

NIH-EDUCAUSE PKI INTEROPERABILITY PROJECT PHASE THREE PROJECT REPORT

Peter Alterman, Ph.D., Russel Weiser, Scott Rea, Deborah Blanchard ¹

Introduction

In 1998, in the Government Paperwork Elimination Act, the U.S. Federal government signaled its intention to move its transactions with citizens, businesses and other governments from paper-based applications to electronic applications accessible through the Internet. The Act required all Federal Agencies to convert paper-based transactions to electronic ones, or have plans to convert them, by October 23, 2003 and explicitly engaging the issue of electronic signatures for authentication and authorization. This first notice was followed by other statutes further empowering electronic identity management and e-business, notably the E-SIGN Act of 2000 and the E-Government Act of 2002.

In 2001, the U.S. Government identified a short list of applications to serve as leaders in implementing e-government services (the Quicksilver Project); among them were two cross-cutting services: enterprise architecture and e-authentication. This paper will focus and address the use of e-authentication. The purpose of the e-authentication program was, and continues to be, to provide electronic identity services to the 24 initial e-Gov applications and to the thousands of other government business processes that may eventually be brought on line. However, specific efforts to automate program applications were under way long before the government selected the projects that today comprise the core e-Gov activities.

The e-Authentication program initially focused on authenticating electronic identity credentials presented to the government for the purposes of authorizing citizen or business access to on-line government applications systems. A second, major category of electronic business transaction, electronically-signed electronic forms, has been acknowledged, but a concerted effort to fit it into the evolving e-authentication architecture has been on hold pending successful implementation of the first priority.

Notwithstanding the focus on access to online applications, since 2000 the Federal PKI Steering Committee has funded a project to develop models and the technology necessary to allow locally-issued digital certificates to be used to sign digital versions of government forms, and for the Federal government to be able to trust and validate those certificates. This project, [the NIH-EDUCAUSE PKI Interoperability Project](#), successfully demonstrated initial proof of concept in January, 2002 and again, using more sophisticated technology, in December, 2003. Since then, the e-authentication program and an increasing number of Federal agencies have recognized the Interoperability Project as the only successful model for implementing digitally-signed electronic forms

¹ Dr. Alterman may be contacted at altermap@mail.nih.gov; Mr. Weiser at rweiser@trustdst.com; Mr. Rea at srea@trustdst.com and Ms. Blanchard at dblanchard@trustdst.com.

processes that relies on federated identity management using X.509v3 digital certificates.

Authentication vs. Authorization

Properly managed PKI X.509v3 certificates are excellent authentication tools, especially for signing electronic documents. Given proper engineering design, they may also be useful tools to authorize access to online systems. For the purposes of Phase Three, we chose to preauthorize uploading of signed forms to the Automated Receipt Server to model secure processes found in many citizen- and business-to-government transactions. To do this, we created a small database in the Automated Receipt Server which was consulted when a signed document was presented for upload. This is a simple and straightforward method of linking authentication and authorization tools, but it has the drawback of requiring preauthorization, which can be burdensome and which adds a requirement to ensure that the same credential is always used to submit a form. A self-contained method of identifying authorization, using a business process-generated attribute, would ensure greater portability of credentials and simplify credential management at both the end user and subscriber sides. Other solutions are readily imagined.

GPEA Compliance

For the purposes of this phase of the Project, we chose to incorporate compliance with Government Paperwork Elimination Act (GPEA) requirements to demonstrate the ability of the signed forms model to satisfy statutory requirements for electronic government services. GPEA is the foundation law that requires Federal agencies to allow individuals or entities that deal with these agencies the option to submit information or transact business electronically with the agency, when practical, and to maintain records electronically, when practical.

Procedures have been identified for agencies to follow in using and accepting electronic documents and signatures, including records required to be maintained under Federal programs and information that employers are required to store and file with Federal agencies about their employees. These procedures reflect and are to be executed with due consideration of the following policies:

- a. maintaining compatibility with standards and technology for electronic signatures generally used in commerce and industry and by State governments;
- b. ensuring that electronic signatures are as reliable as appropriate for the purpose in question;
- c. maximizing the benefits and minimizing the risks and other costs;
- d. protecting the privacy of transaction partners and third parties that have information contained in the transaction;
- e. ensuring that agencies comply with their recordkeeping responsibilities for these electronic records. Electronic record keeping systems reliably preserve

the information submitted, as required by the Federal Records Act and implementing regulations; and

- f. providing, wherever appropriate, for the electronic acknowledgment of electronic filings that are successfully submitted.

GPEA defines electronic signature as a " . . . a method of signing an electronic message that:

- a. identifies and authenticates a particular person as the source of the electronic message; and
- b. indicates such person's approval of the information contained in the electronic message."²

This definition is consistent with other accepted legal definitions of signature. However, GPEA does not endorse one form of electronic signature over another, e.g., signing with a PIN versus using a digital signature. However, agencies and organizations are strongly encouraged to perform a risk assessment to determine which form of electronic signature best mitigates the risk to the agency. For the NIH-EDUCAUSE PKI Interoperability Project, the method utilized for digital signatures utilized X.509 digital certificates that were issued by the research institutions.

It is important to note the second part of the definition for an electronic signature, which requires a mutually understood, signed agreement between the person or entity submitting the electronically-signed information and the receiving Federal agency. Most often this can be accomplished by using a document referred to as a "terms and conditions" agreement. These agreements can ensure that all conditions of submission and receipt of data electronically are known and understood by the submitting parties. This is particularly the case where terms and conditions are not spelled out in agency programmatic regulations.

Products and Services

For Phase Three of the Project, the following products and services were used:

- Infomosaic SecureSign and Infomosaic SecureXML products as signing and validating tools for both the end user desktop and for the Automated Receipt Server;
- Certificate Arbitration Module (CAM) version 4.0, Release Candidate 4;
- Microsoft MSXNL 4.0 SP2 Parser and SDK;
- Microsoft IIS 6.0
- Microsoft Access 2002
- Persits Software AspEmail 4.5 Component

Participating colleges and universities used the following PKI CA products or services:

- Locally-developed CAs based on OpenSSL (two unique implementations);

² The Government Paperwork Elimination Act, section 1709(1)

- Locally-implemented iPlanet CA;
- Digital Signature Trust/Identrus-issued TrustID X.509v3 digital certificates;
- VeriSign-issued X.509v3 digital certificates from a local subordinate CA.

The Automated Receipt Server and the Federal government used ACES X.509v3 digital certificates issued by Digital Signature Trust/Identrus. The prototype Higher Education Bridge CA operated with the RSA Security Keon CA product, version 5.7p1. The prototype Federal Bridge CA contained the CA products from the following vendors: Entrust, RSA Security, BTrusted (formerly Baltimore Technologies), and Microsoft .Net CA.

Partners

In addition to EDUCAUSE, the following academic institutions are participating members of the Interoperability Project with the U.S. Federal government:

- Dartmouth College
- University of Alabama, Birmingham
- University of Wisconsin – Madison
- University of California, Office of the President
- University of Texas – Houston Health Science Center
- University of Virginia

For contact information at each of these schools, please check the Project website, <http://pki.od.nih.gov>.

Accomplishments

The accomplishments of the Interoperability Project may be divided between trust infrastructure development and PKI-enabling an electronic forms business process. During its tenure of operation, the PKI Interoperability Project has successfully created and demonstrated:

1. a certificate path discovery and validation infrastructure for assessing the legitimacy of digital certificates issued by a wide variety of PKIs and CA products;
2. an operational PKI bridge pathway between the prototype instance of the Federal Bridge CA and the prototype instance of the Higher Education Bridge CA, funded by and operated by EDUCAUSE, a nonprofit association whose mission is to advance higher education by promoting the intelligent use of information technology;
3. resolution of multiple certificate configuration and directory interoperability problems;
4. the ability for faculty and staff at academic institutions to download, fill out, sign (twice) and send XML forms to a U.S. Federal government Automated Receipt Server and to obtain an automated email acknowledgement of acceptance;
5. the ability of an Automated Receipt Server to acquire and test an XML version of a standard U.S. government form, SF-424, and to obtain an email acknowledgement of acceptance,

6. the ability of the Automated Receipt Server to automatically validate the affixed digital certificates, in the process discovering certificate validation paths to four different academic institutions using four different CA products and services through the Federal Bridge - Higher Ed Bridge pathway and to return a correct status to the server using the return path;
7. the ability of the Automated Receipt Server to send a digitally signed report containing a copy of the receipt and validation transaction, the certificates validated and a copy of the form to an audit log that satisfies the requirements of the U.S. National Archives and Records Administration for vouching for and archiving records of electronic transactions with the U.S. Federal government.

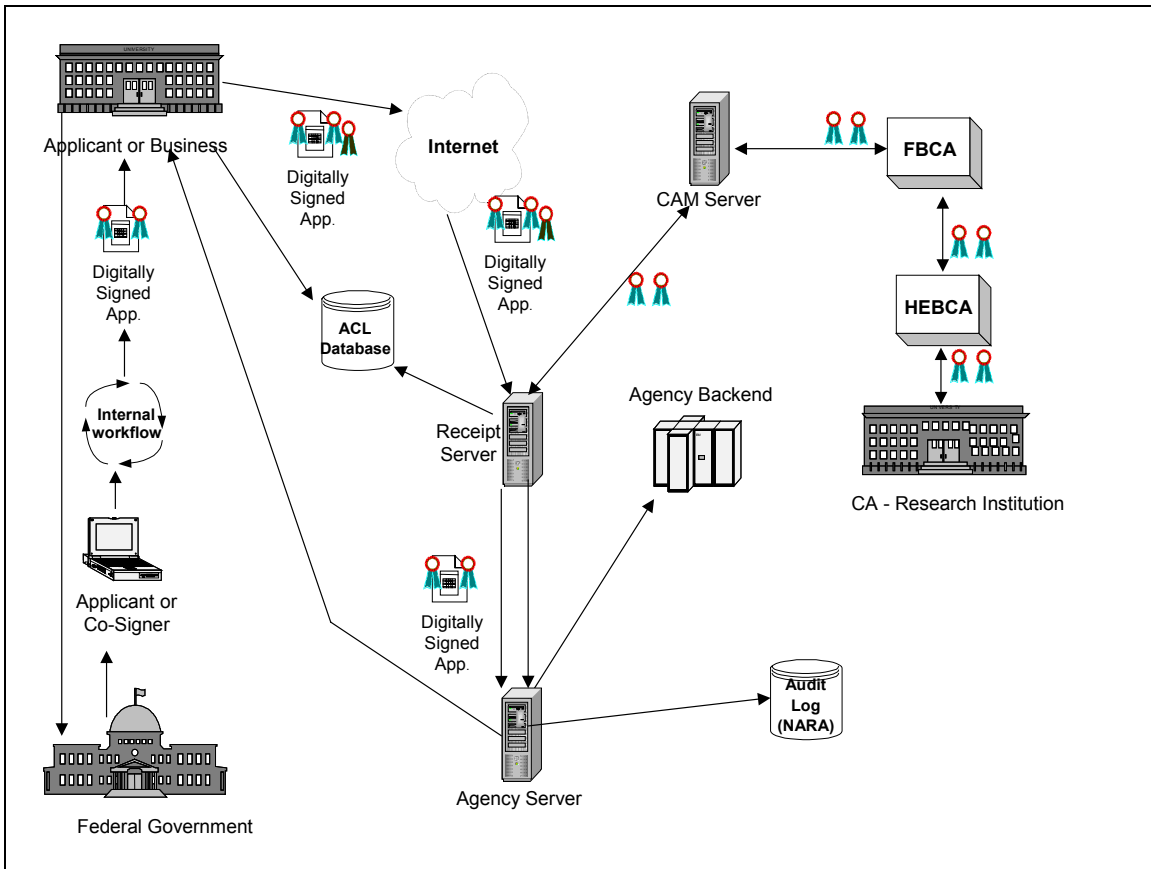
As a result of the success of the NIH-EDUCAUSE PKI Interoperability Project, a number of U.S. Federal Agencies are beginning to adopt all or part of its model to implement e-government business processes based on digitally-signed electronic forms. In addition, the Project has been recommended as a model for supporting the e-Forms initiative of the U.S. Federal government.

In the proposed Phase Four of the Interoperability Project, additional security and functionality elements will be added, especially automated parsing of the received XML form to a back-end database and encryption of all transactions within the model. Also, alternatives to the current validation tool will be evaluated.

Concept of Operations

A graphical representation of the Concept of Operations of the Interoperability Project – Phase Three appears below. Among the foundation assumptions are:

- that the document be standalone to satisfy document privacy and ownership requirements and to allow them to be managed by the end user in his or her unique environment;
- that all elements are standards-based to the extent possible at this time in the commercial environment (and that the least amount of unique code is developed and used);
- that electronic identity credentials (X.509v3 digital certificates) are issued by the institutions to which the end users belong;
- that the Federal government trust the institutional certificates at the test level of assurance;
- that, for purposes of this demonstration, one digital certificate will be preauthorized to upload the signed and completed XML form to the Automated Receipt Server, although many alternative scenarios for authorization are possible; and
- that a version of CAM 4.0 (see below for details) is the path discovery and validation tool used.



Form Conversion

To conform to emerging Federal standards for electronic forms, Phase Three replaced the Phase Two Microsoft Word template version of a PHS 398 “Application for Research Grant” with an XML version of an SF-424 “Request for Federal Funding.” The SF-424 was chosen because it was selected to be the foundation form for government-wide electronic grant applications by Grants.gov, and even though the Interoperability Project is completely separate from that project, using the same form demonstrated the broad applicability of digital certificate signing to cross-agency electronic government initiatives. The standard SF-424 has just a single signature for attestation, whereas the XML version used in this Project provides for signer and co-signer attestations. Therefore, a slight modification to the SF-424 was incorporated to demonstrate multiple signature capabilities.

The conversion of the SF-424 to *standards-based* XML was performed by mapping each requested data item in the original SF-424 form to an associated XML element and then constructing an XML schema logically patterned after the physical layout of the SF-424. The resulting schema consisted of a root element <Signed_Doc> with 3 complex-type sub-elements:

1. <TBS> (referring to the ToBeSigned portion of the document) contained all data elements being attested grouped into logical data records

2. <Signature1> the details about the first signer making attestation
3. <Signature2> the details about the co-signer making attestation

Once the schema for the Test XML SF-424 was completed, an investigation of available COTS based XML signing tools was conducted to determine the most suitable product for presentation of the Test XML form and execution of the required digital signatures. Note that the choice of signing tool was separate from the forms conversion process. Form creation and presentation/signing are standards-based, not product-specific. InfoMosaic's SecureSign Desktop was chosen (refer next section for details) for this aspect of the Project.

An InfoMosaic signature template was then designed to present the Test XML form to the user with a familiar HTML form interface. The signature template (*.tss file) is itself an XML file with a root element <SecureSignTemplate> containing minimally 6 complex type sub-elements:

1. <Header> where template name and identifying information are contained along with control elements for the form such as number of signers, hierarchy of signers and permissions for duplicate signers
2. <SchemaData> where an XML schema may be placed for validation purposes on the XML data (not used for the Project)
3. <XmlData> where the XML elements requiring data that is to be signed are defined (the entire Test XML document was encapsulated in a CDATA element here for the Project)
4. <HTMLData> where the presentation layer for the XML elements requiring data that is to be signed are defined. There is a complex type <HTMLData> element for each digital signer of the document (for the purposes of the Project there were two such elements created as individual HTML documents each encapsulated in a CDATA element). The purpose of having an <HTMLData> element for each digital signer is to allow for explicit enabling and disabling of HTML form fields in accordance with the different XML data population requirements for each signer.
5. <XPathData> where the respective elements to be signed are specified for each signer of the document (for the Project, the first signer attests the <TBS> and <Signature1> elements, while the co-signer attests <TBS>, <Signature1>, <Signature2>, and first signers signature [D-Sig]). NOTE: the co-signer is attesting the first signer's signature – this represents nested digital signatures and not just a peer signature to the original.
6. <HTMLMap> where the HTML form elements from the <HTMLData> element are mapped to XML data elements in the <XmlData> element. This allows the XML data values to be populated in the <XmlData> element from the corresponding HTML form data collected in the <HTMLData> element upon initiation of an appropriate event.

When the template is used with the InfoMosaic SecureSign application, the n-th user is presented with an HTML form created from the n-th <HTMLData> element of the template and their input is saved to the <XmlData> element based on the <HTMLMap> specified transformations upon initiation of an appropriate event (such as a Save operation). Control elements in the <Header> element determine how many signers there

are and whether file attachments are permitted. Control elements in the <XPathData> element determine whether digital signatures are nested signature or peer signatures. The SF-424 template for the Project is available from the Project.

When the InfoMosaic SecureSign application is used to provide template-based application of digital signatures, the <SignatureN> element (where N is the numeric value of the n-th signer) must be present and contain two child elements <SignatureDetailsNameN> and <SignatureDetailsDateN>. At the point of signing, the commonName attribute of the certificate to be used to verify the signature is populated in the <SignatureDetailsNameN> element and a corresponding timestamp (based on the local computer time) in the <SignatureDetailsDateN> element.

When the InfoMosaic SecureSign application is used to provide template based application of digital signatures, the template is base-64 encoded and included in the D-Sig XML output as a <SignedObject> element. Any document file(s) to be used as supporting documentation for the SF-424 submission may also be base-64 encoded and included in the D-Sig XML output as additional <SignedObject> elements.

The application of each digital signature appends a <Signature> element to the D-Sig XML output. The InfoMosaic SecureSign application can be configured to include CAM-based validation results for the signing certificate as an authenticated attribute of the signature. This is achieved by making a CAM validation request at the point of signing for the certificate chosen by the user to sign and including the response from CAM in the signature.

Signing Tool

From a number of COTS signing tools available, we selected Infomosaic SecureSign, because at the present time only SecureSign (in both desktop and server versions) implements the recently-adopted D-Sig standards for electronic document signing. In addition to being able to sign, or affix, multiple digital certificates to an XML form, or sign any file type supported by the operating system, SecureSign is able to read, display and write to templated XML forms. Thus, the same tool used to sign and validate the digital certificates is also the tool used to complete the form. This is a convenience, but does not preclude users from using other XML display applications. Information on InfoMosaic SecureSign may be found at www.infomosaic.net.

SecureSign is CAM-enabled and therefore is able to communicate with the CAM 4.0 Release Candidate 4 path discovery and path validation tool used to link certificate validation queries to the issuing PKIs through the FBCA-HEBCA mesh. (CAM is the Certificate Arbitration Module, created by the U.S. Federal ACES program; beta version 4.0 includes the Discovery and Validation Engine developed as part of Phase Two of this project by Mitretek Systems, Inc.) SecureSign was the only application discovered that provided for inclusion of certificate validation responses from CAM as an authenticated attribute of the signature. Including the CAM response as an authenticated attribute helps us to generate self-contained transactions that comply with the National Archives and Records Administration (NARA) requirements for processing and storage of electronic

records of electronic business transactions. SecureSign also currently supports native OCSP validation, SCVP validation and CRL-based validation of certificates.

The output from the InfoMosaic SecureSign Desktop application is a gzipped D-Sig XML document, making for compact storage and reduced bandwidth requirements when submitting to the Automated Receipt Server. Initially, the SecureXML server product only accepted input in the D-Sig XML format for verification and validation purposes. Due to the requirements of the project, InfoMosaic added gunzip functions to the server product so that the server could easily handle compact files submitted from the desktop product.

During the testing phase of the project, we discovered that some signed, text-based documents created for archival purposes by the SecureXML server could not be verified at a later date. Yet, when the signed document data that was being flagged as corrupt was compared to the original data used to create the document it appeared to be identical. We discovered that different character encoding was being used in creating and displaying the signed data than was used for the original data. To overcome this, InfoMosaic provided some base-64 encoding/decoding routines in their product and these were used to ensure that data remained in a consistent format by base-64 encoding the data prior to signature. This allowed for proper creation and verification of signed text-based documents.

For organizations that already have an HTML-based form and are looking to add digital signing capabilities, InfoMosaic also offers a template form designer product that will automatically create a SecureSign Template file (*.tss) for use with the Desktop product. This product was developed as a result of the template based signing capabilities incorporated in the SecureSign product for the purposes of this project.

Clearly, substantial product development by InfoMosaic was part of Phase Three. A brief summary of the modifications to SecureSign include:

1. Template based XML/HTML Form signing support in SecureSign;
2. CAM based certificate validation in SecureXML and SecureSign;
3. Development of SecureForm Designer product, a XML/HTML form designer for SecureSign;
4. Addition of gunzip feature in SecureXML;
5. Addition of various Base64 encoding/decoding APIs to SecureXML;
6. Netscape 7.X integration for signing with certificates in Netscape browsers;
7. Redesign of the SecureSign GUI making it more document oriented;
8. Master Cosigner concept - allowing a master cosigner to delete previous signatures and modify previously entered data in SecureSign;
9. Saving of unsigned documents for future completion and signing in SecureSign;
10. Signed document auto save feature in SecureSign.

Fully licensed copies of SecureSign for the desktop were provided to institutional participants. A fully licensed copy of the server version of SecureSign was employed by the Automated Receipt Server to enable automatic validation of received signed forms.

Bridges

The certificate trust infrastructure of the Project is based upon interoperation of two PKI bridges, the prototyping instance of the [Federal Bridge CA](#) and the prototype Higher Education Bridge CA. The bridge-bridge interoperability details have been published by NIST and in the report of Phase Two of the Project at the First Annual PKI R& D Workshop at <http://www.cs.dartmouth.edu/~pki02/Alterman/>.

Path Discovery and Validation Tools

Path discovery and validation through a bridge can be a complicated process – from the validation trust anchor up to the appropriate cross certificates, to another cross certificate and back down to the issuer of a the entity certificate being validated, including validating all of the certificates in the discovered path. This is further complicated by inherent issues with the bridge environments. In general, too many certificate extensions and too many certificate options make path processing too complex, too difficult, and too confusing. For example, too many options in the certificate profiles used by Bridge CAs (BCAs) introduce many complexities in the certificate path discovery and validation process. Another complication for a BCA environment is key rollover of the CA certificate and the creation and signing of CRLs. This also complicates path discovery and further complicates the validation process.

The path validation tool that was utilized in this Project was the Certificate Arbitration Module (CAM) that was enhanced with a discovery and validation engine (DAVE). CAM/DAVE was created as open source, “government-off-the-shelf” (GOTS) software. We used a beta version of this combined product known as CAM 4.0 Release Candidate 4. The package is comprised of the following public domain libraries:

- ◆ OpenSSL-0.9.7c from OpenSSL.org,
- ◆ SNACC ASN.1 Compiler created by DigitalNet,
- ◆ the Certificate Management Library (CML), version 3.2 created by DigitalNet for the Department of Defense, and
- ◆ Netscape Libraries and DLLs used for http-based or LDAP queries for AIA extensions and CDP extensions

Near the end of Phase Three, new protocols and commercial products have been announced that claim to be more robust than the CAM/DAVE validation toolset. We plan to test these in Phase Four of the Project. Additionally, updates to CAM 4.0, incorporating improvements to the CML, have recently been delivered and they, too, will be tested.

Directory

A discussion of directory issues was published in the Phase Two Report referenced above. At that time, we discussed the dependency of bridge environments on the ability to find and retrieve CA certificates, Cross Certificates, Certificate Authority Revocation Lists (CARLs), and certificate revocation lists (CRLs) to enable path discovery constructions and validation. Currently, the FBCA and HEBCA environments rely on the directory infrastructures for their proper operation. These environments differ. The FBCA Directory is based on the X.500 directory infrastructure (X.500 DSP protocol) to chain requests and knowledge reference information automatically to other external distributed X.500 directories that are participants within the FBCA environment. The requested objects are returned through the DSP protocol to the requesting directory back to the original requesting Path Discovery and Validation (PDV). The HEBCA, on the other hand, uses LDAP directories and LDAP V2 referrals to facilitate referral of the requesting client to another participant in the HEBCA environment via a URL-based process (smart referrals) to the actual institution's directory holding the needed PKI-based objects.

Both environments leverage LDAP V3 as the primary client access protocol for querying the directories, although the FBCA's X.500 directory also supports the X.500 Directory Access Protocol (DAP) clients for backward compatibility to agency applications. The HEBCA Registry of Directories (RoDs) leverages a rather simple and innovative use of an LDAP directory referral mechanism to provide centralized "smart referral" to the HEBCA participants' directories, which actually contain the PKI Objects need for PDV.

Both the FBCA and the HEBCA directories for this Project have been in use for several years (at least in the prototype BCA environment) and have had limited use. Several items of interest have come from the operation of these directories and the shifting perception of the how both models might play in BCA environments in the future. The FBCA has taken several steps to increase the FBCA flexibility. Most recently, the FBCA directory service was changed to the ISODE M-vault directory, which supports chaining LDAP referrals on behalf of PDV queries to the FBCA directory. This opens up the possibility of the FBCA directory containing smart referrals while still allowing X.500 queries to the FBCA entries with referral to be resolved on behalf of the requesting PDV.

Certificate chaining and locating objects in these directories is easier if a digital certificate utilizes the *AIA* extension. Without an *AIA* extension in the certificate, the issues related to chaining and locating objects become significant. The certificate profiles needed to support this more generic method of finding caCertificates and crossCertificatePair should include the use and population of the *AIA* extension. Very little software makes use of the *AIA* extension; however, DAVE and CAM both use the *AIA* extension if it is present. If an HTTP URL form is present, DAVE will bypass directory lookups and use HTTP directly. If an LDAP URI form is presented to DAVE, the module directly queries the given LDAP server for the given distinguished name (DN) and associated attributes and values; the same logic applies for URL-based CRL distribution point (CDP) fields to retrieve CRLs and ARLs.

An example of AIA and CDP is highlighted in the following hierarchy³.

Root CA (straight SubjectDN retrieval)

- *SIA extension* → URL=
ldap://ldap.trustdst.com/cn=DST ACES Root CA X6,ou=DST ACES,o=Digital Signature Trust,c=US?cACertificate;binary,crossCertificatePair;binary

SubCA cert

- *AIA extension* → URL=
ldap://ldap.trustdst.com/cn=DST ACES Root CA X6,ou=DST ACES,o=Digital Signature Trust,c=US?cACertificate;binary,crossCertificatePair;binary
- *CDP extension* → URL=
ldap://ldap.trustdst.com/cn=DST ACES Root CA X6,ou=DST ACES,o=Digital Signature Trust,c=US?certificateRevocationList;binary,authorityRevocationList;binary

EE cert

- *AIA extension* → URL=
ldap://ldap.trustdst.com/cn=DST ACES Unaffiliated individual CA A3,ou=DST ACES,o=Digital Signature Trust,c=US?cACertificate;binary
- *CDP extension* → URL=
ldap://ldap.trustdst.com/cn=DST ACES Unaffiliated individual CA A3,ou=DST ACES,o=Digital Signature Trust,c=US?certificateRevocationList;binary

Registry of Directories

The Registry of Directories (RoD) is a centralized list of pointers to other directories within an Internet domain. First fielded by the Internet2 Middleware Initiative for the U.S. National Science Foundation, we created a RoD for the “.gov” space and connected it to the existing Internet2 RoD used for the “.edu” space. Each RoD contains pointers to PKI directories that support the participating CAs in government and higher education. For more information on Registries of Directories, see the Phase Two Project Report referenced above.

Open Issues for the Registry of Directories

- The referral URI used in the smart referrals of the RoD must be pre-escaped. In other words, adherence to the URI definition rules must be strictly followed such that space characters must be translated to the %20 in the URI.
- Referral management will require institutional administrators to be aware of changes to the local directory tree that could affect RoD smart referrals. The LDAP Browser/Editor version 2.8.2 by Jarek Gawor was utilized for the creation of the smart referrals in the RoD. This version of the LDAP Browser/Editor was

³ Note that the AIA and CDP may simply be populated with HTTP based URLs in the case of HTTP AIA a .p7b file works for CA and cross certificate pair certificate as separate der encode binary certificates. The HTTP base CDP would have a .crl file of the CRL.

used as the native administration interface of the directory server was found to be cumbersome.

It would probably be wise to write a simple tool or script to provide a subjectDN, Institutional Directory IP address and Port. This tool could then be used to build the Smart Referral as a LDIF file that would easily be imported in the correct RoD. This would simplify the management of the RoDs and reduce errors.

Archive

One of the requirements of the Project is to archive transactions in compliance with NARA guidelines. These state, "If an electronically signed record needs to be preserved, whether for a finite period of time or permanently, then the agency needs to ensure its trustworthiness over time."⁴ Reliability, authenticity, integrity and usability are the characteristics used to describe trustworthy records from a records management perspective. Each of these characteristics was considered when implementing the archive strategy for this Project:

- ◆ Reliability – a reliable record is one in which content can be trusted as a full and accurate representation of the transactions, activities, or facts to which it attests and can be depended upon in the course of subsequent transactions or activities. To meet the goal of reliability, the Project implementation creates an XML archive record consisting of the submitted form, validation responses on each of the signing certificates, signed with the Receipt Server's own archive certificate and including a timestamp as an authenticated attribute.
- ◆ Authenticity – an authentic record is one that is proven to be what it purports to be and to have been created or sent by the person who purports to have created and sent it. To meet the goal of authenticity, the Project implementation utilizes the PKI reliance properties of the signing certificates included in the transaction as proof of origin and the PKI reliance properties of the server's own certificate as proof of acceptance into the system.
- ◆ Integrity – the integrity of a record refers to it being complete and unaltered; it is necessary to protect records against alteration without appropriate permission. To meet the goal of integrity, the Project implementation utilizes the PKI integrity properties inherent when digital signatures verify.
- Usability – a usable record is one that can be located, retrieved, presented and interpreted; any subsequent retrieval and use of the record should imply a direct connection to the business activity that created it. To meet the goal of usability, the Project implementation utilizes an appropriately indexed database to store the digitally signed archive records.

The Project utilized a Time-Contextual approach to ensure the trustworthiness of its electronically signed records over time. This was done by maintaining adequate documentation of the record's validity, such as trust verification records (signature verification and certificate validation), gathered at or near the time of record signing. This is achieved by including the CAM response in the archive documents as an authenticated

⁴ Records Management Guidance for Agencies Implementing Electronic Signature Technologies – NARA Modern Records Program, Office of Records Services, October 18, 2000

attribute of the archive XML document's signature. This approach was preferred to a Time-Independent approach since the Time-Contextual approach is less dependant on technology and much more easily maintained as technology evolves over time for records that have permanent or long-term retention requirements.

The Project implemented the following steps to ensure trustworthy records relating to electronically signed transactions:

- Created and maintained documentation of the systems used to create the records that contain the signatures
- Ensured that the records were created and maintained in a secure environment that protected them from unauthorized alteration or destruction
- Implemented standard operating procedures for the creation, use, and management of these records and maintained adequate written documentation of those procedures.
- Ensured electronically signed SF-424 trustworthiness by the following:
 - Stored the original digitally signed SF-424 form in the archive XML document
 - Digital signature on archive XML document included authenticated timestamp as part of the signature
 - Archive XML document included digital certificate for verification purposes for each signatory on the original digitally signed SF-424 form
 - Archive XML document provided for signature verification at any time for each signatory on the original digitally signed SF-424 form
 - Archive XML document included certificate validation result (from CAM) for each signatory on the original digitally signed SF-424 form and the receipt signer's own certificate validation result and an authenticated attribute of it's signature
 - Long-term integral storage of all of the above items will be achieved by optical media back-up of the archive database.

Email Receipt to the Submitter from the Server/Recipient

In support of GPEA requirements, one of the objectives of the Project was to have the SF-424 form received via the Automated Receipt Server with an automated process for verifying the signatures, validating certificates, and confirming the receipt of SF-424 to the submitter, and the signatories via an e-mail message. The following mandatory requirements for automated receipting, verification, validation, and notification were addressed:

- 24 hour real-time processing of submitted SF-424;
- SF-424 forms processed in real time as they are uploaded to the Automated Receipt Server;
- Submission received confirmation via HTML response message and email;
- Automated digital signature verification;
- Automated Certificate Validation via CAM;

The Automated Receipt Server is a web-based service that allows registered users to connect and upload their Test SF-424 XML forms. If the submitted document is not the

correct format, or if the submitter is not the co-signer on the document, then the submission file is rejected. The Automated Receipt Server uses email to send file upload receipts to each of the Test SF-424 XML Signatories – assuming that their email address is contained in the DN of their certificates; else only to the registered submitter - when processing of the submission is complete.

An administrator email account was created for the Project on the NIH mail server, allowing the Automated Receipt Server to create emails from a valid NIH email address. Access to the Administrator email account is assigned to the NIH personnel responsible for administering the Project site.

The Applicant's certificate and Co-signer's certificate are extracted from the submitted SF-424 document by the SecureXML server without any human intervention. The respective signature blocks are then verified utilizing the corresponding public keys. Once a signature is verified, the respective certificate is handed to the CAM for certificate validation. Upon successful verification of the signatures, and validation of the associated certificates, a confirmation e-mail message is sent to the certificate holders (providing there is an email attribute in the signing certificate subject DN) using the e-mail address contained within the digital certificate. If no email address is found in the certificates subject DN, a confirmation e-mail message will be sent to the registered e-mail address of the submitter stored in the access control list.

Certificate-Based Access Control

The Project used a web-based upload function to allow Project participants to submit forms. A certificate-based access control list and SSL Mutual Authentication process control access to the Project site submission service. Authorized Test SF-424 form co-signers are the only individuals who may submit forms and they are required to be pre-registered in order to do so. As a part of the registration process, the prospective submitter will be asked to nominate which certificate they will use to authenticate themselves (it must be the same certificate they use to co-sign their documents) and it is recorded in the Access Control List. The Project was originally designed for an Administrator to review all registration requests and approve or disapprove the registration, but currently operates in an auto-approve mode on all registration requests.

When a registered submitter connects to the submission service they are asked to authenticate themselves via 128-bit SSL Mutual Authentication. If the certificate chosen by the submitter for mutual authentication is not from one of the NIH trusted PKIs (FBCA cross-certified, HEBCA cross-certified, ACES or TrustID PKI), the SSL server trust list is configured for each participating Institution by installing and trusting the appropriate certificate chain for their end entity certificates, then their connection to the service is rejected. After a successful connection and a local file are submitted, the server checks the format of the file: if it is not a signed SecureSign XML document, then it is rejected. If the document is in the correct format it is saved to the server cache and the document is verified and parsed to extract the two signer's certificates. If the document does not verify or if the name of the co-signer on the document does not match the name

of the submitter, then the user is notified and the document rejected. If the co-signer is the submitter, then the two signing certificates are validated via the CAM. If either certificate fails validation, then the document is rejected. If the certificates validate, then an archive XML record is created and the transaction recorded.

Each time a new school wishes to participate in the Project, it is necessary to obtain the certificate chain for the end entity certificates that the Authorized co-signers will use to register with and subsequently sign and submit files. This is necessary because the access control mechanism for the Project site is certificate based and the web server hosting the Automated Receipt Server needs to trust the end entity chain for SSL Mutual Authentication so that the Authorized Co-signer is presented with the option to select their certificate from the list presented by the browser when they connect to the Automated Receipt Server site. If the chain is not installed and trusted on the web server, then the user will not have the option to select their certificate for authentication and will be unable to connect.

Two Digital Signatures on the Form

In a continuation from previous phases, multiple digital signatures were utilized, validated, and verified for the form submittal. However, the SF-424 in a production state only requires one signature. Since in this phase we were not building a true production system, we had the flexibility to add as many signature blocks as necessary. We chose to require two signatures as a proxy for multiple signatures, to extend the model to as great a degree of flexibility as possible.

We also incorporated limited rules around the digital signatures on the form. First, the digital certificates used for digital signing were required to be different. In other words, a participant may not use the same digital certificate to create the two digital signatures. Second, the digital certificate used for digital signing was validated and verified during submission and optionally during the digital signing ceremony. This ensured for the subscriber that the digital certificate being used was valid and that neither had it expired nor been revoked. Finally, the person submitting the SF-424 must be the second signatory and must have been registered in the ACL Database, as discussed above. Registration in the ACL Database was done using a valid digital certificate.

These design decisions were made to demonstrate that using digital certificates for digital signing, for access control, and for workflow is viable in many situations. Additionally, we were able to demonstrate that digital certificate technology can satisfy fully the business rules required for government business processes.

Next Steps

Phase Three of the Interoperability Project successfully demonstrated proof of concept in December, 2003 at the EDUCAUSE offices in Washington, D.C. Since then, several different academic participants have run the demonstration independently in a variety of venues, proving that the model is successful for school-school and agency-agency

business transactions using electronic forms signed with multiple digital certificates. Phase Four is planned to enhance the security of the model and includes designs to parse the signed form into a back end form automatically, to prove the functionality of implementing digitally signed electronic forms into a larger electronic business process scheme.

Acknowledgements

Grateful appreciation for their participation in the pilot project is acknowledged to: Clair Goldsmith, University of Texas System; Jill Gemmill, University of Alabama at Birmingham; Keith Hazelton, University of Wisconsin-Madison; Eric Norman, University of Wisconsin-Madison; Robert Brentrup, Dartmouth College; Ed Feustel, Dartmouth College; David Wasley, University of California Office of the President; Bill Weems, University of Texas – Houston Health Science Center; Barry Ribbeck, University of Texas – Houston Health Science Center; Mark Luker, EDUCAUSE; Steve Worona, EDUCAUSE; Debb Blanchard, Identrus/Digital Signature Trust; Andrew Lins, Mitretek Systems; Scott Rea, Identrus/Digital Signature Trust; Russ Weiser, Identrus/Digital Signature Trust; Manoj K. Srivastava, Infomosaic; Judy Spencer, Chair, Federal Identity Credentialing Committee and Tom Turley and Frank Newman of the NIH Office of Extramural Research.

References

1. The Government Paperwork Elimination Act, section 1709
2. “Implementation of the Government Paperwork Elimination Act”, Office of Management and Budget, <http://www.whitehouse.gov/omb/fedreg/gpea2.html>
3. Final Report: EDUCAUSE – NIH PKI Interoperability Project Project, Prepared for National Institutes of Health (NIH) Office of Extramural Research (OER), Under Contract No. GS00T99ALD0006, May, 2003
4. Memorandum to the Heads of all Departments and Agencies, “E-Authentication Guidance for Federal Agencies”, M-04-04, Joshua B. Bolton, December 16, 2003
5. PKI: Implementing and Managing E-Security, Nash, Duane, Joseph, and Brink, McGraw-Hill Publishing, 2001
6. Planning for PKI, Housley and Polk, John Wiley and Sons, 2001
7. Records Management Guidance for Agencies Implementing Electronic Signature Technologies – NARA Modern Records Program, Office of Records Services, October 18, 2000
8. Report: EDUCAUSE - NIH PKI Interoperability Pilot Project, Peter Alterman, Russel Weiser, Michael Gettes, Kenneth Stillson, Deborah Blanchard, James Fisher, Robert Brentrup, Eric Norman, 1st Annual PKI Research Workshop, <http://www.cs.dartmouth.edu/~pki02/Alterman/>.