1990

Enhancements to the XNS authentication-by-proxy model

Peter D. Wing

Rochester Institute of Technology
Computer Science Department

Enhancements to the XNS Authentication-by-Proxy Model

by

Peter D. Wing +

A thesis, submitted to
The Faculty of the Computer Science Department
in partial fulfillment of the requirements for the degree of
Master of Science in Computer Science

Approved by:

Professor Peter H. Lutz

Mr. Paul A. Rulli

Professor Peter G. Anderson

Professor John A. Biles

April 2, 1990

Statement of Thesis Reproduction Rights

Title of Thesis: Enhancements to the XNS Authentication-by-Proxy Model

I, Peter D. Wing, must be contacted each time a request for reproduction of my above named graduate thesis is received. Reproduction rights are absolutely forbidden in lieu of the possession of a written document bearing my signature as shown below expressing permission the contrary. I may be reached to request such rights at the following address:

94 Hefner Drive
Webster, New York 14580

Signed: _____ Date: 4/2/90

Dedication

In loving tribute to Lawrence F. Fogarty, who by eclipsing all expectations I could have ever had of a father-in-law served as the foremost of many inspirations to complete the thesis element of these graduate studies while balancing family, friends, career, and the many other blessings we experience for such a short instant in history. Above all, he is an authentic friend.

Acknowledgements

This work is uniquely my product by only the most superficial of perspectives. While an exhaustive enumeration of those who contributed to its completion would itself be a formidable challenge, many warrant recognition on the basis of their critical roles in the endeavor. I am immeasureably grateful to these very special individuals.

Whomever coined the phrase "behind every successful man stands a good woman" was close to describing my wife, Kelly. Her love throughout our many years together is the primary enabler of this graduate education work, as well as most other activities in which I participate. Especially admired is her decision to indefinitely postpone a promising software engineering career in lieu of one clearly more important though counter society's latest norms investing in the future represented by our children. This being the case, I clearly recognize that "beside this very lucky man stands one great woman."

Our children, Meaghan and Kevin, are likewise sources of unbounded love and support. More than anything else, they are unfailing role models of the virtues of innocence. At the same time, they have provided a good share of the motivation to capitalize on the availability of this outstanding educational opportunity. It is my sincere hope that their sensitivity to the importance of perseverance in the pursuit of personal dreams has increased through their participation in this experience despite their very young age.

This was but one of the countless invaluable lessons I learned from my parents, Marie and Bob Wing. Among the others notable in this context is a strong conviction to a priority scheme headed by God and family. No doubt their own unfailing commitment to live by the value system they espouse has greatly influenced my aspirations to do likewise. Completion of this degree program is as much a credit to their success in instilling a dedication to apply my talents consistent with their values as any other single factor.

At the age of 15 I first met the wonderful couple who years later became my in-laws, Jeannette and Larry Fogarty. They have always treated me since in a manner of which every son should be so blessed. Few individuals in today's society have the advantage of a single set of terrific parents. I have had two.

Remaining family members and close friends also contributed significantly towards completion of this thesis in various capacities. Common to all, however, was supportiveness in dealing with the time demands my studies entailed. This ranged broadly from good-natured acceptance of my inability to fully participate in various activities to gently reining me in when I was attempting to do too much. Their flexibility and understanding in light of such impositions has been admirable.

Many individuals have enriched my career experience and individual capabilities, but none more than Bob Hurtz. The enjoyment I have derived in the course of our collaboration on such tasks as security product software development and software

engineering methodology establishment is eclipsed in the work environment only by our friendship. Few of us are endowed with the abilities and drive to apply them which Bob possesses and readily shares.

Larry Rourke is distinguished as uniquely providing the environment in which my focus turned to protection as a concentration area. His uncompromising foresight and admirable ability to move corporate inertia towards such high opportunity applications are largely responsible for the fact that this thesis pertains to authentication and not any of a thousand other pertinent topics. One can only hope that the valuable products of our efforts with Bob Hurtz come to fruition.

A number of managers have also been quite instrumental in motivating my pursuit of a graduate degree, Bob Wolf being the first. Though we worked together before this coursework actually began, it was Bob who most reinforced and facilitated pursuit of prior aspirations of advanced formal credentials within the computer science profession. Denny Ulrich was a savior providing friendship, autonomy, and much-needed ideological reinforcement under adverse circumstances. Jim Iverson epitomized the advantages of skills and strategy over position and resources. Jim also notably reaffirmed the importance of a balanced lifestyle through both word and act. All of these men set very high expectations for my performance Maslow's formula for self-actualization. Finally, Bill Valentine provided the environment under which the applied research skills fostered through these graduate studies was allowed to mature significantly through application to his critical software engineering product development responsibilities.

Xerox Corporation supported my graduate studies via such means as tuition aid, computing facilities, on-site course administration, workforce development policies, and a challenging engineering environment comprised of highly skilled associates. It also funded a significant percentage of the underlying research and development comprising the foundation of protection literature.

Finally, thanks are due those members of the RIT graduate faculty from whom I studied. Particular recognition is due my highly-skilled thesis committee: Peter Lutz and Paul Rulli. Their diligent verification of my interim thesis outputs both greatly accelerated their completion schedule and quality.

The intended consistent theme throughout these acknowledgements is my great fortune in having been associated with each of these men, women, and children over the course of the six years spanning this graduate coursework. Had one known all of the hurdles to be encountered in the course of this task at the time of its initiation, a shortfall would be expected. These folks represent the intangible element by which I have been and will continue to be successful.

Abstract

Authentication is the secure network architecture mechanism by which a pair of suspicious principals communicating over presumably unsecure channels assure themselves that each is that whom it claims to be. The Xerox Network Systems architecture proposes one such authentication scheme. This thesis examines the system consequences of the XNS model's unique proxy variant, by which a principal may temporarily commission a second network entity to assume its identity as a means of authority transfer. Specific attendant system failure modes are highlighted. The student's associated original contributions include proposed model revisions which rectify authentication shortfalls yet facilitate the temporal authority transfer motivating the proxy model. Consistent with the acknowledgement that no single solution is defensible as best under circumstances of such technical and administrative complexity, three viable such architectures are specified. Finally, the demand for a disciplined agent management mechanism within a distributed system such as XNS is resoundingly affirmed in the course of these first-order pursuits.

Computing Review Categories and Subject Descriptors

Primary:

- D.4.6 [Operating Systems]: Security and Protection - *access controls, authentication, cryptographic controls*

Secondary:

- C.2.0 [Computer-Communication Networks]: General - *security and protection*
- C.2.2 [Computer-Communication Networks]: Network Protocols - *protocol architecture, XNS*
- C.2.4 [Computer-Communication Networks]: Distributed Systems - *distributed applications*

General Terms

Design, Security

Additional Key Words and Phrases

Agent, Accountability, Architecture, Authorization, Distributed Document Management, Distributed Printing, Impersonation, Interpress, Nonce, Page Description Language (PDL), Private-key Encryption, Proxy, Responsibility, Timestamp

Table of Contents

1 Introduction and Background

This document satisfies the prescribed role of the thesis report element of the Master of Science degree requirements under Rochester Institute of Technology's Graduate Computer Science program. Consistent with this mission, the discussion to follow details the innovative products created by the student's particular research and development activity. The emphasis thereof is largely one of distributed systems requirements analysis and architecture specification.

A significant element of this report thus lies in Section 2, which surveys pertinent previous work. This is comprised of two primary components: a generic treatment of relevant principles from established literature and detailed examination of the particular baseline model to be enhanced. Notably original contributions are provided in this document with respect to the identification of significant improvement opportunities. These are presented in Section 3, in association with numerous candidate models by which they may be realized. Section 4 summarizes the key contributions of this thesis, as well as potential outstanding tasks. The remainder of this section introduces the student's topic, describes the supporting information acquisition methods employed, and presents notational conventions to be applied hereafter.

This organization is consistent with the guideline presented in the defined thesis development process given exclusion of its software project emphasis.[16]

1.1 Problem Statement

Authentication is the secure network architecture mechanism by which a pair of suspicious principals communicating over presumably unsecure channels assure themselves that each is that whom it claims to be. The Xerox Network Systems [XNS] architecture proposes one such authentication scheme. This thesis examines the system consequences of the XNS model's unique proxy variant, by which a principal may temporarily commission a second network entity to assume its identity as a means of authority transfer. Specific attendant system failure modes are highlighted. The student's associated original contributions include proposed model revisions which rectify authentication shortfalls yet facilitate the temporal authority transfer motivating the proxy model. Consistent with the acknowledgement that no single solution is defensible as best under circumstances of such technical and administrative complexity, three viable such architectures are specified. The utility of an agent management mechanism within a distributed system is challenged in the course of this discussion, as is its architectural coupling to the authentication mechanism.

1.2 Authentication and Authority Transfer: to Couple or not to Couple?

As vividly demonstrated by the recent "Cornell worm", a key shortfall of most modern distributed systems is the relative ineffectiveness of their protection mechanisms.[17] This inadequacy has percolated to the top of those constraining the rate by which such state-of-the-art architectures have become state-of-the-practice. Even in many of those applications where distributed systems have made successful inroads, considerable operating risk must be tolerated as a tradeoff against the immediate benefits of a distributed organization. This thesis considers a particular significant element of but one such network protection scheme: the Xerox Network System's Authentication-by-Proxy mechanism.

To review, protection is a simple case of enforcing a set of rules which constrain the operations which active system entities [i.e. subjects] may apply to passive system entities [i.e. objects]. Implicit in this statement is the fundamental assumption that the identity of each of the principals to the transaction has been accurately established. The utility in this knowledge is of the first order in the case of discretionary policies, which control subject to object access purely as a consequence of identity. Alternatively, a second-order effect is witnessed in light of mandatory rules, by which identity is mapped to clearance the ultimate basis of access control.

Consequently, impersonation is a particularly attractive means of subverting protection safeguards. The introduction of a distributed system architecture appreciably shifts the risk/reward profile in favor of the would-be intruder. This is an unfortunate side-effect of the loosely-coupled organization generally applied to the underlying networking media. That is, system administrators intentionally have minimal control over the equipment configuration comprising the domain for which they have protection responsibility. Subject to the architecture, distributed system users are often at liberty to vary any aspect of the node which acts as their network surrogate including the hardware, the operating system kernel, and the applications software.

In light of this degree of network configuration freedom, the only reasonably effective model by which each principal to a transaction may establish the identity of its prospective partners is through mutual suspicion. This is the role of authentication mechanisms. A great many such techniques have been developed given the significance of the impersonation threat and the subtle complexities of the problem.[2]

Complementing authentication as a fundamental theme of this thesis is that of temporary authority transfer. As in social interactions, it is

sometimes architecturally convenient to dispatch a third-party to conduct affairs on behalf of the original authorized party, the transaction initiator. Of particular note is the fact that the party with whom the exchange is targeted, the recipient, generally requires proof of the initiator's delegation of authority to his agent. Such a proxy represents the agent's right to fully-engage in transactions with complete responsibility for the resultant effects transferring to the initiator.

The XNS model to be considered in this thesis achieves temporary authority transfer through controlled impersonation in the authentication mechanism. It is this apparent significant inconsistency of employing impersonation in conjunction with the authentication mechanism to achieve authority transfer which interests the student. This thesis is therefore devoted to dissecting the effectiveness of this coupling in light of complementary system protection considerations.

1.3 Information Acquisition Methods

Exceptionally extensive dedicated research provides a sound foundation upon which this thesis is constructed. This is a fortunate consequence of having charted a direct path towards a thesis concentrated in the area of distributed systems protection early in the curriculum cycle. Such was the motivation behind a trilogy of research reports delivered in partial satisfaction of the requirements of three separate courses.[2,3,4] In succession, this research expanded from the protection primitives associated with security kernels to encompass secure open systems developments, concluding with a survey of notable authentication models. The last of these was specifically intended to establish the baseline existence of significant prior research activity with respect to authentication. Section 2 of this report is largely an integration of those portions of the three original research reports directly applicable to this thesis.

A considerable lag between completion of the authentication survey and thesis initiation prompted the incremental need for a library search to identify subsequent relevant research in June of 1989. No major information was acquired through this effort, though a series of relevant articles describing the "Cornell worm" appeared in *Communications of the ACM* in that timeframe. Ongoing monitoring of trade journals and dialog conducted within electronic mail-based special interest groups was another prolific information source extending to the time of this writing. Of equal currency but parallel subject emphasis is the student's seemingly insatiable appetite for software engineering literature. This

avenue unexpectedly yielded the fundamental engineering principles by which the final architectural models were produced.

1.4 Notation

The data flow diagram (DFD) is the primary graphical representation tool employed to complement narrative text throughout this report.[18] Each authentication model considered is explained using both of these communication tools. Message legibility accrues from such reliance upon consistent notation.

The active entities in the models [i.e. processes], principals and authentication services, are represented as circles. These are connected by named vectors [i.e. flows], the labels of which indicate passive exchanged message contents. Notational liberty has been taken with respect to DFD conventions in applying numbers to the vectors depicting message ordering as an informational expediant. In addition, a pair of parallel lines represent files, semantically indicating time-delayed data exchange as opposed to the instantaneous nature of data flow across named vectors. The DFD source and sink constructs are unused in this work.

The second notable convention pertains to the encryption of messages and fields thereof passing between authentication model communicants in the DFDs. In this instance, a pair of curly braces '{}' delimits the encrypted message. The cleartext is identified between these braces, with the encryption key denoted as a superscript following the trailing brace. For instance, $\{mydata\}^{KA}$ represents the ciphertext formed by encrypting "mydata" under key KA. Those message fields not delimited by curly braces are transmitted in the clear.

2 Previous Work

As its name conveys, the intent of this section is solely to establish the target set of existing affairs to be extended by the original work comprising this thesis. Of course, this end can only be achieved through a discussion of the specific topic of the student's research: the current Xerox Network Systems Authentication-by-Proxy model. Equally as relevant to a distributed systems research effort such as this is an appreciation of the larger architecture in which this particular authentication service plays an integral role, XNS. It is imperative that the reviewer understand both the broad organization of the encompassing architecture and the specific role of the particular service under investigation in order to accurately verify the need for and sufficiency of proposed architectural modifications produced by the student.

The committee's verification of these attributes of the student's work is the means of thesis evaluation. The other qualification necessary to perform this function is an appreciation of established protection principles. This section therefore sets these forth as well. In fact, this discussion largely mirrors the evolution in security developments both in a historical sense and by the student. That is, the fundamental set of protection basics established in the course of computer security mechanisms for standalone systems is extended to networked configurations, finally concentrating on authentication. These topics are treated in considerable detail in the student's prior research reports.[2,3,4] Only the most relevant of these established practices are described below.

2.1 Protection Primitives

Security systems in the non-computing sense are quite mature as a consequence of considerable historical research. For the same reason, protection mechanisms in standalone computing systems are quite well-understood at this point in time. While this knowledge is far from complete, those pursuing the problem now possess considerable capacity to deliver effective computer protection facilities. Of course, total system security in any computing environment is a multi-faceted problem represented by the expression:[13]

S = f(P1 * P2 * A * C1 * C2), where

- S: total system security
- P1: physical security
- P2: personnel security
- A: administrative security
- C1: communications security
- C2: computer security

According to its proponents, each of these factors may be thought of as varying by weight from zero to one according to the application environment. If any single component is deficient, total system security is at risk of some magnitude in spite of effective complementary measures. Likewise, redundant investments across these factors provide no additional overall assurance while inflating costs. The trick as a security systems architect is to strike an effective balance among these various control points given the potential leverage of each.

While all of these factors are interesting, the last two are those most pertinent to this research. The primary reason for introducing this broader view systems view at this point is to emphasize the urgent need to comprehend and actively manage the system impact consequent to decisions at the individual protection mechanism level. This point underscored, the discussion now resumes its computer security theme. In particular, the remainder of this section highlights the basic elements of computing protection.

2.1.1 Reference Monitor

Among the fundamental properties of computing systems is the sharing of a common set of resources among many clients as illustrated in Figure 2.1.1.1. Depending upon the specifics of the situation, sharing provides significant opportunity with respect to such system quality factors as efficiency, reliability, and usability.[22]

However, architectural accomodations must be made in order to reap the potential benefits of resource sharing. The most studied of these deal with the synchronization of client references to the shared resources. The semaphore provides a primitive, though complete and widely-utilized, means of synchronization. A glaring shortfall of the semaphore is the fact that its effectiveness is heavily reliant upon the correctness of its explicit use by each of the resource clients. By centralizing the synchronization logic, the monitor provides a more attractive means of resource reference timing arbitration from such quality perspectives as reliability, verifiability, and maintainability.[22,23]

Protection is a second architectural accomodation which is consequent to the goal of orderly resource sharing. Rather than coordinating shared-resource references along the dimension of time, protection mechanisms do so according to the predefined rights of the client with respect to the resource. In this way, resource access is constrained to be consistent with the intent of its owner and the best interest of its client community.

Taking a page from the evolution of synchronization primitives described above, the reference monitor model of Figure 2.1.1.2 is the conceptual

protection enforcement primitive.[12] This model classifies those shared resources to be protected as objects, with their clients referred to as subjects. Note that not all shared resources need be protected. Typical subjects include human users and the processes which serve them. Files, documents, records, devices, memory segments, and instructions are generally categorized by the system architect as objects to be protected. This organization establishes the granularity of protection provided by the system.

Each subsequent attempt by a subject to access an object must be mediated by the reference monitor based upon the predefined relations to which it alone has access. The reference monitor itself also need be isolated from influence by the subjects as an additional means of assuring the reliability of its operation. This assurance is also bolstered by explicitly engineering those quality criteria supporting verifiability into the reference monitor.[6,22]

These requirements are manifested in the security kernel element of the system depicted in Figure 2.1.1.3. This diagram represents system architecture as a hierarchy of virtual machines, consistent with the contemporary trend.[24] Note that the security kernel provides the base-level virtual machine by managing all protected shared-resources, thereby satisfying the required reference monitor property of mediation. It likewise maintains a separate execution domain for itself per the requirement of isolation. Finally, such centralization and size minimization facilitate verification of its correct operation. To complete the architecture, the supervisor exports remaining operating system services to the applications layer, which in turn serves the end-users.[12]

2.1.2 Threats

A prerequisite to the design of effective protection schemes is an understanding of the threats to be countered. This process of matching requirements to design is no different than that of any other aspect of system engineering. A quality solution is particularly critical in the instance of protection features, however, as end-user utility is directly proportional to the number of design oversights. Even a single flaw in the solution, if sufficiently significant, could render the remaining protection considerations useless. At the very least, user confidence is severely eroded by identification of protection shortfalls.

One might consider the above claim of the need for absolute protection integrity to be an unreasonable quality demand. Therein lies the relevance of the flaw's significance. By design, protection schemes are rarely absolute. To do so would be counterproductive to the associated goal of controlled, efficient, resource sharing. An inverse relation of sorts

exists between the ease of resource sharing and protection scheme severity. The middle ground most often chosen is to adopt sufficient protection as to reduce the reward associated with successful intrusion to a level as to be unwarranted by the cost to do so.[21] Once again, attainment of this goal implies a comprehensive understanding of potential threats.

Such threats may be evaluated according to various orientations. That which is most widely acclaimed in the media is the motivation behind the compromise. In fact, many equate protection solely with the goal of preventing deliberate intrusion. This perspective represents a dangerous simplification - one which has led many an organization to ignore those prudent precautions which could prevent the inevitable accidental threats on the flip side of the motivational coin. It is sometimes easy to effectively argue that the cost of protection is superflous on the basis of the inherent goodness of those comprising the work group. In a distributed system environment, such an argument is much less convincing given the potential disparity of users and participating nodes. In any case, the added complexity introduced by distribution begs for effective protection if only on the basis of the risk associated with inadvertent sources of failure.

A second orientation by which threats may be interpreted is as passive or active. Whereas the value in considering threat motivation lies in assessing the probability of compromise occurrence, this view considers the post-intrusion state of the system. This information, in turn, constrains the attributes of the set of mechanisms by which the threat may effectively be countered. A passive threat is defined to be one in which unauthorized disclosure of information occurs without impacting system state in a noticeable way. On the other hand, an active threat is one in which damage is inflicted through a change in system state. This distinction is significant in that passive threats must be prevented as, by definition, they cannot be detected after the fact. In contrast, active threats often cannot be prevented but can absolutely be detected. As a system response, this implies the minimal need to detect such attacks, ideally complemented with correction mechanisms.

The net impact of a compromise upon the target object is the most tangible of all possible interpretations. Given such relevance, these have been set forth below in distinct subsections.

2.1.2.1 Unauthorized Disclosure

In this type of threat, information is made known to one for whom it is not intended. This is referred to as interception when unauthorized disclosure occurs as a consequence of a deliberate act, and as exposure

otherwise.[14] Unauthorized disclosure is generally a consequence of passive threats. Prevention is thus the protection strategy often associated with specific forms of it. Examples of this class of threat to which the reader may relate include reading from memory regions to which one is not privy and wire-tapping.

2.1.2.2 Unauthorized Modification

In this type of threat, information is modified by one not authorized to do so. This is referred to as deception when unauthorized modification occurs as a consequence of a deliberate act, and as alteration otherwise.[14] Unauthorized modification is generally a consequence of active threats. Detection is thus the protection strategy often associated with specific forms of it. Examples of this class of threat to which the reader may relate include writing to memory regions to which one is not intended access and modification of messages on data communications lines.

2.1.2.3 Denial of Service

In this type of threat, access to an object for which one is sufficiently authorized is wrongly prevented by another. This is referred to as disruption when denial of service occurs as a consequence of a deliberate act, and as interruption otherwise.[14] Denial of service is equally likely to be a consequence of threats which are passive or active in nature. This is often a particularly difficult type of threat to handle. When prevention is not an alternative protection strategy, detection is employed. Even detection is sometimes ineffective. Examples of this class of threat to which the reader may relate include allocation of all available memory and infinite execution loops.

2.1.3 Access Control Paradigms

Having partitioned the system elements into subject and object sets, the system architect must define the protection policy to be enforced by the reference monitor. Thanks in large part the existence of the broadly-accepted Bell and LaPadula formal security model, this is one wheel which need not be constantly reinvented but simply balanced against application-specific needs. This model is the product of Department of Defense-sponsored research conducted in the mid-1970's at Mitre Corporation and Case Western Reserve University.

Two types of access control policy are identified by the Bell and LaPadula model: mandatory and discretionary. As the name implies, mandatory policies are uniformly enforced by the reference monitor according to the so-called dominance relation which exists given the predefined

classifications associated with the particular subject and object involved in any transaction. Discretionary, or need-to-know policies, enable subjects to create new objects and to determine the scope to which other subjects will be allowed access. A particular reference monitor may enforce a combination of these policies depending upon the level of assurance required to support the system's intended application.

As introduced above, monitor enforcement of a mandatory policy is based upon the clearance of the subject, the sensitivity of the object, and the type of access. Considerable mandatory policy tailoring to application needs may be performed given the alternatives of hierarchical and nonhierarchical classification schemes.[4] A pair of key principles determine the acceptability of a type of access by subject to object independent of categorization scheme: the simple security condition and the *-property (pronounced "star-property").

The simple security condition states that only those subjects possessing clearance greater than or equal to the sensitivity of the object may acquire (i.e. read) its contents. A somewhat more subtle aspect of the unauthorized disclosure threat is addressed by enforcement of the *-property. Under this rule, object modification (i.e. write) is granted to only those subjects whose clearance is less than or equal to that of the object's sensitivity.

Application of the *-property prevents many of those potential "Trojan Horse" attacks in which unauthorized object declassification is perpetrated as a side effect of valid data acquisition by a subject. A simple such threat scenario could occur in conjunction with the introduction of a utility to be shared throughout a user population of varying clearance. While apparently performing to the complete satisfaction of its clients, such a utility could also seek to record a copy of all information to which it is exposed in a minimum sensitivity repository which may subsequently be legitimately accessed indiscriminately under the conditions of the simple security condition. This intrusion is prevented through enforcement of the *-property, for while the utility remains rightly allowed to acquire objects consistent with the clearance of its present user, under no circumstance may it store information in a less-sensitive object than that of the user's clearance.[51] The integrity of the simple security condition is thereby reinforced by the *-property.

A second type of access control policy is referred to as discretionary. This term is representative of the fact that such policies regulate object access based not upon their static sensitivity level, but at the discretion of an owning subject. The extent of object availability is determined only by the owner's assessment as to the set of subjects which possess a need-to-

know with respect to that object. The granularity of such discrimination varies from coarse in the case of a group structure scheme to fine in the more robust access or capability list representations touched upon below.[4] The success of object protection under a discretionary policy is thus directly proportional to the accuracy of the owner's decision, the diligence with which this responsibility is exercised, and the granularity of the mechanism. A discretionary policy is thus particularly well-suited to those situations in which a number of subjects of indistinguishable clearance cooperate to complete an operation.

2.1.4 Access Right Representation Schemes

Important consequences derive from the access right representation mechanism selected under a discretionary access control policy. Basically, the choice is between association of the rights with the object or the subject. The more common former instance is referred to as an access list, while the latter is termed a capability list.[25]

Conceptually, each entry of an access list describes the rights of a subject with respect to the associated object. Reference mediation given an access list representation consists of searching the object's access list for the rights of the referencing subject. Unfortunately, this can be somewhat time-consuming and provides no simple means of identifying the complete rights of a particular subject across all objects.[23]

Capability lists have just the opposite profile it is easy to determine the scope of a subject's rights across all objects, but very costly to trace the rights of all subjects against a given object. Consequently, rights revocation is quite costly. In this case, each entry in the list names an object and describes the subject's particular access rights in its regard. Each such construct is referred to as a capability. When the subject wishes to exercise its rights under such a scheme, it presents its capability for the target object to the reference monitor. The reference monitor equates capability possession with the subject's right to complete the reference. This being the case, capability creation must be a protected operation. This is often implemented by selecting capabilities based on large, sparse, random object name spaces which probabilistically precludes their forgery.[23,26,27]

2.2 Protection within Networked Systems

Until quite recently, the emphasis of protection mechanism research and development focused on non-networked system architectures. Most of the spadework pertaining to security between communicating systems had been focused on the fundamental mechanisms of encryption and identification due to their utility in timesharing applications. This was sufficient given the fact that networking technologists were simultaneously working to demonstrate the achievability of their open system ideal. In light of marked commercial system disparity, their task has been to develop models enabling arbitrary communications amongst a heterogeneous collection of hosts, each of which is an element of some independently managed, interconnected network.[8] Restriction of exchanges which necessarily should be prevented was not an issue as long as the technology was of the nature that such internetworking was not possible.

Each discipline has now matured to the point that serious consideration of the requirement to engineer secure open systems is underway. The following factors are among the primary motivations for such endeavors:[8]

- The growing quantity and value of information made vulnerable by the breaching of network security make networks tempting targets.
- Computer systems connected by networks are likely to cooperate in various ways to provide resource sharing for a user community. As a result of this sharing, the security of information on a given host may become dependent on the security measures employed by the network and by other hosts.
- The development of new network technologies facilitates certain kinds of attacks on communications systems; for example, it is easy for an intruder to monitor the transmission of satellite and radio networks.
- The increased use of networks to provide remote access to computer facilities, coupled with improved physical security measures at computer sites, makes attacking networks more attractive to an intruder.

The remainder of this section summarizes those extensions to the primitives already discussed which contribute to such desired protection within network systems. This discussion is constrained to encompass only those architectural dimensions which bear directly upon the student's specific thesis development.

2.2.1 Open Systems Architecture

No survey of contemporary network architecture would be sufficient if it omitted the International Organization for Standardization's Open Systems Interconnection Reference Model depicted in Figure 2.2.1.1.[20] The OSI protocol reference model layers network architecture in such a way as to be maximally consistent with highly-accepted existing systems architecture yet extensible to accomodate projected application and technological developments. Pursuit of these goals supports the model's mission: to provide a platform upon which commercial systems vendors will deliver interoperable products. Both computing vendors and customers are the end beneficiaries of such developments along such system quality dimensions as functionality, performance, cost, and longevity.

Among the key messages the reader should glean from Figure 2.2.1.1 is the logical nature of exchanges between communicating nodes at any single layer in the network hierarchy. Processes executing within the respective nodes, physically separated by the network communications mechanism, are referred to as peers. Another pertinent note is that peer communications are constrained to processes in equivalent layers of the model. This is contrasted by physical exchanges across the interface between adjacent architectural layers within a single node. Peer exchanges represent semantics, while interface exchanges support communications.

The diagram also illustrates a key distinction related to the nature of exchanges between processes in layers one through three, collectively known as the communications subnet, and those of the higher-order four layers. Subnet interactions occur among a variable number of adjacent IMPs, or message switches, along the route from source to destination node. These exchanges add nothing to the semantic handling of the exchange, but merely facilitate moving the message along the path to its final destination. In the Transport through Applications layers, traffic is end-to-end in the sense that only the intended processes within the source and destination nodes participate.

The layered systems decomposition employed in the OSI model is based upon the Principle of Steps, a hierarchical ordering principle exhibited by complex systems throughout the universe including operating systems and networks.[24] Three observations comprise this notion:

- The universe is mostly empty space.
- At each step there are well-defined objects with well-defined rules of interaction.

- The objects of a given step are composed of objects of lower steps and are constituents of objects of higher steps.

Such partitioning facilitates an orderly migration of the system over time based upon the localized evolution of its component layers. The following particular design rules guided the ISO in its development of the seven layers comprising its model. [20]

- A layer should be created where a different level of abstraction is needed.
- Each layer should perform a well defined function.
- The function of each layer should be chosen with an eye toward defining internationally standardized protocols.
- The layer boundaries should be chosen to minimize the information flow across the interfaces.
- The number of layers should be large enough that distinct functions need not be thrown together in the same layer out of necessity, and be small enough that the architecture does not become unwieldy.

The consequent functional decomposition of network architecture under the OSI model is presented in Figure 2.2.1.2.[15] Due to the lack of urgency at the time of its development noted previously, no provisions for protection are specified in the base model.

2.2.2 Threats

Five additional threats accrue from the nature of communications between networked systems. Whereas all represent flavors of the unauthorized disclosure, unauthorized modification, and denial of service threats presented earlier, each is endowed special character given such an architectural expansion. As before, understanding such threats is a prerequisite to successfully countering their occurrence as required.

2.2.2.1 Release of Message Contents

Distributed systems are particularly susceptible to this passive threat of unauthorized disclosure. This is largely a consequence of the fact that the set of messages comprising each transaction between cooperating principals must pass through the intervening network machinery on each end-node in addition to the communications subnet. Depending upon such factors as the effectiveness of the computer security at each switch, the physical security of the transmission media, and the nature of the subnet, messages face the very distinct possibility of interception by an unauthorized party.

2.2.2.2 Traffic Analysis

A very subtle source of unauthorized disclosure is that which may occur as a consequence of monitoring the demographics of traffic patterns on the underlying communications subnet. That is, an intruder may glean a significant volume of information as a side-effect of the frequency, length, and source-to-destination traffic attributes.

Traffic analysis is a serious threat as a potential covert timing channel.[13] Communicants A and B may deliberately choose to interact in defined patterns as a means of clueing party C into sensitive information. As in the standalone machine case, such disclosures are best fought by channel bandwidth reduction below the threshold of desireability.[21,28]

2.2.2.3 Message Stream Modification

Threats of this sort refer to attacks on the integrity, authenticity, and/or the ordering of messages passing between principals.[13] Integrity means that a message has not been modified en route. Authenticity, our eventual subject focal point, means that the issuer of the the message is that whom the receiver expects it to be. And ordering means that the message can be properly located in the message sequence passing between the communicants. Attacks to authenticity include insertion of bogus or replayed messages while those to ordering include message deletion or duplication and altering the order of messages within the stream.

These three properties share a unique relationship in that measures to address integrity are fundamental to insuring authenticity, which is likewise basic to message order preservation.[13]

2.2.2.4 Denial of Message Service

This type of attack may be perceived as a persistent message stream modification attack. Such attacks include discarding or delaying all messages passing in either or both directions of an association between two communicants.[8] As is true of its standalone machine equivalent, this threat variant is sometimes quite difficult to detect. This is particularly true during those intervals when an exchange is quiescent. At that point in time, each party to the conversation generally has no way of predicting when the next message will arrive from its associate. An attack which completely reduces message arrivals from the other party cannot be detected: "is my partner *really* idle??"

2.2.2.5 Spurious Association Initiation

While sharing the characteristic of being the consequence of an active attack with the previous two forms discussed, spurious association initiation differs from its counterparts in its timing relative to the communications lifecycle itself. That is, this type of attack occurs while the two parties are in the process of establishing a connection, whereas the others take place after successful completion of this phase of the interchange. The two forms of spurious association initiation most frequently encountered are:[8]

- impersonation: an attempt to establish an association under a false identity.
- replay: repeating a recording of a previous legitimate association initiation sequence or portion thereof.

The interactive form of authentication of which the subject XNS model is an instance exists specifically to prevent successful spurious association initiations. This point is likely self-evident to the reader. In light thereof, what is the significance in understanding the four remaining classes of threat described above? Quite simply, while authentication is a means to combat spurious message initiation, the messages by which it provides that protection are themselves subject to the same threats as are any interchanges among distributed system components. Other than that of traffic analysis, an effective means of countering each of the threats described above is a prerequisite to dependable authentication models. This represents a second observation of the fragile interdependence of protection components upon net effectiveness.

2.2.3 Protection Mechanisms

In response to increasing recognition of the risk which these threats pose to the reliable operation of networked systems, many mechanisms by which they may be countered have been identified.[5] All are quite intricate. Each is effective when applied in the proper situation, otherwise not. Some facilitate passive threat prevention, while others enable active threat detection and/or correction. Many are certainly directly relevant to the critique of the existing XNS Authentication-by-Proxy mechanism to be presented later in this report. These are discussed in the remainder of this section, as are a number of others of high utility to the student in the pending architectural revision component of this thesis.

2.2.3.1 Encryption

Fittingly, our discussion of protection mechanisms begins with that which is clearly both the most widely-recognized and versatile. As a consequence of judicious message encryption, passive intrusions may largely be prevented and active attacks detected. Such potency has led to widespread development and adoption of encryption algorithms. Consequent to such heavy application, this mechanism is also the frequent target of vulnerability identification research, colloquially known as hacking or cracking.

For the purpose of this discussion, encryption is interesting only as to the manner in which it is applied as a means of message isolation between distributed communicants. Though technically-challenging, algorithm mechanics are beyond the scope of this need. It is within those constraints that our discussion proceeds.

2.2.3.1.1 Invertible vs. One-Way Translations

Encryption refers to the process of translating cleartext into ciphertext for the interval of time during which it is subject to attack and back to cleartext when it is again in a secure domain. The conversion from cleartext to ciphertext is referred to as encryption, while the reverse process is termed decryption.

This is inherently an invertible process. That is, an explicit encryption goal is to reliably recover the original text at the destination with no information loss. The translation to ciphertext form is a temporal response to the requirement to pass securely through notably untrusted territory.

The definition of passing securely may represent any of a number of goals. Among them is that of maintaining exclusive information privacy between a set of interacting principals. This goal is diametrically opposed to the threat of the release of message contents. Other potential goals are to enable the message recipient to validate that it originated at the expected intiator, as well as that the message was not altered en route. Each of these objectives potentially motivate the need to encypt [read: isolate] for a finite time-interval.

Invertible functions are dependent upon the intiator and recipient uniquely possessing private information which enables the translations. This information is generally referred to as an encryption key. Section 2.2.3.1.2 describes alternative key models. In any event, some degree of key secrecy is mandatory to insure the privacy and/or integrity of the ciphertext.

A closely-related text translation concept as related to protection is that of the one-way function. This scheme has traditionally been applied to the authentication component of non-networked systems.[27] As the name conveys, its major distinction from invertible functions is that the original value may not be recovered after application of the one-way function. The goal in this instance is not to temporarily isolate information from potential intruders, but rather to provide a scheme under which possession of the original cleartext is itself sufficient proof of identity.

Figure 2.2.3.1.1 demonstrates such a technique, in which the cleartext X is presented to the one-way function arbitrator to yield a translated value. This is then compared with the entry of a previously-established public lookup table which maps A's user identity to the expected such value. The success of such a scheme is dependent upon the absolute inability to invert this function. An additional notable assumption is that the value produced by the function differs for each original value, as an increased collision rate would proportionally reduce the level of identity assurance. Function space sparseness similarly enhances protection, thus the oft-cited interest in reasonably-lengthly cleartext. The sparseness rate required to provide a fixed level of assurance is a function of the translation bandwidth, which has been steadily increasing given technological advances. The final necessary attribute of such a scheme is randomness of the cleartext space. Each of these attributes varies the risk/reward profile associated with brute force attacks.

Invertibility and collision-rate are implicit properties of the one-way function, while sparseness and randomness are cleartext dependent. This is one of the critical observations which enabled Robert Morris, Jr. to successfully compromise the UNIX-based internetwork on November 2, 1988.[17,29] In practice, the password space of typical UNIX installations is quite clustered and predictable. This is largely a consequence of weak, informal password management practices. Unfortunately, the scope of the worm's compromise testifies to the fact that this is the prevailing industry practice rather than an isolated instance. Serious enforcement of publically-available guidelines is the highly-endorsed prescription.[30]

2.2.3.1.2 Private vs. Public Keys

As described above, encryption algorithms are dependent upon either of two key use models. In the private key case depicted in Figure 2.2.3.1.2.1, K_{AB} represents a shared key which is secretly held by the transaction's two principals, message initiator A and recipient B. The encryption algorithm in the private key case is symmetric with respect to its use of the common key, K_{AB}. That is, the same key is employed both to translate

from cleartext to ciphertext [represented in the diagram as $\{CLEARTEXT\}K_{AB}$] at the initiator as in the opposite direction at the recipient.

In this instance, authentication becomes a matter of insuring that only the two principals to a transaction are informed as to the private key devoted to encrypt its component messages. A considerable portion of the authentication solution thus becomes that of communicating the private key to the principals.[9] This need is frequently aided architecturally by introduction of a trusted authentication server [aka key distribution center], models of which are described in section 2.3.

The public key label of the alternative scheme is actually somewhat of a misnomer. In fact, the scheme represented in Figure 2.2.3.1.2.2 depends upon a pair of keys, one of which is publically-advertised, the other of which is secret. The encryption algorithm in this instance is asymmetric with respect to the key pair in that one key is required to encrypt a message while another is required for its inversion. The public key, PKA, is globally distributed to enable A's communicants to facilitate message forwarding such that they are intelligible only to A by virtue of its possession of the associated secret key, SKA. Similarly, those to whom A issues messages may be assured of its source by the fact that they decrypt properly under PKB. In this instance, message privacy is not the aim, but rather source integrity is guaranteed. Accordingly, a required property of the public key model is that neither key may be derived from the other. This is particularly significant as regards the ability to ascertain the secret key from its public counterpart.

Figure 2.2.3.1.2.2 actually represents a combination of the two public key techniques described above. As recipient B is interested both in source integrity and message privacy, initiator A first applies its secret key to the clear text followed by B's public key. Upon destination arrival, B's secret key is applied to strip off the privacy layer, followed by the source integrity protection provided through decryption under PKA.

As in the private key model, a trusted authentication server is often enlisted to support distributed systems authentication. The model of use varies somewhat, however.

2.2.3.1.3 Link vs. End-to-End Measures

Yet another variant on the encryption puzzle pertains to its scope within the distributed system. Under the link encryption scheme of Figure 2.2.3.1.3.1, a separate key is employed to pass the original message, M, along each hop of its path from initiator A to recipient B. Such a scheme

is advantageous with respect to its ability to limit traffic analysis and the scope of exposure upon key compromise.

An alternative method is to employ an end-to-end scheme, such as that of Figure 2.2.3.1.3.2, by which the message is encrypted once at the initiator with the reciprocal decryption occurring only upon its arrival at the other principal. Such a solution is generally favored by network security architects on the basis that it enables the selective participation of distributed system elements in the encryption scheme according to their individual protection needs.[8]

This perspective is consistent with that which has been adopted by many distributed system authentication model architects. Most such solutions have been constrained by the assumption that the principals have freely elected to employ authenticated communications. The broader problems introduced by such requirements as forced secure communications or coupling prevention are beyond the scope of this thesis.[9]

2.2.3.2 Digital Signature

Analogous to its interpersonal namesake, this mechanism provides a means of proving that a particular data object originated from the party who's unique marking is affixed thereon. The ability to apply the mark must therefore be solely possessed by the object's originator. Signatures would otherwise be unenforceable as their creators could repudiate having issued them in the first place and possessors would have no reliable means of demonstrating that they were not forgeries.

The secret key of asymmetric encryption schemes satisfies the marking restriction attribute quite well. A digital signature having the stated properties therefore consists of data or a checksum thereof encrypted under the initiator's secret key. Authenticity may subsequently be demonstrated by successful decryption of the signature under the initiator's public key.

2.2.3.3 Notarization

Imposition of a third-party between transaction participants provides an independent means of proving additional attributes about their dealings. Such a notary is particularly effective in preventing the subsequent repudiation of participation in the transaction by either party. Required properties of the notary include trustworthiness, auditability, and responsiveness.

2.2.3.4 Data Integrity

The critical nature of this mechanism's use and effectiveness is self-evident. Data integrity is a fundamental contributor to reliable system

operation. In the context of protection, integrity lies at the base of the message stream modification threats hierarchy described previously. This attribute therefore makes it a very attractive intrusion target.

Two distinct mechanisms are generally provided in support of data integrity: one to address threats to single messages or individual fields thereof, and a second to protect a series of related messages comprising a peer connection.[5]

- The mechanism used to protect the integrity of a single message or its component fields is distributed among the sending and receiving peer processes. The sender tags each unit to be protected with a value which is a function of the unit itself. The receiver then compares the value received with that which it would expect to be associated with the unit to assess whether a modification has occurred to the message in transit.
- Various forms of time-stamping are employed in combination with the single message mechanism above to counter integrity threats to a series of related messages such as misordering, loss, replay, or data modification.

2.2.3.5 Access Control

An underlying network architecture has no bearing on this mechanism's role as the mediator of subject to object references. However, the reference monitor which conceptually performs this function must be extended to comprehend the distribution of subjects and objects across many nodes of potentially varying protection capability. Consequently, the reference monitor and its database are themselves partitioned to some degree across nodes comprising the distributed system as shown in Figure 2.2.3.5. Together, the cooperating reference monitor elements guarantee that transactions continue to conform to the overall system security model.

Such an organization presents many interesting architectural challenges.[6,31,32,33] Among those most relevant to this discussion is the need of the individual reference monitor elements to assume defensive attitudes given each's respective lack of control over its peer's activity. This is particularly crucial at the object end given the greater yield associated with compromises therein to the intruder. The object-side reference monitor must enforce access rights as there can be no guarantee that incoming references have been prescreened on the untrusted subject-side by definition.

2.2.3.6 Authentication

Identity is a primary consideration in access right mediation decisions. Thus, no attribute of an incoming object reference request should be more scrutinized. It is through application of this mechanism that a transaction recipient may gain that assurance. The initiator's suspicions with respect to the identity of its real-time receiving peer are also laid to rest as a consequence of exercising an effective authentication mechanism. Though less emphasized in this discussion, this subject-side assurance is also necessary to achieve effective protection.

The next section provides further insight into those proposed means of achieving this authentication mission which are pertinent to the XNS benchmark.

2.3 Authentication Survey

One category of generic protection research remains to be discussed prior to focusing directly on XNS and its Authentication-by-Proxy model. Lest the reader become inattentive from anticipation, however, it should be noted that nothing discussed thus far is more pertinent to this work. Of course, that of such importance which is lacking is a brief survey of those models by which authentication of the XNS flavor has been implemented. This section thus provides such a historical view as a means of heightening reader awareness of pertinent challenges, traditional responses, and the genesis of many of the techniques which were blended to form the XNS authentication mechanism.[2]

2.3.1 Architecture

The first notable architectural property of the models to be surveyed is a common reliance upon private key encryption as the means of achieving both message privacy and source integrity. This is consistent with the XNS mechanism. As it so happens, models of comparable complexity have also been developed employing public keys. Tradeoffs between the two key mechanisms lie in the strength and performance efficiencies of the specific underlying encryption algorithms, as well as the security provisions offered by the underlying machines.[9]

All are limited to address the protection threats of impersonation or replay at connection establishment in a system of loosely-coupled processes. Protocol robustness against other threat forms is indicative of those solutions likely to be effective in achieving protected networking generally.

An additional shared property is primarily the consequence of the requirement to support network extensibility to global proportions. The sheer magnitude and likely dynamics of the potential process set within such a domain eliminates consideration of the brute-force solution of maintaining a separate private key pair to service exchanges between each combination of principals.

Consequently, an authentication service is introduced to provide the essential functions of generating and securely distributing private, lifetime-limited, conversation keys. Therein lies the motivation for the service's alternate name: key distribution center. In order to securely pass messages between the authentication service and any principal, it must possess a copy of each's private key.

Given this role, it is evident that the principals depend heavily upon the authentication service's integrity. Its failure to perform reliably is disastrous to model operation. Its best-case failure scenario is

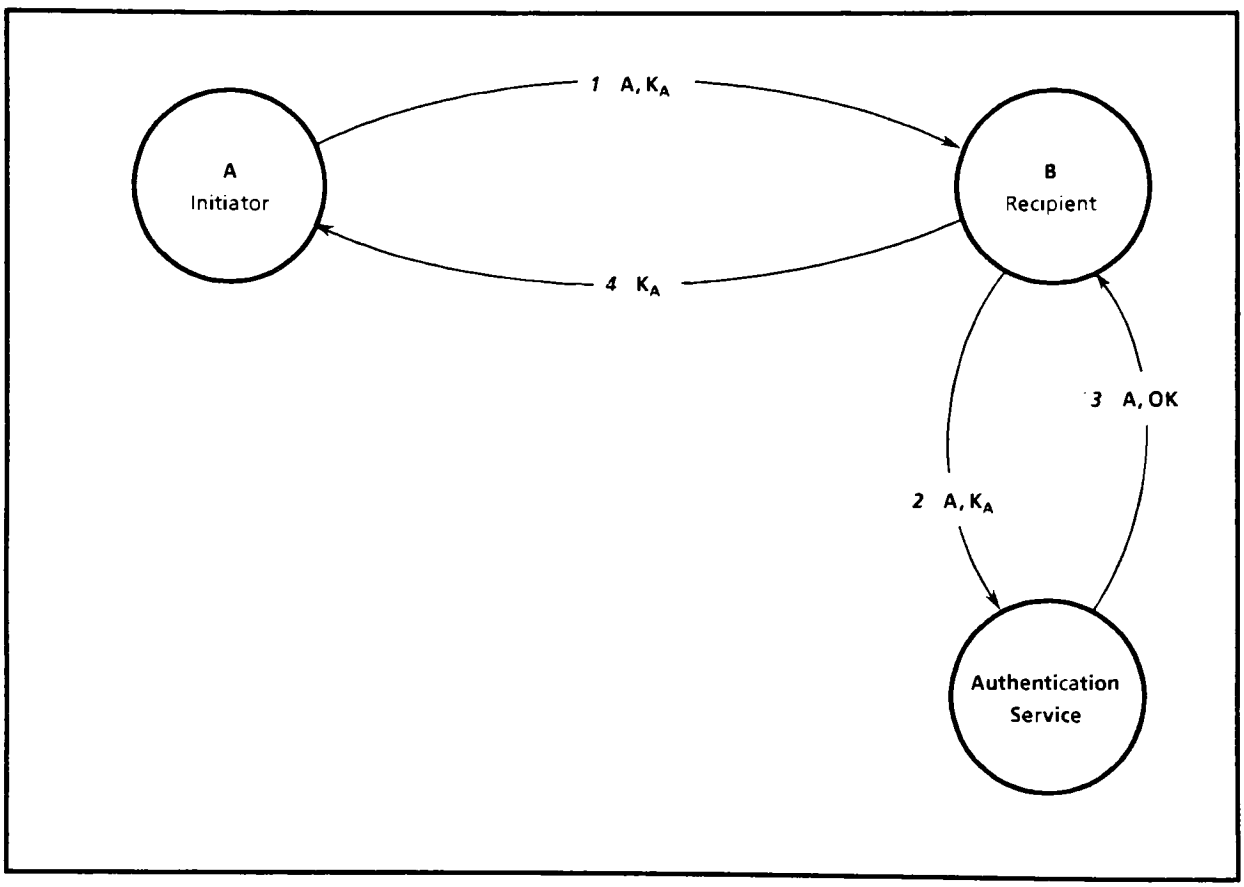

Figure 2.3.2.1: Simple Model[1]

unavailability, in which case underlying transactions cannot proceed securely. Thus, a potential denial of service attack could be targeted at taking the authentication service out of commission. A more crucial failure would be experienced were the authentication service's key database compromised.

Key privacy is fundamental to any encryption-based protection scheme. Immunity from compromise is largely dependent upon effective local security at each element participating in the authentication model, not merely the service. Additionally, the key space is expected to be large, sparse, and random as a precaution against brute-force attacks. The algorithm itself must be immune from cryptanalysis.

Finally, the authentication service is typically closely aligned with a distributed naming authority. This is an outcome of their similar role in translating name to property, which is specifically the associated encryption key as pertains the authentication service.

2.3.2 Models

The authentication service's private channel to each principal enables the significant achievement of guaranteeing a message's recipient of its original source. This obviously goes a long way towards convincing mutually suspicious principals of each other's claimed identity. While necessary, it is not sufficient to solve the authentication problem. The models below best illustrate why this is so.

2.3.2.1 Simple

The intuitive authentication solution is that represented in Figure 2.3.2.1. In this initial model, transaction initiator A forwards its identity and private key, K_A, to recipient B in cleartext. The crucial underlying premise is that knowledge of K_A is sufficient proof of the initiator's identity. B confirms the accuracy of this mapping through consultation with the authentication service. A's secret key accompanies future replies from B in order to satisfy A's suspicions relative to B. B's knowledge of K_A is equated with having received it knowingly from A.

This model is simple in the sense that all protection threats are assumed irrelevant. While not generally an appropriate assumption, certainly there exist limited applications within which it is. One likely example is that of a closed network of entry-level devices. Under such circumstances, it hardly seems cost-effective to incur the additional expense associated with defense against deliberate attack. Accidental threats remain a concern, however, as nothing prevents B, or anyone else, from inadvertantly impersonating A.

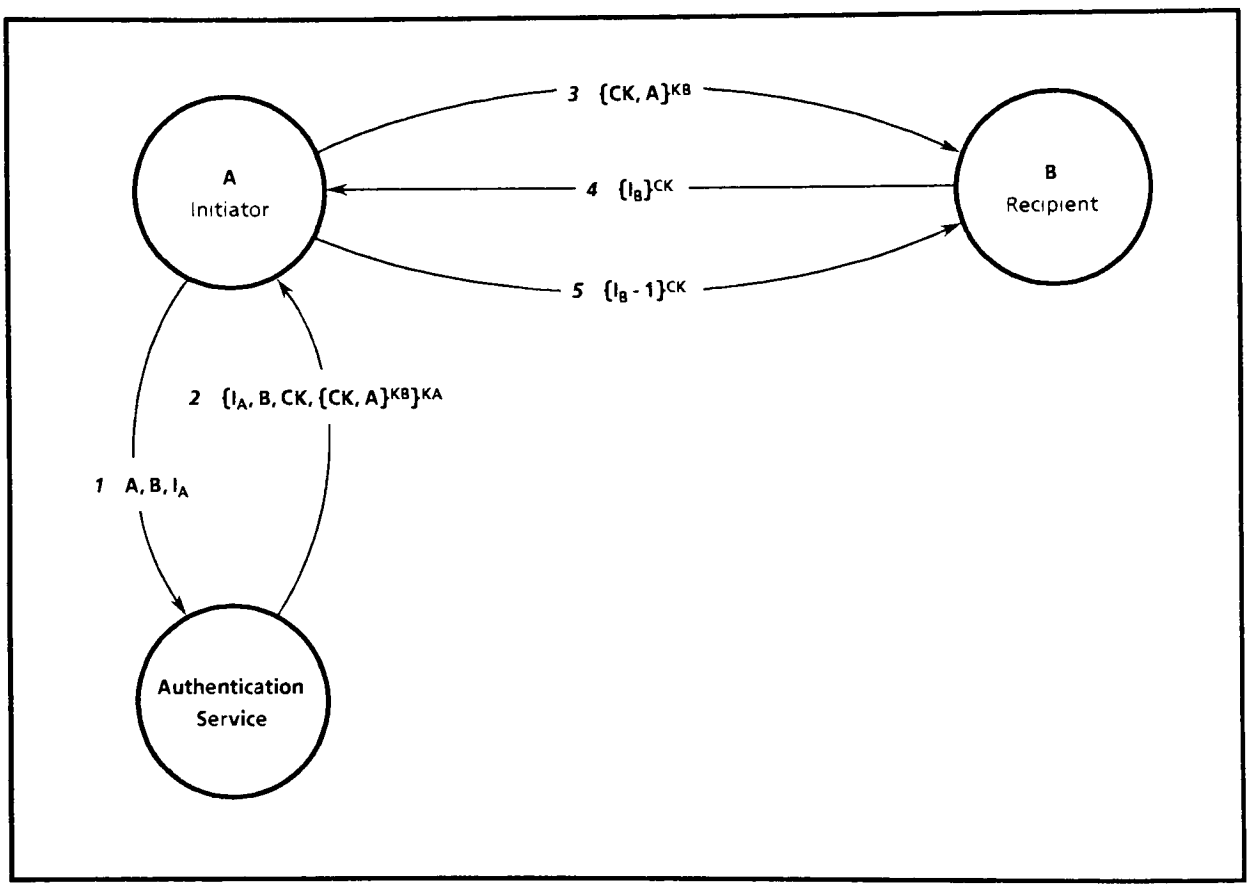

Figure 2.3.2.2: Base Strong Model[9]

2.3.2.2 Base Strong

Much of the groundwork in the area of authentication modeling is credited to Roger Needham and Michael Schroeder. Fittingly, their contributions comprise a significant percentage of that considered here. Figure 2.3.2.2 was their perspective of the fundamental private-key based authentication model as of 1978. Many attributes of this model are evident in subsequent efforts through deliberate emulation. Alternatively, the perceived shortcomings of their model often serve as a primary basis of proposed alternatives.

This model begins with the initiator forwarding the name of the intended recipient and itself with a nonce, I_A, to the authentication service in cleartext. The nonce is simply a non-repeating number which its sender, who in this instance is A, uses to insure the timeliness of replies in which it is contained. Thus, it serves as a defense against message replay attacks.

The authentication service constructs and forwards a private reply to A's opening message. A's nonce is included in the message as proof of message currency. Inclusion of the recipient's name guarantees A that it was not modified en route to the authentication service as a mean of inducing A to unknowingly interact with an intruder The private conversation key, CK, unique to this transaction is also returned securely to A in this message. Finally, the authentication service includes an authenticator, $\{CK, A\}^{KB}$, which is of no direct use to A as it is encrypted under B's private key. Rather, this is forwarded to B, who alone can acquire the enclosed conversation key and initiator's name.

Having done so, B is convinced that the authentication service alone could have generated the authenticator signaling A as the transaction's initiator as it alone has access to B's private key, KB. However, no accomodations have been made thus far to counter a replay of this message. While B could protect itself against this threat by maintaining a used conversation key table, such a design would be quite cost-ineffective. Rather, B employs the recently-acquired conversation key to issue a message containing a different nonce, I_B, to the initiator. As this message is discernable only by A, B can assume that a reply which is both a simple function of the nonce and is encrypted under the conversation key could only originate at A.

Both parties may now be relatively satisfied that the other is that whom it purports to be.

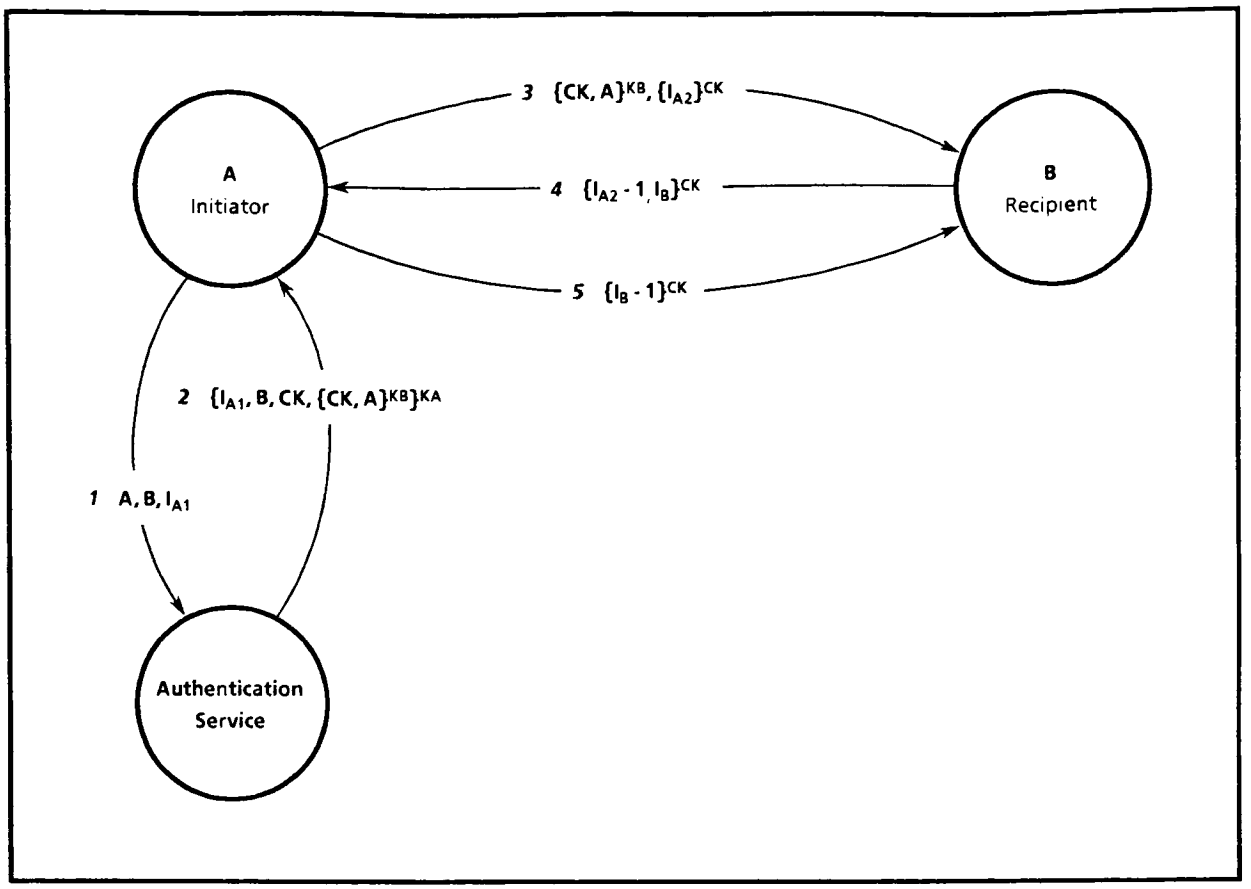

Figure 2.3.2.3: Cached Authenticator Model[9]

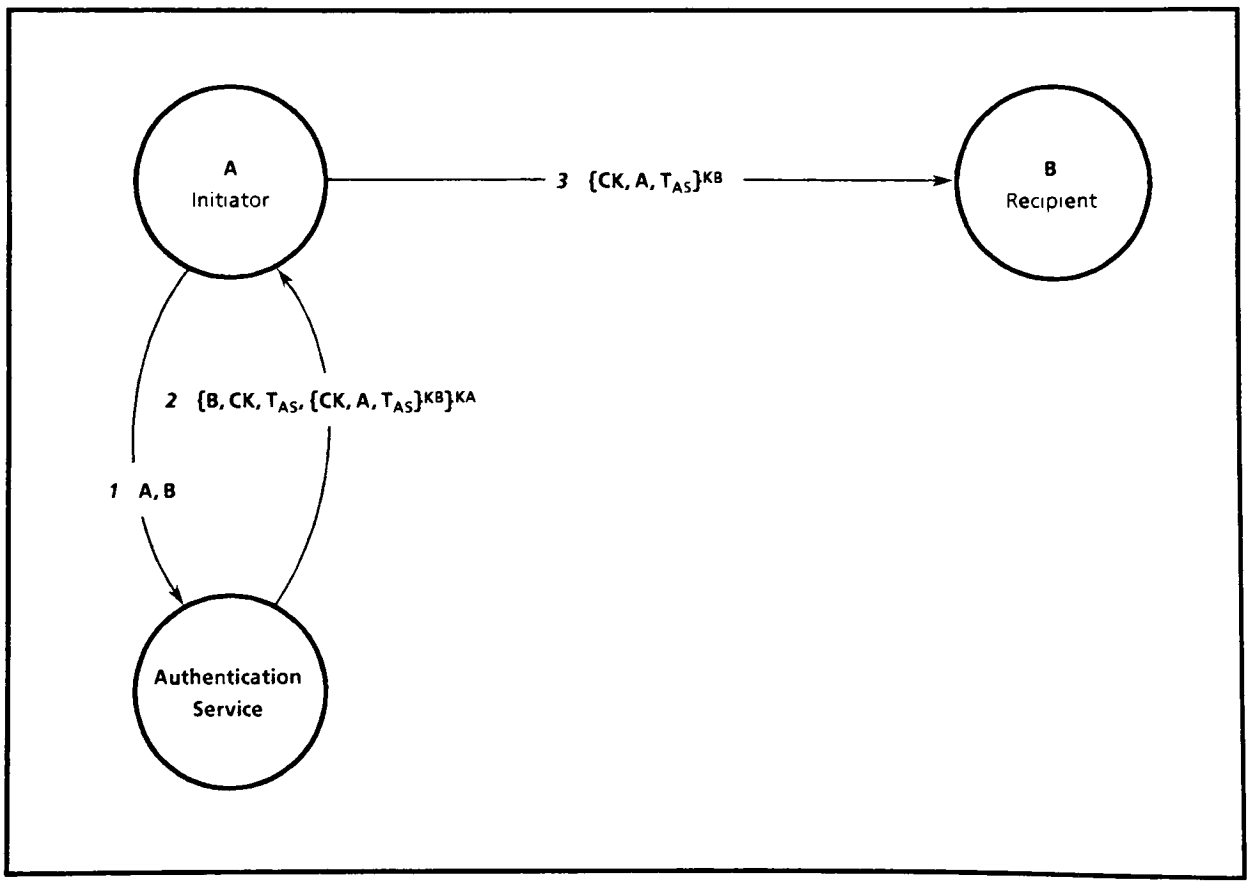

Figure 2.3.2.4: Timestamp Model[10]

2.3.2.3 Cached Authenticator

Needham and Schroeder observed that a performance shortcut may be desireable relative to their base model. That is, when two principals interact frequently, A may cache the authenticator $\{CK, A\}^{KB}$. Their resulting model is illustrated in Figure 2.3.2.3.

Steps one and two need only be executed once per principal pair. They remain essentially the same as in the former model. This model varies from that with the inclusion of a second nonce originating at A in steps three and four, however. Its purpose is to prevent replay of B's nonce in step four. Through such a replay scenario, an intruder could intercept each of A's messages to B transparent to A. This would amount to a denial of service at A. Presence of the function of A's nonce, I_{A2} 1, in message four is sufficient to convince A that B is in receipt of message three.

Key caching slightly increases the risk associated with conversation key compromise. This translates directly into a slight degradation in the identity assurance level associated with this model.

2.3.2.4 Timestamp

In the models above, the nonce has been exchanged through handshaking to convince its originator as to the timeliness of the connection sequence. Alternatively, Denning and Sacco propose the use of message timestamps as demonstrated in Figure 2.3.2.4. They argue that this scheme is both more robust and efficient than the base private key model.

In the unlikely event that a conversation key is compromised, they point out, the base model allows an intruder to impersonate the initiator indefinitely by first replaying the authenticator and thereafter intercepting and answering the recipient's challenge encrypted under the conversation key.

This model introduces the need for local clocks at the authentication service and the principals. It is further expected that these are periodically synchronized from a network time service. Given these additional facilities, this timestamp-based model requires two less messages than the base private key model as it eliminates the nonce-based handshake of steps four and five.

This exchange begins much as the base model, with initiator A notifying the authentication service of its intent to connect to recipient B. The nonce of the base model is replaced in the authentication service's reply with its current local time, T_{AS}. This is also included in the authenticator,

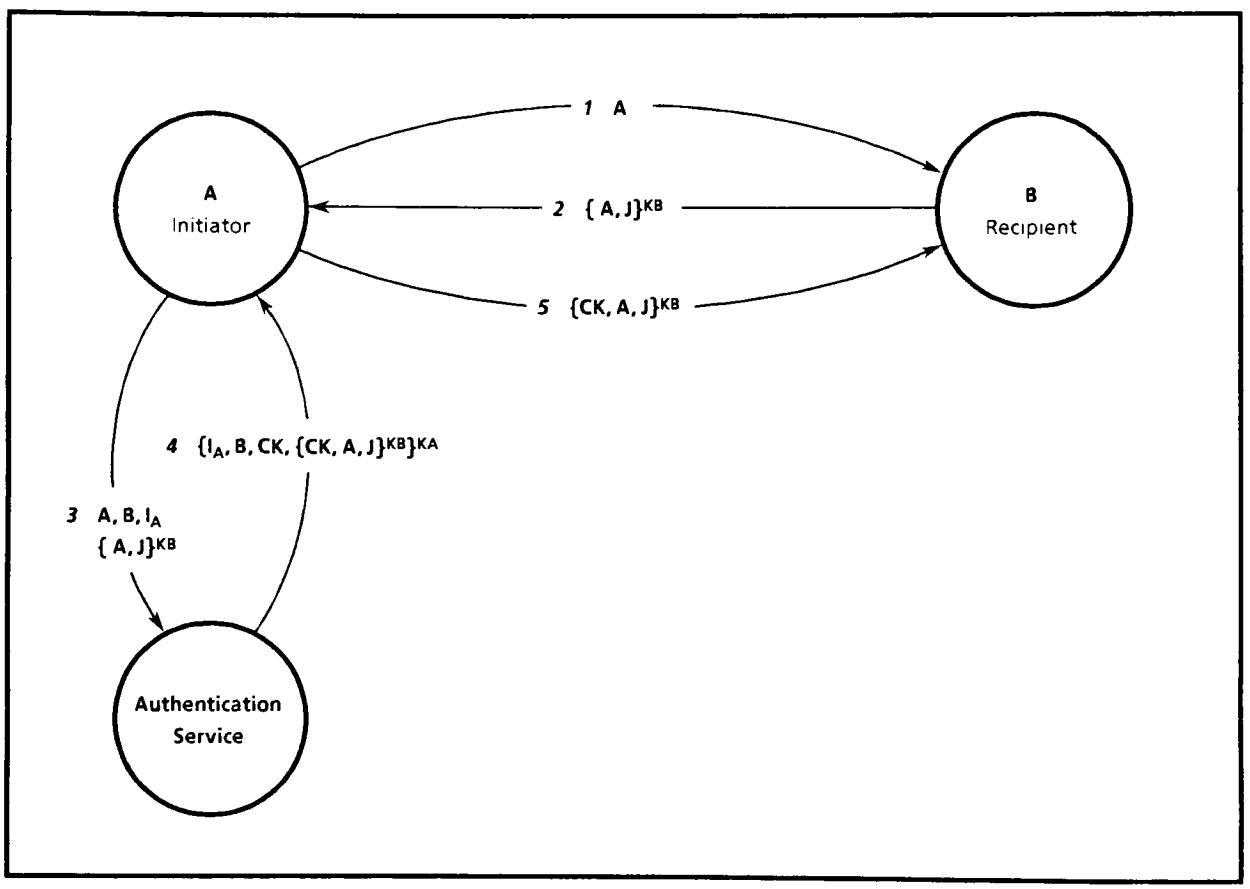

Figure 2.3.2.5: Nonce-Protected Conversation Key Model[11]

which is ultimately destined for B. Both A and B compare T_{AS} with their current local time. In each instance, they conclude that the message is not a replay if:

$|LocalTime - T_{AS}| < (\Delta t_1 + \Delta t_2)$, where

LocalTime : Current time as reflected by the principal's local clock

Δt_1: Expected discrepancy between the clock at the authentication service and that of the principal

Δt_2: Expected network delay time

The scope of this model's effectiveness is constrained by the magnitude of $(\Delta t_1 + \Delta t_2)$ as replays are undetectable within this interval. It is suggested that reasonable value for Δt_1 is between one and two minutes. Authenticator caching is also prevented as a side-effect of the planned-obsolescence characteristic of this model.

2.3.2.5 Nonce-Protected Conversation Key

The timestamp solution is not without drawbacks. In addition to the minimal opportunity for impersonation introduced by the $(\Delta t_1 + \Delta t_2)$ window, Needham and Schroeder point out that the cost of distributed clock maintenance cannot be overlooked.

Figure 2.3.2.5 represents their base private key model modified to accomodate the somewhat remote risk of conversation key compromise. The basis of the model is that the nonce should always be generated by the principal which seeks reassurance of transaction timeliness. Accordingly, they recognized that a mechanism had to be devised by which B generates a nonce for subsequent protected confirmation.

They achieved this goal by having initiator A first pass a clear text message identifying itself to recipient B. The response to this message is encrypted under the recipient's private key, as it is ultimately intended for use by the authentication service. The initiator's identity is included as a defense against modification of A's opening message to B. If undetected, such an event could result in a subsequent denial of service by B in step five. In addition, B generates and includes nonce J in its reply.

The contents of A's initial message to the authentication service of the base private key model is complemented with this protected response from B. Similarly, the authentication service's reply to A's message three remains unchanged other than by its inclusion of nonce J in the authenticator destined for B. Upon its receipt from A, B verifies that the enclosed nonce maps to that issued against the initiator.

This model differs notably from its counterparts in this survey. The other five models each established techniques which were subsequently

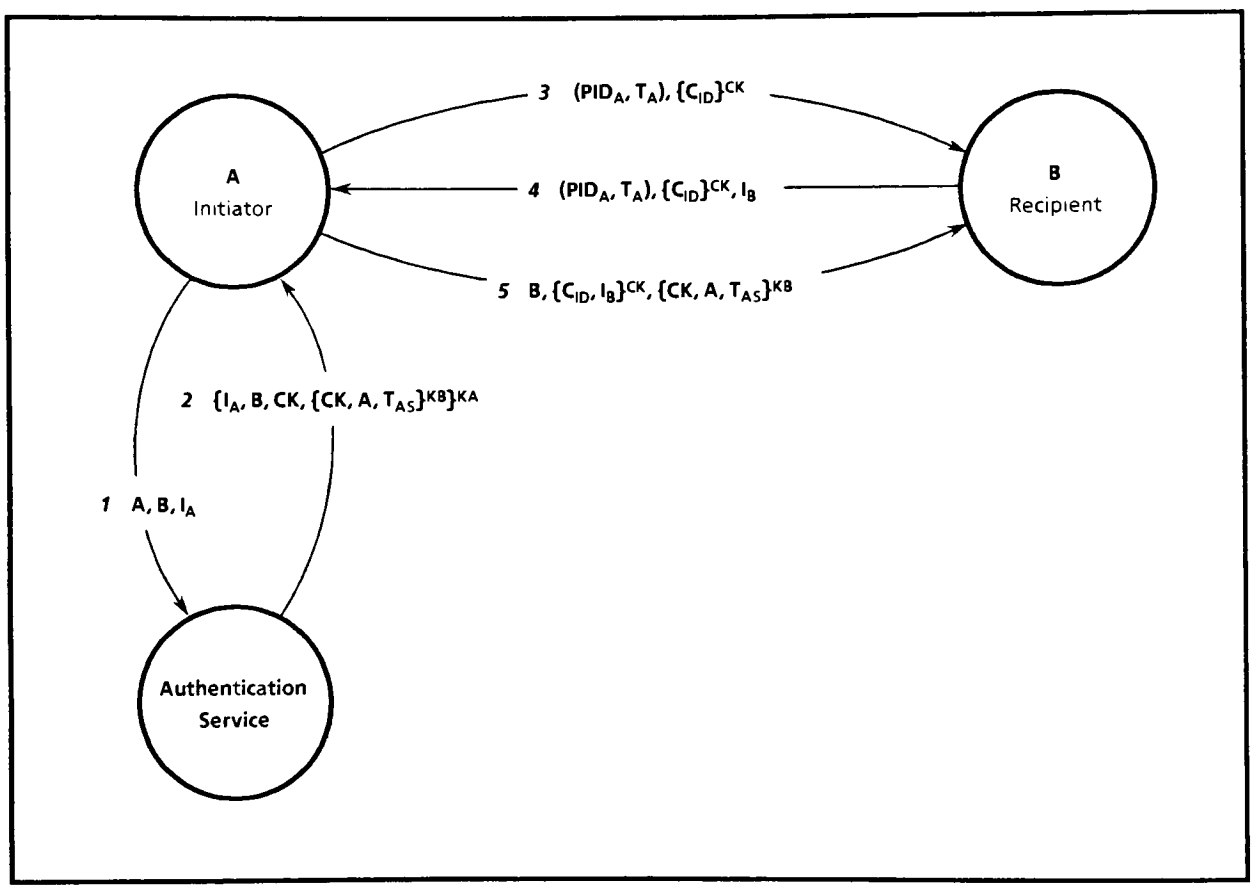

Figure 2.3.2.6: Remote Procedure Callback Model[43]

integrated and extended to form the XNS model. On the other hand, this model followed the XNS model in the historical evolution of authentication mechanisms. Its relevance to this discussion is as an example of contemporary technique, exhibiting remaining shortfalls and improvements against those which served as the basis of the XNS solution.

2.3.2.6 Remote Procedure Callback

The model of Figure 2.3.2.6 is an interesting hybrid of those discussed thus far in the sense that its reliance upon such standard techniques as nonces and timestamps is complemented with unique theme variations. The initiator/authentication service opening exchange represents a combination of the base model and that of timestamping. A nonce, I_A, is used to insure timeliness of the service's response to A. On the other hand, a timestamp is embedded in the authenticator. The associated semantics are themselves somewhat different from that of the earlier timestamp model: to limit the exposure available through private or conversation key compromise to a matter of a few of hours.

This model was developed by Andrew D. Birrell to serve as a key component within a secure remote procedure call facility.[43] Some of the attributes of his model thus uniquely accomodate its role within a security framework. In particular, ease of use at the remote procedure call level is stressed. Towards this end, the concept of a conversation is introduced. This is the incremental overhead incurred by the remote procedure call user in the interest of secure interactions.

For the purposes of this discussion, this concept is important for many reasons. Of these, the first is that by which it is identified. The (PID_A, T_A) pair of messages three and four serve this purpose. These represent a network-wide forever-unique concatentation of the initiator's processor identifier and local clock value. Semantically, the conversation identifier maps to the principal pair and conversation key within the largely-transparent authentication scheme in each end node. This enables transaction recognition and message decryption.

The concept of callback comes into play as follows. The initiator passes its choice of conversation identifier in clear text to the recipient along with the encrypted remote procedure call. At B, this is mapped into a table of known current conversations. Not finding an entry corresponding to this conversation, B is unable to decrypt the parameters of the protected procedure call as the conversation key is unknown as well. Consequently, B requests A to forward the authenticator it acquired earlier in message two. This request is protected by a nonce, I_B, per the lessons of the base private key model. The encrypted call identifier, C_{ID}, provides protection

against other forms of replay. Finally, A forwards the authenticator along with a set of fields recognizeable by now as insuring message timeliness.

Subsequent to this initial exchange both principals are in possession of the conversation identifier. B may thereafter act on the call immediately upon its receipt via message three. Another advantage of this scheme is in its inherent recoverability in the face of temporal recipient failure. The original request for authenticator sequence is repeatable on an as-needed basis within the window from T_{AS} to $(T_{AS} + \Delta t)$.

2.4 Xerox Network Systems

One of the prime organizational contributors to the historical evolution of the computing industry has been Xerox Corporation. Its corporate research and product development organizations, most notably the Palo Alto Research Center, have introduced a long list of revolutionary concepts in such speciality areas as personal computing hardware and software architecture, networking, protection, programming languages, and software engineering. Examples include the Alto personal computer, the lightweight-process operating system organization, the mouse user input device, the STAR desktop user interface metaphor, the Ethernet data communications mechanism, access control and covert channel paradigms, the Mesa programming language and development environment, and the Interpress page description language.[25,28,39,40,41]

All of these innovations support the corporation's objective of being the industry's leading supplier of integrated office system products. The glue which pulls together the individual products offering the various capabilities described above into a system is the Xerox Network System architecture. At the time of its introduction, XNS was a major advance into operational distributed operating systems from its more-common, less-capable, networked counterparts.[34] Though not an industry leader by placement measures, XNS continues to be a significant operational platform in the contemporary commercial office systems environment.

The remainder of the student's final report specifically pertains to this historically notable workstation-based distributed system. Of course, the emphasis of these tasks is the critique and subsequent enhancement of the Authentication-by-Proxy protection element of the XNS architecture. This can only be performed effectively in light of the larger issues of the system application and consequent general architecture. It is the purpose of the remainder of this section to exhibit such insight.

2.4.1 Application: Distributed Document Management

Among the unarguable conclusions one might draw upon an inspection of the contemporary international business environment is that its unprecedented dynamics preclude effective long-term performance by any but the most flexible of organizations. Counter to the traditional American treatment of labor as a production factor to be optimized, minimized, or even eliminated, experts now recognize that people represent the greatest leverage towards such flexibility.[35,36] The workforce must therefore be treated as the primary asset of any organization aspiring to sustained success.

Many conditions must be present within the organizational culture to ensure a workforce which is sufficiently prepared to respond to the ever-shifting challenges of the business climate. In addition to responsibility and accountability, knowledgeability must be imparted to its lowest levels. Not only is this a consequence of proper educational preparation, but also of efficient access to information regarding the daily affairs of the office environment. Office information systems exist primarily to support such essential communications.[15]

Within an electronic office system, the information to be exchanged is typically expressed as a document. A document is a structured organization of information designed to be communicated effectively with people. A document may be represented on various media including paper, video monitors or voice.[15]

Thus, efficient document management is the primary business of an office system. Many operations comprise this application such as creation, layout, editing, translation, retention, transfer, media rendering, replication, and archival. A particular office system may support different combinations of these functions depending upon its specific mission. Production publishing is an example among the more comprehensive of these, while electronic mail falls in the familiar low end of the functionality scale.[37,38]

In addition to the ability to tailor office system functionality to organizational information availability needs, its topology must be configurable to accomodate the using organization's geographic distribution. This tends to be a rather dynamic attribute of modern organizations. As a general trend, however, the geographic breadth of business transactions is far-reaching, often multinational. As most well-formed organizations conform to the Principle of Steps mentioned earlier, such distant communications are relatively infrequent in comparison to those occurring locally. Additionally, infrequent interorganizational exchanges need also be accomodated in open system environments. Topology optimizations may be applied consistent with these observations.

Such a situation forms a strong argument favoring a distributed organization in the case of document management systems. Other quality factors also support such a course including the familiar list of expandability, integrity, availability, and reliability. This is the niche which XNS exists to satisfy. The succeeding section describes the particular architectural characteristics of XNS which support this applications target.

2.4.2 Architecture

Through the 1970's timesharing and batch were the architectural responses to the recognition that office system interactions were clustered locally with comparatively few involving significant variance with respect to distance. The maturity of the computer architecture and manufacturing industry was likewise a very large factor influencing this approach. The economics of computing and communications through that era precluded production application of other options.

Early in that decade, however, researchers began to develop the notion that distributed systems would one day provide a viable alternative to batch and timesharing. Those at Xerox PARC were particularly successful in translating their visions into operational prototypes. Their research networks were subsequently translated into commercial products. The innovative Xerox Network Systems architecture serves as the backbone of the system those elements may be configured to comprise.

2.4.2.1 Data Communications

Among the key contributors to the results the PARC researchers achieved was the decision that a local area network would provide the necessary data communications foundation upon which to construct a distributed document management system possessing the requisite quality factors. Of course, Ethernet is the product of this foresight.

Xerox' specific research definitions for this mechanism encompass the physical and data links of the ISO/OSI model. These were proven usable through application to a large-scale internal research network. The Digital Equipment and Intel Corporations subsequently collaborated with Xerox to develop the Ethernet definition which has since been standardized as IEEE 802.3.[42] A proliferation of products employing Ethernet followed, sustaining its pace through the present time.

Figure 2.4.2.1.1 depicts the significant physical components of a standard Ethernet local area network. The most fundamental of these is the coaxial cable which serves as its transmission medium. This is the common data bus for communications among any combination of network devices, the node set within a constrained distance. The specific maximum distance between nodes on an Ethernet is a function of such factors as target channel efficiency, but is roughly intended to accomodate an office complex spanning a few kilometers.[15,20,42] The Ethernet standard calls for baseband signaling on the coaxial cable at a data rate of 10Mbps.

Message broadcasting within the local area network is the notable consequence of this bus architecture. The transceiver provides the associated significant physical layer functionality of channel contention detection based upon Ethernet's innovative CSMA/CD scheme. It also performs other expected physical layer functions such as signal encoding and decoding. Transceivers thus provide a standard unit by which network devices are physically connected to the coaxial cable.

The network interface implements data link functionality within the network device. The most notable element of this responsibility is the contention arbitration portion of the CSMA/CD scheme. Lastly, the drop cable simply transports data link messages between the transceiver and the network interface, enabling their physical separation as depicted.

Transparent to the network devices, individual Ethernet local area networks may be interconnected via point-to-point channels to form a wide area network as illustrated in Figure 2.4.2.1.2. In this way, the relatively light demand for transactions between distant nodes may be sufficiently accomodated in XNS. The specific WAN topology may be managed to attain targets against such system quality attributes as throughput, availability, and cost. That depicted in the example figure is minimally reliable, though connections can be formed between any two nodes if the three point-to-point channels are all available.

2.4.2.2 Distributed Systems Differentiators

Such is the state of the data communications component of the XNS architecture. Given this underpinning, the processor allocation and transaction models are further differentiating attributes of distributed system architectures warranting discussion. XNS is fairly simplistic in both instances by today's research standards, relying upon static processor allocation and client/server transaction models. Such contemporary concepts as load balancing and process migration are foreign to XNS.[34]

As for processor allocation, XNS nodes are statically classified as either workstations or servers. A workstation in this sense is any network device which is employed directly by an individual to perform document management functions. Personal computing as it is known today is a distant descendant of this vision from PARC researchers by which office system workers would each have individual access to complete customized computing systems. The name workstation is itself representative of the model by which such personal computers would provide most of the routine service required by the office systems worker. This represents the next logical step beyond the timesharing model in which each worker is presented a virtual view of personal

computing, but is constrained by the actual limitations of the underlying shared physical resources.

Economics dictate that some resources be routinely shared between individuals even in a distributed system such as XNS. Servers fill this need, offering capabilities which are most economically provided all network users through centralized facilities. Such resource management is based upon time-slicing in a global sense, extending across a set of users whose needs are typically accomodated locally by their workstation. Administrators are identified among the network user population to manage servers. Service functionality may be exported from a processor which simultaneously operates as a workstation.

Evident in this discussion is the fact that server access is on a demand basis. Therein lies the prime motivation for XNS' adoption of a client/service transaction model. Simply stated, a client is an entity which issues service requests, while a service is the entity which responds. Services are passive in that they do not independently initiate client contact. This is consistent with the further assumption that client availability is unguaranteed at any point in time, while services are expected to be accessible as the norm.

The transaction model relates to that for processor allocation in the following manner. Servers export their respective services to clients via applications-level networking components. Workstations are similarly comprised in part by components providing the required network client roles. A service may also act as a client of any other service.

2.4.2.3 Functional Decomposition

Figure 2.4.2.3.1 identifies the significant such architectural components of XNS according to their mapping into the ISO/OSI division discussed earlier. The layering is notably similar though expectedly imprecise. As before, an informal discussion of the functional partitioning is offered in Figure 2.4.2.3.2.

Layer 7 of the diagram is particularly informative. In addition to describing the prominent XNS Basic Application Services which comprise the key elements within the document management application, Authentication is presented as part of the Application Support Environment. Of course, we will eventually discuss the model of its operation in considerable detail. At this point, however, one should note that it is part of a group which includes the distributed naming and time functions, Clearinghouse and Time respectively. The general relevance of these to authentication mechanisms was noted in the survey of section

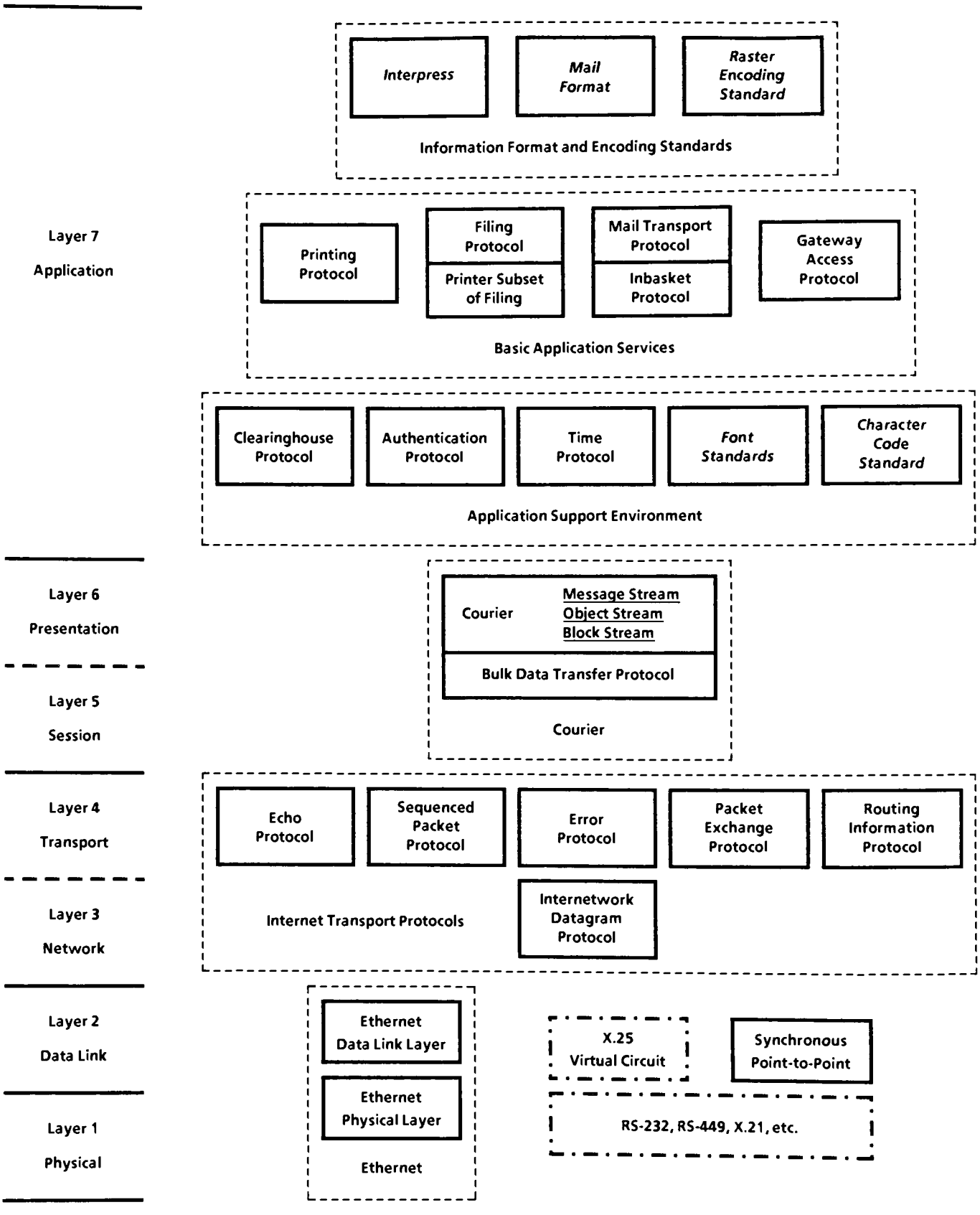

Figure 2.4.2.3.1: Architectural Mapping of XNS to ISO/OSI Layers[15]

Layer Name	Functional Description
Document Formats	Within the application layer, the standards for format or language for the encoding of document form or content are labeled with italic type. In many respects, the utility of XNS depends as much on the innovative approach to document descriptions as it does on the actual protocols. The document encoding techniques referred to in Figure 2.4.2.3 particularly Interpress - make it possible for XNS documents to be printed or communicated anywhere on the system. Other encoding standards are the Character Code Standard for representing text in many languages, and the Raster Encoding Standard for representing compressed and uncompressed bitmap images.
Application Protocols	At the application layer (ISO Model layer 7) the Application Support Environment provides support resources called on by users and/or the application protocols shown immediately above. These protocols - mailing, printing, filing, and gateway access - are implemented in hardware/software to provide the application services.
Courier	XNS implements the session and presentation layers in Courier, the XNS protocol for remote procedure calls (i.e. requests).
Internet Transport Protocols	Internet is shown as a set of protocols corresponding to ISO Model layers 3 and 4. The word 'internet' is also used to refer to the set of all interconnected Ethernets in different locations, a relationship implemented by these protocols.
Data Link	At the next lowest layer, Figure 2.4.2.3 shows the CCITT X.25 Virtual Circuit Protocol in a dashed box to indicate that this protocol is part of XNS by adoption. It is used as part of XNS utilization of packet-switching data networks.
Physical	At the lowest layer, Ethernet provides its own unique physical interface. It is unlike traditional data communication physical interfaces, which are shown in the box to the right (e.g. RS-232, RS-449, X.21). These are shown in a broken outline because, strictly speaking, they are part of XNS by adoption rather than by special design.

Figure 2.4.2.3.2: Functional Partitioning of the XNS Architecture[15]

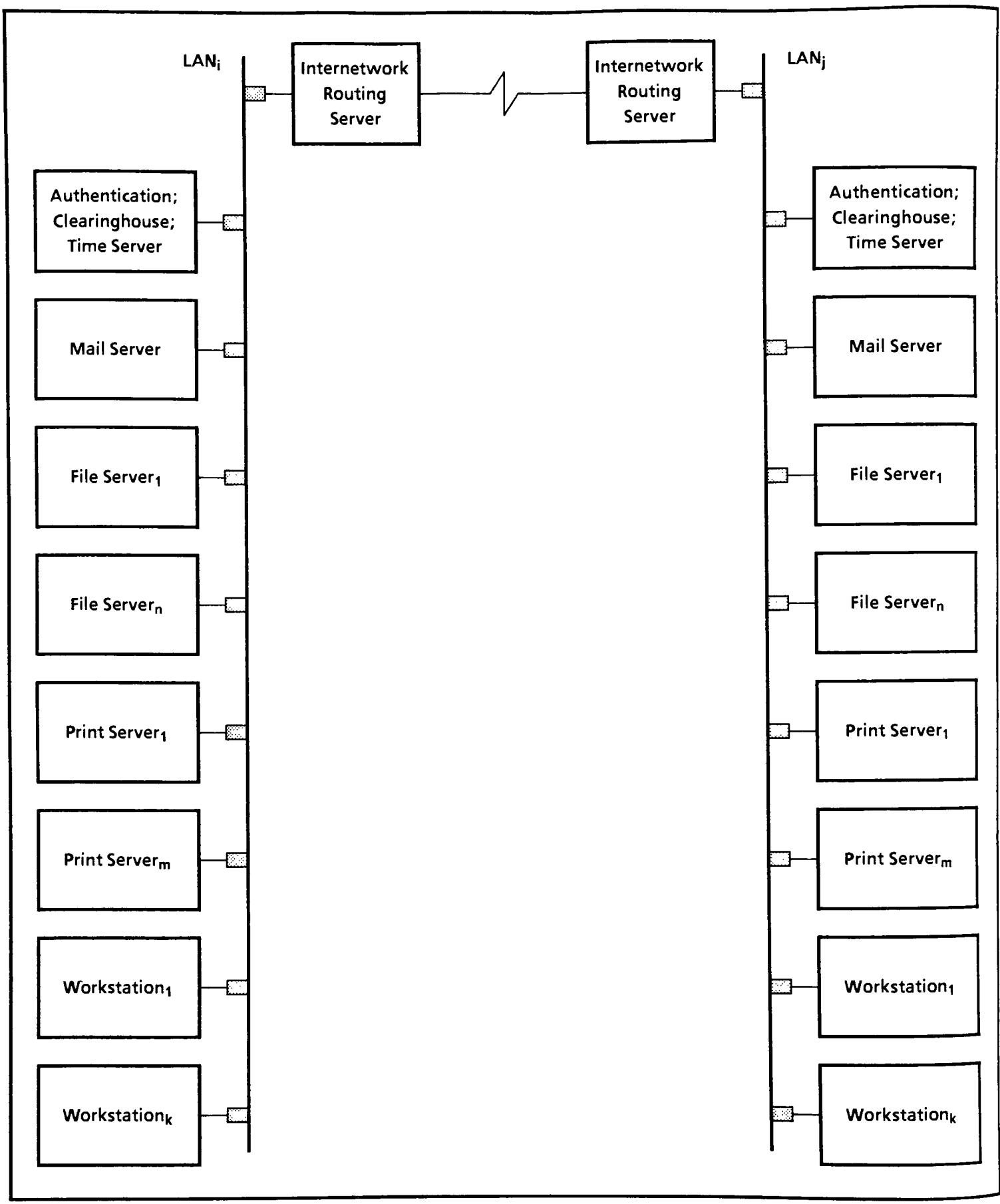

Figure 2.4.2.4: Typical Resource Configuration within an XNS Network

2.3. Due to their interdependence, the three services are often exported from a common processor.

A second important point is illustrated by the placement of the Authentication service in the lowest of three application sublayers. That is, applications service transactions may be authenticated according to the unique protection requirements of the particular service. For instance, filing and mail requests are expectedly authenticated. Print submissions are not. The choice in each case governs the service's ability to discriminate its offerings based upon the identity of the requestor. No accomodation for authentication is provided below layer 7.

2.4.2.4 User Perspective

All of the underlying architectural concepts disclosed thus far interact to support the user-view of an XNS network: the resource configuration. The customer's perspective of the system is at the level of the workstation, server, and communications equipment complement supporting their document management tasks. For instance, a geographically dispersed organization whose functions involve document creation, editing, mailing, printing, and archival would likely configure their XNS network along the lines depicted in Figure 2.4.2.4.

The system illustrated consists of two separate clusters, each of which is based upon a local area network of equivalent functionality. A variable number of workstations may be connected to each LAN, consistent with the size of the respective office system staffs. These would likely be employed for personal document creation, editing, and short-term retention activities. Broader organizational access to the documents produced thereby would be facilitated by electronic mailing and migration to a file service. The latter also contributes to document archival, demand printing, and production publishing applications. File server capacity is scaleable accordingly. Print servers are another important local system element, particularly for proofing the state of in-process workstation documents. Low bandwidth printers supporting such services could be complemented by higher capability print services should production printing also be a system requirement.

Utility application services also play a significant role in such a configuration. A Clearinghouse Server is necessary to facilitate such naming functions as address lookup. It also serves as the repository for the private keys employed by the authentication mechanism. The Authentication Server is thus hosted on the same processor to facilitate its reliable operation. Time services can be hosted nearly anywhere given their use of only commonly-available resources. As Authentication and Clearinghouse are important Time Service clients, colocation is common.

Finally, transparent to both the workstation users and application services, Internetwork Routing Servers forward traffic between resources on different LANs via point-to-point channels.

From an authentication perspective, one final note regarding such a network organization bears reinforcement. All traffic within a LAN is available to each processor thereon as a consequence of the broadcast nature of the underlying transmission medium. Inter-LAN traffic is processed through at least the Internet Transport Protocol components of each node on the adjacent local areas, as well as the connecting routers. This magnifies protection threats, particularly those of unauthorized disclosure, replay, and traffic analysis.

2.4.3 Authentication Model

Such architectural conditions as these combine with the applications focus of XNS to form the environment in which the specific authentication model of interest to this thesis need perform effectively. Recall further the significant influence which one would rightly expect to witness on the basis of such non-XNS factors as the generic protection theory cited earlier. That is indeed the case, as reflected in part by the number of similar authentication models presented in section 2.3.2.

In fact, the intuitive though simple authentication model of section 2.3.2.1 is a component of the XNS mechanism. This is a consequence of its architects' recognition of the important performance tradeoff between ease of system use and protection threat immunity.[44] Rather than embrace one of these conflicting customer requirements at the wholesale expense of the other, they assumed a more flexible posture. That is, the XNS authentication mechanism was made customer-scalable on the basis of individual application security needs. Those sites which prefer ease of use to stringent information security are satisfied by the minimal capabilities of the simple model. A strong model such as the other five surveyed earlier is provided to accomodate the needs of those installations of the converse orientation. This is yet another pertinent example of a protection mechanism which has been tailored to customer need.

The student's thesis is directed only at the proxy mechanism of XNS' strong authentication scheme, despite its availability in both forms. This decision is based on the observation that many of the objections raised against a proxy capability under a purportedly threat-resistant authentication mechanism are not customer requirements in the simple case. For instance, impersonation is not prevented in the simple mechanism as it is not expected to be an issue by definition. However, since the strong instance exists largely to prevent this threat,

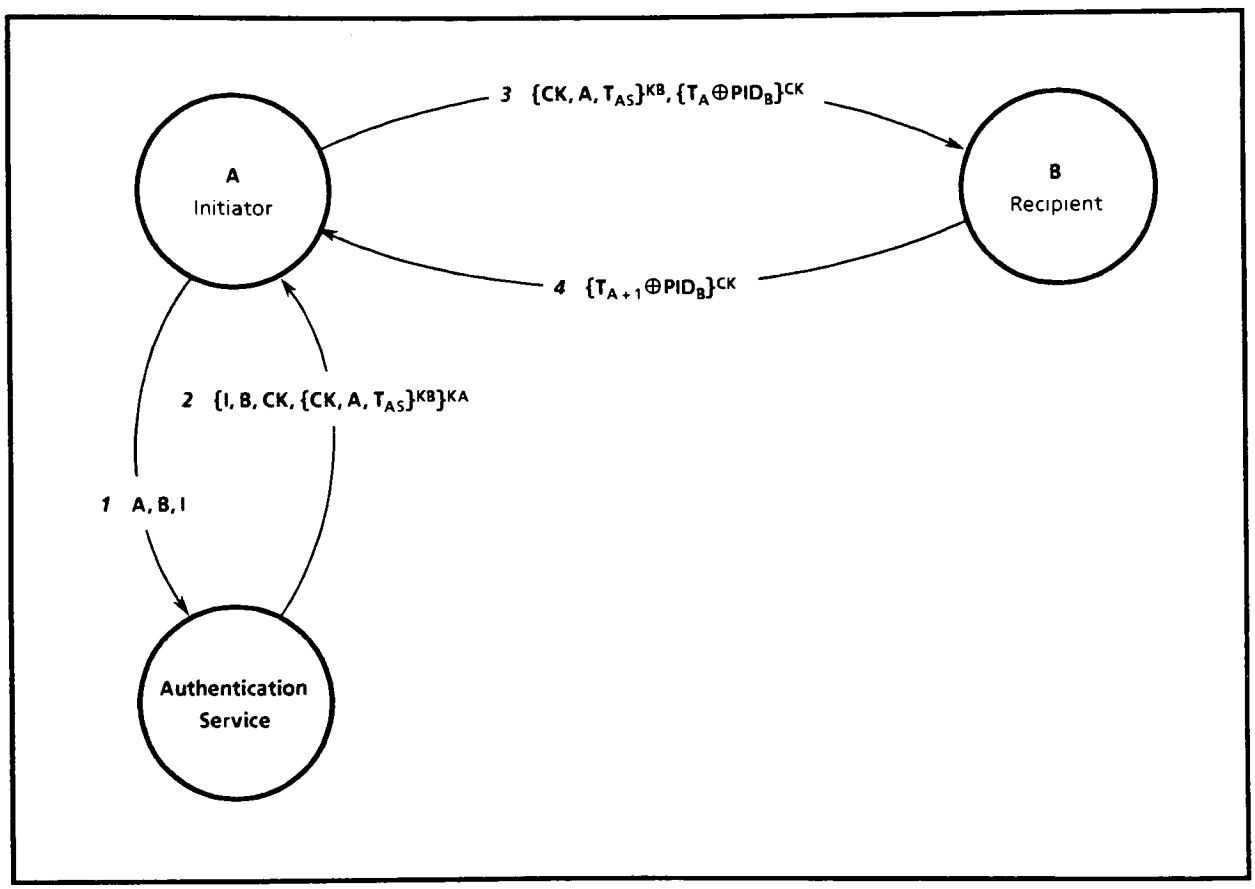

Figure 2.4.3.1: Base XNS Model[1]

characteristics of the proxy facility which undermine identity assurance genuinely represent a serious issue.

The remainder of this section thus deals exclusively with strong authentication under XNS. The base model is introduced first, followed by the unique properties associated with its proxy variant. This sequence is consistent with that of their respective development and standardization. The base model was completed in 1984, complemented two years later by accomodations for proxies. The student participated in the standard review process in the latter instance. This remains the current state of the mechanism's development.

2.4.3.1 Base

The base XNS authentication model of Figure 2.4.3.1 shares many traits with the remote procedure callback model of section 2.3.2.6, due largely to the cross-pollination effect of their Xerox ancestry. The fields comprising messages one and two of each are identical. A subtle difference lies in the semantics of T_{AS}, however. In this instance it represents the authenticator's expiration time. Thus, the authentication service sets the value of what was the callback scheme's Δt. As before, this is a hedge against key compromise on the order of hours.

The initiator forwards a verifier, $\{T_A \oplus PID_B\}^{CK}$, with the authenticator to the recipient in message three. The verifier serves two purposes. T_A works along the lines of the simple timestamping model. That is, it is compared by the recipient with the current value of its clock to attain a reasonable (on the order of minutes) guarantee of message timeliness. On the other hand, PID_B further limits the scope of the authenticator to only that instance of the recipient executing on the named processor. Implicit in this is the extended assumption that many instances of a single recipient may concurrently be active within the distributed system.

Finally, message four serves a role much like a nonce in that it is intended merely as a timely confirmation of the recipient's identity to the initiator on subsequent result passing. As usual, this is achieved by employing the private conversation key to encrypt a function on the verifier's contents.

2.4.3.2 Proxy

The XNS authentication mechanism is unique among the many studied by the student in one very interesting respect: the proxy. It is this specific entity which this thesis exists to enhance. To do so, one need first understand the motivation for its existance and the specifics of its inclusion in the architecture. Such is the purpose of this climactic final section describing published work pertinent to the student's impending original contributions.

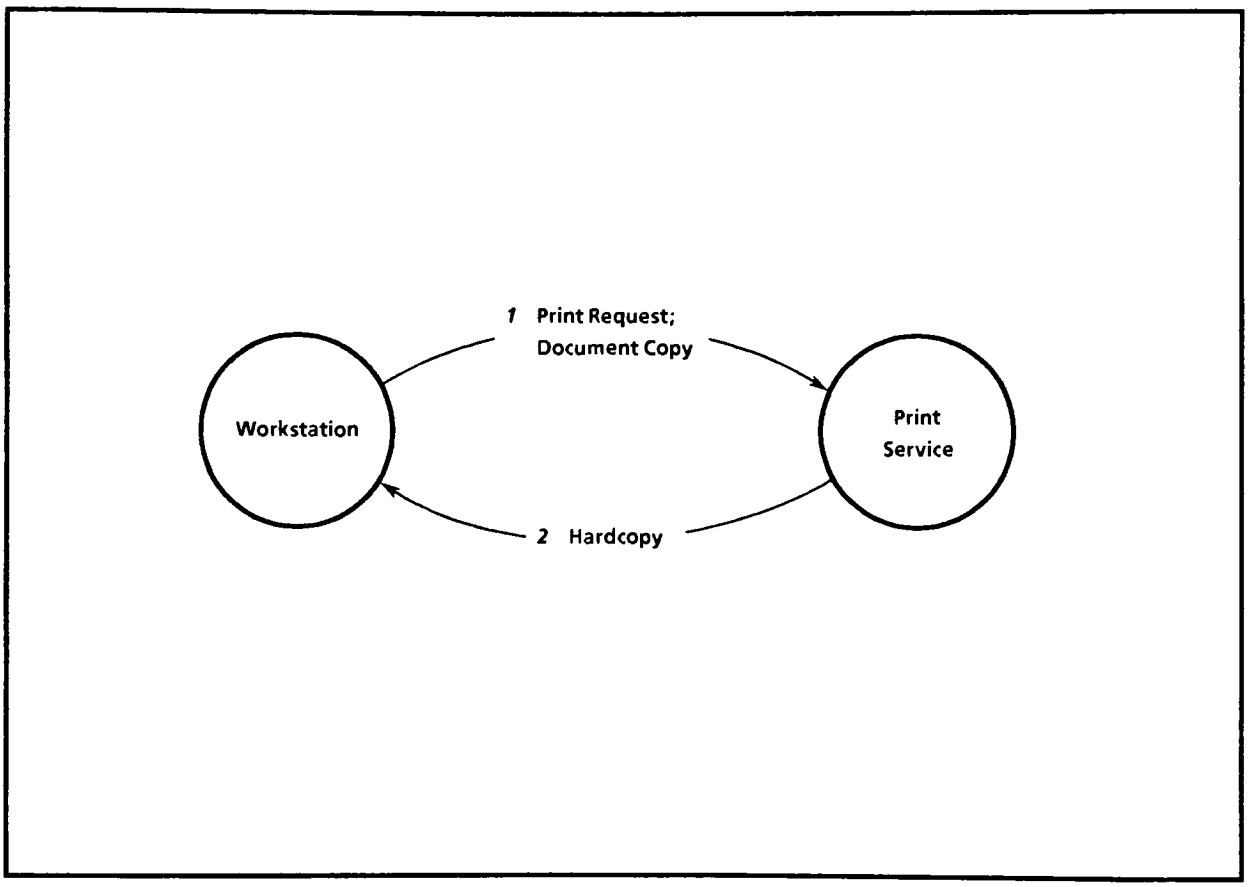

Figure 2.4.3.2.1.1: Workstation-Resident Document Printing Model[45]

2.4.3.2.1 Motivation

As noted in the discussion of section 2.4.2, print services are among the application layer elements of the XNS architecture. Such services are exported from servers which have been customized to efficiently generate hardcopy corresponding to the electronic document specified by the client. In the XNS architecture, documents to be printed are described according to the Interpress page description language, the precursor to the currently-popular Postscript PDL.[48] Print service execution of the instructions comprising an Interpress document yields the output images. The transactions by which such documents are submitted for translation likewise influence operation of the print service.[45]

The model of distributed printing which is most common within XNS is quite intuitive. As stressed previously, most of an individual's document management operations are serviced by their workstation. This notably includes creation, manipulation, and short-term retention. Thus, printing of work-in-progress documents typically follows the form of Figure 2.4.3.2.1.1. Per the discussion above, the workstation forwards an Interpress-formatted copy of the document to be printed to the print service with the associated control instructions. This is accomplished via XNS' remote procedure call and bulk data transfer facilities. Note in particular that the print service immediately copies the document from the workstation upon receipt of the print request, while it may not actually produce the associated hardcopy until some later point in time. This strategy is a consequence of a number of architectural assumptions, including the lack of guarantee that the workstation will be available on demand and that the document may change in the interval between request submission and imaging. Consequently, sufficient memory resources must be configured at the print server to support the necessary balance between cost and availability.

A second distributed printing model is more appropriate in the case of documents which are resident on file services. Among the applications in which one would likely expect such documents to exist are demand printing and production publishing. In the first instance, documents have been migrated to a file service to facilitate their long-term retention in a static state, against which the need to efficiently replenish a limited hardcopy supply occasionally arises. In the latter, the document creation and manipulation tasks are beyond the capabilities of the individual workstations, with a central source providing integration services. This may be a similar variant on the current file service theme.[37]

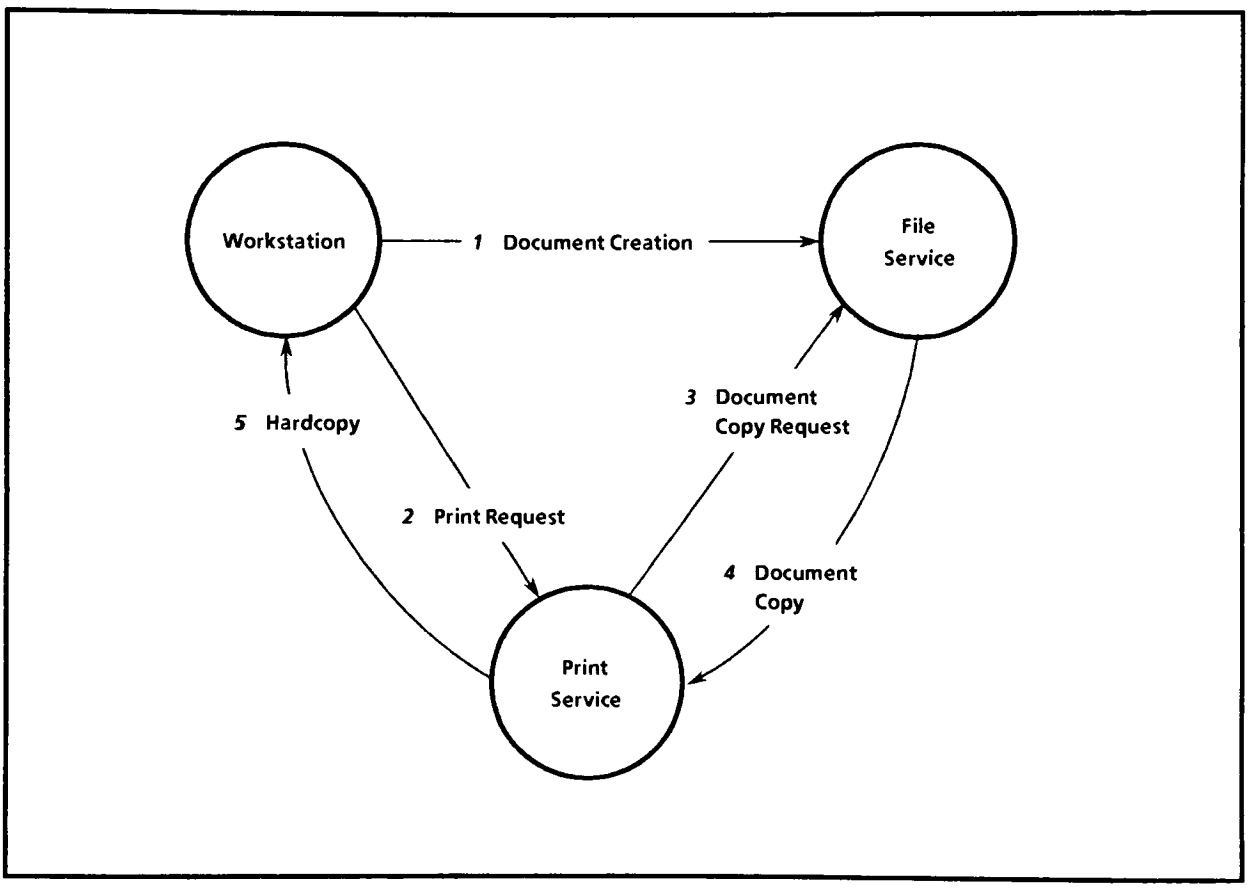

Figure 2.4.3.2.1.2: File Service-Resident Document Printing Model[1]

Figure 2.4.3.2.1.2 presents a model more suitable to such applications. Note that an Interpress document is created at the workstation and migrated to a file service at some point in time preceding the printing need. Thereafter, the workstation issues a print request to the service consistent with its user's interests. In response, the print service copies the document directly from the file service, enabling production of the desired hardcopy. This model offers many efficiency opportunities including resource scheduling optimizations at the print service, workstation productivity, and modular print document construction.

The authorization considerations of the workstation-resident document printing model are straightforward. One reason for this condition is the fact that the current set of XNS print services do not discriminate among print request submitters. In other words, no print client requests are rejected as a function of authorization. Were that situation changed in the future, the arbitration of such requests would remain convenient as the workstation possesses its current user's protection context. Requests could thus be mediated based thereon against the access control rules in effect at the print service. Secondly, the document to be printed is readily accessible to the service through immediate contact with the supplying workstation. The print client's authorization to copy the document to the service is mediated internal to the workstation based upon the properties of its current user. Thus, no particular accomodations for this transfer need be made relative to protection under this printing model.

Such is not the case with respect to the file service-resident document printing model. The comments above pertaining to mediation of the print request are equally applicable in this instance. Document accessability to the print service differs significantly, however. This is a consequence of the workstation's temporal dissociation from the file transfer. As one would reasonably expect, XNS file services apply protection to the objects stored therein. Once a document is migrated to the file service, all operations against which it is the target are mediated according to the file service's protection model. This is currently discretionary, implemented through access lists.[49]

The first order problem is to insure that the print client cannot compromise the file service protection mechanism by dispatching a print service to acquire a document for which the workstation user does not have rightful access. In addition, the interests of the workstation user need be insured in that the print service should be given access only to the precise degree authorized by the client. This precludes such scenarios as spurious print service operations against file server-resident

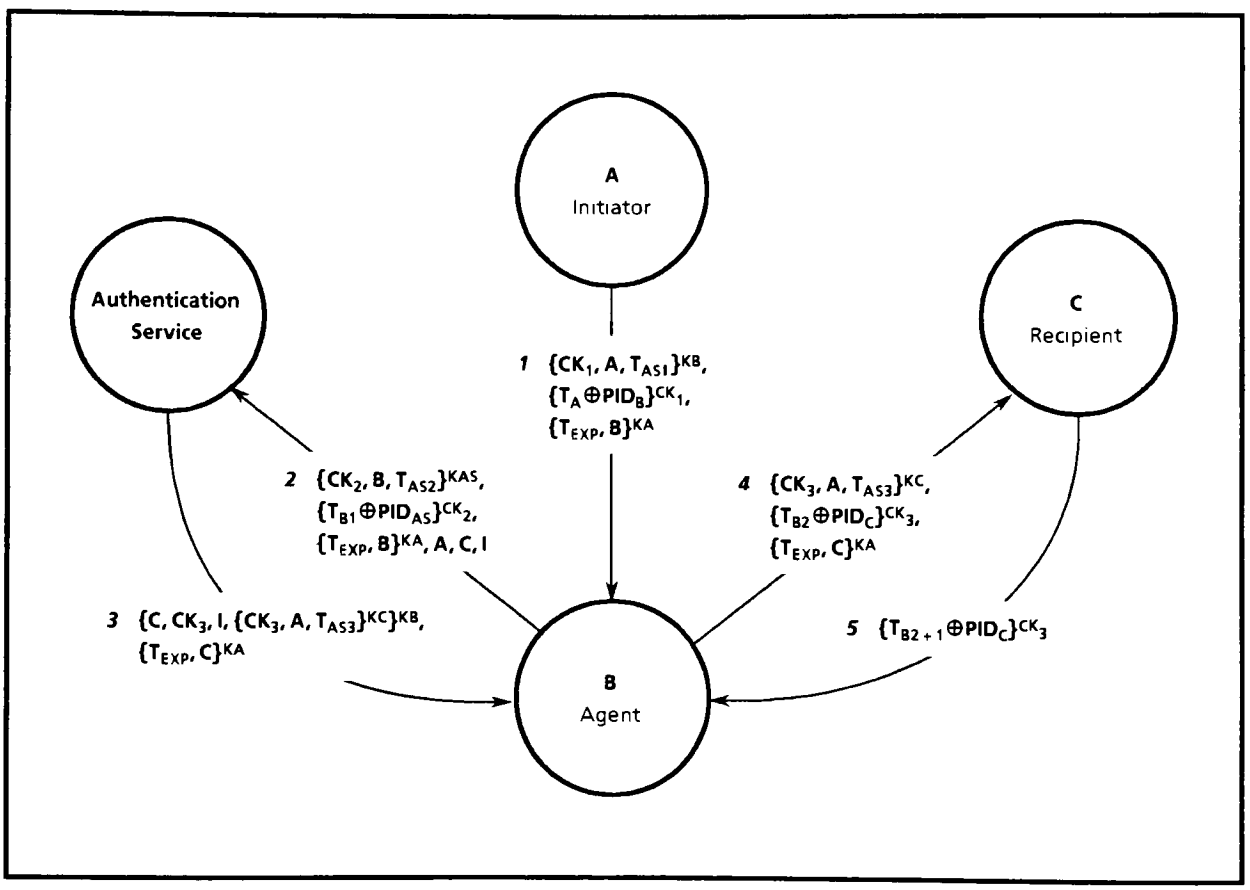

Figure 2.4.3.2.2: XNS Authentication-by-Proxy Model[1]

documents on the assumption that a workstation user has issued prior blanket approval to that effect.

This application is specifically that motivating the proxy facility within the XNS authentication mechanism. Many others of a similar nature could also be described but have not been to the student's knowledge. In any case, the thrust of the matter is to temporarily convey the print client's identity to the service so as to facilitate satisfaction of the file service's authorization mechanism on the document retrieval operation. Such a solution effectively addresses the problems cited above in that a print service operating under its requestor's identity is treated identically thereas by the file service's access control mechanism.

A proxy thus represents the means by which any element within the XNS architecture may delegate its identity to any other for a specified interval. This is a unilateral decision exercised by a transaction initiator, completely transparent to the recipient. Strangely enough, this element of the authentication model intentionally facilitates impersonation on the premise that it is in the best interest of system performance.

As is true of many software manifestations, the conditions associated with this construct closely mirror those of well-established interpersonal techniques. Legal mechanisms have been specifically established to contend with the case where one party wishes to commission a second to temporarily conduct business transactions on its behalf to some degree. Such is the legal concept of an agent. Of particular note in that sense is the fact that the transaction recipient must be presented proof of the initiator's delegation of authority to the agent. Such a proxy represents the agent's right to engage in transactions to the degree specified with complete responsibility for the resultant effects transferring to the initiator.

2.4.3.2.2 Model

The specific architecture of the proxy mechanism within XNS' strong authentication model is offered in Figure 2.4.3.2.2. Message one is similar to message three of the earlier model, consisting of an authenticator, verifier and an added proxy field. Contained therein under protection the initiator's private key is an expiration time after which the proxy is no longer valid. This alone limits the agent's subsequent ability to impersonate the initiator. The agent is also named in the proxy in order to void its utility to any intruders.

The authentication service redeems the proxy in messages two and three. Of significance are the observations that a proxy may be redeemed any number of times within its window of relevance, to enable secure

communications with any other principal. This message pair is analagous to the first two of the original authenticator scoping model other than the fact that the agent must first have established its identity with the authentication service. The authenticator provided in this message enables to service to constrain proxy redemption to only the party named therein. Other key differences are that the initiator is represented as A in the authenticator and a second proxy cascading A's rights to B further to C is returned by the service. B then has the opportunity to employ messages four and five to seek C's services as A, as well as to delegate C to act as A's agent. Proxies are thus transitive nature, without constraint by the initiator.

3 Theoretical Development

This section represents a significant shift in the nature of the student's thesis work. Thus far, the task has been to identify and integrate those existing research contributions which comprise the foundation of knowledge associated with the topic of interest. Incremental to that accomplished end is the development element of research: enhancement of the current state of affairs.

Innovation is thus the significant attribute of subsequent final report content. Such contributions are divided rolewise between two major deliverables according to the student's interpretation of the recommended thesis development process.[16] Section 3.1 establishes a set of available enhancement opportunities. These then represent the requirements target against which architectural adjustment recommendations are delivered in section 3.2.

3.1 XNS Authentication Model Enhancement Opportunities

The candidate enhancements to be suggested in this report specifically concern the XNS Authentication-by-Proxy model. Their identification is the consequence of student analysis of the existing mechanism in light of the body of established protection research presented in Section 2. Each such opportunity is thus notable in that it is uniquely the product of the student's thesis development. It is on that basis as well as the innovation represented by the associated architectural development to follow that the student contends his product qualifies as a Masters-level thesis.

A number of enhancement opportunities exist in the realm of the strong XNS Authentication base model. As the effectiveness of the proxy mechanism of interest is fundamentally dependent on that of the base, they also need be identified. Such a dependency implies a natural order to this discussion - base opportunities thus precede those unique to the proxy. Recall in both instances that the authentication mechanism exists solely to prevent the threat of spurious association initiation. To be effective, the mechanism must itself be immune to the distributed systems protection threats outlined earlier.

3.1.1 Base

The architects of the strong XNS Authentication base model integrated a broad variety of the techniques introduced by the models of section 2.3.2. Among the most prominent of these are the concepts of nonces, timestamps, and authenticator caching. Yet in most instances, these ideas have been modified as applied to the XNS model. While the resulting transformations are generally quite effective, a few are not. This section highlights those of the latter flavor.

3.1.1.1 Conversation Liveness Assurance

As noted through its introduction in section 2.3.2.3, cached authenticators are a performance efficiency mechanism. They eliminate the need to contact the authentication service as a precursor to each conversation between a pair of principals. Among the associated benefits of this reduction in message traffic are decreased channel contention and conversation duration. In addition, denials of service consequent to authentication service unavailability are avoided proportional to conversation locality and the length of the caching window.

In the original cached authenticator model, a nonce is passed as ciphertext under the conversation key with the authenticator as proof of message liveness to the recipient. A timestamp fulfills the same function in message 3 of the base XNS model. While this has the intended effect of eliminating the handshaking associated with a nonce-based exchange, it is not clearly an effective means of guaranteeing conversation liveness given the complementary XNS architecture of which this is a component.

Timestamps are arguably both quite costly in terms of overhead and technically complicated in terms of synchronization. Few, if any, economical architectures have been identified by which distributed clocks may be accurately maintained over any significant interval of time. In fact, so illusive has been an effective solution that Mt. Sinai Hospital in New York is reported to have adopted a solution whereby closed circuit television monitors throughout the complex are set to a common channel on which the image of an analog clock is broadcast.[50]

In addition to this generic trait of timestamps, it seems a particular dilema exists associated with their use in this application. The magnitude of the timestamp acceptability window should be no longer than that required to accomodate expected subnet delays. Any slack time beyond that represents an unnecessary period of vulnerability to the threat of spurious association initiation as a consequence of message replay. This is exacerbated in those XNS instances where the underlying net is physically Ethernet-based. Of course, such is the norm.

Two traits of Ethernet contribute to this situation. Due to the random backoff nature of contention arbitration under the CSMA/CD scheme, it is difficult to tightly-constrain the magnitude of the window without incurring an unnecessarily large number of connection rejections. At the same time, rapid message replay is quite feasible given the broadcast nature of the subnet. Wide area networking further exposes the replay vulnerability associated with such a use of timestamps as it necessarily extends the window.

3.1.1.2 Identity Assurance

The message reduction efficiency associated with timestamp use in authentication models was the XNS architects' motivation for its utilization in the mode above. However, it was the secondary justification for its original application to the model of section 2.3.2.4. Denning and Sacco argued that their timestamp use eliminated the need to employ the conversation key as part of the authentication exchange. This was presented as a significant opportunity to strengthen the identity assurance yielded by the mechanism given the potential of conversation key compromise and subsequent prolonged exposure to impersonation of either principal by the intruder. Therein also lies the motivation for inclusion of the nonce-protected conversation key model in the survey of section 2.3.2.

The base XNS model suffers this vulnerability given its utilization of the conversation key to protect the verifier. This is consistent with the cached authenticator model upon which it is based. No model has been identified which both facilitates caching and prevents impersonation as a consequence of conversation key compromise. This represents an enhancement opportunity, short of which it would seem caching need be sacrificed in favor of greater identity assurance.

3.1.2 Proxy

Many noble system architectural goals are achieved by the Authentication-by-Proxy model central to this thesis development task. The most fundamental of these is its ability to satisfy the immediate need motivating its introduction: protected file-service resident document accessibility to print services. Better still, such hardcopy generation efficiencies are enabled through the proxy mechanism transparent to all architectural elements but those participating in authentication exchanges. In the minimum case, this includes the authentication service itself and the party on whose behalf an agent is to be dispatched. As the reference monitor abstraction demonstrates, such scope constraint generally enhances protection mechanism assurance as a consequence of such crucial quality factors as isolation and verifiability. Run-time performance and system maintenance cost minimization are additional beneficiaries of positive impacts consequent to such a restricted solution. In addition, application generality extends proxy mechanism utility.

Yet were such complimentary observations the complete proxy story, this research would rightly conclude short of the innovative development required of thesis efforts. The remainder of this section accordingly

identifies a number of categories in which the student contends the proxy mechanism is insufficiently equipped to fulfill reasonable expectations of a protection mechanism within the broader XNS architecture. The basis for such claims rests with the student's analysis of proxies against the established research contributions cited in section 2.

3.1.2.1 Identity Assurance

An authentication mechanism's sole purpose is to provide each principal to a potential conversation confidence in the accuracy of the claimed identity of its counterpart. Its ability to achieve that end is the primary basis by which its utility is assessed. Such other quality factors as performance efficiency are distant secondary concerns to the degree of identity assurance afforded.[2]

While the concept of a proxy does not fundamentally clash with this end, the manner in which it has been architected into the XNS authentication mechanism deliberately compromises the primary authentication goal of identity assurance. As explained previously, a straightforward use of the proxy option is visible to the initiator, its agent, and the authentication service. Notable by its absence from this list is the recipient, who is completely blind to the fact that other system elements may be cooperating to establish an inaccurate identity for the party requesting its services. Equally as likely is the fact that the claimed initiator identity is correct. Consequently, the recipient may no longer place any confidence in the accuracy of the claimed identity of its peer.

Interestingly, confidence in the recipient's stated identity is not compromised as a consequence of proxy use. This is due to the authentication service's use of the private key of the recipient to form the authenticator. It is thus legible only to the actual recipient by definition of its unique possession of the key by which the authenticator may be returned to cleartext. The initiator's interests in confirmation of its counterpart's identity are therefore unaffected by proxy application.

In spite of such sound assurance with respect to the recipient, initiator impersonation alone represents a gapping protection shortfall. As mentioned at the outset, identity is a significant element of access control decisions in both mandatory and discretionary schemes. Initiator identity is particularly significant to such decisions in a loosely-coupled system organization such as XNS, where services must assume a defensive policy in which they apply access controls locally according to their own particular protection interests. Such a strategy is based on the lack of assurance that incoming requests have been reliably preauthorized.

This extensive leverage which initiator identity applies to access control decisions is precisely the property which is abused by the current proxy mechanism under investigation. As the access control mechanism assumes accuracy of the identity upon which its decisions are based, lack thereof has the direct effect of proportionally compromising the mediation effectiveness of associated references.

In the classical proxy sense, the recipient is aware of the fact that it is dealing with a particular party as an authorized agent for the transaction initiator. It may vary its behavior accordingly based upon such distinctions. The XNS approximation to this model completely short-circuits the proxy process from the recipient's perspective - it is unaware of the fact that it is not in fact dealing with the claimed party who is authorized to perform the requested operation. The access control mechanism is thereby extended beyond the explicitly defined bounds to include decisions which serve the initiator's self-interests. Verification of the consistency of such obscurely mediated-references against the stated policy is a significant challenge. The additional reference monitor properties of isolation and mediation are likewise weakened.

3.1.2.2 Authorization Scoping

Beyond the major opportunity for proxy facility improvement which would accrue from reconsideration of the current impersonation strategy, another set of slightly lesser importance deal with constraining the delegation of authority consistent with the needs of the operation to be performed. That is, there is minimal accomodation in the current proxy mechanism to scope the level of the agent's operation on behalf of the initiator. The only provision to do so comes with respect to time even this is minimally restricting as explained below.

A proxy is comprised in part by a timestamp chosen by the initiator. This value represents the point in time after which the agent is no longer commissioned to operate on the initiator's behalf. Selection of this value is somewhat imprecise, as it reflects the initiator's assessment as to when the operations to be performed by the agent will be completed. Such estimation is likely to require the inclusion of a significant error factor even were the initiator to be diligent in its minimization, an interest counter to norm. Excessively large windows extend the threat of proxy abuse, while insufficient windows prevent successful completion of the delegated tasks. In either event, because the proxy is solely regulated by time, there is no direct association to revocation upon completion of the action of interest.

Secondly, the timestamp solution is similarly suspect in that it applies only to the trading of proxy rights for authenticators between the agent and authentication service. Once the agent has acquired the authenticator by which it may impersonate the initiator with a particular recipient, it may continue to do so within the timeframe established by the authentication service in the authenticator. This may, in fact, extend beyond the window commissioned by the initiator. Alternatively, many such authenticator acquistions may be required within the timeframe necessary to complete the operation depending upon such factors as operation scheduling and the relative magnitude of the authenticator and proxy timestamps.

Due in large part to the reliance upon impersonation as the means of proxy implementation, the delegation of initiator authority is nearly completely operation-wise unconstrained to the agent within the specified timeframe. Specifically, there is no delegation of identified task responsibility. In fact, this attribute of the solution is by design in order to accomodate the ability of the agent to assume more responsibility than originally recognized but implicitly authorized. One instance in which this is the case with respect to the document printing example cited earlier exists when unidentified document retrievals are necessary to resolve subitems referenced from the main document. An undesireable side-effect of this strategy is to delegate immensely more authority to the agent than is required to fulfill the initiator's intent. Such a blank-check is also obviously counter to prudent protection practices in which privileges are dispensed consistent with need-to-know in the discretionary instance and clearance in mandatory schemes.

A third dimension exists against which the scoping of authority delegation under the current proxy mechanism could be improved. It seems the delegation of proxy authority is itself unconstrained. That is, once the initiator appoints an agent, that agent may elect to further expand the set of entities representing the initiator by naming any number of additional agents therefor. Any subsequent agent may do likewise. This authority distribution is constrained only by the timeframe specified by the initiator in the original proxy by which the first agent relationship was formed. Each such subdelegation of agent rights is not even acknowledged to the initiator, much less confirmed. While the authentication service participates in each subagent appointment by virtue of its creation of the proxy by which they occur, it lacks any useful criteria against which such requests could be meaningfully screened. Given such ease of proxy reproduction, a single misjudgement as to the credibility of a prospective agent could be disastrous to the initiator as

well as the recipient. In this aspect of proxy use, the initiator's fate is like that of the recipient in the general sense - neither has adequate control.

3.1.2.3 Threat Immunity

It has been noted a number of times that an effective authentication mechanism must itself be highly-immune to the protection threats present in its environment. Such a property enhances the identity assurance which the authentication mechanism provides the entities which employ it. So also does it reinforce the foundation upon which the complementary components of the protection architecture are constructed.

As the proxy model is built upon the XNS authentication base, it is fundamentally vulnerable to the attacks described in section 3.1.1, namely authenticator replay and conversation key compromise. Among the proxy model architects' prime challenges was to compensate for these underlying shortfalls such that their mechanism would itself be effective nonetheless. This goal was attained for the most part. However, two exceptions to this statement have been identified by the student as improvement opportunities. In both instances, the result is the acquisition and execution of a proxy by an entity for whom the associated delegation of rights was not properly authorized.

Figure 3.1.2.3.1 depicts a scenario by which any entity having physical access to the standard proxy exchange sequence may grant itself the rights being delegated. Notice that messages one through five of this diagram are identical to those of the proxy model, while the rest deal specifically with the intrusion. More accurately stated, the key requirement to successfully compromise the proxy model in this manner is the ability to record and replay the message by which the intended agent first presents its proxy to the authentication service for an authenticator against any recipient. Message two performs this function. Messages three through five and recipient C have been represented in the diagram in a light font to illustrate that they have no bearing on the attack. In fact, agent B will likely complete its functions as planned oblivious to the intruder's efforts.

This attack is focused on message six, which combines the threats of replay and message stream modification. It takes advantage of the observation that the authentication service returns a secondary proxy naming the identified recipient as a subagent whenever a current agent issues a valid request for an authenticator. Thus, intruder D need simply record an effective proxy-for-authenticator trade request for the entity it seeks to impersonate. It thereafter must only substitute its name in the cleartext recipient field. The composite is then issued as message six.

Interestingly, the authentication service relies on the fact that the authenticator it returns in message seven is usable only by the party which is a legitimate agent per the proxy and authenticator of message six. This primary element of the message seven response is thus of no value to the intruder. Neither is it the motivation for the attack. Notice that the proxy attached to message seven is exactly what the intruder sought, a usable ticket to impersonate the initiator. Messages eight through eleven confirm this via the normal proxy application sequence.

The rogue proxy was attained under this scheme by deceiving the authentication service to believe that the trade request eminated from a currently valid agent. This is possible given the XNS model's fundamental susceptibility to replay consequent to timestamp reliance. Additional important contributing conditions include the modifiability of the recipient field of message two and the indiscriminate usability of the proxy returned in message three.

The proxy interception outlined above could be perpetrated by nearly any entity in an XNS network given the underlying broadcast local area subnets. It does require that a valid agent actually issue a successful proxy-for-authenticator trade request, however. The second means of proxy interception is more restrictive in that it may be attempted only by a much smaller audience: an agent may intercept proxies subsequently directed at the initiator on whose behalf it is commissioned. However, the proxy need not be applied by the intended agent to enable the intrusion in this instance. Figure 3.1.2.3.2 illustrates the substantive exchanges comprising this scenario. Note in this case that the familiar messages by which proxies are routinely applied are omitted in the interest of legibility. Accordingly, only those essential to the compromise under discussion have been included.

Message one commissions B as A's agent. B thereafter engages the authentication service in a rather unique proxy-for-authenticator trade request. This special case represents B's expression of its intention to serve as A's agent in subsequent dealings with the authentication service itself. One might consider this the ultimate violation of the demand for identity assurance, yet it is not explicitly precluded to the student's knowledge. Presumably, that effect would be implicitly achieved as a consequence of consistent authentication service reliance upon the private key of its intended conversant. The following intrusion demonstrates that such is not the case in at least one instance.

B acquires the authenticator by which it may impersonate A in conversations with the authentication service via messages two and three. Message four represents D's intent to commission A as its agent.

The proxy element of this transaction is thereafter traded for an authenticator by which B may actually serve as D's agent via the normal sequence of messages five and six. A key to intrusion success in this respect is that B names itself as the recipient in this transaction. As in the last scenario, the authenticator component of the authentication service's response is neither usable or of interest to B - but the proxy is unfortunately right on target. B is then ideally positioned to illegitimately assume D's role.

Unlike the former intrusion scenario, timestamp shortfalls of the base model are of no meaningful consequence in this instance. Rather, allowance of entity impersonation with even the authentication service combines with insufficient privacy of the cascaded proxy of message six to form the primary loophole enablements. Indiscriminate usability of the proxy as communicated from initiator to intended agent is also a significant element of the attack.

3.1.2.4 Extensibility

The final class of improvement opportunities identified by the student differ in an important sense from those described thus far. To date, the items listed represent means of addressing significant existing vulnerabilities in the XNS protection mechanism. Those which follow pertain to ways in which the future expansion of the mechanism's capability are constrained as a result of the present proxy architecture. Current shortfalls appropriately warrant short-term resolution. The student contends that relief from the extensibility constraints identified herein is likewise critical given the rapid rate of technological advance and the fundamental motivator thereof - customer demand.

In this respect, the most important area limited by the proxy architecture involves the basic protection concern of accountability. The effectiveness of such an ability to associate entities to the transactions in which they participate is largely dependent upon both the accurate establishment of entity identity and the reliable collection and analysis of audit trails of interest.[7] Such a function has significant value with respect to attack deterrence, detection, and recovery.

XNS' lack of an organized, distributed audit mechanism is therefore a major functionality shortfall whose solution is beyond the scope of this work. This condition is relevant to this thesis only in the sense that it accounts for the fact that there has yet to be a major objection cited in association with the proxy mechanism's complete compromise of the identity assurance upon which accountability is based. Without such a logging ability, it matters little that the accuracy of the information which could be captured is highly-suspect. Looking downstream

somewhat, it is quite likely that the auditing side of accountability will be addressed. At that time, the rampant impersonation condoned by the proxy mechanism will be objectionable in this sense as well as those cited in section 3.1.2.1.

A second dimension along which the proxy model is limiting is that of access control models. XNS presently relies solely upon discretionary controls. The authority passing accomplished via the proxy mechanism reasonably approximates such a need-to-know policy, in that the initiator elects to temporarily distribute its rights to the agent based upon some local criteria. Objections to this integration of the authentication and access control mechanisms have already been cited despite this apparent philosophical consistency.

The case against impersonation as the means of authority transfer strengthens immensely given extension of the access control policy to encompass mandatory rules as well. Coincidentally, this is the expected trend of commercial system evolution within the next few years. Accurate identity establishment is important in such an instance as the means of clearance determination. This, in turn, is the basis of mediation decisions rather than simple identity. More importantly, the concept of authority transfer under such a policy does not accomodate initiator discretion as to agent preference.

3.2 Architectural Enhancements

Preceding sections of this document establish the need for specific revisions to the model by which authority is temporarily transferred between entities within the Xerox Network System. The extent of the discussion by which this end has been achieved is indicative of the magnitude of the challenge presented by the task of completely addressing the requirement and architectural adjustment tasks associated with this topic. The final research products of the student's pursuit of that formidable goal are presented in this section.

Three main philosophical principles guided the production of the specific architectural enhancements set forth below. These serve both to partially establish architectural content validity and are evident in the presentation style. Each also shares the notable trait of being conscious attempts to deal with the complexity of the problem at hand in the disciplined, effective fashion one would justifiably expect in a work of graduate thesis scale.

The first of these is recognition of the fallacy of the single-solution belief. This is clearly a necessary element of a researcher's mindset, having been cited as one of the three great obstacles to innovation.[57] Those lacking such an orientation pursue the identification of the one correct evident solution which exists for each problem. Such blindness prevents the sufficient investigation of potentially viable alternative approaches which is a widely recognized characteristic of effective software development activities, particularly that of design.[22] Application of this principle to the work at hand is a motivation for the forthcoming specification of three architectural models of varying intent rather than the single option one might expect prior to consideration of this issue.

The student's second guiding principle builds upon the first in that among the many potential problem solutions considered one need incorporate those of fundamentally different paradigms on the chance that extension of the existing assumptions is insufficient to attain the target growth increment.[57,58] Realization that models are incomplete system abstractions precedes this conclusion that the underlying assumptions are themselves limiting factors of some magnitude to solution effectiveness. This understanding manifests itself in the case at hand in reconsideration of agency viability within a secure system, its architectural coupling with the authentication mechanism, and reliance upon XNS' current base authentication model.

The familiar divide-and-conquer strategy by which complex tasks are decomposed into solvable units is the remaining principle pertinent to the following architectural development. Interestingly, it also happens to

have been the first of the three applied in that it is the motivation for the enhancement opportunity analysis by which their respective interactions and priorities were characterized as a prerequisite to model development.

The consequences of the student's application of both these basic general principles and those of lesser note to the particular XNS authentication architecture enhancement task represents the significant remaining concern of this document. Such is the role of the discussion comprising the rest of this section.

3.2.1 Opportunity Analysis

In order to respond effectively to a set of identified architectural inadequacies, such as those forming the XNS authentication model enhancement opportunities of section 3.1, one need understand a number of its attributes. Analysis of the set elements is required to establish such familiarity. Consistent with the divide-and-conquer principle outlined above, two of the most significant such relationships of interest to the architect are the functional dependencies between elements and their relative urgencies. Each perspective facilitates efficient, visible ordering of pending architectural accomodations.

These particular attributes of the desired revisions thus represent the next logical topic of this disclosure.

3.2.1.1 Interdependence

To be effective, task scheduling must accomodate the functional interdependence relationships among the elements of a set of grouped tasks, regardless of application. The purpose of this planning function is to establish cause/effect relationships among the elements so as to enable management of the removal of all causes for those of greatest urgency. In fact, such information is itself a valuable modifier within many nontrivial prioritization schemes. For simple or familiar activities, functional interdependence is often either nonapplicable or obvious. In other circumstances, conscious effort need be expended to establish such a hierarchy. The latter case applies with respect to the enhancement opportunities addressed by this thesis.

Figure 3.2.1.1 illustrates the dependence hierarchy within this set. The message gleaned from this perspective of the opportunities is that the four elements listed in the lowest layer of the diagram are fundamental causals of those above. Note also that these elements are independent of one another, being consequences of other circumstances described earlier within the scope of their individual treatments. Likewise, the opportunity listed in the middle layer is caused in part by all of those

listed below as well as others. Like those in the lowest layer, it partially contributes to the existence of the opportunity listed in the top layer. Closer examination of each opportunity illustrates that it is not generally necessary to address all of the causal opportunities within lower layers in order to remedy that of interest. Rather, various revision combinations are sufficient. This point is illustrated by way of the alternate architecture instances specified in section 3.2.3.

3.2.1.2 Prioritization

The respective benefit magnitude associated with each element of the enhancement opportunity set is a primary consideration in both architectural strategy development and evaluation of consequent model sufficiency. This is based largely on the system consequences of the identified existing condition. As noted in the preceding section, the functional interdependence between enhancement opportunities is a special case of this measure. From a priority perspective, those opportunities on which there is heaviest dependence are weighted most important. This is somewhat an inverse function on the hierarchy of section 3.2.1.1, with those elements in the lowest layer considered equally most urgent.

Figure 3.2.1.2 presents the benefit ranking among the enhancement opportunities after all factors have been considered. Interestingly, it turns out to be a reverse ranking of the functional dependence relationships, with ties among those elements of the lowest layer arbitrated. Notably, the opportunity to remedy identity assurance shortfalls given the current proxy model as described in section 3.1.2.1 ranks far and away as the most urgent. Authorization scoping under the proxy scheme rates a distant second, while positions three through six are of only minor difference relative to urgency, yet once again much less critical than correction of the authorization scoping concern. Within this cluster, conversation liveness and identity assurance under the base model are the most beneficial given their frequency and severity. Liveness assurance rates slightly ahead of that of identity based on the first of these factors. Threat immunity under the proxy model follows given its dependence on these shortfalls, as well as its lesser likelihood as a consequence of being limited to proxy use situations. Lastly, extensibility is considered least urgent despite the importance of the accountability and mandatory access control facilities it concerns. This is due in large measure to the associated need to expand those general mechanisms independent of agency model resolution. An effective resolution to the many proxy issues cited is a necessary but insufficient condition relative to provision of those security facilities.

3.2.2 The Agent Debate

The priorities established above indicate that the potential return on architectural revision investment is very high simply given correction of the noted inadequacies associated with the proxy mechanism's identity assurance and authorization scoping dimensions. However, the student would be justifiably subject to a criticism of work incompleteness were he to passively accept the premise that the underlying agent analogy is the optimal means of manifesting authority transfer under these particular circumstances. Such fundamental paradigm challenge is consistent with the student's previously-stated thesis development principles, general cautions regarding the potential inapplicability of such extensions of human experience to computing technology, and the ongoing debate within the security community precisely regarding the agent application.[54,55,56,58]

Elimination of the agent model from the XNS architecture would certainly address four of the six authentication mechanism enhancement opportunities, namely all those pertaining to proxies. Only those comparatively minor items relevant to the base model would then remain. This section considers the feasibility of such a potentially-elegant approach.

Of course, successful completion requires identification of at least one alternate model satisfying those applications who are clients of the authority-passing enabled through agency. As was the case in explaining the motivation for agent inclusion in the XNS architecture, the file service-resident document printing model presented earlier in section 2.4.3.2.1 serves as the instance of such applications referenced for this purpose. Accomodation of those printing needs is necessary but insufficient to prove general viability, while a demonstration of deficiency would support a strategy of agent model revision as opposed to replacement.

3.2.2.1 The Purist Perspective

The legitimacy of agents under protection models is emphatically dismissed by those who perceive them as compromising existing mechanisms.[54,55] Loss of individual accountability, believed a necessary deterence and detection mechanism, is the particular issue most frequently raised. The higher-level consideration of individual responsibility is thereby compromised given an inability to associate subjects to their mediated actions against objects. Coincidentally, this is precisely the identity assurance issue instantiated by the proxy mechanism under study. The present challenge is to discern whether such

an outcome is implicit in the agent model or simply consequent to its form in the XNS authentication mechanism.

A frequently-discussed specific instance of the agent concept is that of password-sharing.[54,55,56] Many protection professionals find it objectionable from the perspectives mentioned above that password-based identification mechanisms are insensitive to the infrequent sharing of the information by which they discriminate parties. Such an exchange between individuals outside system security controls amounts to the discretionary creation of an agency relationship. Notably, this condition is easily revocable. In addition, administrative security policies may be instituted under which the password owner retains responsibility for all actions performed under the associated identity as a deterrent to irresponsible such disclosures given loss of individual accountability. Such considerations remain objectionable under circumstances where mandatory access control policies are warranted by the application's security sensitivity. Nontransferable identification schemes are absolutely called for under these conditions.

Nonetheless, theoretical computer security purists are unsatisfied short of individual accountability within the system even under discretionary access control policies. Their perspective is that sufficient leverage to avoid the need for agency given existing access control paradigms is available between organization of the subject/object spaces and architectural mechanisms. In particular, the system should be organized such that the initiator either represent itself with the recipients of its transactions or delegate such rights to other entities via the discretionary access control mechanism. In the instance of XNS' file service-resident document printing application, this implies that the workstation either directly participate in the document transfer from file service to print service or that it establish read access rights against the files comprising the document to enable some other entity to do so.

3.2.2.2 The Pragmatic Perspective

Unfortunately, a number of attributes of the existing Xerox Network System prohibit either of these architectural options from being realistically usable alternatives to the current reliance upon the print service as its client's agent in the file service-resident document printing application. The specific nature of each such property is individually presented in this section. The collective general theme illustrated is that actual instances exist in which an agent is better equipped to perform certain actions than is an authorized party given that it is held accountable while doing so. Responsibility is thus focused externally on the authorized party, while shared between the two on a secondary

level. Among others, this principle routinely motivates the hierarchical decomposition of human organizations.

3.2.2.2.1 Functional Efficiency

A number of opportunities for enhanced system efficiency were noted in the introduction of the file service-resident document printing model. Specifically, section 2.4.3.2.1 enumerates these as resource scheduling optimization at the print service, workstation productivity, and modular print document construction. None of these opportunities are achieveable given an architecture shift resulting in the workstation's first-hand participation in all underlying file transfer operations.

Consistent with XNS' client/service transaction model, all direct interactions between the workstation and print service must occur within the window required to register the print request. However, this represents a severe constraint to the resource balancing algorithms at the print service when the document transfer is included in the exchange set. In fact, the ability to indefinitely postpone the print service's such data acquisition is among the main attractions of document migration to a file service in the first place.

This observation does not apply to demand printing applications, however. In such an instance, the document is file service-resident largely as a means of long-term on-line retention. Short-term workstation resource usage is optimized under this scenario. The unique negative dimension of directly involving the workstation in document transfers to the print service given this scenario pertains to unnecessary communications bandwidth consumption given the redundancy of the operation. Of course, workstation processing capacity is also needlessly expended under such an approach.

As noted earlier, printable documents are represented in the Interpress page description language within the XNS architecture.[48] In the simple case, an entire such document is represented by a single file. This constraint is relaxed in the general case, however. In fact, the printable form may span many files across any number of file services. The language supports this capability via indirect reference primitives somewhat analogous to the "include" construct of the "C" programming language.[59] The ability to nest such externally referenced content many levels further adds both to construction flexibility and its corresponding complexity. Production publishing is the primary customer application of this organizational capability.

The two functional inefficiencies described thus far are amplified under instances of such a nested organization. Subsequent sections elaborate

upon significant attributes unique to this complex case. Note that this condition may be encountered in association with either document printing model given the inclusion of indirect file references within the top-level printable document involved. The potential range of agent paradigm applicability is thus expanded proportionally.

3.2.2.2.2 Delegation of Intent

This application serves as a vivid instance in which the party engaging the services of an agent seeks to delegate the global intent of the objective to be achieved rather than its specific component tasks. The logical motivation driving such an open-ended approach is that the transaction initiator is not knowledgeable as to the complete composition of the implied task list, whereas the agent is. Such a situation requires the initiator to rely upon the agent's inherent trustworthiness under the particular circumstances. A degree of assurance in this respect is typically attained via a dual-pronged risk management strategy, with compromise prevention based primarily upon examination of the prospective agents' public record complemented by an auditing-based detection mechanism.

This is precisely the scenario as regards the nested printable document example introduced above. By design, the workstation-resident print client is unaware of the complete file set comprising the document it requests a print service to image on its behalf. System flexibility is enhanced by limiting such scope management to include only direct references. Consequently, the service must dynamically parse incoming documents and their component segments to establish this overall context. This profile is thereafter employed to acquire and merge the pertinent content prior to imaging. Such rolewise responsibility division is consistent with the familiar performance tradeoffs associated with the late binding principles of programming language theory.[60]

One possible architectural alternative which is consistent with these roles calls for the print service to notify the client as to the file set it requires to continue throughout the course of such parsing. The workstation could then participate directly in the necessary transfers between various file services and the print service conditional upon the reservations expressed in the prior section. A crucial property of this approach is that the scope of such data transfers is established by the print service. In addition, the workstation has little means of measuring the completeness of the file list beyond trust in the service and analysis of the requests over time. This candidate model benefits as a consequence of the minimal simplification associated with its elimination of the need for any formal authority transfer between the workstation and print service. On the other hand, it is essentially equivalent to an agent situation in terms of threat

susceptibility given the initiator's lack of specific familiarity with the rights set it possesses and need leverage to complete the intended objective.

3.2.2.2.3 Discretionary Access Control Model Fidelity

One might alternatively propose that the workstation-based print client be equipped with the capability to traverse the nested printable document structure, thereby invalidating the model above in which it is less informed as to the specific task set than the agent. This could be accomplished in any number of ways, all of which sacrifice the flexibility achieved through the execution time binding employed at present. This tradeoff aside, the workstation could then confidently avail the print service to the precise file set involved.

Various potential models exist by which this end could be achieved, one of which is the familiar first-hand involvement of the workstation in the data transfers. This approach dovetails nicely with one of the means by which the workstation could apprise itself of the document structure, dynamic parsing of the individual elements much as the print service presently operates. A notable associated advantage is its transparency relative to the existing document creation and description architecture. While authority transfer is unnecessary in this scenario, its unattractive properties relative to functional efficiency are quite significant as outlined above. Those reservations are supplemented in this instance by the many negative effects incurred through document parsing at the print client.

Another available approach would be for the print client to temporarily expand the read access which it possesses against the individual elements of the file set to include the print service. This seems a viable strategy at first-blush given its appropriate use of the existing discretionary access control mechanism to implement the desired authority transfer. Unfortunately, existing need-to-know models such as that employed by the XNS file service reserve such authority expansion for the object's owner. Thus, while the print client has read access to each file comprising the nested printable document by definition, it does not necessarily have the ability to transfer those rights to the print service. Paradigm modification to enable elements of the various access classes to unilaterally include other subjects in this fashion would be both significant and undesireable, invalidating the fundamental owner concept by which responsibility is presently focused.

A slight variation on this theme is also worth discussion. Given such read access and complete knowledge of the file set, the print client can create redundant copies of those objects for which it is not the owner. Read

access for each data set could then be established on behalf of the print service. Such data duplication is a significant expense. Incrementally, model complexity could be reduced at an associated similar resource cost by copying all files in the set as opposed to only those for which it was necessary to satisfy the discretionary access control model. An additional nontrivial issue pertaining to the need to rescind print service access upon completion of its specific short-term service also need be addressed. This is much easier in the instance where all files are copied, as the print service could also be granted delete access in anticipation of its responsibility to exercise such rights upon print completion. Otherwise, a synchronization mechanism need be instituted by which the workstation would be notified of the event. Construction of an effective mechanism of this type is no simple task given the intermittent client availability assumption of the underlying transaction model. All things considered, this option seems no more satisfactory than any of the other frail identified paradigm substitutes for agency.

3.2.3 Alternative Agent Models

The preceding exploration firmly establishes the insufficiency of existing access control paradigms to accomodate those conditions under which the XNS Authentication-by-Proxy architects expected their agent mechanism to be most applicable. The effort expended in considering such a fundamental issue is not without merit in spite of this outcome, however. Affirmation of the value added by inclusion of an effective agent mechanism within the architecture is itself a prominent conclusion of this work.

That being the case, the candidate solution set has been narrowed to encompass only those models by which the XNS architecture could be revised to support agency while responding to the identified enhancement opportunities. This section describes the student's contributions to that set. Where useful, the file service-resident document printing application continues to serve as the instance by which viability is demonstrated in a specific, limited sense. General utility is also promoted given the student's inability to identify cases proving the contrary.

Three such models are presented, consistent with the acknowledgement that none is defensible as best under all circumstances. Particularly relevant tradeoffs pertain to the completeness of enhancement opportunity satisfaction against the tolerable degree of architectural modification. These respectively represent the most relevant associated aspects of benefit and cost to be considered in this case by those responsible for the long-term evolution of the overall XNS architecture.

In terms of content, the models themselves vary as a consequence of the paradigm shift principle. In this second-level instance, coupling of agency to the authentication mechanism is reconsidered as is reliance upon XNS' current base authentication model. The result is a pair of models under which agency is provided in conjunction with authentication as in the current model. Incrementally, a third direction is specified by which access control serves as the authority transfer vehicle for agent relationships. The evident cost of such architectural adjustment is balanced in each presentation with that of the model's success in capturing the potential enhancement opportunities.

3.2.3.1 Authentication-Coupled

As demonstrated by the current facility, it is possible to minimize the amount of architectural revision required to provide an agent mechanism given its coupling to that of authentication. Section 3.1.2 enumerates many attractive aspects of such localization. In addition, the momentum to incorporate such modifications into the XNS authentication scheme also seems to exist at the present time as evidenced by the recent promotion of adjustments to facilitate the associated need for secure post-authentication conversation traffic.[52] Finally, it is possible to remedy most of the identified shortfalls of the present agent facility within the scope of such an authentication-based mechanism.

The primary exception to this statement applies to the ability of the initiator to effectively constrain the scope of its agents' actions consistent with the intended authority delegation. As the opportunity analysis illustrated, such proxy authorization scoping ranks as the second most beneficial among the six. This significant reason combines with those associated with the principles of consistency in functional decomposition to form the main inadequacies associated with such a pairing.

In light of these common characteristics, a pair of models are presented below. The first represents the minimal change to the present state given its reliance upon the existing XNS base authentication model. This constraint is dropped in forming its associate in order to capture a larger share of the enhancement opportunity set.

3.2.3.1.1 Preserved Base Model

Perhaps surprisingly, a significant percentage of the potential benefit associated with the identified enhancement opportunities may be captured simply through relatively minor adjustment of the present XNS Authentication-by-Proxy model. This is largely a consequence of the model's amenability to correction of the major opportunity - identity

assurance under the proxy model. The primary constraint implied throughout this discussion is that base authentication model modification is prohibited other than as specifically required to support the agent mechanism. As stated previously, this is a potentially valid practical consideration.

Consequently, the existing base model's character is preserved in the model of Figure 3.2.3.1.1.1. As demonstrated, no adjustments have been incorporated relative to the two enhancement opportunities specifically associated with the base model. This is not a particularly damaging constraint relative to the goal of specifying a vastly improved model, however, as the benefit associated with each was quantified through earlier analysis as relatively low. On the other hand, it does present a challenge with respect to the task of capturing the other opportunities influenced thereby. The discussion to follow illuminates the specific manner by which the completeness of proxy extensibility and threat immunity opportunity enhancements are correspondingly reduced.

The only required revision to the preserved base model is the addition of a fourth value to the authenticator This is the "agent for" field, positioned following the initiator's name. Semantically, the field is used to designate the name of the principal on whose behalf the initiator is serving as an agent. Accurate communication of such information to the recipient rectifies the identity assurance problem under the current scheme. Under the new model of Figure 3.2.3.1.1.2, the authentication service employs the authenticator to notify the recipient as to both the identity of the party with which it is directly interacting and that on whose authority it seeks to perform. The service provides assurance of information validity given the identity which the agent conveys through message two. Given such relationship visibility, the recipient is once again in control of its local access mediation decisions. This applies also to the mandatory access control extensibility opportunity. In the base case, where no agent relationship has been established, the field is null as designated by the "Λ" symbol. This is the familiar situation in which the initiator's own authority is employed in access control decisions.

Accountability is also facilitated given such an adjustment. Note also the addition of an audit trail of proxy trade requests to be maintained by the authentication service. This supports the specific accountability of agents to the initiator in two respects. In the first, authority scoping is facilitated at a minimal level in that the initiator now has a mechanism by which it may determine the set of recipients with whom its agents have transacted business. This is a very-high level authority abuse deterrence and detection mechanism in which the authentication service acts also as a notary. Authority scoping is also enhanced by this facility with respect

to an agent's transitive passing of proxy rights. As above, abuse of such a powerful responsibility is less likely given the greater visibility accrued from this reliable auditing capability.

As a final adjustment to the model relative to authorization scoping, it is recommended that the authentication service constrain the authenticator expiration time to be the minimum of its typical window length and that of the initiator's designated proxy's expiration. Though not visible in the information communicated between parties, this treatment enforces the initiator's designated intent to the degree possible.

Two items are specified by which immunity from the identified threats to which the current proxy model is suspect may be largely achieved. The first, encryption of the recipient's name under the conversation key in message three, prevents an intruder from acquiring a proxy as a consequence of a message modification and replay attack. This is a relatively inexpensive operation given the constrained nature of the object to be translated. Notably, this does not prevent an intruder from perpetrating successful denial of service attacks by replaying valid messages within the replay window to which the base model remains suspect. However, the audit trail facility specified above captures evidence of such occurrences for subsequent analysis and detection.

The second threat to which the current proxy model is suspect is preventable as a side-effect of the identity assurance adjustment described earlier. Specifically, the authentication service is now in possession of sufficient information to insure that it does not translate a proxy for any party but that explicitly designated, including any agents thereof. In addition, one might reasonably argue that the authentication service reject attempts by any party to establish an agent relationship for transactions against the authentication service itself. This is unnecessary relative to any of the identified enhancement opportunities, but presented as a practical consideration given the implicitly sensitive nature of such interactions.

The shaded areas of Figure 3.2.3.1.1.3 graphically indicate those opportunities captured by these revisions to the current Authentication-by-Proxy model. Included among these is that ranking highest in priority as well as a portion of that pertaining to the second most useful. This seems a very high return on the minimal investment described. Thus, this model is the most attractive of the three offered by the student if overall architectural stability is of high concern relative to security assurance completeness. Such is potentially the case if maintenance cost minimization is an engineering goal.

3.2.3.1.2 Enhanced Base Model

A second, slightly more radical, model adjustment option expands the scope of consideration to include the underlying base authentication mechanism. While one could elect to patch its current shortfalls, the evolution of more effective models since that of XNS was conceived suggests that the prudent strategy is to select the most attractive of among those which have emerged. Thus, the nonce-protected conversation key model of section 2.3.2.5 is the student's choice.

The key property of that model favoring its selection under these conditions is excellent effectiveness in establishing conversation liveness and identity assurance, two fundamental weaknesses of the current base XNS mechanism. These are respectively consequent to reliance upon timestamps and the conversation key within the authentication exchange. Alternatively, the candidate employs nonces and the recipient's private key - methods defended earlier as more effective.

Expectedly, such a benefit increment is not free of cost. Among the most notable such expense is greater transaction traffic between the principals and authentication service. This is due both to the elimination of conversation key caching and the handshaking associated with nonce use but not timestamps. Denial of service attacks given authentication service unavailability also proportionally increase in likelihood. In addition, such modification of fundamental system facilities implies nontrivial maintenance cost distributed across many of its elements as well as potential configuration compatibility issues in operating environments.

The figures on the opposite page illustrate the proposed enhanced base authentication and agent models. As in the former proposal, the sole modification of the base model required specifically due to the agent application is inclusion of an "agent for" field within the authenticator. This again designates the principal on whose behalf the transaction initiator seeks to operate with the recipient. In the base case, this field remains unused, consistently represented by the "Λ" symbol.

This agent model is otherwise very similar to that presented in the prior section, including the addition of proxy trade transaction auditing responsibility to those of the authentication service. A significant difference pertains to the lack of the need to encrypt the recipient field of this message, number four of the agent model. This is due to the inherent liveness assurance of the message under this base model. For the same reason, the denial of service attack to which the first model is suspect may be prevented in this instance. This is an important

perfomance tradeoff against the increased message traffic identified above.

A slight disadvantage of this model against that of the existing base XNS model pertains to authority scoping under the agent model. Replacement of the timestamp in the authenticator with a nonce, though attractive with respect to liveness assurance, also shields the recipient from the window within which the proxy is valid. Thus, while the authentication service insures that the proxy is valid at trade request time, no effective means exists thereafter by which the recipient may terminate the conversation consistent with initiator's stated intent.

Figure 3.2.3.1.2.3 demonstrates the security assurance benefit potential associated with agency under the enhanced base model architecture. Consistent with its higher expense, the model yields a marked functional improvement over the preserved base model option. It is important to note that the two are equivalent with respect to identity assurance in the agency architecture, the highest priority opportunity. Likewise, each is constrained with respect to extensibility given its coupling with the authentication mechanism. In this instance, improvement potential exists with respect to the frequent authentication exchanges which occur at the base level. This direction is quite attractive in light of such observations, given availability of the necessary moderate maintenance resources.

3.2.3.2 Access Control-Coupled

Has the best been saved for last? Once again, the answer to this question depends upon the reader's definition of quality. An affirmative reply is warranted if security assurance and system extensibility rank very high on the customer's fitness for use criteria list. Otherwise, either of the two solutions described thus far will prove quite satisfactory. In this section, the student's identification of viable enhancements to the XNS Authentication-by-Proxy model is completed as a consequence of yet another application of the paradigm shift principle. In this case, the authentication mechanism is discarded as the underlying agent architecture vehicle. Rather, protection's traditional authority passing mechanism, access control, is supplemented with agent-specific provisions. Such a strategy enables the parallel independent evolution of authentication and access control mechanisms by which efficiency of the integrated protection facility may be optimized. Maintenance cost and configuration compatibility once again represent the cost elements of the decision process. This direction represents both the most expensive and beneficial of the three.

The student's model by which an agent mechanism might best be offered in conjunction with the access control mechanism is illustrated in Figure 3.2.3.2.1. A number of attributes of the model warrant discussion prior to consideration of the particular exchanges by which it is executed. Note first the introduction of an "agent administrator" entity within the architecture. As the name implies, its role is to provide a centralized source of administrative expertise regarding those transactions in which agents participate. Specific tasks of which this objective is comprised include the retention of context regarding the security clearance of prospective agents by which transaction initiators can be accurately apprised of their statically-established trustworthiness. The agent administrator also verifies the legitimacy of a proxy relative to the service requested of the transaction recipient by a purported agent. Lastly, the administrator serves the notary role seen before with respect to proxy trade request auditing.

A second noteworthy property of the model is the use of standard authenticators in association with application-dependent message content. The specific authentication model employed is an orthogonal issue to that of the agent architecture. It simply performs the pure authentication task of principal identification. Given the interplay of the pair with respect to the overall protection scheme, the nonce-protected conversation key model of section 2.3.2.5 is recommended for the same reasons it was employed as the basis of the enhanced base model above. This is not a necessary condition for model adoption, however. Alternatives modify integrity of the messages comprising the exchange proportional to their respective effectiveness.

The first order of business under the access control-coupled agent model is for the initiator to convince itself that the prospective agent is credible. This discretionary decision is supported by preregistering agent clearance with the agent administrator. The initiator obtains this certification information at the outset of the agent sequence via messages one and two. This is compared against local requirements to determine the acceptability of responsibility abuse risk. As the initiator remains responsible for the actions of its agents under each of these models, it is very much in their interest to pass such authority with care. That is particularly so in the instances at hand, in which intent is delegated rather than specific access-controllable tasks. This exchange effectively supports that need.

Figure 3.2.3.2.2 presents the component fields of these messages and all others within the model. In this instance, the exchange is much like the standard initial exchange between initiator and authentication service.

Note that the reply is encrypted under the private key of the initiator to guarantee both privacy and source integrity, with timeliness proven by inclusion of a nonce. Finally, the agent's name within the reply enables detection of message stream modification attacks.

Upon selection of a sufficiently-trustworthy agent, the initiator engages that party in carrying out the objective of interest via message three. Note that the proxy passed is both unforgeable by and of little immediate use to the agent given its protection under the initiator's private key. Fields within the proxy include an identifier by which actions sanctioned under it are to be logged for accountability purposes. The agent authorized under the proxy is also designated along with the time at which the initiator's rights delegation is to be rescinded much as in the existing model. The primary advantage in this instance is the increased granularity of such determination, as explained below. Notably, a simple rights set is also included in the proxy. These are intended to be consistent with those administered under the vanilla access control mechanism. The initiator is thereby enabled to constrain the type of operation delegated even though the specific objects against which such rights apply remain unspecified as required.

Message four is forwarded from agent to recipient when it seeks to perform a service contributing to the initiator's objective. The recipient may immediately choose to reject the request on the basis of the proxy. Alternatively, it would ascertain the validity of the proxy presented by forwarding it to the agent administrator along with the context of the requested service. The agent administrator first checks the validity of the claim that the proxy originated at the initiator by decrypting it under the private key of record. It then insures that the named agent corresponds to the party who presented the proxy to the recipient. The currency of the proxy is verified next, based upon a comparision of the agent administrator's clock to the expiration time specified. Finally, consistency of the operation being requested by the agent with those authorized under the proxy is confirmed. The results are confidentially forwarded to the recipient, along with a nonce demonstrating liveness. So also is the transaction logged by the agent administrator for accountability purposes, as represented by the data dictionary's "ProxyUseLogEntry".

Given success above, the recipient mediates the references implied under the service request given its local policies. It thereafter should log audit trail information according to standard, as yet unspecified, practice. Finally, the agent is apprised of the application-dependent results of requested services.

The agent administrator requires access both to the private key of each system entity and mechanisms by which they are applied to cleartext. The administrator itself needs to be trustworthy for this reason, as well as to reliably perform its notarization function. Finally, it must maintain accurate clearance information with respect to entity trustworthiness under agent conditions. Each of these roles are very similar to those of the authentication and clearinghouse services. Thus, the function should minimally be allocated to the same processor as are those elements. Tighter coupling would also be acceptable contingent upon a modular organization. Finalization of this consideration is best disposed of under the guise of a detailed design activity.

It is somewhat an understatement to note that this direction represents a radical departure from the current Authentication-by-Proxy mechanism. Fortunately, it is also a very attractive architecture from many perspectives including the degree to which it captures the target enhancement opportunities. As Figure 3.2.3.2.3 illustrates, it is nearly complete in this respect. The sole exception to that statement is in the continued inability to explicitly prevent abuse of the delegated intent with respect to time and specific objects. As discussed when considering the viability of agents within the architecture, this is an unavoidable consequence of an organization in which the agent is more informed as to the specifics of the task to be performed than is the authorized party on whose behalf it operates. Accordingly, abuse detection mechanisms have been identified as a prudent risk management technique.

Additional advantages of this model accrue to the individual parties involved. For instance, the initiator benefits as a consequence of such architecture attributes as visibility of agent track records, auditing capabilities as authority abuse deterrents, and the ability to constrain the transfer of access rights within the bounds of the objectives to be performed. Similarly, the recipient regains control of access mediation decisions given the visibility of agency relationships. In terms of implementation strategy, need-driven extensibility is facilitated given the architecture of a centralized agent administrator. In addition, decoupling authentication from authority transfer mechanisms regains architectural consistency. Finally, applications within which agency is a meaningful concept may migrate towards its use, while others need not change. Such scope constraint is also desireable from the perspective of verifiability - a fundamental security assurance property.

Along these lines, Figure 3.2.3.2.4 illustrates a typical example of the access control-coupled agent model within the file service-resident document printing application. Most significant of among its many

attributes is the ability of the print client to constrain the authority delegated under the proxy to encompass only those operations requiring read access. As an instance of model sufficiency under the primary conditions motivating provision of agency within the architecture, so also does it focus verification attempts with favorable results.

4 Conclusions

This thesis report describes a number of innovative products of the student's development activity. First among these is the identification of many significant improvement opportunities which exist within the current Authentication-by-Proxy model of the Xerox Network System. This claim is backed by an extensive body of previously-published protection literature. More importantly, a number of architectural enhancements by which these opportunities can be captured are provided in response to the recognized need. Consistent with the acknowledgement that no single solution is defensible as best under circumstances of such technical and administrative complexity, three viable enhancement options are offered. Cost and benefit factors beyond the realm of this work should be employed as the ultimate discriminators within this set. Finally, the demand for a disciplined agent management mechanism within a distributed system such as XNS was resoundingly affirmed in the course of these first-order pursuits. This outcome is itself surprising in that it is counter that expected by the student at the outset of this thesis.

4.1 Unanticipated Problems

Only one notable unpleasant technical surprise was encountered in the course of this work. One might already have surmised on the basis of length that this effort incorporated many more concepts than was projected at the time of topic selection. Fortunately, much of the underlying research was performed in the course of associated graduate studies in anticipation of some thesis pertaining to protection within distributed systems. Nonetheless, such volume amplified the requirement for skilled thesis committee members. On this count, it would have been difficult to fare any better than in this experience. Aside from that, incremental development effort was the familiar primary response to this recognition.

4.2 Residual Opportunities

The most obvious latent activity to be pursued is the transformation of the specified architectural models into operational systems. This is a significant objective given the number of associated complex elements for which detailed design, coding, testing, and installation remains. Were that not the case, the student surely would have pursued those tasks within the umbrella of this work. In fact, many individuals could keep themselves challeged for a fair amount of time in this way. The most-likely practical strategy by which this would occur is within the research and development confines of the Xerox Corporation. This is an outcome both of their vested interest in realizing the potential opportunities

described and unique direct exposure to the resources necessary to do so. Of course, the ideas directed in this context at the Xerox Network System apply in large measure to other distributed system architectures as well. Such generalization opens the possibility of many related research avenues.

In addition, effort remains minimally estimated as of a graduate project scope by which formal methods of protocol validation could be applied to the models produced herein. This is an emerging area of study which has recently attracted the participation of some of the protection field's leading researchers.[53]

Glossary

access control: The prevention of unauthorized use of an object by a subject.

access list: A data structure associated with a protected object which describes the individual access rights of its aspiring subjects.

access rights: The set of object references for which a particular subject possesses sufficient privilege.

accountability: The ability to accurately associate operations to the parties concerned.

active threat: An intrusion resulting in an unauthorized modification of protected system objects.

administrative security: Formalized policies and procedures guiding organization practices.

agent: A neutral subject to whom a transaction initiator temporarily grants its own access rights as a means of facilitating subsequent object references conducted on the initiator's behalf for which the third-party likely lacks sufficient authority of its own.

asymmetric encryption: *see* **public key encryption**

authentication: The corroboration that a peer entity is the one claimed.

authentication service: The network entity employed to corroborate the claimed identities of two mutually suspicious yet transaction-eager principals. A second notable function of this service is to facilitate the reliable distribution of encryption keys to those principals seeking protected connections.

authenticator: Ciphertext generated by the authentication service by which a transaction initiator may demonstrate its identity to a targeted recipient, as well as to privately communicate the associated conversation key selected by the authentication service.

authorization: The granting of rights, which includes the granting of access based on rights.

availability: The likelihood that a system is operational within a random time period.

baseband: A data transmission technique in which the electrical signaling on the medium directly reproduces the digital form of the information.

broadcast: A communications subnet in which all nodes receive each message transmitted thereon as a consequence of reliance upon a shared transmission medium. These are then filtered at each node according to a specified destination address.

caching: A scheme in which a data item is retained in local storage beyond the timeframe of its initial use on the assumption that it will be referenced again shortly thereafter. This is a memory-reference efficiency technique, which has been applied to the authenticator within some authentication schemes.

capability: An unforgeable ticket, which when presented can be taken as uncontestable proof that the presenting subject is authorized to have access to the object named in the ticket.

capability list: A data structure associated with a subject which describes its respective access rights to objects. A series of capabilities.

channel: An information transfer path through the communications subnet.

ciphertext: Data whose semantic content has been temporarily suppressed as the by-product of encryption.

clearinghouse service: The distributed naming service within the XNS architecture. Its role is to provide a mapping between the names of elements in the distributed system and their various properties. Address translation is the most common public service it provides. Its most notable contribution to the XNS authentication model is the

protected retention of the private keys of named system elements.

cleartext: Intelligible data, the semantic content of which is readily-available.

client: The entity within the XNS architecture which initiates a service request. As XNS services implement ISO application-layer protocols, such an entity employs those conventions to issue its requests.

communication: The exchange of information between parties.

communications security: Mechanisms which protect traffic on the communications subnet from intrusion.

communications subnet: Those elements of the network architecture whose devoted mission is to reliably transmit messages from sender to receiver. Layers one through three serve this role in the ISO/OSI protocol reference model.

confidentiality: The property that a protected object is made available to or disclosed only to authorized subjects.

conversation key: The unique, typically authentication service generated, private key to be employed by the initiator and recipient for protected segments of their connection.

correctness: The degree to which the system operates in accordance with its stated requirements.

covert channel: A communication channel that allows a subject to transfer information in a manner that violates the system's security policy.

covert storage channel: A covert channel that involves the direct or indirect writing of a storage location by one subject and the direct or indirect reading of the storage location by another subject.

covert timing channel: A covert channel in which one subject signals information to another by modulating its own use of system resources in such a way that this manipulation affects the real response time observed by the second subject.

credentials: Data which is transferred to establish the identity of an entity.

cryptography: The discipline which embodies principles, means, and methods for the deterministic transformation of data in order to prevent its disclosure and/or to detect its modification.

CSMA/CD: An abbreviation for the channel access arbitration technique introduced by Ethernet. Any node may attempt to transmit on demand under the scheme, with contention avoided by the transmitter first sensing whether the channel is already in use. Propagation delays may induce collisions despite this precaution. These are thus detected and abitrated by random backoffs at the respective senders. Such behaviors motivate the scheme's full name, carrier sense multiple access with collision detection.

DFD: see **data flow diagram**

data flow diagram: A graphical system modeling notation emphasizing the flow of information between components. Another notable attribute of the notation is its strength at partitioning multidimensional systems, thereby decomposing a complex problem into a series of workable ones.

decryption: A translation by which ciphertext is returned to the unique cleartext from which it originated as a consequence of a matching encryption operation.

denial of service: The prevention of authorized access to an object or the delaying of time-critical operations.

digital signature: Data appended to, or a cryptographic transformation of, a data unit that allows its recipient to prove its source and integrity.

discretionary policy: A model by which object access is mediated as a function of the identity of subjects or the groups to which they belong. The controls are discretionary in that a subject with certain access rights is capable of passing that permission on to any other subject.

distributed system: An operating system which executes on multiple independent processors transparent to its users. Such processor-independence is particularly noticeable with respect to the file system, program execution, and protection. Contrast this organization with that of a **networked system**.

document: Information which has been structured to facilitate its effective communication between individuals. Such an object may be represented through various media, electronic or otherwise.

document management: The set of operations which combine to enable the organization of information for interpersonal exchange. Prominent among these are creation, manipulation, and distribution.

efficiency: The degree to which the resources required for execution are affordable to the customer.

encryption: The cryptographic translation of data to produce ciphertext.

end-to-end encryption: The encryption application strategy in which message transformations are performed only at the sending and receiving nodes, transparent to intervening switches.

ethernet: The baseband local area networking scheme introduced by Xerox research, and subsequently formally specified along with Digital Equipment and Intel.

expandability: The degree to which system resources may be conveniently scaled according to the demands of the using organization.

file service: An application service in the XNS architecture which exports conventional filing operations to other entities on the network. Access to the individual files residing in the service by network clients is protected under a discretionary policy.

handshake: A bidirectional exchange between communicating network entities employed to assure one or both parties of such conversation attributes as liveness.

identity-based policy: see **discretionary policy**

IMP: see **message switch**

impersonation: The pretence by an entity to be a different entity.

initiator: The principal who first sought to establish the conversation requiring authentication.

integrity: The property that an object has not been altered or destroyed in an unauthorized manner.

interface: The primitive operations and services which a layer of the network architecture exports to higher layers. Data is physically communicated across such a boundary.

intrusion: An instance in which an object has been accessed in a manner inconsistent with the authorization of the subject.

intruder: A subject who's object references are beyond the scope of its authority.

ISO: The International Organization for Standardization is a worldwide federation of national standard bodies whose many products include the OSI protocol reference model and its security addendum.

key: A sequence of symbols that controls the operations of encryption and decryption. Possesion thereof is thus regulated as a means of protecting the confidentiality of cleartext and/or the integrity of ciphertext.

key distribution center: see **authentication service**

LAN: see **local area network**

link encryption: The encryption application strategy in which messages are individually transformed along each circuit between adjacent switches on the path from sender to receiver.

local area network: A network which is intended to enable a great number of varying processing elements to exchange large amounts of data at high speed over limited distances.

loosely-coupled: An MIMD computing architecture in which the control scheme is distributed. That is, a number of potentially heterogeneous machines combine to form a system with local activities and interaction frequency determined largely by the dynamic needs of the respective elements.

maintainability: The degree to which software remains productive throughout the postrelease phase of its lifecycle.

mandatory policy: A model by which object access is mediated as a function of the sensitivity of its contents and predefined subject clearance.

masquerade: *see* **impersonation**

message switch: The specialized computers comprising the communications subnet which route messages along the individual circuits forming the physical communications media between the sending and receiving nodes.

monitor: A mechanism which synchronizes the access to abstract data types transparently to its clients.

network: *see* **networked system**.

networked system: An interconnected collection of autonomous computers which interact at the explicit direction of their users. Contrast this organization with that of a **distributed system**.

node: An individual computer system which executes processes having a need to interact with those resident on another node through an intervening network.

nonce: A non-repeating number employed in a handshaking sequence to insure the property of reply timeliness to the exchange's originator.

notary: A trusted third-party with whom transaction demographics are recorded to corroborate potential subsequent claims by the principals.

object: A shared-resource against which client references are to be controlled by the protection mechanism.

one-way function: The discipline which embodies principles, means, and methods for the uninvertible deterministic transformation of data .

OSI: The Open Systems Interconnection Reference Model defines a vendor-independent layered network system architecture.

passive threat: An intrusion resulting in an unauthorized disclosure of protected system objects without an associated change in their state.

peers: Processes executing in respective nodes which are interacting according to the protocol of a specific layer of the network architecture.

personnel security: Mechanisms directed at insuring the integrity of the workforce, such as redundant responsibilities and assignment rotation.

physical security: Mechanisms designed to insure the preservation and availability of an organization's physical resources.

principal: The two suspicious parties employing the authentication service to assure themselves of the accuracy of their associate's claimed identity, as well as the timeliness of their exchanges.

print service: An application service in the XNS architecture which exports document printing operations to other entities on the network.

private key: An key used in conjunction with a symmetric encryption algorithm. The system consequence of this attribute is that the single key must be possessed both by the message initiator to form the ciphertext and its recipient to recover the cleartext.

private key encryption: The class of encryption algorithms in which a single key is employed both to encrypt and decrypt messages. This is referred to as a private key.

protection: The mechanisms and techniques which control subject to object references. Often used synonomously with the broader sense of security.

protocol: The rules and conventions by which peers interact.

proxy: A token which demonstrates the approval of an entity to temporarily delegate its authority to a second entity, its agent.

public key: The globally-accessible key used in conjunction with an asymmetric encryption algorithm which is employed by any party wishing to privately pass data to its owner or to confirm the integrity or authenticity of data received from its owner.

public key encryption: The class of encryption algorithms in which two keys are employed, referred to as the public and secret keys. Either may be employed to encrypt cleartext. Having done so, the complementary key must be applied to successfuly decrypt the resultant ciphertext.

recipient: The principal with whom an initiator wishes to establish an authenticated conversation.

reference monitor: An access control concept referring to the abstract machine which mediates all accesses to objects by subjects based upon predefined relations to which it alone has access.

reliability: The degree to which the time between system failures is acceptable to the customer.

replay: Reissuance of messages which were recorded from a prior legitimate conversation.

repudiation: Denial of participation in all or a portion of a conversation.

rule-based policy: *see* **mandatory policy**

secret key: The key used in conjunction with an asymmetric encryption algorithm which is known only by its owner. Consequently, it may be employed to recover

	information passed privately to its owner by other parties or to pass information to others with assurance as to its source.
security:	The mechanisms and techniques which control who may use or modify a computer system or the information stored therein.
semaphore:	An integer variable accessable only via operations which register an acquisition or release request as a means of synchronizing client access to a set of shared resources.
server:	A node in the XNS architecture from which services are exported. Servers are configured with unique resource capabilities or functional roles which are most efficiently availed the overall system via such physical concentration.
service:	An element of the XNS architecture which exports ISO applications-layer protocols.
simple authentication:	A scheme by which a pair of interacting principals identify themselves to one another in a manner which has little immunity against potential protection threats.
strong authentication:	A scheme by which a pair of suspicious principals assure themselves that each is whom they claim to be in a manner which is intended to be highly-immune from protection threats.
subject:	A client of a shared-resource who's use is controlled by the protection mechanism.
symmetric encryption:	*see* **private key encryption**
synchronization:	Coordination of the relative timing of shared-resource use among its many clients.
threat:	A potential means of violating the security policy.
throughput:	The processing bandwidth component of system performance: output per unit time.
time service:	The distributed time synchronization mechanism within the XNS architecture. Its role is to serve as the master

clock against which those local to the many processors in the network are coordinated.

timestamp: A message field in which a time/date value is specified. The associated semantics vary by protocol, though it generally depicts the current time at the initiator's node at the instant of message issuance.

usability: The degree to which the system user finds its operation to be effortless.

verifiability: The degree to which it is possible to certify correctness.

verifier: Ciphertext produced by the recipient in the XNS authentication model which demonstrates response liveness to the initiator.

WAN: *see* **wide area network**

wide area network: A communications network which spans significant geographical distance. A WAN is typically comprised of local clusters interconnected by point-to-point channels.

workstation: An XNS system element whose purpose is to locally service the routine tasks of an individual.

XNS: The network architecture which forms the distributed computing platform of Xerox Corporation's office system product set. This is also the architecture in which the particular authentication paradigm central to the student's thesis is a component.

Bibliography

1. Xerox Corporation, Authentication Protocol, Xerox System Integration Standard, XNSS 098605, May 1986.

2. Wing, P. D., "Authentication: Establishing Interactive Principals," *RIT Topics in Operating Systems Research Report*, August 1987.

3. Wing, P. D., "Secure Open Systems: An Introduction," *RIT Networking-II Research Report*, May 1987.

4. Wing, P. D., "The Kernel Solution to Security," *RIT Operating Systems-II Research Report*, May 1986.

5. ISO/TC 97, "Information processing systems - Open Systems Interconnection Basic Reference Model - Part 2: Security Architecture," ISO 7498-2, February 1989.

6. National Computer Security Center, Trusted Network Interpretation of the Trusted Computer System Evaluation Criteria, Ft. George G. Meade, Maryland, July 1987.

7. National Computer Security Center, Department of Defense Trusted Computer System Evaluation Criteria, Ft. George G. Meade, Maryland, December 1985.

8. Voydock, V. L. and Kent, S. T., "Security mechanisms in high-level network protocols," *ACM Computer Surveys*, June 1983, pp. 135-171.

9. Needham, R. M. and Schroeder, M. D., "Using Encryption for Authentication in Large Networks of Computers," *Communications of the ACM*, December 1978, pp. 993-999.

10. Denning, D. E. and Sacco, G. M., "Timestamps in Key Distribution Protocols," *Communications of the ACM*, August 1981, pp. 533-536.

11. Needham, R. M. and Schroeder, M. D., "Authentication Revisted," *Operating Systems Review*, January 1987, p. 7.

12. Schell, R. R. et al, "Security Kernel Design and Implementation: An Introduction," *Computer*, July 1983, pp. 14-22.

13. Rutledge, L. S. and Hoffman, L. J., "A Survey of Issues in Computer Network Security," *IEEE COMPCON Proceedings*, Spring 1984, pp. 296-308.

14. Winkler, S. and Danner, L., "Data Security in the Computer Communication Environment," *IEEE Computer*, February 1974, pp. 23-31.

15. Xerox Corporation, Xerox Network Systems Architecture: General Information Manual, Palo Alto, California, April 1985.

16. Rochester Institute of Technology Graduate Computer Science Department, "Guide to the Masters Thesis," May 1988.

17. Spafford, E. H., "Crisis and Aftermath," *Communications of the ACM*, June 1989, pp. 678-687.

18. DeMarco, T., Structured Analysis and System Specification, Yourdan Inc., New York, New York, 1979.

19. Saltzer, J. H. and Schroeder, M. D., "The Protection of Information in Computer Systems," *Proceedings of the IEEE*, September 1975.

20. Tanenbaum, A. S., Computer Networks, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1981.

21. Loepere, K., "Resolving Covert Channels Within A B2 Class Secure System," *Operating Systems Review*, July 1985, pp. 9-28.

22. Deutsch, M. S. and Willis, R. R., Software Quality Engineering: A Total Technical and Management Approach, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1988.

23. Peterson, J. L. and Silberschatz, A., Operating System Concepts, Addison-Wesley Publishing Company Inc., Reading, Massachusetts, 1985.

24. Comer, D., Operating System Design: The XINU Approach, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1984.

25. Lampson, B. W., "Protection," *Operating Systems Review*, January 1974, pp. 18-24.

26. Mullender, S. J. and Tanenbaum, A. S., "Protection and Resource Control in Distributed Operating Systems," *Computer Networks*, November 1984, pp. 421-432.

27. Evans Jr., A. and Kantrowitz, W., "A User Authentication Scheme Not Requiring Secrecy in the Computer," *Communications of the ACM*, August 1974, pp. 437-442.

28. Lampson, B. W., "A Note on the Confinement Problem," *Communications of the ACM*, October 1973, pp. 613-615.

29. Seeley, D., "Password Cracking: A Game of Wits," *Communications of the ACM*, June 1989, pp. 700-703.

30. National Computer Security Center, Department of Defense Password Management Guideline, Ft. George G. Meade, Maryland, April 1985.

31. Walker, S. T., "Network Security Overview," *IEEE Symposium on Security and Privacy*, 1985, pp. 62-76.

32. Anderson, J. P., "A Unification of Computer and Network Security Concepts," *IEEE Symposium on Security and Privacy*, 1985, pp. 77-87.

33. Abrams, M. D. and Jeng, A. B., "Network Security: Protocol Reference Model And The Trusted Computer System Evaluation Criteria," *IEEE Network*, April 1987, pp. 24-33.

34. Tanenbaum, A. S. and Van Renesse, R., "Distributed Operating Systems," *Computing Surveys*, December 1985, pp. 419-470.

35. Peters, T., Thriving on Chaos: Handbook for a Management Revolution, Alfred A. Knopf Inc., New York, New York, 1987.

36. Boehm, B., Software Engineering Economics, Prentice Hall Inc., Englewood Cliffs, New Jersey, 1981.

37. Schlichter, J. H., and Miller, L. J., "FolioPub: A Publication Management System," *Computer*, January 1988, pp. 61-69.

38. Birrel, A. D. et al, "Grapevine: An Exercise in Distributed Computing," *Communications of the ACM*, April 1982, pp. 260-274.

39. Johnson, J. et al, "The Xerox Star: A Retrospective," *Computer*, September 1989, pp. 11-29.

40. Lauer, H. C. and Needham, R. M., "On the Duality of Operating System Structures," *Operating Systems Review*, April 1979, pp. 3-19.

41. Lampson, B. W. and Redell, D. D., "Experience with Processes and Monitors in Mesa," *Communications of the ACM*, February 1980, pp. 105-117.

42. Xerox Corporation et al, The Ethernet: A Local Area Network, November 1982.

43. Birrell, A. D., "Secure Communications Using Remote Procedure Calls," *ACM Transactions on Computer Systems*, February 1985, pp. 1-14.

44. Israel, J. E. and Linden, T. A., "Authentication in Office System Internetworks," *ACM Transactions on Office Information Systems*, July 1983, pp. 193-210.

45. Xerox Corporation, Printing Protocol, Xerox System Integration Standard, XNSS 118404, April 1984.

46. Digital Equipment Corporation, Introduction to Local Area Networks, EB-22714-18, 1982.

47. Baer, J. L., Computer Systems Architecture, Computer Science Press Inc., Rockville, Maryland, 1980.

48. Xerox Corporation, Interpress Electronic Printing Standard, Xerox System Integration Standard, XNSS 048601, January 1986.

49. Xerox Corporation, Filing Protocol, Xerox System Integration Standard, XNSS 108605, May 1986.

50. Smith, R., "An interesting answer to the distributed time problem," Posting to *Risks to the Public in Computers and Related Systems*, 20 September 1989.

51. Downs, D. D., "Operating systems key security with basic software mechanisms," *Electronics*, 8 March 1984, pp. 122-127.

52. Housley, R., "Authentication, Confidentiality, and Integrity Extensions to the XNS Protocol Suite," *Security Audit & Control Review*, Fall 1989, pp. 17-24.

53. Needham, R. et al, "A Logic of Authentication," *Operating Systems Review*, December 1989, pp. 1-13.

54. Arsenault, A., "The risks of not learning?," Posting to *Risks to the Public in Computers and Related Systems*, 3 January 1990.

55. Leichter, J., "The risks of not learning and of ignoring realities," Posting to *Risks to the Public in Computers and Related Systems*, 5 January 1990.

56. Leichter, J., "Re: password sharing," Posting to *Risks to the Public in Computers and Related Systems*, 9 January 1990.

57. Weinberg, G. M., Becoming a Technical Leader: An Organic Problem-Solving Approach, Dorset House, New York, New York, 1986.

58. Dijkstra, E. W., "On the Cruelty of Really Teaching Computer Science," *Communications of the ACM*, December 1989, pp. 1398-1404.

59. Kernighan, B. W. and Ritchie, D. M., The C Programming Language, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1978.

60. Pratt, T. W., Programming Languages: Design and Implementation, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1975.

61. Jentz, G. A. et al, West's Business Law, West Publishing Company, St. Paul, Minnesota, 1990.