# Access Control and Policy Enforcement in Internetworks

by

Gene Tsudik

A Dissertation Presented to the

FACULTY OF THE GRADUATE SCHOOL

UNIVERSITY OF SOUTHERN CALIFORNIA

In Partial Fulfillment of the

Requirements for the Degree

DOCTOR OF PHILOSOPHY

(Computer Science)

April 1991

# Dedication

To *R&D* and 179 steps of wisdom.

## Acknowledgements

This is one of the most difficult sections of this thesis as there are so many people I would like to thank.

First and foremost, I can only scratch the surface of my deepest gratitude to my advisor, Dr. Deborah Estrin, whose guidance, friendship and piquant sense of humor I enjoyed during my years at USC. Our innumerable meetings, discussions and (at times heated) debates at various exotic locations have contributed greatly to this dissertation and my intellectual maturity. Most importantly, she is responsible for convincing me not to view research as a sequence of sprints, and to use analogies sparingly.

I would also like to thank other members of my dissertation committee, Drs. Silvester and McLeod for their helpful comments and for enduring my ramblings, both written and spoken.

I sincerely thank all former and current members of the Computer Networks and Distributed Systems Lab for maintaining a stimulating and friendly research milieu. I am indebted to Lee Breslau and Steve Hotz for comments on this thesis and its presentation. Kamal Anand, Danny Mitzel and Ron Cocchi have contributed to various stages of *Visa* protocol implementation, testing and experiments. I am also grateful to fellow IDPR-ers: Lee Breslau, Tony Li and Katia Obraczka with whom I enjoyed having many a brainstorming session. Special thanks to Debbie Galtman for *conquering* DESNC. I am also indebted to Sharon Anderson who is, tragically, no longer with us.

I am thankful to all members of the Internet Open Routing Working Group who contributed to the design and implementation of IDPR. In particular, I would like to thank Martha Steenstrup and Helen Bowns at BBN, and Robert (Woody) Woodburn at SAIC.

Last, but certainly not least, I wish to thank my family. My parents have always been an endless source of warmth, understanding and support. My wife, Rima, is impossible to thank properly in this context; any expression of gratitude will be an understatement. Finally, my daughter, Daniela, has been instrumental to this dissertation by being the chief motivation for its eventual completion, and by staying exceptionally quiet in difficult moments.

# Contents

# Abstract

A collection of independent Administrative Domains (ADs) can be joined together in an internetwork in order to support inter-organizational communication and resource sharing. Despite the interconnection, ADs are concerned with maintaining varying degrees of autonomy by exerting local control over their network resources. Each AD specifies its own network **policy** that represents a tradeoff between autonomy and interdependence. However, existing internetworking approaches attempt to achieve full connectivity without much regard to policy.

This research represents the first broad-scale treatment of policy enforcement in internetwork environments. We explore the range of potential policies and consider policy enforcement in the context of end-point and transit ADs. A well-known *end-to-end* argument is applied to provide a comprehensive framework for the placement and composition of access controls to support policy enforcement. For end-point ADs, we propose fine-grained enforcement provided by *Visa* protocol which controls the flow of packet traffic at AD boundaries. In order to control *transit* internetwork traffic, policy support must be integrated into internetwork routing and packet-forwarding protocols. Secure protocols for the Inter-Domain Policy Routing architecture are presented.

The central theme of protocol security is addressed throughout. In order to minimize the performance impact of typically costly security services, innovative cryptographic protocols are introduced and the cost of security is evaluated in detail. Furthermore, methods for strong, yet inexpensive, encryption-free data integrity and authentication are presented.

Prototype implementations of the proposed policy enforcement mechanisms have been developed. We discuss implementation issues and performance results.

# Chapter 1

# Introduction

## 1.1 Overview

Increasing use of computers for communication has prompted widespread interconnection of autonomous networks. In an environment of interconnected Administrative Domains (ADs), access to network resources is an issue of growing concern. In the absence of special mechanisms, network interconnection using existing internetworking protocols (e.g., IP [73] or OSI [43]) attempts to achieve full connectivity. However, ADs should be able to interconnect without exposing their internal resources to unrestricted external access [24, 27]. Moreover, internetwork components should be able to control incoming and outgoing traffic by specifying or constraining the ADs to, and through, which the traffic can flow [26].

While complete autonomy implies no interconnection, increased *"openness"* sacrifices autonomy. Thus, each participant organization must reach its own tradeoff between autonomy and interdependence. The particulars of this tradeoff are embodied in what we refer to as **policy**.

The purpose of this thesis is to present the design of a policy enforcement architecture for an environment of interconnected ADs. In this chapter, we introduce some basic concepts necessary for the appreciation of the underlying problem and propose a framework for the subsequent design of policy enforcement mechanisms. In Chapter 2, we discuss current policy enforcement approaches and a number of support mechanisms to be used as basic building blocks in our design. Chapter 3 is dedicated to the treatment of access control at end-point AD boundaries and Chapter 4 addresses the control of transit internetwork traffic. Chapter 5 describes the experimental results obtained from the implementations of the mechanisms proposed in Chapters 3 and 4. Finally, Chapter 6 summarizes the results of this thesis and discusses topics for future research.

### 1.1.1 Organization of This Chapter

This chapter is organized as follows.[1] We begin in Section 1.2 by addressing the internetwork environment and exploring the range of network policies that an organization might wish to express. In Section 1.3, we identify three objects of internetwork access control:[2] end-systems, network-layer resources, and internetwork routes. We then consider the well-known *end-to-end* argument for the placement of controls in network layer protocols [78]. Section 1.4 describes the security services needed to address each of these requirements and the corresponding design choices of enforcement location, protocol, and granularity. Section 1.6 concludes this chapter with an overview of the rest of the thesis.

## 1.2 Interconnection of Autonomous Networks

In order to provide appropriate background for the subsequent discussion, this section defines our terminology and assumptions regarding internetwork environments, policies, and protocol design principles.

[1] Portions of this chapter appeared in [30].

[2] Throughout this thesis, the terms *access control* and *policy enforcement* are used interchangeably.

### 1.2.1 Administrative Domains

In the context of this thesis, an internetwork is composed of a number of Administrative Domains, or ADs. An AD is defined as a collection of network resources under control of a single administrative entity [26]. We distinguish between two types of ADs: *stub* and *transit*. *Stub* ADs are interested mainly in communication with other stub ADs, i.e., providing communication for their constituent end-systems. A campus network is an example of a stub AD. *Transit* ADs provide communication service (i.e., bandwidth and switching) for stub AD traffic. Finally, there are also hybrid ADs that combine transit service with end-system communication.

### 1.2.2 Policies

As frequently happens with a new concept, an analogy can lead to better understanding of the problem at hand. We can view ADs as sovereign countries, each with a specific set of foreign policy statements regarding interaction with foreign entities (other ADs). For example, a country may have policies restricting foreign visitors to specific areas or restricting travel privileges of the local populace when visiting foreign countries. Countries may also have specific policies pertaining to transit travelers, e.g., restricting entry on the basis of the traveler's itinerary. Security policies regarding international travel can express policy with regard to passport and visa requirements, length of stay, etc. Accounting or billing policies may concern, for example, visa fees or departure taxes.

ADs can express similar policies regarding communication with external entities, e.g., restrict internal systems available for external access or restrict external systems available for internal access. Transit traffic may or may not be allowed, or it may be restricted to specific source, destination ADs or end-systems. Policies can also embody security requirements, e.g., authentication and authorization for inter-AD traffic, as well as accounting and billing conditions [26].

Network level policies are primarily concerned with unauthorized access to resources, denial of service, and inappropriate accrual of communication-related charges. These threats can all come about through attacks on the authenticity and the integrity of internetwork packet traffic. Some concerns are of greater importance to stub networks and others, to transit networks.

#### 1.2.2.1 Stub and Transit Policies

Due largely to the nature of service provided, stub and transit ADs tend to express different policies. Most policies expressed by stub ADs protect internal resources from external access, while those expressed by transit ADs tend to be cost-related. Another way of making this distinction is to observe that transit ADs, by virtue of providing transit service, are inherently more *open* than their stub counterparts. Furthermore, subversion of transit AD's policies will, in the worst case, result in denial of communication services, whereas subversion of stub network polices can potentially disrupt the end-systems themselves. Another reason for separating the respective policies is the difference in accounting and billing requirements. Stub ADs are more likely to bundle communication costs into billing for end services, if any such billing occurs. Transit ADs are more likely to charge for the communication itself. Finally, stub AD policies include route selection criteria, which dictate how the AD's packets travel to their destinations.

In some respects, the requirements for transit policy enforcement are simpler than those for stub policy enforcement. However, several factors complicate the implementation of the latter. First, in an internetwork, a packet may travel through a number of transit ADs on its way to the destination. Consequently, applicable policies from all transit ADs must be considered when a packet is being sent; whereas for control of stub resources, only the policies of the two end-point ADs need to be taken into account. In addition, transit control has to be reconciled with topology changes (routers or links going down). If in the middle of a connection any component of the route becomes disabled, entirely different policies may come into effect. Also, when a transit AD decides to account or charge for resource usage, coordination is required to pass charges back to the end points. Moreover, stub AD route selection criteria must be integrated with transit control policies to determine the appropriate routes. These factors add to the complexity of potential enforcement mechanisms.

Based in part on the difference in policies, and in part on the functionality required in any routing (i.e., transit) mechanism, transit and stub AD *mechanisms* also differ. By analogy with international travel, in most countries transit travelers are set apart from other visitors. They are issued special *transit* visas and are restricted in movement and length of stay. We discuss transit mechanisms further in later sections.

#### 1.2.2.2 Policy Attributes

Policies can be based upon a number of attributes:

- **Endpoint** policies place restrictions on the source and/or destination of traffic.
  Example: *[No traffic to/from AD X is accepted]*
- **Path** policies place restrictions on other ADs of the path in addition to the source and destination ADs.
  Example: *[Transit traffic must enter/exit through AD Y]*
- **Security** attributes express requirements for authentication, data integrity, replay detection and privacy.
  Example: *[All incoming traffic must be encrypted]*
- **Temporal parameters** include restrictions on usage based on time of day, day of the week or other time-related parameters.
  Example: *[Traffic from AD X is only accepted between midnight and 6 am]*
- **Type of Service (ToS)** policies discriminate according to the service parameters (e.g., delay, throughput) made available to different users.
  Example: *[High-bandwidth, low-delay traffic is not handled]*
- **Accounting/Billing** policies express conditions related to charging and accounting.
  Example: *[Transit service is charged for on per packet basis]*

A typical policy statement can be based upon several policy attributes. For example, the policy statement below applies to transit traffic and combines ToS, temporal and accounting/billing attributes:

*[Priority transit traffic from $AD_a$ is accepted between 2 and 6 am with a per packet charge]*

Further examples of policy types can be found in [26].

#### 1.2.2.3 Problematic Policies

Policy types discussed thus far involve static attributes and are deterministic in nature, i.e., a policy either permits or prohibits communication between a set of entities. Policies can also be based upon highly dynamic parameters such as current load or link availability. Such policies are known as *non-deterministic* policies. For example, $AD_a$ may have a policy to carry transit traffic as long as it does not interfere with local communication. Or, $AD_a$ will carry transit traffic as long as it consumes less than, say, 30% of $AD_a$'s total bandwidth. The ubiquity of these policies is that an AD can express conditional policy statements based on the constantly changing state of the network. The difficulty is that, outside of the AD that expresses non-deterministic policies, it is generally impossible to determine whether a particular policy permits or prohibits communication at a given point in time.

### 1.2.3 Internetwork Topology

Some routing protocols place restrictions on internet scale and topology, e.g., EGP [77]. Any inter-AD routing protocol should have the potential of supporting very large scale internetworking. We anticipate on the order of $10^5$ ADs.[3] In an internet of such enormous size, it would be impractical to design a protocol that relied on topological restrictions; enforcement would be near impossible. Consequently, one of our design goals is to allow for maximum degree of flexibility in regard to the configuration of the internetwork. The protocols discussed below do not place restrictions on the internetwork topology.

Figure 1.1 depicts an example of AD interconnection topology. It resembles a traditional three-level hierarchy of long haul, regional and stub ADs. However, there are exceptions to the hierarchy in the form of lateral and bypass links. These exceptions to the otherwise regular topology are not dispensable and must be supported, perhaps at the expense of routing protocol overhead. Absence of restrictions on AD interconnection allows us to accommodate this, or any other, topology that may evolve in future internetworks.[4]

[3] Although the majority will be stub ADs, our model assumes a large number of transit and hybrid ADs as well.
[4] For further discussion of internetwork topology see [26].

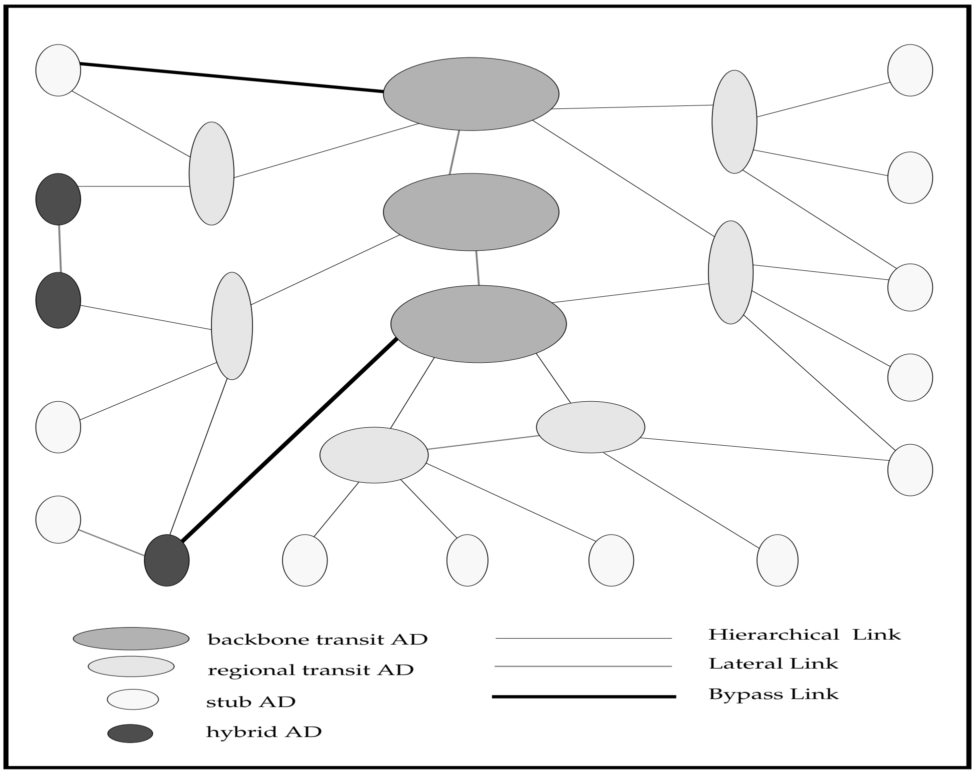


Figure 1.1: Example of AD interconnection.

## 1.3 Access Control Requirements

This section addresses access control requirements for all objects of policy. It begins with a brief discussion of end-system control in an inter-AD context. We then argue for controlling access to AD network resources independently of end-system access control.

### 1.3.1 End-systems and Applications

End-system security is a requirement for all stub ADs. Previous work has addressed the design of secure applications, operating systems, as well as the adaptation of secure systems to a network context [89, 51, 90, 89, 46, 27].

Modern distributed operating systems, e.g., Andrew [80] and Amoeba [64], illustrate methods for efficient implementation of security features in a distributed computing environment with high availability of services. In the realm of secure applications, Privacy-Enhanced Electronic Mail [51], for example, provides valuable insights pertaining to the implementation of security services in a large-scale distributed environment with a very large and volatile user population.

We emphasize end-system and application-level controls as a point of comparison for the other controls addressed, and comment on the division of labor between end-system and network controls. But first, we briefly discuss how inter-AD connections influence the requirements for end-system controls themselves.

Inter-AD connections impact ADs that have relatively open internal computing environments more so than ADs with closed, protective internal environments. For an AD that employs rigorous security mechanisms on *all* end-systems, introduction of an inter-AD link need not induce significant modification of the end-systems themselves. However, many ADs employ a more *laissez-faire* approach to security–implementing stringent controls on only a small subset of critical systems and leaving the balance of end-systems relatively vulnerable to intra-AD access. This should not be viewed as a criticism or failing of intra-AD security. Rather, to the extent that controls may inhibit internal resource sharing, it may be in the organization's interest to maintain a relatively open environment. Introducing additional controls throughout an AD is often impractical because of the large number of internal end-systems and the subsequent difficulties in verifying the correct operation of these controls. Nevertheless, introduction of inter-AD links may require some reassessment of this approach.

In addition, some communication-oriented applications will require special consideration. Among them are: electronic mail, video conferencing, file sharing and process migration. Each demands a unique combination of security-related services. In video conferencing, for example, privacy is often important while data integrity is somewhat secondary; also, bandwidth availability is critical. Hence, denial of service (malicious or otherwise) is of significant concern.

### 1.3.2 Network Resources

Many discussions of *network security* are actually discussions of end-system protection in a network environment, e.g., [85, 69, 48, 46, 27, 90]. While this is an important consideration, we claim that it is not adequate in the multi-AD context. For both stub and transit ADs, there are valuable network resources that are also the object of policy.

This is in agreement with a well-known design principle, the *end-to-end* argument [78]. It states that the placement of controls should be in the highest protocol layer at the end-points of communication. From a functional standpoint, features such as reliability and security must take place in the highest layers if they are to cover all sources of vulnerability. Therefore, the argument is that lower-layer efforts are always redundant and should be implemented only to the extent that they improve efficiency. In the case of security, the argument suggests that end-system resources are best protected by the end-systems themselves, e.g., security services should be provided in the transport layer as opposed to the network or data link layer [90]. We observe, however, that in the sense of the *end-to-end* argument, network resources are *endpoints* to the extent they require protection in their own right. Therefore, it is imperative to address the protection of network resource in addition to the more traditional end-system protection. This implies the need for controls at the AD exit and entry points.

If control is left to the end-systems, valuable stub-AD network resources may be consumed by unauthorized traffic. Rejecting packets at the end-system is *too late* from the perspective of network resource usage. Moreover, unrestricted network access increases the vulnerability of ADs to denial of service attacks in the form of packet storms. In other words, the network interfaces of end-systems are themselves network resources and should be subject to access control (which can only be achieved below the transport layer). Similarly, some ADs have no relevant end-systems (they provide transit services only) and therefore must implement desired controls in the network-layer routing protocol. Since routing is a network layer function, these controls must also involve network level entities and can not be left to transport session endpoints.

Network interconnection is typically done at the network layer for reasons of transparency, flexibility and performance. Consequently, transport and higher layers are generally not well-suited for the implementation of network resource controls. Moreover, lower layers (physical and data-link) do not provide access to subject and object information necessary to make policy decisions with respect to network resources. This leads to our assertion that the network layer is most appropriate for the implementation of network resource controls.

In summary, to the extent network resources require protection, the highest relevant endpoint is the network router and associated packet forwarding and routing protocols. In this sense, the *end-to-end* argument supports implementing these controls at the network layer.

#### 1.3.2.1 AD Boundaries

An AD represents a set of resources that are governed by common policies. We argue that the enforcement of policies pertaining to inter-AD communication is best carried out at the AD boundaries.[5] Throughout our discussion we will exploit the AD abstraction to decouple requirements and mechanisms for intra-AD communication from those for inter-AD communication. In particular, we assume network layer enforcement only in those routers that connect the AD to other ADs, i.e., border routers (also referred to as Policy Gateways [50]).

Another motivating factor for implementing controls at border routers is the trend towards commercialization of portions of the Internet. Some network resources may be charged for on a usage-sensitive basis. This opens new

[5]Policies that apply to intra-AD communication can be enforced by each AD independently.

incentives and opportunities for fraud. Consequently, border routers must be equipped to detect such abuse before valuable network resources are consumed and charges are accrued.

In the following sections, we address such security concerns in the context of both stub and transit ADs.

#### 1.3.2.2 Stub ADs

Stub ADs need to protect the integrity of their internal network in the presence of inter-AD connections. Network resources that may require protection include links, bridges, routers, and end-system network interfaces. In the following paragraphs we further justify the need to control access to communication resources themselves, and the need to include end-system network interfaces among the resources being protected.

Most internal networks are implemented without explicit access controls in the nodes or routers. In the context of intra-AD use, limitations on access to private information or resources applies primarily to end-systems. Communication is most often treated as a *free good* within organizations. It is neither charged for, nor controlled on a usage-sensitive basis. Most often, internal network access is unrestricted if a user has legitimate end-system access. Even within the AD this can lead to undesired dependencies among hosts, departments, etc; for example, misbehaving hosts generating broadcast storms, or poor transport protocol implementations degrading performance of a bottleneck resource. However, in most cases, the existence of a common administrative umbrella alleviates the need for intra-AD control of communication resource usage.

When an inter-AD connection is first established, it may violate some of the assumptions under which these intra-AD protection decisions were made. There is no longer a common administration, nor the common organizational goals that can be assumed in the intra-AD context. In addition, there is no control over implementation and configuration of end-systems and routers in other ADs. The number and nature of the potential users increases qualitatively in the presence of an inter-AD link. In this context, the communication resources can no longer be considered *free* to all potential users. For this reason, stub AD network resources themselves must be considered as objects of access control policy in a multi-AD internet.

The external connection in a stub AD usually exists for the purposes of a small subset of internal end-systems [24]. It may be undesirable to expose all internal end-systems. Furthermore, as discussed in section 1.3.1, it is impractical to assume that all end-systems in every AD will implement adequate defenses to be considered secure in the face of a greatly expanded and diversified user community (e.g., a global internetwork).[6] Therefore, ADs require mechanisms that allow them to designate those end-systems that will be reachable from the outside, and those that will not. Similarly, ADs may wish to restrict origination of outgoing traffic to those end-systems that are explicitly authorized, e.g., to restrict the usage of pay-per-packet transit services or to reduce the risk of undesired information exportation. Hereafter, we refer to the externally-accessible (reachable) end-systems as *equipped*, and the rest as *unequipped* end-systems.

Access to unequipped end-systems is treated as a network-level access control problem. In other words, the end-system network interfaces are viewed as network resources that require protection. By controlling access to end-system interfaces, ADs can also address the susceptibility of all end-systems to denial of service attacks through flooding [27].

Finally, even communication with equipped end-systems must be controlled at the network layer. Each time an unauthorized external user attempts to communicate with an equipped end-system, stub AD network resources are consumed. Even if the end-system knows to reject the transaction, it is too late with respect to the expended network resources.

In summary, if a stub AD wishes to control usage of its internal network resources (links, bridges, routers, and end-system network interfaces) it can not rely solely on the protection mechanisms in end-systems.

#### 1.3.2.3 Transit ADs

Transit ADs are concerned with controlling usage of, and access to, their internal routers and links. Their policies may be based upon the source or destination AD, the previous or next hop AD, or other characteristics such as user classes, charge codes, or type of service [26]. Transit resources may be billed on a usage-sensitive basis. In addition, service quality is dependent upon adequate capacity to meet demand. Consequently, control of access to these resources is critical to their operation.

[6] We consider physical isolation of reachable end-systems from strictly internal end-systems to be an overly restrictive solution. For most environments, it would infringe on internal communication and integration to an unacceptable extent.

As with stub ADs, transit resource protection can not be left to end-systems. By the time traffic reaches the end-systems, the communication resources would have been used. Moreover, in the case of transit ADs, the destination end-system is not within the particular transit AD's administrative control and can not be expected to recover the transit AD's costs. Furthermore, transit ADs should not rely on stub ADs to enforce transit policies; this represents an excessive compromise of transit ADs' autonomy. Even if transit ADs are open to all paying customers, they need to monitor and charge for traffic. Monitoring and charging are just different types of network-layer control mechanisms. In any case, network layer controls are needed at the boundaries of transit ADs to protect against unauthorized resource usage and denial of service attacks.

### 1.3.3 Route selection

The last policy requirement faced in the multi-AD context is control of route selection. Different routes have different characteristics. For example, some routes may travel via ADs that charge on a usage sensitive basis. Other ADs may be avoided because they are not trusted. Stub ADs may wish to express policies regarding the particulars of their traffic flows, i.e., where a packet can be sent, how it should get there, and which internal systems can originate it. Similarly, some ADs will want to enforce policies regarding the routes taken by incoming traffic. Control of transit AD network resources and route selection are similar to the extent that they both restrict access to transit resources. However, route selection is based on preferences of stub ADs, whereas transit ADs protect their resources on the basis of local policies.

Therefore, a stub AD may need to control access to inter-AD routes according to their characteristics. Not all internal traffic sources will be given external access. Even those sources allowed external access may not be permitted to use routes that incur usage-sensitive charges. Still others may be permitted to transmit only over routes composed of highly trusted ADs. Some AD destinations may even be considered off limits to all outgoing traffic. These policies can not be enforced by transport layer protocols as routing is transparent to them. Hence, enforcement must take place at the level of internetwork routing and packet forwarding protocols.

## 1.4 Design Choices

This section addresses the design of enforcement mechanisms to meet all three control requirements: end-system, network resource, and route selection. It begins with the discussion of applicable security services and proceeds to discuss appropriate enforcement locations, enforcement protocols, and granularity of principals. In the final subsection the object granularity and mode of enforcement are discussed. The results of our discussion are summarized in Section 1.4.6.

### 1.4.1 Security Services

The OSI model specifies fourteen security services [44]. (They are summarized in Table 1.1). These services, however, are not uniformly applicable to all types of objects:

1. End-systems
   Due to the variety of applications and services provided, end-systems are subject to a wide range of potential security threats. To counter these threats, end-systems may require all or most of the (fourteen) OSI security services.

2. Network resources

   (a) Stub ADs
   In order to prevent unauthorized access to their internal network resources and unauthorized export of traffic across AD boundaries, stub ADs need to enforce access control policies. This entails the authentication of principals (peers) involved. Furthermore, since a compromised communication channel between an authorized pair of principals can lead to improper usage of stub AD's internal network resources (i.e., compromised data still consumes stub AD network resources *en route*), data origin authentication and data integrity must be maintained as well. Non-repudiation of origin may be needed if an AD needs to account for usage of network resources. On the other hand, confidentiality of the end-to-end data is logically a function of higher protocol layers in end-systems.

| No. | ISO Security Service | Applicability |
|---|---|---|
| 1 | Peer Entity Authentication | Y |
| 2 | Data Origin Authentication | Y |
| 3 | Access Control Service | Y |
| 4 | Connection Confidentiality | N |
| 5 | Connectionless Confidentiality | N |
| 6 | Selective Field Confidentiality | N |
| 7 | Traffic Flow Confidentiality | N |
| 8 | Connection Integrity with Recovery | N |
| 9 | Connection Integrity without Recovery | N |
| 10 | Selective Field Connection Integrity | N |
| 11 | Connectionless Integrity | Y |
| 12 | Selective Field Connectionless Integrity | N |
| 13 | Non-repudiation at Origin | Y |
| 14 | Non-repudiation at Delivery | ? |

**Legend:**
Y – applicable
N – not applicable
? – potentially applicable

Table 1.1: Applicability of ISO Security Services

(b) Transit ADs
Because transit AD network resources are similar to those of stub ADs (with the exception of end-systems) they are subject to the same security threats, and, hence, require much the same security services. As discussed later, differences arise in other characteristics of policy enforcement.

3. Route selection policies can be thought of as access control restrictions with respect to internetwork routes. This requires authentication of the principals requesting access to routes. Furthermore, it also requires authentication and integrity of routing information provided by the transit ADs (because this routing information is used to associate cost and security characteristics with computed routes).

In summary, we are primarily concerned with five security services: access control, peer entity authentication, data origin authentication, data integrity and non-repudiation. The remaining services may be of concern to some end-systems but, in general, are not affected by the particular issue of AD interconnection. Not surprisingly, the Standard for Interoperable LAN Security (SILS) recommends attention to these same five security services [42]. Moreover, the SILS document points out the inter-dependencies among these five services: access control on authentication and integrity, authentication on integrity, and non-repudiation on authentication and integrity.

### 1.4.2 Enforcement Location

One of the most critical decisions in the design of enforcement mechanisms is their physical location. We address the enforcement of network policies in the context of end-systems, border routers, and specialized servers. Internal AD routers (i.e., those that speak the AD's interior routing protocol) are potentially large in number. Modification, in the form of additional access control mechanisms, raises concerns regarding the cost of implementing and verifying system configurations, and interference of inter-AD protection mechanisms with intra-AD communication. In Section 1.3.2 we justified the placement of controls only in those routers that act as points of connections to other ADs, i.e., the border routers.

In determining the appropriate locations for the enforcement of specific policies, we based our decisions on the principle that, ideally, unauthorized resource usage attempts should be detected before any resources have been consumed, i.e., at the earliest possible point.

1. **End-system resources**: equipped end-systems can be controlled by mechanisms in the transport and application layers. Controls for unequipped end-systems are placed with the stub AD's network resource controls (see next item). This is because no external traffic whatsoever should be allowed to reach these end-systems as they are assumed to be unprotected.

2. **Network resources**: Stub and transit AD network resources are both controlled by mechanisms located in the border routers of the ADs. Border routers may work in conjunction with servers (e.g., authentication and access control servers, policy servers, and route servers) that will also enforce policy. Such servers are needed to offload time and space consuming functions for performance-critical routers, and as a point of coordination and integration of policy.

3. **Route selection**: Stub ADs control access to routes within the route servers that compute external routes and within border routers that may validate traffic.

### 1.4.3 Enforcement Protocol

Closely related to the enforcement location is the enforcement protocol. We consider mechanisms in the end-system transport (and higher layer) protocols, the internetwork packet-forwarding protocol, and the internetwork routing protocol. We distinguish between mechanisms implemented in the intra-AD protocols, and those required in the border router protocols only.

1. End-systems can protect themselves at any (or all) layers in the protocol hierarchy (transport layer being the most applicable). Of course, this only holds for the reachable end-systems that are sufficiently equipped.

2. (a) The unequipped end-systems and the rest of the stub AD network resources are protected by the border routers as discussed above. Since ADs interconnect at the network layer and the network-layer **packet-forwarding** protocol is the highest layer with respect to these resources, it is the most appropriate protocol for the enforcement of access restrictions to unequipped end-systems.

   (b) In order to protect network resources of transit ADs, access controls must also be incorporated into the network-layer **routing protocol**. This is necessary because, unlike stub AD policies, transit AD policies may prevent communication even when a viable route exists. If a stub AD's policy disallows communication, the user may be inconvenienced. However, the policy is having its desired affect. On the other hand, transit AD policies have more far-reaching impact. The shortest route computed by a traditional routing algorithm may not be usable by a particular source due to a policy of one of the transit ADs. Without access to transit policy information, the routing protocol has no means of finding an alternative, perhaps *longer*, route for that source.

   As a result, transit ADs can not simply enforce policy restrictions on a unilateral basis at packet forwarding time. Instead, policies pertaining to transit AD network resources must be either implicit in the topology of an internetwork, or advertised to the anticipated resource users as part of the network-layer routing protocol.[7]

3. Route selection policies are enforced at both border routers and route servers. In route servers, the enforcement protocol is the **routing protocol** application that computes internetwork routes and distributes them to appropriate end-systems. In addition, border routers have to validate route selection made by the route servers at packet forwarding time. The latter function needs to take place in the network-layer packet-forwarding protocol.

### 1.4.4 Principal Granularity

We refer to the subject of a security policy as a *principal* [79] (i.e., a principal is permitted to access, or is restricted from accessing a particular object). Policies may be applied to ADs as a whole, to user classes that are location-independent, to particular end-systems within ADs, or to particular users or user processes. The coarser grain policies (i.e, those based on AD or user class) are easier to manage but, by definition, less precise. Of these policies, ones based on user classes are more difficult to implement, but offer flexibility of grouping users independent of the physical location. The difficulty arises from having to bind a user or user class to lower-level units of communication, i.e., packets. Whereas packets routinely include end-system addresses that can be mapped more easily into AD addresses through existing mechanisms.

---

[7] As argued in [7], it is difficult to reflect a rich set of policies in the topology when the internetwork is of this scale and heterogeneity.

1. For policy enforcement at the end-system level, the choice of principal granularity depends on the protocol layer. Since end-systems can implement controls at several protocol layers, principals of different granularity can be specified. For example, end-systems at the transport layer and application protocols or users at higher layers.

2. The ends-points of inter-AD communication with respect to stub AD network resources are the end-point stub ADs This implies that the principal granularity for stub AD access control should be at the level of ADs or user classes. However, as mentioned before, network resources of stub ADs include the *unequipped* end-systems. To preclude all access to these end-systems, it is necessary to discriminate incoming traffic on the basis of the end-system destination address. Therefore, principal granularity for stub AD network resources may be end-system as well as AD.

3. Since transit ADs provide communication facilities for large numbers of stub ADs, it is impractical for their resource usage policy to specify fine-grained principals such as end-systems.[8] Therefore, the subjects of transit resource policy enforcement are expected to be ADs and user classes.

4. For route selection policies, we must consider the granularities of the two end-point (source and destination) principals. They may be the same (e.g., a single route between a pair of ADs) or they may differ, e.g., a policy may prescribe a specific route for a given {*end-system, destination-AD*} pair. It is expected that most route selection policies will restrict based on source AD (or user class) and destination AD.

### 1.4.5 Communication Granularity and Enforcement Mode

Policies may be applied to communication units of different granularity. In particular, there are tradeoffs associated with implementing controls per packet, per end-system association (i.e., connection), and per AD association. We also consider the appropriateness of using a priori (preventive) or post facto (accounting) detection, of unauthorized resource use, in the context of different policy objects and security services.

1. At the end-systems, the granularity of controlled communication units depends on their reachability. For the reachable end-systems, granularity depends upon the particular enforcement protocol. For example, per-packet at the network layer, and per end-system association at the transport layer. Regardless of the protocol, the mode of enforcement has to be preventive (at real-time) since end-system disruption can not usually be tolerated.

   Access control for unequipped end-systems has to be done on a per packet preventive basis in border routers. This is because even a single packet can disrupt these unprotected end-systems.

2. It is less straight forward to determine the appropriate basis for protection of other stub AD network resources. If reachable end-systems implement controls, and external traffic to unequipped end-systems is prevented at AD boundaries, the only potential for unauthorized use of stub AD network resources is the compromised traffic addressed to reachable end-systems. Such traffic consumes network resources *en route*.[9] However, authenticating the origin and checking data integrity of every incoming packet may be expensive in terms of performance and implementation. Thus, some ADs may elect to allow for some unauthorized resource usage of this sort in return for faster packet switching in border routers, while others may go to the trouble of scrutinizing every packet to assure non-interference of compromised traffic. In summary, for network resources other than end-systems, both real-time and post facto (accounting-based) detection methods can be used.

3. For transit AD network resources, the communication unit granularity is dependent upon the particular security service. It is impractical to control access on the basis of packets or end-system associations due to potentially large numbers of them. It is more manageable to apply enforcement to stub AD associations. Because AD-pair associations are coarse-grained (i.e., can encompass large volumes of packet traffic), establishment of such associations may be verified and validated by intervening transit AD. However, due to the potential risks involved, this type of enforcement should be addressed using preventive methods.

   Once a stub AD-pair association is validated by a transit AD, individual data packets must be associated with a particular AD-pair, e.g., to pass charges appropriately. The implication is that at least some enforcement must take place on per-packet basis. One possible scenario is to enforce certain *inexpensive* controls (based on addressing and/or route) in real-time, (i.e., without expensive data authentication and integrity checks), combined with bulk packet-based accounting to later detect fraudulent (or unaccounted for) resource usage.

---

[8] Transit ADs are also more likely to bundle usage charges on the basis of ADs rather than end-systems.

[9] However, it poses no direct threat to end-systems, as they are sufficiently protected.

In summary, the design choices of object granularity and enforcement mode are more context-dependent than are enforcement location, protocol, or principal granularity.

### 1.4.6 Summary

We conclude this section with the results of the above discussion summarized in Table 1.2.[10]

| Resources | Security services | Enforcement location | Enforcement protocol | Principal granularity |
|---|---|---|---|---|
| End-systems | All | End-systems, Access Control Servers | Transport and above | Any |
| Stub AD Network Resources | 2,3,11,13 | Border Routers, Access Control Servers | Network | End-system, AD, user class |
| Transit AD Network Resources | 2,3,11,13 | Border Routers, Access Control Servers | Routing | AD, user class |
| Route Selection | 2,3 | Border Routers, Route Servers | Routing and Route Computation | Any |

Table 1.2: Policy Enforcement Parameters

## 1.5 Conclusions

In conclusion, an integrated view of access control is needed in an environment of interconnected ADs in order to achieve effective and efficient placement of function for different types of policy, and to define consistent policies across the many network elements, protocols, and servers that are involved. In this chapter, we proposed a framework for placement of access control functions. We applied the original *end-to-end* argument to network resources and concluded that resources other than sufficiently protected end-systems are best protected at the border routers. This argument is reinforced by the increasing concerns with respect to resource usage feedback and cost recovery that are raised by the commercialization of internetwork transit facilities [33].

## 1.6 Overview of This Thesis

The remainder of this thesis is organized as follows. Chapter 2 begins by reviewing and discussing related research in policy enforcement. We concentrate on two areas: i) network security and internetwork access control, and i) internetwork routing. Subsequently, in the second part of the chapter, we identify and discuss several basic support mechanisms used as building blocks in our design.

Chapter 3 addresses stub AD policy enforcement mechanisms. We introduce *Visa* protocol for controlling packet traffic at stub AD boundaries; its main purpose is the establishment of authorized and authenticated end-system associations. The key features of the protocol are:

- A special ticket, called a *visa* is required for communication outside an AD.
- Only select end-systems (those that are sufficiently protected) are granted visas.
- Access Control Servers in both end-point ADs must authorize communication before a visa is issued.

[10] We exclude the issues discussed in Section 1.4.5 because the suggested approach is very context-dependent and does not lend itself to accurate representation in a table.

- Visas are distributed to the authorized end-systems and border-routers in the end-point ADs.
- Every packet that attempts to leave or enter an AD is expected to be stamped with a valid visa thereby proving its authenticity.

After describing the protocol and discussing numerous design issues, we analyze its security and associated overhead costs.

Chapter 4 is concerned with controlling access to transit AD network resources, i.e., control of transit internetwork traffic. It begins by attempting to extend network access control methods to control of transit traffic. In response to some fundamental deficiencies of existing approaches, we conclude that novel protocols are necessary in order to address the problem effectively. Such protocols are embodied in the Inter-Domain Policy Routing (IDPR) architecture which is used as a springboard for our design. IDPR has the following key features:

- Each AD expresses its policy in Policy Terms (PTs) which are disseminated to all other ADs along with the inter-AD topology information.
- Individual end-system connection are aggregated into coarser-grained Policy Routes (PRs). A PR is an AD-level source route that also includes PTs necessary to make policy authorization decisions in transit ADs.
- PRs are installed (and authorized) in all intervening ADs before any communication can take place.
- Since the "expensive" part of policy enforcement is done at PR installation time, subsequent data packets encounter little scrutiny, i.e., delay.

After identifying a number of security issues and threats facing IDPR, we specify secure PR setup and packet forwarding protocols, analyze their security and address the performance costs.

Chapter 5 supports the choices made in our protocol design by demonstrating and evaluating experimental results obtained from prototype implementations of protocols proposed in Chapters 3 and 4. In conclusion, Chapter 6, reviews the contributions of this thesis and discusses topics for future research.

# Chapter 2

# Background

This chapter sets the stage for protocol design by reviewing related work in network security and internetwork routing and identifying the inadequacies of the existing policy enforcement approaches. It also discusses a number of support mechanisms that are employed throughout the rest of this thesis.

## 2.1 Related Work

Policy enforcement is not an entirely new subject. Much effort has been put into the enforcement of certain types of policy, especially, access control in stub AD environments. Most of the related work comes from two areas: *Network Security* and *Internetwork Routing*. Previous results in these two areas form a solid background for the design of stub and transit policy enforcement, respectively.

### 2.1.1 Network Security

Research in network security dates back to the mid-seventies when computer networks first began to proliferate. There has been a lot of research in the field, as evidenced by the enormous amount of literature. It is impossible to treat all of it thoroughly; related work considered below was selected for being the most applicable to the subject of this thesis.

Since early networks were, for the most part, technically and administratively homogeneous, security issues concerned basic services, such as session and user authentication, data integrity and confidentiality.

In their pioneering work [67], Needham and Schroeder introduced third party authentication protocols based on both conventional and public key encryption. The purpose of these simple, but elegant, protocols is to allow the establishment of a secure channel between two mutually suspicious principals by providing them with a shared secret, that can consequently be used a session key. The protocols make use of a trusted Authentication Server that shares pairwise keys with all principals and can be trusted to to generate *good* session keys.

Variations of the Needham-Schroeder protocols are used in existing control mechanisms, most notably, Kerberos Authentication Server[85].

Voydock and Kent in [90] treated security in high-level network protocols by considering a broad range of security risks and possible attacks and suggesting a number of encryption-based countermeasures. Their main contribution lies in outlining the relationship between cryptographic and network protocols in the context of structured protocols such as the OSI reference model.

In the late seventies, the DARPA Internet[1] evolved into the first truly heterogeneous, dynamic internetwork [12]. This prompted increased concern with access control across autonomous network boundaries.

This subject and other security-related issues in inter-organizational setting were first discussed in a series of papers by Estrin [24, 25]. The main contribution of this work is twofold: i) it identified a number of issues that set inter-AD access control apart from the more traditional access control scenarios and demonstrated the need for network-layer controls, and ii) it suggested a range of possible solutions, some of which later served as the basis for protocol design in this thesis (e.g., *Visa* Protocol [24, 32]).

[1] Hereafter referred to as simply *Internet*.

At the same time, (non-military) government agencies became aware of the security issues and produced a number of reports and guidelines, most notably, a paper by Gomberg describing a model for inter-administration network authentication and access control[38]. Also, work by Nessett at DOE analyzed the security implications of the heterogeneous network administration[69].

A subject that received a lot of attention is the authentication of principals in a suspicious distributed system environment. In particular, a paper by Birrell et al. presents an authentication service without global trust[6]. However, since the same (rather elaborate) authentication mechanism is prescribed for use by all intended participants, autonomy and flexibility are sacrificed. Another authentication scenario for a distributed system environment is described by Sollins[83] in the paper on cascaded authentication. In it, authentication between parties not enjoying mutual trust is achieved by chaining pairwise authentication between adjacent parties that enjoy such trust.

For the most part, interconnection of autonomous networks takes place at the network layer to provide datagram-level connectivity. (Interconnection at higher layers is possible commensurate with losses in performance and flexibility. A comparison between network and higher-layer access control approaches can be found in [25]). At the network layer, communication between a pair of end-systems in different ADs involves traversing a sequence of (at least two) network-layer border routers. Therefore, more than two principals would have to be involved in network-layer policy enforcement. For this reason, methods described above are not applicable to access control across network-layer AD boundaries.

One major effort to provide network-layer access control for the internetwork environment is the Security Protocol 3 (SP3) [81]. SP3 originated from a larger project, Secure Data Network Systems (SDNS), overseen by the National Security Agency.[2] The goal of SP3 is to provide *transparent* security services (connectionless confidentiality and integrity, access control and data origin authentication) for the constituent end-systems. SP3 is implemented in so-called SP3 systems, each SP3 system serves a set of end-systems. (An SP3 system can be viewed as a border router). Because it is decoupled from the end-systems, SP3 assumes a *trusted path* between an end-system and its SP3 system. Since individual end-systems are not authenticated, SP3 does not (as specified) protect against masquerading attacks. Furthermore, SP3 has no facility for on-demand association establishment between SP3 systems with no history of previous communication. Last (but not least), SP3 is not specified to detect replay attacks, i.e., duplicate or out-of-order packets.

An instantiation of SP3 architecture is the The Blacker system[5].[3] It is is a hardware unit designed to secure user traffic in sensitive packet networks. It protects data from disclosure during transit, ensures correct identification of packets by address, and enforces security labels.

Another workable approach is *Visa* protocol [32, 27], a network-layer mechanism for establishing authorized and authenticated inter-AD network connections. (A simpler and less secure variant using packet filtering is described in [61]). In ISO parlance, *Visa* protocol provides connectionless integrity, data origin authentication and access control services. It involves Access Control Servers (ACSs), border gateways and select end-systems (those that are allowed external access). Before an inter-AD connection is established, both end-systems must be authorized and authenticated by their respective ACSs. After establishing authorization, ACSs jointly issue a visa to the requesting end-systems. The same visa is distributed to the border routers in each AD. A *visa* is a certificate authorizing two end-systems to communicate. Included in a visa is a visa-key, a secret quantity which end-systems use to sign subsequent data packets. Intervening border routers authenticate data packets by verifying packet signatures. We describe *Visa* protocol in greater detail in Chapter 3.

### 2.1.2 *Ad hoc* Methods

In order to further motivate the need for a comprehensive policy enforcement architecture, this section reviews some simple, ad hoc stub AD access control methods currently in use. Some are direct applications of the related work described in the previous section.

- Physical isolation
  Physical isolation of externally accessible resources is, by far, the simplest and the most drastic of all methods. It requires configuration of a separate network and increased end-point security for all externally accessible resources. While this provides perfect security (only in a sense of separation) communication between externally accessible and internal resources is impossible. It also requires that every distinct set of externally accessible resources be isolated, an impractical task in case of multiple overlapping sets of externally-accessible resources.

[2] In cooperation with several other government agencies and private corporations.
[3] The name refers to the proverbial black box.

- Protection of all internal resources
  If stronger security mechanisms are incorporated into all internal resources, policy can be enforced, but at the price of infringing upon intra-AD communication. Moreover, when the set of externally-accessible resources is small as compared to the rest of the AD, this becomes a highly impractical approach[24].

- Application-specific filtering
  If traffic is restricted to a specific application, e.g., mail or voice, application-specific filters can be build to implement this policy. In general, however, a separate filter must be built for each application anticipated[61]. Moreover, performance overhead of application-level filtering may prove prohibitively high, especially, for throughput-oriented applications, e.g., real-time voice and video[25].

- Access control lists
  If external access is confined to a relatively static set of entities, access control lists can be used, once again, to filter traffic. Unfortunately, this will not accommodate dynamic requirements. In addition, list-based filtering is subject to spoofing as network addresses can be easily modified.

- Bilateral policy agreements
  Two or more ADs can always agree out of band to follow a specific policy or agree on a set of policies. Such an agreement may include adopting a common charging scheme or a common authentication protocol. This can work for a limited number of policies, but requires that autonomy be sacrificed.

Any of the above methods can be effective under special circumstances. Their main flaw is the lack of flexibility. Each addresses a small subset of possible policies, while compromising performance, flexibility or autonomy.

### 2.1.3 Internetwork Routing

Network routing has received a lot of attention since the late fifties as evidenced by the enormous amount of literature in the field. Several fundamental routing algorithms were developed, most notably, Dijkstra's Shortest Path [22] and Ford and Fulkerson's Max Flow algorithms [35]. The former gave rise to a family of routing protocols known as *link state*, and the latter, to a collection of protocols known as *distance vector*. Link state protocols are characterized by each node keeping a "map" of the entire network over which it computes shortest paths to all destinations. Each node contributes to this "map" by flooding the network with a link state packet, i.e., a packet that contains the status of all incident links. In distance vector protocols, nodes keep tables of the best paths and associated metrics for all possible destinations and periodically exchange the contents of this table with neighbors.

As mentioned before, the DARPA Internet evolved into the first large, decentralized and dynamic datagram network. At first, its routing protocol was of a distance vector variety, as described in [55]. However, as the network grew, shortcomings of the distance vector became more apparent. Frequent oscillations and otherwise unstable behavior that it exhibited were due mostly to long propagation delays with respect to changes in topology. The successor [56], was a link state protocol with a relatively short convergence period and looping avoidance. Neither protocol incorporated any notion of security or policy.

As the Internet grew, it began to encompass a greater number of autonomous networks, or ADs, using our terminology. With regard to routing, this growth presented two problems:

- **Autonomy**
  while electing to become part of the Internet, ADs do not necessarily wish to expose their internal network structure to the rest of the world. In other words, there is a need to limit the dissemination of routing information.

- **Scale**
  the size of the Internet makes the deployment of a global routing protocol undesirable. Since routing information is typically propagated throughout the entire domain of a routing protocol, routing tables in participating gateways grow in proportion to the size of the Internet. Therefore, in order to avoid an *information explosion* routing information granularity must be coarser than end-systems or networks.

In response to these problems, two types of routing protocols were defined:

- **Interior Gateway Protocols (IGPs)** are intended for use within a single administrative entity, i.e., AD. Routers employing an IGP exchange reachability information pertaining to entities within an AD [65].

- **Exterior Routing Protocols (EGPs)** are used by AD border routers to learn about reachability of networks in other ADs [65]. (We discuss EGPs in the following subsections).

Some of the notable IGPs are: IGRP [40], OSPF [62] and DEC IS-IS [21]. IGRP, a distance vector protocol, supports ToS routing indirectly by distributing several different metrics (e.g., delay, bandwidth). Routers assign a weight to each metric before combining them into a single composite metric which then serves as a basis for "shortest" path selection. Both OSPF and IS-IS are links state protocols. They may include a number of metrics corresponding to different ToSs in link state updates.

These IGPs are well-suited for their intended application domain, i.e., a single-AD environment. However, they do not scale to a large number of ToS-s and require route computation to be repeated for each ToS supported [7]. In addition, most IGPs are designed with little concern for security as components are assumed to share a certain level of trust. This assumption is unreasonable in a multi-AD environment where routing information may not be "trusted" across AD boundaries.

#### 2.1.3.1 Exterior Gateway Protocol

As the Internet incorporated a more diverse organizational mix, coexistence of multiple administrations (and the associated security implications) was recognized as an important problem. The Exterior Gateway Protocol *(EGP)* [77] was the first routing protocol to address this issue.[4]

*EGP* was designed to communicate reachability information among administrative regions that do not enjoy mutual trust. It includes an authentication facility for validating routing information exchanged among the regions. The regions hooked together by *EGP* can be viewed as ADs.

*EGP* supports a very limited notion of policy. Individual ADs are allowed to hide portions of their routing database that they are not willing to share. Also, ADs are free to manipulate route metrics that they assign to other ADs in order to favor or preclude certain transit AD hops. However, *EGP* does not provide for ToS-based or other fine-grained policy enforcement. In *EGP*, an AD makes routing decisions based only on its own policy, since *EGP* provides no facility for the distribution of policy information across AD boundaries.

Furthermore, in order to avoid routing loops *EGP* imposes a topological restriction on AD interconnection in the form of a cycle-free hierarchy. As Clark points out in [13], *EGP's* restriction on the interconnection topology has proved unsatisfactory. In general, topological restrictions are undesirable as they inhibit autonomy and are near impossible to enforce [7, 26].

#### 2.1.3.2 Border Gateway Protocol

BGP is a recently proposed addition to the Internet Protocol family[53]. It was designed to be a successor to *EGP* and a variant has been submitted as an international standard[2]. Its foremost goal is to provide efficient and robust Inter-AD routing with rapid convergence and loop detection for arbitrary internetwork topologies.[5] It is primarily intended for use by transit ADs and inter-operates with other interior routing protocols.

BGP is designed under the following assumptions:

1. Policies can be expressed using information about the full AD path that packets will travel to a destination.
2. Transit policies apply uniformly to all sources.

BGP uses hop-by-hop routing and a distance vector algorithm for the next hop selection [55]. One common benefit of traditional distance vector algorithms is the ability to hide network structure. Neighboring nodes exchange reachability information for a specific destination in the form of distance metrics corresponding to each next hop. Nodes do not exchange information about subsequent hops to the destination. BGP augments this traditional approach by distributing full AD-level paths. In other words, for each destination advertised, nodes specify the AD-level path to that destination.

[4] In this section, the term *EGP* denotes a specific protocol. Whereas, EGP, as referred to in the previous section, denoted a class of protocols.

[5] BGP and *EGP* use the term *Autonomous System* and *Routing Domain*, respectively. We use the term *Administrative Domain*. They are not completely equivalent but, for the sake of this discussion, they can be interchanged. See [54, 26] for further discussion.

As a result, BGP provides less information hiding in return for the ability to detect routing loops quickly. By using full AD paths to detect loops BGP avoids convergence problems [56] without imposing topological restrictions on AD interconnection. In addition, AD path information can be used as policy criteria for route selection.

BGP allows for limited policy-based route selection. A BGP router can select its next hop based on the information provided in the full AD path, in addition to the distance metric. For example, $AD_A$ can reject all routes through $AD_B$. On the other hand, each AD must apply the same route selection decision to all packet sources, including itself. For example, $AD_A$ can not reject all routes through $AD_B$ for itself without affecting its neighbors, and vice versa. Similarly, an AD can not apply one policy to one neighbor and a second policy to another neighbor. Since BGP *was not* intended to implement policies that discriminate between traffic end-points with arbitrary granularity, the approach achieves its goals [53].

Each BGP router can be configured according to its AD's local policy. Even though local policy is not distributed among ADs, it is represented in a universal *policy language*. A policy in this language is an expression:

*[Network-list,AD-path]=preference*

The semantics of a policy are as follows: if a routing update for a network in *Network-list* is received via *AD-path* and its *preference* metric is better than that of a path currently in use, then, this update must be used for subsequent routing.

#### 2.1.3.3 Inter-Domain Routing Protocol

Inter-Domain Routing Protocol (IDRP) is an extension of BGP that has been proposed as an international standard. IDRP augments the BGP protocol by including (among other features) distribution lists along with route information [2]. The list may be inclusive or exclusive and is propagated along with next hop and full-AD path information. Each border router along a path may further restrict a distribution list before advertising a route, i.e., ADs may be deleted from the inclusive list or added to the exclusive list but no router can relax or ignore the list.[6] This feature allows IDRP to support some source-specific policies. However, IDRP has no built-in support for enforcing source-specific policies at packet forwarding time. Another departure from BGP is the IDRP's ability to include policy-related (e.g., ToS or User Class) information in routing updates.

IDRP is thus able to support a wider range of policies than BGP. Nevertheless, because IDRP is a hop-by-hop protocol, it only allows a single route per every $[destination, ToS]$ to be advertized. However, multiple routes for a given $[destination, ToS]$ combination may be necessary in order to allow traffic sources to apply route selection policies. (See [7] for an in-depth discussion of this and other related issues).

#### 2.1.3.4 Routing with Multiple Hierarchical Addresses

A novel approach to policy routing is the use of multiple hierarchical addresses (MHA). In [87], Tsuchiya suggests that multiple addresses be assigned to end-systems (stub ADs, in our parlance). A single address is formed as $[stub.regional.backbone]$ indicating that the corresponding route: $[backbone \Longrightarrow regional \Longrightarrow stub]$ satisfies the policies of its component ADs. A given end-point may have a number of such addresses differing in the *regional* and/or *backbone* fields.

Routing in this approach can be viewed as a variant of source routing. More specifically, a route between $AD_a$ and $AD_b$ is the combination of $AD_a$'s address and the *inverse* of $AD_b$'s address, e.g., $AD_a.regional_1.backbone_1$ followed by $backbone_2.regional_2.AD_b$. To route a packet, a stub AD simply selects (according to its policy) one of the addresses for the intended destination.

The main benefit of MHA is its simplicity and low overhead with regard to route computation. There are a few important drawbacks, though. First, a *shallow* (three-level) hierarchy is assumed. As pointed out above, such restriction is undesirable for two reasons: i) lateral and bypass links must be supported as the Internet is not expected to conform to a strict hierarchy, and ii) even if a strict hierarchy is possible, limiting it to three levels may be inadequate in the context of a global Internet. A related problem is the assumption regarding backbones. If the backbone components of a route are not identical, multiple transit backbones have to be traversed. Such backbones, however, are not included in either of the two addresses. Consequently, policy enforcement is severely limited from the perspective of the end-points (since transit backbones are *hidden*) from them. Conversely, transit backbones, not included in the "route" must

[6] The proposed standard includes several other extensions which are not directly relevant to our discussion.

enforce their policy on a per-packet basis, i.e., they are denied any opportunity of restricting traffic in advance of actual communication.

#### 2.1.3.5 IDPR

Routing protocols discussed thus far have been developed with limited concern for policy enforcement. Designed with more conventional routing in mind, these protocols either impose topological restrictions and do not scale well (e.g., EGP), or can not support large numbers of diverse and dynamic policies (e.g., BGP, IDRP and MHA).

In his landmark paper, Clark [13] first motivated the need for the integration of policy support into the routing function and presented a *blueprint* for policy routing in the Internet. The Internet Inter-Domain Policy Routing Working Group (IDPR-WG) has since developed an architecture for Inter-Domain Policy Routing (IDPR)[7] that is largely based on Clark's policy routing proposal.

IDPR represents a significant departure from the more traditional routing protocols. In brief, the distinctive features of IDPR are (we discuss IDPR in greater detail in Chapter 4.1):

- **Policy Terms (PTs)** are units of policy expressed in a universal policy syntax. Every AD includes its PTs in a link state update which it distributes to *all* other ADs via a flooding mechanism.
- **Policy Routes (PRs)** are source routes at the granularity of ADs. A PR is computed by a source AD and then installed at all intervening ADs in advance of the actual communication. Subsequent data traffic flows along established PRs.[8]

IDPR was designed to support a wide range of policies while alleviating global consistency requirements. Stub ADs enforce their policy at the time of route computation. Transit ADs advertize their policies to their stub counterparts that, in turn, use these policies to compose PRs. Transit policy enforcement takes place when PRs are installed at the transit AD hops.

IDPR supports the following policy attributes: source and destination ADs, previous and next AD hop, Quality-of-Service, Time-of-Day, User Classes, authentication and security requirements, and charging conditions. In return for this functionality, IDPR presents a number of challenging problems relating to complexity and scale [7].

In Chapter 4, we discuss IDPR in greater detail concentrating on security aspects of its policy enforcement mechanisms.

#### 2.1.3.6 Secure and Robust Routing

At an *extreme* of robustness and security is *Network-layer Protocol with Byzantine Robustness* (NPBR) by Perlman [71]. NPBR concepts are suitable for use as either IGP or EGP, mainly because it makes no assumptions about trust among its components.

NPBR comes in two flavors: flooding and link state. Flooding NPBR is a highly-robust protocol where communication between two nodes is guaranteed as long as there exists a non-faulty path between them. This robustness is achieved at the expense of: (1) flooding data packets, (2) per-packet public key encryption at every hop, and, (3) significant state in routers. The value of this protocol is largely theoretical as it illustrates the limits of achievable network-layer robustness and security.

Link state NPBR is slightly less robust. It guarantees communication between two nodes as long as there exist $n$ node-disjoint paths between a given pair of nodes, and at most $(n\text{-}1)$ node failures exist simultaneously in the network. Reduced robustness in link state NPBR is counter-balanced by the use of a link state protocol in conjunction with source routing. Link state update dissemination is performed using highly-robust flooding NPBR method, whereas data packets are source-routed. This design, while still quite costly, demonstrates some useful techniques:

- Public key signatures for link state updates to defend against tampering and repudiation of origin.
- Per-node nonwrapping sequence numbers to achieve replay detection and packet reordering.

[7] Not to be confused with IDRP. For the record, IDPR was named first!
[8] This bears some resemblance to a traditional virtual circuit model. However, without packet sequencing or reliable delivery.

- End-to-end and hop-by-hop packet delivery acknowledgements to determine dynamically the status of links and nodes traversed by a packet.

In Chapter 4, we take advantage of these techniques in our design of secure transit policy enforcement.

## 2.2 Support Mechanisms

This section briefly addresses a number of common support mechanisms used as basic building blocks in our policy enforcement protocols.

### 2.2.1 Encryption and Signature Support

Protocols that implement security services for policy enforcement will have to make use of encryption to support authentication, data integrity, and confidentiality (if applicable).

Two dominant types of encryption are: conventional (or symmetric) and public key. Conventional cryptography has been in use for quite a long time [49]. Typically, it involves a function and a single key used for encryption, as well as decryption, of data. The key must be shared among every group (usually, of size two) of principals wishing to communicate in secret.

The best-known (if not the most notorious) of the contemporary conventional cryptosystems is the Data Encryption Standard (DES)[66]. It is, at present, a United States standard which makes it (and a number of derivatives) the most widely used cryptosystem.

Some conventional cryptosystems are also suitable for generating digital signatures, e.g., FEAL-8 [82] and DES [66]. A digital signature is a value that, when attached to a message, proves that the message has been generated by a party in possession of the key used in the signature computation.

For the most part, conventional cryptosystems lend themselves to efficient implementations and are able to reach encryption rates of several megabytes per second (thereby matching some LAN speeds). On the other hand, conventional cryptography has several drawbacks:

- Sharing a distinct key for every pair of principals makes key management extremely difficult ($N^2$ keys are required for $N$ principals).
- An unfortunate consequence of key sharing is the inability of attributing encrypted messages to a single principal. In other words, a message encrypted (or signed) with a shared key can be generated by any of the (at least two) principals in possession of this key.
- Finally, authenticated multicast and broadcast communication is rather laborious. In order to send a message to a group of $N$ principals, the sender must produce $N$ distinct message signatures, one for each intended recipient.

Public key encryption[9] addresses some of the drawbacks of conventional encryption. In a public key cryptosystem, each principal is associated with a unique key-pair. A key-pair consists of a *public* (encryption) and a *private* (decryption) key. The former is made available to anyone who wants to communicate with the principal in question, while the latter is kept secret. Encryption is performed with the public key, and decryption with the corresponding private key. The most important feature is that, given a principal's public key, computing the corresponding private key is computationally infeasible.[10]

The best-known example of a public-key cryptosystem is the RSA [76]. In RSA, the difficulty of attacking the system is equivalent to factoring large numbers. RSA has been extensively researched and scrutinized in the past and is considered one of the most secure cryptosystems available. Another example of a hereto unbroken public key cryptosystem is the *El Gamal* scheme[23] which is very slow (even by public key standards), but well-suited for certain specialized tasks, e.g., key distribution.[11]

[9] First proposed by W. Diffie and M. Hellman in 1976 [20].

[10] Derivation of a private key from its public counterpart is usually equivalent to solving a hard, e.g., NP-complete, computational problem [20].

[11] El Gamal's slowness is due to exponentiation complexity and expansion in ciphertext size.

Public key encryption addresses many of the conventional encryption's drawbacks.

- Since a principal only needs a single key-pair, key management is no longer a problem.
- A principal generates signatures with a private key known only to it. Therefore, there is no ambiguity in tracing a signed message to its origin.
- Authenticated multicast or broadcast are easily achieved by a single message signature computed with the sender's private key. Any of the intended recipients can authenticate the origin and the contents of the message by verifying the signature with the help of the sender's public key.

In return for all the benefits it provides, public key encryption takes a heavy toll in terms of performance. As compared to its conventional counterpart, public key encryption is extremely slow.

Methods for reducing the high cost of public key encryption have been proposed. Specifically, in RSA, the cost of signature computation (with private key) is roughly the same as the cost of signature verification (with public key).[12] For example, Privacy-Enhanced Mail[51] uses a *scaled-down* version of RSA where signature verification is significantly less expensive than signature computation. This is accomplished by using large private keys in conjunction with relatively small public keys.

In order to avoid the respective drawbacks and combine the respective benefits, conventional encryption is frequently used for data integrity and confidentiality (encryption), while public key encryption is reserved for session key distribution and authentication. The end-result is a *hybrid* cryptosystem.

In the context of policy enforcement, public key encryption is particularly useful for disseminating routing information, especially in the form of link state updates [71]. A link state packet can be signed once by its originator and the signature can be easily verified by all potential recipients, assuming that the originator's public key is readily available. Also, as described in the next section, inter-AD associations can be established using public key certificates. Conventional signatures are especially well-suited for data integrity once the relevant principals are authenticated.

As discussed in Chapter 1, confidentiality is not a relevant security service in the context of policy enforcement, hence, applicable security services do not require bulk encryption. Message signatures suffice for origin authentication and data integrity services.

It is not necessary to sign the entire message [17]. Instead, a short (e.g., 128-bit) digest of a message is produced and then signed, thereby greatly reducing the costs. The security of this type of signature is dependent on the digest computation function, usually referred to as a *hash function*. A *strong* (or *one-way*) hash function must have the following properties:

- It must be computationally difficult to find two messages that hash to the same digest.
- It must be computationally difficult to find a message that hashes into a given digest.

In practical terms, it is important for a hash function to lend itself to fast implementations, at least an order of magnitude faster than the signature function. In recent years, some hash functions conjectured to be one-way have been proposed. Some have been successfully attacked, e.g., Merkle's SNEFRU [57, 3]. Others, such as Rivest's MD4 and MD2 [75], appear to be more resilient. (MD4 is discussed in detail in Appendix A).

The use of hash functions for message authentication entails signing only a short fixed-length message digest value. Nonetheless, encryption-based signatures are still quite expensive. In Appendix A, we describe two methods of encryption-free message authentication based entirely on the use of one-way hash functions. (A similar scheme was developed independently by the Internet Security and Privacy Working Group [36]).

### 2.2.2 Certification

Our internetwork model assumes a very large number of interconnected ADs. Assuming that all ADs participate in a global public key encryption scheme, each AD would need to have reliable knowledge of *all* other ADs' public keys in order to support arbitrary communication patterns. Moreover, a global, secure key management scheme would be

[12] Usually, both keys are of equal (or near-equal) length.

necessary to distribute, grant and revoke, keys. Overhead due to both storage and maintenance makes this scenario rather undesirable. Nonetheless, since policies are most often based on AD allegiance, principals must have means of proving their identity. Principals include the various AD-level servers: Access Control, Route, Policy, Name and Time to name just a few.

To allow for more dynamic, yet secure, on-demand binding between names and public keys, a technique known as *certification* is used.[37] Certification is performed by well-known trusted Certification Authorities (CAs). Included in a certificate is a name (e.g., a DNS name [60] or an X.500 distinguished name [10]) and a corresponding public key. A certificate also contains the name of the issuing CA, and the expiration date and time. Most importantly, a certificate is signed with the private (secret) key of the issuing CA. All interested parties are thus only required to possess the CA's public key as opposed to a multitude of principals' public keys.

In the simplest case, there is a single universally trusted CA. However, in a heterogeneous internetwork environment, it is unlikely that a single CA will suffice for reasons of scale and policy. A more realistic scenario is the CA hierarchy as described by Gasser et al. in [37]. For example, a worldwide internetwork may require at least a four-level hierarchy. The top level CA is responsible for certifying individual countries' CAs which, in turn, certify ADs in their domain. Bottommost, AD-level CAs may be employed to certify constituent users and end-systems.

In this model, an association between two principals (A and B) is established as follows. If A and B are under the jurisdiction of the same CA, A forwards its certificate to B and B verifies the signature and the expiration time. (The procedure is then repeated with B sending its certificate to A). Otherwise, the least common ancestor (LCA) in the CA hierarchy is established. Then, A supplies B with not one, but a list of, certificates starting with A's own certificate and ending with the certificate of the LCA. Since B is able to verify LCA's certificate, it iterates through the list of certificates culminating with the verification of A's certificate.

Certificates are currently being used in Privacy-Enhanced Electronic Mail (PEM) [51]. PEM certificates are used for proving identity of mail message originators. Certificates can be obtained from either a private company (RSADSI) that manages the PEM's authentication hierarchy for a fee, or an organizational notary (ON) who will vouch for people in an organization. (ONs are under contract to only issue Certificates legitimately).

This approach is well suited for the distribution of signed routing updates and for route setup. Key distribution will involve route servers and border routers and will be at a coarse granularity of ADs. Public Key certificates can be used to identify ADs that have no previous history of association. For example, a source AD can include its public key certificate as a part of route setup when one (or more) ADs in the route are being used for the first time. Certificates can also be very useful in stub policy enforcement (see Chapter 2) where a new association between two stub ADs may begin with an exchange of the certificates as a form of *secure introduction.*

Throughout the remainder of this thesis, we assume the existence of a certification hierarchy as described above. At the very least, every relevant principal (servers and some border routers) is assumed to possess a certificate.[13]

### 2.2.3 Time Synchronization

The purpose of a time service is to provide its clients with continuous, accurate time synchronized with global (or national) standards [58]. Time service is a major contributor to the proper function of a network as many network protocols assume the presence of a reliable time service to achieve clock synchronization.

One example of a working time service is the Network Time Protocol (NTP) [59] currently used in the Research Internet. NTP time servers are arranged in a three-level hierarchy. The hierarchy is flexible,[14] i.e., it is resilient to certain types of node failures. The actual source of precise time is usually an external system (e.g., an atomic clock) with which top-level time servers are synchronized.

In an environment of interconnected ADs, there are several security threats facing a time service [4]:

1. impersonation of a time server
2. modification of time server messages
3. replay of previously recorded time server messages

[13] The granularity of certificates can vary between ADs. Some ADs may issue distinct certificates to different servers and/or routers, while others may have a single certificate shared among all servers and routers.
[14] Except for the top-level servers.

4. prevention of time server messages from reaching their intended destination

5. delay of time server messages (e.g., by flooding the network)

In [4], NTP is analyzed for susceptibility to these threats. The conclusions and recommendations made are applicable to any time service. In particular, threats (1) and (2) can be countered by using message authentication while protection from (3) requires maintaining state with respect to all possible peers. Threats (4) and (5) are not addressable within the time service itself.

In the context of this thesis, secure and available time service is utilized in stub and transit policy enforcement for insuring the timeliness and uniqueness of the various resource access requests. In other words, a timestamp is treated as as a unique sequence number (i.e., a nonce [67]) as well as a freshness indicator.

We assume that the principals' clocks are loosely synchronized with a maximum clock skew of $\Delta_T$[15] between any pair of principals. Furthermore, clocks are assumed never to run backwards. In general, when a principal A receives a packet from its peer B, the packet timestamp, $T$, is validated as follows:[16]

If A and B have communicated recently (within last $\Delta_T$ interval) A is required to keep the timestamp of the last packet ($TOLP_B$) received from B. $T$ is considered valid if: (1) $T$ is greater than $TOLP_B$ and (2) $T$ is within the maximum clock skew. This insures that $T$ is timely (within the limits of the clock skew) and has never been used before.

If A and B have not communicated recently, i.e., $TOLP_B$ does not exist, A can establish $T$'s freshness by making sure that it is within the maximum allowed clock skew, i.e. it differs from the current clock reading by at most $\Delta_t$. If this condition is satisfied, A is implicitly assured of $T$'s uniqueness (if $T$ had been used in the past, A would have kept a record of it).

[15] The value of $\Delta_T$ depends on the particular protocol.

[16] Origin authentication and data integrity services are assumed.

# Chapter 3

# Stub Policy Enforcement: *Visa* Protocol

In this chapter we address policy enforcement for stub AD communication. The key design goals and guidelines have been outlined in Chapter 1. As discussed in Chapter 2, there are several existing approaches for controlling inter-AD communication at stub AD boundaries. *Visa* protocol, the mechanism described below, has been selected for several reasons:

- Flexibility - protocol operation is almost entirely dependent on the particular access control and authentication policies of the participating AD.
- Functionality - protection against: i) unauthorized AD entry/exit, and ii) modification, substitution and replay of legitimate inter-AD traffic.
- Layering - the entire protocol is situated at the network-layer.

## 3.1 Overview

*Visa* protocol is a mechanism for controlling the flow of packet traffic to and from end-systems in a stub AD. Before an end-system can communicate across its AD boundary, the communication has to be authorized according to the policies of both local and destination ADs. Authorization can be obtained via a dialog with an Access Control Server (ACS) on local and destination ADs. The need for and particulars of this dialog are determined independently by the administration of each AD involved. When the communication is approved by both end-point ADs, the respective ACSs issue *visas* to the requesting end-system.

A visa is a cryptographically sealed certificate issued by an Access Control Server (ACS). It contains, among other things, a secret quantity, known as the *visa-key*. Each packet belonging to an authorized stream carries a *visa-stamp*, which indicates that the packet is allowed to leave (or enter) an AD's network. A visa-stamp is a function of the visa-key and the packet's data. It is attached to the packet by the originating end-system and is then re-computed and verified by the border routers of the end-point ADs.

In *Visa* protocol, border routers do not bear sole responsibility for making access control decisions. By issuing a visa, an ACS has pre-computed a decision such as *"end-systems $H_a$ and $H_b$ are allowed to communicate"*, or *"end-system $H_a$ can be trusted to pay its bills"*. The task of a router is reduced to ensuring that a visa is valid and is being used correctly; the expensive part of the policy enforcement is done once per connection, by the ACSs of the end-point ADs, rather than once per packet, by the border routers.

## 3.2 History

The term *visa* was first suggested by D. Reed, and documented by J. Mracek [63]. A detailed analysis of the issues associated with multi-AD internetworks, as well as the original motivating factors leading to the development of *Visa* protocol, can be found in [24]. The first detailed description and the informal analysis of the protocol appeared in [32].

Subsequent research [27] resulted in the development of two protocol models based on different philosophies with regard to state in visa-routers. The original *stateful* model requires that participating border routers maintain **reliable** tables of active visas. In it, ACSs explicitly distribute visas to visa-routers. Although the loss of state in a visa-router is not fatal to communication, overhead is incurred in the process of re-establishing the necessary state. In contrast, the *stateless* model avoids the necessity for the distributed state, but requires some additional encryption steps. The stateless model has several advantages: higher fault tolerance (insofar as routers), lower router storage requirements, and the ability to accommodate multiple visa-routers without additional overhead. All this is gained at a price of higher per-packet processing costs and increased packet size.

In the remainder of this chapter, we use the experience from previous work to design the next-generation *Visa* protocol. In the next section, the goals of *Visa* protocol are formalized. Network environment is discussed in Section 3.4. *Visa* protocol participants and their respective requirements are addressed in Section 3.5. Section 3.6 presents the new *Visa* protocol, and Section 3.7 addresses the key design issues and choices. Section 3.8 analyzes the security of *Visa* protocol and Section 3.9 evaluates the storage requirements. Protocol implementation and performance results are presented in Chapter 5.

## 3.3 Goals

The primary goal of *Visa* protocol is to allow an AD to control communication between its constituent end-systems and end-systems in other ADs. If the end-systems involved can be trusted, then a stronger goal can be met: we can control the transmission of packets to and from *a specific* end-system in another AD. In a datagram network, as opposed to a circuit-switched network, the only information available about a packet must be attached to the packet rather than inferred from the route the packet follows. Therefore, we can state these goals more directly as follows.

1. A packet can leave the source AD, $AD_{src}$ if and only if $ACS_{src}$ has authorized the source end-system, $H_{src}$ to communicate with the destination end-system, $H_{dst}$.
2. A packet can leave $AD_{src}$ if and only if it originated at $H_{src}$ within a reasonable time interval, has not been modified in transit and is addressed for $H_{dst}$.
3. A packet can enter the destination AD, $AD_{dst}$ if and only if $ACS_{dst}$ has authorized $H_{src}$ to communicate with $H_{dst}$.
4. A packet can enter $AD_{dst}$ if and only if it originated at $H_{src}$ within a reasonable time interval, has not been modified in transit and is addressed for $H_{dst}$.

Another fundamental goal is not to impact intra-AD communication, nor to impose additional security measures upon unequipped end-systems, i.e., those that do not participate in inter-AD communication. Similarly, we wish to limit the overhead imposed upon ADs that are not concerned with controlling external access.

Finally, we would like to minimize the costs imposed by *Visa* protocol, including:

- Additional per-packet processing in border routers and end-systems
- Storage requirements for routers and end-systems
- Extra communication during connection setup
- Additional packet length (additional length increases latency and decreases throughput)
- Cost of recovery from router crashes

## 3.4 Network Environment

We assume that the internetwork closely follows the model of the DARPA Internet [70], which is substantially similar to the Open Systems Interconnection (OSI) model [43]. The essential features of this environment are:

- End-systems are autonomous and cannot necessarily be trusted.

- ADs are interconnected with routers; between any pair of end-systems in different ADs there are at least two routers, one belonging to each of the ADs. Conceptually, the connection between two ADs is a pair of half-routers connected via a trusted link. Each half-router can be trusted by its own AD but not necessarily by any other AD. The terms *border router* and *inter-AD router* are equivalent.
- All information flows via datagram packets. A packet consists of a *header* that includes addressing and other control information, and a data segment that is not intelligible to routers.
- A packet may flow through several *untrusted* ADs on its way to the destination.
- End-system addresses, both source and destination, can be forged. It is not possible (using hardware methods) to determine reliably which end-system actually sent a packet or to prevent a packet from being seen by unauthorized end-system.
- Packets traveling across an internetwork may be: i) lost, ii) duplicated, and iii) re-ordered.
- Successive packets between a given end-system pair may travel along different routes.

Lastly, there must exist a global name service which, in a secure and reliable fashion, provides a mapping from end-system network addresses to AD identifiers in addition to the more traditional mapping between end-system names and addresses. Along with AD identifiers, the name service has to provide:

- Addresses of ACS-s within an AD.
- Public key certificates for a given AD (or an ACS assuming each ACS is assigned a distinct certificate)

## 3.5 Participants

*Visa* protocol involves the following participants: access control servers, border routers, and end-systems. These participants and their responsibilities are described in this section.

### 3.5.1 ACSs

An ACS is an end-system, usually dedicated for security reasons, that is primarily concerned with access control. Each AD that implements *Visa* protocol has at least one ACS, responsible for authorizing its constituent end-systems for communication with end-systems in other ADs.[1] Multiple ACSs may be necessary for availability and performance reasons.

Each ACS knows of a number of local border routers that implement *Visa* protocol. ACSs are trusted and are sufficiently secure to defend against hostile attacks. The security of the overall protocol requires that ACSs be secure and that they employ an authenticated and secure channel for communication with local end-systems and routers. Furthermore, each ACS must be *identifiable* by a unique public key pair [ $EK_{ACS}, DK_{ACS}$ ] where $EK_{ACS}$ is the ACS's public (encryption) key, and $DK_{ACS}$ is the corresponding secret (decryption) key. Also, each AD is assumed to be a participant in a global, internet-wide certification scheme, whereby each AD (or each ACS therein) has a public-key certificate, $CERT_{ACS}$, issued by a well-known certification authority (as described in Chapter 2). Each ACS certificate contains (among other fields): ACS's address, $EK_{ACS}$, the name of the issuing authority and the certificate signature. This signature is computed with the issuing authority's private key, hence, anyone in possession of the corresponding public key can verify the certificate's validity and thus authenticate the certificate holder.

### 3.5.2 Border Routers

A border router is an end-system dedicated (for reasons of performance and security) to packet forwarding. Routers that use *Visa* protocol to enforce access controls are called visa-routers. All inter-AD connections must be implemented

[1] If a participant AD does not have an ACS, its end-systems will still be able to communicate with the end-systems in other ADs, although the AD in question will be subject to risks associated with the uncontrolled access.