node will ignore this part but the sender which will decrypt all the local secret keys and store them for later communication.

Refreshing message security would deploy the same schemes as the above except the reduced usage of PKS cryptography. The error message, confirmation message and teardown message security is fairly simple. For detail secure message format, please refer to protocol specification.

4.5 Interoperability and Performance consideration

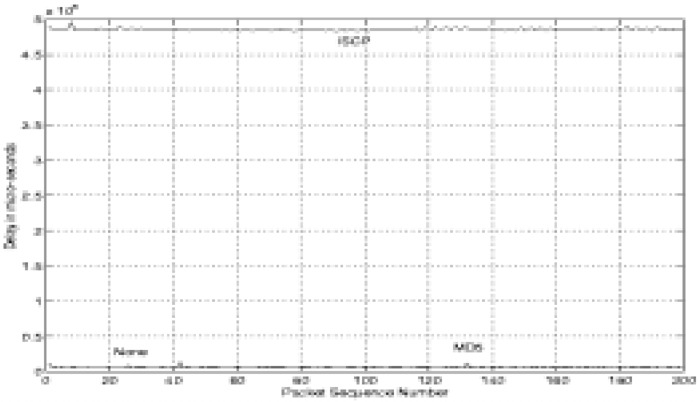
ISCP daemon does not need to be installed in every router in order to provide the satisfying security service. The most relevant part of end-to-end security service provisioning is the border network devices and security gateways/firewalls. Therefore, the domain border network devices and security gateways are most likely equipped with the ISCP and leave other routers/switches as they are. The SMA/ISCP-equipped nodes can coordinate and provide end-to-end security services without relying on any cooperations from those non-participating nodes. For routers not supporting ISCP, the discovery message will be forwarded transparently as normal IP packets.

The scalability problem arises in RSVP from hop-by-hop per-flow state and processing. There is less scalability concern in ISCP although it still exists. First of all, most likely the devices to install ISCP are border routers and security gateways. Most of the interior routers do not need to participate and have no obligation to process any per-flow state. Furthermore, in two phase security context establishment, the first round of ISCP message transmission is the most expensive one for the amount of public key algorithm computation. However, after all the local keys have been set up, the ISCP periodic refreshing will become much cheaper by cached information reuse and fast shared key algorithm employment. The result of our performance testing of several cases is presented in next section.

5. Performance Evaluation

Our performance testing was using one Pentium II machine as the sender and three 386 machines as two intermediate nodes and one receiver. The objective is to compare the performance of ISCP message transmission in three cases: 1) Without any security protection (only transmit plain text) 2) With all local shared key algorithm for security protection 3) With full security protection (exactly what ISCP protocol does for message security and shared key transmission in the first round). Please note that, after the first round, every node can simply reuse cached information or use shared secret key to protect the updated information such that the first and the second case approximate the time cost of refreshing.

We did experiment for each type 200 times and depict result in figure 4. In the figure, the horizontal axial is message sequence number (let the sender issues 200 ISCP messages) and the vertical axial is the time (ms) spent for a complete round



of ISCP message transmission. The curves marked with None, MD5 and ISCP are the time cost of case 1, 2 and 3 respectively. We can see from the figure that the first round is the most expensive while the refreshing is much cheaper to perform.

6. Related work

Policy Based Security Management (PBSM) System: To address the problems of policy and Sas management to provide end-to-end IPSec service, BBN/GTEI proposes and develops the Policy Based Security Management (PBSM)[18] system. The PBSM system focus on the solutions of firewall traversal, policy resolution and SA management within IPSec while Celestial also intends to address the efficient utilization and management of heterogeneous security capability/policy in all protocol layers in addition to those within IPSec to provide end-to-end security service. Because of the different objectives, the protocol processing is also different: PBSM exchanges and merges security policies PS-to-PS (Policy Server) to determine the security context by finding non-nil intersection of the policies while Celestial system collects security capability/policy information all along the path to have receiver make decision based on the collected information.

CiscoAssure: Cisco proposes CiscoAssure[19] policy system to ease the policy management by using LDAPv3 policy server to centrally administer policies. Since LDAP policy server does not address inter-domain policy exchange and negotiation, this solution is good for large distributed enterprise to facilitate security policy resolution and management while Celestial also provides security service for end-to-end communication within different administrative domains. Furthermore, besides various security policy, Celestial system also address heterogeneous security capability management at all protocol layers.

7. Conclusion and Future work

In order to support service management for network security, the Celestial project [9] develops a security service management system to systematically support end-to-end secure communications. ISCP protocol is core protocol within Celestial framework to exchange security capability and policy information among security management agents in each policy domain. ISCP protocol is reliable, scalable, secure and extensible. In this paper, we presented ISCP protocol design as well as performance and evaluation using our ISCP prototype implementation.

As mentioned earlier, the scalability problem in ISCP is not as significant as in RSVP. It is still desirable to minimize it. We are currently working on a more scalable design by aggregating refreshing messages and utilizing the existing SAs, like VPN tunnel etc., to further reduce set up overhead. Furthermore, we are going to extend this model to support multicast transmission by merging the reservation messages. The new version of ISCP design and prototype implementation are underway and will be reported in future publications.

REFERENCE

[1] R. Braden, D. Clark, S. Shenker, Integrated Services in the Internet Architecture: an Overview, RFC 1633, June 1994

[2] S. Kent, R. Atkinson, Security Architecture for the Internet Protocol, RFC 2401, Nov. 1998

[3] S. Kent, R. Atkinson, IP Authentication Header, RFC 2402, Nov. 1998

[4] S. Kent, R. Atkinson, IP Encapsulating Security Payload, RFC 2406, Nov. 1998

[5] A. Freier, P. Karlton, and P. Kocher, The SSL Protocol, Version 3.0, Nov. 1996, http://www.db.opengroup.org/sib.htm#SSL3

[6] T. Dierks, C. Allen, The TLS Protocol, Version 1.0, RFC 2246, Jan. 1999

[7] D. Stevenson, N. Hillery, G. Byrd, Secure Communications in ATM networks, Communications of the ACM, 38(2):45-52, Feb 1995

[8] ATM CellCase product information is available from http://www.celotek.com

[9] Fengmin Gong, Y. Frank Jou, Chandru Sargor, S. Felix Wu, Architecture design of the Celestial security management system, Jan. 28, 1998

[10] D. Maughan, M. Schertler, M. Schneider, J. Turner, Internet Security Association and Key Management Protocol (ISAKMP), RFC 2408, Nov 1998

[11] P. Srisuresh, L.A. Sanchez, Policy Framework for IP Security, Internet Draft, <draft-ietf-IPSec-policy-framework-00.txt>, February 1999

[12] M.C. Richardson, Authenticated Firewall Traversal with IPSec, Internet Draft, April 1996.

[13] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, Resource Reservation Protocol (RSVP), RFC 2205, Proposed Standard, September 1997

[14] F. Baker, B. Lindall, and M. Talwar. RSVP Cryptographic Authentication, Internet Draft, November 1998, Network Working Group.

[15] Tsung-li Wu, S. Felix Wu, Zhi Fu, He Huang, Fengmin Gong, Securing QoS: Threats to RSVP Messages and their Countermeasures, IWQoS'99

[16] M. Blaze, J. Feigenbaum, J. Lacy , Decentralized Trust Management, in Proceedings of the 17th Symposium on Security and Privacy, IEEE Computer Society Press, Los Alamitos, 1996, pp. 164-173.

[17] M. Condell, C. Lynn, J. Zao, Security Policy Specification Language, Internet draft, October 1997

[18] L.A. Sanchez, M.N. Condell, Security Policy System, Internet draft, Nov. 98

[19] Delivering End-to-End Security in Policy-Based Networks, Cisco paper, http://www.cisco.com/warp/public/cc/cisco/mkt/enm/cap/tech/deesp_wp.htm