Policy-based Hybrid Management Architecture for IP-based VPN

Seung-Jin Baek, Moon-Sang Jeong, and Jong-Tae Park
School of Electronic and Electrical Engineering
Kyungpook National University
{sjbaek, msjeong}@ain.knu.ac.kr, park@ee.knu.ac.kr

Abstract

Using IP-based Virtual Private Network (IP-VPN) technology, a company can establish its virtual private network over public networks. Recently, IP-VPN emerges as one of main technologies for increasing business competitiveness with reasonable costs. The standards for IP-VPN are currently being developed by Internet Engineering Task Forces (IETF) and others. In this paper, we identify the limitations of current standard of IETF for the global policy management. We propose a new policy-based hybrid management architecture for IP-VPN services which makes a use of a centralized global management server. A global poilcy is classified into a set of levels depending on the administrator's authority. The procedures for automatically verifying, decorrelating, and resolving the global policies have been designed for the various policy levels by extending *mntner* class and *policy* class of Security Policy Specification Language (SPSL). Finally, we have built a prototype of VPN service management system to demonstrate the functionality.

1. Introduction

Due to the recent explosive use of the Internet, Intranet and Extranet are widely being deployed in order to facilitate the communication and group working activities among business partners and customers. There is an increasing need to extend networks to specific exterior.

Virtual Private Network (VPN) is simply defined as the 'emulation of a private wide area network (WAN) facility using IP facilities' [1]. In IP-VPN, the VPN service is provided over the Internet by establishing a virtual communication link between end-points using various tunneling protocols such as L2TP and IPSec [2,3]. As VPN service becomes widely popular, the need for efficient VPN service management technology becomes very important [4]. In VPN service management, the policy-based management scheme is being employed.

There are research efforts for VPN management including VPN management in ATM networks [5,6], and others [7,8,9]. There have also have been several research efforts on policy-based management including works on the managed object modeling, formalization of policy representation, and policy-based management architecture [10,11,12]. However, there have been few research works on IP-VPN management. Recently, IETF proposes a few standards for policy-based IP-VPN security management. These include Internet Drafts on Security Policy Specification Language, Security Policy Protocol, Security Policy Data Model, and Security Policy System [13,14,15,16,17].

The architecture of the security policy system described in [16] is a fully distributed system, interworks various policy servers which may be placed at different internet domains. However, the existing system has some difficulties in introducing the global policy concept because it was designed without any consideration on that. Its capability is limited to the adaptation and distribution of security policies between local policy servers. In its architecture, each policy sever is responsible for the adaptation and distribution of security policies. However, it is difficult for a VPN service provider or a VPN administrator to enforce and maintain dynamically the global policy. In other words, even if one of the local policy administrators changes the existing policy database in such a way that the local policy violates the global policy, there is no mechanisms provides to check this violation, verify the consistency, and maintain the integrity of the global policy throughout the entire security policy system. There is another problem in the architecture in [16] with respect to performance and efficiency. When we construct another VPN within a VPN, the policy negotiation procedure proposed in [16] may be too complex for some cases.

In order to eliminate these limitations mentioned above, we propose a new policy-based hybrid security management architecture for IP-VPN in which a centralized management server is located along with distributed local policy servers proposed by IETF. The centralized security policy manager maintains and enforces the global policies, and provides several management functions including verification, decorrelation and resolution functions. Finally, we have designed the prototype of VPN service management system using CORBA technology in which the global security policy can be automatically enforced into the distributed local servers.

This paper is organized as follows. Section 2 describes the limitations of current IETF standards, and in Section 3, we present the policy-based hybrid security management system and policy verification procedure. Section 4 describes generic VPN service management system using CORBA technology. Finally, we conclude in Section 5.

2. Limitation of Current Standards for the Management of Global Policy

Researches related to security policy are in progress by IETF IPSec Working Group. IETF has defined Security Policy Specification Language (SPSL) [13] to describe security policies, and Security Policy Protocol (SPP) [14] to distribute policies. SPP enables policy information to be exchanged between a policy server and a client.

Servers refer to local policy database to provide policy information to clients. When a server can not determine the authority of specific host or gateway, it requests information exchange to other policy servers. The security policy system is a fully distributed system without any centrally monitoring or controlling components. However, the architecture of security policy system has some difficulties in introducing the global policy concept as described below.

The limitation of IETF-proposed architecture is a lack of consistent and automated policy enforcement or modification mechanism. When service providers make global policies that cover the whole provider's network, he/she should perform a policy update process to every local policy server which is related to new policies. In addition, the local administrator has all authorities on the local policy management in distributed environments. For example, for the VPN shown in Figure 1, a global VPN administrator can enforce the global policies on each local policy server (PS) at the initialization step of a local PS. However, after the VPN starts to operate, the VPN administrator has no way to monitor and control the local policies that are created or modified by the local administrator because the administrator can set up only initial policies. The only way to control is that the VPN administrator accesses the entire policy information of local servers.

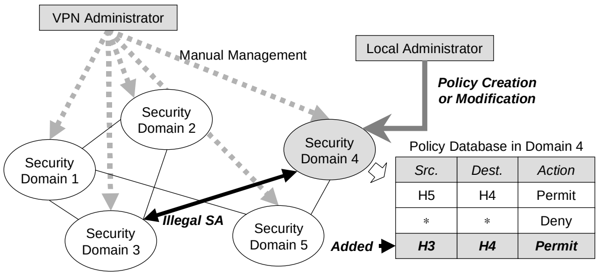


Figure 1: Illegal Policy Modificaiton in Local Policy Servers

Let's suppose that the policy database of domain 4 is initialized by VPN administrator as shown in Figure 1. If the local administrator of domain 4 creates a new policy or modifies the existing policies which may be violating the rules of the global policies, domain 4 can make security associations with domain 3. Physically, this does not cause any problems. However, from the view of service management, this security association can not be permitted. Moreover, if domain 4 should be protected in the VPN, illegal policy modification can cause some problems in security aspects. SPS proposed by IETF does not provide ways to inhibit this situation to occur. A VPN administrator should have mechanisms to detect and correct this problem automatically. We solve this problem by placing a another central global policy-based management system to which could access dynamically the local policy information stored in each local SPS database, and control each local server to enforce the global policies.

The second problem may occur in the policy negotiation procedures between multiple domains. Figure 2 depicts the negotiation procedure when host 1 and host 3 make security associations. As a user wants to communicate between host 1 and host 3, host 1 sends a policy query message to PS1 to know whether it is possible or not. PS1 has its own local policies only, and it can not determine the authority to host 3. Therefore, PS1 sends a query messages to PS2 through SG1, SG2, and PS2 send it to PS3. PS3 deliver the policy information of host 3 to PS2. Finally, host 1 receives it from PS1 and PS2. In this procedure, PS1 and PS2 should update their own SPD (Security Policy Database) in accordance with the policy response message.

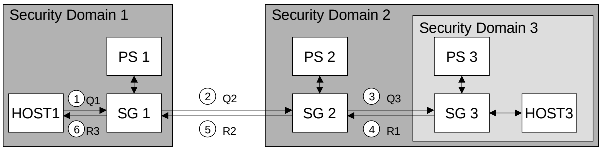


Figure 2: Security Policy Negotiation in Multiple Domains

For the case of Figure 2, if we ignore the transactions between policy servers and security gateways, the total number of policy query/response messages is 6. The total number of messages exchanged between the two end-points is proportional to the number of gateways. If we take into account the times to perform transactions to databases, the total delay may be intolerable due to inefficient and complex negotiation procedure.

3. Policy-based Hybrid Security Management System

Policies describe management activities ranging from lower-level management operations to higher level objectives that are determined by economical or social requirements. Policy is defined as “information which influences the behavior of managers and managed objects” in [10]. A policy determines the management objectives, the roles, and the responsibilities between a subject set containing managers and a target set containing managed objects. The manager applications can be made responsible for the enforcement of policies, i.e. they have to interpret policies and to align their behavior with the policies.

When the managed range is relatively small, the administrator can use manual technique. However, in a large-scale network, it is necessary to provide an automated service management based on policies

3.1. Design of Hybrid Security Policy System

In this section, we present a new hybrid security policy system in order to eliminate the limitations imposed by the existing security policy system proposed by IETF. The hybrid security policy system plays a role of generic policy system. It distributes global policies which are set up by VPN administrator, and verifies the consistency of the policies which are created or modified by local administrators. Figure 3 shows the overall structure of proposed security policy system, its components, and operations. Each component shown in Figure 4 has the functions as described below.

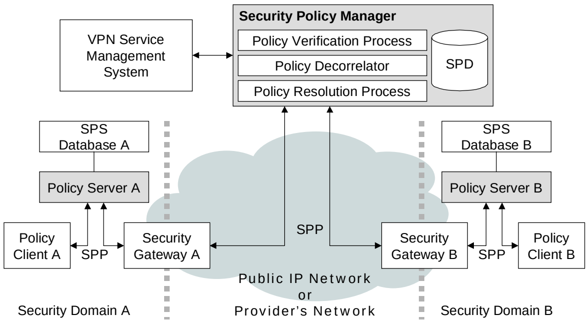


Figure 3: Interworking between Policy Server and Security Policy Manager

- **Security Policy Manager (SPM)**

In this system, Security Policy Manager (SPM) has two major roles as follows. SPM transfers not only policy to Security Policy Server (PS), but also response for policy query that PS has asked. It has global SPD to retain policy information of the entire network. The existing PSs communicates information with SPM, which does with the individual server by means of SPP, in order to exchange policy information. In other words, from the Policy Server’s point of view, it only differs communication object but any other change. In addition to this, SPM only needs to retain location information of

Security Policy Manager, instead of having location information of other Security Gateway.

SPM performs verification, decorrelation, and policy resolution process about a newly created and/or modified policy and stores the results at SPD. It checks whether the local policy is consistent with the global policy through verification process. Policy decorrelation and policy resolution process is performed to add in the previous policy as well. Algorithm and procedures of decorrelation and resolution process are described in [16]. SPM can interoperate with VPN service management system for automatic provisioning and management of VPN service, which is described in subsequent Section 4.1.

- **Security Policy Database (SPD)**

In security policy system, every security domain must maintain a database containing the policy information. There are two kinds of databases. One is the global policy database and the other is the local policy database. The local policy database contains all the policies for the security domain. It is populated with information coming from the master file of the security domain. The incoming new policy is merged with those in SPD by using the policy resolution process. The global policy database also contains all the policies that is related to the entire provider's networks. The policies in the global policy database are set up by VPN administrator either automatically by using automated NM tools or manually. IETF standardizes the format of the database, security policy specification language (SPSL).

- **Security Policy Server (PS)**

The security policy server plays a role of enforcing local policies into the security gateways. Policy servers exchange policy information with policy clients using security policy protocol. In the architecture proposed by IETF, PS communicates with other PSs to check the authority for the specific host. However, in our system, servers interacts with not other PSs, but SPM. Policies which are made by local administrator must be verified by SPM through the policy verification process.

- **Security Policy Protocol (SPP)**

Policy clients and servers exchange information using the security policy protocol. The protocol defines how the policy information is exchanged, processed, and protected by clients and servers. The protocol also defines what policy information is exchanged and the format used to encode the information. The protocol specifies six different message types used to exchange policy information.

3.2. Policy Verification Process

SPM inspects the requested policy through policy verification process. To achieve this function, this process may include user authentication and verification of the authority on the request

In SPSL, the *mntner* class defines objects that can create, delete, and replace SPSL objects., and all policies have *mnt-by* attributes to specify the authorized user on the policy. The *mnt-by* attribute is a list of *mntner* class. If a new policy is created, mnt-by attribute should contain the value of all the authorized users. From the view of service provider, he or she must restrict the customers within the scope of the contract. On the other hand, customers may change their services within the scope of the contract. If a customer endows all the end-users the authority on a new policy, the length of the *mntner-list* of the policy may be very large. In this case, the various policy levels and user classes are needed.

- **Extension in Security Policy Specification Language**

In our system, we extend some parts of SPSL in order to verify the authority.

```
Mntner:     <object-name>
Char-set:   <char-set>
Notes:      <free-form>
Auth:       <scheme-id> <auth-info>
Address:    <free-form>
Phone:      <phone-number>
Fax-no:     <phone-number>
Email:      <email-address>
Mnt-by:     list of <mntner-name>
Certs:      list of <cert-name>
Changed:    <mntner-name> <date>
Signature:  <mntner-name> <cert-name>
            <signature-alg>   <signature-
            data>
domain:     list of <domain-name>
```

Figure 4: Extension of *mntner* class

We have extended *mntner* class defined by IETF as shown in Figure 4. We add the *domain* attribute to indicate where users are. We define the value of this attribute as list of domain-name, because administrators may participate in multiple domains.

```
Policy-name:    <object-name>
Char-set:       <char-set>
Notes:          <free-form>
Association:    <node-name>
Cache-expiry:   <integer>
Policy:
Mnt-by:         list of <mntner-name>
Changed:        <mntner-name> <date>
Signature:
Mnt-level:      <integer>
```

Figure 5: Extension of *policy* class

We have also extended *policy* class defined by IETF as shown in Figure 5. We add the *mnt-level* attributes to indicate the authority level on the policy.

- **Policy Verification Algorithm**

Policy verification process verifies the authority of users as follows.

(1) Authenticate user identity.
(2) Get the authorized domain list in *mntner* class
(3) Get the user *mnt-level* form Policy Authority Table
(4) Break up the policy C into the smaller sets $C_1 \times C_2 \times \cdots \times C_k$, as described below, such that there is no loss of policy information.

$C = C_1 \times C_2 \times \cdots \times C_k$

C : Policy that is requested by users

C_j : The *j*th policy in C

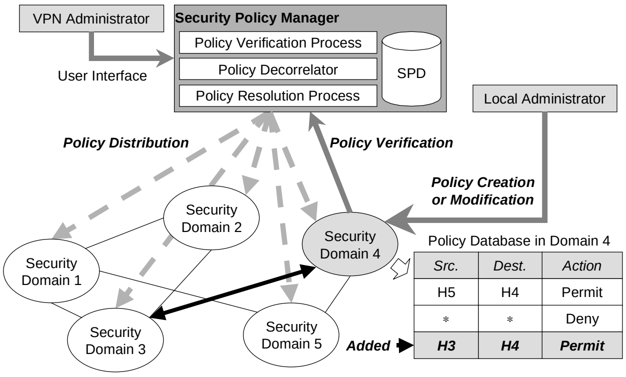
(5) Compare every C_j with policies in SPD.
(6) Send SPP-denied message to the user when a user is found to have no authority. This means user *mnt-level* is lower than that of the existing policy.
(7) Send SPP-accepted message to the user when a user has the authority. The new policy is added to SPD through the decorrelation and resolution process, and its *mnt-level* value is set up to the same as that of user's *mnt-level*.

Procedures or algorithms for resolution and decorrelation are described in [16].

3.3. Operations in Security Policy Manager

In this section, we will the operations of security policy system in policy-based management. In the architecture of IETF, there are no automated mechanisms for global policy distribution. The existing system reflects the local policies which are defined in the master files of local SPDs. For the enforcement of global policies, the VPN administrator must change the master files, and the contents of all local PSs that are correspond to the policy.

SPM distributes the global policies to each local PS in the proposed architecture. The VPN administrator can make and modify the global policies. Policies that are made by the VPN administrator are stored in SPD through the policy decorrelation process module in SPM. Policies stored in SPD are distributed to the local servers using SPP. It is possible that the global policies are enforced in VPN management. In our system, a global VPN administrator can enforce the global policies to local servers automatically as follows.



Policy Database in Domain 4

Src.	*Dest.*	*Action*
H5	H4	Permit
*	*	Deny
H3	***H4***	***Permit***

Figure 6: Policy Verification Process

(1) A VPN administrator makes a new global security policy.
(2) The policy is delivered to SPM by either manual manner or using visualized tools.
(3) The received policy is decorrelated and merged with existing policies.
(4) The policy is stored in SPD.
(5) SPM sends SPP policy messages to related policy servers.
(6) Local policy servers merge a newly received message with their own policies through resolution process.

(7) Finally, the new policy is enforced at local policy servers.

As we described in Section 2, local administrator can make or modify the policies. In Figure 6, creation/modification processes are performed in our system under administrator’s control. In our system, all policy change must obtain the approval of SPM. SPM replies to the policy modification request according to the contents of SPD in SPM. Therefore, policy changes in local PSs can be done within the scope of the contract. In the proposed system, SPM participates in the policy negotiation process.

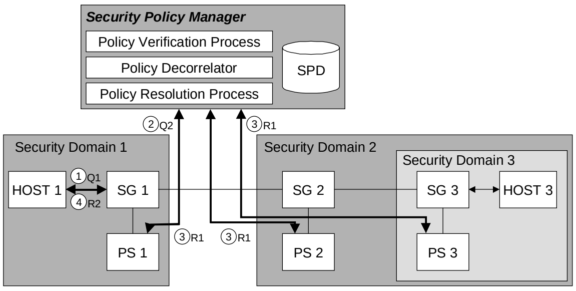


Figure 7: Security Policy Negotiation using Security Policy Manager

When VPN is composed of as shown in Figure 7, policy negotiation procedure between host 1 and host 3 is done as follows.

(1) User application attempts to send a message from host1 to host 3.
(2) IPSec on host 1 gets the packet and finds a policy for it in the SPD.
(3) Host 1 sends a policy query message to PS1 to determine whether it is possible to establish security association with host 3.
(4) As PS1 can not determine the authority on host 3, PS1 request policy information that is related to host 3 from SPM.
(5) SPM sends the information to all servers which are on the communication path, and every server should update their local database in accordance with the response.
(6) Finally, host 1 receives the response from PS1.

If we ignore the detailed transactions between policy servers and security gateways, the total number of policy query/response messages is six. As the number of the security gateways which participate in the negotiation procedures becomes N, the total number of policy messages becomes 2+N. And the number of procedure steps is 4. Although the number of security gateways increases, the number of steps in the execution of transactions is constant. This shows that the proposed architecture performs in more efficiently than SPS of IETF does.

4. Policy-based Service Provisioning : An Application Example

In this section, we show an architecture of CORBA-based VPN service management system based on security policy management which is built by extending our previous work [5,18]. The procedure of end-to-end management based on security policy for end-to-end environments are described.

4.1. Architecture of VPN Service Management System

Figure 8 shows the architecture of VPN service management system in which security policy manager is embedded. The management system is composed of customer network management (CNM) system, end-to-end management system, gateway for supporting the interworking between network infrastructure and management system, modules for business management, and security policy manager for supporting the global security policy.

Security policy manager applies security policy to services. It determines whether new and existing services observe the policies stored in each local policy servers. It provides several functions such as verification, decorrelation and resolution, for the management of local policy. End-to-end service management system is a module of managing services between end-points that related to VPN service. Management functions of this module include service creation, maintenance, deletion, and so on. Management system needs authentication system for the only authorized users to use the system. Authentication manager module executes user authentication procedure according to the authentication table in the management system. The details on CORBA-based service management system are out of the scope of

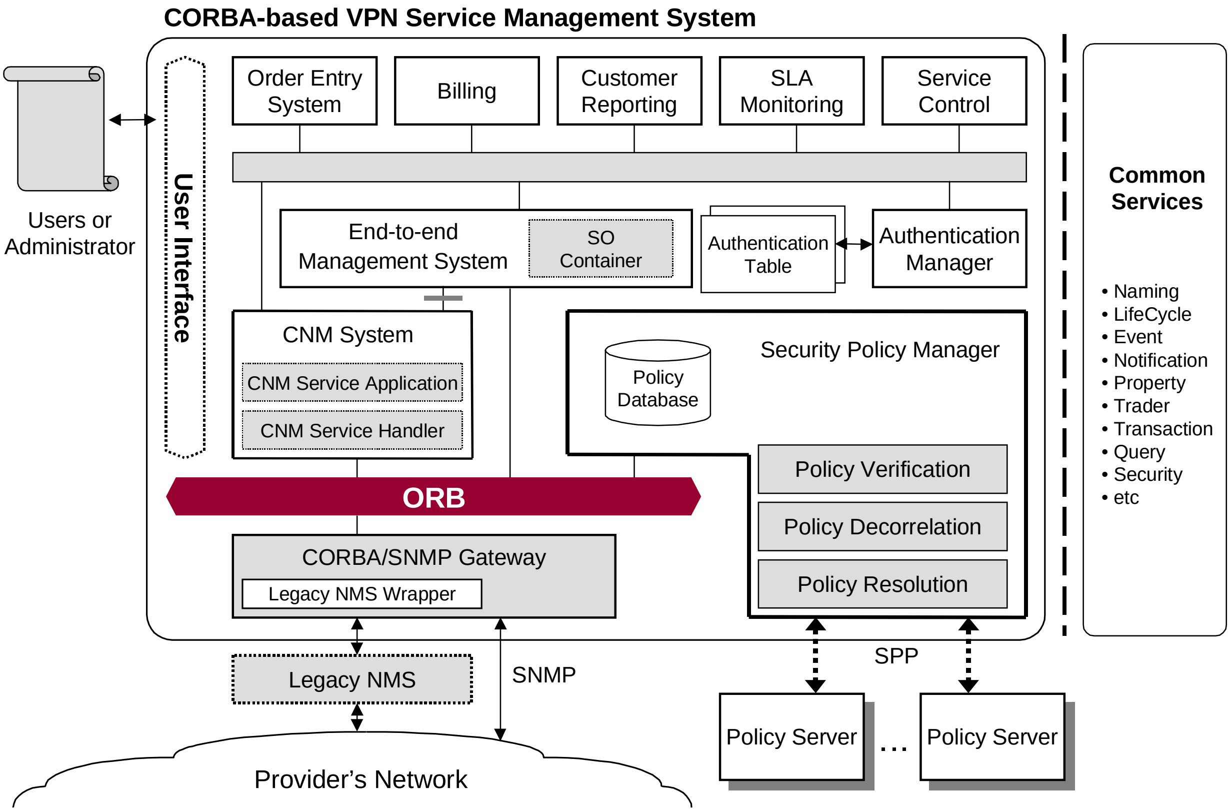


Figure 8: Architecture of CORBA-based VPN service management system

this paper, but can be obtained by referring to the paper [5,18].

4.2. Automated Service Provisioning based on Security Policy

In order to provide end-to-end VPN services, VPN service management system should have a function for automated service provisioning. End-to-end manager performs functions of service creation, maintenance, modification, and so on. End-to-end management system creates an object for a created service and puts it to service object container. When customer requests a service creation, this module accepts the request if the service creation does not violate the security policies. The service object factory creates a new object instance according to the accepted request. Service object is composed of service related parameters – VPN ID, Customer ID, Endpoint address, Tunnel Mode, Encryption Algorithm, and QoS parameters. Service object requests CNM system to reflect its own information to real network resources at service creation/modification. Service object container is a kind of repository that stores service objects. Service object container has service object factory as well as service objects, so that end-to-end management system can perform operations like creation, modification, detection, and deletion.

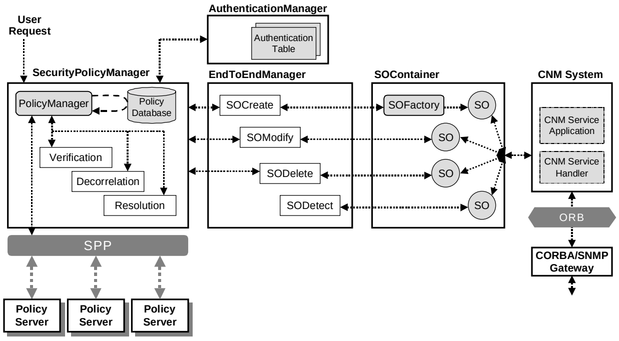


Figure 7: End-to-end Management based on Security Policy

Figure 7 presents the interactions for end-to-end management among authentication manager, end-to-end manager, service object container, and security policy manager. The interactions among components are described below in detail.

(1) Local policy server sends SPP policy query message.

(2) SPM verifies the authentication of the received message using authentication manager

(3) Using verification process, SPM inspects the validity of the policy which is requested by local policy server.

(4) SPM sends SPP response message.

(5) If SPM permits PS's request, the new policy is stored in policy database through decorrelation and resolution process.

(6) SPM sends SPP-policy messages to the related policy servers in local.

(7) SPM invokes SOCreate method in end-to-end manager.

(8) SOFactory creates new service object, and newly created SO is stored in SOContainer.

(9) SO can be used in CNM system, Billing, Accounting.

5. Conclusion

In this paper, we have identified the limitations of current standard of IETF for the global policy management. We have proposed a new policy-based hybrid management architecture for IP-VPN services which makes a use of a centralized global management server. The security policy manager enforces global policies, and provides the management of local policy servers with several functions, which are necessary for the verification, decorrelation, and resolution. We have designed the global policy verification process and algorithms by extending *mntner* class and *policy* class of SPSL for the various policy levels. For an illustrating example, the VPN service management system has been built, and provides end-to-end service management functionality that is based on security policies by using end-to-end manager and security policy manager. The system simplifies the service provisioning procedure and policy negotiation procedure, and achieves more efficient management performance. In the future, we will evaluate the performance and scalability of the proposed system.

[References]

[1] Bryan Gleeson, Arthur Lin, Juha Heinanen, Grenville Armitage, and Andrew Malis, "A framework for IP based Virtual Private Networks," Internet Draft, 1999.

[2] W. M. Townsley, A. Valencia, A. Rubens, G. S. Pall, G. Zorn, and B. Palter, "Layer Two Tunneling Protocol L2TP," Internet Draft, 1999.

[3] R. Thayer, N. Doraswamy, and R. Glenn, "IP Security Document Roadmap," RFC 2411, November 1998.

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

[5] J. T. Park, J. H. Lee, J. W. Hong, Y. M. Kim, and S. M. Kim, "A VPN Management Architecture for Supporting CNM Services in ATM Networks," Proceeding of the IEEE/IFIP International Symposium on Integrated Network Management, May 1997, pp.44-57.

[6] Isabelle Hamchaoui and Fabrice Guillemin, "Resource Management in Virtual Private Networks over ATM," Proceedings of the IEEE ATM'97 Workshop, May 1997.

[7] M.C.Chan, A.A.Lazer, and R.Stadler, "Customer Management and Control of Broadband VPN Services," Proceedings of the Fifth IFIP/IEEE International Symposium on Integrated Network Management, May 1997.

[8] Eun Chul Kim, Choong Seon Hong, and Joong Goo Song, "The Multi-layer VPN Management Architecture," Proceedings of the Sixth IFIP/IEEE International Symposium on Integrated Network Management, May 1999.

[9] Walfrey Ng, Andrew Do-Sung Jun, HungKei Keith Chow, Rouf Boutaba, and Alberto Leon-Garcia, "MIBlets:A Practical Approach to Virtual Private Network Management," Proceedings of the Sixth IFIP/IEEE International Symposium on Integrated Network Management, May 1999.

[10] B. Alpers and H. Plansky, "Concepts and Application of Policy-Based Management," Proceeding of the IEEE/IFIP International Symposium on Integrated Network Management, 1995, pp. 57-68

[11] Stephen Howard, Hanan Lutifyya, Michael Katchaban, and Michael Baurer, "Supporting Dynamic Policy Change Using CORBA System Management Facilities," Proceeding of the 5th IFIP/IEEE International Symposium on Integrated Network Management, May 1997.

[12] Burkhard Alpers, Herbert Planksy, and Rianer Sauerwein, "Applying Domain and Policy Concepts to Customer Network Management," ISS'95-International Switching Symposium, World Telecommunications Congress-Proceedings - Volume 2, April 1995.

[13] M. Condell, C. Lynn, and J. Zao, "Security Policy Specification Language," Internet Draft, July 1999.

[14] L.A. Sanchez and M.N. Condell, "Security Policy Protocol,", Internet Draft, July 1999.

[15] R. Pereira and P.Bhattacharya, "IPSec Policy Data Model," Internet Draft, February 1998.

[16] L. A. Sanchez and M. N. Condell, "Security Policy System," Internet Draft, November 1998.

[17] S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol," RFC 2401, November 1998.

[18] J. W. Baek, T. J. Ha, J. T. Park, J. W. Hong, and S. B. Kim, "ATM Customer Network Management Using WWW and CORBA Technologies," Proceeding of the IEEE/IFIP Network Operations and Management Symposium, New Orleans, Louisiana, February 1998, pp.120-129.

백승진

1998.2 경북대학교 전자공학 학사

1998 – 현재 경북대학교 전자공학 석사과정

관심분야: 가상사설망, 웹기반 관리, 인터넷관리, CORBA

정문상

1998.2 경북대학교 전자공학 학사

1998 – 현재 경북대학교 전자공학 석사과정

관심분야: 네트워크 관리, TMN, 인터넷관리, CORBA

박종태

1978 경북대학교 전자공학 학사

1981 서울대학교 전자공학 석사

1981-1987 Univ. of Michigan , 전기 및 전산학 박사

1987 - 1988 미국 AT&T Bell 연구소 연구원

현재 경북대학교 전자전기공학부 부교수
한국통신학회 통신망운용관리연구회 위원장. 정보통신부, 과학기술부 및 한국통신 전문위원. 경북대학교 정보통신학과 학과장. 경북대학교 차세대 정보통신 연구소 소장.

관심분야: TMN, 인터넷관리, 멀티미디어 통신시스템, 이동통신관리