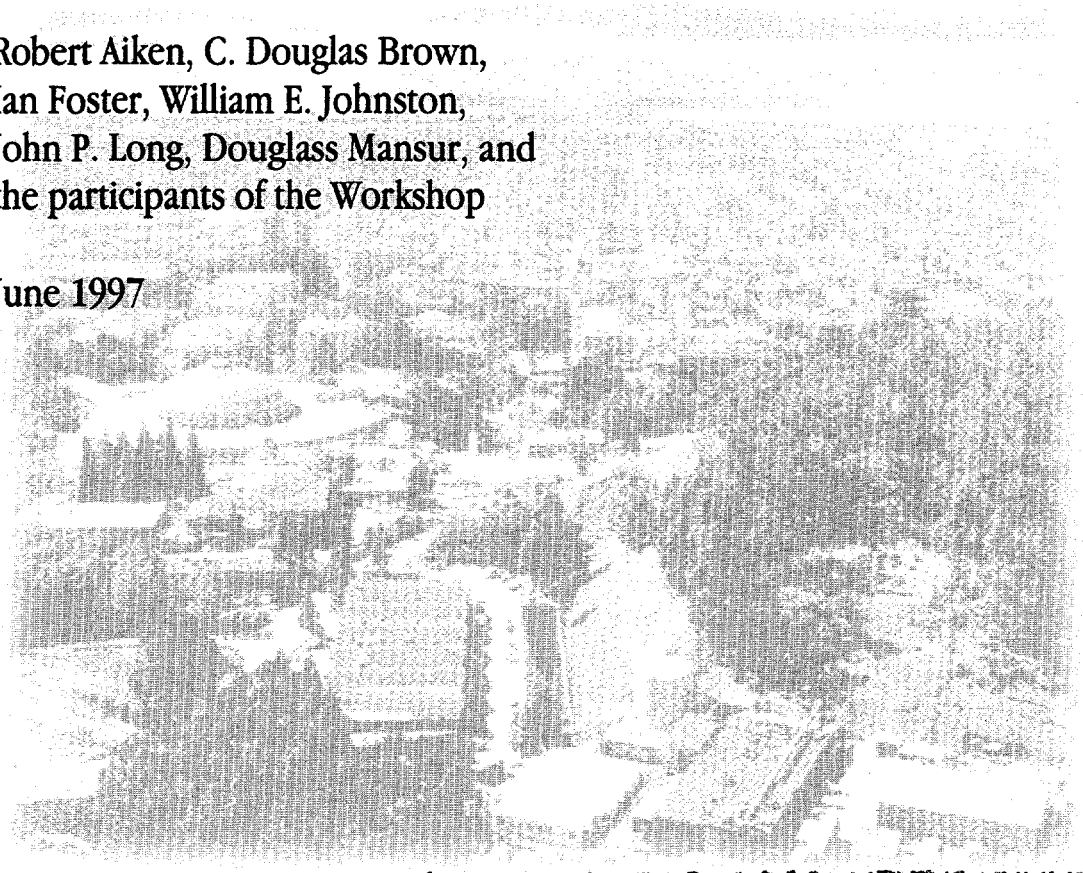
LBNL-40028
UC-405

ERNEST ORLANDO LAWRENCE
BERKELEY NATIONAL LABORATORY

Public Key Infrastructure for DOE Security Research: *Findings from the U.S. Department of Energy, Joint Energy Research/Defense Programs Computing-Related Security Research Requirements, Workshop II*

Robert Aiken, C. Douglas Brown,
Ian Foster, William E. Johnston,
John P. Long, Douglass Mansur, and
the participants of the Workshop

June 1997



LBNL-40028
UC-405

Public Key Infrastructure for DOE Security Research[1,2]

Findings from "U. S. Department of Energy, Joint Energy Research / Defense Programs Computing-Related Security Research Requirements, Workshop II" Dec 11–13, 1996, Albuquerque, New Mexico

Robert Aiken[3], C. Douglas Brown[4], Ian Foster[5], William E. Johnston[6], John P. Long[7], Douglass Mansur[8], and the participants of the Workshop

June, 1997

---

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 W-31-109-Eng-38 with Argonne National Laboratory and Contract DE-AC03-76SF00098 with the University of California. This document is report LBNL-40028, and is available at http://www-itg.lbl.gov/DOE_Security_Research .

2. For information contact William E. Johnston. U.S. mail address: Lawrence Berkeley National Laboratory, 1 Cyclotron Rd., MS: 50B-2239, Berkeley, CA 94720. Tel: +1-510-486-5014, fax: +1-510-486-6363, wejohnston@lbl.gov, http://www-itg.lbl.gov/~johnston

3. U. S. Department of Energy (aiken@oerhp01.er.doe.gov)

4. Sandia National Laboratories (cdbrown@sandia.gov)

5. Argonne National Laboratory (foster@mcs.anl.gov, http://www.mcs.anl.gov/people/foster/)

6. Lawrence Berkeley National Laboratory, Information and Computing Sciences Division (wejohnston@lbl.gov)

7. Sandia National Laboratories (jplong@sandia.gov)

8. Lawrence Livermore National Laboratory (mansur@llnl.gov)

Abstract

This document summarizes the Department of Energy's Second Joint Energy Research / Defence Programs Security Research Workshop. The workshop, built on the results of the first Joint Workshop which reviewed security requirements represented in a range of mission-critical ER and DP applications, discussed commonalties and differences in ER/DP requirements and approaches, and identified an integrated common set of security research priorities. One significant conclusion of the first workshop was that progress in a broad spectrum of DOE-relevant security problems and applications could best be addressed through public-key cryptography based systems, and therefore depended upon the existence of a robust, broadly deployed public-key infrastructure. Hence, public-key infrastructure ("PKI") was adopted as a primary focus for the second workshop.

The Second Joint Workshop covered a range of DOE security research and deployment efforts, as well as summaries of the state of the art in various areas relating to public-key technologies. Key findings were that a broad range of DOE applications can benefit from security architectures and technologies built on a robust, flexible, widely deployed public-key infrastructure; that there exists a collection of specific requirements for missing or undeveloped PKI functionality, together with a preliminary assessment of how these requirements can be met; that, while commercial developments can be expected to provide many relevant security technologies, there are important capabilities that commercial developments will not address, due to the unique scale, performance, diversity, distributed nature, and sensitivity of DOE applications; that DOE should encourage and support research activities intended to increase understanding of security technology requirements, and to develop critical components not forthcoming from other sources in a timely manner, and; that testbeds that enable the deployment of security architectures and protocols in realistic environments are an important component of the research efforts, since only in open, heterogeneous testbeds are certain weaknesses and interoperability problems likely to be exposed and addressed.

Contents

Executive Summary

This document summarizes the findings of the Department of Energy (DOE)'s 2nd Joint ER/DP Security Research Workshop, held in Albuquerque in December 1996. The workshop built on the results of the first Joint Workshop, held in Chicago in November 1995, which reviewed security requirements represented in a range of mission-critical DP and ER applications, discussed commonalties and differences in ER/DP requirements and approaches, and identified an integrated common set of security research priorities. (See http://www-itg.lbl.gov/DOE_Security_Research for a summary of the first workshop's findings.) One significant conclusion of the first workshop was that progress in a broad spectrum of DOE-relevant security problems and applications could best be addressed through public-key cryptography based systems, and therefore depended upon the existence of a robust, broadly deployed public-key infrastructure. Hence, public-key infrastructure ("PKI") was adopted as a primary focus for the second meeting.

The two and a half day Second Joint Workshop brought together 35 security researchers and practitioners from Defense Programs (DP) and Energy Research (ER) laboratories. The workshop comprised a mixture of presentations and discussion. Presentations covered a range of DOE security research and deployment efforts, as well as summaries of the state of the art in various areas relating to public-key technologies.

Goals:

Workshop participants were charged with:

- Clarifying DOE requirements for public-key infrastructure,
- Reviewing approaches to meeting those requirements,
- Identifying "problem areas" in which commercial developments would not obviously meet DOE requirements, and
- Identifying concrete, short-term steps that could be taken toward building security infrastructures and testbeds to support research and development on the identified problem areas.

Findings:

The principal findings of the workshop are contained in this report, and can be summarized as follows:

- A broad range of DOE applications can benefit from security architectures and technologies built on a robust, flexible, widely deployed public-key infrastructure, i.e., one that extends beyond the DOE and government scope and is an integral part of the PKI used by academia and industry.
- A collection of specific requirements for missing or undeveloped PKI functionality were identified, together with a preliminary assessment of how these requirements can be met (see Section 2.1).

- While commercial developments can be expected to provide many relevant security technologies, there are important capabilities that commercial developments will not address, due to the unique scale, performance, diversity, distributed nature, and sensitivity of DOE applications.
- Hence, DOE should encourage and support research activities intended to increase understanding of security technology requirements as well as support applications and to develop critical components not forthcoming from other sources in a timely manner.
- Testbeds are an important component of these research efforts. Testbed activities enable the deployment of security architectures and protocols in realistic environments so that we can evaluate and make progress on the research issues. Only in open, heterogeneous testbeds are certain weaknesses and interoperability problems likely to be exposed and addressed.

In addition, as in the First Joint Workshop, there was a general feeling that the meeting provided an invaluable opportunity for ER and DP researchers and practitioners to discuss common interests, and hence should be continued on an annual basis to ensure an integrated ER and DP infrastructure and supportive security research. The ER MICS Division support for these workshops has proven invaluable in ensuring their success.

The rest of this report is divided into four sections. Section 1 provides background material on public-key technology. Section 2 summarizes DOE-specific challenges in the area of public-key infrastructure, as identified in the workshop. Section 3 lists "testbeds" proposed at the workshop. Finally, three appendices provide the workshop program, participant list, and a PKI technology introduction.

1.0 Public-Key Infrastructure and DOE Applications

1.1 The Problem

Multiple DOE-critical applications must protect against unauthorized access and/or assure privacy of proprietary, confidential, or otherwise restricted data. For example:

- Access to remote instrumentation resources via open network, whether for monitoring, control of experiments, or collection and manipulation of data.
- Computing, when the data, computing elements, and users are scattered all over the world.
- Fine-grained access control to high-value data bases.
- Normal DOE and Laboratory electronic financial and business communications.

In each of these applications, authentication and access control mechanisms are required, and indeed are essential to safe and secure operations. Awkward, intrusive, and/or expensive security mechanisms will delay the wide-spread use of these applications in open environments. Hence, we require security techniques, tools, and architectures that are flexible and effective (to provide required capabilities in a manageable, cost effective, and usable manner) and that are easily deployed, administered, and used. Furthermore, these architectures must automate all aspects of the security process once certificates are defined, to provide for generalized and transparent access control. (We use the term *certificate* in this report to refer to "documents" used for authentication and authorization, and whose integrity is assured through cryptographic techniques. These documents are also known as "public-key certificates," "digitally signed certificates," or simply "signed certificates.")

The overall goals of security architectures, then, are to encode, distribute, protect, and then act on, information that is needed to provide for the routine, secure availability of remote resources in a widely distributed environment, where the users, resources and stake-holders may all be in different places at different times.

This statement of the general problem encompasses many DOE applications, and some of these are described below.

1.2 Overall Approach: Public-key Certificates

The emerging approach to address the distributed environment problem described above makes use of the scalable and distributed characteristics of public-key certificate infrastructure ("PKI"), which obtains its name from the underlying technology of public-key cryptography.

Public-key cryptography has unique characteristics that make it invaluable as a basis for security functions in widely distributed environments. Briefly, this form of cryptography uses a pair of asymmetric keys with the property that what one encrypts with one key of the key-pair can only be decrypted with the other key of the key-pair, and visa versa. A user ("A") generates or is issued this key-pair, and one key is designated the "private-key"

— and must be kept secret by the user — and the other is the "public-key." The public-key is widely and openly distributed (some commercial organizations publish their public-keys in the New York Times). For example, anyone may send user A a confidential message by encrypting plain text using A's public key and then sending the encrypted message to user A. Only user A can decrypt this message since only she has the private key (of the key pair) necessary to properly decrypt the message.

The private-key is also used for authentication of user-A. A message that can be decrypted with a given public-key could only have been encrypted with the corresponding private-key. If that private-key has been kept secret by A, then only A could have sent the encrypted message. One important use of this characteristic is the digital or cryptographic "signing" of documents or messages. The purpose of the signing is to prove both that the message originated with A, and is un-altered from its original version. This is accomplished by producing a hash code (a short and unique code based on a mathematical transformation of the message) and then encrypting this hash with the private-key. This message digest is appended to the message (or sent separately). The receiver decrypts the hash with user-A's public-key, recomputes the hash of the message, and compares the two hash codes. If they match, then the received message must be identical (bit-for-bit) with the one that user-A generated originally. A variation of this procedure is used in authentication exchanges for functions such as remote login and remote shell.

Public-key identity certificates associate a public-key with a (typically user or system) name. A public-key identity certificate is a document that contains a name, say of A, together with A's public-key. This document is then cryptographically "signed" by a "trusted" third-party. The purpose of the third-party is usually to attest to a relationship between the name in the certificate (e.g., that it belongs to a "real" person of the "same" name). This function is called a "certificate authority." There are different forms of certificates, including those supporting the authorization to do something, delegation of authority, etc. (See "Appendix 3: Public-key Certificates — Background" for more introduction on PKI technology.)

Hence, the term *Public-key Infrastructure* refers to the combination of four components: (a) public-key cryptography, (b) digitally signed documents, (c) trusted third-parties that provide some guarantee about the relationship between names, public-keys, and "real" entities, and (d) the mechanisms for publishing and using the certificates.

The PKI approach as just described has emerged from a concurrent evolution of the information systems being protected and the security mechanisms protecting them. This evolution continues, and in fact PKI is now being extended beyond the classical notions of "security" and is being acknowledged as a critical component of distributed enterprises of all types ([GASD96] [John96a]). In general, we can say that PKI concerns not just security, but also addresses the more general problem of handling distributed information in ways that enable new capabilities (e.g., brokering resources for construction of distributed systems). From this perspective, it is perhaps not surprising that the concerns of the financial services industry are remarkably similar to those of DOE: technologies that are components of a security architecture for distributed environments (especially as related to contracts and conduct of business) are also essential components of electronic

commerce and globally distributed enterprise. Table 1 (abstracted from [GASD96]) summarizes the view of the financial community.

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)

In reasoning about security it is useful to decompose the situation into several levels of abstraction: security model, architecture, infrastructure, and operations.

1.2.1 Security Model

The security model addresses what is being protected, and how. A brief example of such a model is the following (taken from [John96a]):

> *Our general model addresses access control and data confidentiality for computer mediated resources. The resources have use-conditions specified by the responsible parties that are expressed as cryptographically signed documents. Potential users have matching attributes that are also attested to in cryptographically signed documents. Access control is then based on, first: producing a set of credentials that verify that the use-conditions have been satisfied; second, at the point and time of access to a resource, validating the credential presenter and performing any required real-time ("check-immediate") actions (such as revalidating certificates,*

payment-like exchanges, etc.).

From an abstract point of view, this model is taken directly from how human organizations deal with similar sorts of resources today.

The security model drives the design of the architecture.

1.2.2 Architecture

A security architecture specifies how various components fit together to form a system. These components may include key management, certificate management, security context establishment, user authentication, message authentication/confidentiality, application communications libraries, etc. These components must interoperate to provide the capabilities needed to protect the resources in question.

Significant aspects of architectures are also driven by operational requirements: who generates authorization certificates and access control lists, how computing systems administrative domains and application security domains interact (they are likely to have different policy concerns), etc. Architectures are also driven by the availability of components, current directions in the commercial and standards communities, etc.

A fairly general architecture that appears to meet the needs of scientific computing applications is the IETF Generic Security Service [GSS]. This architecture provides authenticated and confidential messaging between components of distributed systems, together with the collection of functions needed to establish security contexts: various certificate and private-key management, certificate generation, storage, location, and use techniques, tools, and APIs, as well as other tools needed to build a useful and deployable security system.

1.2.3 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 mechanisms for establishing identity and for distributing the cryptographic information needed to use that identity for user and message authentication. Additional mechanisms are then required for locating and certifying these certificates. The computing community is in the process of defining and deploying these infrastructure components. While there is agreement on some of the basics (e.g., X.509 certificates for identity), many aspects of this infrastructure remain without common agreement or implementation. For example, it is not obvious whether the Web will be used to store, locate, and retrieve various types of public-key certificates, or whether LDAP and/or X.500 will be the common certificate distribution mechanism, or whether and how all of these mechanisms will be used.

1.2.4 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.

1.3 Overview of the PKI Architecture

The Public-Key Infrastructure Working Group of the IETF Security Area ([PKIX]) has produced a comprehensive description of the various sub-functions and components of a general and open PKI ([APKI]) and are working on specification of detailed interface and data format specifications for all of the components and subcomponents. Implementation of this architecture and its open interfaces would go a long way toward promoting a uniform approach to security in a wide range of applications. However, it is not clear at this point that vendors will implement this architecture and its open and interoperable interfaces[9] unless pressured to do so.

The "Architecture for Public-Key Infrastructure" [APKI] document groups PKI components into the broad functional categories and relationships indicated below.

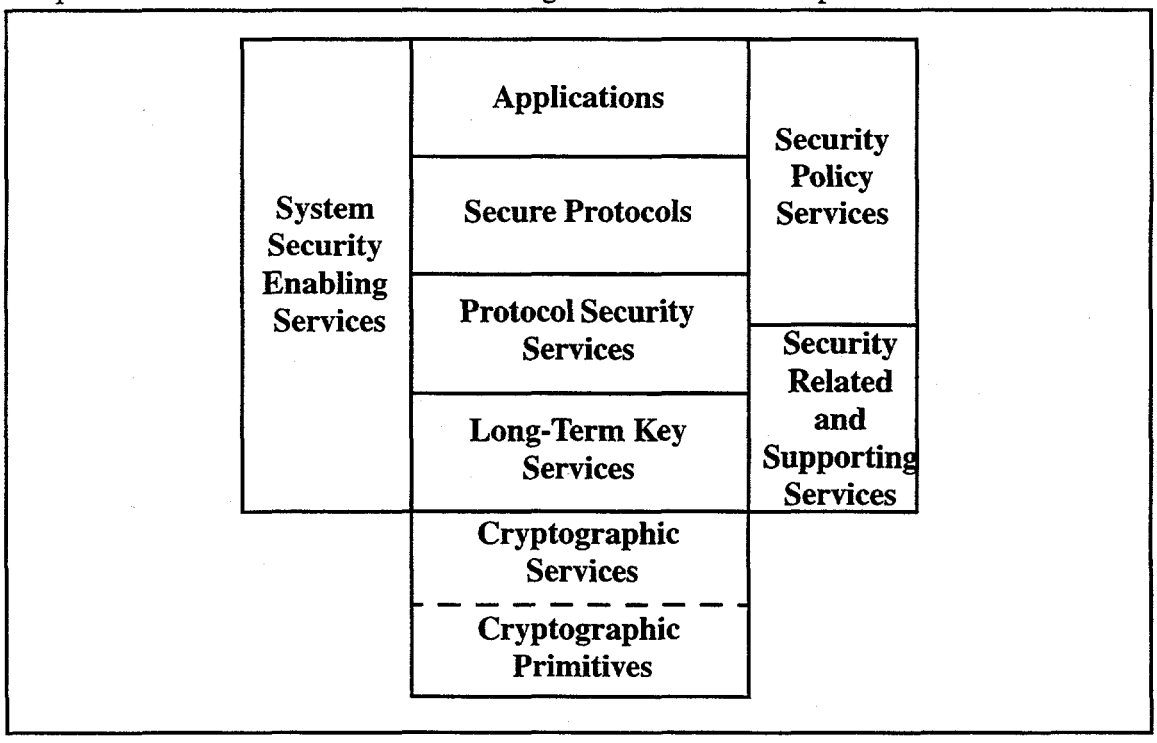


Figure 1 **PKI Architecture**

A brief description of the major PKI components, as defined by the APKI Working Group follows:

9. Open interfaces do not necessarily imply interoperable interfaces. There are many ways to inhibit interoperability, either deliberately or mistakenly. See [Lar96].

- Cryptographic Primitives and Services provide the cryptographic functions on which public-key security is based (including secret-key primitives such as DES). They provide access to low-level cryptographic primitives such as key generation, hashing application data to a buffer, encryption of a data buffer using secret-key or public-key algorithms, decryption of a data buffer using secret-key or public-key algorithms, *etc.*...

  The architecture's cryptographic primitives may be provided by hardware (e.g., smart-cards or cryptographic modules) or by software.

  Candidate interfaces for access to cryptographic primitives include The RSA BSafe library, The X/Open GCS-API and, The Microsoft CryptoAPI 1.0.

- Long-term Key Services permit users and other principals to manage their own long-term keys and certificates and to retrieve and check the validity of other principals' certificates. The components include:
  - Key Lifecycle Management provides key revocation, key repudiation, key expiration, and related services.
  - Key Escrow and Key Recovery provide for secure storage of keys for later recovery under policy control.
  - Virtual Smartcard Service (e.g., RSA's PKCS #11) enables users and other principals to:
    - store long-term personal security information (including private-keys, certificates, and other information) in protected storage,
    - activate personal keys for use via an authentication procedure,
    - use those keys for encryption, decryption, and signature activities.
  - Certificate Management enables users, administrators and other principals to request certification of public-keys and revocation of previously certified keys. It may optionally generate key pairs and provide key-pair recovery services. There are four sub-components:
    - Local Registration Authority provides interfaces for requesting generation of key-pairs and corresponding certificates, requesting certification of existing public-keys, and requesting revocation of existing certificates.
    - Certification Authority Agent (CA Agent) provides interfaces for certifying existing public-keys, generating and returning key pairs and corresponding certificates, revoking existing certificates. The CA Agent implements these interfaces via calls to a Certification Authority (CA).
    - Certification Authority certifies public-keys (returning the generated certificate) and generates certificate revocation lists. In some configurations these CA "singing" functions will be "off-line".
    - Publication Authority provides interfaces through which CAs and CA Agents can place certificates and Certificate Revocation Lists (CRLs) into public repositories or transmit them directly to requestors.
    - Public-Key Delivery and Verification allows a program to retrieve any principal's certificate, verify its validity, and extract the principal's certified public-key from the certificate.

- Protocol Security Services provide security functionality (data origin authentication, data integrity protection, data privacy protection, nonrepudiation) suitable for use by implementors of security-aware applications such as secure protocols. The components include:
  - Session-Oriented Protocol Security Services: The APKI preferred interface for these services is the IETF GSS-API.
  - Store & Forward Protocol Security Services and Non-Repudiation Services: The preferred interface for these services is IETF IDUP-GSS- API.
- Secure Protocols provide secure inter-application communications for security-unaware and "mildly" security-aware applications. Secure protocols provide protected data transfer between communicating partners without requiring any calls to security services. (I.e., applications operating in a security context established by someone else — e.g. SSH secure proxy ports, IP security, etc.) Examples of secure protocols include: Secure RPC, SSL, SHTTP, OMG SECIOP, and IPSec.
- System Security Enabling Services provide the functionality which allows a user's or other principal's identity to be established and associated with his actions in the system.
- System Functions (e.g., Operating System functions) are needed to support user logon, user credential acquisition, and association of security state information with user processes and threads. For example, once a user has acquired credentials by authenticating himself to a smartcard, that user's processes should be able to use the smartcard interface to sign data using a private-key stored on the smartcard. This will only be possible (and secure) if the system has maintained security state information associating the user's processes with the handle returned when the user authenticated himself to the smartcard. Examples include Pluggable Application Modules [PAM] and Secure Shell [SSH].
- Security Policy Services provide the policy-related information which must be carried in secure protocols to enable access control, and provide access-control checking facilities to security-aware applications which must enforce policy. Security Policy Services manage information about users' (and other principals') privileges and resource access control policies, and make access control decisions based on that information.
- Supporting Services provide functionality which is required for secure operation, but is not directly involved in security policy enforcement. Examples include security auditing services, secure time service, and various directory services.

1.4 DOE Example Applications

Two examples of DOE applications that are inherently distributed — in system architecture, use, and administration — and that therefore require a scalable security architecture, are remote collaboratories (distributed collaboration and remotely controlled instruments) and metacomputing (distributed supercomputing).

Security architectures supporting these applications require interoperating mechanisms for secure remote user access to system components (for administration) and for secure

interprocess communication (IPC). In the future, these applications will also require use and resource-owner certificate definition and management functions, as remotely defined authorization and attribute certificates replace access control lists as the user validation mechanism.

The secure IPC can be provided by several different mechanisms, all of which can use PKI for user authentication.

1.4.1 GSS-API

The GSS-API is a secure messaging standard that is used for data communication in distributed programs. (See [GSS] and [SPKM].) Programs that use GSS-API directly are "mildly security aware", in that they must specify a few simple parameters that specify whether, for example, messages are authenticated or encrypted. The application may, or may not, get involved with the establishment of the GSS security context (GSS initialization). GSS is intended to provide application security services independent of the underlying security architecture, and the SPKM version of GSS uses PKI.

GSS is used by the LBNL distributed storage systems [DPSS]) to provide independent security contexts for enforcing the relationship among the storage system components and resources (e.g. which servers and how much disk may be used by a particular instance of DPSS) as well as providing enforcement of data-owner imposed file access restrictions.

1.4.2 CORBA

CORBA is an IPC mechanism used by remote instrument control systems.

CORBA provides a high level "RPC" mechanism with function call-like semantics (as opposed to GSS, which is a low-level two-way messaging system). The CORBA Security Services draft standard [CORBASEC] defines a way of using GSS-API (and therefore PKI) to add authentication and authorization to the existing CORBA RPC mechanism. The GSS security functionality can be added to some CORBA implementations (depending on the modularity of the implementation — see, e.g. [Desai]).

1.4.3 Zipper

The scale and performance requirements of distributed supercomputing applications can place heavy demands on IPC mechanisms. These applications can span hundreds or even thousands of processors, located at multiple sites. They can require low latencies and high bandwidths, and often have little tolerance for overhead. Hence, specialized techniques can be required both for security context establishment and for secure IPC. Zipper [Zipper] is a secure communications library that provides these functions. Security context is performed by using authentication mechanisms provided by Globus [Globus]; high-performance transport is provided via security-enhanced versions of the Nexus and MPI communications libraries, based on multimethod communication mechanisms provided by Nexus. These libraries allow security mechanisms to be enabled selectively, and used over multiple low-level communication protocols and substrates.

Zipper does some of its own security context establishment, and also uses GSS.

1.4.4 SSH

Secure access to remote systems can be provided by utilities such as SSH [SSH] or by PAM-aware applications (see below). Both of these mechanisms provide secure remote log-in, copy, etc.; SSH provides in addition secure X-windows sessions (and a general secure port proxy mechanism). Both of these systems have security context establishment mechanisms that can (potentially) use PKI.

2.0 Public-Key Infrastructure Challenges for DOE Applications

Commercial organizations are developing public-key infrastructure products, and these products can be expected, in the future, to meet many DOE requirements. Indeed, there are significant areas of DOE interest (such as financial transactions) in which there is no reason to expect that commercial developments will not provide all required functionality.

At the same time, it would be a mistake to conclude that DOE can afford to wait for commercial vendors to meet all DOE needs. As emphasized at the First Joint Workshop, DOE applications in areas such as

- On-line instruments,
- Distributed computing,
- Collaborative environments, and
- Defense Programs Stockpile Stewardship and Management

often have unique characteristics in terms of scale, performance requirements, widely distributed nature, and high consequences. The development of adequate security solutions for these applications will require both new research and the integration of DOE-specific components with multiple existing commercial technologies in an integrated heterogeneous environment. Together, these lead to three significant challenges for the DOE community:

1) Specific DOE-unique requirements must be identified, and the research efforts required to address those requirements pursued.

2) The need for integration of both general and application-specific security architectures with commercial technologies means that *open interfaces* and *interoperability* become key concerns for PKI. Yet these issues have not been addressed by vendors: for example, almost no commercial PKI applications interpret certificates in a uniform way, and so can only use certificates issued by the PKI provided as part of that application environment; and none offer the private-key functions needed to support a uniform security architecture.

3) Testbeds and large-scale application experiments are needed to help identify DOE-unique requirements, evaluate potential solutions, verify interoperability, and test the effectiveness and strength of the resulting security.

In the rest of this section, we list PKI issues identified by workshop participants as particularly challenging, missing but potentially useful or necessary, or present in theory but not in practice. This list is certainly not complete, nor is it the case that only DOE will address these issues. However, each of these issues is currently an obstacle to DOE use of PKI and an obstacle to some classes of DOE applications. We conclude the section with a discussion of open interfaces and interoperability.

2.1 Identified Requirements

The Workshop discussions identified a range of required functionality, some of which may be expected to show up in commercial products as they are currently evolving, some of which will show up in commercial products if the vendors are "pushed" in the "right"

directions (e.g. with RFP requirements), and some of which we do not expect to show up in commercial products. The main topics are:

- OID-based Policy Statements in Certificates
- Archives for "Old" Keys
- Multi-keyed Certificate Authorities
- Security Gateways for Legacy Applications
- Authorization and Attribute Certificates
- Multiple Digital Signatures and Policy Engines
- Pluggable Authentication Modules
- Open Interfaces and Interoperable Independent PKI Components
- Firewalls and PKI Authentication
- Key-agile Environments
- Certificate Databases
- N-way Security Contexts
- Long-lived Channels and Autonomous Server Recovery
- DOE Participation in Standards work

These are discussed below.

2.1.1 OID-based Policy Statements in Certificates

Background:
X.509v3 certificates have the (potentially very useful) provision for supporting "standard" policy statements. These statements can allow construction of a "transitive trust chain," thus enabling the automation of cross certification authority (CA) operation once the policy has been agreed upon.

Rationale:
This automated cross-CA operation can be an important scaling issue. Practical CAs will almost certainly be at the organization (e.g., Laboratory) level, if not below, so the number will grow rapidly to the point where the manual exchange of CA public-keys needed for cross-CA use of certificates will be difficult to support. If policy statements can be standardized for various certificate uses (e.g., a level of identity surety sufficient for secure e-mail exchanges), then local CAs can issue certificates that just say something like: "any of the listed CAs with this OID policy statement can be trusted to issue personal identify certificates sufficient for a function X (e.g., secure e-mail)."

Issue:
No one has experience in defining and representing such standard policies, nor any experience with the intended and unintended results of such automated cross-CA operation.

2.1.2 Archives for "Old" Keys

Background:

Rationale:
When digital signatures are used to ensure the authenticity of a long-lived document (e.g., a purchasing contract) then it will be necessary to preserve both the document and the public-key of the signer in order to verify the signature validity in the future.

The same archive can provide some protection against compromised private (signing) keys if the archive itself is secure and un-modifiable. (If the time of the compromise can be identified, then a secure backup prior to that time will be known to contain valid signatures.)

Recovering previously encrypted data may also be a concern, so it may be necessary to archive encryption keys that have been used to encrypt data for storage.

Issue:
The many operational mechanisms needed for such archives have yet to be determined.

2.1.3 Multi-keyed Certificate Authorities

Background:
Certificate authorities ("CA") will be used for many different purposes: issuing key-pairs (adding new identities to the CA), signing certificates, archiving private-keys, delegating institutional authority, etc.

Rationale:
Most of these operations require either high availability of the signing function and/or high assurance of the signers' integrity. One way to address both of these issues is through the use of multi-part mechanisms (e.g., multi-part cryptographic keys) for releasing the CA private-key for signing. That is, any N out of M people can release the CA private-key to perform signing functions, and/or at least N out of M must participate to release the CA key in order to do key recovery (i.e., obtain a copy of a user's private-key).

Issue:
Neither the mechanisms nor the cryptographic algorithms to do this are widely practiced or understood.

2.1.4 Security Gateways for Legacy Applications

Background:
Most existing applications cannot be fully integrated with modern security architectures because they are too complex, lack sufficient justification, or the source code is not under the control of the DOE community.

Rationale:
It may still be possible to bring the protection of modern security to these applications through standard gateway architectures that act as security-aware proxies.

Issue:
Various legacy applications need to be examined, and gateways written and experimented with, in order to evaluate the value of this approach. Early work has indicated that there may be common architectures and/or common architectural elements for such application gateways which would ease the job of building the gateways. (See [John96b].)

2.1.5 Authorization and Attribute Certificates

Background:
While the various "standard" PKI functions like digital signatures, document origin authentication, secure communication via publicly available keys, etc., will be very useful, the full potential of PKI will be realized through authorization and delegation certificates. These are generalized public-key certificates that carry organizational authority delegation, functional authorization, and user attributes.

In this certificate-rich model (as in the financial services industry work described in [X9.45]) certificates will be used to encode securely a wide array of conditions, delegations, and characteristics, that are handled in a variety of ways in the enterprise today. For example: cost of services; limits on disk and CPU utilization in distributed systems; identification of corporate functions (e.g., person A is a purchasing agent); record-level data access rights (e.g., access to disciplinary records in the personnel database require signatures of two people with HR function); delegation of contract authorizations from CEO Head of the Purchasing Dept.; or "a level of safety training is required for access to this instrument."

The motivation of both the scientific and the financial communities to use such conditions and authorizations encoded in certificates is to automate the process of maintaining control and proper functioning of all aspects of the enterprise in a widely distributed environment where people increasingly do not necessarily experience face-to-face contact with colleagues and supervisors while doing their work. A similar motivation exists for control over information streams in the global network environment, as typified by the Web.

Rationale:
This use of P-K certificates will provide an important tool for global electronic commerce, and an important tool for enabling secure and highly distributed computing, collaboration, and remote control environments. By enabling this type of distributed enterprise, the PKI will reach its potential for benefiting distributed user communities, and provide an important justification for the effort involved in deploying PKI. See [John96a].

Issue:
Security architectures that support this certificate-rich environment need access to, and will much more widely distribute, all levels of the PKI, which is expected to be widely distributed. Certificate generation, management, acquisition, and validation are functions that, like the organization that uses them, will also be widely distributed. All of this depends on a flexible and open PKI, where all of the relevant interfaces are exposed for use by the security architecture, and interoperating components because the many different participating sites and groups will all have different suppliers of the PKI components.

2.1.6 Multiple Digital Signatures and Policy Engines

Background:
A deficiency of X.509-style PKI is that it associates a single key with a distinguished name, and in some circumstances multiple isomorphic certificates that differ only in key length would be very useful. Further, automatic generation of trust relationships based on analysis of "chains" of certificates is a complicated process that is central to the use of P-K certificates for action authorization that depends on multiple conditions and attributes.

Rationale:
In practice, we often want to be able to use keys of different "strengths" for different purposes. A "robust" digital signature architecture would allow multiple keys (e.g., of different sizes) to be associated with the same distinguished name.

Automatic generation of trust relationships based on analysis of "chains" of certificates is an important part of application security architectures. The fixed set of rules in the approach of X.509 is difficult and inflexible. The view of X.509 as a once-and-for-all standard of trust expression represents a weakness. Although it is not required to be implemented this way, the typical implementation of bit-field meanings and interpretation of X.509 fields is hardwired into the client software, libraries, and so on. This makes it expensive (in terms of the upgrade burden on clients) to introduce changes to the X.509 standards.

The Policymaker approach [BFL96] of a trust-expression language, or the general idea of a "policy engine" (that takes "any piece of signed code" and evaluates trust conditions) allows for an evolving trust model, and thus provides a much more flexible security condition rule evaluator, and this is also being integrated in SPKI.

Issue:
SPKI and SDSI potentially address the multiple signature requirement, but they offer an approach to PKI that is fairly different from X.509. (See [SPKI] and [SDSI].) SPKI, Policymaker, and SDSI may well be merged in the IEFT work, and this approach should be investigated as an alternative / adjunct to an X.509 infrastructure. (The SPKI approach may be able to accommodate X.509 as a subset.)

2.1.7 Pluggable Authentication Modules

Background:
Pluggable Authentication Modules ("PAM") are an emerging industry standard for providing a standard way of integrating security into system services like loin, ftp, etc. Such applications that wish to use security services invoke PAMs to get authentication and access control results, but the specific security mechanism may be different for different environments. This approach is not unlike GSS-API for secure communication channels for applications. PAM as been adopted by Sun, OSF, CDE, Linux, etc. (See [PAM].)

Rationale:
PAM represents a potentially important component for providing a uniform view of security to users by providing the possibility of using common security context establishment mechanism for secure system services, as well as several IPC mechanisms.

Issues:
Integration of PAM with PKI.

2.1.8 Open Interfaces and Interoperable Independent PKI Components

Background:
Certificate-based software is evolving and at an early stage of deployment. Some mistakes in infrastructure building are:

- Requiring the presence of a number of complex pieces to be built and understood before the infrastructure can be used for even the most basic operations. This raises the "cost of entry" for both the implementor and the user such that alternative infrastructures are proposed. The user should have as low a cost of entry as possible. Being able to integrate into existing and future environments without much user intervention (or effort) is a key design point. This may require predicting or influencing vendor design.

  SPKI and SDSI could be considered alternatives to X.509 proposed for this very reason although the audience is programmers not users. SSH could be considered a response to the initial costs of establishing a Kerberos/DCE-like environment when all that is needed is a secure login to a remote system.

  SSH:
  - single client, single server (small amount of software)
  - user keys are generated by user (little admin. effort)
  - transparent X11 proxy — no modification of existing applications
  - ad-hoc, point-of-use key registration — no new management function needs to be created.

  Kerberos:
  - secure key-server hardware, software and management (additional hardware and admin. effort)
  - modification of X11 clients at the library level (requiring multiple vendor effort)
  - user key registration (additional user and admin. effort)
  - service key registration (e.g. login) (additional admin. effort)
- There will be a mixture of commercial and custom implementation. Some pieces are best acquired off-the-shelf because of the development cost required to implement and the closeness of fit to actual needs:
  - certificate authority management interfaces may require database development, GUI development and are only needed in small quantity.
  - cryptographic libraries are ubiquitous and need to be both implemented efficiently, and correctly on many platforms.

  Other pieces, such as policy modules and application specific authorization, need to be custom developed, but will need to fit into and be able to extend the off-the-shelf software.
- There will necessarily be explicit component boundaries between:

- unrelated third-party software interacting within the PKI
  + security services and application
  + communicating applications, such as E-Mail, Web, V-conf, client/server
- customer/business relationship, e.g. CA services, notary services

Rationale:
The infrastructure should be usable with a small number of pieces in place. The infrastructure will necessarily be a mixture of off-the-shelf and custom components.

The interfaces between application software and services could restrict the components which can interoperate, and thus reduce the general applicability of PKI. These interfaces should therefore be widely accepted and practiced, i.e. not vendor specific. Moreover, these interface definitions should be easily obtainable: no document fees, use fees, disclosure agreements, licensing, etc.

Issues:

- Availability and interoperability of the various components (boxes in the APKI document)
- Suitability of components for application specific needs. Can the necessary additional functions be added, or does the component need to be reimplemented? E.g. in the case of the private-key component, can we perform a digital-signature that can be understood by other components? If this function is not provided, it cannot be emulated because it is cryptographic in nature, requires the private-key, and the private-key is not easily retrieved in most implementations.
- Can, and will it be useful for, DOE to require open and interoperable components in procurements of security software?

2.1.9 Firewalls and PKI Authentication

Background:
Packet filters in site border routers are a common way to protect against certain types of unwanted traffic. Currently these "firewalls" are fairly statically configured, with configuration changes being a fairly "weighty" process. Further, "opening" a particular port for some IP addresses is, at best, a weak assurance of legitimate use.

Proxies (application-specific activity filters that operate in conjunction with firewalls) work, but they are high cost in terms of development, hardware, administration, etc.

Rationale:
If firewalls could use PKI to authenticate users seeking access from the outside, then easily maintained access control lists could address a whole range of currently un-scalable administrative problems associated with current firewall practice, as well as simultaneously increasing the security and flexibility of firewalls.

Issues:
There are a range of architecture and implementation issues associated with this idea, including characterizing the types of flows and connections that are permitted on a per-user basis, etc.

2.1.10 Key-agile Environments

Background:
How do you manage the generation and distribution of hundreds of keys/second? Even multi-level session keys (e.g., symmetric keys that are used for individual connection encryption are passed between communicating end points (systems) by using longer-lived session keys known by the end systems) will have to be regenerated and distributed periodically via PKI. This problem has characteristics similar to the problem of managing long-lived client-server connections, but in this case is below the application level: all key operations must be managed automatically.

Issues:

- Fast CRL processing
- Fast, high volume certificate distribution

2.1.11 Certificate Databases

Background:
PKI effectively establishes a database associating public-key certificates and meaningful names, e.g. host names, human names. Other certificates, such as authorization, delegation, trust, may associate human names with authorizations, roles and trust. Rationale:

These names may already be present in a database elsewhere, e.g. personnel, DNS, organizational charts. It is necessary to maintain "database integrity", such as when a person is removed from the personnel database, identity certificates must be revoked. Or when an employee switches roles, authorizations are transferred over to someone else.

Rationale:
Managing the various forms of certificate data is necessary if the data is to be trustworthy enough to be used for authorization for sensitive applications.

The certificate authority issuing/revoking operations are a low-level means of maintaining data.

A higher-level means would be to tie in the certificate authority function to existing data-management tools, e.g. databases with rule systems. Higher level operations, such as "Give user Y exclusive access to resource X for N amount of time" or "Assign user X the rights associated with role Y with default override.", may translate to issuance of (multiple) certificates with particular attributes and lifetimes and perhaps even establishment of rules to enforce exclusivity.

This heavyweight management may not be required for most applications — such as that of a single person managing a small set of certificates of limited scope. The CA should be designed with this possibility in mind.

Issues:

- Maintenance of consistency among databases.
  - traditional databases
  - ad-hoc databases, such as DNS

- Integration of CA functions with existing databases.
  - CA functions (e.g. issuance, revocation) should be observable by other software (e.g. database glue), and modifiable by other software (e.g. approval/denial of certificate issuance depending on conditions).
  - CA functions should be under programmatic control, e.g. invocation of CA functions from within database glue.
- The whole range of domain specific data management issues
  - this may be out of our scope, but common to other problem areas.

Some of the identified issues are in the realm of research and development, and basic work will have to be done before ideas and implementations are ready for testbeds.

2.1.12 N-way Security Contexts

Background:
Many applications are not a client-server architecture but rather involve multiple peers. (For example, the multiple storage servers of the Distributed-Parallel Storage system (see [DPSS]) and the metacomputing environment (see, e.g.,[Zipper]). In this case, setting up a security context involves authenticating and establishing a "shared secret" among many participant processes.

The situation is similar with multicast groups, though the loser association presents a somewhat different problem.

Rationale:
With 1000's of hosts communicating securely together, traditional 2-party security protocols, e.g. SSLv3, SPKM, do not scale well (N^2 scaling).

Key-sharing is really trust-sharing, so if a large group of hosts (such as a group of data servers) trust each-other (because the software and hardware are owned by the same entity), then they can all use the same key for communication. Likewise, if a client trusts a group of hosts equally (that the software on each of them is operating), then it can share one key with the group of hosts.

Server load balancing and fault tolerance through request redirection or response "stand-in" doesn't require encryption/decryption with different keys since the keys shared between the servers and client are the same.

(Claim:) N-way security contexts explicitly allow for failure tolerance while 2-way security protocols do not — if one party of a two party context goes away, the context has been torn down. If one party of an N-way security context goes away, it doesn't necessarily mean the security context is no longer valid.

Issues:
These include scalability (we must be able to create rapidly N-way security contexts involving hundreds or thousands of participants), security (the use of a shared context must not compromise security), collective operations (e.g., multicast, scatter/gather, collective computations such as reduction), dynamics (addition/deletion of participants), and heterogeneity (we must be able to establish N-way security contexts that span resources with mul-

tiple local security mechanisms).

- A single GSSAPI security context among multiple components of a large jitterbugged application
- Related issue is multicast key distribution for M-bone (see, e.g., [Pessi])

2.1.13 Long-lived Channels and Autonomous Server Recovery

Background:
There are many circumstances where communication between distributed processes last for a long time. (Again, both the DPSS and metacomputing provide examples of this, as do secure remote shells or remote windows.) Under these circumstances, public-key authentication is followed by an exchange of session keys (e.g., DES) used for channel encryption. These session keys are changed periodically to protect against cryptoanalysis. However, a different problem arises when a server crashes. It is frequently highly desirable that a server recover from a crash and resume its service in an autonomous way. However, unlike when the server is initially configured and enabled — potentially with a human participating in the initial server authentication process — crash recovery usually has to happen without human intervention. In this case, what is the mechanism by which the server (and its platform) can authenticate?

Rationale:
As noted, secure long-lived services that operate autonomously are an important component of widely distributed computing.

Issues:
Delegation of credentials to processes (no operating system support for digital signature or crash recovery!)

OS services, daemon processes running with particular credentials need to be started back up with those credentials. The OS needs to support this. Unix supports the notion of setuid which is one form of associating credentials with a newly started process. Credentials in a public-key environment may be much richer.

3.0 Security Infrastructure Testbeds

The workshop participants agreed that PKI "testbeds" are important in order to provide environments in which researchers and users can conduct experiments with security technologies by involving DOE-relevant applications on a meaningful scale. The term "testbed" is used here to denote, typically, a collection of systems and sites, generally connected over an existing network such as ESnet, that are applying security techniques to a common problem. Alternatively, "testbeds" could be defined in terms of the scope of interoperating PKI components, in which several applications, as well as the interoperation of the PKI components, are the subject of the testbed. That is, the testbed is an experiment in interoperable PKI components, and the applications motivate the various ways in which the components are used (e.g., all aspects of certificate management in a "certificate-rich" application environment).

As discussed above, some of the proposed testbeds are focused on aspects and components of PKI for which it seems likely that commercial developments will *not* provide required advances: for example, the dynamic use of certificates for real-time authorization of remote instruments, or certificate management in distributed, high-performance computing environments. In contrast, commercial developments are highly likely to provide required advances in areas such as electronic commerce, long-term retention of signed documents, and PKI interoperability; in these areas, testbeds might focus on establishing and refining operational procedures. In all cases, DOE should attempt to be a market and technical force to drive the commercial vendors toward open system solutions with interoperable components.

3.1 Testbeds

We expect to highlight and demonstrate two principal PKI scenarios in the testbeds. First, the groups that are designing and testing general scientific application security architectures require an open PKI architecture that permits heterogeneous implementations to access many different functions (as indicated in [APKI]). Such functions include private-key operations, small (individual) as well as institutional scale certificate servers that have different types of certificate organization and search strategies, etc. On the other hand, the second scenario involves groups interested primarily in the "traditional" PKI service functions (e.g., for electronic commerce), will be more concerned with operational and procedural aspects of the problem. For example, a testbed investigating the issues exposed in the AM-NII [AM-NII] work may use a single PKI product — as they do now with Entrust — but focus on issues relating to long-term archiving of signed documents, encrypted documents, keys, etc.

We envision both kinds of PKI testbeds being implemented, with one of the goals being interoperation of these two environments.

3.2 General Testbed Issues

We can use testbeds to investigate, especially in widely distributed environments, the general issues involved with:

- Obtaining and using identity certificates
- Obtaining and using authorization certificates
- Setting up secure application control and data flows
- Using PK certificates for generalized access control
- Automatic cross-operation of CAs, conditioned on what uses may be made of remote certificates
- Interoperability of certificates from different PKIs and among commercial PKI components
- Use of a rich variety of PKI components interfaces (a la' APKI)
- Scalability issues that arise when dealing with large numbers of participants
- Performance issues relating to scale, data rates, etc.
- Authenticated non-human entities (processes, servers, systems, devices, etc.)

Independent of the particular application considered, each testbed will also provide a framework in which DOE researchers can develop an understanding of how to deploy and manage a variety of PKI technologies, including security architectures designed to protect whole classes of applications.

3.3 Application Drivers

Testbeds should be driven by applications that require the use of a common security infrastructure in a distributed environment. The following applications appear particularly promising.

- Some DOE 2000 pilot projects
- Document signing tools
- Multiple signatures
- Verifying authorization to sign business documents

3.3.1 Remote Collaboration and Instrument Control

Background:

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]

Issues:

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.

3.3.2 Metacomputing

Background:
The 1995 I-WAY project showed how high-performance networking can enable new approaches to scientific computing, in which, for example, a supercomputer computation is monitored and/or controlled by a user at a remote site; accesses data stored remotely; processes a data stream from a remote instrument, or controls that instrument; or harnesses multiple distributed computers. Several groups are developing the software technologies required to make these applications commonplace (e.g., the Globus project [Globus]) and deploying these technologies in testbeds. Lack of security facilities and security testbeds is a significant problem. An interesting set of applications exist that could be deployed rapidly in a metacomputing security testbed, enabling validation of the security mechanisms.

Issues:
To be useful, a metacomputing security testbed must involve high-end resources (supercomputers, mass store systems, instruments) at multiple sites. The heterogeneous nature of these resources makes the deployment of a uniform security solution a significant challenge. Testbeds will probably need to include resources at non-DOE sites. N-way security context establishment mechanisms are required; a large testbed can provide an opportunity for verifying the scalability of these mechanisms. Security-enhanced versions of parallel computing tools such as MPI are required. Finally, we note that a metacomputing testbed will include many nonhuman entities (e.g., servers) that must be authenticated. We discuss this issue below.

3.3.3 Remote Data Access

Background:
As high-speed networks become commonplace, it is increasingly common that users compute remotely but maintain their data locally. This is the case with many users at NERSC[10], Argonne, and no doubt other supercomputer sites. Hence, there is a need for mechanisms that can allow users to access remote data in a relatively seamless fashion, while providing acceptable performance and without compromising security.

Issues:
Several mechanisms are currently being used for remote data access. We note here two that appear particularly natural to explore in a security testbed. The Remote I/O (RIO) library developed at Argonne provides for high-performance access from (parallel) programs to re-

10.National Energy Research Scientific Computing Center

mote (parallel) file systems. For example, this library can allow an application running on a NERSC supercomputer to access files maintained on the Argonne storage system. In a security testbed linking these sites, we can explore and demonstrate the mechanisms required to authenticate such accesses, and optionally encrypt data transfers. More ambitiously, we can experiment with the deployment of a distributed file system spanning multiple DOE sites. Again, authentication is critical, as are access control and encryption.

3.3.4 IPv6 Key Management

An area that was recognized as important, but was not discussed directly at the Workshop.

3.3.5 Collaborative Engineering

Background:
In collaborative engineering applications, engineers use high-end virtual environments (e.g., CAVEs, ImmersaDesks) located at different sites to cooperate in the design of an engineering process. The process in question can be simulated via a simulation code running on a supercomputer. One example of such a system is the BoilerMaker system constructed by Lori Freitag and her colleagues; this simulates an industrial incinerator and allows engineers to collaborate on the design of retrofitted emissions control equipment. A refined version is being constructed that allows people to participate via VRML-enhanced web browsers.

Issues:
Applications of this sort often involve proprietary information and hence can require authentication and potentially privacy. Implementation is complicated by the diverse entities involved (computers, high-end display devices, web browsers, people) and by the multimedia data that must be communicated: simulation data, tracking data, sound, etc.

3.3.6 Mbone Session Key Distribution

An area that was recognized as important, but was not discussed directly at the Workshop.

3.3.7 Secure CORBA for Collaboratories and Instrument Control

An area that was recognized as important, but was not discussed directly at the Workshop.

3.3.8 Secure Access Control for Information Sources

An area that was recognized as important, but was not discussed directly at the Workshop.

3.4 Specific Testbed Issues and Topics

The workshop identified a collection of specific issues that need to be addressed in testbeds.

3.4.1 Uniform Security Architecture

Background:
One of the important drivers for testbeds will be our ability, using currently available PKI

components, to create a uniform security architecture. Users are typically interested only in their application, but have to put up with security to operate in an open environment. Hence, having a uniform security implementation that protects all of the tools needed to accomplish their job (e.g., Web access, logon authentication, secure e-mail, secure file transfer, application and information access control) is critical to the successful deployment and use of security mechanisms and policies. If the "real" users continually "stumble" over the security then they will either figure out how to short-circuit it, or revert to a non-distributed mode of operation.

Issues:
What tools are available to participate in a uniform security environment? Can modified SSH, ssl-ftp, PAM-aware utilities (with at PKI module for PAM), etc., all use a single security context?

3.4.2 Use of PKI in Scientific and Technical Applications

Investigate ability to use PKI in scientific and technical applications:

- Encapsulation
- Post-design integration
- By-design

3.4.3 PKI-based Security Gateways to Legacy Systems

Investigate the issues of using PKI-based security gateways to legacy systems.

(An area that was recognized as important, but was not discussed directly at the Workshop.)

3.4.4 PKI Interoperability

Interoperability between diverse PKI components, including testing of the standard APIs from different vendors. The APKI Working Group document [APKI] is the "touchstone" for open PKI interfaces.

3.4.5 Authentication of Non-human Entities

Background:
Multiple DOE applications require the ability to authenticate non-human entities, such as processes and physical hardware. For example, a distributed computing system must maintain a variety of "servers," providing process creation, status information, and resource brokering services. A user needs to be able to verify the identify of these services to avoid, for example, initiating a computation that involves private data on an untrusted machine. In remote instrument control applications, we may need to be able to verify that we are really interacting with a remote instrument, and not some process spoofing that instrument.

Issues:
This authentication can be achieved by using PKI, but there are also challenges relating to management of the private keys. In the case of a long-lived server, the private key must be maintained somewhere on stable storage so that if the server crashes, it can be restarted with

the same identity. The key may be stored in a file, but this is not necessarily very secure. Alternatively, we may connect a smart card to the machine on which the server is running, and require the server to read the smart card. This can be more secure, but may be impractical for large numbers of servers. In some situations, we may require that a server run only on a specific machine, in which case the machine's identity needs to be verified.

Secure DNS — with its ability to provide public-keys for IP platforms — should be an integral part of the testbeds.

3.4.6 Long-term Key Management Applications

A recognized issue for records management, where the records have been involved in, or generated in a PKI environment.

3.4.7 Digital Notary Services

- Secure time-stamping service
  (Maybe ESNet could provide this until the commercial sector decides to do so. USPS may provide this service, though nothing has happened in the two years since they first indicated that they were going to start the service.)

3.4.8 Other Issues

- Common policy specification
- Ability to define what certificates can/cannot be used for
- Who are the testbed users
  - DOE 2000 Collaboratories

4.0 References

Ada96

"IDUP and SPKM: Developing public-key-based APIs and mechanisms for communication security services". Carlisle Adams. 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.

AM-NII

"Advanced Manufacturing-National Information Infrastructure"

"The U.S. Department of Energy (DOE) created AM-NII to help advance American manufacturing. DOE's information technology expertise can be leveraged to help lower costs and improve processes. Five of its national laboratories are working together on pilot R&D solutions. Each task is led by one lab, with the other four participating.

For example, LLNL is leading the electronic commerce task. One of its major projects is with the California Industrial Leadership Council (CILC). Their goal is to cut the costs of dealing with small and medium-sized enterprises (SMEs). This CILC pilot project features an intelligent hub with Internet browser links to three small manufacturers in Southern California. Its first application is to exchange purchase orders with this SME co-op.

The other four national labs lead related virtual enterprise tasks. DOE's Kansas City Plant (KCP) is developing a next-generation information repository for improved technical data exchange. Oak Ridge National Laboratory (ORNL) is developing a Technical Solicitation Server (TSS) for Web-based electronic bidding. Los Alamos National Laboratory (LANL) is responsible for enterprise-integration modeling. Finally, Sandia National Laboratories (SNL) has been asked to look at security and intelligent agents."

(From http://zephyr.llnl.gov/zf03T1-3_ElecCom.html)

APKI

"Architecture for Public-Key Infrastructure". Internet Engineering Task Force draft: draft-ietf-pkix-apki-00.txt, available from http://www2.es.net/pub/internet-drafts/draft-ietf-pkix-apki-00.txt

BFL96

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

CORBASEC

"CORBA Security". Object Management Group. Document 95-12-1. Available from ftp://ftp.omg.org/pub/docs/

Desai

See "Kunal's report on OMG's CORBA Security Architecture Specification." at http://www-itg.lbl.gov/~kdesai/thesis.html

DPSS

"The Distributed-Parallel Storage System (DPSS)". Imaging and Distributed Computing Group, Lawrence Berkeley National Laboratory, 1996. See the DPSS home page at http://www-itg.lbl.gov/DPSS.

GASD96

"Information security — transforming the global marketplace" (a panel discussion). D. Gary, J. M. Anderson, F. Sudia, and K. Daguio. 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).

Globus

"Globus: A Metacomputing Infrastructure Toolkit". Ian Foster, Carl Kesselman, *International Journal of Supercomputer Applications*, 1997. See http://www.globus.org/.

GSS

"Generic Security Service Application Program Interface, Version 2". IETF Request for Comments: 2078. Available from http://ds.internic.net/rfc/rfc2078.txt

The Generic Security Service Application Program Interface (GSS-API), as defined in RFC-1508, provides security services to callers in a generic fashion, supportable with a range of underlying mechanisms and technologies and hence allowing source-level portability of applications to different environments. This specification defines GSS-API services and primitives at a level independent of underlying mechanism and programming language environment, and is to be complemented by other, related specifications: documents defining specific parameter bindings for particular language environments, and documents defining token formats, protocols, and procedures to be implemented in order to realize GSS-API services atop particular security mechanisms.

John96a

"A Use-Condition Centered Approach to Authenticated Global Capabilities: Security Architectures for Large-Scale Distributed Collaboratory Environments". William Johnston and Case Larsen, Ernest Orlando Lawrence Berkeley National Laboratory. Available from http://www-itg.lbl.gov/~johnston/Security.Arch.Global.Cap.html

John96b

"Real-Time Generation and Cataloguing of Large Data-Objects in Widely Distributed Environments". William Johnston, Jin Guojun, Case Larsen, Jason Lee, Gary Hoo, Mary Thompson, and Brian Tierney. Submitted to "Research and Technology Advances in Digital Libraries", May, 1997. Available at http://www-itg.lbl.gov/~johnston .

Lar96

"Open Interoperable Interchangeable Interfaces for Managing Private-key and Sensitive data" Case Larsen and William Johnston, December 1996. Available at http://www-itg.lbl.gov/~clarsen/security-html .

PAM

"Unified Login with Pluggable Authentication Modules (PAM)". Open Software Foundation, Request For Comments: 86.0, V. Samar (SunSoft), R. Schemers (SunSoft), October 1995.

Available at http://www.pilgrim.umass.edu/pub/osf_dce/RFC/rfc86.0.txt, also see http://www.redhat.com/linux-info/pam/

Pessi

"Secure Multicast" in Proceedings of the HUT Network Seminar '96. Available at http://www.tcm.hut.fi/Opinnot/Tik-110.501/1995/

This seminar series provides a very readable introduction to several relevant topics:

Chapter 1. Practical Cryptosystems and their Strength
Chapter 2. The IP Security Architecture
Chapter 3. Secure Multicast
Chapter 4. Introduction to and comparison of formalisms
Chapter 5. A Logic of Authentication by Burrows, Abadi and Needham
Chapter 6. Zero Knowledge Protocols and Small Systems
Chapter 7. Invisible communication
Chapter 8. Secure Electronic Mail
Chapter 9. Models of Electronic Commerce
Chapter 10. Mechanisms of Electronic Money
Chapter 11. Legal and Ethical Issues Related to Cryptography and Information Security
Chapter 12. Controlling and Securing Personal Privacy and Anonymity in the Information Society

PKIX

"[The Public-Key Infrastructure (X.509) (pkix) group focuses] on tailoring and profiling the features available in the v3 X.509 certificate to best match the requirements and characteristics of the Internet environment. Other topics to be addressed potentially include:

- Alternatives for CA-to-CA certification links and structures, including guidelines for constraints
- Revocation alternatives, including profiling of X.509 v2 CRL extensions
- Certificate and CRL distribution options (X.500-based, non-X.500-based)
- Guidelines for policy definition and registration
- Administrative protocols and procedures, including certificate generation, revocation notification, cross-certification, and key-pair updating

- Naming and name forms (how entities are identified, e.g., e-mail address, URN, DN, misc.)"

Current Internet-Drafts:

"Internet Public Key Infrastructure Part I: X.509 Certificate and CRL Profile"
"Internet Public Key Infrastructure Part III: Certificate Management Protocols"
"Architecture for Public-Key Infrastructure"
See http://www.ietf.org/html.charters/pkix-charter.html

SDSI

"SDSI — A Simple Distributed Security Infrastructure." R. Rivest and B. Lampson. Available at http://theory.lcs.mit.edu/~rivest/publications.html

SPKI

"Simple Public-key Infrastructure." 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. Send "subscribe spki" to majordomo@c2.org to join. Also see http://www.clark.net/pub/cme/html/spki.html

SPKM

"The Simple Public-Key GSS-API Mechanism (SPKM)." IETF Request for Comments: 2025. Available from http://ds.internic.net/rfc/rfc2025.txt

This specification defines protocols, procedures, and conventions to be employed by peers implementing the Generic Security Service Application Program Interface (as specified in RFCs 1508 and 1509) when using the Simple Public-Key Mechanism.

SSH

"The SSH (Secure Shell) remote login protocol." Tatu Ylonen. See http://www.cs.hut.fi/ssh/.

X9.45

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

This document is an introduction to the Financial Industry Security Standards (ANSI X9), including ANSI X9.45 draft of "Enhanced Management Controls Using Digital Signatures and Attribute Certificates"

Zipper

"A Secure Communications Infrastructure for High-Performance Distributed Computing." Ian Foster, Nicholas T. Karonis, Carl Kesselman, Greg Koenig, Steven Tuecke, Mathematics and Computer Science Division, Argonne National Laboratory. See http://www.mcs.anl.gov/zipper/index.html. Submitted for publication.

Appendix 1: Second Joint Workshop Program

DOE Security Research Workshop-II

Agenda

WEDNESDAY, Dec. 11

9:00

Intro

Framing the workshop - what we want to accomplish:

- A description of perceived security infrastructure requirements
- Proposals for how to meet these requirements, together with an identification of areas in which uncertainties remain regarding implementation strategy
- A description of a set of research projects that could use such a security infrastructure, and/or contribute to its development
- Ideas for testbeds that could be used to support these activities

9:15

Session 1: Issues in Implementing and Operating Public-Key Infrastructure

9:15

o TALK: "PKI: Principles and Operation" (W. Johnston),

10:15

o TALK: "Current experience (AM-NII PKI project)" (John Long, Sandia)

11:20

o Discussion on DOE PKI requirements

12:20

Lunch

1:20

Session 2: Issues for Building Security Architectures and Applications on PKI

1:20

o TALK: "ATM Link-level Encryption"- Lyndon Pierson, Sandia (lgpiers@sandia.gov)

2:20

o TALK: "Secure Communications for High-Performance Computing" - Ian Foster, ANL (itf@mcs.anl.gov)

3:40

o TALK: "A Use-Condition Centered Approach to Authenticated Global Capabilities: Security Architectures for Large-Scale Distributed Collaboratory Environments" - W. Johnston, LBNL

4:40

o Discussion on the requirements of DOE applications and architectures

END

THURSDAY, Dec. 12

9:00

o TALK: Web security - Kevin McCurley, Sandia

10:00

o TALK: LLNL Closed Testbed (Doug Mansur)

```
11:20
   o Talk: ASCI: Pete Deane, Sandia
12:20
   Lunch
1:20
   Session 3: Research Issues Related to DOE use of
              PKI-based Architectures and Applications
1:20
   o TALK: SPKI Certificates - Carl Ellison
2:20
   45 min + 15
   o TALK: "X.509 - past, present, and future" Warwick Ford
3:40
   o TALK: "Draft Requirements for Modular Private Key
            Management" Case Larsen, LBNL
4:40
   o TALK: Alternatives to commercial CAs (SecuDE,
            SESAME, Web CAs) Case Larsen and W. Johnston, LBNL
5:25
   o Discussion on research areas
6:25
   End

FRIDAY, Dec. 13
------
9:00
   Session 4: Security Testbeds
9:00
   o Talk: DOE topics: Judy Moore, Sandia
10:00
   o TALK: Sandia Closed Testbed (Mike Sjulin)
11:20
   o TALK: Applications for open testbeds (Wm. Johnston,
   LBNL)
11:50
   o Discussion on testbeds
12:35
   o Wrap-up
1:00
   END
```

Appendix 2: Second Joint Workshop Participants

NAME	ORGANIZATION	PHONE NUMBER
Harry Leake	Yucca Mountain Project (YMP) M&O	702-794-7705
Aaron Engel	Yucca Mountain Project (YMP) M&O	702-794-7231
C. Douglas 'Doug' Brown cdbrown@sandia.gov	Sandia National Laboratories (SNL)-04621 MS 0806	505-845-8699
Stephen T. 'Steve' Elbert elbert@ameslab.gov	Ames Laboratory	515-294-1307
Sharon Jacobsen	Lockheed Martin/Oak Ridge	423-574-0900
Armin R. Miller	Ames Laboratory	515-294-4286
John Volmer	AWL (Argonne)	630-252-5447
Dale Sparks	DOE Albuquerque Sparks & Associates	505-845-5221
Ray W. Surface	DOE Albuquerque Sparks & Associates	505-845-5337
Larry E. Parker lep@lanl.gov	Los Alamos National Laboratories (LANL)	505-667-3943
Douglas 'Doug' E. Engert deengert@anl.gov	Argonne National Laboratory (ANL)	630-252-5444
Mary Anne Scott scott@er.doe.gov	DOE-ER31	301-903-6368
Case T. Larsen CTLarsen@lbl.gov	Lawrence Berkeley National Laboratory (LBNL)	510-486-5778
Thomas 'Tom' A. Harper taharper@pnl.gov	Pacific Northwest National Laboratory (PNNL)	509-375-2150
William 'Bill' E. Johnston WEJohnston@lbl.gov	Lawrence Berkeley National Laboratory (LBNL)	510-486-5014
Ian T. Foster itf@mcs.anl.gov	Argonne National Laboratory (ANL)	630-252-4619
James A. Rome romeja@ornl.gov	Oak Ridge National Laboratory (ORNL)	429-574-1306
Sandy Goldston	LMES/Oak Ridge	423-574-5212
Frank W. Ploof fploof@llnl.gov	Lawrence Livermore National Laboratories (LLNL)	510-422-6990
Douglass 'Doug' L. Mansur mansur@llnl.gov	Lawrence Livermore National Laboratories LLNL	510-422-0896
Ronald 'Ron' W. Wilkins ronw@lanl.gov	Los Alamos National Laboratories (LANL)	505-665-1879

James 'Jamey' N. Maze mazejn@ornl.gov	Oak Ridge National Laboratory (ORNL)	423-574-6355
Victoria 'Vickie' A. Hamilton vahamil@sandia.gov	Sandia National Laboratories (SNL)-06513 MS0449	505-845-8779
Gabi G. Istrail ggistra@sandia.gov	Sandia National Laboratories (SNL)-06512 MS 0449	505-845-7749
Robert 'Bob' J. Aiken aiken@anl.gov	Argonne National Laboratory (ANL)	301-271-2919
Peter 'Pete' Dean pwdean@sandia.gov	Sandia National Laboratories (SNL)-08910 MS 9011	510-294-2656
John P. Long jplong@sandia.gov	Sandia National Laboratories (SNL)-04621 MS 0806	505-845-8622
Brian Desind bdesind@kcp.com	Allied Signal	505-844-2987
Kevin McCurley	Sandia National Laboratories (SNL)	505-845-7378
Peter S. Gemmell psgemme@sandia.gov	Sandia National Laboratories (SNL)-09222 MS 0820	505-845-7604
Warwick Ford	Verisign	617-492-2816
Pat Moore pcmoore@sandia.gov	Sandia National Laboratories (SNL)-04621 MS 0806	505-844-3588
Bernard 'Bernie' P. Clifford bpclifford@sandia.gov	Sandia National Laboratories (SNL)-06522 MS 0974	505-284-3102

Appendix 3: Public-key Certificates — 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.)

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

The enabling underlying technology for many aspects of our approach is the ability for one person ("A") to encrypt a piece of information with a private-key, and another person to decrypt that information in 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.

Digital / Cryptographic Signatures

Digitally signing a document ensures it authenticity 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[11]. 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[12]. 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 documents in that their function is typically to participate in the authentication and authorization phase of a security system.

---

11. Throughout this paper the term "private" key will refer to the private portion of a public/private key pair.

12. 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.

Certificates

In general, certificates are small documents, some of which may have a standardized format (e.g., X.509) and some of which do not. Public-key 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.

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 at the approach taken by 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]).

Certification Authorities

CAs serve a dual role in our model. On the 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 at one end of the spectrum (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.

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.

References and Notes for "Appendix 3: Public-key Certificates — Background"

[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.

[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.

[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.

[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.

[Ken93] 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.

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

[LBN96] LBNL. The Distributed-Parallel Storage System (DPSS) home page, 1996. Available at http://www-itg.lbl.gov/DPSS.

[LDA96] Lightweight Directory Access Protocol, 1996. Available at http://www.umich.edu/~rsug/ldap/ldap.html.

[Lin93] 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.

[Lin96] John Linn. The Kerberos version 5 GSS-API mechanism, June 1996. Available at ftp://ds.internic.net/rfc/rfc1964.txt.

[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.

[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/.

[SES] SESAME — a Secure European System for Applications in a Multi-vendor Environment. Available at http://www.esat.kuleuven.ac.be/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.

[SPK] 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.

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

[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.

[You] Eric Young. SSLeay implementation of the SSL protocol. Available at http://www.psy.uq.edu.au:8080/~ftp/Crypto. SSLeay is a free implementation of Netscape's Secure Socket Layer — the software encryption protocol behind the Netsite Secure Server and the Netscape Browser. What started as an effort to implement the SSL protocol has turned into a fairly complete cryptographic library (which Eric is still working on). There is also quite a bit of ASN.1 sup port, with routines to convert and manipulate the base ASN.1 types, X509v3 certificates, certificate requests, certificate revocation lists (CRL), RSA private-keys and DH parameters. There are routines to load and write these objects in base64 encoding and routines to convert ASN.1 object identifiers to/from ASCII representations and an internal form. There are functions for verification of X509 certificates and for specifying where to look for certificates to "climb" the x509 "tree". This last part of the library is still evolving. The big number library is quite complete and has no restrictions on the size of the numbers manipulated. RSA and Diffie-Hellman routines have been layered on top of this library.