Policy-Based IPsec Management

Man Li, Nokia Research Center

Abstract

Security is vital to the success of e-commerce and many new valued-added IP services. As a consequence, IPsec is an especially important security mechanism in that it provides cryptographic-based protection mechanisms for IP packets. Moreover, in order for IPsec to work properly, security policies that describe how different IP packets are protected must be provisioned on all network elements that offer IPsec protection. Since IPsec policies are quite complex, manually configuring them on individual network elements is inefficient and therefore infeasible for large-scale IPsec deployment. Policy-based IPsec management strives to solve this problem: Policy-based management employs a policy server to manage a network as a whole; it translates business goals or policies into network resource configurations and automates these configurations across multiple different network elements. Policy-based IPsec management significantly simplifies the task of defining, deploying, and maintaining security policies across a network, thereby significantly simplifying large-scale IPsec deployment. This article describes the motivations, key concepts, and recent IETF developments for policy-based IPsec management. It then applies the key concepts to an example of IPsec VPN service provisioning and further describes an example of an IPsec policy server as well as experience gained from implementing such a server. Challenges facing policy-based IPsec management are also discussed.

IP networks play an increasingly important role in our daily life by enabling new approaches to doing business. Transactions that previously were carried out face to face can now be conducted remotely through the Internet. However, new services such as e-commerce and teleconference services will not be widely used unless they have well designed security protections. In other words, security that consists of measures to deter, prevent, detect and correct security violations involved in the transmission and storage of electronic information is of paramount importance to the success of e-commerce and new value-added IP services.

IPsec is a set of Internet Engineering Task Force (IETF) open standards that provides cryptographic-based protection mechanisms for IP packets [1]. The IP layer protections include:

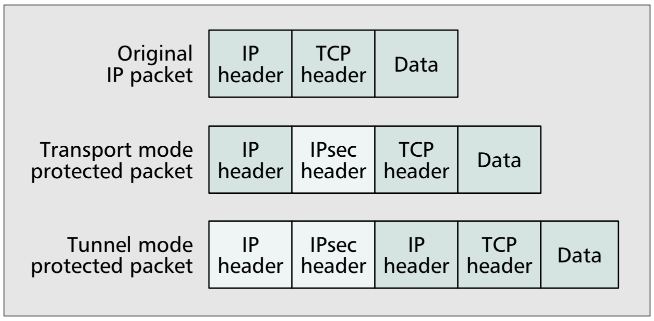
- Packet confidentiality: Packets are encrypted before being sent over the network so that only authorized entities can read them.
- Packet integrity: Packets are protected so that any alterations during transmission over the network can be detected.
- Packet origin authentication: Packets are protected to ensure that they are indeed from the claimed sender whose IP address is contained in the source address of the IP header.
- Protection against replay: Packets are protected from being captured and resent at some later time.

IPsec can be implemented on routers, gateways, hosts, and any electronic appliances where a secure IP connection is required. However, in the remainder of this article, when it is not necessary to distinguish between these physical elements, *IPsec-enabled devices* is used to refer to network elements or devices that implement IPsec.

Compared to firewalls, IPsec-enabled devices provide a much richer set of security protections. A firewall largely provides access control by examining packets and determining which are permitted to pass through; in contrast, an IPsec-enabled device, in addition to access control, also provides confidentiality, integrity, authentication, and replay protection. This richer set of protection concomitantly makes IPsec management more complex than firewall management.

A policy defines a definite goal, course, or method of action to guide and determine present and future decisions [2]. An IPsec policy consists of rules to provide security protections to IP traffic. However, an IPsec-enabled device must be provisioned with appropriate security policy in order for the device to provide essential security protections. At the same time, the rich set features of IPsec protections makes it very complicated to provision a large-scale IPsec policy across many network elements. In such an environment the traditional network-element-oriented approach of configuring one element at a time is impractical. The latter procedure is inefficient and makes it difficult to achieve policy consistency across the network. Policy-based management solves these challenges.

A policy-based management system employs a *policy server* to define, store, and configure policies on multiple network elements. Network elements controlled by a policy server are called *policy targets*. To provision an IPsec policy across a network, an administrator would define the IPsec policy using the policy server. The policy is then stored in the repository of the policy server. The policy may be provisioned at policy targets through either a push from the server or a pull from the policy targets; in either case, the server retrieves the corresponding IPsec policy and translates it into device-specific formats that are comprehensible to the policy targets. The policy server then transports the translated policy to the policy targets, where the targets then install the policy in their local database and accordingly execute the policy.



■ Figure 1. *Transport mode vs. tunnel mode.*

A policy-based management system manages an entire network. It translates business goals or policies into network resource configurations, and then automates these configurations across multiple network elements. Clearly, the use of a centralized server ensures consistent configurations across these network elements. A policy-based management approach simplifies IPsec policy provisioning and shortens the time for new IPsec policy deployment. The remainder of this article examines policy-based IPsec management in detail.

IPsec and IPsec Policy

IPsec Overview

IPsec provides confidentiality, integrity, authentication, and replay protection through the use of two security protocols: Encapsulating Security Payload (ESP) [3] and Authentication Header (AH) [4]. ESP can provide all four protections, whereas AH can provide only three: integrity protection, origin authentication, and replay protection.

Both ESP and AH can operate in two modes [1], *transport mode* and *tunnel mode*, as shown in Fig. 1. Transport mode is typically implemented between two hosts, whereas tunnel mode is customarily implemented between two security gateways, or between a security gateway and a host.

In transport mode an IPsec header is inserted between the IP header and the upper-layer protocol header. As a result, transport mode protects upper-layer protocols only. In contrast, in tunnel mode the entire IP packet is encapsulated into another IP datagram, and an IPsec header is inserted between the outer and inner IP headers. Consequently, tunnel mode protects the entire IP datagram.

One of the most important concepts of IPsec is referred to as a *security association*. A security association (SA) is a simplex "connection" that affords security services to the traffic carried by it [1]. SAs describe security services used in AH or ESP. If both AH and ESP protections must be applied to a traffic stream, two or more SAs are created to provide protection to the traffic stream.

Integrity, authentication, and encryption services require secretly shared cryptographic keys, which can be distributed either manually or dynamically. IPsec specifies an Internet key exchange (IKE) [5] for dynamic key establishment, that is, the ability to establish shared secret keys when they are needed, without the need for manual key distribution. Furthermore, IKE offers two negotiation modes: *main mode* and *aggressive mode*. Aggressive mode is faster than main mode, but not as flexible for key negotiations. Prior to IKE negotiation, the two devices involved in the negotiation must be provisioned with a list of authentication methods, cipher algorithms, and hash algorithms, each having their own preference order. During the IKE negotiation, both parties agree on an authentication method, a cipher algorithm, and a hash algorithm used to generate shared cryptographic keys. The authentication method allows the two parties to authenticate each other and build a trust relationship. After successful IKE negotiation, the two endpoints have the required shared secret cryptographic keys to permit secure negotiations to create IPsec SAs.

Listed below are three possible outcomes when an IPsec-enabled device processes inbound and outbound IP packets:
- Discard: The packet is dropped.
- Bypass: The packet is allowed to pass without further IPsec protection.
- Protect: The packet is enhanced with IPsec protection.

These rules are stored locally in a security policy database (SPD). When an SA is established according to policies in an SPD, it is stored in a local SA database (SADB) until its lifetime expires or it is deleted. In other words, an SPD contains rules that govern the establishment of SAs, whereas an SADB contains information on all currently active SAs.

Perhaps the most popular application for IPsec is in virtual private networks (VPN). In a VPN, IPsec provides protection for IP traffic between geographically disparate sites of a corporation, and for IP traffic between a mobile user and a corporate site. Other IPsec applications include protecting critical signaling and network management IP traffic for mobile networks [6].

IPsec Policy

A policy defines a definite goal, course, or method of action to guide and determine present and future decisions [2]. A policy consists of both conditions and actions. When the policy conditions associated with a policy rule are found to be true, the associated policy actions are applied.

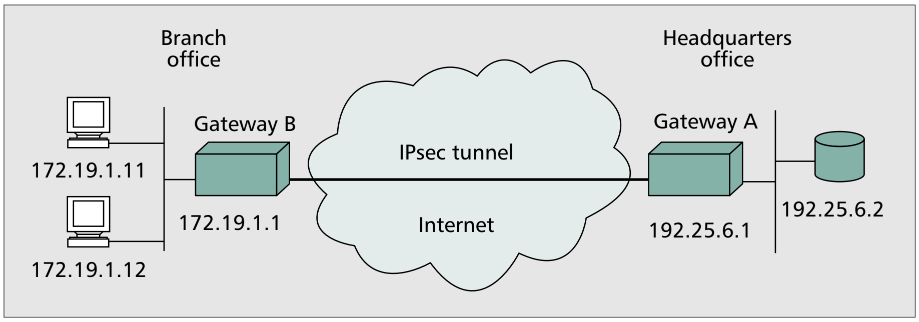
In the context of IPsec management, it can be seen that an IPsec policy consists of rules for providing security protection to IP traffic. These IPsec policy conditions are also called *selectors*, and are stored as access control entries in the SADB. A selector may contain source and destination addresses, protocol, and port numbers that are matched against IP packets. IPsec policy actions may be to discard, bypass, or protect a particular IP communication. If IPsec protection is to be applied, the policy action proceeds according to the SA and IKE specifications. As an example, a policy may call for all traffic originated from a given IP address to be protected by AH in tunnel mode using an integrity protection method based on a key hashing algorithm called SHA-1. A complicated IPsec policy may include multiple nested SAs. An example of an IPsec policy is provided next.

Since IPsec policy governs the establishment of SAs, a correct setup requires that the IPsec policies at both ends of an association be consistent.

The Need for Policy-Based IPsec Management

In explaining the need for policy-based IPsec management, let us begin with an example of VPN provisioning. Though that scenario is presented from the perspective of a service provider, the same challenges apply to organizations that deploy and manage their own large-scale VPNs.

Consider the policies for a service provider offering two types of VPN services: VPN Gold, for data authentication, integrity, and confidentiality; and VPN Silver, which does not provide confidentiality. The Gold service uses IPsec ESP. It provides data integrity and authentication using the cryptographic method known as HMAC-SHA-1, and provides confidentiality using the cryptographically strong encryption method known as 3DES (or "triple DES"). The Silver service uses IPsec AH and, as mentioned above, only provides data



■ Figure 2. *An example of a VPN service.*

integrity and authentication, and uses the cryptographic method HMAC-SHA-1.

Suppose that a company named XYZ has requested VPN Gold service between its headquarters and a remote branch office. In particular, there are two workstations at the branch office, and they need to transfer files to and from a server located at the corporate headquarters. To protect this IP traffic, an IPsec tunnel must be established between the two security gateways at the sites shown in Fig. 2.

Once a service agreement has been signed, the service provider configures the two security gateways with details of selectors, actions, SAs, and key management for Gold VPN service. Accordingly, suppose that the IPsec policies to be configured at the two gateways are as follows:

- At gateway A: Create an IPsec tunnel to gateway B (172.19.1.1) with ESP (3DES, HMAC-SHA-1) to protect the TCP traffic between the two workstations (172.19.1.11, 172.19.1.12) and the server (192.25.6.2).
  For key exchange, establish an IKE tunnel to gateway B (172.19.1.1) with main mode, pre-shared key authentication, 3DES cipher algorithm, and the HMAC-MD5 hash algorithm.
- At gateway B: Similarly, create an IPsec tunnel to gateway A (192.25.6.1) with ESP (3DES, HMAC-SHA-1) to protect the TCP traffic between the two workstations (172.19.1.11, 172.19.1.12) and the server (192.25.6.2).
  For key exchange, establish an IKE tunnel to gateway A (192.25.6.1) with main mode, pre-shared key authentication, 3DES cipher algorithm, and the HMAC-MD5 hash algorithm.

The above IPsec policies for Gold VPN service are written in a simplified and readable format for illustrative purposes. An actual configuration is considerably more complicated. The service provider must configure each gateway with detailed parameters associated with selectors, IKE negations, and IPsec negotiations. These parameters must include, but are not limited to:

- The source addresses, destination addresses, protocol, and port numbers of the traffic to be protected
- The IP address of the peer gateway
- The cipher and integrity algorithms of ESP
- The lifetime of the IPsec security association
- The negotiation mode of IKE
- The cipher, hash, and authentication methods to be used in IKE
- The lifetime of the IKE association

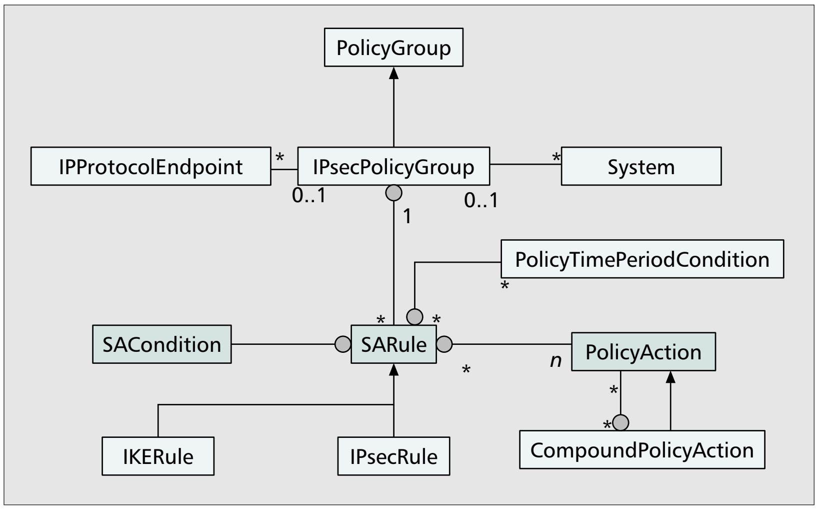
For a single SA in the example above, there is a minimum of 20 parameters to be configured at each gateway. Table 1 lists these parameters for the IPsec policy to be installed at gateway A for outgoing traffic protection. This is a minimal list since in many cases there will be more than one acceptable cryptographic algorithm for protecting data between two sites, and each of the choices must be described in individual *proposals*. Hence, a policy may include a series of proposals and a list of protection algorithms for each proposal. In the example here, there is only one proposal. In addition, the authentication method for IKE in this example is pre-shared key. Each gateway uses a pre-installed secret to authenticate each other. If another authentication method is used (e.g., authentication with digital signature), even more parameters must be configured.

Conditions	**IPsec selector parameters**
	Source IP address = 192.25.6.2 Destination IP address = 172.19.1.11 or 172.19.1.12 Protocol = TCP Source port number = * (matches all destination port number) Destination port number = * (matches all destination port number)
Actions	**IKE negotiation parameters**
	Exchange mode = main mode Cipher algorithm = 3DES Hash algorithm = HMAC-MD5 Authentication method = pre-shared key Diffie-Hellman group = 1 Maximum lifetime in seconds = 1,728,000 Maximum lifetime in kilobytes = 150,000,000
	IPsec negotiation parameters
	Protection mode = tunnel mode IP address of peer gateway = 172.19.1.1 Use perfect forward = true Diffie-Hellman group = 2 Integrity algorithm of ESP = HMAC-SHA-1 Cipher algorithm of ESP = 3DES Maximum lifetime in seconds = 691,200 seconds Maximum lifetime in kilobytes = 200,000,000

■ Table 1. *IPsec policy installed at gateway A for a Gold VPN service*

In order for the IPsec tunnel to work properly, policies at both ends of the tunnel must be consistent. For example, if gateway A is instructed to protect outgoing packets with ESP, while gateway B is instructed that incoming packets from gateway A must be protected with AH, gateway B will drop all packets from gateway A that are ESP protected. As a result, this tunnel fails to be established.

Although tedious, individually configuring IPsec policies on each of the two gateways is possible. But now imagine that the customer requests VPN Gold service between its headquarters and 500 branch offices around the country! Configuring one gateway at a time becomes prohibitively time consuming. To further appreciate the enormity of the task, now contemplate a situation in which there are 20 more corporate customers requesting either VPN Gold or VPN Silver services across the country. This means that there would obviously be a huge number of gateways whose IPsec policies must be provisioned. Besides the expense of individually configuring each, it is also difficult to achieve consistency. A small configuration error at either end of a tunnel will cause dropped packets. Clearly, this is where policy-based management is invaluable.



■ Figure 3. *IPsec information model.*

Policy-Based IPsec Management

A policy-based management system employs a policy server to configure policies on multiple network elements. As mentioned before, the network elements controlled by a policy server are called *policy targets*. In this section we discuss the key concepts of policy-based IPsec management. Later these concepts will be applied to the VPN provisioning example described earlier.

The Concept of Role

Many factors determine the policies to be applied to an interface. As an obvious example, interfaces serving a branch office will almost surely have policies that differ from those serving a headquarters office. On the other hand, policies applied to many different interfaces may be the same. Rather than explicitly specifying policies for each interface of all devices in the network, policies are instead specified in terms of interface functionality. The IETF introduced the concept of *Role* to describe these functionalities. A Role is simply a string associated with an interface [7]. A given interface may have any number of Roles simultaneously. Additionally, a policy has an attribute called a *Role Combination* (i.e., a lexicographically ordered set of Roles). The policy is applied to an interface if and only if the set of Roles in the Role Combination matches the set of Roles of the interface. Thus, Roles provide a level of abstraction for the application of a set of policies to specific interfaces [7]. Examples of Roles might be Company_X_VoIP or Company_X_Branch_Office.

Using the Role concept, an administrator can specify IPsec policies for a Role instead of specifying them for every individual network interface. Similarly, the administrator can modify existing IPsec policies within a Role, instead of modifying them on every individual network interface. In this way, administrators can use Roles to generate network-wide policies.

The Roles for policy target interfaces may be either configured at the time of policy target installation or assigned by a policy server when the policy target first connects to the server. The example later describes the use of Roles in much more detail.

IPsec Policy Information Model

An IPsec policy can take on a variety of forms as it travels from a policy server to policy targets. At each step along the way, the policy needs to be represented in a manner that is consistent and convenient for the current task. As an example, the policy may exist as a schema in a database, as an on-the-wire representation over a transport protocol like the Common Object Policy Service (COPS) [8, 9], or a text-based policy specification language suitable for editing. To ensure consistency, each of these task-specific representations should be derived from a common information model that precisely specifies the content and semantics of the IPsec policy.

A common information model also facilitates interoperability in a multivendor environment. Security devices from different vendors may need to be configured through different protocols or interfaces, such as COPS, Simple Network Management Protocol (SNMP), and Command Line Interface (CLI). Each protocol may require a different on-the-wire representation of the IPsec policy. As long as all the representations are derived from a common information model, policy consistency across multivendor devices is achievable. The remainder of this section describes the IPsec policy information model being developed in the IETF [10].

The IPsec policy information model is an object-oriented representation of the content and semantics of the IPsec policy to be applied to an individual interface. Figure 3 presents part of the model. The following notations are used:

- Boxes represent classes. Attributes of each class are not shown due to space limitations.
- A line that terminates with an arrow denotes inheritance.
- Associations are used to model a relationship between two classes. Classes that share an association are connected by a line.
- A line that begins with an "o" denotes aggregation or containment.
- Next to a line representing an association appears cardinality. Cardinalities indicate the constraints on the number of object instances in a set of relationships. The letter *n*, for example, indicates from 1 to many. An asterisk indicates any number of objects, including zero.

In Fig. 3, the IPProtocolEndpoint and System classes both represent interfaces. The IPsec information model is derived from the PolicyGroup class of the Desktop Management Task Force (DMTF) policy core information model [11]. Through its association with either the IPProtocolEndpoint or System class, the IPsec PolicyGroup represents a set of policies that are used on an interface. Dividing policies into groups provides a convenient way to specify IPsec policies for different user groups.

The SACondition and PolicyAction represent policy conditions and actions, respectively. The class SARule serves as a base class for IKERule and IPsecRule. It defines a common connection point for associations of conditions and actions for both types of rules. The action classes are used to model the different actions an IPsec device may take when the evaluation result of the associated conditions is TRUE. The CompoundPolicyAction class is used to model multiple compound actions (e.g., nested SAs).

The IKE and IPsec rules are stored in the SPD and govern the establishment of SAs. The IKERule associates conditions and actions for IKE negotiations. The IPsecRule associates conditions and actions for IPsec security association negotiations. The SARule (and therefore IKERule and IPsecRule) also has an association with PolicyTimePeriodCondition that specifies the validity period of the rules.

Selectors are derived from the SACondition class. IKE and IPsec actions are derived from the PolicyAction class. These additional details are not shown in the figure.

In an object-oriented approach, the IPsec policy information model presents an IPsec policy as consisting of policy conditions and policy actions. When the policy conditions associated with a policy rule evaluate to TRUE, the associated policy actions are applied.

The DMTF also defines a data model of IPsec policy [11]. It can be seen as a superset of the IETF IPsec configuration model. In addition to configurations, the DMTF also models runtime IPsec information. For example, one can identify all active IPsec tunnels between two IPsec-enabled devices with the DMTF model.

Besides the IETF and DMTF information models, there have been many other efforts to specify security policies. A survey on these various approaches is provided in [12].

IPsec Policy Distribution

The IETF has standardized policy distribution protocols between policy servers and policy targets. The protocols are COPS [8] and COPS for Provisioning (COPS-PR) [9]. COPS is a simple query and response protocol optimized for policy management. It may operate in outsourcing or provisioning mode. In outsourcing mode, when a policy target receives an event that requires a new policy decision, the policy target sends a request message to the policy server. The policy server replies with corresponding policies in real time. In the provisioning mode, a policy server may proactively provision the policy targets, often as a result of administrative inputs. COPS-PR extends the COPS protocol to meet the needs of provisioning operations. An alternative to COPS-PR is the SNMP protocol. The IETF is also working to specify a best current practice for using SNMP to configure policy [13].

Each application, such as IPsec or quality of service (QoS), needs to define its own data structure to be carried by the policy distribution protocols. If the distribution protocol is COPS-PR, the data structure is defined in the format of a policy information base (PIB) [14]; if the distribution protocol is SNMP, the data structure is in the format of a management information base (MIB)[15, 16]. The rest of this section describes the IPsec PIB developed at the IETF [17]. (The IPsec MIB being developed at the IETF contains similar information [18].)

An IPsec PIB consists of approximately 30 tables with references between the tables. The columns of a table contain IPsec parameters that serve a common purpose. For example, the selector table contains selector parameters that include source and destination addresses, source and destination port numbers, and protocol. Each row of the selector table specifies an instance of a selector. There is a unique identifier for every table and each row of that table. The IPsec PIB specification is based on the IPsec policy information model [10]. Consequently, all parameters in the IPsec PIB tables have a correspondence in the IPsec policy information model.

The IPsec PIB specification is additionally tailored to the policy distribution. For that reason, the number of tables is kept small in the IPsec PIB specification, although the IPsec policy is itself quite complex.

Capability Reporting

IPsec-enabled devices from different vendors may have different capabilities (e.g., they may support different sets of encryption algorithms). Some may support nested SAs while others do not. If a policy target implements COPS-PR [9] together with an IPsec PIB [17] and a framework PIB [7] for policy-based management, the policy target can report its capabilities to a policy server via COPS-PR. Based on the reported capabilities and supported Roles, the policy server constructs the policy for the policy target. As a result, an administrator does not need to keep track, in great detail, of the capabilities of each IPsec-enabled device. A policy server can further notify the administrator at the time of policy deployment if a policy target does not support a specified action. Hence, the capability reporting function simplifies policy management, particularly in a multivendor environment.

Unfortunately, for policy targets that do not support capability reporting, as is the case in many current existing products, capabilities associated with each IPsec-enabled device must be manually entered into the policy server by an administrator, who must also make necessary modifications when the software is updated on each policy target.

An Example of Policy-Based IPsec Management

Concepts discussed in the above section are now applied to the VPN provisioning example described earlier. The Roles for the security gateway interfaces may be either configured at installation time or assigned by a policy server when the gateways first connect to the policy server. In either case, this information is stored in the repository of the policy server.

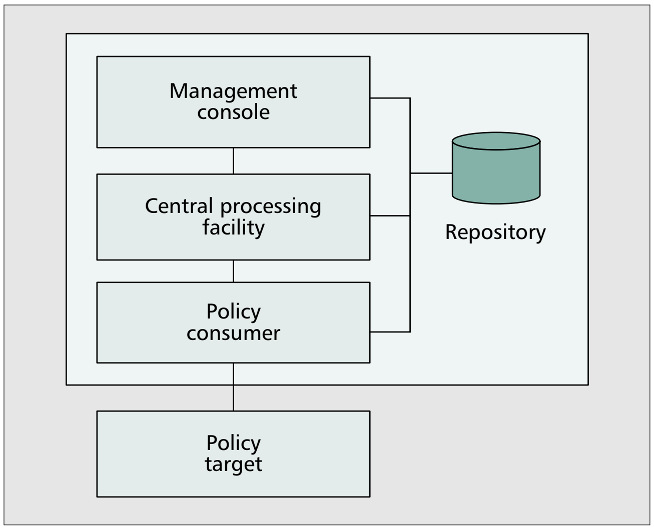
When a service provider plans to roll out VPN Gold (ESP with 3DES, HMAC-SHA-1) and VPN Silver (AH with HMAC-SHA-1) services, the following steps are necessary:

1) The service provider enters the definitions or IPsec parameters associated with VPN Gold and Silver services into a policy server. The policy server stores the service definitions in its repository. The schema of the repository is derived from the IPsec policy information model [10].

2) The service provider also creates a template that will be filled when these services are provisioned for customers. This template is also stored in the repository. The template may include fields for customer name, VPN service (e.g., Gold or Silver) to be offered, VPN service topology, type of traffic to be protected, and the Role names of the gateways on which this policy is to be deployed. The VPN service topology refers to how the security gateways that serve a given customer are to be connected with IPsec tunnels. Examples of VPN service topologies include mesh and hub-and-spoke.

3) When customer XYZ requests VPN Gold service between its headquarters and 500 remote branches, the service provider simply fills in the template to indicate that this customer will be offered VPN Gold service with a hub-and-spoke topology to protect all TCP traffic. In addition, this policy is to be deployed on all gateway interfaces with Role name XYZ_VPN. When the service provider instructs the policy server to deploy the policies, the policy server translates the policies into detailed configurations for the 501 security gateways and configures them automatically.

Behind the scenes, the policy server must first ensure that this new VPN service does not conflict with services already deployed. The policy server then attaches VPN Gold service with its associated IPsec parameters. These parameters have already been stored in the repository, as described in step 1. Next, the policy server looks into its repository to determine the IP addresses of the security gateways that have Role name XYZ_VPN. The policy server then encodes the policies into a proper format, such as that specified by the IPsec PIB. Note that the gateways may be from different vendors. As a result, the policies may need to be formatted and encoded differently for different types of gateways. In this situation, the protocol used for configuration by each gateway is also *a priori* stored in the policy server. The final step is to distribute the policies to the 501 gateways with COPS-PR, SNMP or a proprietary protocol.



■ Figure 4. *A policy server example.*

Now suppose the customer adds a new branch office. After the security gateway at the new branch office is installed, it connects to the policy server to announce its existence, its Role name of XYZ_VPN, and its capabilities (assuming that it supports COPS-PR, IPsec PIB, and framework PIB). After authenticating this gateway, the policy server automatically downloads IPsec policy to the gateway. With the addition of this new security gateway, the policy server also updates the policy of the gateway at the headquarters office. If COPS-PR is used for policy distribution, not all 30-plus IPsec PIB tables need to be loaded again to the headquarters gateway. Instead, since COPS-PR supports incremental updates, only an additional row from each of the five affected IPsec PIB tables needs to be sent. Since each row of a table has a unique identifier, assigning new identifiers to the additional rows indicates that these rows are new additions instead of replacements of existing rows. This update would constitute only a small amount of network traffic.

If a customer wants to change their service from VPN Gold to VPN Silver, the administrator simply modifies the policies on the policy server. All gateways with Role name XYZ_VPN will be downloaded with the updated policies. Any modifications to existing services can be performed in a similar way.

This example demonstrates that a policy-based IPsec management system allows an administrator to define new IPsec services, assign Role names to the interfaces of IPsec-enabled devices, and bind IPsec services to Role names. The policy server translates business security policies into device-specific configurations and automates the configurations on multiple IPsec-enabled devices. This approach is fully scalable because the administrator need not specify IPsec policies for every gateway; nor does he need to manually configure every gateway.

An Example of an IPsec Policy Server

Figure 4 depicts an example of a policy server. The four functional entities are described below.

The **repository** stores service definitions, templates, IPsec policies, and so on. The repository may be implemented with flat files, LDAP directories, or databases. To store IPsec policies for large networks, directories or databases are preferable because data access is faster and schema modification easier with these types of repositories.

The **management console** is a user interface that allows administrators to define new services and enter high-level business policies for customers. Large networks usually have multiple administrators. It is therefore important to allow multiple users to concurrently use a policy server. Some users could remotely connect to the policy server. It may also be useful to assign users with different access rights. For example, some might have read and write access to the repository, while others may only have read access.

The **central processing facility** translates high-level business policies into detailed IPsec policies and stores the translations in the repository. The central processing facility also performs policy verification, conflict detection, and conflict resolution. In its simplest form, the central processing facility translates business policies into IPsec policies. However, when a policy server is positioned to provision many different services, the central processing facility becomes a critical entity. Rule processing capabilities, such as rule verification, conflict detection, and conflict resolution, may become a differentiating factor for future policy servers. An interesting mechanism for IPsec policy rule verification, conflict detection, and resolution is described in [19].

The **policy consumer** acquires policies on behalf of policy targets, adapts the policies into device specific formats and deploys them to policy targets. The policy consumer may be closely coupled with the central processing facility and management console, thus having all functions in one management station. A more scalable approach may be to pack policy consumers into separate physical elements. Each policy consumer is able to connect to, for example, a few hundred IPsec-enabled devices. These policy consumers may be distributed around the network to increase scalability.

The policy target shown in Fig. 4 interprets and executes the IPsec policies.

The policy consumer and policy target execute the functions of the policy decision point and policy enforcement point, respectively, as defined at the IETF [2]. The schema for policies stored in the repository can be derived from the IPsec information model [10].

Because IPsec policies are applied to IPsec-enabled devices, a policy server needs to know the names or IP addresses of all policy targets. Although this information can be manually entered into a policy server, it would be a tedious process, and keeping up with network updates would be challenging at least, perhaps impossible. On the other hand, existing network element management databases already store the information of all network elements. A policy server should be able to import this information from the network management databases, process the information, and store the processed results in its own repository.

The policy distribution protocol used between a policy server and policy targets may be the standardized COPS-PR, SNMP, or a proprietary approach. An example of a proprietary approach is to put IPsec policies into flat files and distribute the files to policy targets through FTP. Another example of a proprietary approach is to distribute policy through a command line interface (CLI). A CLI allows an administrator to connect to a policy target through Telnet, and then type in commands to configure policies. The commands to configure the IPsec policy differ from vendor to vendor.

Experiences with Policy Distribution Protocols

We have developed policy server prototypes that manage IPsec and QoS policies. The architecture of our policy server is shown in Fig. 4. When the policy targets are third-party products, their proprietary CLI is used for policy distribution; otherwise, the policy is distributed through COPS-PR. Two types of policy consumers have been developed, supporting COPS-PR and CLI, respectively.

In order to keep track of the state of a policy target, the policy consumer must process responses from a policy target. We found that it is hard to build a policy consumer that distributes policy through a CLI because a CLI is designed for human use, not for machines. As a result, a CLI does not provide or maintain a well structured state machine, something the human brain can readily do. A policy consumer has difficulty in determining the current state of a policy target: whether the policy target is ready for new commands, has found errors in the command, or has detected failure conditions. In addition, the policy target responds to queries and policy deployments in phrases, which, although ostensibly readable, do not have a standard and well defined structure. Furthermore, although vendors promise backward compatibility of CLI configuration commands, the reply phrases from policy targets may change with new software releases. The lack of states and well structured responses means that the policy consumer must be very intelligent and therefore very complex.

On the other hand, our experience has shown that it is relatively easier to implement a policy consumer with COPS-PR and IPsec PIB. The well defined reporting mechanism and state of COPS-PR, together with well structured IPsec PIB tables, mean that the policy consumer does not need to be excessively complicated.

Challenges Ahead

Policy-based IPsec management is a new field that faces many interesting challenges. Some of these challenges are discussed below, although this is far from an exhaustive list.

Seamless Integration with Other Policy Management — Current policy-based IPsec management systems configure only IPsec policies on policy targets. Similarly, there are policy-based QoS management systems that configure only QoS policies on policy targets. Just as we do not need many dedicated Web servers to store different types of files, we do not need dedicated policy servers to provision different policies. With a single policy server, an administrator should be able to manage many different types of policies. For example, a service provider may want to define a VPN Platinum service as having ESP (3DES, HMAC-SHA-1) IPsec protection and high-priority QoS traffic treatment. In this case, the service provider should be able to enter both security and QoS policies into a single policy server, and the policy server should configure both security and QoS policies simultaneously on multiple policy targets. Reducing the number of management servers reduces operational cost.

When there are multiple policies to be deployed, a policy server must be able to handle dependencies among different types of policies. The policy server must conduct policy verification, rule conflict detection, and rule conflict resolution. An expert system-based rule processing engine is well suited to these tasks. As more and more services are added and policies become more complicated, including rule processing expert systems into policy servers may become inevitable. Rule processing capability may become a differentiation point for future policy servers. Interesting research results for policy conflict detection and resolution are reported in [20, 21].

Multiple Domain IPsec Policy Management — Our discussion so far has been focused on policy-based IPsec management within a single administrative domain (i.e., a domain in which both the policy server and policy targets belong to the same administrator). However, we have not fully investigated how the two corresponding gateways obtain the necessary IPsec policies when two hosts residing in different administrative domains wish to communicate under IPsec protection. For interdomain communications, the policy servers of the two domains may need to negotiate interdomain security policy. The IPsec policy information model described earlier can still serve as a guideline for policy negotiation. An interesting proposal to solve interdomain security policy management can be found in [22].

Seamless Integration with Existing Management and Monitoring Systems — If an administrator already uses one type of database or directory system (e.g., an LDAP directory server), a policy-based management system should provide flexibility to use the same type of database or directory as its repository. This commonality will facilitate data exchange between policy servers and existing network management systems. To facilitate scalability, after necessary processing, essential device-specific information such as IP addresses could be directly imported into a policy server repository from an existing database.

Evolution toward a Standard Policy Distribution Interface — To operate in a multivendor environment, it is essential that both policy servers and policy targets implement standard policy distribution interfaces. To evolve toward universal standard policy distribution interfaces, policy management systems must also support those devices that have already been deployed in the field and do not implement standard interfaces. One solution is to develop a proxy that communicates to a policy server using a standard interface and also communicates to non-IPsec-enabled devices with their proprietary interfaces. The proxy may be a software module that resides on either policy targets or a policy server.

Secure Policy Distribution — IPsec policies contain sensitive information. As a result, policy servers and their policy distribution processes must be secure. A policy server needs to be protected from break-ins. In addition, a policy server needs to authenticate policy targets to ensure that the policy server downloads policies to legitimate network elements. Policy targets in turn need to authenticate the policy server to ensure that the targets obtain policies from a legitimate server. Furthermore, the policy distribution protocol needs to ensure that IPsec policies are not altered during transportation over the network. IPsec or TLS [23], for example, may be used to provide such protections.

In large networks, mutually authenticating policy servers and policy targets, and securing large numbers of policy distribution channels can be very challenging.

Seamless Integration with Trust Management for IPsec — An important aspect of IPsec policy management is trust or credential management for IPsec peers. Although the current IETF IPsec policy management specifications (in particular, the IPsec information model, IPsec PIB, and IPsec MIB) touch on this topic, many questions remain unanswered. How credentials are obtained and used as well as how the public key infrastructure works together with an IPsec policy management system need to be investigated. An interesting discussion on trust management for IPsec can be found in [24].

Conclusions

IPsec offers a rich set of security protections. IPsec policy management is complex, and the proposed policy-based IPsec management simplifies large-scale IPsec policy deployment and management. A policy-based management system employs a policy server to configure policies on policy targets. The Role concept provides a level of abstraction for the application of a set of policies to specific interfaces. An IPsec policy information model is being developed in the IETF to serve as a foundation for different policy representations. An IPsec policy may be distributed to policy targets through IETF standard protocols (COPS-PR, SNMP) or proprietary protocols. The capability reporting function provided through COPS-PR, IPsec PIB, and framework PIB also helps to make policy management in a multivendor environment easier.

We have shown, with an example, how these key concepts of policy-based IPsec management are applied to VPN service provisioning. We also give an example of a policy server. Our implementation experience shows that a well defined policy distribution protocol such as COPS-PR and a well structured policy database such as IPsec PIB make the implementation of a policy server much easier.

As a relatively new area, policy-based IPsec management faces many challenges. Continuous research and standardization efforts are required to meet these challenges.

Acknowledgments

The author thanks the anonymous reviewers for their comments and suggestions. She also thanks Curtis Siller, Jon Graff, and William Papp for their help in writing this article.

References

[1] S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol," IETF RFC 2401, Nov. 1998.
[2] A. Westerinen *et al.*, " Terminology for Policy-Based Management," IETF RFC 3198, Nov. 2001.
[3] S. Kent and R. Atkinson, "IP Encapsulating Security Payload," IETF RFC 2406, Nov. 1998.
[4] S. Kent and R. Atkinson, "IP Authentication Header," IETF RFC 2402, Nov. 1998.
[5] D. Harkins and D. Carrel, " The Internet Key Exchange (IKE)," IETF RFC 2409, Nov. 1998.
[6] 3GPP TS 33.210: "Third Generation Partnership Project; Technical Specification, Group Services and System Aspects; 3G Security; Network Domain Security; IP Network Layer Security."
[7] M. Fine *et al.*, "Framework Policy Information Base," IETF RFC3318, Mar. 2003.
[8] D. Durham *et al.*, "The COPS (Common Open Policy Service) Protocol," IETF RFC 2748, Jan. 2000.
[9] K. Chan *et al.*, "COPS Usage for Policy Provisioning (COPS-PR)," IETF RFC 3084, Mar. 2001.
[10] J. Jason *et al.*, " IPsec Configuration Policy Information Model," IETF draft-ietf-ipsp-config-policy-model-07.txt, work in progress, Mar. 2003.
[11] DMTF: http://www.dmtf.org
[12] M. Sloman and E.C. Lupu, "Security and Management Policy Specification," *IEEE Network*, Mar./Apr. 2002. pp. 10–19.
[13] M. MacFaden *et al.*, "Configuring Networks and Devices With SNMP," IETF RFC 3512, Apr. 2003.
[14] K. McCloghrie *et al.*, "Structure of Policy Provisioning Information (SPPI)," IETF RFC 3159, Aug. 2001.
[15] K. McCloghrie *et al.*, " Structure of Management Information Version 2 (SMIv2)," IETF RFC 2578, Apr. 1999.
[16] D. Perkins, E. McGinnis, *Understanding SNMP MIBs*, Prentice-Hall. 1997.
[17] M. Li *et al.*, "IPsec Policy Information Base," IETF draft-ietf-ipsp-ipsecpib-08.txt, work in progress, May 2003.
[18] M. Baer *et al.*, "IPsec Policy Configuration MIB," IETF draft-ietf-ipsp-ipsec-conf-mib-06.txt, work in progress, Mar. 2003.
[19] Z. Fu *et al.*, "IPsec/VPN Security Policy: Correctness, Conflict Detection, and Resolution," *LNCS*, no. 1995, Jan. 2001, pp. 39–56.
[20] E.C. Lupu and M. Sloman, "Conflict Analysis for Management Policies," *Proc. 5th IFIP/IEEE Int'l. Symp. Integrated Net. Mgmt.*, 1997, pp. 430–43.
[21] E.C. Lupu and M. Sloman, "Conflicts in Policy-Based Distributed Systems Management," *IEEE Trans. Software Eng.*, vol. 25, no. 6, Nov./Dec. 1999, pp. 852–69.
[22] J. Zao *et al.*, "Domain Based Internet Security Policy Management," *Proc. DARPA Info. Conf. Expo.*, vol. 1, 1999, pp. 41–53.
[23] T. Dierks and C. Allen, "The TLS Protocol," IETF RFC2246, Jan. 1999.
[24] M. Blaze *et al.*, "Trust Management for IPsec," *Net. and Distrib. Sys. Security Symp.*, Feb. 2001.

Biography

MAN LI (man.m.li@nokia.com) is a principal research engineer in the Nokia Research Center, Burlington, Massachusetts. She received Ph.D. and Master's degrees from the University of Ottawa, Canada, in 1991 and 1987, and a Bachelor's degree from Beijing University of Aeronautics and Astronautics in 1984; all in electrical engineering. Before joining Nokia in 2000, Man Li also worked at Verizon Laboratories and Nortel Technology respectively. Her current research interest includes end-to-end policy and service management in 3G mobile networks.