NHSE Distribution of HPCC Software: Legal Issues and Technological Options

Shirley Browne, Paul McMahan, and Danny Powell

Report submitted to the NHSE Technical Committee

December 4, 1996

1 Introduction

The goal of the NHSE is to distribute HPCC software to as broad a U.S. audience as possible, so as to maximize the return on HPCC agency investment in developing this software by promoting further research and increasing U.S. market competitiveness. Where possible, HPCC software should also be made available to foreign researchers who are collaborating with U.S. scientists. However, the distribution mechanisms must provide reasonable assurances that intellectual property rights are protected and that export regulations are abided by.

The approach of the NHSE is to encourage the development of HPCC repositories that are maintained by experts and that provide access to software and documents within their specific domains. An example of such a repository is the Netlib mathematical software repository. The NHSE will then link together these domain-specific repositories and provide access for HPCC users in a convenient way. The NHSE is developing a Repository in a Box (RIB) toolkit that will include tools for carrying out basic repository setup and maintenance tasks.

A repository might provide access just to software catalog records containing pointers to the actual software that is available elsewhere (i.e., be a "virtual repository"), or it might distribute the actual software itself. Similarly, software may be shared between interoperating repositories at two levels: 1) at the level of catalog information that describes the software, 2) at the level of actual software files. Advantages of the direct distribution approach may be provision of faster and reliable service to users, as well as a single point of contact for administrative procedures such as license agreements. Problems with the direct approach include liability for enforcing legal restrictions and proper crediting of download and usage statistics to the originating site.

The NHSE plans to work within the HPCC agencies' rights management and software distribution policies to distribute HPCC software to authorized users in as streamlined and efficient manner as possible while providing adequate security for enforcement of restrictions on software distribution. Effective use of encryption and authentication technologies that have been incorporated into the current established base of secure Web browsers and servers will enable secure electronic distribution of restricted software to only authorized users. Use of public key cryptography will allow digital signing and online execution of license agreements and contracts. Recipients of downloaded software may be identified and authenticated by their public key certificates, and records of software transfer transactions will be kept.

The remainder of this report is organized as follows. Section 2 gives an overview of the software rights management process. Section 3 discusses different options for distributing restricted software from NHSE repositories. Section 4 discusses policy and liability issues. Section 5 describes a prototype implementation of a secure software distribution mechanism based on public key certificates and the Secure Socket Layer (SSL) protocol. Section 6 lists decisions that remain to be made and legal agreements that need to be in place before production distribution of access-restricted software by NHSE repositories can occur.

2 Overview of the Software Rights Management Process

This section describes a generic software rights management process that might apply to any of the HPCC agency laboratories and centers. An example such a process is given in the draft NASA draft software release policy dated June 3, 1996 [1].

A request to release a piece of software is typically made by the author to the organization for which he works. The author fills out a form that captures the information required to resolve rights issues and detect possible infringement of patents, trademarks, or copyrights. Then an intellectual property assessment is carried out by the organization's intellectual property counsel. Then an intellectual property assessment is carried out by the organization's intellectual property counsel. This assessment determines the following:

- whether any rights infringements have been made by the author
- whether the organization should seek to protect intellectual property rights embodied in the asset (e.g., by filing a patent application)
- whether the asset is eligible for foreign release and, if so, whether or not an export license will be required

After the intellectual property assessment has been carried out, the software is labeled with a copyright legend and, if applicable, patent and/or trademark notices. Finally, a release category is chosen for the asset. Organizations typically have several release categories such as public domain, royalty-free license (e.g., freeware), beta-testing, and licensed. If a category requiring a license is chosen, one or more appropriate licenses are written. Different licenses might be required for different end users (e.g., research vs. commercial use).

If the organization which contributes the software is different from the organization which runs the repository from which the software is to be distributed, then an agreement needs to be in effect between the two organizations specifying how the intellectual property rights owned by the contributing organization will be managed by the repository. Similarly, when interoperating repositories exchange non-public-domain software, agreements need to be in effect specifying how repositories will manage intellectual property rights associated with the exchanged software. The types of intellectual property rights and legal restrictions that are of main concern for NHSE software are copyright, patents, trademarks, and export restrictions. Some backgound information on these topics is contained in the following subsections.

2.1 Copyright

Authors of original works fixed in any tangible medium of expression can obtain limited protection for their intellectual property through the copyright laws of the United States. Copyright protection is in effect as soon as the work is fixed in a tangible medium of expression, but the copyright owner cannot bring an infringement action until the copyright has been registered with the U.S. Copyright Office. The copyright usually belongs to either the author or his or her assignee. The exception is a work for hire, which can either be a work authored by an emmployee within the scope of his or her employment or a commissioned work. The latter requires a written agreement declaring that the task is a work for hire. In the case of a work for hire, the employer owns the copyright. If owned by the author, copyright protection remains in effect for the author's lifetime plus fifty years. For a work for hire, the duration of the copyright is 75 years from publication or 100 years from creation, whichever is earlier.

The copyright owner may assign or license the rights to the copyrighted work. The owner of a copy of a copyrighted work may loan, sell, or lease the copy without restriction. The owner of a copy of a computer program can install and execute the program on a single computer. Additionally, the owner can make a copy of the program for archival purposes. The copyright owner may assign reusers the right to make and and distribute copies of the program, but unless the program has been explicitedly placed in the public domain, such permission may subsequently be withdrawn.

Works of the U.S. government are public domain and cannot be copyrighted. However, although the U.S. government cannot get copyright for its own works, it can have an existing copyright assigned to it. For example, an independent contractor working for the government owns the copyright to the work it produces but may assign the copyright back to the government.

2.2 Patents

A patent protects an idea and gives an inventor the exclusive right to prevent others from making, using, or selling his or her invention for seventeen years after the patent is issued. The governing law for patents in the United States is Title 35 of the United States Code, or 35 USC. In order to be patentable, the invention must fall into one of the following five statutory classes of things that are patentable: 1) processes, 2) machines, 3) manufactures, 4) compositions of matter, and 5) new uses of any of the preceding. Most software patents fall under the category of processes. In addition, the invention must be useful, novel, and nonobvious.

Patents are awarded by the U.S. Patent Office. Under U.S. patent law, a patent will not be granted to an applicant unless the application is filed less than one year from the date that the invention was first sold or offered for sale within the United States. The patent will also be denied unless the application is filed within one year of the date the invention was described in a printed publication anywhere in the world. Under 35 USC section 287, a patent owner is required to mark goods embodying the invention with the patent number.

2.3 Trademarks

A trademark is any word, slogan, or symbol which is used in trade with goods and services to indicate their source of origin and to distinguish them from the goods and services of others [1]. Trademark rights may be used to prevent others from using a confusingly similar mark, but not to prevent others from offering the same goods and services under a non-confusing mark. In the United States, trademark rights are created when use of a trademark begins. However, these rights are often limited. Greater rights are available by registering the trademark with the state or federal government. Trademarks used in interstate or foreign commerce may be registered in the U.S. Patent and Trademark Office (PTO). A trademark application can be submitted to the PTO based upon actual use in commerce or a bona fide intent to use the mark in commerce. A trademark Examiner is assigned to each application and considers the registrability of the mark in light of the statutory guidelines.

2.4 Export Restrictions

Export restrictions on software are governed by the International Traffic in Arms Regulations (ITAR) (22 CFR parts 120-130), and the Office of Foreign

[1] This information is taken from `http://www.malloylaw.com/trademk.html`

Assets Control (OFAC) (31 CFR parts 500-585), and the Export Administration Regulations (EAR) (15 CFR Parts 768-799) Release to a foreign national living in the United States is considered an export.

ITAR specifically controls cryptographic software. Export of ITAR controlled software to any destination except for Canada requires a validated export license issued by the Office of Defense Trade Controls of the State Department.

OFAC imposes a complete export embargo on Cuba, Iraq, North Korea, and Yugoslavia. All software exports to OFAC-embargoed countries must be authored by OFAC.

The Bureau of Export Administration (BXA) of the Department of Commerce issues the EAR which regulate software that does not fall under ITAR. In its new ruling issued in March of 1996, which is effective immediately and becomes mandatory November 1, the BXA is dropping the terms “general license” and “validated license”. Instead, the term “license” is used to refer only to authorization issued by BXA upon application. The many existing general licenses have been converted into a smaller number of ”exceptions”. The redesigned Commerce Control List (CCL), to be used in tandem with a new Country Chart, indicates whether a license is required for a given Export Control Classification Number (ECCN) to any country in the world and the reasons for control. In using a license exception, the exporter certifies that all terms, conditions, and provisions for use of that license exception have been met. A license requirement may be based on the end-use or end-user. There are no license exceptions to General Prohibition Four (the Table of Denial Orders list of prohibited companies and individuals) or General Prohibition Five (End-Use and End-User having to do with nonproliferation). Various end-use and end-user limitations are placed on certain license exceptions.

When a U.S. domestic party is releasing software to another U.S. domestic party, there is no need for an export control provision in the agreement, even when it is known that the recipient will be exporting the software. The only exception is that the software may not be delivered domestically if the supplier has reason to believe that it will be exported illegally.

There is special concern for parallel software in that the Commerce Department has placed it in a separate category which continues to require a validated export license[2]:

> ECCN 4D03A: Operating system software, software development tools, and compilers specially designed for multi data stream processing equipment, in source code.

In one case, the Commerce Department has denied export for a parallel iterative solver package developed at Sandia Laboratories [3].

However, if a project passes the “fundamental research” litmus test, then results results from that project, including software, are exempt from Commerce Department export regulations [4]. The main litmus test for fundamental research is whether research results are reviewed for the purpose of withholding the release of information from publication. Review to ensure that proprietary information used for the research project is not disclosed is allowed.

3 Options for Distributing Restricted NHSE Software

3.1 Agency Requirements

The NHSE has not been given any official guidelines specifying HPCC agency requirements and conditions for distribution of restricted NHSE software, but judging from the available information, the requirements appear to be the following:

- Anyone downloading the software will execute an agreement which binds them to abide by stipulated terms and conditions concerning copying, use, and distribution of the software.
- Recipients of software will be accurately identified and the transaction logged for future reference.
- Distribution of HPCC agency software must be restricted to domestic U.S. citizens unless approval to export the software is obtained from the appropriate agency official. Unrestricted access on the Internet is considered export.

3.2 Options for User Authorization and Authentication

Authorization means making a judgment as to whether or not a user should be permitted access. Authentication means verifying that the party attempting access is who he claims to be. The initial authorization and authentication process requires manual intervention to check the user's credentials and determine access privileges. Pre-authorization for one or more software packages or entire classes of software may be carried out once and for all, with subsequent access requiring only authentication which may be done automatically.

The following three subsection describe options for user authentication.

3.2.1 Pretty Good Privacy (PGP)

PGP is a public key system for encrypting electonic mail using the RSA public key cypher [2] [5]. It encrypts the message using the Swiss IDEA cypher with a randomly generated key. It then encrypts the key using the recipient's public key. When the recipient recieves the message, PGP uses his private RSA key to decrypt the IDEA key and then uses that IDEA key to decrypt the message.

[2] `http://axion.physics.ubc.ca/crypt.html`

PGP can also be used to sign messages. It does so by first computing a "hash" of the message using the hash function MD5. It then encrypts this hash output (128 bits or 16 bytes) with the secret RSA key of the sender. Any recipient can calculate that same hash output of the received message and use the sender's public key to decrypt the signature. If the output to this decryption agrees with the recipient's calculated hash output, then the recipient knows both that the sender actually sent that message and that not a single bit of that message has been changed.

The PGP software is freely available for non-commercial use to U.S. and Canadian citizens from the MIT PGP Web site [3]. This distribution is done in cooperation with Philip Zimmermann, the author of PGP, and with RSA Data Security, Inc., which licenses patents to the public-key encryption technology on which PGP relies.

The PGP software will generate a public /private key pair for the user, of the length specified by the user (up to 2048 bits). Since it is the user that generates the key pair, one of the problems is that of trust. How do you know that the public key claimed to be from the intended recipient is not that of an enemy instead pretending to be the recipient? PGP uses the idea of a "Web of Trust". It advises anyone generating a public key to have it signed by a number of other trustworthy people who are in effect affirming that the key belongs to the one it claims to belong to. Thus the hope is that at least one of the signers is someone known to the sender as a trustworthy person, or that he is someone vouched for by a known trustworthy person. MIT maintains a public key server that can be accessed via the Web or by email to retrieve the public key of a registered party.

An informal Web of Trust would not be sufficient for reliably identifying NHSE users. The following possibilities would improve this situation:

- Have HPCC agencies and organizations sign keys and have users submit them to an existing PGP public key server
- Have the NHSE run a PGP public key server
- Use the Four11 commercial PGP key certification service ($20/certificate)

Because current secure Web servers and browsers aren't compatible with PGP, the NHSE would have to write its own applications software to verify PGP signatures. Also, the X.509 certificates (discussed below) used by current secure Web servers and browsers take advantage of a more scalable hierarchy of certification authorities for handling the trust problem discussed above. The incompatibility problem may change with future versions of PGP.

The National Center for Supercomputing Applications (NCSA) currently uses PGP for secure email correspondence with users [4]. Two members of

[3] `http://web.mit.edu/network/pgp.html`

the NCSA computer security staff maintain the NCSA key ring by personnally signing users' public keys and placing them on this key ring. All transactions requiring user authentication are currently carried out via email.

PGP is in use at the distributed set of Netlib sites to allow Netlib editors and authors to sign software and users to verify the authenticity and integrity of software they retrieve from Netlib [6]. Actually what is signed are MD5 fingerprints of the software files rather than the actual software files themselves. The public keys for the Netlib editors are published in the SIAM Newsletter.

The Resource Cataloging and Distribution System (RCDS) [7] currently under development at the University of Tennessee uses PGP to allow authors to sign software descriptions. PGP is invoked by the RCDS publisher tool to sign a description that an author has created using an HTML form. The publishing tool automatically creates additional fields, such as the MD5 fingerprint of the software file being published, signs the description using the author's private key, and uploads the PGP-signed description to an RCDS catalog server.

In a similar manner, an NHSE software request tool could invoke PGP to sign a user's request for software using his or her private key. Presumably the request would take the form of a license agreement that the user PGP-signs to indicate his willingness to comply with the terms of the agreement if he obtains the software. The request tool could then send the request to a specially designed file server that would verify the requestor's identity by checking a PGP key ring. Then, provided that the requestor were authorized to access the requested software (such authorization would need to be checked by lookup in a separate database, because there are no provisions for including additional information with a PGP public key), the server would return the software in an encrypted form, along with a key to decrypt it. The software files would be best encrypted using a symmetric encryption scheme. The symmetric encryption key would be encrypted using the requestor's public key so that only he could decrypt the software file. The symmetric key should probably be a "session key" generated solely for the purpose of that transmission.

3.2.2 Kerberos

Kerberos is a network authentication system for use on physically insecure networks that allows entities communicating over those networks to prove their identity to each other while preventing eavesdropping or replay attacks [4]. Kerberos provides for data stream integrity (detection of modification) and secrecy (prevention of unauthorized reading) using cryptography systems such as DES. Kerberos works by providing principals (users or services) with tickets that they can use to identify themselves to other principals and with secret cryptographic keys for secure communication with other principals. Kerberos does not provide for authorization or accounting, although applications can use their secret keys

[4] `http://nii.isi.edu/info/kerberos`

to perform these functions securely. The NHSE would need to either incorporate Kerberos into a software distribution application as part of Repository in a Box, or contract with a company that provides commercial support for Kerberos, such as CyberSAFE Corporation, Cygnus Support, or OpenVision Technologies.

In a Kerberos system, there is a designated site on the network, called the Kerberos server, which performs centralized management and administrative functions. The server maintains a database In a Kerberos system, there is a designated site on the network, called the Kerberos server, which performs centralized management and administrative functions. The server maintains a database containing the secret keys of all users, generates session keys whenever two users wish to communicate securely, and authenticates the identity of a user who requests certain network services. If the server is compromised, the integrity of the whole system fails. With Kerberos Version 5, multiple Kerberos systems, called "realms", may interoperate.

Kerberos authentication, but not message content encryption, via HTTP is supported in NCSA HTTPd 1.5 (and 1.6b1) as well as XMosaic 2.7b.

Secret-key authentication systems such as Kerberos were designed to authenticate access to network resources, rather than to authenticate documents, a task which is better achieved via digital signatures. Because authentication of documents is needed for distribution of restricted NHSE software, for example for electronic license agreements, we do not consider Kerberos further in this report.

3.2.3 X.509 Certificates and Certification Authorities

The following is excerpted from the RSA Security FAQ Version 3.0 [5]:

> ITU-T Recommendation X.509 [8] specifies the authentication service for X.500 directories, as well as the widely adopted X.509 certificate syntax. The initial version of X.509 was published in 1988, version 2 was published in 1993, and version 3 was proposed in 1994 and considered for approval in 1995. Version 3 addresses some of the security concerns and limited flexibility that were issues in versions 1 and 2.
>
> Directory authentication in X.509 can be carried out using either secret-key techniques or public key techniques, with the the latter is based on public key certificates. An X.509 certificate consists of the following fields:
>
> - version
> - serial number
> - signature algorithm ID

[5] `http://www.rsa.com/`

- issuer name
- validity period
- subject (user) name
- subject public key information
- issuer unique identifier (version 2 and 3 only)
- subject unique identifier (version 2 and 3 only)
- extensions (version 3 only)
- signature on the above fields

This certificate is signed by the issuer to authenticate the binding between the subject (user's) name and the user's public key. The major difference between versions 2 and 3 is the addition of the extensions field. This field grants more flexibility as it can convey additional information beyond just the key and name binding. Standard extensions include subject and issuer attributes, certification policy information, and key usage restrictions, among others.

The X.509 standard is supported by a number of protocols, including PEM, PKCS, S-HTTP, and SSL.

The SSL (Secure Socket Layer) Handshake Protocol was developed by Netscape Communications Corporation to provide security and privacy over the Internet. The protocol supports server and client authentication. The SSL protocol is application independent, allowing protocols like HTTP, FTP (File Transfer Protocol), and Telnet to be layered on top of it transparently. The SSL protocol is able to negotiate encryption keys as well as authenticate the server before data is exchanged by the higher-level application. The SSL protocol maintains the security and integrity of the transmission channel by using encryption, authentication and message authentication codes.

The SSL Handshake Protocol consists of two phases, server authentication and client authentication, with the second phase being optional. In the first phase, the server, in response to a client's request, sends its certificate and its cipher preferences. The client then generates a master key, which it encrypts with the server's public key, and transmits the encrypted master key to the server. The server recovers the master key and authenticates itself to the client by returning a message encrypted with the master key. Subsequent data is encrypted with keys derived from this master key. In the optional second phase, the server sends a challenge to the client. The client authenticates itself to the server by returning the client's digital signature on the challenge, as well as its public-key certificate.

An X.509 certificate binds an identity to a pair of electronic keys that can be used for encrypting and signing digital information. The pair consists of two related keys – a public key and a private key. The public key can be used by anyone to verify a message siged with the private key or to encrypt a message that can only be decrypted using the private key. The private key must be kept secure and protected against unauthorized use.

Certificates are issued by a Certification Authority (CA), which is a trusted party that vouches for the identity of those to whom it issues certificates. In order to prevent forged certificates, the CA's public key must be trustworthy. The CA can either widely publicize its public key or provide a certificate from a higher level CA which attests to the validity of its public key. The latter leads to a hierarchy of CAs. To obtain a certificate, an individual generates his own key pair and sends the public key to the CA with proof of his identity. Different CAs may issue certificates with different levels of identification requirements. For example, Verisign is a commercial CA that offers four classes of certificates, with increasing levels of assurance (and cost) [6]. Using the requirements for the particular level applied for, the CA checks the identification and then sends the requestor a certificate attesting to the binding between the requestor and his public key, along with (possibly) a hierarchy of certificates verifying the CA's public key.

The National Institute of Standards and Technology (NIST) is taking a leadership role in the development of a Federal Public Key Infrastructure that supports digital signatures and other public key-enabled security services [7] [9]. In doing this, NIST is coordinating with industry and technical groups developing PKI technology such as the Federal PKI Steering Committee and its Technical Working Group (TWG), CommerceNet, Internet's PKIX, and the Open Group. NIST chairs the TWG, which is composed of technical representatives from Federal agencies and industry. Active since October 1994, the TWG has developed initial versions of a requirements document, a concept of operations, a technical security policy, an X509 v3 certificate profile, and an interoperability report. Laboratory activities include the development of a Reference Implementation and the initial implementation of a root Certification Authority (CA) for the Federal PKI.

To protect agaist long-term factoring attacks, a certificate has a validity period which should be shorter than the expected factoring time. The validity period, together with the need for security and the expected strength of an attacker, determines the appropriate key size which should be chosen when generating the key pair. In the event that someone's private key is compromised before it expires, he must let others know by adding the associated certificate to a Certificate Revocation List (CRL). The CRL is maintained by the Certification Authority that orginally certified the key. When verifying a signature, one can

[6] `http://www.verisign.com`

[7] `http://csrc.ncsl.nist.gov/pki/`

check the CRL to make sure the signer's key has not been revoked.

Someone who obtains the private key of a CA could then forge certificates. Thus, a CA must ensure that its private key is extremely secure, for example by keeping it in a tamperproof box called a Certificate Signing Unit, or CSU. Furthermore, to protect against long-term factoring attacks, a CA should use very long keys and change keys regularly.

An individual can use a certificate to identify himself to secure servers such as membership-based or access-controlled Web servers. Multiple certificates can be attached to a message or transaction, forming a certificate chain in which each certificate attests to the authenticity of the previous certificate. The top-level CA in the chain must be independently known and trusted by the recipient. When installed in a Web browser, a certificate functions as electronic credentials, eliminating the need for typing in a username and password. Similarly, a secure Web server uses its own certificate to assure clients that the server is run by the organization claimed and to verify the integrity of the provided documents.

X.509 client certificates are currently supported by Netscape in its Navigator 3.0 browser. Microsoft has also announced support for X.509 certificates in its client applications. X.509 server certificates are currently used in server products from IBM, Microsoft, Netscape, OpenMarket, and Oracle.

NCSA HTTPd 1.6, currently is beta testing, provides support for the Secure-HTTP (S-HTTP) protocol and SSL version 2.0 and 3.0 [8]. A version of XMosaic which supports S-HTTP and SSL is also available.

3.2.4 Simple Distributed Security Infrastructure

Simple Distributed Security Infrastructure (SDSI) has been proposed as a simpler alternative to X.509 [10]. SDSI combines a simple public-key infrastructure design with a means of defining groups and issuing group membership certificates. SDSI's groups provide terminology for defining access-control lists and security policies. SDSI's design relies on linked local name spaces rather than a hierarchical global name space.

3.2.5 Digital Signatures

To attach a digital signature to a document, an author uses a secure one-way hash function to compute a digest, or "digital fingerprint" of the document. A secure one-way hash function ensures that it is impossible to create a second, different document with the same fingerprint, or to derive the original document from the fingerpint. The author then encrypts the fingerprint with his private key. The encrypted fingerprint is the digital signature for the document. Given a digitally signed document and the author's public key, one can verify both that the document was actually signed by the author and that the document has not been altered since it was signed. In the case of a contract signed by two

[8] `http://hoohoo.ncsa.uiuc.edu/beta-1.6/`

parties, both may attach their digital signatures to the same document so that it may later be verified that both agreed to it.

3.2.6 Digital Time-stamping Services

After a key expires, everything that was signed with it will no longer be considered valid. If a signed document must remain valid after the key used to sign it expires, then the document should be time-stamped by a digital time-stamping service (DTS). A DTS issues a time-stamp which associates a date and time with a digital document in a cryptographically strong way. In addition to verifying the author's identity and the integrity of the document, the time-stamp also proves that the digital document existed at the time stated. Even if a private key used to sign the document is later compromised, the document remains valid. The contents of the document need not be revealed to the DTS. The author can compute a message digest of the document using a secure hash function and then send the message digest to the DTS, which returns a digital time-stamp consisting of the message digest, the date and time it was received, and the digital signature of the DTS. Strong cryptographic techniques must be used to ensure that time-stamps cannot be forged. One way to satisfy the strong cryptographic requirements is to store the private key of the DTS and an accurate clock inside a tamperproof box. The DTS must also have a long key that will be valid for several decades. A cryptographically strong DTS which avoids the need for tamperproof hardware has been implemented by researchers at Bellcore [11]. A digital time-stamping service is currently offered by Surety Technologies [9], and Bellcore has plans to offer such a service in the future [10].

3.3 Options for Restricting Access

3.3.1 By Domain Name

A simple method for enforcing access restrictions for NHSE software would be to test the hostname of the machine from which a request originates. This hostname could be used to determine whether or not the request was made from a host within a certain domain, such as `.gov` or `.edu`. Access to software could then either be allowed or denied based on the domain name.

The amount of effort required to enable this type of access control would be minimal. On the file server side, enabling this feature with most mainstream HTTP servers would be as simple as adding a few lines to the configuration file. On the client side, the whole process would be invisible unless, of course, the download request was denied.

Unfortunately, this type of access control presents some problems. One problem is that sometimes the partitioning of hosts created by domain names

[9] `http://www.surety.com`
[10] `http://www.bellcore.com/`

would not match the restriction criteria. For example, access permission to software is often based on what country the request originates from. However, some domain names, such as `.org`, `.com`, and `.net`, do not indicate where the host is geographically located while others, such as `.us`, `.fr`, and `.uk` do make this distinction.

Even if one chose to err on the side of caution and allow access only to those hosts that are partitioned correctly by domain names then the identity of the person who initiated the download would still be in question; a host with access privileges might be used merely as a waystation for software headed towards a host in another domain. Another problem with this type of access control is that there are many "hacker tools" that could be used to break into or impersonate hosts from trusted domains.

Another problem with domain name based access restriction is that a browser can be set to use a proxy server to fetch documents, and the server doing the acccess restriction will only know about the domain name of the proxy, not the real user. If the proxy is in an allowed domain, then anyone can use the proxy to access files, unless the proxy does its own restriction.

Despite its weaknesses, access control by domain name is currently in use at MIT for distributing PGP [11], at Lucent Technologies for distributing Inferno [12], and at NCSA for distributed cryptographically-enhanced versions of NCSA httpd and Mosaic [13].

3.3.2 Username/password Access Restriction

When a user attempts to access a file that is protected by username/password access restriction, he or she is asked to enter a correct username and password before being allowed to download the file. In the case of an HTTP server, this type of access restriction is implemented by means of a configuration file which may be either global or directory-specific. In the case of an FTP server, accounts are set up for the allowed users on the file server machine, and access permission bits and ownership of files are set appropriately. With both HTTP Basic Authentication and commonly used FTP applications, passwords are sent over the network unencrypted. In HTTP MD5 Message Digest Authentication, the password is not sent over the network at all. Rather, a "digest" that is generated based on the password and other information about the request is hashed using MD5 and sent over the network. Digest Authentication is more secure over the network, but requires more rigorous security on the server machine, because the stored information cannot be encrypted with a one way function, whereas with Basic Authentication the server stores password using a one way encryption function.

[11] `http://web.mit.edu/network/pgp.html`
[12] `http://inferno.bell-labs.com/inferno/`
[13] `http://usa-only.ncsa.uiuc.edu:8080/`

3.3.3 Encryption Using Public Key Cryptography

With the public key cryptography method of access control, both the request for the software and the software itself are encrypted so that they cannot be read by anyone but the intended recipient. This method is intended to be combined with one of the public key authentication mechanisms described in section 3.2 – e.g., PGP, X.509, or SDSI. The request would take the form of a license agreement that the user signs using his public key to indicate agreement to the terms and conditions for using the software. The user's public key also identifies him or her so that the software server can check whether or not that user is authorized to obtain the software. The software itself would be best encrypted using a symmetric session key which would be generated for the purpose of this transmission only.

3.3.4 Options for Access Control Specification

When a user attempts to electronically download a piece of software, the access control mechanism must decide whether or not the user is authorized to access the software. This determination would be made by checking the user's certificate serial number against the software's access control list. Options for how access control lists could be managed include the following:

1. A separate access control list is maintained for each piece of software which lists the serial numbers of the certificates of users who are authorized to access this software.
2. Access control lists are maintained that apply to classes of software. Software would be placed into a particular class during the intellectual property assessment discussed in section 2. The classes could be the same or different across HPCC agencies.
3. Access control lists are maintained that list the classes of users that are authorized to access the software. The classes to which a user belongs would be determined at the time he applies for a certificate, or could be added later, and are attached to his certificate. The classes could be the same or different across HPCC agencies.
4. A combination of 2 and 3.

3.4 Options for Who Distributes the Software

There are the following options for who distributes NHSE software:

1. The software is distributed by the individual authors. This option places the most burden on the author. The NHSE could conceivably develop a tool that would automate user authorization and authentication and

the recording of the software transfer transaction as much as possible. However, distribution by individual authors would be less reliable and more error-prone than more centralized distribution.

2. The software is distributed from repositories run by HPCC agency programs (e.g., NASA/ESS, NASA Ames NAS Division).

3. The software is distributed from NHSE domain-specific repositories, such as PTLIB for parallel tools and systems software. This option has the advantage that the user need only go one place to find software in a particular domain. This option has the disadvantage that continued funding support for the external repository is required. Once the repository is established, however, this cost may be kept minimal, and may be less costly than the redundant effort involved with option 2.

4. The software is replicated on and distributed from a highly reliable and available set of NHSE server machines, which are distributed nationwide. To be most effective, this option should be combined with an automated name-to-location resolution system that resolves a location-independent name for a piece of software to a list of locations, with hints as to which location is likely to be the "best" (e.g., closest, fastest response time, or highest bandwidth). Such a name resolution system is currently under development at the University of Tennessee [14]. This option could also be combined with the encryption option discussed in 3.6.

3.5 Labeling with Legal Restrictions

The NHSE uses the Reuse Library Interoperability Group (RIG) Basic Interoperability Data Model (BIDM) for software catalog records. The BIDM is an IEEE Standard for software repositories and specific the minimal information about software that interoperating repositories should be able to exchange. The BIDM contains an attribute called "Restrictions" which includes copyright, patents, government rights, export restrictions, etc. The NHSE will use this field and expand it if necessary to clearly label software with its intellectual property rights and legal restrictions. Such labeling will allow users and importing repositories to be made easily aware of the restrictions that apply to a particular piece of software so that they do not inadvertantly violate them. Although copyright and patent notices usually appear in the software itself, additional labeling in the catalog record for the software alerts the user to information that might otherwise be missed, and can also be used for filtering prior to looking at the actual software.

[14] `http://www.netlib.org/utk/projects/rcds/`

3.6 Distribution of Encrypted Software

An alternative to restricting access to the actual software files is to encrypt these files and allow anyone to download them, but require authorization and authentication to obtain the decryption key. This approach is used in the Crytolope technology of IBM Infomarket [15]. In this case, the SSL protocol would be used by the server providing the descryption key to authenticate the user, sign a license agreement, and deliver the decryption key in a secure manner. A separate application would be needed to decrypt the software files.

4 Policy and Liability Issues

Distribution of HPCC software technologies requires special attention to a number of legal issues. Non-compliance with these issues can result in civil or even criminal actions. The NHSE has been working with legal counsel to identify these issues. Since the NHSE is not expert in these types of matters, definitive legal advice must come from outside counsel and from lawyers at the federal agencies. The role of the NHSE will be to provide technical methodologies which, according to the legal experts, will enable compliance with US laws and protect both the owners and the distributors of the technologies from prosecution.

Software to be distributed via the NHSE falls into one or more of several categories:

- Public Domain software technologies and documentation,
- Software requiring limited licensing restrictions, but still freely distributable within those restrictions,
- Commercial codes or other codes requiring fees, royalties or other types of payments. These codes typically require licenses stipulating stronger restrictions in copying, use, and redistribution of the software,
- Software technologies or documentation that fall under the control of Federal Export Regulations.

With the exception of public domain software, software distributions managed via the Internet must include one or more methods designed to:

- provide legally binding licensing restrictions, which when affirmatively acknowledged by the recipient (and possibly the licensor), are enforceable and provide adequate protection for the software developer and the software distributor [12];

[15] `http://www.infomkt.ibm.com/`

- accurately identify (authenticate) the recipient and provide access only to those technologies the recipient is legally allowed to receive;
- verify the integrity of the software technology being distributed;
- accurately log the transaction for future reference.

Although the NHSE will need to set some policies of its own to protect itself against liability, the NHSE will rely on the federal agencies to set policies as to what users are permitted to access what software.

4.1 Legal Admissability of Digital Signatures

Although in theory digital signatures are far more secure than physical signing, they have not been tested by court cases. Court rulings will determine which digital signature methods, key sizes, and security precautions are acceptable for a digital signature to be legally binding.

Efforts are underway to legislate the legality and use of digital signatures [13] [16]. Legislation creating procedures and support for digital signatures for public and private sectors use was passed in Utah in March of 1995, and similar legislation has been passed in California, Washington, and Florida, with other states to follow The National Institute of Standards and Technology has proposed the Digital Signature Standard, or DSS, as an algorithm standard for digital signatures. The NIST standard for generating keys and signatures is compatible with standards in the state legislation. A Government Accounting Office decision requested by NIST opined that digital signatures will meet legal requirements for valid contracts under federal law. The Department of Defense has notified NIST that DSS can be used by the Defense Department to sign unclassified data and, in some cases, classified data. The American Bar Association's Information Security Committee is developing model legislation for digital signatures for a compatible system for interstate commerce.

5 Proposed Secure Software Distribution Mechanism

The process of client/server certificate authentication is documented in some detail in the Netscape web pages [17]. Here is a brief outline of how the NHSE could use this technology:

1. The party requesting a software package fills out an HTML form which allows them to input whatever information is required by the Certification Authority, e.g. organization name, phone number, software requested,

[16] `http://www.SoftwareIndustry.org/issues/1digsig.html`
[17] `http://home.netscape.com/eng/security/certs.html`

etc. The new Netscape Navigator (version 3.0) supports an HTML tag, $< KEYGEN >$, which is imbedded in the HTML form. When submitting a form using this browser, the browser will generate an RSA key pair whose public key is digitally signed and sent to a cgi script. The browser passes this form's input off to a cgi script on the Certification Authority's machine and stores the private key in a local key database.

2. The cgi-script which runs on the Certification Authority's machine processes the input from the HTML form and places the request for a certificate into a queue. The queue is routinely checked and each request is then accepted or denied based on some set of policies set forth by the entity which controls the release of the requested software.

3. If the request was accepted, then the Certification Authority creates and digitally signs a certificate which can be used to download the requested software. The Certification Authority contacts the party who requested the software and points them to a URL to retrieve the new certificate. The new certificate contains the public key that was generated by the $< KEYGEN >$ tag.

4. The party who requested the software points his browser at the URL that was provided by the Certification Authority. The browser is notified that the incoming data contains a new certificate because it is specified by the Certification Authority's http server as being MIME type "application/x-x509-user-cert". The browser uses the public key encoded in the certificate to associate the certificate with the appropriate private key in its local key database. The certificate has now been installed in the browser.

5. When an HTTP server wants to require authentication based on client certificates, it uses the Secure Sockets Layer (SSL) protocol to negotiate the transfer between the browser and the http server. The HTTP server is set up so that it will only accept requests that are signed by a certain Certification Authority. When the server asks for a certificate, the browser is prompted to choose which certificate it wants to send. Depending on whether or not that certificate has been signed by the proper Certification Authority, access to the software is either accepted or denied.

Rather than issuing certificates on a per software package basis, the NHSE would most likely use one of the options for access control specification listed in 3.3.4. As soon as the Netscape Certificate Server becomes available, the NHSE developers will implement a prototype that demonstrates user authorization, authentication, and access to controlled software.

6 Issues Remaining to be Resolved

The following issues will need to be resolved before secure distribution of restricted HPCC software can be put into production mode:

1. Will restriction by domain name and voluntary identification of users suffice, or is secure user authentication required?
2. If the PGP technology is chosen, who will be responsible for signing users' public keys and maintaining the public key server(s)?
3. If the X.509 Certificate technology is chosen, who will serve as the Certification Authority(ies) who initially authorize and authenticate users and issue them certificates?
4. Who will distribute restricted software?
5. Will access control be placed on the actual software files or on decryption keys for unlocking encrypted versions of the software files?
6. Will digital signatures be acceptable for license agreements? If so, will the agreements need to be digitally timestamped?

After these decisions have been made, the NHSE will be able to incorporate the appropriate user authentication, access control, and license agreement mechanisms into the Repository in a Box toolkit. The NHSE will work with federal laboratories and agencies to provide technologies and methodologies for disseminating federally developed HPCC software technologies in a streamlined manner to an appropriate set of users, as determined by agency and/or laboratory policies and regulations.

References

[1] Uniform methodology for releasing NASA computer software. Commercial Development and Technology Transfer Division, NASA Headquarters, June 3, 1996 draft version.

[2] Department of Commerce Bureau of Export Administration: Export Administration Regulation; Simplificatin of Export Administration Regulations; Final Rule. *Federal Register*, 61(58), March 1996.

[3] Warren D. Siemens. Private communication, July 1996.

[4] Kenneth Rowe. Private communication, November 1996.

[5] Simon Garfinkel. *PGP – Pretty Good Privacy.* O'Reilly and Associates, Inc., 1995.

[6] Eric Grosse. Private communication, November 1996.

[7] Keith Moore, Shirley Browne, Jason Cox, and Jonathan Gettler. The Resource Cataloging and Distribution System. (Submitted for publication), November 1996.

[8] *CCITT. Recommendation X.509: The Directory - Authentication Framework*. 1988.

[9] Warwick Ford. *A Public Key Infrastructure for U.S. Government Unclassified but Sensitive Applications*. National Institute of Standards and Technology, 1995.

[10] Ronald Rivest and Butler Lampson. Sdsi - a simple distributed security infrastructure. http://theory.lcs.mit.edu/ rivest/publications.html, September 1996.

[11] S. Haber and W. W. Stornetta. How to time-stamp a digital document. *Journal of Cryptology*, 3(2):99–112, 1991.

[12] Fred M. Greguras, Trudy A. Golobic, and Rebecca Duncan. Software marketing, licensing and distribution in cyberspace. *Cyberspace Lawyer*, 1(3), June 1996.

[13] Brian Miller. How to sign on the digital line. *Government Technology Magazine*, June 1995. Accessible at `http://www.govtech.net/`.