DECEMBER 1988

WRL
Research Report 88/5

Visa Protocols for Controlling Inter-Organizational Datagram Flow: Extended Description

Deborah Estrin
Jeffrey C. Mogul
Gene Tsudik
Kamaljit Anand

Western Research Laboratory 100 Hamilton Avenue Palo Alto, California 94301 USA

Visa Protocols for Controlling Inter-Organizational Datagram Flow: Extended Description

Deborah Estrin
Computer Science Department, University of Southern California

Jeffrey C. Mogul
Digital Equipment Corporation Western Research Laboratory

Gene Tsudik
Kamaljit Anand
Computer Science Department, University of Southern California

December, 1988

This research was funded in part by the National Science Foundation, Presidential Young Investigator Award, with matching funds from GTE Inc. and NCR Inc., and by the University of Southern California Faculty Research Initiation Fund. Portions were funded by the Digital Equipment Corporation Western Research Laboratory.

digital **Western Research Laboratory** 100 Hamilton Avenue Palo Alto, California 94301 USA

Abstract

The increasing use of internetworking protocols to connect administratively heterogeneous networks has raised the question of how an organization can control the flow of information across its network boundaries. One method for doing so is the use of *visas*, a cryptographic technique for authenticating and authorizing a flow of datagrams. This report presents and evaluates two *visa* protocols — one that requires distributed state information in gateways and one that uses additional encryption operations instead of distributed state. Applications for such *visa* protocols include access control, accounting and billing for packet transit, and network resource management.

This technical report is based, in large part, upon a shorter paper [8]. We have extended the discussion of design issues and added an appendix describing a visa protocol using dual-key (public key) encryption.

Key Words: Computer networks, network interconnection, network security, access control, authentication, cryptographic protocols.

This report is simultaneously published by the Computer Science Department, University of Southern California, as Technical Report TR 88-50.

1. Introduction

The local-area and long-haul networks of many distinct organizations can be joined together into an *internetwork* through which datagrams flow without regard to organizational boundaries. The transparency of an internetwork is both a blessing and a curse: a blessing because it provides universal connectivity without requiring application-specific gateways, and a curse because it makes it much harder to control the flow of information between organizations.

Early internetworks ignored the issue of control, either because they connected organizations within a larger administrative unit (such as a single corporation, university, or governmental body) or because they connected research institutions with little need to limit information flow. Current internetworks connect organizations that may have competing interests. Thus, we can no longer ignore the need for controlling inter-organizational information flow. Similarly, in a multi-organization internetwork, costs must be billed to individual organizations or departments, resulting in a growing need for secure protocols to account for datagram traffic.

One approach is to introduce controls at a number of levels in the protocol hierarchy. We would like to preserve the useful properties of datagram-level transparency by controlling the flow of individual datagrams. We assume that higher-level controls will be implemented as appropriate to the particular applications and organizations involved.

To provide datagram-level control, Estrin and Tsudik have proposed the *Visa* scheme [9]. Conceptually, a secret key is used to compute an unforgeable mark placed on a datagram to assure a gateway that inter-organizational transmission of that datagram is properly authorized. This mark is called a *visa*, by analogy with the stamp made on a passport that allows the bearer to cross a border. We bind each visa to a single datagram in order to guarantee the authenticity of datagram contents. Visas were first suggested by David Reed, and documented by J. Mracek [13]. A detailed analysis of the issues associated with inter-organizational networks, as well as the motivations behind the visa scheme, can be found in [6].

In general, a host on a visa-controlled network that wants to communicate across its organizational boundary initially engages in a high-level authorization and authentication procedure with the Access Control Servers (ACSs) on both source and destination networks (see figure 1). The need for (and particulars of) ACS authorization is determined individually by the owners of the end-point networks. When a source-destination connection has been approved by an ACS on each network, the ACSs allocate *visas* to the requesting host. The host uses the visas to stamp all

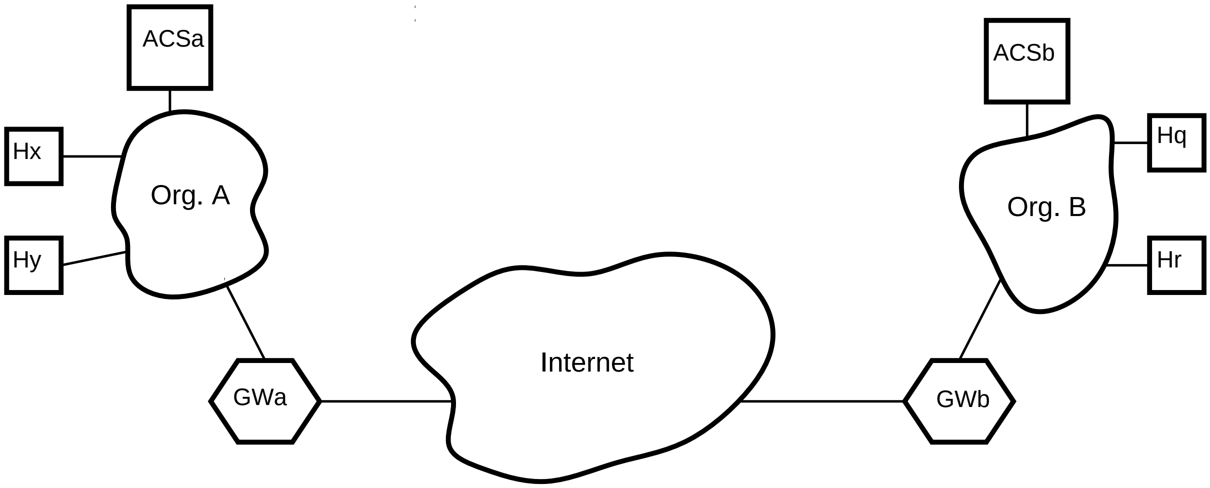

Figure 1: Two interconnected organizations running the visa protocol.

datagrams belonging to that connection. The border gateways (‘‘visa-gateways’’) of the endpoint organizations check all datagrams for appropriate stamping, and pass authorized datagrams until a visa expires or is revoked. Each gateway checks the authorization of a datagram to enter or exit the attached network, not whether the datagram is authorized to travel all the way from source to destination. Visa-gateways may also use visa information to ensure that the proper parties are billed for the cost of carrying the datagrams.

In this report we present two variations of the protocol originally proposed by Estrin and Tsudik [9]. One is an improved version of the original (‘‘stateful’’) protocol, in which the ACSs distribute visas to the gateways involved. The other (‘‘stateless’’) variant avoids the necessity for distributed state, but requires additional encryption steps. We then analyze the drawbacks and advantages of these two protocols based on conventional single-key (‘‘private-key’’) cryptography. (A public-key variation of the stateless protocol is discussed in Appendix I.) This technical report is based, in large part, upon a shorter paper [8]. We have extended the discussion of design issues and added an appendix describing a visa protocol using dual-key (public key) encryption.

1.1. Policies

Visas are a *mechanism* for authenticating the source, destination, and contents of a datagram. Authentication in itself is not an end but a means for implementing a policy, such as access control or accounting. An access control policy, applied to datagrams, requires a gateway to determine if the authenticated parties are indeed authorized to communicate. (Visa protocols described in this report allow only authorized pairs of hosts to be authenticated.) An accounting policy requires a gateway to charge the resources used to an authenticated host; in this context a visa is a certificate that the host has promised to pay its bills. A resource management policy requires a gateway to ensure that the authenticated host has not used up its quota of resources (for example, if datagram charges must be prepaid).

In the visa protocols we describe, gateways do not bear sole responsibility for making policy decisions. By issuing a visa, an ACS has precomputed a decision such as ‘‘these hosts are allowed to communicate,’’ or ‘‘this host can be trusted to pay its bills.’’ The task of a gateway is reduced to ensuring that the visa is valid and is being used correctly; the expensive part of the policy implementation is done once per connection, by the ACS, rather than once per datagram, by the gateway.

This report emphasizes problems of access control; visa protocols described here are designed for that purpose. Accounting and resource management appear to be simpler problems; for example, one may tolerate moderate ‘‘leakage’’, resulting in slightly incorrect bills, if the net result is a lower overhead cost for doing the billing. Also, it is necessary to authenticate only one party (the one who is paying) if the only application is billing. Therefore, in an environment where visas are used for accounting and not for access control, somewhat different protocols may be appropriate; this is the subject of work in progress [10].

1.2. Network environment

We will assume that the internetwork closely follows the model of the DARPA Internet [17], which is substantially similar to the Open Systems Interconnection (OSI) model [21, 24]. The essential features of the environment are:

- Hosts are autonomous and cannot necessarily be trusted.
- Organizational networks are connected by gateways; between any pair of hosts in different organizations there are at least two gateways, one belonging to each of the organizations. Conceptually, the connection between two organizations is a pair of half-gateways connected via a trusted link. Each half-gateway can be trusted by its own organization but not by any other organization.
- All information flows via datagrams. A datagram consists of a *header* that includes addressing information and a data segment that is not intelligible to gateways.
- A datagram may flow through several ‘‘untrusted’’ organizations on its way to the destination.
- Host addresses, both source and destination, can be forged. It is not possible (using hardware methods) to determine reliably which host actually sent a datagram or to prevent a datagram from being seen by unauthorized hosts; in other words, many Local-Area Network (LAN) technologies can be wire-tapped.
- Duplicate datagrams and occasional lost datagrams are natural consequences of using a datagram network. Therefore, if a malicious host duplicates datagrams from time to time, we are willing to accept the covert channel created by this method.

1.3. Design goals

The purpose of the visa protocols is to allow an organization to grant certain privileges to select, trusted hosts and to provide a means for preventing the abuse of such privileges. This is but one component in the provision of complete security. The success of a visa-based system assumes the ability to trust certain hosts not to misuse visas.

Our primary goal is to allow an organization to control the transmission of datagrams to and from hosts in other organizations. If the specific hosts involved can be trusted then we can meet a stronger goal: we can control the transmission of datagrams to and from *a specific* host in another organization. In a datagram network, as opposed to a circuit-switched network, the only information available about a datagram must be attached to the datagram rather than inferred from the route the datagram follows. Therefore, we can state these goals more directly as follows. An organization can guarantee that: a datagram can leave the source organization O_{src} only if O_{src} has authorized the sender to send datagrams to the apparent destination host, and a datagram can enter the destination organization O_{dst} only if O_{dst} has authorized the sender to send datagrams to the apparent destination host. Visa protocols also allow each controlling organization to revoke the privileges it has granted.

Another goal is to add no cost to intra-organizational datagram traffic, nor to impose additional security measures upon hosts that do not participate in inter-organizational traffic. Similarly, we wish to limit the overhead imposed upon organizations who are not concerned with controlling external access.

Finally, we want to minimize the costs imposed by the visa protocols, including: additional per-packet processing time in both hosts and gateways, additional storage requirements for hosts and gateways, extra datagrams sent during connection setup, increments in the length of datagrams (increasing length increases latency and decreases throughput), costs of recovering from gateway crashes, and complexity of the implementations.

The security of visa protocols depends upon the secure operation of participating ACSs, gateways, and hosts, as well as upon secure distribution of visas from ACSs to gateways and hosts. Discussion of mechanisms to implement such security is beyond the scope of this report and can be found elsewhere [15].

1.4. Structure of this report

The remainder of this report is organized as follows. Section 2 describes the notation and the general features of the visa protocols. Section 3 describes an improved version of the original single-key visa protocol (with state information in gateways). Section 4 describes a stateless variation of the single-key protocol. Section 5 presents an evaluation and analysis of the two protocols. Experimental results are discussed in section 6. Section 7 touches upon several design issues that space does not permit us to cover in detail. Finally, section 8 summarizes our findings.

2. Visa protocols

2.1. Notation

We use the notation of Needham and Schroeder [15] to show encryption operations; for example,

$$\{F_0, F_1, ..., F_n\}^K$$

denotes the encryption of a record containing fields F_0 through F_n with key K. For active entities involved in the visa protocols, we use the symbol *H* to denote a host, *O* to denote an organization, *ACS* to denote an Access Control Server, and *GW* to denote an inter-organization gateway. *VKEY* denotes a visa key issued by an ACS for use in creating visas in the stateful visa protocol, and *V* denotes a visa issued by an ACS for use in the stateless protocol.

Any of these symbols can be subscripted *src* to indicate the *source* of a datagram, *dst* to indicate the *destination* of a datagram, *trans* to indicate an organization through which a datagram passes *in transit* between the source and destination organizations, *exit* to indicate the gateway via which a datagram *exits* an organization, and *entr* to indicate the gateway via which a datagram *enters*. For example, H_{dst} denotes the destination host of a datagram, and GW_{exit} denotes a visa-gateway of the source organization through which a datagram leaves that organization's network.

2.2. Components

Both visa protocols involve the following components: visas, access control servers, gateways, and hosts. These components and their responsibilities are described in this section.

2.2.1. Visas

A visa is an unforgeable stamp, created by cryptographic means, that is attached to a datagram. Its presence in a datagram indicates that the datagram is allowed to leave (or enter) an organization's network. A visa can be validated by the gateways of the organization that issued the visa (or that issued the means for its generation)[1]. We describe how visa values are computed in section 2.4.

Each datagram carries at most two visas — one (V_{exit}) for entering and exiting the source organization network, and one (V_{entr}) for entering and exiting the destination organization network. This is necessary because the agents of one organization may not trust the agents of another organization, so source and destination visas for a datagram must be issued separately by the respective organizations[2].

For our experimental modification of the Internet Protocol (IP) [18], visa-related information is carried in the OPTIONS field of the IP header, and so does not affect the normal processing of datagrams (see figures 2 and 3). Datagrams traveling between visa-hosts that do not require visas (as decided by the ACSs of each organization) contain dummy visa values in the appropriate header fields to avoid calling undue attention to those datagrams that warrant visa protection; only the visa-gateways know which datagrams need to contain verified visas. Other IP gateways need not recognize IP options; therefore, visas are transparent to non-visa gateways.

A visa key is allocated to an identifiable source-destination pair. In this discussion we assume that the uniformly-available granularity of control and identification is a host; that is, visas are allocated for (H_{src}, H_{dst}) pairs.

[1] Estrin and Tsudik [9] originally used the term ‘‘visa’’ to indicate the cryptographic key used by the source and gateway to compute the unforgeable stamp. ‘‘Visa’’ now indicates the stamp itself, a usage closer to the English meaning of the word.

[2] In this report we assume the use of *two-way visas*; that is, a single visa key is used to generate visas for datagrams traveling into and out of an organization's network between a particular source-destination pair. However, if an organization wants to carry out separate authorization/authentication dialogs for incoming and outgoing traffic, it may do so — at the cost of double the connection setup overhead.

2.2.2. ACSs

An ACS is a host, usually dedicated for security reasons, that is primarily concerned with access control. Each visa-controlled organization has at least one ACS, responsible for authorizing hosts within its organization to communicate with hosts in other organizations[3]. Multiple ACSs may be necessary for availability and performance reasons. Specific policies regarding who may communicate with whom are embodied within ACSs and are not addressed directly in this report.

Each ACS knows of a number of local visa-gateways that enforce its decisions. ACSs are trusted and assumed to defend against attempted abuse. The security of the overall protocol requires that ACSs be secure and that they employ an authenticated and secure channel for communication with local hosts and gateways.

2.2.3. Gateways

A gateway is a host dedicated (for reasons of performance and security) to packet forwarding. Gateways that use the visa mechanism to enforce access controls are called visa-gateways[4]. All inter-organization connections must be implemented with visa-gateways. Each visa-gateway knows the ACSs in its organization, is willing to accept visa assignments from these ACSs, and trusts their decisions about authorizing and terminating sessions. A visa-gateway allows any external party to communicate with any registered, internal ACS; similarly the gateway allows all registered, local ACSs to communicate with any external party[5].

Assuming that each organization employs a visa-gateway, each inter-organization datagram travels through at least two such gateways. Each visa-gateway is equipped with some means of verifying a visa. Visa protocols described in subsequent sections vary in the particular validation techniques used.

A visa-gateway must scrutinize every packet it receives; datagrams without visas cannot be forwarded (except for those to or from trusted entities of the gateway's own organization). In sections 2.3 and 7.2.2 we describe a mechanism for a gateway to inform a host that visas are required for an inter-organizational connection. Datagrams must be dropped if they contain neither a valid visa nor a ‘‘dummy’’ placeholder visa indicating that a host wishes to be informed via this mechanism.

If the two organizations' networks are not directly connected, packets will pass through the gateways of transit networks. Visa-gateways in a transit network trust each other, and transfer transit packets via secure channels to prevent unauthorized entrance or exit; this is described in

[3] If a participant organization does not have an ACS, its hosts will still be able to communicate with the hosts of other organizations, although the organization in question will be subject to risks associated with the uncontrolled access.

[4] Some gateways may not be involved in visa-enforcement (for example, gateways internal to an organization). We therefore distinguish between *visa-gateways* and *non-visa gateways*.

[5] Such trust is reasonable because ACSs are known to be defensive and to enforce organization policy. Other special servers such as a name server may be given a similar ‘‘carte blanche’’ for external communication if they too are known to be secure.

more detail in section 7.1.3. Non-visa gateways in transit networks treat visa datagrams as regular internet packets.

2.2.4. Hosts

The source host (H_{src}) of an inter-organization connection must obtain a pair of visas, one from the ACS of its organization (ACS_{src}) and one from the ACS of the destination organization (ACS_{dst}). These visas must be included in the header of every datagram sent from H_{src} to the destination host, H_{dst}.

A host, unlike a gateway, does not have to have reliable knowledge of the local ACS's address; this may instead be supplied by a gateway when a host attempts to communicate across the organizational boundary (see section 2.3). The host must still use an authentication protocol to make sure it is really talking to the ACS.

Since datagram reception is a passive operation, the destination host (H_{dst}) is not required to initiate any actions. Of course, in almost any protocol, datagrams flow in both directions, so each host is both a source and a destination. Therefore, to avoid additional overhead we assume that an organization allows its ACS to allocate *two-way* visas automatically *if* authentication of the remote destination is not required.

By themselves, visa protocols do not provide for multi-level security, nor do they eliminate a variety of covert channels. In the absence of additional host-level, non-discretionary controls, an authorized host may still subvert these protocols by ‘‘willingly’’ serving as a conduit for communications between unauthorized hosts.

2.3. Establishing Authorization

In the scheme originally proposed in [9], H_{src}, when opening a connection to H_{dst}, initially sends a datagram with an ‘‘empty’’ visa; if the datagram reaches a visa-gateway, the gateway replies with a REJECT message directing H_{src} to an appropriate ACS. The source host requests a visa from that ACS, which (if necessary) obtains visas from ACSs in other organizations, distributes visa information to the appropriate gateways, and returns the valid visas to H_{src} (and, possibly, H_{dst}). The purpose of the REJECT mechanism is to accommodate hosts that do not know when a visa is required.

However, a host may already know that its intended destination is in a different organization, either because it has previously communicated with that host (and cached the fact that at some point it had received a REJECT), or it may have discovered this through some external mechanism (for example, a name server). If so, it may communicate immediately with an ACS of its own organization to obtain visas, rather than going through the extra two-packet step of attempting to send the initial datagram and receiving a REJECT. The REJECT mechanism is a ‘‘fallback’’ mechanism to inform hosts that they are crossing an organizational boundary, rather than an integral part of connection setup. Note that a REJECT may actually be sent in the middle of a connection, if a visa expires or if a gateway table overflows and active visas are purged. For further detail on the REJECT mechanism see section 7.2.2.

Many inter-organizational connections are brief: in the Internet, for example, most such connections are either electronic mail transfers, which usually involve no more than a dozen datagrams, or name translations, which are even briefer. A visa authorizes datagram transmission between two hosts, not a specific high-level connection. Therefore, we do not require hosts to obtain a fresh visa for every connection, nor do we expect hosts to inform the gateways when a visa-controlled connection terminates. ‘‘Least-recently-used’’ mechanisms can keep gateway caches or tables from filling with stale data. We rely upon the ACSs to enforce specific visa expiration and revocation policies.

2.4. Computing visa values

A visa value must protect against subversion in two ways. First, it must prove that the source of a datagram is authorized to send datagrams to the destination (in other words, that an imposter cannot pose as an authorized source merely by faking its internet address). Second, it must prove that the particular data carried in a datagram is the same data that the source intended to send to the destination. We refer to this second proof as ‘‘data integrity.’’ In general, transformation of a data value to guarantee its provenance is known as a ‘‘digital signature’’ [5, 15, 20].

The integrity of a visa protocol depends on the method by which the visa values are calculated. To avoid ‘‘playback attacks’’, a visa value must be derived from a visa key and some unique property of each individual datagram. In other words, *visa = F(visakey, datagram*) where F is some cryptographically strong one-way (trapdoor) function that computes a cryptographic signature of the datagram. The function chosen for F must have good cryptographic properties, yet be inexpensive to compute. In this report, we assume that *F* is a function such as the DES-based Message Authentication Code (MAC) [3].

Note that the sizes of both visas and visa keys affect the cost of computing visas; they also affect the likelihood that a visa system can be compromised. Unfortunately, although signatures and keys with larger sizes are more resistant to attack, they also increase the cost of computing the value of *F*.

3. Single-key protocol with state information in gateways

This section describes the first single-key variation of the visa protocol, derived from the one proposed in [9]. In this protocol, all non-transit visa-gateways along all possible routes of a datagram must contain an appropriate entry in their tables. Therefore, in order to set up a path between two hosts, each such gateway must communicate with its organization’s ACS to obtain the visa key for the source-destination pair.

This is the distinctive feature that separates this protocol from the stateless protocol discussed later in the report. Here, each component (hosts, ACSs, and gateways) must maintain a *visa-table*, a database of active visa information. An entry in the visa-table pertains to the state information of a specific inter-organization connection. In the stateless protocol, in return for slightly greater per-packet header length and encryption overhead, only the hosts must maintain reliable databases. The stateless-protocol gateways use caches to improve their performance, without requiring extra packet exchanges for database maintenance.

3.1. Creation and distribution of visa keys

In this variant, a *visa key* is a unique value (a cryptographic key) assigned by an ACS to a session between two hosts on distinct networks. The visa value carried in the datagram is computed as a cryptographic signature of a datagram.

Whenever an ACS issues a visa key to a host via a VISAGRANT message, it must also send the visa key to all the border visa-gateways for the organization. If there is more than one ACS for an organization, it might also be useful to distribute the visa information to other ACSs so as to improve the availability of the information in the case of host failures[6].

3.2. Verification of visas

Once the visa keys are in place, H_{src} is able to send datagrams to H_{dst}. Every outgoing datagram addressed to H_{dst} is stamped with both exit and entrance visas, V_{exit} and V_{entr}. Both values are calculated as described above. GW_{exit} and GW_{entr} each calculate V_{exit} and V_{entr} respectively (using the values $VKEY_{exit}$ and $VKEY_{entr}$ from their visa-tables), and compare them with the values found in the datagram. If the two values match, the datagram is passed, otherwise it is REJECTed. This procedure simultaneously verifies that a visa is valid, that a visa allows H_{src} to communicate with H_{dst}, and that the contents of a datagram are those that were sent by H_{src}.

3.3. Connection revocation

Because many protocols do not have an explicit ending phase (for example, the delta-T protocol [11, 23]) an ACS imposes time limits on visas that it issues. The time limits are passed along with the visa keys to the local visa-gateways, which delete the connection's entry from their visa-tables as soon as the connection times out. A host that anticipates exceeding the time limit of its current visa may request a visa extension before the visa expires, in order to avoid reapplication delays. In addition to exceeded time or resource limits, a REVOKE message may be used to revoke a visa. A REVOKE message, triggered by a request from H_{src}, H_{dst}, or an ACS itself, is sent to the appropriate gateways by the ACS. The system is vulnerable to the extent that REVOKE messages may be dropped or delayed.

3.4. Problems

The main drawback of this protocol is that each visa-gateway between a pair of communicating hosts must include a visa-table entry for that host-pair. This is undesirable because:

- The setup mechanism used to get visas into the visa-tables generates a number of extra datagrams. At least two visas must be sent from ACSs to gateways, requiring at least that many datagrams[7].

[6] If one-way visas are used, this same procedure will be carried out in reverse when the first return datagram is generated.

[7] This is in addition to whatever datagrams need be exchanged between the source host and the ACSs involved in order to authorize the visas.

- One of the commonly-held advantages of datagram networks is their ability to efficiently and dynamically switch packets along multiple routes, thus providing some immunity to failed gateways or links, and spreading load across the available bandwidth of a well-connected network. In order to take advantage of routing redundancy when using visas, every local visa-gateway along any potential route is given the visa information at setup time, which can potentially result in *(M+N*) datagrams to be sent by source's and destination's ACSs to their respective visa-gateways (M and N are the number of visa-gateways in each of the organizations' networks).
- A gateway must maintain its visa-table, which can potentially be quite large (*O(n*) in the number of communicating host pairs). Table overflow is not fatal, but when a purged entry turns out to be active, part of the setup mechanism must be reinvoked. The storage overhead of visa-tables is per visa-gateway, not simply per gateway-pair, since the two gateways belong to different organizations and cannot trust one another.
- When a visa-gateway crashes, unless its visa-table is held in stable storage it must be reloaded from the organization's ACS. If the ACS crashes as well, the setup mechanism must be reinvoked for every active connection. The resulting burst in overhead traffic is likely to create congestion.

4. Stateless single-key protocol

In order to avoid some of the problems listed in section 3.4, we present a different visa protocol without the requirement that the gateways know about every visa. This means that we no longer have to pay the costs for setting up and storing visa-tables, although the per-packet processing costs are slightly higher, and revocation is more disruptive.

The primary difference between the two protocols is where the gateways find the authorization information. In the first, or *stateful* protocol, a gateway keeps all authorization information about active connections in its visa-table, which must be loaded by the ACS. In the second, or *stateless* protocol, the authorization information is attached by cryptographic means to each datagram; a gateway needs no authorization database. In effect, the visa information is piggybacked on each datagram rather than being directly communicated between ACSs and gateways. A digital signature system is used to maintain the integrity of this piggybacked information, and caching is used to reduce the amount of encryption overhead.

The particular protocol described here uses a single-key (private-key) cryptosystem such as DES [14]. A public-key version is quite similar; see Appendix I.

4.1. Overview of the stateless mechanism

Suppose that H_{src} in O_{src} intends to send a datagram to H_{dst} in O_{dst}. Before sending the datagram, H_{src} must obtain a ‘‘visa-pair’’, consisting of an exit visa for O_{src} and an entrance visa for O_{dst}. It does so by contacting ACS_{src}, proving its identity, and asking for the appropriate visa-pair. If communication is in fact authorized, ACS_{src} negotiates with ACS_{dst} to obtain an entrance visa for O_{dst}, issues the exit visa for O_{src}, and returns the visa-pair to H_{src}.

When H_{src} sends a datagram to H_{dst}, it first attaches the visa to the datagram (in a manner to be described shortly) in such a way that the visa-gateways can verify that the communication is authorized. This verification is done *solely* by applying cryptographic mechanisms to the datagram; the gateways need not maintain any databases.

A gateway can verify that a visa attached to a datagram is valid because the visa itself is signed by the issuing ACS. Signature is accomplished by encrypting the visa with a key known only to the ACSs and gateways of an organization; this is known as the ‘‘organization key’’. If the cryptosystem is secure, there is no chance of forgery.

It is harder to see how to protect against a malicious host that obtains a valid visa by monitoring the network and attaches this visa to its own datagrams. The trick is to have the source host sign every datagram using a secret session key known only to the source host and the visa-gateways (and to the ACSs trusted by those gateways). This key is embedded in the visa attached to the datagram, but because the visa is encrypted with the organization key, the session key is not available to interlopers. It is available to the visa-gateway as a side-effect of verifying the authenticity of the visa. Because this key becomes known to ACS_{dst} and GW_{entr}, which may not be entirely trustworthy to H_{src}, a new signature key should be generated for each path, and different keys should be used for exit and entrance visas. In this protocol, the function FSIG(*data*) returns a signature of the data (for example, a DES-based Message Authentication Code) using the secret session key, K.

4.2. Creation of visas

H_{src} begins the process of visa creation by generating two signature keys, $KSIG1_{H_{src}}$ and $KSIG2_{H_{src}}$. It then contacts ACS_{src}, proves its identity[8], passes the signature keys to ACS_{src}, and requests a visa-pair for use with H_{dst}. If communication is authorized, ACS_{src} negotiates with ACS_{dst} (passing $KSIG2_{H_{src}}$) to obtain an entrance visa for O_{dst}, issues an exit visa for O_{src}, and returns the visa-pair to H_{src}.

The exit visa issued by ACS_{src} is

$$\mathrm{V_{exit}} = \{\mathrm{H_{src}}, \mathrm{H_{dst}}, \mathrm{KSIG1_{H_{src}}}, \textsc{expiration}\}^{\mathrm{KPRIV_{O_{src}}}}$$

where $KPRIV_{O_{src}}$ is the organization key for O_{src}, and EXPIRATION is a timestamp indicating when the visa expires; this allows an ACS to limit the lifetime of the visas it issues, since (in this protocol) explicit visa revocation is expensive (see section 4.5)[9]. Any gateway belonging to O_{src} can verify that the visa was actually issued by O_{src} by computing $\{Vexit\}^{KPRIV_{O_{src}}}$ and verifying that $KSIG1_{H_{src}}$ produces the data signature for this datagram.

[8] Authentication methods for both single-key and public-key cryptosystems are described by Needham and Schroeder [15, 16].

[9] If the visa is encrypted in separate blocks, the EXPIRATION field must not be in a block by itself, as this would allow a malicious host to ‘‘renew’’ an expired visa by substituting the block from an unexpired visa. The fields of the visa could be staggered across block boundaries to prevent this attack.

The entrance visa issued by ACS_{dst} is similar

$$\mathrm{V_{entr}} = \{\mathrm{H_{src}}, \mathrm{H_{dst}}, \mathrm{KSIG2_{H_{src}}}, \mathrm{EXPIRATION}\}^{\mathrm{KPRIV_{O_{dst}}}}$$

and likewise can be verified by any gateway belonging to O_{dst}.

Note that because the visas are signed using a single-key system, $KSIG1_{H_{src}}$ and $KSIG2_{H_{src}}$ are kept secret.

Once it has a visa-pair, H_{src} can send datagrams. Assume that the datagram that it wishes to send is

$$\mathrm{DGRAM} = \{\mathrm{HEADER}, \mathrm{DATA}\}$$

and that the header is

$$\mathrm{HEADER} = \{\mathrm{H_{src}}, \mathrm{H_{dst}}, \mathrm{SEQNUM}, \textit{other fields}\}$$

where SEQNUM is an ID that is unique to this datagram (these IDs can be recycled after a period at least as long as the expiration time of a visa).

H_{src} must create a ‘‘safe’’ version of the datagram as follows:

$$\mathrm{DSIG_{exit}} = \mathrm{FSIG}(\{\mathrm{HEADER}, \mathrm{DATA}\}, \mathrm{KSIG1_{H_{src}}})$$
$$\mathrm{DSIG_{entr}} = \mathrm{FSIG}(\{\mathrm{HEADER}, \mathrm{DATA}\}, \mathrm{KSIG2_{H_{src}}})$$
$$\mathrm{SAFEHDR} = \{\mathrm{H_{src}}, \mathrm{H_{dst}}, \mathrm{SEQNUM}, \mathrm{V_{exit}}, \mathrm{V_{entr}}, \mathrm{DSIG_{exit}}, \mathrm{DSIG_{entr}}, \textit{other fields}\}$$
$$\mathrm{SAFEDGRAM} = \{\mathrm{SAFEHDR}, \mathrm{DATA}\}$$

$DSIG_{exit}$ and $DSIG_{entr}$ are the data signatures. They are constructed so that all fields of the original datagram whose values must be checked are signed by H_{src}[10]. The safe datagram still includes the contents of the original datagram header in the unencrypted form, so it can be handled by non-visa gateways without additional mechanism. The new fields in the header are purely for the benefit of visa-gateways.

4.3. Verification of visas

Once the safe datagram has been constructed, it is sent along whatever route has been chosen by the usual means, and eventually reaches GW_{exit}. GW_{exit} must verify that (1) V_{exit} is valid, (2) V_{exit} allows H_{src} to send datagrams to H_{dst}, and (3) the contents of the datagram are those that were sent by H_{src}. The first condition is checked by computing

$$\{\mathrm{H_{src}}, \mathrm{H_{dst}}, \mathrm{KSIG1_{H_{src}}}, \mathrm{EXPIRATION}\}=\{\mathrm{Vexit}\}^{\mathrm{KPRIV_{O_{src}}}}$$

and verifying that the EXPIRATION time is reasonable and has not passed; also, if the visa is not valid then the extracted $KSIG1_{H_{src}}$ will be meaningless and consequently will not produce $DSIG_{exit}$. The second condition is checked by verifying that the H_{src} and H_{dst} extracted from the visa are those found in the datagram header. The third condition is checked by reconstructing the original HEADER and using the $KSIG1_{H_{src}}$ extracted from the visa to check that

$$\mathrm{FSIG}(\{\mathrm{HEADER}, \mathrm{DATA}\}, \mathrm{KSIG1_{H_{src}}}) = \mathrm{DSIG_{exit}}$$

If all three conditions are met, then the datagram is what it purports to be, and SAFEDGRAM may be forwarded out of the organization.

[10] It may be necessary to include copies of other header fields in the data signatures; see section 7.1.4.

Eventually the datagram reaches GW_{entr}, which must verify that V_{entr} is valid, V_{entr} allows H_{src} to send datagrams to H_{dst}, and the contents of the datagram are those that were sent by H_{src}. These conditions are checked in the same way as they were checked for the exit visa. If they hold, the datagram can be delivered to H_{dst}.

4.4. Avoiding the cost of visa decryption

Because V_{entr} and V_{exit} are constant for as long as they do not expire, a gateway can cache both encrypted and decrypted values of the visas it uses. When a datagram arrives, a gateway uses the encrypted visa found in the datagram as a key to find a cache entry. If an entry exists, the gateway can use the contents of the decrypted visa, instead of paying the cost of visa decryption (the data signature must still be checked).

The size of the cache, unlike the size of the visa-tables used in the stateful protocol, is relatively unimportant. In the event of cache misses only one additional encryption step per datagram is required, instead of a flurry of message exchanges[11]. If a gateway crashes and reboots, it need only retrieve its organization’s key before continuing to process datagrams; no other messages need be exchanged.

4.5. Revocation

In some cases it might be necessary to revoke a visa. The primary mechanism for revocation is the expiration time contained in the visa’s cleartext. If visas are issued with relatively short lifetimes (on the order of minutes or hours) then it is unlikely that they will need to be explicitly revoked. In the stateful protocol, visas may be revoked explicitly. In the stateless protocol, if an ACS must revoke an unexpired visa, it needs to choose a new organization key and distribute that key to all boundary gateways and ACSs of its organization. Unfortunately, this invalidates all visas issued by that organization; because of this, and because a visa might expire before a connection is finished, all visa users must be prepared to reapply for new visas at any point in a connection.

4.6. Variations on the theme

Visas in the stateless protocol have more internal structure than those in the stateful protocol. Because that structure is visible only to the ACSs and gateways of their issuing organization, this allows some flexibility in their use.

One possibility is to use different cryptosystems for visa generation and signature generation. Since signatures cover entire datagrams, they are best done with an inexpensive single-key system such as DES. On the other hand, visas themselves are relatively small, and given the caching scheme described in section 4.4, visa decryption is done infrequently. Visas could therefore be generated using a public-key system such as RSA. Use of a public-key organizational key instead of a single-key one would reduce the danger of compromising the secret organizational key, since it would never leave the ACS.

[11] The size of a cache entry is twice the size as in the stateful protocol; this is because both cleartext and ciphertext versions of visas are cached.

It is also possible to include additional datagram-header fields in the visa, thereby allowing visas to be issued on, for example, a process-to-process basis rather than a host-to-host basis. Additional informational fields for use by gateways, such as a limit on the packet rate or packet count for the connection, could also be included in the visa. Any additional visa fields, however, increase the processing time in both hosts and gateways, and risk exceeding limits on datagram header size.

5. Evaluation and comparison of single-key protocols

In this section we evaluate and compare the two proposed protocols on the basis of their respective overhead costs. We separate the costs into per-connection costs and per-datagram costs for authorized datagrams. Per-connection costs include the extra datagrams exchanged among visa hosts, ACSs, and gateways; and the storage requirements in gateways and hosts. Per-datagram costs include encryption and decryption, additional packet length due to the visas, and table lookups in hosts and gateways.

5.1. Per-connection costs

In the stateful protocol, there are several kinds of per-connection costs:

1. **Negotiations (supported by datagram exchanges) between H_{src} and the ACSs involved**: At least 2 datagrams must be sent to request the necessary visas, and at least 2 datagrams are required to return the visas to H_{src}[12].

2. **Distribution of visas from ACSs to gateways (more datagram exchanges)**: Visas must be passed to at least two visa-gateways (GW_{exit} and GW_{entr}); this requires at least two datagrams. In total, $M+N$ such datagrams are sent if there are M potential exit gateways and N potential entrance gateways.

3. **Table storage space and maintenance costs**: Storage overhead, consisting of both space and runtime costs, is introduced in this protocol mainly by the need for all participants, but especially gateways, to keep visa-tables. Significant costs are associated with both the space required to store the table, because many connections may be active, and the cost of lookups, since one is performed for every datagram forwarded.

In the stateless protocol, some per-connection costs are reduced:

1. **Distribution of visas from ACSs to gateways**: This is not done at all. The only communication between ACSs and gateways is the distribution of keys at infrequent intervals.

2. **Table storage space and maintenance costs**: Since the only state stored in the visa-gateways is the cache of decrypted visas, which can be refilled at minimal cost, there is no need to maintain a complete table. Table storage space can be allocated to the extent that it is available. Average per-datagram costs will increase if the cache size is so small as to significantly reduce hit ratios.

[12] In practice, any visa protocol may require additional datagrams to be generated in order for H_{src} to authenticate itself to ACS_{src} and ACS_{dst}.

The stateless protocol does require each ACS to perform an encryption operation to create a visa. It is also more expensive, in the stateless protocol, to revoke an unexpired visa because there is no way to do this without revoking all unexpired visas.

Overall, the minimum number of datagrams required to set up a connection in the stateless protocol is lower at least by two (more precisely, by $M+N$) since no visa distribution to gateways is done. In addition, the table storage space and maintenance costs are lower for the stateless protocol.

5.2. Per-datagram costs

The per-datagram costs for visas are the additional fields in datagrams, table look-ups, and cryptographic operations.

Each datagram must carry header fields for both exit and entrance visas. In the stateful protocol, space is required only for two rather small visas, each being a data signature. In the stateless protocol, space is required not only for two data signatures, but also for two rather large visas, each containing (in encrypted form) two source addresses, a signature key, and an expiration time.

In our implementation using 32-bit DES keys, the visas in the stateful protocol together require 8 bytes, while in the stateless protocol, the two visas and data signatures together require 40 bytes (see figures 2 and 3; note that IP requires an additional 4 bytes to indicate the presence of this option). This difference between the stateful and stateless protocols cannot be ignored, but is becoming less significant as network bandwidths increase.

Both protocols require essentially the same number of table lookups; the cache lookups done in the stateless protocol should cost about the same as the table lookups required in the stateful protocol. The only difference is the size of the lookup key, which is twice as large in the stateless protocol.

The cryptographic operations required depend upon the data integrity scheme used. They also depend upon whether the operation involves passing over the entire datagram or over only part of the datagram. For the single-key visa protocols described in this report, the cryptographic costs are: 4 cryptographic operations for the stateful protocol, 6 operations for the stateless protocol without cache hits, and 4 operations for the stateless protocol with cache hits (see table 2). These values include the cryptographic operations at the source host and at both intervening gateways.

Using this analysis we see that, given a reasonable cache hit rate for the stateless protocol, the per-datagram encryption costs are roughly equal for the two single-key visa protocols. The main determinant of cryptographic cost is the strength of the signature function, and thus the vulnerability of the system, rather than the particular visa protocol.

5.3. Summary

In summary, the stateless visa protocol has lower setup costs, possibly lower storage costs for the gateways (depending upon the cache size), but slightly higher per-datagram processing costs than the stateful protocol. A natural consequence of this statement is that the stateless protocol provides for more efficient handling of brief connections, since its setup cost is lower; in particular, the critical path is shorter by one packet-delay. For longer connections, once the difference in setup costs has been amortized and the gateway caches are loaded, the stateless protocol is slightly less efficient because it requires longer packet headers. A choice between the stateless and stateful protocols may depend on other factors, such as the higher cost of selective revocation in the stateless protocol, and the higher cost of gateway table overflow in the stateful protocol. Alternatively, one could implement a hybrid protocol that would employ either the stateless or the stateful protocol depending upon the connection type.

Either protocol depends upon the availability of a high-performance cryptosystem. While public-key methods do not yet appear to meet this need (the fastest commercially available hardware, the Cylink Corporation CY1024, is specified to encrypt up to 2 Kbits/second [4]), single-key systems such as DES are already capable of matching high-speed LAN bandwidths (the AMD AMZ8068 is specified to encrypt up to 1.7 Mbytes/second [1]).

6. Experimental results

The purpose of our experiments was to evaluate per-datagram, connection setup, and overall network costs of visa protocols. This section presents a brief description of our implementation, and analyzes performance measurements of a prototype implementation of both stateful and stateless protocols.

We conducted two sets of experiments, the first on a logical internet in our laboratory at USC, and the second across the DARPA Internet. The laboratory data provide a basis for comparing the relative overheads of the various visa protocols presented. The Internet data prove the feasibility of implementing visa protocols in an operational internet environment, and illustrate the relatively low overhead of visas in a context of relatively high transmission delay.

6.1. Visa implementation

For both laboratory and Internet experiments, visa protocols were implemented as modifications to the IP code in 4.3BSD Unix running on IBM PC RTs[13]. Visa-gateways, hosts, and ACSs all used RTs with 4 megabytes of internal memory. The RTs were connected to an Ethernet with standard Ungerman-Bass Ethernet adaptors. DES encryption, in Electronic Code-Book (ECB) mode, was done in hardware using prototype cards from the Information Technology Center of Carnegie-Mellon University (CMU-ITC). Although the AMD AMZ8068 chip used on the card is specified to encrypt up to 1.7 Mbytes/second [1], the prototype board itself encrypts large data blocks at only 200 Kbytes/second due to slow I/O.

[13] The IBM PC RT scores 2690 on the ‘‘Dhrystone benchmark’’, compared with 2993 for SUN 3/50 and 1577 for Digital Equipment Corporation MicroVax II.

The IP option definition for the stateful visa protocol is depicted in figure 2, and for the stateless visa protocol in figure 3.

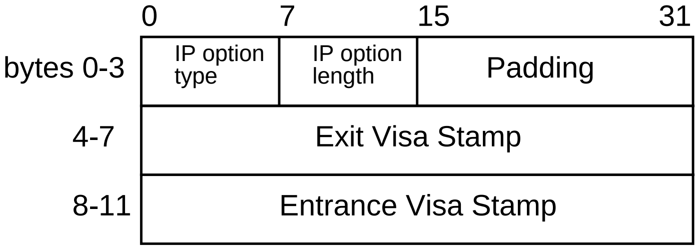

Figure 2: Stateful Visa protocol IP option definition.

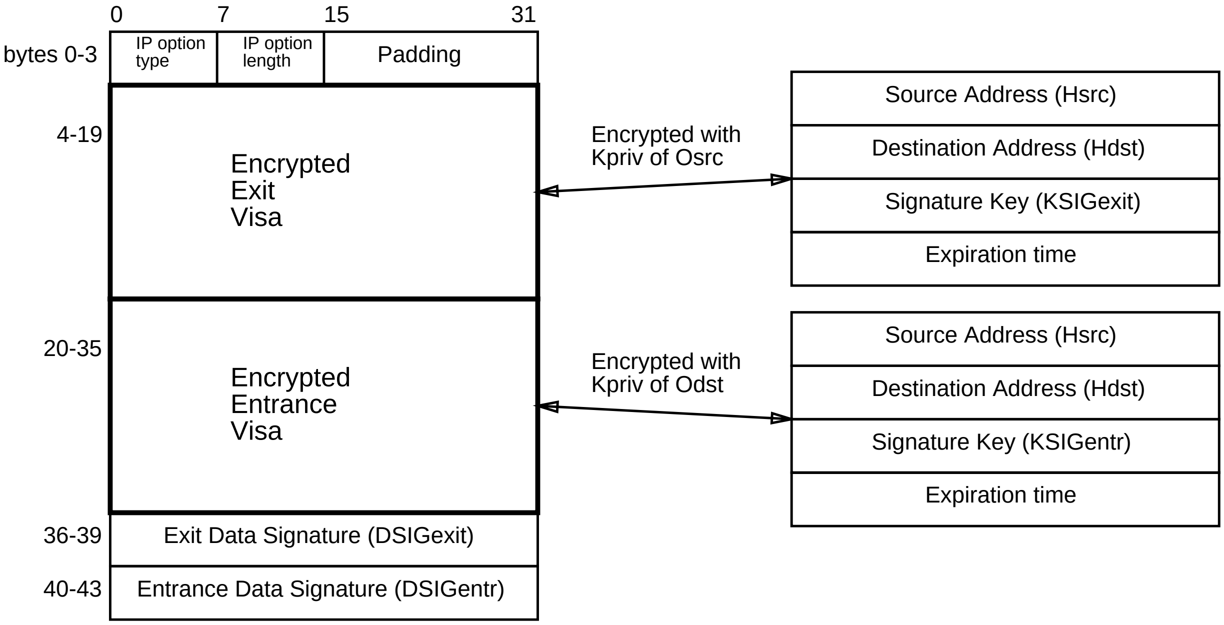

Figure 3: Stateless Visa protocol IP option definition.

We encountered a significant problem with our first implementation of the stateless protocol — we exceeded the maximum IP header size of 60 bytes! In order to implement the stateless protocol within existing IP, we cut down the size of DES keys and data signatures from 64 to 32 bits. Although clever encoding techniques could be used used to pack additional key bits into the header, the stateless protocol is unlikely to coexist with any other IP options, due to the header length limit.

6.2. Experimental configurations

For the laboratory experiments, we created logically separate networks on top of a single physical network by manipulating the routing databases for local hosts (see figure 4).

Our Internet configuration consisted of networks in two universities, USC and UCLA, each connected to the ARPAnet. The visa networks sit within campus networks which each connect to the ARPAnet (see figure 5).

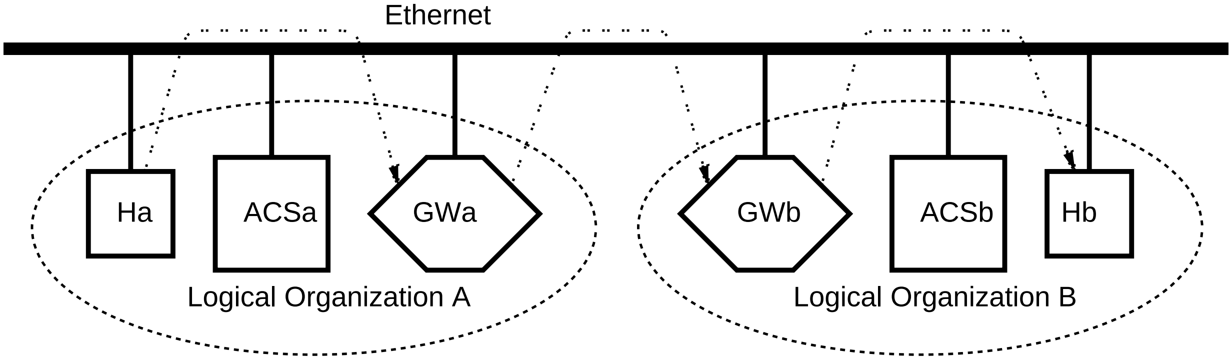

Figure 4: Laboratory configuration. Logically separate networks on a single physical network.

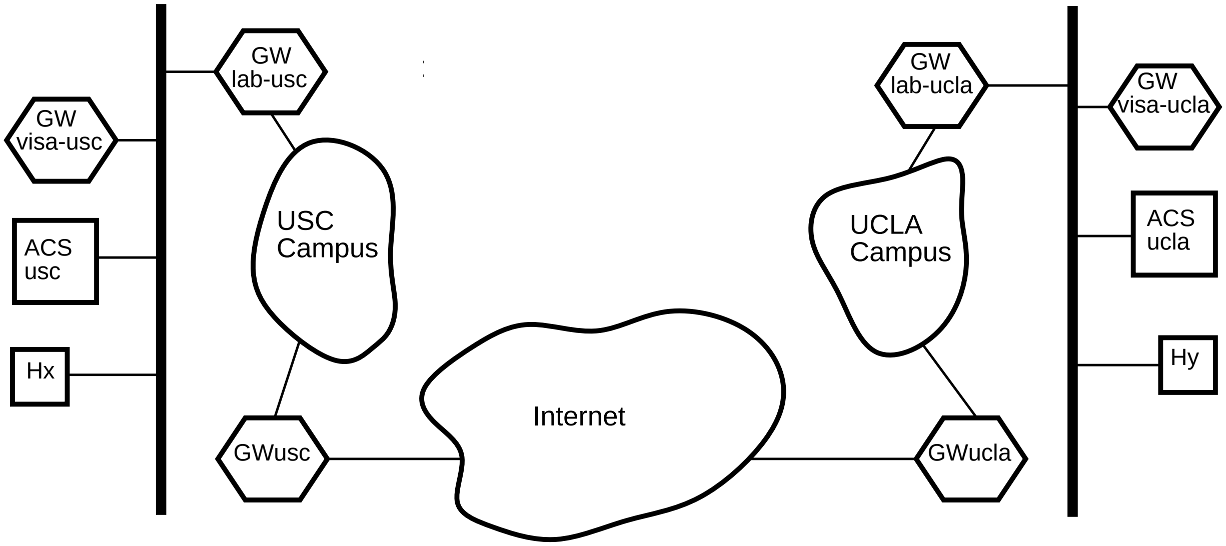

Figure 5: Internet configuration. Physical connections between USC and UCLA visa networks.

6.3. Laboratory measurements

In the laboratory experiment we measured the round-trip datagram times for both visa and non-visa implementations under conditions of similar network load. We measured six protocol variations: no visas, the stateful and stateless visa protocols without encryption (to measure the overhead due to the additional header length of visa packets), the stateful protocol, and the stateless protocol with and without cache hits.

After the initial connection setup, datagram round-trip time was measured using the ICMP Echo protocol [19]. In this protocol, a request datagram travels from H_{src} to the H_{dst}, which immediately returns a reply datagram. We used ICMP Echo instead of an application protocol (such as file transfer or remote login) to isolate, as much as possible, the overhead associated with the visa protocols.

Table 1 shows measured round-trip datagram times for datagrams of varying data length. The results are also presented in graphical form in figure 6. The slight performance advantage of the stateful protocol comes from the shorter header used, compared to the stateless protocol.

Round-trip times (milliseconds)						
	Datagram size (bytes)					
Version	16	64	250	500	750	900
Without Visa	10	11	21	32	42	49
Stateful without encryption	12	14	23	34	46	52
Stateful	19	23	40	61	83	95
Stateless without encryption	14	16	25	36	48	54
Stateless with cache hits	20	25	41	63	84	96
Stateless with no cache hits	24	29	44	66	88	100

Table 1: Round-trip datagram times for the laboratory experiment.

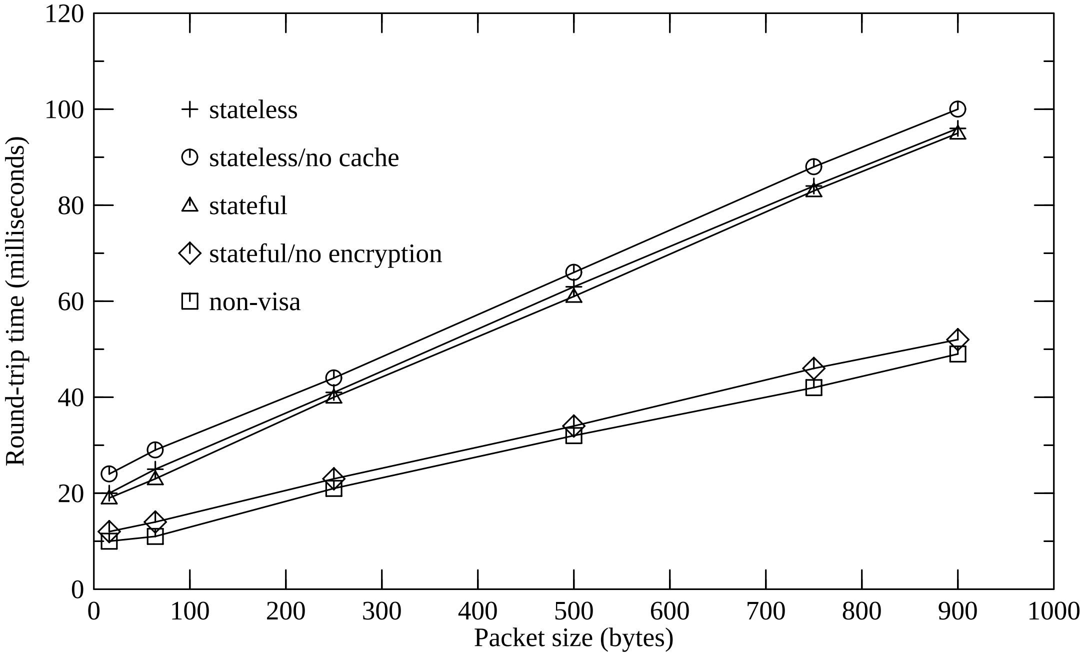

Figure 6: Graphical representation of the laboratory results.

A significant portion of the visa protocol overhead is due to encryption. Table 2 summarizes the per-datagram cryptographic costs for the three variations described in sections 3 and 4. Note that the encryption overhead for the stateless protocol with cache hits is the same as that for the stateful protocol. The table gives one-way overhead; for the round-trip measurements we made, twice as many encryptions are performed.

Actual measurements of the total encryption costs are shown in table 3.

Operation	Stateful version	Stateless version with no cache hits	Stateless version with cache hits
H_{src} creates V_{exit}	X		
H_{src} creates V_{entr}	X		
H_{src} creates $DSIG_{exit}$		X	X
H_{src} creates $DSIG_{entr}$		X	X
GW_{exit} checks V_{exit}	X	X	
GW_{entr} checks V_{entr}	X	X	
GW_{exit} checks $DSIG_{exit}$		X	X
GW_{entr} checks $DSIG_{entr}$		X	X
Total number	4	6	4

Table 2: Per-datagram cryptographic operations.

Encryption overhead (milliseconds)				
	Datagram size (bytes)			
Version	16	64	500	1000
Stateful	8	10	31	53
Stateless with cache hits	8	10	31	53
Stateless with no cache hits	13	15	36	58

Table 3: Per-datagram encryption costs of stateful and stateless visa protocols.

These measurements correspond closely to calculations based upon the number of encryption operations. For example, a round-trip for a 1 Kbyte datagram requires 8 encryptions; at an encryption rate of 200 Kbytes/second, encrypting 8K bytes should take 40 ms. The measured value is 53 ms. The discrepancy comes from per-datagram overhead in using the encryption hardware, which is not reflected in the nominal 200 Kbyte/second rate (measured for encryptions of much larger data blocks).

Since it should be possible to employ the AMZ8068 DES chip to encrypt data at up to 1.7 Mbyte/sec., we also present an estimate, in table 4, of the round-trip times attainable with encryption at the realistically attainable rate of 1.0 Mbyte/sec; this illustrates the importance of faster DES hardware.

The connection setup time for the stateful visa protocol ranged from 30 to 40 ms, averaging about 33 ms. This number represents the time from when the first unstamped datagram is sent to the time that the visa arrives at H_{src}, allowing stamped datagrams to be sent. The REJECT mechanism is employed, but the ACS to GW communication is not secured by encryption or other privacy mechanisms.

Round-trip times (milliseconds)						
	Datagram size (bytes)					
Version	16	64	250	500	750	900
Stateful	12	14	24	36	49	56
Stateless with no cache hits	14	17	26	38	51	58
Stateless with cache hits	14	16	26	38	51	58

Table 4: Projected round-trip times for the laboratory experiment with 1.0 Mbyte/sec encryption rate.

6.4. Internet measurements

The laboratory Ethernet has higher bandwidth, and is more lightly loaded, than the typical inter-organizational network. Therefore, we also conducted experiments over the DARPA Internet to demonstrate the visa protocols in a more realistic context. The path between USC and UCLA includes a highly-congested, low-bandwidth (56 Kbit/sec) hop, as well as several non-visa gateways.

In this configuration, not only is the average delay much higher, but the *variance* in queueing delay is larger than the difference between the visa and non-visa protocol overheads. Consequently, we must emphasize that the results *cannot* be used to compare the various visa protocols to one another, but are presented primarily to demonstrate the reduced significance of visa overhead in the context of other sources of network delay.

In order to obtain the most meaningful average values for visa and non-visa protocols, we ran suites of measurements at different times of the day and week in search of a period of relatively low delay variance. The numbers presented in table 5 (and graphically in figure 7) are from a suite run during a three hour interval when delay varied least. In addition, we excluded the highest delay values when calculating the averages for each protocol.

Round-trip times (milliseconds)						
	Datagram size (bytes)					
Version	16	64	250	500	750	900
Without Visa	120	149	280	441	609	696
Stateful without encryption	131	172	298	468	616	706
Stateful	138	176	316	478	631	719
Stateless with no cache hits	190	228	342	507	661	745

Table 5: Round-trip datagram times for the Internet experiment.

These measurements may understate the cost of visa protocols, since the encryption operations involved were probably being performed in parallel with the transmission of other packets over a congested link. In an uncongested network, through gateways that handle only visa-controlled datagrams, this parallelism might not be available, and the additional end-to-end delay imposed by the visa protocols could be as large as it is in our laboratory experiments.

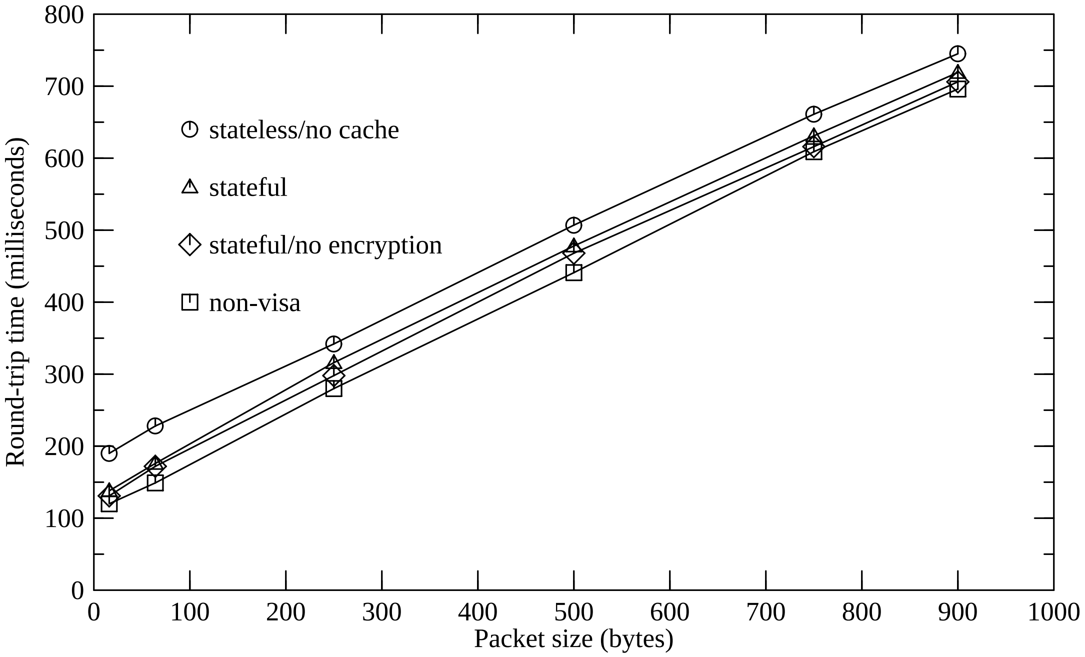

Figure 7: Round-trip travel time across the Internet for datagrams of varying length.

6.5. Analysis

Our results demonstrate the function of the stateful and stateless protocols in both laboratory and actual inter-organizational internet arrangements. They show that, while the overhead for our implementation is significant, it is not prohibitive.

The laboratory results provide a basis for comparing the protocols to one another. They confirm our prediction that the stateless protocol performs nearly as well as the stateful one, on per-datagram delay, only when the cache-hit rate is reasonably high. (In the operating region where the number of active conversations is greater than the size of a gateway’s visa-table, the stateless protocol may perform better than the stateful protocol.) These results also show that comparing the cost of connection setup in the stateful protocol to the cost of setting up cache entries in the stateless protocol, for connections involving only a few datagrams, the stateless protocol may have a small edge. In the steady state, the difference in delay of approximately 2 msec per datagram is due to the additional length of stateless-protocol visa options.

The Internet results demonstrate that when visa mechanisms are added to subsets of existing Internet gateways and hosts both variations of the protocol work without interfering with non-visa, local network or internet, operations. These results also demonstrate that the overhead of visa protocols is much less significant in high-delay environments. The results from the laboratory experiments provide an upper bound on the overhead of visa protocols; the relative overhead in actual inter-organizational networks will be lower, since over such paths the delays due to visa protocols stay fixed, while delays due to transmission and gateway processing generally are higher.

The critical prerequisite for practical application of visa protocols is faster encryption hardware. If encryption rates are not improved by an order of magnitude over that of the equipment we used, visa-related encryption processing will present an overwhelming burden to high-speed gateways that can otherwise process several thousand packets per second. We believe that acceptable encryption rates are feasible with current technology.

7. Other design issues

In this section, we discuss several issues related to the use of visa protocols, in the areas of security, connection setup, and datagram fragmentation.

7.1. Security

A visa protocol is only one component in a system for providing network security. Other mechanisms and policies, used in conjunction with a visa protocol, determine the level of security. Here we consider mechanisms for authenticating the parties to a visa protocol, avoiding denial-of-service attacks, protecting transit organizations, and reducing covert channels.

Security policies, as embodied in an ACS, are as important as security mechanisms. As described in [7], access control decisions are most appropriately made according to a group or class affiliation and associated category sets that determine access rights. The visa protocol itself does not dictate or constrain the particulars of the authorization policies; in this report we are describing the visa interface of an ACS, not the ACS design itself. Regardless of the policy used, the visa mechanism assumes only that a YES/NO decision is provided by the ACS.

Security policies and mechanisms for application-specific access control are left to the end-point hosts and applications; visa protocols address only controlling access to the hosts on a network.

7.1.1. Authenticating hosts and ACSs

Hosts and ACSs must authenticate themselves to each other, in order to prevent an unauthorized host from obtaining a visa from an ACS, or to prevent a malicious host from imitating an ACS and interposing itself between a gateway and an ACS, and thereby providing itself with a visa. The visa protocols described in this report do not specify how a host authenticates itself to an ACS, and vice versa. The authentication process may involve a higher-level conversation between the host and the ACS, which can include the exchange of passwords, keys or other authenticating information. Depending on local policies, the authentication process may require direct communication with the end-user; alternatively, some information may be provided by the system on the user’s behalf.

Each organization could individually choose the authentication mechanism used by its ACSs, but this would require a visa implementation to be tailored to a specific organization, making it hard for vendors to supply turn-key systems. Also, since a host must obtain an entrance visa from a foreign organization, each participant host (or an ACS acting on its behalf) would have to understand the authentication protocol used by the ACS of each organization it communicates with.

It is impractical to expect every source host to ‘‘speak’’ an unbounded set of ACS authentication protocols; it is nearly as impractical to expect each ACS to implement the authentication protocol of each possible foreign organization. The simplest solution is to adopt a standard protocol for host-ACS (and ACS-ACS) communication. Such a standard is a subject for future work.

7.1.2. Denial of service

Visa protocols present the possibility of certain novel denial-of-service attacks. For example, a malicious host could interpose itself between a victim host and an ACS, and ‘‘issue’’ visas that would prove useless. Interposition can be prevented by a suitably strong host-to-ACS authentication protocol.

The REJECT mechanism described in sections 2.3 and 7.2.2 also introduces potential denial-of-service attacks. A malicious host could send spurious REJECTs to a visa host, since the normal behavior of the visa host upon receipt of a REJECT is to interrupt the connection in progress until a new visa is obtained. This attack can be prevented by the use of an authentication protocol between hosts and gateways, such as public-key digital signatures on REJECT messages.

Standardization of these defenses is a subject for future work. Environments where denial-of-service is of sufficient concern should use secure means of authentication between hosts, gateways, and ACSs.

7.1.3. Protecting transit organizations

Recall that a transit organization is one through which a datagram flows, but that does not contain the source or destination hosts of that datagram. In the future, we anticipate the existence of policy-sensitive routing protocols to protect transit networks, while visa protocols would be used to protect endpoint networks [2]. However, in the interim, if visa protocols are used to protect transit services, then when a datagram flows through the gateways of a transit organization O_{trans}, they must ensure that the datagram is in fact what it appears to be, rather than a forgery designed to bypass the information-flow controls of O_{trans}.

There are two problems that must be solved:

1. A datagram may leave O_{trans} appearing to have originated in another organization O_{src}, but might actually be a forgery generated by a host in O_{trans} that is not authorized to send datagrams to O_{dst}.
2. A datagram may enter O_{trans} apparently on its way to another organization O_{dst}, but might actually be meant for and received by an unauthorized host in O_{trans}.

These problems only arise for visa-gateways at the borders of O_{trans}, and only for transit organizations that wish to control information flow[14]. In a well-organized internetwork, most transit traffic should travel over common carriers or similar backbone networks. Carrier organizations presumably have no interest in controlling information flow (as opposed to resource control

[14] In certain network technologies, such as a point-to-point network, it is not possible to fake a source address at the data-link layer, or to receive a datagram meant for another host. In this case, the two problems discussed in this section do not arise.

and billing, which are separate issues), so they need not expend effort to solve these two problems.

To protect against illegal exits, we require that an in-transit datagram be *sealed* by the gateway through which it enters an organization. When an in-transit datagram tries to leave an organization, the exit gateway must verify that it is properly sealed. If it is, then it cannot have been generated within this organization and thus the exit-visa need not be checked.

To protect against illegal entrances, a gateway must not allow an apparently in-transit datagram to arrive at any untrusted host within its organization. If the network can be tapped by any host, the only secure way of doing this is to encrypt the entire datagram.

Transit-sealing could be done by adding a signature, computed as in section 4, to the datagram header at the entrance gateway. Since both parties to the sealing are visa-gateways of O_{trans}, they trust each other and can use a single signature key to compute the seal. But, since the entire datagram is being encrypted anyway to avoid unauthorized reception while it traverses O_{trans}, there is no need to perform a separate sealing encryption. This method, in effect, encapsulates transit datagrams in a secure point-to-point protocol between gateways of O_{trans}, adding a cost of 2 encryption operations for transit sealing and unsealing. (If there are N_T transit organizations along the path of a datagram, the total addition cost is $2N_T$ encryption operations.) The gateways can use any suitably efficient and secure encryption mechanism for this purpose.

7.1.4. Covert channels via header fields

A data signature method must cover not only the data segment, but any datagram header fields whose authenticity cannot be checked by the gateways. Any unchecked field leaves a potential covert channel, since a malicious host could copy a valid datagram, change the unchecked field, and send the modified copy without raising suspicion.

We could protect against this by including the entire datagram header under the data signature, but in most internetworking protocols there are some header fields that are modified by the gateways, and hence cannot be included in the signature. (All gateways may have to modify the header, not just visa-gateways, and we assume that non-visa gateways cannot regenerate the signature. If a public-key method is used, not even visa-gateways can do so.)

In the IP protocol, there are two such variable fields. One is the header checksum; this cannot be forged because it is a function of the other fields in the header, and is already recomputed by each IP gateway. The other is the 8 bit wide ‘‘Time-To-Live’’ (TTL) field, used to prevent datagrams from following routing loops. The TTL must be decremented by each gateway, and must never be incremented. A malicious host could communicate approximately 6 or 7 bits per datagram by manipulating the initial value of the TTL field in copies of otherwise validly-signed datagrams.

If this covert channel is considered too broad, there are a number of steps that can be taken. The visa-gateways could make use of their knowledge of network topology to reduce the TTL value to near the minimum necessary for the datagram to safely arrive at H_{dst}. Since the diameter of most internetworks is closer to 15 than 255, this reduces the width of the covert channel to perhaps 1 or 2 bits per datagram; unfortunately, since most gateways cannot know the exact route a datagram will follow, this approach might lead to complete loss of datagrams that follow a slightly longer route than expected. The use of ‘‘Strict Source Routing’’ [18] might sufficiently constrain the routes, but is not currently practical in the Internet.

Alternatively, since the visas themselves will stop certain kinds of loops (a datagram cannot reenter O_{src}, nor leave O_{dst}, because it does not carry visas to do so), GW_{exit} and GW_{entr} could each set the TTL to its maximum value. This erases any manipulation, but it violates the letter of the IP specification, and might confound protocols that use the TTL field to limit the lifetime of a datagram.

7.2. Connection setup

There is a tradeoff between the cost and flexibility of connection setup mechanisms. Shortcuts can be programmed into the visa-gateways to reduce the overhead. At the same time, the use of lazy evaluation increases the overhead for the sake of increased flexibility.

7.2.1. Reducing the cost of connection setup

In the simplest case, when H_{src} wishes to initiate a bi-directional connection it acquires a pair of visas, sends a datagram to H_{dst}, and then must wait for the destination to go through the process of acquiring its own pair of visas. This can result in long connection setup times, and in particular it makes it much harder to predict the round-trip time for the connection. It would be more efficient if the return visas could be issued simultaneously with the forward visas.

If a public-key visa protocol is used (see Appendix I), this is easily accomplished. Suppose that H_{src} has H_{dst}’s public key. (It might have obtained it from the name server used to find H_{dst}’s address, and in any case would need it to protect its communications with H_{dst}). When H_{src} requests its own visas, it can also pass H_{dst}’s public key to ACS_{src} and request reverse visas for H_{dst} to use. If the ACSs approve, they return both pairs of visas to H_{src}. There is no problem in doing so, since only H_{dst} can make use of its visas. H_{src} may then pass them to H_{dst} in the initial datagram of the connection.

If a private-key stateless visa protocol is used, H_{dst} must generate its own secret signature keys, and so it must be involved in the generation of the return visas. ACS_{dst} must ask H_{dst} to participate in creating visas perhaps before H_{dst} knows that it is about to be called by H_{src}. This is not a serious problem, but it requires additional asynchrony at H_{dst}.

The private-key stateful visa protocol, and other private-key visa protocols that do not require hosts to generate their own keys, may avoid involving H_{dst} in this asynchronous manner. In this case, ACS_{dst} could generate the required keys and send them in a signed, encrypted ‘‘envelope’’ back to H_{src} for conveyance to H_{dst}.

7.2.2. Details of the REJECT mechanism

As described in section 2.3, one approach to connection setup is to use the REJECT mechanism to discover the need for visas, rather than to require H_{src} to know in advance if a visa is required. This is how a host acquires a visa using the REJECT mechanism:

1. When a host, H_{src}, wants to communicate with a another host, H_{dst}, it initially sends a datagram addressed to H_{dst} with a special ‘‘dummy’’ visa in the datagram header. This eliminates the need for each host to know if a visa is required for communication with a given destination. The normal routing mechanism is used to choose a path for the datagram.

2. The datagram reaches a gateway, GW_{exit}, on the boundary of O_{src}. GW_{exit} traps the datagram and upon discovering that it is not stamped with a valid visa, drops it and sends a special REJECT message back to H_{src}. The REJECT message, among other things, contains the addresses of one or more ACSs trusted by that gateway, eliminating the need for H_{src} to reliably know the address of an ACS. If a gateway receives a datagram that has neither a valid visa nor a dummy visa, then the source host presumably does not understand the visa protocol at all; instead of sending a REJECT message, the gateway sends an ICMP ‘‘Destination Unreachable’’ message.

3. Upon receiving the REJECT, H_{src} sends a special REQUEST message to an ACS (ACS_{src}) that contains addresses of H_{src} and H_{dst}[15]. If the ACS chosen is down, H_{src} should choose a different ACS from the list in the REJECT message, and try again. Because H_{src} and GW_{exit} may be ‘‘neighbors’’ of different ACSs in their organization, allowing H_{src} to choose the ACS not only eliminates the need for GW_{exit} to know which ACSs are up, but can improve performance because H_{src} might have to exchange more datagrams with ACS_{src} than does GW_{exit}.

4. ACS_{src} authorizes and authenticates H_{src} (and maybe H_{dst}) and sends a similar REQUEST message to H_{dst} (on behalf of H_{src}). Because this datagram is sent to H_{dst}, ACS_{src} and the gateways of O_{src} do not need to know the addresses of the foreign ACSs. GW_{exit} passes this datagram because each visa-gateway passes datagrams to and from its local ACSs. ACS_{src} records in its database that this REQUEST is pending; pending entries are flushed periodically.

5. If the destination organization is not visa-controlled, the REQUEST message is received by H_{dst} which promptly replies with special VISAGRANT message containing a ‘‘dummy’’ visa. Otherwise, the REQUEST message is trapped by GW_{entr}, the gateway via which the datagram enters O_{dst}. GW_{entr} is programmed to reroute the REQUEST message to ACS_{dst}.

6. ACS_{dst} receives the REQUEST, and, after authenticating and authorizing H_{dst} (and maybe H_{src}), sends either $VKEY_{entr}$ (for the stateful protocol), or V_{entr} (for the stateless protocol) back to ACS_{src} in a special VISAGRANT message (and to GW_{dst} for the stateful protocol).

7. ACS_{src} receives the VISAGRANT message from ACS_{dst} and now issues either $VKEY_{exit}$ (for the stateful protocol), or V_{exit} (for the stateless protocol). It sends both $VKEY_{exit}$ and $VKEY_{entr}$ (or V_{exit} and V_{entr}) to H_{src} (and to GW_{src} for the stateful protocol), also by means of a VISAGRANT message. The ‘‘pending REQUEST’’ records in the databases of both H_{src} and ACS_{src} may be removed at this time.

8. H_{src} adds the visa information contained in the VISAGRANT message to its database, associated with the foreign host H_{dst}.

[15] During the time between steps (1) and (3), H_{src} may continue to send datagrams to H_{dst} and they will result in REJECT messages sent back by GW_{exit}. However, in order to prevent confusion, H_{src} should ignore all but the first REJECT message. To do this, H_{src} keeps a database of pending REQUESTs that it has issued.

In the stateful variant of the visa protocol, during this procedure the visa information must also be distributed to the gateways; this is described in more detail in section 3.1.

After this procedure, all the interested parties have the visa information they need.

Note that neither H_{src} nor ACS_{src} is required to use the REJECT mechanism to acquire the appropriate ACS addresses. Each is free to address a REQUEST message directly to the appropriate ACS, if its address is known. (That is, H_{src} sends its REQUEST to ACS_{src}, and ACS_{src} sends its datagram to ACS_{dst}.) This can reduce latency of visa setup by up to 3 packet transfers (since in the REJECT protocol all of the packet transfers occur serially).

7.3. Visas and fragmentation

In a number of internetworking protocols, including IP, a gateway may have to fragment a datagram if it cannot be transmitted in a single packet. Data signatures complicate the use of fragmentation; with data signatures, the fragments must appear to have been signed by H_{src}, but the signatures would have to be computed by the fragmenting gateway. With public-key signatures, this is impossible, since only H_{src} can compute the signature. Even with private-key visas, fragmentation is a problem because only a visa-gateway can do it while preserving the data signatures.

Fragmentation is at best a necessary evil [12]; it is almost always better to set datagram sizes at H_{src}, to make the best possible use of the available bandwidth and to provide acknowledgements for each transmission unit. In this document, rather than try to devise a protocol for fragmenting visa-carrying datagrams, we insist that the source host avoid sending datagrams that will have to be fragmented. (Methods have been proposed for accommodating fragmentation [22].) A gateway should assist in this by returning an error datagram when it is unable to transmit a datagram without fragmenting it; in fact, the IP protocol includes a mechanism for doing so (through the ICMP ‘‘Destination Unreachable/fragmentation needed’’ message [19]).

8. Conclusions

We have described two variations on the original visa scheme [9] for controlling datagram flow between organizations. The first involves direct transfer of authentication information between ACSs and gateways, state maintenance in the gateways, and a cryptographic mechanism to mark authorized datagrams. In the second variation, authentication information is ‘‘piggybacked’’ on the controlled datagrams, rather than directly communicated between ACSs and gateways, and the gateways maintain caches rather than true databases. The two protocols vary in the number of datagrams required to authorize a connection, their behavior under load and during failure recovery, and the amount of encryption performed on each datagram; experimental results illustrate these tradeoffs.

Adaptation of visas in actual internetworks depends on several prerequisites: resolution of a few design choices and parameters, the widespread availability of inexpensive, fast, and secure cryptosystems, and sufficient coordination among organizations to make the system worthwhile. Visas are at best a robust *mechanism* for enforcing information flow control *policies*; the choice and specification of these policies will present difficult and interesting problems.

9. Acknowledgements

Our thanks to numerous colleagues in the Internet community for their comments on previous drafts, including: Bob Braden, Annette deSchon, Michael Schroeder, Robert Sansom and the anonymous reviewers.

Particular appreciation to: Paul Crumley of the Carnegie Mellon University ITC for making available the DES hardware and assisting us in putting it to work, Mark LaRouche of UCLA for providing temporary access to UCLA network facilities in support of our Internet experiments, and to Mark Brown of the USC Computing Services for helping to configure our experimental laboratory internet.

Appendix I. Public key protocol without state information in gateways

It is possible to construct public-key variants of visa protocols. In this appendix, we show how this might be done. Public-key methods have certain inherent advantages over private-key methods, but today they are much more expensive to implement; consequently, practically available data rates are inadequate.

The public key variant of the stateless protocol is quite similar to the single-key stateless protocol (see section 4). As before, it begins with H_{src} contacting ACS_{src} to request the issuance of a visa-pair; in this case, instead of passing two private keys, H_{src} provides its (single) public key.

The exit visa issued by ACS_{src} is

$$\mathrm{V_{exit}} = \{\mathrm{H_{src}}, \mathrm{H_{dst}}, \mathrm{KPUB_{H_{src}}}, \mathrm{EXPIRATION}\}^{\mathrm{KPRIV_{O_{src}}}}$$

$KPUB_{H_{src}}$ is either passed by H_{src} to ACS_{src} when it asks for a visa, or more likely is known to ACS_{src} as part of the mechanism it uses to confirm the identify of H_{src}. EXPIRATION is a timestamp indicating when the visa expires.

The entrance visa issued by ACS_{dst} is similar:

$$\mathrm{V_{entr}} = \{\mathrm{H_{src}}, \mathrm{H_{dst}}, \mathrm{KPUB_{H_{src}}}, \mathrm{EXPIRATION}\}^{\mathrm{KPRIV_{O_{dst}}}}$$

and likewise can be verified by any gateway belonging to O_{dst}.

H_{src} then creates the ‘‘safe’’ version of the datagram as follows:

$$\mathrm{SAFEDATA} = \{\mathrm{H_{src}}, \mathrm{H_{dst}}, \mathrm{SEQNUM}, \mathrm{DATA}\}^{\mathrm{KPRIV_{H_{src}}}}$$
$$\mathrm{SAFEHDR} = \{\mathrm{H_{src}}, \mathrm{H_{dst}}, \mathrm{SEQNUM}, \mathrm{V_{exit}}, \mathrm{V_{entr}}, \mathrm{KPUB_{H_{src}}}, \textit{other fields}\}$$
$$\mathrm{SAFEDGRAM} = \{\mathrm{SAFEHDR}, \mathrm{SAFEDATA}\}$$

SAFEDATA is constructed so that all fields of the original datagram whose values must be checked are signed by H_{src}; we refer to this as the *data signature*. The safe datagram still includes the contents of the original datagram header in an unencrypted form, so it can be handled by non-visa gateways without additional mechanism. H_{dst} must be able to invert the ‘‘signing’’ of the data segment, which is why a copy of $KPUB_{H_{src}}$ is passed in ‘‘unsigned’’ form in SAFEHDR. The other new fields in the safe header are purely for the benefit of visa-gateways.

Once the safe datagram has been constructed, it is sent along the chosen route by the usual means, and reaches gateway GW_{exit}. GW_{exit} must verify that the exit visa is valid, the exit visa allows H_{src} to send datagrams to H_{dst}, and the contents of the datagram are those that were sent by H_{src}. The first condition is checked by computing

$$\{\mathrm{H_{src}}, \mathrm{H_{dst}}, \mathrm{KPUB_{H_{src}}}, \mathrm{EXPIRATION}\} = \{\mathrm{V_{exit}}\}^{\mathrm{KPUB_{O_{src}}}}$$

and verifying that the EXPIRATION time has not passed. Also, if the visa is not valid then the extracted $KPUB_{H_{src}}$ will be meaningless and consequently will not produce correct values for H_{src} and H_{dst} when the third condition is checked. The second condition is checked by verifying that the H_{src} and H_{dst} extracted from the visa are those found in the datagram header. The third condition is checked by using the $KPUB_{H_{src}}$ extracted from the visa to compute

$$\{\mathrm{H_{src}}, \mathrm{H_{dst}}, \mathrm{SEQNUM}, \mathrm{DATA}\} = \{\mathrm{SAFEDATA}\}^{\mathrm{KPUB_{H_{src}}}}$$

and then verifying that the fields in the datagram header (specifically H_{src}, H_{dst}, and SEQNUM) match those extracted.

If all three conditions are met, then the datagram is what it purports to be, and SAFEDGRAM can be forwarded out of the organization. The procedure followed when the datagram reaches GW_{entr} is analogous.

Because O_{dst} may want to ensure that no unauthorized hosts on its network see the contents of the datagram, GW_{entr} may have to encrypt the data segment one more time, using $KPUB_{H_{dst}}$ so that only H_{dst} can read the datagram[16]. GW_{entr} can acquire $KPUB_{H_{dst}}$ by using some method external to the visa system, or ACS_{dst} can supply this key by including it as an additional field in V_{entr}.

When SAFEDGRAM finally reaches H_{dst}, the actual data segment can be extracted using the copy of $KPUB_{H_{src}}$ in SAFEHDR, perhaps after inverting the encryption done by GW_{entr}. By postponing the final decryption to this point, we provide the assurance of digital signatures on an end-to-end basis with minimal additional cost. Alternatively, GW_{entr} is the last gateway that needs, for the purposes of inter-organizational information-flow control, to invert the signature on the data segment. Therefore, it can reconstruct the original, unsigned datagram at this point (since it has already done the decryption).

This variant has the advantage over private-key signatures that H_{src} need do one less encryption (generating one signed data segment instead of two signature values). On the other hand, H_{dst} (or GW_{entr}) does have to decrypt the data segment in order to read it.

[16] O_{dst} cannot trust H_{src} to encrypt the data so that only H_{dst} can read it, so this encryption can only be done at GW_{entr}.

References

[1] Advanced Micro Devices.
Advanced Micro Devices MOS Microprocessors and Peripherals Data Book.
Advanced Micro Devices, Inc., Sunnyvale, CA, 1987.

[2] David Clark.
Policy Routing in Internet Protocols, Version 1.1.
To be published as an Internet RFC and available from the author at MIT Laboratory for Computer Science, 545 Technology Sq., Cambridge MA 02139.

[3] D. W. Davies and W. L. Price.
Security For Computer Networks.
Wiley, New York, NY, 1984.

[4] W. Diffie.
The First Ten Years of Public-Key Cryptography.
Proc. IEEE 76(5):560-577, May, 1988.

[5] W. Diffie and M. E. Hellman.
New Directions in Cryptography.
IEEE Transactions on Information Theory IT-22(11):644-654, November, 1976.

[6] Deborah Estrin.
Interconnection Protocols for Interorganization Networks.
IEEE Journal on Selected Areas in Communication SAC-5(9):1480-1491, December, 1987.

[7] Deborah Estrin.
Controls for Inter-Organization Networks.
IEEE Transactions on Software Engineering SE-13(2):249-261, February, 1987.

[8] Deborah Estrin, Jeffrey C. Mogul, and Gene Tsudik.
Visa Protocols for Controlling Inter-Organization Datagram Flow.
IEEE Journal on Selected Areas in Communication , 1989.
In press (Special Issue on Secure Communications).

[9] Deborah Estrin and Gene Tsudik.
Visa Scheme for Inter-Organization Network Security.
In *Proceedings of the IEEE Symposium on Security and Privacy,* pages 174-183. IEEE, April, 1987.

[10] Deborah Estrin and Gene Tsudik.
Issues in Secure Policy Routing.
TR 88-54, University of Southern California, Computer Science Department, December, 1988.

[11] J. G. Fletcher and R. W. Watson.
Mechanisms for a Reliable Timer-based Protocol.
Computer Networks 2(4/5):271-290, September/October, 1978.

[12] Christopher A. Kent and Jeffrey C. Mogul.
Fragmentation Considered Harmful.
In *SIGCOMM87,* pages 390-401. ACM SIGCOMM, Stowe, VT, August, 1987.

[13] J. Mracek.
Network Access Control in Multi-Net Internet Transport.
S.B. Thesis, M.I.T. Department of Electrical Engineering and Computer Science.
June, 1983.

[14] National Bureau of Standards.
Federal Information Processing Standards.
Publication 46, National Bureau of Standards, 1977.

[15] R. M. Needham and M. D. Schroeder.
Using Encryption for Authentication in Large Networks of Computers.
Communications of the ACM 21(12):993-998, December, 1978.

[16] R. M. Needham and M. D. Schroeder.
Authentication Revisited.
Operating Systems Review 21(7):7, January, 1987.

[17] M. A. Padlipsky.
A Perspective on the ARPANET Reference Model.
RFC 871, SRI-NIC, September, 1982.

[18] Jon Postel.
Internet Protocol.
RFC 791, SRI-NIC, September, 1981.

[19] Jon Postel.
Internet Control Message Protocol.
RFC 791, SRI-NIC, September, 1982.

[20] R. Rivest, A. Shamir, and L. Adelman.
A Method for Obtaining Digital Signatures and Public-key Cryptosystems.
Communications of the ACM 21(2):120-126, February, 1978.

[21] Andrew S. Tanenbaum.
Computer Networks.
Prentice-Hall, Englewood Cliffs, NJ, 1981.

[22] Gene Tsudik.
Internet Datagram Authentication: Implications of Fragmentation and Dynamic Routing.
IEEE Journal on Selected Areas in Communication , 1989.
In press (Special Issue on Secure Communications).

[23] R. W. Watson.
Delta-T Protocol Preliminary Specification.
UCRL 52881, Lawrence Livermore Laboratory, November, 1979.

[24] H. Zimmermann.
OSI Reference Model - The ISO Model of Architecture for Open Systems Interconnection.
IEEE Transactions on Communications COM-28:425-432, April, 1980.

Table of Contents

List of Figures

List of Tables