A Reference Model for Firewall Technology and its Implications for Connection Signaling*

J. Bryan Lyles
Computer Science Laboratory
Xerox PARC
3333 Coyote Hill Road
Palo Alto, CA 94304–1314
lyles@parc.xerox.com

Christoph L. Schuba
COAST Laboratory
Purdue University
1398 Department of Computer Sciences
West Lafayette, IN 47907–1398
schuba@cs.purdue.edu

Abstract

This paper concentrates on one particular aspect of providing communication security: firewalls between domains of trust. We argue that signaling support for providing scalable security services is a design requirement. On this basis we outline a reference model for firewall technology. It captures the current state of the art and proves suitable for connection-oriented high-performance networks.

The architecture is an improvement in network management and provides a controlled exposure of the internal network structure to the outside, and transparency to the user. Its components are endpoint authentication, call admission control, connection authentication, audit, and a distributed architecture with centralized policy. The paper discusses implications of this reference model for the design of signaling protocols.

1 Introduction

Data communications networks have become an infrastructure resource for businesses, corporations, government agencies, and academic institutions. However, new technologies introduce new threats, and networking not only puts corporate resources, plans and data at risk, but ultimately the company's reputation and potential survival. Protection from network–enabled threats cannot be achieved by a single technology or work practice. While this paper concentrates on a particular aspect of providing communication security, firewalls between domains of trust, we want to stress that a balanced approach to network protection draws from several other fields: such as physical security, personnel security, operations security, and communication security.

For the purpose of this paper we adopt the following working definition for firewall technology:

Firewall Technology: *Mechanism to help enforce access policies about communication traffic entering or leaving networks.* (1)

In classic firewall technology access control security services for distributed systems were provided in an ad hoc fashion. To date there is neither a well designed reference model nor any theoretical background.

*Published in *Proceedings Open Signaling Workshop*, Columbia University, New York, NY, Oct. 14-15 1996.

The integration of classical TCP/IP networks and new highspeed network technologies, such as ATM, offers new opportunities to address some of the current shortcomings of firewall technology. Additionally, the development of new networking technologies offers the opportunity to investigate the question of what capabilities it must provide and where.

We are stepping back and are asking what security services need to be present in connection oriented networking technologies to support a wide variety of applications ranging from native ATM devices to complex distributed systems. In particular, we are investigating what basic mechanisms need to be available in their supporting signaling protocols.

1.1 Previous Work

The value of firewall technology has long been recognized. Several research papers describe the different approaches ([2], [1], [14], [16], [21], [26],[10], [8], [13] and [4]). In the past two years a few text books on the topic have been published ([5], [23] and [9]).

Little has been published on firewall issues in connection-oriented communication networks. In a standards contribution, Lyles ([17]) motivates the development of authenticated signaling as part of the ATM signaling standards: a fundamental prerequisite for our approach. Smith and Stidd ([24]) were the first to propose concrete solutions to the problems of user authentication and billing for services and products provided by end systems in B–ISDN. Further development and prototyping efforts are underway by several groups, e.g., Tarman et al. at Sandia National Laboratories ([19] and [25]), Bullard et al. at Fore systems, and the ATM Forum ([11]).

Tarman et al. at Sandia National Laboratories focused mainly on hardware and software encryption in high speed networks, as well as signaling support for encryption, authentication, and key exchange. They did not put any emphasis on the issue of network layer access control.

"Domain Type Enforcement (DTE)" was introduced by Boebert and Kain in [6]. It investigates issues in access control that are relevant to our approach. The DTE approach is actively being used by a group at Trusted Information Systems.

2 Background

2.1 Current Firewall Technology

Firewall technology in TCP/IP internetworks provides a mechanism to help enforce access policies on communication traffic entering or leaving networks. Usually an "inside" network domain is protected against an "outside" untrusted network, or parts of a network are protected against each other. A firewall is a security architecture placed on the data transmission path between networks, or on a bastion host placed in a demilitarized zone network between the inside and the outside.

In current firewall practice, security policies are translated into simple lists of rules. Each rule explicitly or implicitly allows or denies data through the firewall based on some semantic interpretation of the data contents. Rules interact with each other, for example through their order. Different types of firewalls operate on different layers of abstraction of passed data: network layer (packet–filtering), transport layer (circuit–level), and application layer (application–level).

2.2 Packet Filters

At the lowest level of abstraction, data is transmitted in packets, called IP datagrams in a TCP/IP network. In a packet–filtering firewall each datagram that arrives at the firewall router is passed to a packet filtering mechanism. The filter discards or forwards packets according to specified rules based on the fields of the TCP/IP packet header, e.g., source and destination addresses and port numbers. The rules operate solely on the contents of the datagram, because no context is maintained across datagrams that belong to the same connection.

2.3 Circuit–Level Gateways

Circuit–level firewalls group packets into connections, e.g., TCP connections, by maintaining state across packets. This association is typically done by inserting a proxy process into the connection. An alternative approach is to build "on the fly" tables at the packet forwarding process based on examining the SYN/ACK flags of TCP packet headers. In the case of "on the fly" table creation, the firewall implements a policy of forwarding packets belonging to connections initiated from within the firewall, but not trusting connections initiated from the outside. If proxies are present, processes on the inside cannot directly establish connections to destinations on the other side of the firewall either, but rather connect to the proxy. The proxy then uses access rules to determine if the connection should be established or blocked. Circuit–level gateways can implement elaborate access control mechanisms, including authentication and additional client/proxy protocol message exchanges. Programs initiating connections must be modified in order to use circuit–level proxies. Only minor changes are necessary, but the availability of source code, the heterogeneity of system platforms, the distribution of programs, and the education of the user population make this a difficult task.

2.4 Application–Level Gateways

Application–level firewalls interpret the data in packets according to particular application protocols. Essentially they are proxies: special purpose implementations of the applications whose purpose is to add security features and to prevent the applications from being misused. They are application specific: for each application, a different application–level firewall must be provided.

2.5 Discussion of Firewall Technology

Security firewalls neither provide perfect security nor are free of operational difficulties. They do not protect against malicious insiders. There is no protection against connections that circumvent the firewall, e.g. modems attached to computers inside the firewall. There is only limited protection against tunneled connections and novel attacks. Because current practice does not provide a check of internal system configuration against the firewall access lists, changes in system configurations may inadvertently produce security holes. Firewalls offer only limited protection against data driven attacks, such as the contents of downloaded Java applets. Because of the reactive character of the concept of firewalls there is only little reason to believe that effective protection against novel attacks is guaranteed. Indeed, there is a history of attack scenarios that initially succeeded against firewalls and that prompted advances in the state of the art.

Firewalls are useful because many currently deployed computing systems and networked applications do not provide strong security. Some argue that firewall technology is more than just a retrofit patch for shortcomings in systems and protocol design. Even in the presence of secure hosts and network protocols, firewalls are desirable because they serve as a centralized focus of security policy and as a place to collect comprehensive security audits. They improve administrative control and network management via controlled exposure of internal network structure, topological flexibility, and transparency to the user. Lastly, and perhaps most importantly, firewalls represent a technology that is widely accepted, available, and justifiable to management in charge of purchasing decisions.

Overall it is important to understand that in spite of their advantages firewalls are neither a panacea nor a replacement for good host security, but an additional protection mechanism.

3 Firewall Reference Model

In this section we describe a reference model for firewall technology in accordance with Definition (1). The reference model is designed to provide strong basic security services and integration with other existing security mechanisms, in particular firewall approaches as mentioned in Section 2.1.

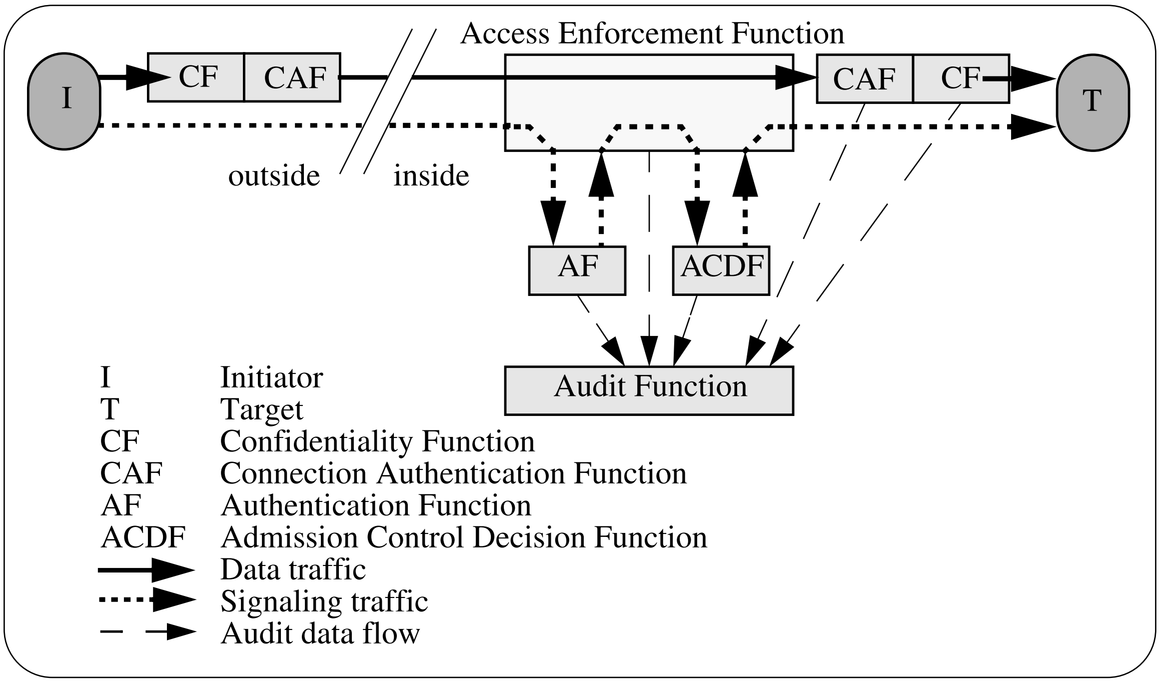

Figure 1: Reference Model of Firewall Technology

Figure 1 depicts the high level view of our generic model of network security. It combines the security services of endpoint authentication, connection authenticity, data integrity, (data confidentiality,) call admission control, and accountability through the application of a combination of security functions, such as authentication, and audit.

Services are displayed as functional blocks. As we explain in the following sections, our model is more distributed than this compact representation suggests. The coupling between functions can be tightly integrated to very loose, functions may be replicated and distributed across a large distributed system. The concepts described in the figure are not restricted to an end–to–end, end–to–intermediate, or intermediate–to–intermediate discussion, nor to unilateral authentication. As we describe in the following paragraphs, an iterative application of this figure allows us to argue about a combination of endpoints, as well as mutual authentication.

Figure 1 is useful in explaining the conceptual interaction between components. An initiator attempts a certain access request to a target – in our framework a connection establishment request. The access enforcement function located in the communication path between these two principals requests the authentication of the initiator and eventually the access control decision, and acts upon the results of these functions. Access enforcement function, authentication function and access control decision function all have write access to the audit function.

3.1 Assumptions

Our reference model takes advantage of the notion of connection oriented communication. Although we discuss the reference model primarily with respect to the asynchronous transfer mode (ATM), it can be applied to other connection oriented protocols, such as TCP, including those with soft state connections, such as RSVP flows. We assume the existence of a secure public key distribution infrastructure and a naming service. Furthermore, we assume that the binding between communicating principals and their associated keys cannot be compromised. We require the integrity of the trusted computing base and the appropriate strength of utilized cryptographic algorithms and parameters.

To satisfy Definition (1), we require five essential elements:

1. Endpoint Authentication
2. Domain Based Call Admission Control
3. Connection Authentication
4. Audit
5. Centralized Policy with Distributed Service and Enforcement

3.2 Endpoint Authentication

All connections traversing the network perimeter are positively identified by their authenticated endpoints, which can be labeled "unknown".

Authentication provides assurance of the claimed identity of an entity. Entity authentication provides corroboration of the identity of a principal, within the context of a communication relationship. A principal is an entity having one or more distinguishing identifiers associated with it. Authentication services can be used by entities to verify the purported identities of principals. Examples of principals in our framework are network service access points (NSAPs), and possibly higher layer entities strongly bound to those NSAPs, such as server processes or even users.

It is necessary that the identifier be interpretable at any place along the connection establishment that is involved in the authentication and access control process. If identifiers have global significance this requirement is trivially satisfied. However, this is usually not necessary. If an endpoint cannot be authenticated, or its identifying label cannot be interpreted, its identity is labeled as "unknown". It is the responsibility of the security policy to comprehend this case.

Distinguishing identifiers are required for unambiguous identification within a security domain. They can be distinguished at a coarse level by virtue of group membership, or at the finest degree of granularity identifying exactly one entity. The term claimant is used to describe a principal for the purpose of authentication. The authentication verifier is an entity which is or represents the entity requiring an authenticated identity. Authentication between a claimant and a verifier is called unilateral authentication. An entity involved in mutual authentication will assume both claimant and verifier roles.

Authentication methods rely on one or a combination of the following principles: something known (e.g., password), something possessed (e.g., security token), some immutable characteristic (e.g., biometric identifier), trust (e.g., third party information), or context (e.g., address of principal).

There are authentication schemes with and without trusted third party involvement (see [20, Figures 1,2]). In the simple case no trusted third party is involved. The claimant establishes his identity with the verifier through a direct exchange of authentication information. Third parties can get involved in a variety of ways: in–line (a trusted entity intervenes directly in an authentication exchange between the claimant and the verifier, e.g., ftp proxy), on–line (one or more trusted parties are actively involved in every instance of an authentication exchange, e.g., Kerberos), off–line (one or more trusted parties support authentication without being involved in each instance of authentication). See [20, Figures 3,4,5]. Our architecture combines the two schemes of in–line and off–line authentication. In–line authentication is used to execute the authentication protocol between claimant and intermediary. In our model authentication between intermediary and verifier is based on trust, because they belong to the same domain of trust and administration. Off–line authentication is utilized by the intermediary or verifier for verification of public key certificates.

3.3 Domain Based Call Admission Control

Call admission control decisions are based on explicit policies that act on the security domain membership of connection endpoint identities.

Our model of access control includes two main principals: the initiator[1] and the target. Initiators can be human beings or computer–based entities that access or attempt to access targets. The connection establishment is the subject of access control requests. Targets represent computer–based or communications entities to which access is attempted. The access enforcement function is located on any possible path between initiator and target and is part of the trusted computing base.

The access control decision function decides upon the access request by the initiator to the target. Information taken into account by the access decision function are the identities of initiator and target, the access request, contextual information, as well as the security policy implemented.

Domain based access control takes a hierarchical approach to dealing with the scaling issues of access control. It is infeasible to specify security policies exhaustively in terms of all possible participating entities in a globally interconnected system. Domain based access control allows to represent the structural relationships among entities in a set theoretic approach, e.g., users can belong to a group of engineers, or files can belong to a certain project.

A fair amount of research effort has been spent in investigating the semantics of access control. Several publications propose languages as tools for the specification of access control policies and their enforcement. A rich set of theories and existing implementations can be utilized. The idea of Domain Type Enforcement as one particular instance of domain based access control goes back to [6].

Authentication and access control are inherently related. If we want identifiers to identify as high level an entity as possible, the labels can become arbitrarily complex. In general it is infeasible for a low level authentication module in the network layer to perform its operation on this scale, because certain high level information necessary to perform the access control decision is not present at the network layer. This problem is described in [18] where Moffet and Sloman argue that general, application-independent access control is infeasible. In [22] Röscheisen and Winograd give an example that shows that the approach of security negotiation in all but the simplest cases becomes a complex coordination problem that can easily lead to deadlock situations. Participants in the negotiation do not know a priori what information the peer requires to make the local access control decision. Including all data that can possibly be needed in the access request is prohibitively expensive and possibly violates privacy concerns of the requester.

Because of these issues our model needs to be one of verified delegation. It is the role of the firewall in complex transactions to ensure that communications occur only with entities (e.g., programs) which are trusted to enforce the security policy appropriately, e.g., a ftp server whose file system security is known to be appropriate for anonymous ftp access.

3.4 Connection Authentication

Connection authentication provides assurance about the authenticity of sender of data in a connection and the integrity of the transmitted data. This becomes important once endpoint authentication and call admission control have been performed. The identity of the sender needs to match the initially authenticated identity. It is important to note that integrity assurance is part of connection authentication. Although possible, and often desired for other valid reasons, it is not necessary to assure integrity through encryption of the whole data stream - a common misconception. Integrity and confidentiality services serve different purposes and have very different characteristics.

3.5 Audit

All components of the system need the opportunity to record information in a consistent manner for use by notification utilities, audit trail analysis, intrusion detection engines, and billing agents.

3.6 Centralized Policy with Distributed Service and Enforcement

The elements described above are distributed and enforced along the path of the connection. In particular, they do not have to be located directly at the network perimeter as classically required by firewall

[1] In our model claimants for authentication purposes are identical with initiators for access control.

technology. The main argument here is scaling.

Indeed, the avoidance of the network perimeter becoming a performance bottleneck (as is currently the case) is a compelling argument for moving or distributing some of the functions further into the network. Consider the scenario where access control verification and enforcement can be negotiated between the network perimeter (or possibly a sequence of modules along the way of the data connection) and the end system. After the initial authentication there might be a cascade of access control decisions to be performed, based on the granularity of access control enforcement at a certain module.

One special case of this it the possibility of complete trust into a certain protocol stack, running on machines inside the boundary of trust, implementing all firewall security services.

In a different scenario the distribution of functionality might be configured at runtime, based on the capability of some involved modules. For example, depending on the capabilities of the operating system that is running on the end system, the access control decision made at a previous node can be different. That approach has operational advantages over current firewall technology. It does not depend on an absolute trust relationship among all components in the protected network. It also allows more "plug and play" type configuration, where system capabilities are detected at runtime.

Non trusted protocol stacks, or stacks that implement only subsets of the firewall security services can be identified at the network boundary via their endpoint identifier in the CONNECT message. Access will be restricted appropriately.

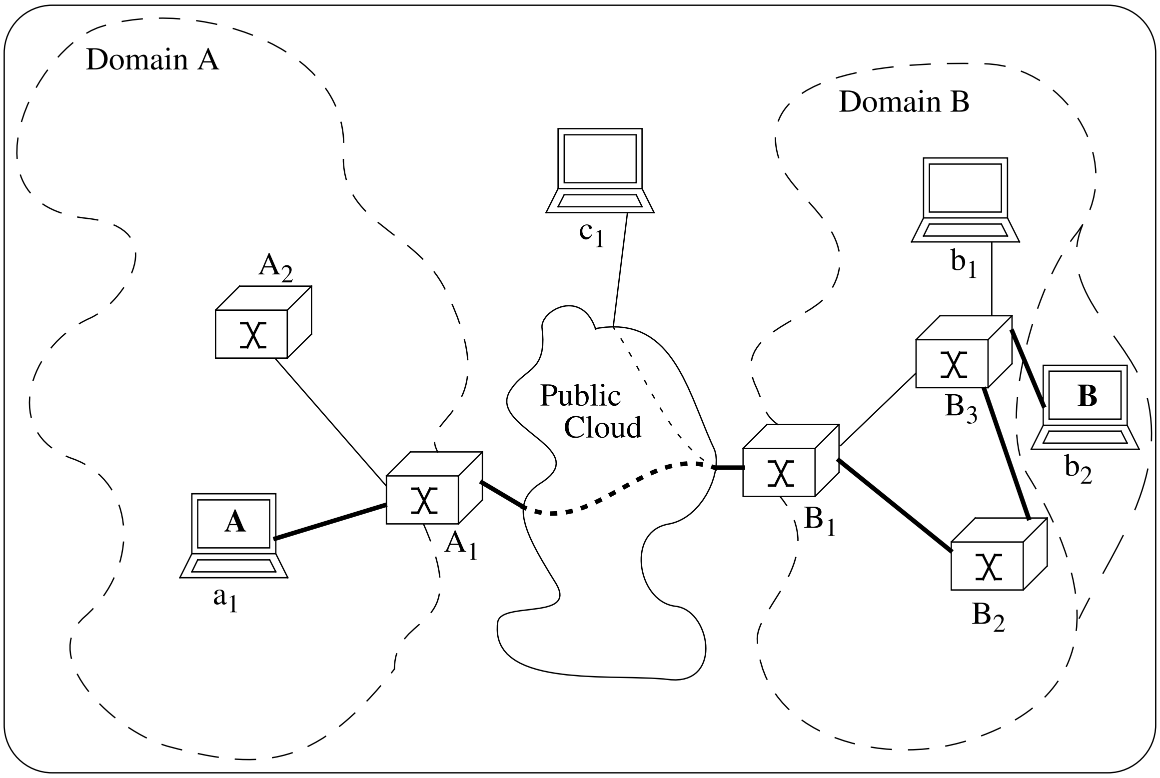

Figure 2: Reference Example

3.7 A Generic Scenario and Example

The following generic network access scenario illustrates the application of the previously explained elements of the reference model.

- Originating principal A initiates a connection to destination principal B. A and B are located on different sides of the network perimeter that is being protected. A and B can be any of a large set of principals, such as hosts, network interfaces, processes, users, etc.
- As part of the connection attempt the originator creates its credentials for the authenticated call setup, e.g. at a_1.
- After the connection request arrives at the destination's network boundary the authentication module located at that virtual boundary verifies the authenticity of A. The connection establishment attempt may be terminated at this point if the authentication fails.
- The call admission control decision anywhere between the authentication module and principal B calculates and enforces its access control decision. The access might be refused at this point and the connection torn down. The decision regarding the access control module to be invoked can be dynamic, based on a negotiation between the boundary switching process and the end point in question.
- A positive access control decision might call for further action, such as the validation of the functionality of an enforcement module at B, or the exchange of enforcement parameters with it.
- Once the call is established, A and B can communicate. If so desired connection authentication is provided on the data stream on an end–to–end basis.

4 Implications for Signaling Protocols

The previous section described what capabilities are required in order to build firewall technology according to Definition (1). This section investigates their implications for signaling protocols. We discuss the implications for Q.2931, the ATM signaling protocol. Our claims and conclusions are validated by a prototype implementation of the reference model.

Conceptually, we need to provide security services that affect

- call establishment and clearing protocol messages,
- data traffic, and
- the signaling system as a whole.

The relationship between these and the fundamental elements is as follows: Call establishment and clearing protocol messages are affected by "endpoint authentication" and "domain based call admission control". Data traffic is affected by "connection authentication". The signaling system as a whole needs to provide support for "audit", and the "centralized policy and distributed service enforcement".

4.1 Endpoint Authentication

Endpoint authentication requires the introduction of a new information element into the signaling protocol. This information element contains endpoint identification information and identifies which authentication protocol and algorithms are used, as well as protocol and algorithm specific information. Appendix A serves as an example for an authentication information element that was used in our prototype implementation[2]. The field `message type` identifies the signaling message type that is being authenticated, e.g., SETUP. Ideally, no information present in any other mandatory information element should be replicated in the authentication information element.

[2]Note that some information was replicated that is present in other information elements, simply for practicality of implementation.

Unilateral or mutual authentication can be achieved by a variety of well known authentication protocols within the limitations of the Q.2931 protocol message flow, i.e., one message authentication. One such protocol is described in Appendix B. It relies on public key cryptography and synchronized clocks. Unilateral authentication of the initiator of a connection is achieved by one authentication information element added to the initial SETUP message. For mutual authentication, the destination of a connection would generate a CONNECT message with an additional authentication information element certifying the authenticity of the destination.

This proposal is therefore sufficient for unilateral and mutual authentication between any two participants: end–to–end, end–to–intermediate, intermediate–to–intermediate. Authentication verification (unilateral and mutual) does not need to be performed by the final destination in the authentication process, but can be performed by any intermediary system with access to the signaling message on the destination's behalf. Verification and any possible action prompted by the result of the verification can therefore be delegated to any trusted intermediary, in particular "firewall switches" located at the logical network boundary.

Nested authentication (authentication of several entities within one message, e.g., end–to–end AND end–to–intermediary) is a simple extension to our approach, where multiple authentication elements can be present within one signaling protocol message. If the data covered by the authentication information element is chosen carefully, assurance for the integrity of a large portion of the protocol message is given. In conjunction with access control this mechanism can be utilized to protect against denial of service attacks by authenticating the source of RELEASE messages.

4.2 Call Admission Control

Call Admission Control requires the signaling system to perform or use the services of the access control function and enforce its result. If no authentication information is present in the protocol messages, a default "unknown" identity is used as the subject for the access control decision request.

Both initiator and destination of a connection have opportunities to enforce access policies, as well as intermediate nodes. Connection release needs to be subject to access control in addition to authentication. It is not sufficient to record who released a connection, but to ensure it happened according to security policy.

The degree of coupling between access enforcement function and access control decision function is important. A collocation of the two modules may have advantages with respect to efficiency and timeliness, however, an access control decision function that serves several access enforcement functions may reduce the need to distribute access control information.

4.3 Connection Authentication

Connection authentication provides assurance about the authenticity of the sender of data in a connection and the integrity of the transmitted data. Connection authentication can leverage off endpoint authentication to determine the initiator of a connection. However, connection authentication still needs to validate that all data received at the destination was indeed sent by the originally authenticated initiator. This protects against threats of active wiretapping, such as connection highjacking, e.g., [12]. The second aspect of connection authentication is the assurance of integrity of transmitted data.

Both components can be provided by the application of cryptographic mechanisms, e.g., a periodically transmitted hash value of previously sent data, signed by a key shared among the two connection endpoints. Such keys can easily be derived from public key information utilized by the initial endpoint authentication together with an update message, such as proposed in SKIP ([3]). At the receiving side delivery of data is verified, which can introduce jitter. It is important to choose the granularity of the data unit for which integrity is enforced carefully in order to optimize the tradeoffs involved between the introduction of jitter, computational overhead, and the amount of security assurance gained. A natural choice is to use the protocol frame size as data unit, e.g., AAL5 frames. Each frame would be followed by an OAM cell containing the digital signature for the preceding frame. Rekeying can also be achieved through an OAM cell, again in a similar fashion as in SKIP ([3, Section 1.9]).

Connection authentication is maintained on a per connection basis. "Signature" messages containing the digital signature for preceding data units and periodic "key resynchronization" messages are sufficient mechanisms to provide for connection authentication.

Confidentiality

According to Definition (1) confidentiality is not part of our reference model. However, one can argue that a confidentiality security service is an important service in any security architecture. We therefore include this brief section on confidentiality. The discussion of implications for signaling for a confidentiality security service are similar to the discussion on connection authentication. Typical data units subject to encryption are ATM cells or whole frames. There is no necessity of "Signature" OAM cells, but for "Resync" messages containing initialization vectors to accomplish recovery from encryption synchronization loss. See Tarman et al. [25, Section 8.1] for details.

4.4 Audit

Audit does not affect the signaling protocol flow, however it requires any implementation of a signaling protocol to provide the necessary calls to the audit function. We cannot stress strongly enough the importance of a secure audit system for the purpose of billing, intrusion detection, and any form of post mortem or audit trail analysis.

All the above discussed mechanisms and implications can be added to a signaling protocol, such as Q.2931 without prohibiting usage of non security aware Q.2931 implementations. This allows for a gradual transition towards a secure infrastructure.

5 Conclusions

Our study shows that the concept of firewall technology is viable in connection–oriented highspeed networks, such as ATM.

We consider the security services of endpoint authentication, domain based call admission control, connection authentication, and audit as essential elements of our reference model for firewall technology. Furthermore, the flexibility of choice of location of services and their enforcement, together with a centralized security policy allow our model to scale to large networks.

The paper investigated the implications of this model on the design of signaling protocols and the associated signaling system. The discussion and our prototype implementation show that simple extensions to the signaling protocol Q.2931 and the data message flow are sufficient to implement this reference model.

Acknowledgments

We are very grateful for funding provided by Sprint Corporation. Dr. Eugene H. Spafford provided valuable guidance and technical advice. We thank Dr. Sandeep Kumar for a review of this paper.

References

[1] Frederick M. Avolio and Marcus J. Ranum. A Network Perimeter with Secure External Access. In *Second Symposium on Network and Distributed System Security (NDSS)*, San Diego, February 1994. Internet Society (ISOC).

[2] Frederick M. Avolio and Marcus J. Ranum. A Toolkit and Methods for Internet Firewalls. In *Technical Summer Conference*, pages 37–44, Boston, June 1994. USENIX.

[3] Ashar Aziz, Tom Markson, and Hemma Prafullchandra. *Simple Key-Management For Internet Protocols (SKIP)*. Internet Engineering Task Force, Reston, VA., August 1996. Internet Draft (work in progress).

[4] Mary L. Bailey, Burra Gopal, Michael A. Pagls, Peterson Larry L., and Prasenjit Sarkar. PathFinder: A Pattern-Based Packet Classifier. In *Proceedings of the First Symposium on Operating System Design and Implementation (OSDI)*, Monterey, CA, November 1994. USENIX.

[5] Steven M. Bellovin and William R. Cheswick. *Firewalls and Internet Security*. Addison–Wesley Publishing Company, Inc., 1994.

[6] W. E. Boebert and R. Y. Kain. A Practical Alternative to Hierarchical Integrity Policies. In *Proceedings 8^{th} National Computer Security Conference*, Gaithersburg, MD, September 1985.

[7] CCITT. *Recommendation X–509 The Directory Authentication Framework*. CCITT, 1988.

[8] D. Brent Chapman. Network (In)Security Through IP Packet Filtering. In *Proceedings of the Third USENIX UNIX Security Symposium*, Baltimore, MD, September 1992. USENIX.

[9] D. Brent Chapman and Elizabeth D. Zwicky. *Building Internet Firewalls*. O'Reilley & Associates, Inc. Sebastopol, CA., September 1995.

[10] William R. Cheswick. The Design of a Secure Internet Gateway. In *Proceedings of the Third USENIX UNIX Security Symposium*, Baltimore, MD, September 1992. USENIX.

[11] ATM Forum Technical Committee. *Phase I ATM Security Specification (Draft)*. ATM Forum, August 1996.

[12] Computer Emergency Response Team (CERT), Carnegie Mellon University, Pittsburgh, PA. *IP Spoofing Attacks and Hijacked Terminal Connections*, January 1995. CA-95:01.

[13] Annette DeSchon and Danny Cohen. The ISI "Tunnel". Technical Report ISI/SR–93–358, University of Southern California, Information Sciences Institute, October 1993.

[14] Digital Equipment Corporation (DEC). *Screening External Access Link (SEAL) Introductory Guide*, 1992.

[15] ATM Forum. *ATM User–Network Interface Specification, Version 3.1*. Prentice–Hall, Englewood Cliffs, New Jersey, September 1994. Q.2391.

[16] David Koblas and Michelle R. Koblas. SOCKS. Socks package documentation, 1994.

[17] Bryan Lyles. Requirement for Authenticated Signaling, April 1994. ANSI Committee T1S1.5/94–118.

[18] J. D. Moffet and M. S. Sloman. Content-dependent Access Control. *Operating Systems Reviews*, 25(2):63–70, April 1991.

[19] Lyndon G. Pierson. Integrating End-to-End Encryption and Authentication Technology into Broadband Networks. Sandia National Laboratories, March 1996.

[20] Karen T. Randall, editor. *Recommendation X–811 Information Technology – Open Systems Interconnection – Security Frameworks for Open Systems: Authentication Framework*. International Telecommunications Union, 1993.

[21] Marcus J. Ranum. Thinking About Firewalls. In *Proceedings of the Second International Conference on Systems and Network Security and Management (SANS-II)*, April 1993.

[22] R. Martin Roescheisen and Terry Winograd. A Communication Agreement Framework for Access/Action Control. In *Proceedings of the Symposium on Research in Security and Privacy*, Oakland, CA, May 1996. IEEE.

[23] Karanjit Siyan and Chris Hare. *Internet firewalls and network security.* New Riders Pub., Indianapolis, IN, 1995.

[24] Ted Smith and John Stidd. Requirements and Methodology for Authenticated Signaling, November 1994. ATM Forum 94–1213.

[25] Thomas D. Tarman, Lyndon G. Pierson, Joseph P. Brenkosh, Barbara J. Jennings, Edward L. Witzke, and Marylou Brazee. Final Report for the Protocol Extensions for ATM Security Laboratory Directed Research and Development Project. Technical Report 96–0657, Sandia National Laboratories, March 1996.

[26] G. Winfield Treese and Alec Wolman. X Through the Firewall, and Other Application Relays. In *Technical Summer Conference*. USENIX, June 1993.

A Authentication Information Element

We define the authentication information element according to [15, Section 5.4.5.1 and Figure 5–23].

A.1 Authentication Information Element – Header

byte	coding	meaning
00	fe	information element identifier[3]
01	80	bit 8 ext=1
		bit 7-6 = 0 – coding standard: ITU-T
		bit 5 flag = 0 – in agreement UNI 3.1
		bit 4 = 0
		bit 3-1 IE action indicator = 0 – in agreement UNI 3.1
02-03	01 fc	0x01fc = 508_d size of IE. In total 512 bytes.
04-1ff	xx xx	508 bytes available for the authentication value

A.2 Authentication Information Element – Body

name	len	type	description
opcode	1	u_char	Opcode for requests
result	1	u_char	Result code
Protocol specific data:			
*message type	1	u_char	Message type
*protocol	1	u_char	protocol identifier
*nonce_no	4	long	Nonce number
*nonce_time	8	long[2]	Nonce Timestamp
*hash alg.	1	u_char	hash algorithm used
*encryption alg.	1	u_char	encryption algorithm used
Endpoint identification data:			
*destination NSID	1	u_char	destination name space identifier
*source NSID	1	u_char	source name space identifier
*destination ID	4	u_int	ID of receiver
*source ID	4	u_int	ID of sender
*destination GID	4	u_int	GID of receiver
*source GID	4	u_int	GID of sender
*destination socket	16	struct sockaddr	Socket address of receiver
*source socket	16	struct sockaddr	Socket address of sender
*called_atm_len	1	short	ATM address of receiver
*called_atm_addr	20	u_char*	
*called_sub_len	1	short	ATM subaddress of receiver
*called_sub_addr	20	u_char*	
*calling_atm_len	1	short	ATM address of sender
*calling_atm_addr	20	u_char*	
*calling_sub_len	1	short	ATM subaddress of sender
*calling_sub_addr	20	u_char*	
Algorithm specific data:			
signature	200	char[200]	Cryptographic signature

B Single Message Authentication Protocol

B.1 Authentication Protocol based on Signed Hashing

In this protocol the hash value of an authentication message is encrypted by the private key of the sender. After successful execution of the authentication protocol principal A (the claimant) has established her authenticity with principal B (the verifier) and ensured the integrity of data message m. The authentication message consists, for example, of a data message, a timestamp, a sequence number, and identifiers for the participants of this protocol A and B. The data message m can be empty, if only the authenticity of A is important. If m is not empty, this protocol establishes its integrity upon successful execution. The data message may consist of the first n octets of the first IP packet for this connection and a combination of information elements. The exact contents, coding, and layout for the authentication message are defined in Section A. This protocol is similar to current proposals in the IETF IP security working group.

B.2 Assumptions

This protocol assumes that the private key of the sender is not compromised, and a secure public key infrastructure exists, such as [7]. K_A, the public key of principal A is a public value. It may be cached for future speedup.

Protocol

1.	^{t_1}A			:	$h_A := h(m, t_1, n_A, A, B)$
2.	A			:	$s := \{h_A\}_{K_A^{-1}}$
3.	A	$\rightarrow$	$B^{\,t_2}$	:	$(m, t_1, n_A, A, B, s) \rightarrow (m^*, t_1^*, n_A^*, A^*, B^*, s^*)$
4.			B	:	lookup K_A
5.			B	:	$h_B := \{s^*\}_{K_A}$
6.			B	:	$h_A^* := h(m^*, t_1^*, n_A^*, A^*, B^*)$

time t_1 : A starts creating the authentication protocol message
time t_2 : B has received the authentication protocol message
time t_Δ : time window in which different sequence numbers are accepted.

B.3 Authentication Verification

After the last step of either protocol is completed, principal B performs a number of tests to determine if the authentication has succeeded. The authenticity of A and the integrity of data message m are not established if any single test fails.

	evaluates to *true*	result
1.	$(h_A^* \neq h_B)$	signature mismatch
2.	(identity of receiving node $\neq B^*$)	destination mismatch
3.	$(t_1 \notin (t_2 - t_\Delta, t_2])$	timing violation
4.	(n_A has been seen by B in t_Δ)	sequence number mismatch

C Notation

Principals participating in communication are denoted in capital letters A or B. A usually plays the role of the initiator (sender), B the acceptor (receiver) of a connection (of data). If the role is not clear from the context the principals are additionally labeled with their role.

Messages that are transmitted in packets are denoted by msg. Received messages are labeled with a superscript * to denote that the data might have been changed during transmission by an active wiretapper. Times are represented by t_i, where the subscript i is used to distinguish between different times. Numbers created by principal X are represented by n_X.

K is the symbol for encryption keys. If it is important whose principal's key it is, we will add the name of the principal as a subscript, e.g., K_A. K and K^{-1} are a public key pair with K^{-1} being the private key part. The same subscript rules apply. Encrypted messages are surrounded by curly braces, with the subscript stating the encryption key, e.g., $\{msg\}_{K_A^{-1}}$. Hash functions are abbreviated by h().