A Use-Condition Centered Approach to Authenticated Global Capabilities:[1] Security Architectures for Large-Scale Distributed Collaboratory Environments

William Johnston and Case Larsen
Imaging and Distributed Computing Group, Information and Computing Sciences Division
Ernest Orlando Lawrence Berkeley National Laboratory
University of California
Berkeley, CA, 94720

Abstract

We are developing a security model and architecture that is intended to provide general, scalable, and effective security services in open and highly distributed network environments. Our objective is to provide, especially for on-line scientific instrument systems, the same level of, and expressiveness of, access control that is available to a local human controller of information and facilities, and the same authority, delegation, individual responsibility and accountability, and expressiveness of policy that one sees in specific environments in scientific organizations.

Our model is based on a public-key infrastructure and cryptographically signed certificates that encode use-conditions that are defined by those directly responsible for a resource. Certificates that encode user characteristics that satisfy the use-conditions are supplied by those who can attest to the characteristic. The collection of certificates specifying use-conditions and their satisfaction are combined with on-line (real-time) access control mechanisms to enable remote instrument operation. The real-time mechanisms are intended to provide the level and scope of credential validation commensurate with the consequences of the actions that are enabled / protected by the security system.

This general approach is not unlike the directions of the financial information industry is taking to enable global distributed enterprise. One of our proposed uses of the model (supporting real-time construction of distributed computing and storage systems based on use-condition certificates) is similar to the distributed enterprise / electronic commerce capabilities envisioned by the financial industry.

We also describe a prototype implementation that we are using to experiment with this model, and that is providing security services for several distributed applications.draft

1. This work is supported by the U. S. Dept. of Energy, Energy Research Division, Mathematical, Information, and Computational Sciences office (http://www.er.doe.gov/production/octr/mics), under contract DE-AC03-76SF00098 with the University of California. Author's address: 50B-2239, Lawrence Berkeley National Laboratory, Berkeley, CA 94720. Tel: +1-510-486-5014, fax: +1-510-486-6363, wejohnston@lbl.gov, http://www-itg.lbl.gov/~johnston. This document is report LBNL-38850, version "Security.Arch.Global.Cap.CERN.fm".

Contents

1.0 Introduction

1.1 The Need

On the one hand, there is a great potential for highly distributed computing to benefit science:

> *The fusion of computers and electronic communications has the potential to dramatically enhance the output and productivity of U. S. researchers. A major step toward realizing that potential can come from combining the interests of the scientific community at large with those of the computer science and engineering community to create integrated, tool-oriented computing and communication systems to support scientific collaboration. Such systems can be called "collaboratories."*
>
> *"National Collaboratories - Applying Information Technology for Scientific Research," Committee on a National Collaboratory, National Research Council. National Academy Press, Washington, D. C., 1993.*

On the other hand, this will not happen in an open network environment until we put into place a strong and flexible security architecture and infrastructure:

> *"The access to a remote collaborative environment, whether it is for monitoring or for control of experiments, requires that security and access limitation mechanisms be in place, so that safeguards against unauthorized access and privacy of proprietary data exist.*
>
> *The advent of collaboratories brings a new class of user to the ALS. These users are likely to be much more occasional and less experienced with the equipment than has been the case in the past. Collaboratories will provide network based access to very expensive equipment and must be designed to avoid several potential security and safety problems. They must also be designed to have automated equipment failure modes with sanity checks on all incoming data and be resistant to network-based tampering. With respect to remote users, one significant issue is to ensure that use-conditions of the remote resource have been met."*
>
> *(From "Spectro-Microscopy Collaboratory at the Advanced Light Source Project Summary", http://www-itg.lbl.gov/~deba/ALS.DCEE)*

Remote operation of instruments and other equipment in distributed collaboratories, and cross-organizational distributed systems, are examples of the scientific community's movement toward distributed enterprise. The financial information and services industry's move toward global operation - especially in the areas of contracts and delegated authority - presents similar issues, and architectures similar to the one being described here are being designed to support global distributed commerce. [GASD96]

1.2 The Problem and Objectives

It is important to understand of the approach described here, what problems and capabilities are, and are not, addressed. We are addressing access control -- both to ensure correct, policy-based, use of resources by legitimate users, and to prevent unauthorized access by pranksters, criminals, and terrorists. There are many aspects of the problem -- see Denning's very readable overview [Denning96] -- that application-level access control does not address. It is, however, one of several essential components for protecting the on-line environment.

Access to remote instrumentation resources via open networks, whether for monitoring, control of experiments, or collection and manipulation of data, requires that authentication and access control mechanisms be in place in order to provide safeguards against unauthorized access, and to assure privacy of proprietary, confidential, or otherwise restricted data. Therefore, awkward, intrusive, and expensive security will delay wide-spread use of remote instrument resources in open environments.

In addition, we would like the security architecture and supporting infrastructure to be general enough to enable enforcement of resource owner specified use-conditions that can also be used for operations like automatic brokering of computing and storage resources. This latter capability should support a scenario where resource owners (either as a mainline business, or as a barter-based use of excess capacity) advertise the resources, and the user agents collect and assemble these resources into useful systems based on satisfying the owner use-conditions. (This could be a very useful service in the world of scientific experiment control, when significant computing resources may be required only for well-defined periods of time, and are idle otherwise.) It is our hypothesis that this capability can be built on the same use-condition based security architecture that is described here.

In order to provide the required capabilities in a manageable, cost effective, and usable manner, the security architecture must be flexible and effective, and easily deployed, administered, and used. In order to realize its potential not only for transparent access control, but also as the basis of an automated resource brokering system, all aspects of the security process after certificates are obtained must be able to be automated.

The overall goal of the security architecture, then, is to encode, distribute, protect, and then act on, information that is needed to provide routine, secure availability of remote resources in a widely distributed environment, and to enable a level of automated processing that will permit resource brokering.

Our approach is to identify and demonstrate use-condition based authentication and access control methodologies that are applicable to all user-level resources - data, storage and processing, control of external devices, etc. - in widely distributed systems, and also that these methodologies are general, scalable, “strong”, non-intrusive, and easily managed (for the administrator, the user, and those who supply the supporting information).

1.3 The Scope of the Infrastructure, and the Related Threats, Vulnerabilities, and Consequences

The Communications and Information Infrastructure provides the access, analysis, and control capabilities for the information and on-line systems that are increasingly and integral and vital part of our society. The communications part of this infrastructure consists of the public switched networks and the global Internet, and the information infrastructure consists of the millions of computer systems that collect, store, organize, and analyze information, and provide control based on this information. It is not possible to isolate systems from this infrastructure for many physical, logistical, and strategic reasons, and we must therefore understand its scope, and related threats and vulnerabilities and their corresponding countermeasures.

The infrastructure consists of three primary aspects: an underlying “link” layer that moves data from point to point, a network and transport layer that deals with addressing, routing, and data transport services, and the data manipulation components (computing systems). The combination of these provides a full spectrum of communication, computation, control, information, and human collaboration systems.

The link layer has historically been quite heterogeneous, consisting of almost any transmission technology that can move digital data: the analogue telephone system, selected television channels in a cable TV distribution system, spectrum multiplexed optical fibers carrying many independent channels, shortwave radio, etc., as well as modern technologies designed for digital data (e.g. the Synchronous Optical Network technology - SONET - that provides transport for most of today's commercial telecommunications, cellular digital packet systems for mobile computing, etc.).

Most wide-area data networks use "public switched networks" - the system of control centers, communication lines, and circuit switches that are maintained and operated by the telecommunications industry - for their link layer.

The network and transport layer provides the mechanisms for assigning addresses to every communicating end-node, for "routing" (finding a path from one end-node to another), and it provides the "services" that allow computers to communicate. The Internet technology dominates today's computer communications. In addition to addressing and routing, it defines how data is transported over many different types of links (in other words, it enables "inter-networking" of many different transport media) and it provides standard data transport services for computer applications (e.g. remote program invocation, video and audio transport for computer mediated, human collaboration and multimedia applications, remote transaction services for distributed databases and instrument control, information transport such as that used by the World Wide Web, e-mail, etc.). The ARPA developed Internet technology differs from the public switched network primarily in that all communication is accomplished by breaking the data communication that provides all of the services mentioned above into "packets" that are independently routed through an inter-network. (As opposed to the entire data stream being carried on a fixed path circuit as with a telephone call, though this distinction is rapidly fading as modern telephony systems migrate toward packet-like transport.)

Most modern computer applications use Internet technology for communication, whether through the global, public Internet, or in private (isolated) corporate Internets (sometimes called "Intranets").

The final layer of the Infrastructure is the computing systems that generate, manipulate, store, display, or control by using, information, facilitate human collaboration and creative activities, etc.

This combined "Communications and Information Infrastructure" is the basis of our modern information society. The future scope of the communications and information infrastructure is difficult to predict, but current trends suggest that it will be directly involved in virtually every aspect of our lives: from control of household appliances by energy utilities to optimize energy usage, to tele-robotic surgery, to remote control of manufacturing; from delivery of every form of multimedia information and entertainment, to the provisioning of our country's defensive capabilities. Therefore, the consequences of the disruption, subversion, corruption, or monitoring of this infrastructure will be roughly equivalent to these activities happening today simultaneously to the telephone, broadcast, and electric power systems.

When considering the threats to this infrastructure, there has been a tendency in the past to separate the layers and examine them independently. This is no longer completely appropriate. Every component of every layer relies on computer control, and increasingly Internet communication is being used to provide control and management of the components of the layers below the Internet: the public switched network and the routing infrastructure of the Internet itself [NCS1]. The same vulnerabilities - albeit with different consequences - exist in each layer: the entire public switched network is computer controlled; the routing infrastructure of the Internet is computer controlled; and at the "top" layer, an incredible and increasing variety of services are computer controlled.

Therefore, many of the research and development issues for securing the infrastructure from attack are common to all of the Communications and Information infrastructure components. What differs between the layers, of course, are the specific techniques, consequences, recovery procedures, etc.

Vulnerabilities at each level provide opportunities (threats of) monitoring, penetration, masquerading, subversion (of correct operation), and denial of service [NCS2].

Monitoring vulnerabilities arise from physical access to transmission media, subversion of switching and routing components (to divert streams through misconfiguration or bogus instructions) or direct access to data at the client systems. The impact of monitoring varies widely with the nature of the information that is disclosed. Passive monitoring may be very difficult, if not impossible to detect, active monitoring is typically the result of successful intrusion of a computer system. Penetration of systems exposes various aspects of the operation of those systems, and the risk of exposure of information. If the target systems are responsible for operation of the infrastructure, then the impact extends to every facet of the infrastructure controlled by the compromised system. Masquerading as a legitimate user or other entity conveys the access and control rights of the legitimate user. Subversion of operation results from successful penetration or masquerading. Denial of service can result from subversion of the communication infrastructure or from direct attacks on limited resources.

The access control approach described in this paper addresses enforcement of use-conditions, confidentiality of data and transactions, and ease of use by the stakeholders setting policy and use-conditions, as well as ease of use by the intended users. Application-level approaches, such as we describe here, do not address the vulnerabilities of the infrastructure: telecommunications, network routing and name resolution, correct and secure functioning of the computing environments in which we embed our applications and control paradigms, etc. However, access control is an increasingly important aspect at all levels of the infrastructure, as the infrastructure components all start to look like, or are controlled by, computer based resources.

1.4 Overall Approach

In the next two sections we will briefly discuss how we decompose the overall problem for the purpose of description, analysis, implementation, and operation. This is followed by a brief discussion of the related cryptographic and security systems concepts.

1.4.1 Security Model

The security model addresses what is being protected, and how.

Our general model addresses access control and data confidentiality for computer mediated resources. We treat these as end-to-end aspects of security, and do not directly the other “infrastructure” security issues noted above (though this general approach has applicability to some infrastructure problems).

The resources have use-conditions specified by the responsible parties and represented as cryptographically signed statements. Each use-condition is matched by an attribute that is also provided as a signed certificate by a party trusted to attest to the required characteristic.

Access control is then based on a set of credentials that verify that the use-conditions have been matched to corresponding attributes. From an abstract point of view, this model is intended to represent how human organizations normally deal with similar sorts of resource access control today.

Access control lists are an easily implemented and important special case of the use of this model that lets us incrementally build and test an implementation of the general architecture.

The security model drives the design of the architecture.

1.4.2 Architecture

The security architecture specifies how the various components fit together into a system in order to provide a complete and flexible suite of capabilities that are needed to protect the resources. For example, key management, certificate management, security context establishment and user authentication, message authentication, confidentiality, application communications libraries, interfaces to all of the necessary mechanisms, etc., all must interoperate.

The architecture is based on public-key infrastructure ("PKI") and signed certificates[2] because these technologies potentially provide for an inherently scalable and secure mechanism for widely distributing the assured information needed for authentication and authorization.

Significant aspects of the architecture are also driven by operational requirements: who specifies use-conditions and access control lists, how do the computing systems administrative domains and application security domains interact (they are likely to have different policy concerns), etc. The architecture is also driven by the availability of components, current directions in the commercial and standards communities, etc.

Our architecture supports both the IETF Generic Security Service [RFC1508] and the Secure Sockets Layer [SSL] to provide authenticated and confidential messaging between components of distributed systems, together with the collection of functions needed to establish the security context. The various certificate and private key management, certificate generation, storage, location, and use techniques, tools, and APIs, as well as some other tools needed to build a useful and deployable security system are supported in various ways that are, in principle, similar to the Architecture for Public-Key Infrastructure [APKI], but certainly do not supply all of the standardized interfaces provided in that work. The architecture and implementation is designed to be modular in order to allow experimentation.

1.4.3 Infrastructure

The infrastructure supports the architecture by providing basic functionality, e.g. public key certificates, certificate location and distribution mechanisms, private key management, certification authorities, etc. From a practical point of view, characteristics of the available and emerging infrastructure also drive the architecture.

Public-key certificates provide the mechanism of establishing identity and distributing the cryptographic information needed to use that identity for user and message authentication. The certificates, in turn must be locatable and certified, the mechanisms for which are two more parts of the required infrastructure. The computing community is in the process of defining and deploying these infrastructure components, and there remain unresolved alternatives (like X.509 vs. SPKI vs. SDSI certificates for identity [PKIX], [SDSI], [SPKI]) providing many aspects of this infrastructure that remain without common agreement or implementation.

Our approach to security infrastructure is to use prototype standard mechanisms and encodings that will integrate with institutional PKI as it is deployed. As will be noted, we are also developing

2. We will use the term "certificates" to refer to documents used for authentication, authorization, and representation of attributes and whose integrity is assured through cryptographic techniques. These are also known as "public key certificates", "digitally signed certificates", or just "signed certificates".

methodology and APIs for using the Web to provide for storing, locating, and retrieving various types of public key certificates. These APIs can supplement LDAP/X.500 APIs if and when they turn out to be the common certificate distribution mechanism.

1.4.4 Operations

Practical operational mechanisms that minimize the intrusive and administrative impact of the security environment are needed for certificate definition and issuance, integration with computing system administration, etc.

1.5 Background

This section gives a brief overview of the relevant security technologies and how they fit into our model and architecture. (See [RSA96] and [For95] for additional background information.)

1.5.1 Asymmetric-key (e.g. public-key) Cryptography

The enabling underlying technology for many aspects of our approach is public-key cryptography which provides the capability for one person ("A") to encrypt a piece of information with a private-key, and another person to decrypt that information at a remote location by using only a widely distributed public-key. Only the public-key corresponding to the private-key can decrypt the original message, thereby ensuring a unique and identifiable origin. Operating in the reverse manner, anyone can encrypt a message with A's public-key, and only A can decrypt that message.

1.5.2 Digital / Cryptographic Signatures

Cryptographic document signing is a technique that ensures the authenticity of the document without the physical possession issues of a holographic signature. The signing process typically involves encrypting a "hash code" of the document with the author's private key[3]. A hash is a much smaller, but unique, code derived by a mathematical transformation of the document. The uniqueness "guarantees" that the document itself cannot be changed without this hash code changing. Encrypting the hash with the author's private key thus assures that only the author could have created (or altered) the contents of the document. The purpose of such signing, then, is to guarantee that any attempt to modify the contents of the document from what was signed by the author can easily be detected[4]. This process of integrity assurance is commonly called a "digital signature". Such digital signatures do not provide confidentially of the contents of the document. One of the characteristics of digital signatures is that they do not change the document contents. Like a holographic signature, the digital signature is usually appended to the clear text of the document. (If confidentiality is desired as well, that is handled separately.)

Certificates are a special case of general signed documents in that they are usually signed by some trusted third party, and their function is typically to participate in the authentication (identity verification) and authorization phase of a security system.

---

3. Throughput this paper the term "private" key will refer to the private portion of a public-private key pair.

4. This does not, of course, prevent the author from changing the document. If a such a guarantee is required then the original hash (or the whole document) can be sent to a cryptographic timestamping service (such as the US Postal System is planning to offer) which adds a timestamp and signs the hash of the original hash or document plus the timestamp, thereby providing a proof of the contents of the original document at that point in time.

1.5.3 Certificates

In general, certificates are small documents, some of which may have a standardized format (e.g. X.509 or SPKI) and some of which do not. Certificates can encode information about a principal, or information expressed by a principal, or a relationship between principals, in a secure and verifiable way. Certificates that provide some policy based assurances of the identity of the principal we call identity certificates. Certificates that encode organizational / group affiliation, creditworthiness, level and scope of training, etc., we call attribute certificates. Certificates that encode authority delegation (and restriction) are called authorization certificates. Certificates that encode use-conditions, e.g., cost, required role attributes (personal identity, group membership, organizational function), required personal characteristics (training, credit worthiness) etc., we call use-condition certificates. Certificates that encode a relationship of a principle to a policy (e.g. that one certification authority operates under the policy of another) we call trust certificates.

All of these certificates are cryptographically signed documents. The certificate is signed using the private key of the certificate issuer. The issuer's identity may, in turn, be verified by obtaining the issuer's public key from a "trusted" source and then using that public key to "decode" the document signature, an operation that can only succeed if the private key and the public key were originally generated as a pair. The job of a certification authority (see below) is to assign that key pair to a person of verified identity. The trusted source of the public key is typically a publicly accessible database maintained by the CA.

1.5.4 Information Encoding

In order to enable automated processing there must be a machine-comprehensible encoding for the representation and automatic manipulation of use-conditions and attributes. While our first implementation will use ad-hoc representations, we are also looking into various approaches such as the DCE "Authorization for Distributed Applications and Groups" (ADAGE) project (see [HMSZ96]), the generalized certificates being discussed in the IETF SKPI working group [SPK], and systems like Policymaker [BFL96], and maybe PICS [Wor96] because they are addressing the problem of how to express many different trust relationships at the same time. There is also work being done in the commercial sector on general encodings for authorization certificates (e.g. X9.45 - see [Ank]).

1.5.5 Certification Authorities

CAs serve a dual role in our model. On one hand they "certify" that the holder of a private-public key pair is associated with a particular identity. The association is made by the CA digitally signing a certificate that contains some personal identification (legal name, e-mail handle, CPU id, MAC address, etc.) and the public key of that individual / system. The strength of this association (e.g. what documentation or other "conventional" / societal proof of identity - driving license, birth certificate, etc. - were required, and how were they checked before an identity certificate was signed by the CA) is a matter of (published) policy for the CA. There may be, as is common for commercially operated CAs, different "levels" of certificate depending on the strength of the identity verification.

The other task that we relegate to CAs is to represent the root of organizational authority and to sign certificates that delegate parts of that authority throughout the organization. This function is different from the identity CA, and will be discussed more later. (We should perhaps call the certification root "CR" to avoid confusion with the conventional CA.)

There must also be a mechanism to establish and represent trust relationships (policy and procedure agreements) among authorizing or verifying entities. The common ways of doing this are represented in a continuum from a centralized root, hierarchical structure of certification authorities that operate under some set of common policies (see [Ken93]), through "webs" of organization-scope CAs, to the completely decentralized approach of PGP (where individuals attest to each other's attributes) at the other end. All of these approaches are in use, and we have focused on the middle ground of independent organization-level CAs that attest to specific facts (such as identity) and establish inter-organizational trust on a pair-wise basis.

Trust relationships between CAs (domains) can be represented by an exchange of trusted public keys. If there is a previously established common point of trust - e.g. a single CA that signs key in both domains - then these public keys (belonging to, e.g., departments at two different hospitals that need to exchange patient information) can be exchanged in certificates whose origin can be verified by the common relationship with the CA. In the absence of a common point of trust (e.g. a top level CA, a la RFC-1422) then a "pair-wise" trust relationship can be established among CAs by a secure, out-of-band exchange of CA public keys that are then offered to the local CA community under the signature of the "home" CA. The availability of a trusted public key from a different domain / organization can then be used to verify certificates passed between parties in the two organizations, thus permitting cross-organization secure transactions. What is "trusted" between the CAs is a matter of policy, but the implied minimum trust is the identity verification process (and operating procedures) of the other CA.

1.5.6 Certificate Distribution

There must be widely and reliably available mechanisms for making certificates locatable based on content. There are several current approaches to this. X.500 directory servers provide, in principle, a standard way of searching for and distributing X.509 identity and attribute certificates. There are already WWW sites and ftp sites that provide PGP certificates. Other distributed information mechanisms (e.g., whois++) are possible. Another approach is to use Web-based searching of textual representations of certificates. In this approach we can use the very fast Web search engines for text searching, and thus search the distributed environment over a large space of textual data sets. The results of a search would be URLs of the certificates that match the search criteria. If the certificates are binary objects, then this approach is essentially the same as an image library capability for indexing large data-objects. (See [TJ96] and [JA95].)

Our current approach is focused on the Web search method mentioned above for all sorts of certificates, and we may also use the "Lightweight Directory Access Protocol" [LDA96] to communicate with X.500 servers when they contain X.509 identity certificates.

2.0 A Use-Condition Centered Security Model

We are evolving a use-condition centered model that is based on: 1) verifiable and securely represented use-conditions imposed and documented by the resource controller / owner / host institution, and; 2) equally secure and verifiable satisfaction of those condition by the "entity" (individual or its agent) requesting access, through the acquisition and presentation of certain certified attributes that satisfy the use conditions. The model provides "decentralized" resource management because responsibility is vested in the people / institutions directly involved in setting use-conditions or verifying their satisfaction. Responsibility for actions is traced back to an individual identity, and/or role in an organization, through the chain of certificates that lead to a permitted access / action.

Our model is trying to emphasize direct checking of satisfaction of use-conditions imposed by a resource controller rather than identity-based access control lists, however, the result of matching user credentials to use-conditions implies an access control list, and using our model, traditional ACLs can be maintained directly by the responsible party from their “home” workstation and accessed from a certificate database.

The basic idea is that the principals that control resources will establish a set of conditions for the use of the resources. The use-conditions are whatever is appropriate to the resource: a level of training for operating an instrument; membership in a group (e.g. health care professionals working for a particular HMO for access to patient data); creditworthiness for access to a resource that requires a commitment of payment; etc. These use-conditions are then encoded by the responsible party in signed certificates and deposited in a “trusted” server for public access.

Paired with the resource owner’s use-conditions are the principals that can attest to the relevant attributes of a user or agent that is seeking access to a resource: A school registrar may certify a level of training; a state examining board may certify a level of professional attainment; an organization can certify employment (and related group membership); a bank, a level of credit; and a certification authority may attest to some form of individual identity. Again, the principal (person or organization) that can attest to some attribute, expresses that fact in a signed certificate that is deposited in a trusted server. See Figure 1.

There are several underlying tenets of the model that we are evolving. One is the notion of decentralized / delegated authority and the other is that of use-condition and attribute certificates.

Delegated authority is essential to automated and scalable operation This approach is how virtually all modern organizations work. The authority “root” - the CEO, Laboratory Director, Campus Chancellor - delegates portions of their authority down the organizational hierarchy, with the authority to “act” becoming steadily more specific (restricted). Finally, that delegation reaches a person who has the actual responsibility to act: sign a contract, operate a piece of equipment, etc. This authority delegation is traditionally accomplished through a (written) collection of policies and procedures that define how employees conduct the day-to-day operations of the organization.

The notion of delegated authority in our model accomplishes the same diffusion of authority, but is implemented differently. We use a collection of signed certificates that may - if necessary - be traced through an unbroken chain of such certificates back to the root of authority. The decentralized aspect comes both from the fact that the authorization for an individual at the “bottom” (broad base) of the organization does not come directly from the root of authority, and that the responsible parties generate and sign certificates commensurate with their responsibility and authority. Again, this is similar to the traditional model: For example, the purchasing agent does not have a personally addressed and signed letter from the CEO saying that he or she can sign certain kinds of contracts - that appropriately restricted authority comes from the head of the purchasing department, and so on up the hierarchy. This same model is appropriate when digitally signed certificates convey the delegated authority. The reason for this is that for the root authority CA to sign a certificate requires a personal actions by the corresponding human authority (e.g. to provide a smart card or PIN to release the private key for the signing). (This is why we call the root authority a CA, and hypothesize that it should probably be separate from the identity CA. One is used rarely and the other frequently, but the mechanisms are essentially the same: such systems probably require multiple parties to be present to accomplish their function and are probably isolated, physically secured systems.)

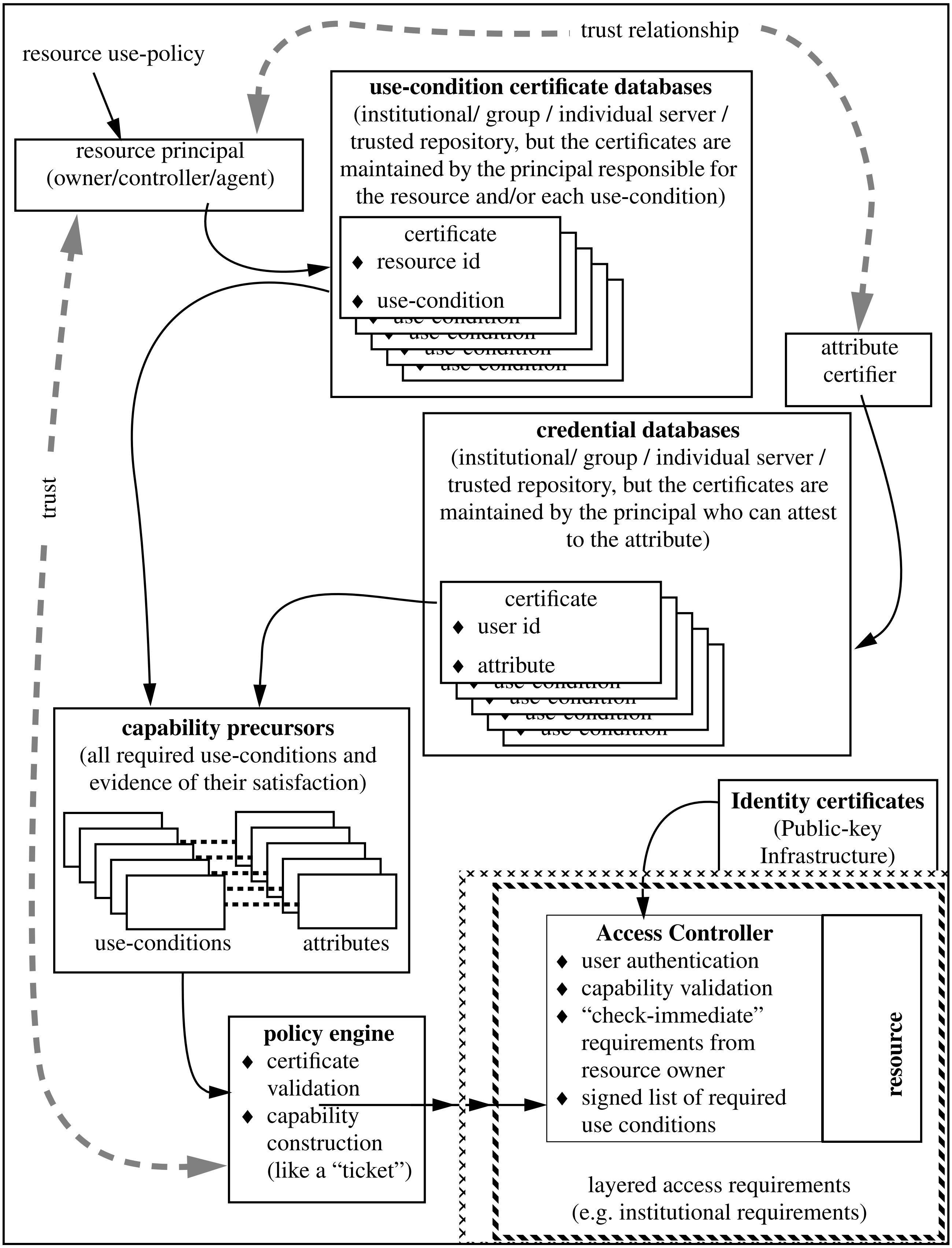


Figure 1 **The Security Model**

As the authority diffuses down the hierarchy, individuals generate and sign certificates in a more routine and location independent fashion than does the authority root. So, if the principle investigator - the person with responsibility and authority over a scientific instrument - is half-way around the world from the instrument (not uncommon in our environment), the PI can set up use-conditions in certificates signed at his/her local workstation and then communicate those certificates in whatever manner is convenient or appropriate (including, for instance, advertising them widely).

The notion of decentralized authority does not in any way imply abrogation of institutional authority. The institution (e.g. university, laboratory, hospital, etc.) in which the resource is located normally has responsibilities for aspects of operation relating to safety, ethical conduct, etc. As such, the institution is also a principal with respect to the resource, and will impose its own use-conditions that operate in “conjunction” with those imposed by the “first level” resource owner (e.g. a scientist who has constructed a piece of apparatus with their grant funds). That is, both sets of use-conditions must be satisfied before access is permitted.

As with many distributed systems, the decentralized aspect of the model is a key to scalability: The administrative task of maintaining an appropriate expression of access policy is only dealt with by those entities (principals, or their agents) that are directly responsible for the policy governing the use of a resource (i.e. they provide a signed certificate containing use-conditions). The operational administration of the resource (e.g. a local human operator or computing system) only has to provide the basic support for the access control mechanisms (e.g. the security infrastructure at the resource to authenticate the user and the verification agent -- e.g. a policy engine -- that signs the capability), as well as implementing resource-specific “check-immediate” conditions (that are distributed directly from the resource owner, or built into the resource access control mechanisms). Organizational certification authorities need only provide identity assurances, together with the infrastructure (such as X.500 or WWW) for distributing the identity certificates. Central administration of information like access control lists, or multiple roles that have to be remembered by resource users and controllers, etc., are not scalable approaches in a global distributed environment -- especially when the responsibility for administering the access control is shared by multiple parties -- and are aspects of security that this model is intended to replace.

Use-conditions are specific requirements that must be satisfied before actions or uses are allowed. Certain resources (e.g. remotely operated scientific instruments) may have a complex set of use-conditions that are imposed by different authorities in different domains, all of which have to satisfied simultaneously. (For example, the instrument remote operation may require a level of operational proficiency (a PI condition), safety training (an institutional condition), group membership (PI condition), time of day restrictions (an operations group condition), etc. It is part of our model that these use-conditions are represented by certificates created and signed by the directly responsible entities in the respective authority domains.

The use-conditions are satisfied by a corresponding collection of attributes that are certified by people / domains qualified to make that certification (e.g. a school or training center or a delegated authority). The matching of use-conditions and attributes, the trust relationships among condition imposers and those that certify compliance, etc., are all aspects of the model (though ones that are not likely to be implemented in an automated fashion in the short term).

While such a model is complex, it, and similar models related to commercial contracts and conduct of business, are essential for global, distributed electronic commerce. [GASD96] As such, there are commercial sector efforts to define and implement these models (e.g. X9.45) The scientific community should participate in these efforts, because while there are significant similarities in the models, there are also significant differences in the basic operational aspects of the commercial and scientific sectors. (One example is that scientific facilities - government labs, universities, etc. - tend to share facilities in various ways, while commercial entities tend not to.) It remains to be seen how much of the commercial sector model than enables sales, contracts, etc., can be used in a facility sharing environment, but the more commonality the better.

In addition to use-conditions and certification of attributes, a collection of ***trust relationships*** must exist. So, for example, in order for a resource controller to accept a certificate that purports to represent satisfaction of a use-condition, the resource owner must trust that the principal that is attesting to (certifying) an attribute of the user, is qualified and/or certified to make such a claim. These trust relationships may be direct: the resource owner and the verifier of conditions may have an explicit, pair-wise agreement; or they may be indirect: a resource owner may accept a statement of policy, perhaps policed by a third party, as an implicit guarantee of the satisfaction of a use-condition.

To gain access to network-based resources, the user, or a broker / agent, acting on behalf of the user, collects the certified use-conditions for a resource, and then collects the credentials that verify the user's satisfaction of the use-conditions (the ***verification agent*** of Figure 1). The combination of resource identity, use-conditions, and matching credentials, each of which is cryptographically secured by the relevant principal, are used to form a "capability". A capability allows the user to access, monitor, manipulate, use, etc., a remote resource. One interesting aspect of the process is how to ensure that all of the stakeholders are represented before a capability is issued.

The last part of the model is that of ***check-immediate / deferred*** actions. These actions may be required in the use-conditions, but are deferred to the resource access control mechanism because they have temporal aspects. In a sense, the consuming of a coupon is an example of this. Other examples include checking all manner of time sensitive requirements, re-verification of user credentials in the case of high-consequence access, etc. In many - perhaps most - cases, user presentation of a valid capability - i.e. all use-conditions matched by corresponding credentials, etc. - would be all that is required for access to the resource. That is, the resource controller would only have to authenticate the user and the signer of the capability (assuming that a trusted third party originally assembled and verified the capability). "Check immediate" requirements beyond this (including possible revalidation of all credentials, checking revocation lists, etc.) would be the result of specific policy requirements of the resource principal, or the institution in which the resource is embedded. (E.g. a general safety requirement imposed by the facility where the resource is physically located.)

Another, perhaps not so obvious requirement of some security systems is that the security must be able to be "broken" under certain circumstances. An obvious example is a medical records system under the circumstances when the patient is in an emergency situation and the normally authorized people are not available. In this - and perhaps similar situations - the access control agent might have a special form of ACL (allowing, for example, anyone with the attribute "physician", or any two members of certain organizations presenting credentials simultaneously, etc.), whose use would also trigger full auditing, real-time notification of responsible parties, etc., but access would be permitted. This is a variation of a check-immediate condition at the resource controller.

Finally, it is a goal to enable new distributed enterprise functions. One example is that systems should be able to be built dynamically from network resources using the mechanisms that we have been describing. As with access control, a broker / agent, acting on behalf of the potential user, collects the certified use-conditions for a resource, and then collects the credentials that verify the user's satisfaction of the use-conditions. In addition to use-conditions we will assume a payment requirement. One way of dealing with this is to use a "cybercash" approach, where the cash or coupons are denominated in appropriate units: monetary, resource units, barter units (resource units for other resources), etc. These coupons are presented along with the capability, but are not "consumed" until the resource access is actually accomplished. One could easily imagine coupling this into a commercial banking / settlements system to handle the accounting for this mechanism.

2.1 Basis in Practice

Our general approach is motivated by the credential model used in most Western societies for consumer contracts. An example that illustrates many of the points is that of renting an automobile at an airport. The essential elements of this process include presenting and verifying a set of credentials, and then being given a capability (authorization certificate) in the form of a validated rental car contract, that is then used to gain access to the resource (the automobile).

In this model, a driving license is an attribute certificate that represents a guarantee by a trusted third party (the issuing state Dept. of Motor Vehicles) that an individual has had appropriate training and is currently authorized to drive a car. The driving license also plays the role of identity certificate when the individual's signature must be verified in order to signify agreement to a contract. An individual's physical signature is a "private" or individual expression (like the private-key part of the public-key/private-key pair). The photograph of the individual's face, and the photo of a written signature together on a driving license certificate constitutes the certified "public" expression of identity.

One's credit card represents a creditworthiness credential, and provides the entree for on-line revalidation of creditworthiness. These two credentials (driving license and credit card) are verified (usually automatically), and a capability - to use the rental car, with restrictions - are represented in a third, combined authorization certificate (or "capability"): the rental car agreement.

The rental car company agents are the verification agents. They verify multiple use-conditions by checking a collection of attribute certificates and then issue a (composite) authorization certificate on behalf of the resource owner, the rental car company. We call this new certificate a "capability" because it associates a set of otherwise unrelated certificates to represent the satisfaction of a set of use-conditions that, together, enable the use of a specific resource. The guard at the rental car lot gate is the resource access controller. He checks that the capability is signed by the verification agent and authenticates the user (checks driving license again). The capability, then enables the use of the rental car (the resource).

The success of this model is clear: most responsible people can rent an automobile - a dangerous and expensive resource - anywhere in the world by producing the commonly possessed set of credentials: the driving license and the credit card to satisfy the use-conditions imposed by the resource owner.

This example embodies most, if not all, of the characteristics of a security architecture that can provide secured global capabilities in an open Internet and the abstraction of these operations are necessary, and probably sufficient, conditions to ensure general resource security in the Internet.

2.2 Strengths and Weaknesses of the Model

One of the expected advantages of the model described here is that it should provide a natural (and distributed) responsibility for information validity, placing the basic expression of policy (issuance of a use-condition certificate or an attribute credential) with those principals that have the direct responsibility for the information (e.g., the home state Department of Motor Vehicles) or for setting use-conditions (e.g., the rental car company).

Automated acquisition and verification of attribute credentials will increase efficiency and reliability, and will play a critical role in scaling remote environments by removing the need for human intervention in repetitive and trivial tasks such as credential checking, especially when the human participants are never in the same place at the same time, may never physically visit the site were service is being requested, and may not have any knowledge of each other outside of the contents of the credentials..

TABLE 1. Financial Industry View of Security

← past ---------------------------------- present ----------------------------------- future →			
Physical Protection	**"Blanket" Info Security**	**Specific Application Security**	**Enterprise Security**
safes	ACLs[a]	public-key technology	public-key infrastructure (CAs and certificate distribution)
guards	firewalls	EBT[b]	
fences	authentication	EFT[c]	enterprise functions
	intrusion detection	EDI[d]	electronic commerce
		identity certificates	authorization certificates
		current Web	
closed box security	secure centralized passwds (e.g. Kerberos)		certificates ("the ultimate decentralization" to support global enterprise)
← past ---------------------------------- present ----------------------------------- future →			

a. access control lists
b. electronic benefits transfer (E.g., see: http://www.itsc.state.md.us/ITSC/techrepts/ebt.html)
c. electronic funds transfer (E.g., see: http://www.itsc.state.md.us/info/EDI/chart/eft.html)
d. electronic data interchange (E.g., see: http://www.itsc.state.md.us/info/EDI/chart/ediinfor.html)

This model is not just about security, but handling distributed information in ways that enable new capabilities (e.g. brokering for construction of distributed systems) In that regard, the view of the financial information and services industry seems to be similar: technologies that are components of a security architecture for distributed environments (especially related to contracts and conduct of business) are also essential components of electronic commerce and globally distributed enterprise.

The ability to audit actions should be greater than today, as every aspect of the model is represented by cryptographically signed statements traceable back to their source.

The approach described here - though developed independently - appears to be a good deal in common with the evolution of commercial distributed enterprise modes. This evolution - which is abstracted from [GASD96] is represented in Table 1.

The two most obvious weaknesses are probably the granularity and its highly distributed nature. The verification agent and the resource access agent both have to perform potentially "heavy weight" operations. This will probably not be an issue in the remote instrument operation environment. Secondly, if there are a lot of use-conditions and correspondingly many attribute certificates, all of which are maintained by different entities, then there may be a lot of searching and information movement in a widely distributed environment (with the attendant possibilities of access failure) needed to build a capability. Again, however, in our environment capabilities will probably be fairly long-lived, so this may not be an issue.

A potential weakness is that there is minimal experience with, and deployment of, this sort of distributed infrastructure. Apart from the underlying PKI, two main features will be needed to enable automatic brokering. One is an approach to encoding the relevant information (both use-conditions and attributes that can satisfy them) in ways that can be parsed, understood, and then manipulated by software systems. The second is widely available (e.g. network-based), searchable databases that can be used to “advertise” resources and use-conditions, and to provide repositories for the certificates that can attest to the satisfaction of use-conditions.

Both the verification agent and the access controller have to have some sort of policy verification “engines” to determine the logical validity of a set of certificates, and these must be developed and/or evolved from current capabilities. (See, e.g., [BFL96]).

However, even with the tools and infrastructure in place, they are still just that: tools and infrastructure. In each given situation where a resource is being protected, those responsible must evolve a security policy model that defines the level of assurances, protection, auditing, etc., that meets their requirements. Flexible tools and infrastructure make it possible to implement specific security policy in a usable and cost effective way, but it remains to be seen how “naturally” these tools can be integrated into work environments. Hopefully this form of security will be seen as enabling new capabilities so that it will be accepted as a routine part of the work environment.

3.0 Objectives of a Prototype Architecture

Implementation of the model described above requires a number of operational components with capabilities beyond what we see today:

- Distributed brokers that understand the encoding of the information in corporate database objects that are represented in signed certificates will be needed to automatically build capabilities.
- Coupling between corporate databases and credential servers that will minimize the administrative overhead.

  (This will (eventually) provide the automated maintenance of attributes and automatic credential generation based on information obtained from the primary corporate database (e.g. employee records, school records, etc.).)
- Rules and mechanisms for encoding requirements (the “use-conditions”) for services will be required, as will mechanisms for determining the level of trust to place in other CAs or principals that attest to an attribute. For example, how are CA operating policies published and evaluated, and from whence is the CA’s authority derived (e.g. does the school registrar that signs a training credential have a certificate signed by the State Board of Education), etc.

The basic technology is a generalized certificate understandable to a variety of distributed entities. (For example, see [MH95], [HMSZ96], [Ell96], [BFL96], and [Wor96].) However, while the certificate idea has been around for some time, there is no widespread use or understanding of many of the issues involved, which run the gamut from sociological to legal to technical. Even if we consider only the technical issues, we are faced with a considerable range of topics to address: expression of attributes and certification policy in machine understandable form; distribution and revocation of certificates; usability and performance implications of certificates and their processing on the part of the servers, brokers, and clients, etc.

3.1 Issues

We are currently examining a restricted set of issues in the interest of early application to the domain of remote collaboratories:

- What is the essential information needed for attribute certificates and how can it be simply encoded (e.g. types of training, granting institution authorization, receiving institution authorization)?
- How are certificates generated, obtained and used?
  - What type of infrastructure is required (e.g. functionality, availability, and performance of directory servers used for certificates)?
  - How will the specifiers of use-conditions select use-conditions that correspond to the ability of the legitimate users to obtain matching attribute credentials? (One goal is to make the process as transparent as possible for legitimate users.)
  - How will the certificates actually be employed (to establish session keys, to digitally sign individual transactions, to build capabilities, etc.)
- How will the resulting use of authenticated and / or confidential transactions affect performance, complexity, etc. of the services being provided
- How transparent can this architecture be made to the applications whose only interest is a “yes” or “no” to the question: “Is this an authorized access request?”
- How transparent can this architecture be made to the users who, ideally, should only perceive the presence of security if some use-condition is not met?
- Can this approach be structured to minimize the burden of managing access control conditions?
  - Is it practical to relegate to the information owner or resource owner the task of defining, encoding, and publicizing the required information or setting access conditions?
- How is local control maintained?
  - What are the mechanisms by which the responsible parties local to a physical facility maintain ultimate control of access?

3.2 Approach

To examine and address the issues noted above, we are building a prototype environment that will:

- Provide some example information encodings (e.g. attribute certificates)
- Implement access control mechanisms that use certificates that contain various kinds of information
- Provide prototype implementations of mechanisms for client applications to use the certificates, or their derivatives, for session and transaction authentication, and then characterize the complexity and performance of this type of use of the architecture
- Characterize the performance and complexity issues for locating, serving, and parsing the multiple certificates needed to form a joint certificate to authorize a specific service

- Determine the overall "practicality" of the approach in use in today's environments by using it in at least one application
- Integrate with a PKI

Based on this experience and analysis we will suggest strategies for using this technology to support routine and secure access to DOE experimental science facilities. It is anticipated that the current work will provide:

- Directions for a follow-on, scalable, distributed, approach to secure access of remote resources
- Provide sufficient information to the DOE community to permit making reasonable choices among some of the currently competing approaches (e.g. X.500, DNS, PGP)
- Provide the community with experience in the issues and complexities of dealing with multiple certificate issuers (e.g., on a per-function basis) - an aspect that is important for relegating the certification of different attributes to the responsible organization agents

4.0 Implementation of the Architecture

Our approach to implementing the architecture involves the following points. Please also see Figure 2 and Figure 3.

(1) A certification authority issues identity certificates.

The CA is implemented using the SSLeay [SSLeay] CA. Requests for certificates are made via the html extension used for this purpose [Netscape1]. The CA is off-line, and the certificate request is forwarded to the CA administrator. After identity verification commensurate with the CA policy is accomplished, an X.509 certificate is issued.

Identity certificates can also be issued and managed using an off-the-shelf product such as Entrust [Ent], or a certificate issuing system part of freely available software [You] [GMD] [SES].

(2) Private-key management via the SSH agent. [Ylo] The agent protects the private key by encrypting it based on a pass phrase. When the key is released, the agent can sign documents on behalf of the principal.

(3) Certificate retrieval is via Web-based servers using a free-form organization and search, or relational-style searches. We are also considering LDAP using an X.500 directory organization [LDA96].

(4) Attribute / authorization certificates are generated by specifying use conditions and corresponding attributes by using a controlled vocabulary that is defined by policy.

(5) Resource access control involves several steps:

(5a) A policy engine analyzes certificates. It authenticates the user and matches use-conditions with attributes.

(5b) For mildly security aware applications, a security context is established via GSS/SPKM, after which secure, authenticated, integrity protected, and optionally confidential exchanges can occur using GSS-API [Lin93]

A similar mechanism is used for context establishment for SSL, which provides secure communication at the transport level.

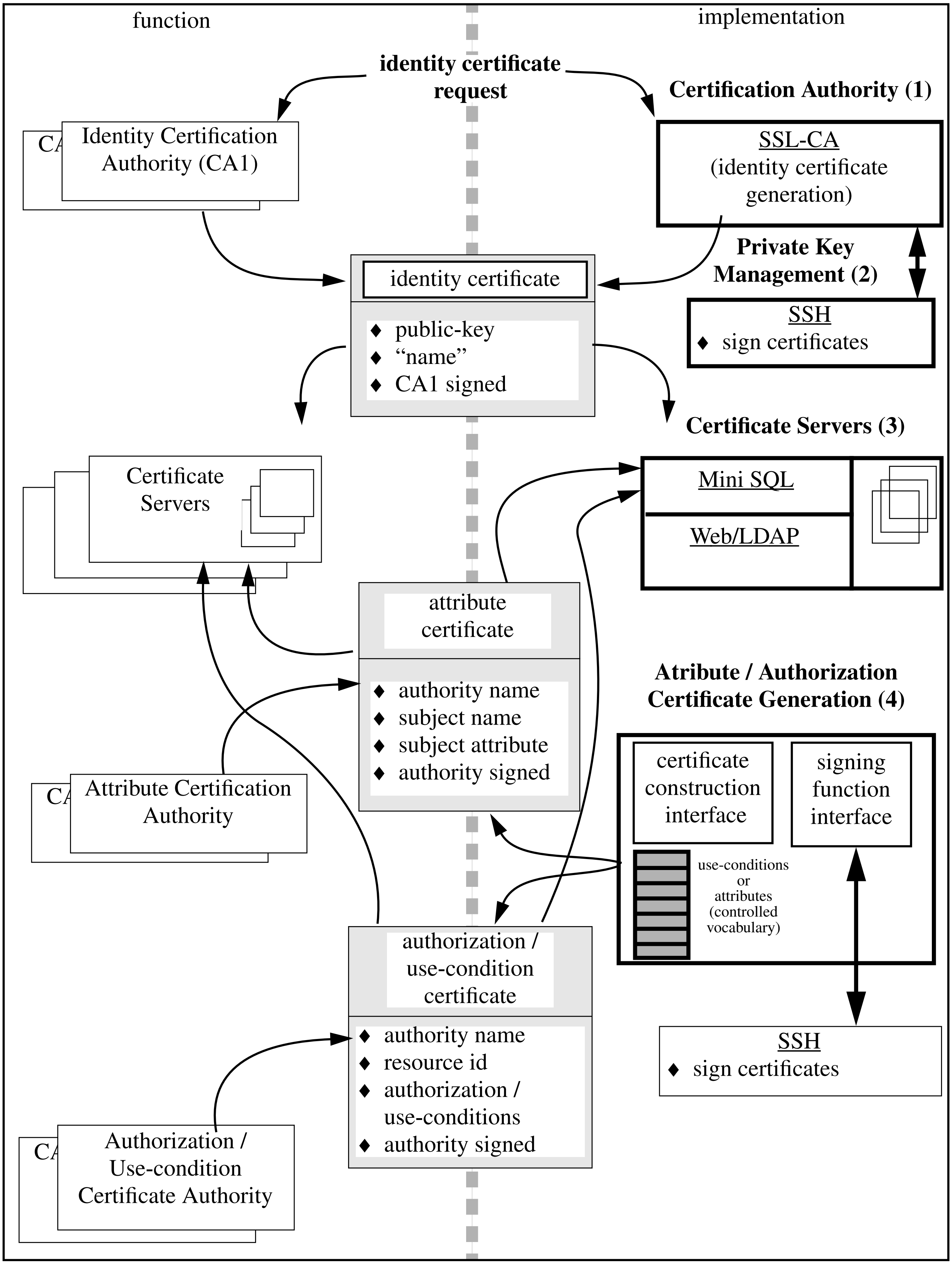
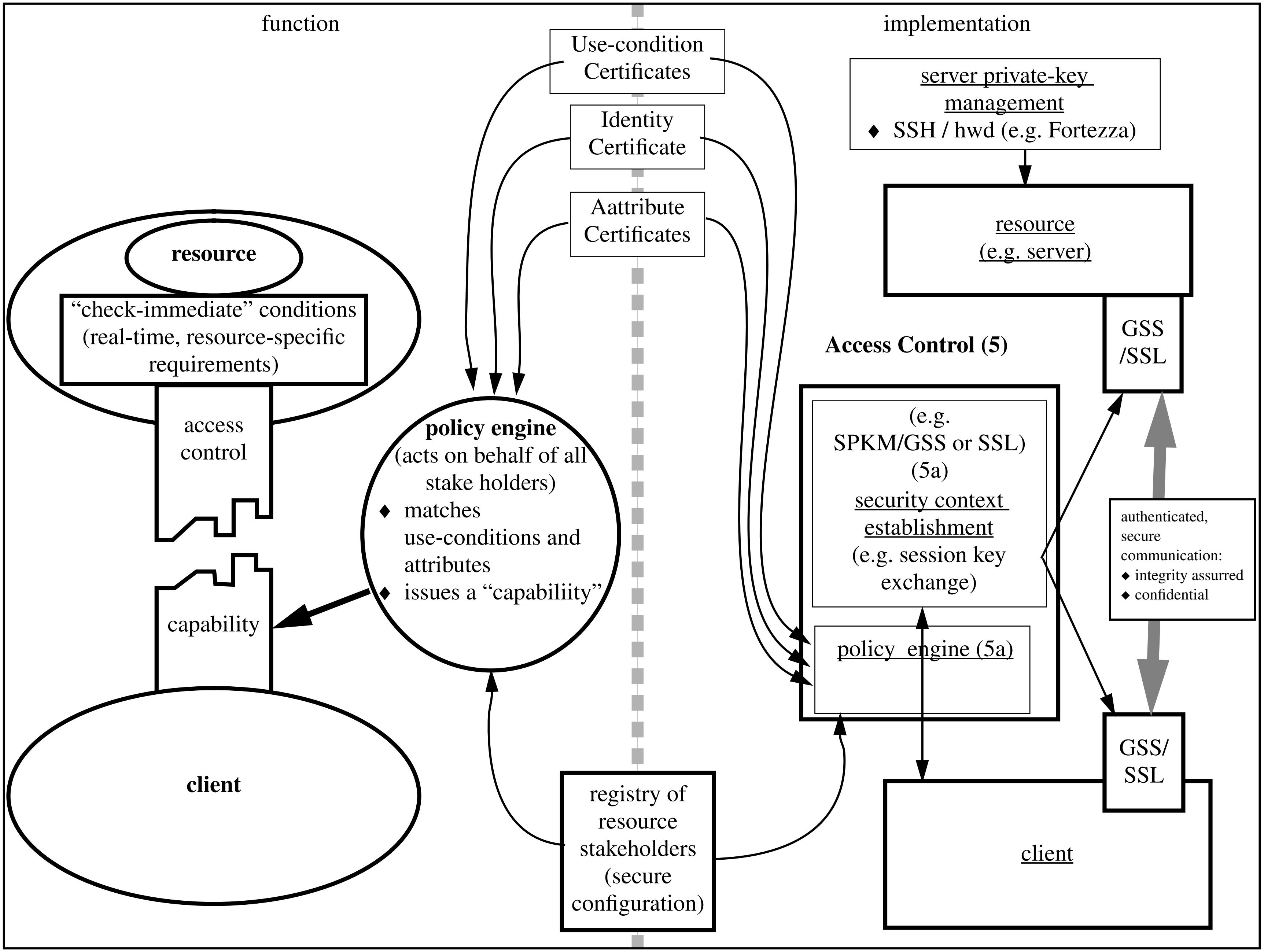


Figure 2 **Use-Condition Security Architecture Implementation**

Lawrence Berkeley National Laboratory
WEJohnston@lbl.gov

Figure 3

Use-Condition Security Architecture Implementation



DRAFT

- For naive applications whose communication can be controlled, e.g. the X11 protocol, secure and authenticated exchanges can occur using the SSH secure communication proxy method.

4.1 The Generic Security Service

The GSS-API [Lin93] (Generic Security Service - IETF Common Authentication Technology working group) provides a simple application interface to send integrity-protected or confidential communication. See Table 2.

Category	Functions
credential management	acquire_cred, inquire_cred, release_cred
context establishment	init_sec_context, accept_sec_context
transmit integrity-protected data	getmic, verifymic
transmit confidentiality protected data	wrap, unwrap

Table 2 A sampling of GSS-API functions. The function names are typically prefixed by "gss_".

The GSS-API interface has been implemented for many security environments, such as Kerberos [Lin96], OSF DCE, SESAME [SES]. The SPKM protocol is a public-key based protocol that provides a GSS-API interface. We have implemented the SPKM protocol and are aware of at least two other implementations: SecuDe [GMD] and Entrust/Toolkit [Ent]. For a good discussion of the rationale of SPKM, and its relationship to the Kerberos GSS-API, see [Ada96].

The SSH design provides convenient and transparent management of the private-keys of the user. We have extended the SSH agent to enable the setup of the SPKM security context.

SSH makes the user's private-key available to the SPKM protocol to establish the GSS security context. The private-key is used to authenticate the user to the target (e.g. access control agent) and transmit a session key to the target. Once this is done, the GSS-API provides an application interface to send validated and/or confidential communication as summarized in Table 2.

To establish a security context with the SPKM protocol, the RSA algorithm is used for two different purposes. The first use is by the initiator to prove its identity to the target through digital signature. The second use is by the target to decrypt a session key that is passed to it through public-key encryption. Whenever it is possible, we perform RSA operations involving the private-key within the SSH agent. This protects the private-key from being revealed.

We have extended the SSH agent with functions that perform signature and decryption. Although we follow the RSA recommended method, PKCS #1, of doing signature and decryption with the private-key, there are noted weaknesses in PKCS #1 [JM96], and the implementor of SSH expresses reservations about exporting functions that could possibly leak the private-key.

While user-initiated applications take advantage of the agent for private-key use, the long running, highly available services cannot make use of the agent. Since these services must be restarted without human intervention, such as after a reboot or a power outage, we store the private-key on disk in unencrypted form. (This is clearly not a good idea and a possible solution is to use a "smart card" that can make the private key available after reauthenticating the server based on a hardware characteristic like a CPU id that has a corresponding public key in the organization CA database.) Other services that need to be secure (at the cost of being available) can be started manually, under the agent, through a user's login shell.

A weakness of SSH is that the login environment currently does not use (third party signed) certificates to verify the authenticity of either user or server public-keys, even though the SPKM protocol does. The next version of SSH should be able to make use of such public-key certificates. We envision having a CA sign the SSH generated public-keys that are transmitted to it through a certificate request mechanism, such as PKCS #10 [RSA93]. (That is, the CA does not generate private-public key pairs, but just verifies the identity of the holder of these keys.)

For the search and retrieval of public-key identity certificates, we are examining both the LDAP and WWW search engine approaches mentioned above.

We currently use an ad-hoc encoding of use-conditions that allow searching by text-based search engines via WWW (see Figure 4) , but are examining other approaches like ADAGE [HMSZ96],

```
attribute-name: group
attribute-value:  1230259968 (0x49544700) "ITG"
key-name: /C=US/SP=California/O=Lawrence Berkeley National
Laboratory/OU=ICSD/CN=Case Larsen2/Email=ctlarsen@lbl.gov
issuer-key-name: /C=US/SP=California/O=Lawrence Berkeley National
Laboratory/OU=ICSD/CN=ITG-CA/Email=itg-ca@lbl.gov
-----BEGIN attrcert
MIIC1zCCAkYDBwBncm91cAADBQBJVEcAMIGaMAsGCSqGSIb3DQEBAQOBigAwgYYC
gYBcmvnmvLqIztgXhQI/rGcpvBlGQVgM2TjIBXOz/rgLkXFnTSEILtQQr/62pRkc
DzGmwnnhpUNODalclAQTWqNH04i17UwDVZQhjGBY+qeMIxcPaGTcKlk0V9KGedSb
...
-----END attrcert
```

Figure 4 Combined text and binary representation of an attribute certificate

Policymaker [BFL96], or PICS [Wor96] noted above. To be able to support automatic brokering, we require the ability to collect, parse, and automatically match use-conditions with credentials, which in turn requires an encoding which is agreed upon by all parties. This is future work.

4.2 Prototype Applications

We are using the prototype implementation to provide security for an experimental health care information application (see [KLP95]) and for the Distributed-Parallel Storage System ([LBN96]).

Distributed-Parallel Storage System: A High-Speed, Network Distributed Cache

At the application level, the DPSS is a semi-persistent cache of named data-objects, and at the storage level it is a logical block server. The system is usually used with an application agent library called a "data set structure access method". This component provides an object-like encapsulation of the data, in order to represent complex user-level data structures so that the application does not have to retain this information for each different data set. The access method converts the application requests into logical block requests. These logical block requests are then sent to the DPSS Master which serves two functions, request and resource management. The Resource Manager maintains data set definitions, and the Request Manager is responsible for mapping the logical block requests to physical block requests. The overall data flow involves "third-party" transfers from the storage servers directly to the data-consuming application (a model used by most high performance storage systems). Thus, the application requests data, these requests are translated to physical block addresses (server name, disk number, and disk block), and the servers deliver data directly to the application. The Resource Manager also deals with interactions with the storage servers to determine available storage (a storage server is an independent entity and may deal with several DPSS Masters).

The security model for the DPSS involves accommodating several different resource owners. The security context established between the Data Set Manager (DSM) (see Figure 4) and the disk/storage

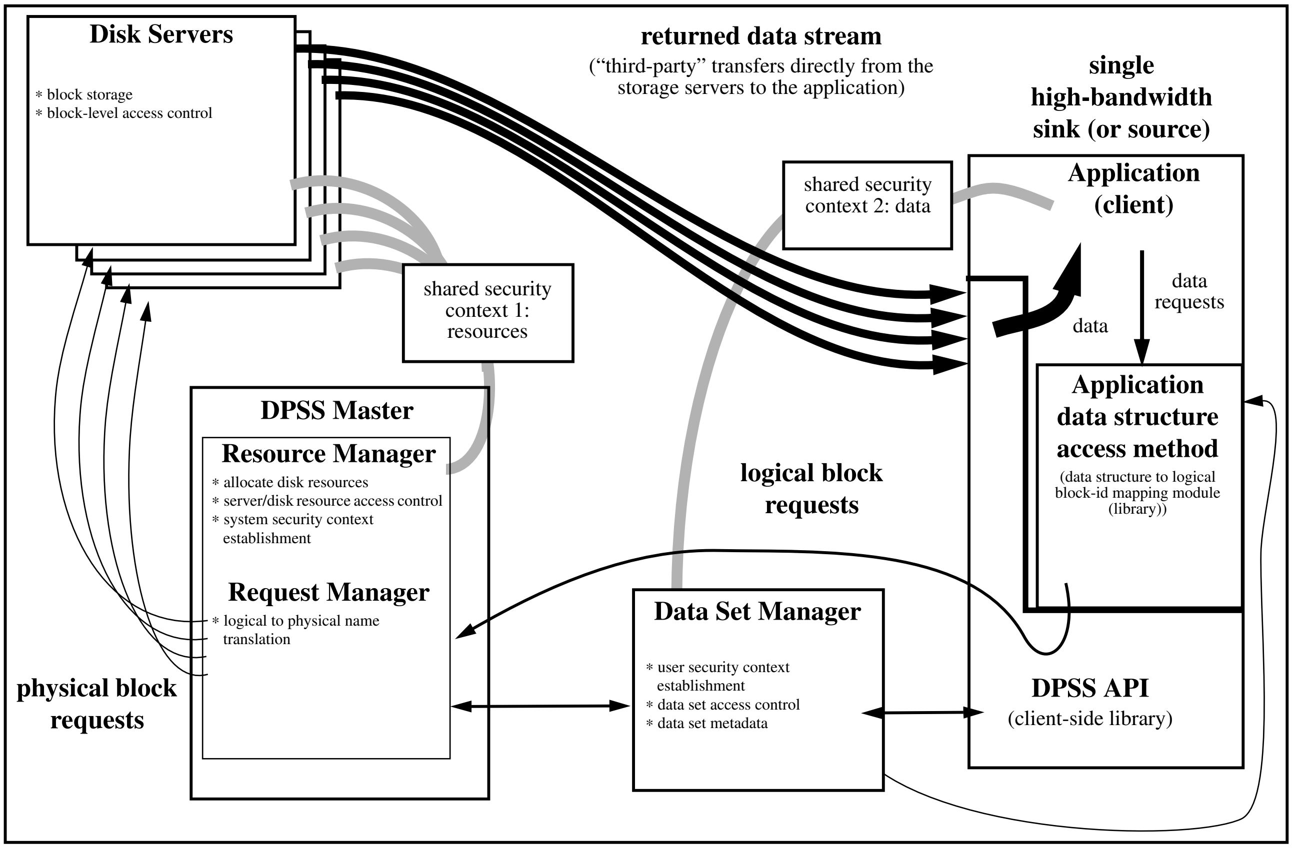


Figure 4 **Distributed-Parallel Storage System Architecture**

servers reflects agreements between the owners of physical resources (disks) and an agent that is providing storage to a user community. This context enforces the disk usage agreements. A separate context established between the DSM and the users reflects the use-conditions imposed by the data "owner", and provides for ensuring access control that enforces those use-conditions.

A Health Care Image Management System

An example of a health care application that uses the DPSS is a system that provides for collection, storage, cataloguing, and playback of video-angiography[5] images using a shared metropolitan area ATM network.

The image data are sent through the network to storage and analysis systems, as well as directly to users at clinic sites. Thus, data can be stored and catalogued for later use, data can be delivered live from the imaging device to remote clinics in real-time, or these data flows can all be done simultaneously. Whether the storage servers are local or distributed around the network is entirely a function of the optimal logistics.

5. Cardio-angiography imaging involves a two plane, X-ray video imaging system that produces from several to tens of minutes of digital video sequences for each patient study for each patient session. The digital video is organized as tens of data-objects, each of which are of the order of 100 MBytes.

Figure 5 illustrates the configuration of the health care application as it is embedded in a Pacific Bell

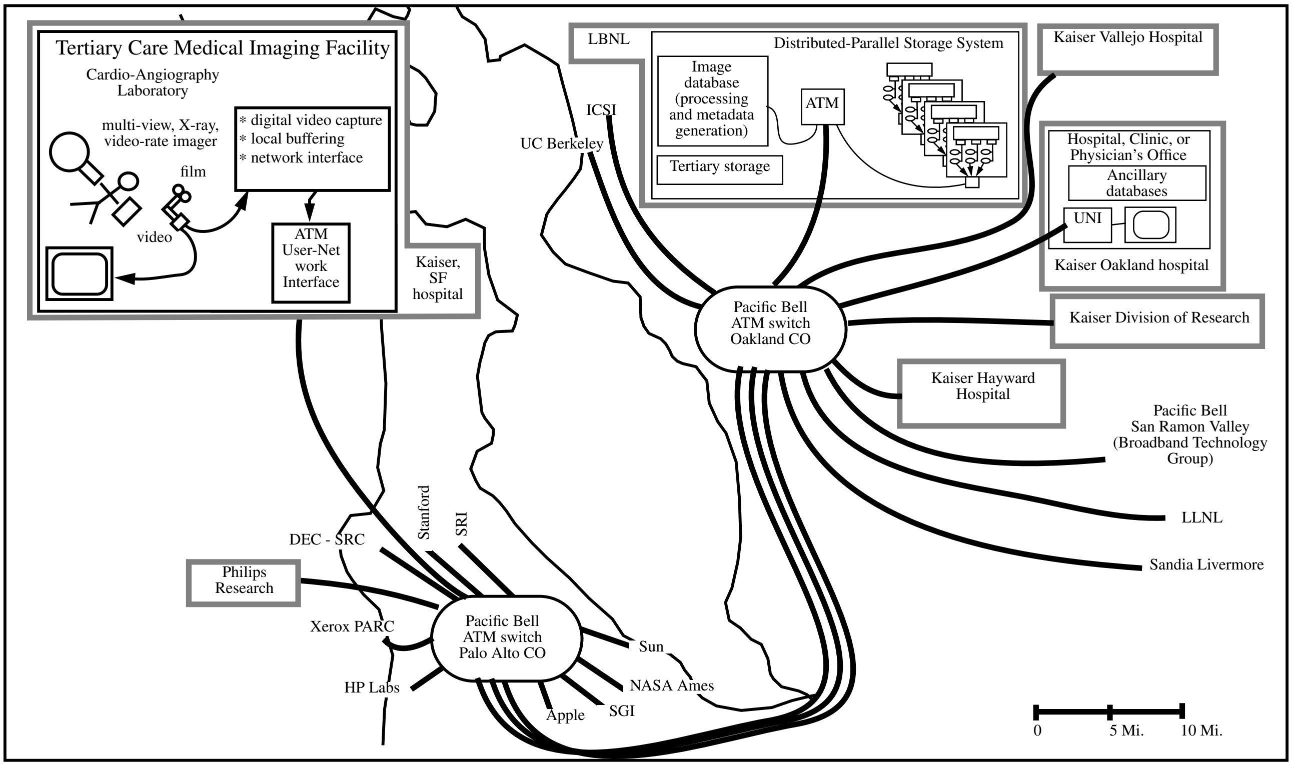


Figure 5 **Distributed Large Data-Object Medical Imaging Application Embedded in the Pacific Bell, S. F. Bay Area, Metropolitan Area ATM Network (BAGNet and Kaiser CalREN Sites Shown)**

ATM network. This application is a joint project of Lawrence Berkeley National Laboratory, Kaiser Permanente, Philips Research (Palo Alto), and the Pacific Bell CalREN program. (See [KLP95].)

The angiography data is collected directly from a Philips scanner by a computer system at the San Francisco Kaiser Cardiac Catheterization Laboratory. This computing system, in turn, is attached to the metropolitan area ATM network. When the data collection for a patient is complete (about once every 20-40 minutes), about 500 MBy of digital video data is sent across the network to LBNL (in Berkeley) and stored first on the DPSS, and then archived to a mass storage system. This process goes on 8-10 hours a day.

While still on the DPSS the data is catalogued in a World Wide Web based image database system [TJ96]. Use of the data involves access through ImgLib to locate the data sets of interest, and then to invoke a viewing application. Department-level Web-based patient databases can refer directly to the data in ImgLib without duplicating the data, or being concerned about tertiary storage management (which is handled by ImgLib).

The data-object viewing application ("Vplayer") is invoked via a WWW image database frontend that uses WWW security to protect the metadata. The application (a specialized, combined video and image browser) then accesses data stored on the DPSS. The Vplayer access to DPSS data is protected by the security architecture described here. (That is, credentialing is handled by the secure shell that runs Vplayer, and the DPSS access is controlled by checking authorization certificates presented by Vplayer.)

5.0 Relationship to Other Work

Our general approach is an application of public-key certificates, and as such it draws ideas from an active community of researchers.

The basic concepts of distributed systems security are laid out in various ISO (International Organization for Standardization) and ECMA (a Europe based international association for the standardization of information and communication systems) documents (see [SIR]). SESAME is an implementation that incorporates many of the concepts in the ECMA standards [ECM94] and [ECM89]. (See [SES].)

We have, however, taken a somewhat different direction than the ECMA and SESAME approaches with respect to attribute certificates, which are central to our approach. In the ECMA approach, most of the emphasis on the use of attribute certificates seems to be in the form of access control lists and roles (party A, who controls a resource, grants to party B the right to use that resource, perhaps under restricted conditions (roles)). Our approach treats use-conditions, rather than ACLs, as the fundamental concept. That is, we see resources primarily being protected by requirements on the user (e.g. level of training) rather than identity. (ACLs are, of course, easily implemented in this model.)

Our implementation is based on some refinements to the “secure shell” [Ylo], combined with the “Generic Security Service Application Program Interface” [Lin93] to provide an easily used application security mechanism.

Although evolved independently, the idea of CAs dealing only with identity certificates and resource principals signing use-condition certificates, is very similar to the generalized certificate ideas described by Carl Ellison [Ell96].

Our use-conditions are also similar to work being done in ANSI X9.45:

> *X9.45 defines enhanced management controls using attribute certificates. The digital signature control models put forward thus far do not contain sufficient security or authorization controls to offer a high-value non-repudiation service for either wholesale financial or other large scale commercial applications. This standard defines strategies for reducing the risks associated with digital signature systems. Much of this builds on the use of public key certificates and attribute certificates, as defined in X9.30 Part 3 and X9.31 Part 3. Attribute certificates are used to convey authorizations and restrictions that inform verifiers when an entity's signature would be considered valid, i.e., when the signature authorizes a document or transaction. These circumstances might include requirements, for example, that the monetary value be less than or equal to a specified dollar amount, or that another entity “cosign” the document. This standard will also define mechanisms for electronic timestamping and notarization of documents, as well as delegation of authority from one user to another. [Ank]*

6.0 Conclusions

The combination of direct reference to the principals responsible for the generation and assurance of information (the resource owners or policy makers), and standardization of representations of the information that permit automated verification, should form the basis of an easily used, scalable, and general purpose authentication and authorization system that provides such services as qualifying remote experimenters to use instrumentation systems in distributed collaboratories, as well as forming the basis for dynamically configured, large-scale, distributed computing resources.

To address deployability, we have evolved an implementation design that manages certificates on behalf of the users, should provide "strong" security for applications that use the security architecture directly, and should provide "good" security for as many unmodified applications as possible.

7.0 Notes and References

Ada96 Carlisle Adams. IDUP and SPKM: Developing public-key-based APIs and mechanisms for communication security services. In *Proceedings of the Symposium on Network and Distributed Systems Security (SNDSS'96)*. ISOC, 1996. Available at http://bilbo.isu.edu/sndss/adams.ps. See also http://bilbo.isu.edu/sndss/sndss96.html.

Ank Richard Ankney. Introduction to cryptographic standards. Technical report, IEEE. Available at http://www.itd.nrl.navy.mil/ITD/5540/ieee/cipher/cipher-crypto-stds.html.

APKI "Architecture for Public-Key Infrastructure," B. Blakley and the APKI Working Group.

This document describes Requirements and an Architecture for Public-Key Infrastructure components, identifies which elements of the architecture should (in the opinion of the authors) be standardized, and identifies candidate interface and protocol specifications which might serve as base documents for the standardization effort.

Available at http://www-itg.lbl.gov/SecurityArch/Reference.docs .

BFL96 Matt Blaze, Joan Feigenbaum, and Jack Lacy. Decentralized trust management. In *Proceedings of the Symposium on Research in Security and Privacy*. IEEE, 1996. Available at ftp://research.att.com/dist/mab/policymaker.ps.

Denning96 "Protection and Defense of Intrusion,' D. Denning. National Security in the Information Age conference at the US Air Force Academy, Colorado Springs, February 28 - March 1, 1996. Available at http://guru.cosc.georgetown.edu/~denning/infosec/USAFA.html

ECM89 ECMA. ECMA-138: Security in open systems - data elements and service definitions, 1st edition. Technical report, European Computer Manufacturers, Dec 1989. Available at http://www.ecma.ch/ecma-138.HTM.

ECM94 ECMA. ECMA-219: Authentication and privilege attribute security application with related key distribution functions, 1st edition. Technical report, European Computer Manufacturers, Dec 1994. Available at http://www.ecma.ch/ecma-219.HTM.

Ell96 Carl M. Ellison. Generalized certificates, 1996. Available at http://www.clark.net/pub/cme/html/cert.html.

Ent Entrust home page. Available at http://www.nortel.com/entprods/entrust/main.html.

For95 Warwick Ford. *Principles, Standards, Protocols, and Techniques*. Prentice-Hall, Englewood Cliffs, New Jersey, 07632, 1995.

GASD96 D. Gary, J. M. Anderson, F. Sudia, and K. Daguio. Information security - transforming the global marketplace (a panel discussion). In *Proceedings of the National Information Systems Security Conference*. NIST, Oct 1996. Available at http://csrc.nist.gov/nissc/1996. The panelists included security and technology officers of major information industry firms, including

Booz-Allen & Hamilton (international management and technology consulting), Anderson, Morgan Stanley (brokerage and financial services), F. Sudia, CertCo / Bankers Trust (global financial services) and the American Bankers Association. Their basic points were that technology has progressed to the point where we are moving from technology driven solutions to solution driven technology, and that security based on PKI, and authorization and attribute certificates, will provide the foundation of electronic commerce - not just protection: electronic contracts, financial instruments, electronic notary, and authority delegation (e.g. setting the scope of business activity of an on-line trader in the global marketplace).

GMD Secude home page. Available at http://www.darmstadt.gmd.de/secude/.

HMSZ96 Marty Hurley, Nimsha Meta, Rich Simon, and Mary Ellen Zurko. Authorization for distributed applications and groups. Technical report, OSF, 1996. Available at http://www.osf.org/www/adage/index.html.

ImgLib "LBNL Image Library," M. Thompson, W. Johnston, et al. http://www-itg.lbl.gov/ImgLib.project/homepage.html

JA95 William Johnston and Deb Agarwal. The virtual laboratory: Using networks to enable widely distributed collaboratory science, 1995. Available at http://www-itg.lbl/gov/~johnston/Virtual.Labs.html. A NSF Workshop Virtual Laboratory whitepaper.

JM96 Don B. Johnson and Stephen M. Matyas. Asymmetric encryption: Evolution and enhancements. *CryptoBytes*, 2(1):1–6, 1996. Available at http://www.rsa.com/PUBS/cryp_s~1.pdf.

KLP95 Kaiser, LBNL, and Philips. The Kaiser - LBNL - Philips CalREN project, 1995. Available at http://www-itg.lbl.gov/Kaiser/LKP/homepage.html.

LBNL96 LBNL. The Distributed-Parallel Storage System (DPSS) home page, 1996. Available at http://www-itg.lbl.gov/DPSS.

LDAP96 Lightweight Directory Access Protocol, 1996. Available at http://www.umich.edu/~rsug/ldap/ldap.html.

MH95 Mendez and Huitema. A new approach to the x.509 framework: Allowing a global authentication infrastructure without a global trust model. In *Proceedings of the Symposium on Network and Distributed System Security*. IEEE, Feb 1995.

NCS1 "Internet/Public Switched Network (PSN) Interconnectivity and Vulnerability Report," Customer Service and Information Assurance Division, National Communications System. Available at http://www.ncs.gov/n5_hp/html .

NCS2 "The Electronic Intrusion Threat to National Security and Emergency Preparedness Telecommunications: An Awareness Document," Information Assurance Branch, National Communications System. Available at http://www.ncs.gov/n5_hp/html .

Netscape1 "Securing Communications on the Intranet and over the Internet," T. Elgamal, J. Treuhaft. F. Chen, Netscape Communications Co. http://www.netscape.com/newsref/ref/128bit.html

PKIX “Public-Key Infrastructure (X.509) Working Group.” http://www.ietf.cnri.reston.va.us/html.charters/pkix-charter.html

RFC1422 Steve Kent. RFC-1422: Privacy enhancement for internet electronic mail: Part II: Certificate-based key management, Feb 1993. Available at http://ds.internic.net/rfc/rfc1422.txt.

RFC1508 John Linn. Generic Security Service Application Program Interface, Sep 1993. Available at http://ds.internic.net/rfc/rfc1508.txt. Also see more recent and related drafts at the IETF Common Authentication Technology home page (http://www.ietf.cnri.reston.va.us/html.charters/cat-charter.html) and at http://www.ietf.cnri.reston.va.us/ids.by.wg/cat.html.

RFC1964 John Linn. The Kerberos version 5 GSS-API mechanism, June 1996. Available at ftp://ds.internic.net/rfc/rfc1964.txt.

RSA93 RSA. PKCS #10, 1993. Available at ftp://ftp.rsa.com/pub/pkcs/ascii/pkcs-10.asc. This describes a syntax for public-key certification requests.

RSA96 RSA. RSA labs’ Frequently Asked Questions about todays’ cryptography v3.0, 1996. Available at http://www.rsa.com/rsalabs/newfaq/.

SDSI “SDSI---A Simple Distributed Security Infrastructure,” R. Rivest and B. Lampson. http://theory.lcs.mit.edu/~rivest/publications.html

SES SESAME - a Secure European System for Applications in a Multi-vendor Environment. Available at http://www.esat.kuleuven.ac.be.fi./cosic/sesame3.html. SESAME (a Secure European System for Applications in a Multi-vendor Environment) is a European research and development project, part funded by the European Commission under its RACE programme. It is also the name of the technology that came out of that project.The SESAME technology offers sophisticated single sign-on with added distributed access control features and cryptographic protection of interchanged data. SESAME is a construction kit. It is a set of security infrastructure components for product developers. It provides the underlying bedrock upon which full managed single sign-on products can be built.

SIR SIRENE. Security in computer networks. Available at http://www.zurich.ibm.com/pub/sti/www/g-kk/sirene/index.html. See also http://www.zurich.ibm.com/pub/sti/www/g-kk/sirene/pointers.html.

SPKI Simple public-key infrastructure (IETF working group). Send “subscribe spki” to majordomo@c2.org to join. Another public-key infrastructure working group has started in the IETF (SPKI). This group has formed to explore public-key certificates that are attribute-based instead of name-based, for use in a variety of Internet applications.

SSL “The SSL Protocol” http://live.netscape.com/newsref/std/SSL.html

SSLeay T. J. Hudson and E. Young. Available at http://www.psy.uq.edu.au~ftp/Crypto. SSLeay is a free implementation of Netscape’s Secure Socket Layer - the software encryption protocol behind the Netscape Secure Server and the Netscape Navigator Browser.

TJ96 Mary Thompson and William Johnston. LBNL image library, 1996. Available at http://www-itg.lbl.gov/ImgLib/ImgLib_intro.html.

Waldo See www-itg.lbl.gov/WALDO

Wor96 World Wide Web Consortium. Platform for internet content selection, 1996. Available at http://www.w3.org/pub/WWW/PICS.

Ylo Tatu Ylonen. The SSH (Secure Shell) remote login protocol. Available at ftp://ietf.cnri.reston.va.us/internet-drafts/draft-ylonen-ssh-protocol-00.tx% t. See also “SSH (Secure Shell) Remote Login Program” (http://www.cs.hut.fi/ssh/). IETF link doesn’t work.