Introduction to Validation for Federated PKI

This paper investigates the security, performance and availability issues associated with establishing trusted identities for large populations spanning many different trust domains. A trusted path building approach is presented that scales to 100s millions of users with transaction response times of a few hundred milliseconds.

Introduction to Validation for Federated PKI

Introduction

Trusted identity is central to all security. This statement has never been more applicable than it is today. Electronic transactions over the Internet are now commonplace and no longer considered the wave of the future.

For government and industry, this brings the challenge of establishing trusted identities for large populations that are certified by many different trust domains. The use of public key technology and digital certificates has emerged as the preferred approach. Both commercial enterprises and government agencies are deploying Public Key Infrastructures (PKIs) to meet this challenge. The promise of PKI is that it can facilitate establishment of trusted identities for huge populations. This promise will only be realized if the building of trust paths between different trust domains can be automated in a scalable, affordable, and manageable way.

The initial focus of every enterprise rolling out a PKI is typically on the issuance of identity credentials. While the vetting of credential recipients and the secure issuance of these credentials is a formidable task, there are significant issues associated with the actual use of these credentials. Validation of these credentials in a timely manner for every transaction has proven to be a formidable challenge. Then, once an enterprise has successfully deployed a PKI for its own internal use, the inevitable question arises as to how to extend their PKI to include external partners, collaborators, supply chain vendors, sister agencies and foreign governments. The answer lies in setting up trust relationships among different trust domains in a manner that allows the verification of trust to be performed in a secure and timely manner. Inherent in this approach is trusting partners to authenticate their own users according to previously agreed upon vetting and credential issuance policies.

This paper explores the role of automated credential authentication and validation in achieving trust in a federated PKI environment. The key word here is "automated". The success or failure of a federated PKI, as with any PKI, lies in the end user's experience. If the deployed PKI is to be successful, users should not experience delay or difficulty in executing a secure transaction at the moment they wish to so. Critical to this success is the ability to scale the verification of trust relationships through automated credential validation.

Establishing Trust in a Hierarchical PKI

For the purposes of this discussion "trust" is defined in terms of the authenticity and validity of an identity credential, specifically a digital certificate:

- **Authenticity** – is the quality of being genuine. An authentic credential is one that has not been forged or tampered with.
- **Validity** – refers to the state of the credential. A valid credential is still in force, is legally binding and has not been revoked.

A trusted credential is therefore one that is both authentic and valid at the time of the transaction. Establishing that a credential can be trusted is a two step process.

In order to understand how trust is established in a federated environment it is essential to fully understand the basic mechanisms used in simple hierarchical PKIs. A typical hierarchical PKI is shown in Figure 1:

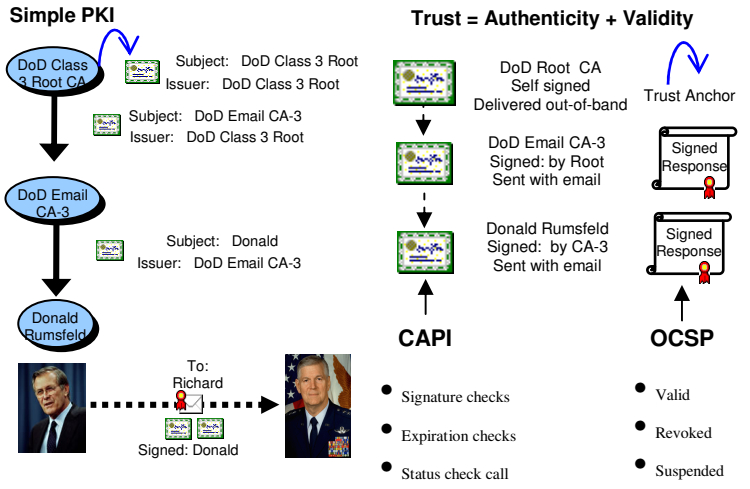

Figure 1: Simple PKI *Establishing trust for a secure email transaction in a simple hierarchical PKI. CAPI checks the authenticity of the certificate chain and checks the status of the certificates it receives as part of the transaction. Both the sender and receiver are members of the same PKI, the US Dept. of Defense (DoD) PKI.*

In this example we show a signed email transaction between two members of the same PKI. The PKI consists of a Root CA and a Subordinate CA. There are three digital certificates present, one from each CA plus the end user certificate which is used to establish identity. This is considered a "hierarchy" because each entity in the hierarchy certifies the entity below it. The Root CA is an exception in that it is self-certified. This means that the Root CA used its own private key to sign its certificate. The Root CA also certifies the CA below it in the hierarchy by signing the Subordinate CA's certificate. The Subordinate CA certifies the end user by signing the end user's certificate. In a typical deployment the Root certificate is delivered out-of-band (i.e., not as part of the transaction) in a trusted manner to all members of the trust domain served by the PKI. This is necessary because the integrity and authenticity of the Root certificate is not protected since it is self-signed.

As shown in Figure 1 the Subordinate CA certificate and the end user certificate were delivered to the relying party as part of the transaction (e.g., along with a digitally signed email). Before trusting the transaction the relying party client (i.e., the recipient of the signed email) must make several security checks, including checking that the certificates it received are both authentic and valid. We have assumed that the client is using Microsoft's Cryptographic API (CAPI) software to perform these checks. In this case CAPI explicitly trusts the Root certificate since it was registered as a trust anchor as part of the out-of-band delivery process. CAPI uses the public key from the Root certificate to verify that the Subordinate CA certificate is authentic (see Figure 2). In a similar manner CAPI verifies the authenticity of the end user's certificate by using the public key from the Subordinate CA's certificate. Finally the signature on the transaction itself is

checked by using the public key from the end user's certificate. This completes the authenticity checking.

CAPI must still check that none of the certificates have been revoked (the Root certificate is assumed to be valid). This is done by retrieving either a Certificate Revocation List (CRL) or an Online Certificate Status Protocol (OCSP) response.[1] If all the certificates are valid and all signatures are authentic the transaction is considered trustworthy. [2]

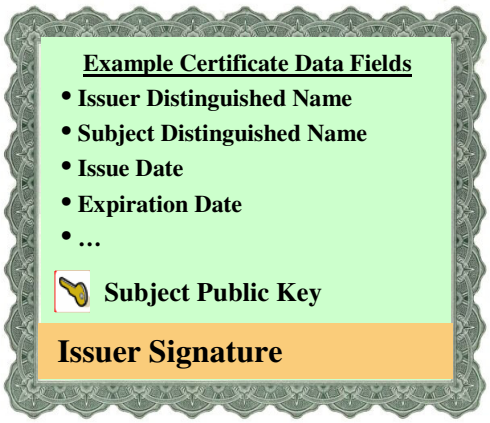

Figure 2: Digital Certificate Content
Certificates contain the subject's public key which can be used to verify signatures made by the subject's private key. The certificate issuer's signature binds the subject's public key to the subject identity and other data fields in the certificate such that they cannot be forged or altered. The certificate issuer's signature is checked using the issuer's public key.

Establishing Trust in a Federated PKI

A federated PKI is by definition a collection of independent hierarchical PKIs, each serving separate trust domains, each with their own Root CA. The federated PKI is established when members of the federation agree on a common set of policies for the purpose of conducting secure transactions among them. Implementing trust across multiple domains is accomplished through the incorporation of a Bridge Certification Authority.[3] The primary function of a Bridge CA is to establish trust relationships among members of the bridged community. This is accomplished by setting the security policies and practices that are adhered to by all bridge members. Bridge CAs are also involved in the issuance of cross-certificates which are used in building trust paths of certificates from a trust anchor in one PKI to an end entity certificate issued by a different PKI. Several examples of PKI bridge communities exist today, including the European Union Bridge, the Pharmaceutical Bridge (SAFE – Secure Access For Everyone), the CertiPath Bridge (aerospace and defense agencies)[4], the US Federal Bridge, and the Higher Education Bridge (EDUCAUSE). Figure 3 depicts potential members of a generic bridge community.

---

[1] For a discussion of validation approaches see "Distributed Certificate Validation", CoreStreet, 2003. Available from http://www.corestreet.com

[2] For completeness we note that there are some additional checks made as part of the validation process such as ensuring the certificate has not expired, that the names chain properly and that the intermediate certificates are CA certificates. These checks are not discussed here since they are unaffected by federated usage.

[3] While there are other trust models for establishing trust across multiple domains the Bridge CA approach has significant operational advantages making it the preferred choice. A discussion of the use of "trust lists" is presented in Appendix A.

[4] The Trans-Atlantic Secure Collaboration Program (TSCP) is actually extending the federation model a step further by establishing trust between different bridge communities.

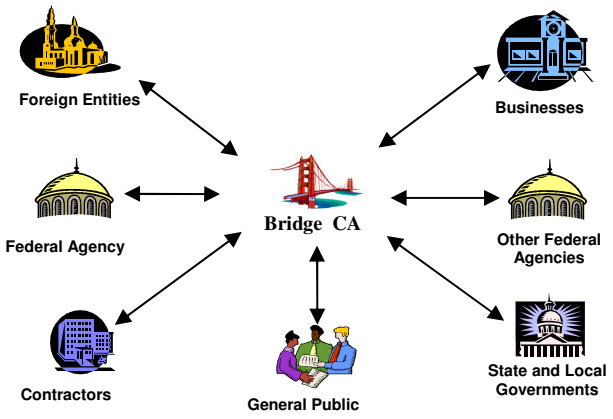

Figure 3: Bridge Community
A PKI bridge community is comprised of multiple trust domains served by independent hierarchical PKIs, each with their own Root CA.

To understand how a Bridge CA is used to provide trust paths within a federated PKI consider the following example. Figure 4 depicts a signed email transaction between members of two separate trust domains. In this example, Donald Rumsfeld, who is a member of the US DoD PKI, sends a signed email to Condoleezza Rice, who is a member of the US State Department PKI. (Condoleezza's trust anchor is the State Department Root. Explicit trust of this Root is achieved by registering the State Department Root certificate on her desktop.) Since she is not a member of the DoD PKI she does not explicitly trust the DoD Root certificate and therefore no complete path of trust exists for this transaction (i.e., there is no way to establish the authenticity of all the certificates in the chain). Consequently, when CAPI is presented with the task of checking the trustworthiness of this transaction the authentication check fails and the result is "not trustworthy".

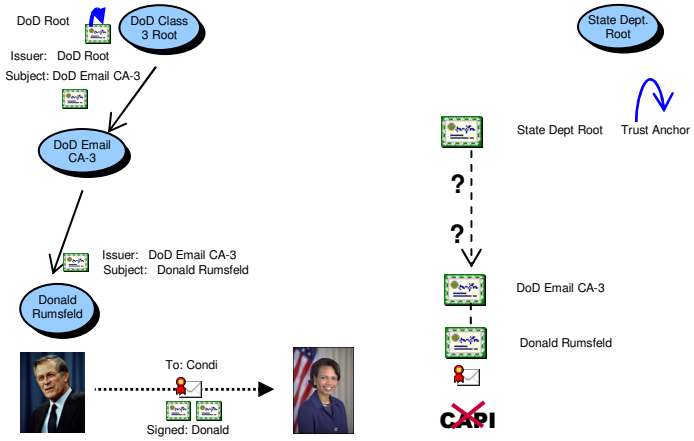

Figure 4: Trust Building
An example of establishing trust for a secure email transaction in a federated PKI. Here the sender and receiver are members of separate trust domains with different PKI trust anchors. Because the certificate chain is incomplete the authenticity check by CAPI fails.

The establishment of a trustworthy path from one PKI to another is accomplished through the use of a Bridge CA. Among the functions facilitated by the inclusion of a Bridge CA is that of providing cross-certificates. Cross-certificates are what "bridge the gap" between different PKIs. For example, they provide a means for the public key of a CA in one trust domain to be certified by a CA in another trust domain. There are no explicit flags identifying these certificates as cross-certificates. Cross-certificates are more likely to have extensions (e.g. policy mapping) that are uncommon in a hierarchical PKI but

otherwise they are basically identical to subordinate CA certificates.[5] For example, in Figure 5 the US Federal Bridge CA (FBCA) has certified the DoD Root CA by using its private key to sign a cross-certificate containing the DoD Root public key. This has the effect of allowing the DoD chain of trust to extend through the FBCA rather than terminate on the DoD Root certificate.

The trust path to the State Department Root is completed using a cross-certificate issued by the State Department to the FBCA. This cross-certificate certifies the FBCA's public key and therefore any certificates issued by the FBCA will be accepted as authentic by anyone who uses the State Department Root as a trust anchor. Therefore the cross-certificate issued by the FBCA to the DoD can be checked for authenticity by using the FBCA's public key contained in the cross-certificate signed by the State Department. The entire chain of certificates can be processed in this manner and will now pass the authenticity checks. There is no change to the process of validating the status of these certificates.

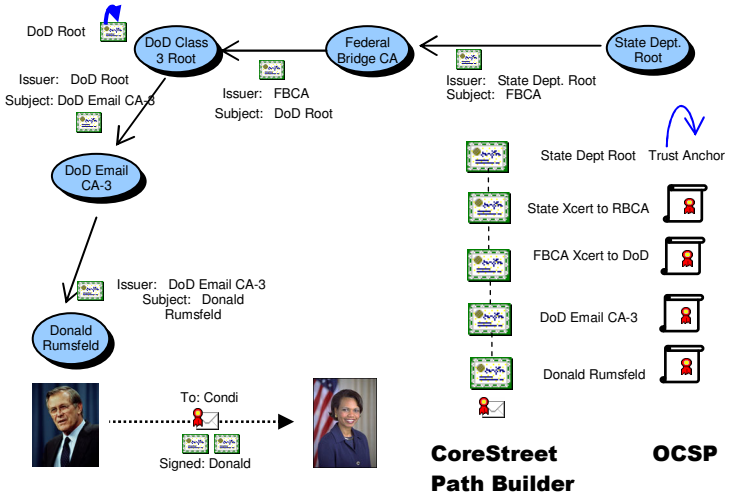

Figure 5: Bridging the "gap" *The certificate chain of trust is completed through the use of a Bridge CA. However non-CAPI software such as* CoreStreet*'s* Path Builder *is required to discover the complete trusted path.*

Even though a complete and trustworthy path now exists, there remains the issue of "discovering" the missing certificates. In general cross-certificates will not be included as part of the transaction or locally available for use by CAPI to perform full path validation. Consequently transactions of this type will always fail to be authenticated by CAPI. It is important to note that this failure of CAPI to authenticate transactions involving cross-certificates is not because there are any different or more complicated checks required to establish trustworthiness for federated PKIs. On the contrary, the required authentication and validation checks at the desktop are exactly the ones used for simple hierarchical PKIs. *The failure stems from the fact that CAPI does not possess the general capability to search external sources for the missing certificates in the chain.*[6]

[5] Cross-certificates do not need to be at the Root CA level. However for this discussion we will assume they are.

[6] CAPI does have the ability to discover paths if all the certificates in the chain have the AIA extension populated. However, this is not the general case. In addition, having each relying party application crawl

CAPI's shortcomings require an alternative solution to be deployed at the desktop in order to facilitate path discovery in federated environments.

Applying Validation Policies

One of the major advantages of using the bridge CA trust model is that it enables centralized management and automatic enforcement of validation policies. Cross-certification establishes a mechanism for building trust paths from one trust domain to another. Validation policies can be used to limit the scope of the trust relationships that are established through cross-certification. Enforcement of these policies at the time of the transaction, allows secure, trusted, business processes to be established among organizations participating in the federated community.

Validation policies contain the specific rules and parameters to be used when validating a certificate. In the Bridge CA model these can be implemented through the use of policy and/or name constraints present in the cross-certificates.[7] Policy constraints are used to constrain certificate usage based on the policies under which that certificate was issued. For example, a given trust domain (Trust Domain A) may issue certificates under two policies which differ according to the level of identity assurance achieved through vetting of the individual to whom the certificate is issued. Low assurance certification may only require that the individual supply his/her identity information in an email. High assurance certification may require in-person application and a background check as part of the vetting process. Another bridge member, Trust Domain B, may want to limit the trust relationship with Trust Domain A to individuals for who this high assurance level of identity vetting has been applied. This can be accomplished by asserting a "high assurance" policy constraint in the cross-certificates which ensures that only high assurance certificates are included in the trust paths. A third bridge member, Trust Domain C, may choose to accept both high assurance and low assurance certificates from Trust Domain A.

Bridge community members may also want to limit the trust relationships with another member to specific sub-domains or to exclude specific sub-domains. This can be accomplished by listing the names (i.e., the X.500 distinguished names) from these sub-domains in the name constraint extension in the cross-certificates. The capability exists to either include or exclude specific names or subsets of names (e.g., everyone from abc.gov address) through this mechanism.

The major benefit of using the bridge CA approach for establishing trust relationships is that the process of building trust paths between the domains, including the enforcement of path constraints, can be automated. Once agreement is reached on what validation policies to implement, these policies are codified in the cross-certificate extensions. This facilitates automatic electronic validation processing for all subsequent transactions without the need to review the agreement terms for each transaction. Automated

---

all the way back to a trust anchor is not a scalable approach, a fact that has prompted the IETF to draft a new protocol as discussed in the *Building Trust Paths* section of this document.

[7] Part of the process of establishing trust relationships between members of a bridge community includes the agreement as to the validation policies to be implemented.

transaction processing is essential to making electronic commerce a reality for worldwide populations that are certified by many different trust domains.

Building Trust Paths

Building trust paths in a hierarchical PKI serving a single trust domain, as shown in Figure 1, is a simple process. Discovering trust paths which cross trust domains, as shown in Figures 4 and 5, is much more difficult. Fundamental to this issue is the problem of finding the correct repositories of the missing path links when cross-certification is used. The IETF has recognized that discovering the missing links and building appropriate trust paths is too difficult and time-consuming to rely on Public Key Enabled (PKE) applications to execute. This has led to the establishment of a new protocol called Simple Certificate Validation Protocol (SCVP).[8]

At the heart of SCVP is the concept of delegating the task of discovering the missing links and building appropriate trust paths. This is articulated in the SCVP Draft RFC:[9]

> "The primary goals of SCVP are to make it easier to deploy PKI-enabled applications by delegating path discovery and/or validation processing to a server, and to allow central administration of validation policies within an organization."

As we have already noted the problem of path discovery is the fundamental issue. The required certificate authentication and validation checks made by the desktop application (i.e., by the client) are exactly the ones used for simple hierarchical PKIs.

The SCVP standard defines two approaches to delegating certificate path building. In the first approach, the client delegates the task of building a valid certification path to an SCVP server but not validation of the returned certification path. This approach is referred to as *delegated path discovery* (DPD). In the second approach, the client delegates both the task of building a valid certification path and the task of confirming that the public key contained in the end user certificate can be used for the intended purpose. This approach is referred to as *delegated path validation* (DPV). While both approaches unburden client applications of the difficult and time-consuming task of path building and provide for central administration of validation policies, the operational characteristics of the two approaches differ significantly[10]. Choosing between them is the subject of the next section.

---

[8] Actually the protocol is anything but "Simple" and the IETF is considering a name change to "Standard Certificate Validation Protocol". At the time of this writing the proposed SCVP standard is a draft but is expected to be adopted shortly.

[9] A copy of the draft RFC can be downloaded from http://tools.ietf.org/wg/pkix/draft-ietf-pkix-scvp/

[10] These "unburdened clients" are often referred to as "thin-clients".

Evaluating Deployment Choices

The crucial factor in deploying any PKI is ensuring that the deployed architecture can scale to meet the needs of the user population. As important as scalability is in simple hierarchical PKIs, [11] it becomes paramount to the successful deployment of federated PKIs where the number of trust domains can easily be in the hundreds and the number of certified identities in the hundreds of millions. It is readily apparent then that the number of PKE transactions will be orders of magnitude larger, in the hundreds of billions. Clearly the validation systems that build, authenticate and validate trust paths as part of validating each transaction must be automated in a scalable and secure fashion.

In choosing between delegated path discovery and delegated path validation architectures, it is important to compare the operational characteristics of each. The relevant criteria that must be met for certificate validation systems to be operationally successful are:

1. **High performance –** The system must provide fast responses to identity validation requests. Ideally the user should not be aware that validation is occurring.
2. **High availability** – The system must be available when the end user wants to use it.
3. **Scalable** – There should be no degradation in performance, availability or security as the system grows to meet the demands of an increasing number of users and trust domains.
4. **Secure** – The system must ensure public trust in the security of information exchanged in any public key enabled electronic transaction.
5. **Interoperable** – The system must be based on open standards to ensure interoperability with all applications conforming to the standards. The use of open standards will also ensure that the system can interoperate with other PKIs in a bridged environment.
6. **Low risk** – The system must be based on technology that has been proven to work in realistic operational scenarios of equal or larger size.

Delegated Path Validation

Figure 6 shows a typical DPV deployment architecture. The DPV SCVP server is a "trusted" server in that the client relies solely on the SCVP server's response as to whether or not the identity certificate in question is authentic, valid and can be used for the intended purpose. *Since the DPV SCVP server is commissioned as the ultimate authority for this determination of trust, much the same as a Root CA is the ultimate authority for the establishment of a trust path, it must be hosted and operated at the same security level as a Root CA.* Unlike a Root CA however, the DPV server must remain online at all times, making it vulnerable to intrusion.

[11] For example, the US Department of Defense has deployed a simple hierarchical PKI consists of 19 certification authorities which have issued more than 17 million certificates to 4.5 million users. The DoD found that scaling their validation system was one of their most difficult challenges. The solution was to deploy a distributed certificate validation system known as Distributed OCSP.

Scaling of a DPV system is accomplished by replicating the trusted server. This has several drawbacks including the cost of the hardware and of securely housing and operating multiple trusted servers. It also complicates the key management plan for ensuring that clients can securely "fail-over" from one "trusted" DPV server to another (e.g., the transfer of the client's trust from one DPV server to another is dependent upon whether these servers all have the same signing key or separate signing keys.)[12]

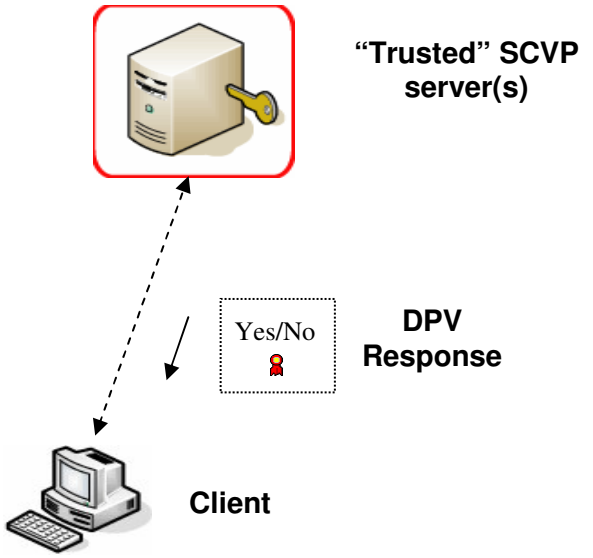

Figure 6: DPV System Architecture
In a DPV system a "trusted" SCVP server provides both path discovery and validation. The response returned to the client is a digitally signed Yes or No answer. In this system the client relies solely on the DPV server for validation which makes this server as security sensitive as a Root CA. It must therefore be secured and operated with the same level of security as a Root CA. Unlike a Root CA however, the DPV server must remain online with an "open port" to the outside world making it vulnerable to intrusion.

Delegated Path Discovery

Figure 7 depicts a DPD system as would be deployed using CoreStreet's Path Builder. In this architecture, Path Builder inserts a second tier of servers between the client and the trusted SCVP server. The trusted SCVP server is used to store all the security sensitive data and to execute all the security sensitive operations (e.g., signing of OCSP responses). While the trusted SCVP server must be housed and operated securely the "untrusted" SCVP servers can be deployed anywhere. Since the data delivered to these untrusted servers (i.e., the certificates that make up the trust path and their corresponding OCSP responses) are digitally signed they may be transmitted and stored in the clear.

[12] The DPV architecture is identical to that of Traditional (or first generation) OCSP. Consequently it has all the same security, performance, availability and scalability drawbacks.

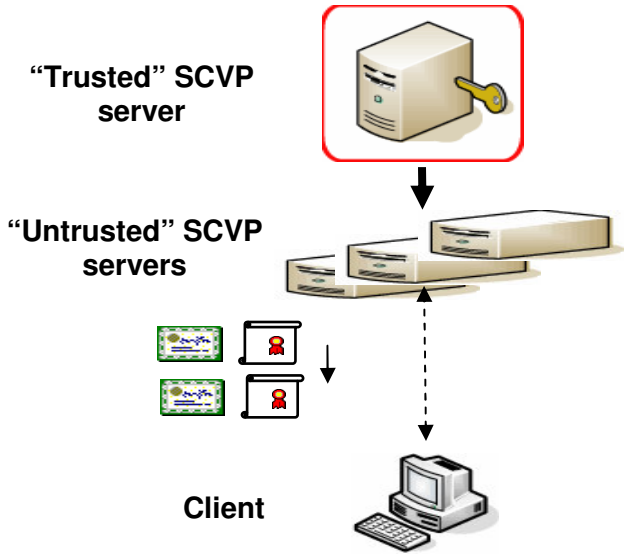

Figure 7: DPD System Architecture
In a DPD system a "trusted" SCVP server builds a trust path from the identity certificate in question to the client's trust anchor. The response returns the trust path and all associated OCSP certificate status responses. The client application then invokes the same authentication and validation checks as it would in a simple hierarchical PKI. While the "trusted" SCVP server must be housed and operated securely the "untrusted" SCVP servers can be deployed anywhere. There are no security requirements for data protection at this lower tier since all the data is digitally signed.

This approach greatly enhances the operational security and effectiveness of the system as a comparison with the previously defined success criteria clearly shows:

1. **High performance** – is achieved by providing the trust path and validation data on "untrusted" servers (called Path Builder Responders) located close to the end users. This eliminates the need to digitally sign responses in real time and decreases the distance, and therefore the number of network hops, between the client and a responder.[13]
2. **High availability** – is achieved by making a large number of Responders available to the end users. This ensures there is always a Responder available when needed. As these Responders are cheap to deploy (i.e., they require no special hardware, can be located anywhere and do not require "cleared" personnel to operate) they can be freely replicated and located anywhere.
3. **Scalable** – true scalability is achieved by separating the delivery process from the security sensitive operations associated with certificate validation. This fundamental design feature allows Responders to be freely added as the demand grows - ensuring high performance and availability without incurring large costs, sacrificing security or degrading the end user experience.
4. **Secure** – Path Builder provides the most secure validation solution available. Key features of the CoreStreet security architecture include:
   - **Intrusion threat eliminated**: Path Builder does not require nor allow any inbound communication to the "trusted" SCVP server (called the Path Builder Authority) from the outside world. Certificate validation requests go only to Responders, not to the Authority. The Authority can literally be "off-line". This eliminates all threats of an outside attack on the security sensitive parts

[13] Typical response times observed by the US DoD in their OCSP implementation of this architecture were less than a hundred milliseconds.

of the system.

- **Denial of service threat mitigated:** the two major characteristics of a denial-of-service vulnerability are: i) highly centralized service, and ii) providing the requested service requires a relatively long computation. This threat is mitigated by deploying multiple, geographically dispersed Responders. In addition, these Responders do not perform any long calculations since all validation and trust path data have been pre-signed and are ready for immediate release to the client.
- **Eliminates single point of failure:** this approach also eliminates the single point of failure threat present in non-distributed validation systems. In fact, if a physical attack or natural disaster were to incapacitate the trusted SCVP server, service would continue uninterrupted via the untrusted Responders for a configurable period of time, allowing for a "recovery" period during which a backup trusted SCVP server can be brought on-line.
- **Key compromise threat mitigated**: scaling the Path Builder system to serve increasingly larger user communities does not require distributing private keys or other security sensitive data or trusted operations to multiple locations. The CoreStreet Path Builder Authority requires a single key housed in a hardware security module, regardless of the number of users, trust domains or Responders being supported. This greatly enhances the ability to securely manage the operation by minimizing the key management process.
- **Built on proven CoreStreet technology:** the CoreStreet Path Builder is built upon the CoreStreet OCSP validation software which has successfully completed the internationally recognized Common Criteria (NIAP) evaluation at EAL 3 augmented for flaw remediation.

5. **Interoperable** – the CoreStreet Path Builder solution is based on open standards (OCSP and SCVP), ensuring interoperability with 3rd party client applications that conform to these standards. In addition, this approach to path discovery allows the addition of new CAs, Root CAs and even entire new Bridge Communities in a matter of minutes without any impact to relying party applications.

6. **Low risk** – CoreStreet's Path Builder provides a solution that is:
   - Proven to work – distributed architecture chosen by the US Department of Defense for its world-wide enterprise OCSP solution and by the US Intelligence Community, due to the inherent security features of this infrastructure, to support classified transactions.
   - Proven to scale – DoD OCSP deployment currently servicing over 17 million certificates with response times of a few hundred milliseconds
   - Based on open standards to ensure interoperability
   - Meets all US NIST PKITS path validation test suite requirements. This ensures product interoperability and conformance to standards.

Summary

Today the Web is available worldwide and heavily used. By simply typing a URL into a browser, any user with access to the Internet can connect to servers anywhere in the world. The underlying infrastructure that makes this possible is both simple to use and efficient. For worldwide electronic commerce to become a reality it must have a simple and efficient underlying infrastructure. The core of this infrastructure must provide a secure and automated way of validating identity credentials from a worldwide population that are certified by many different trust domains. The CoreStreet Path Builder System is not just a better way to provide certificate validation; it is the only solution that guarantees scalability without sacrificing performance, availability, security or cost.

***About** CoreStreet:*

Every day, the world's most demanding governments and commercial enterprises rely on CoreStreet technology to authorize critical events, ranging from opening signed e-mail and documents to granting physical access. More information, including technical whitepapers, industry solution studies and a list of patents awarded to the company is available at www.corestreet.com .

Contact us at:

CoreStreet Ltd.
One Alewife Center, Suite 200
Cambridge, MA 02140
U.S.A.

Email: **info@corestreet.com**
Telephone: +1-617-661-3554

Bibliography

Andrew Nash, William Duane, Celia Joseph and Derek Brink, *PKI Implementing and Managing E-Security*, Osborne/McGraw-Hill, Berkeley, California, 2001

Appendix A: Trust Lists

Perhaps the simplest trust model for facilitating trust among different trust domains is to use a trust list. A trust list is a collection of trust anchors. In this model the user registers each of the trust anchors for all the PKIs that it wants to explicitly trust. While this approach seems attractive at first glance there are several operational drawbacks to the use of trust lists. These include:

- Who negotiates the trust relationship and at what level is this done? Clearly this must be done at a level that has the authority to represent all the users within that trust domain. And it must be done for each relationship.
- There are scalability issues associated with pair-wise relationship building among trust domains. The number of relationships to be negotiated increases rapidly as the number of trust domains to be included grows.
- The trust list approach does not provide for central administration and enforcement of validation policies within an organization, thus missing one of the primary benefits of Bridge CA approach. Local enforcement of validation policies precludes quick and efficient implementation and management of policies.
- The trust list approach requires a mechanism local to the desktop that prevents users from making changes to their trust list.
- The trust list approach raises all the issues of scalability associated with desktop maintenance. The cost of touching each desktop to update a policy or trust list is high and will therefore require an automated approach for updating the trust lists.
- Maintenance of trust lists will be an ongoing effort. It includes adding new entities as new trust relationships are needed and developed and deleting other entities when a trust relationship is dissolved.

Based on the above list of issues one can reasonably conclude that the lack of scalability, the loss of central administration and enforcement of policy and the additional operational costs associated with the trust list approach make it a poor choice for facilitating trust across multiple trust domains.

Appendix B: Definitions

Authentication	Process of proving your claimed identity.
Authorization	Official permission or approval. Process of granting access rights to an identity.
Bridge Certification Authority	A Certificate Authority that only issues cross-certificates to member organization root CAs.
Client	The relying party application involved with the public key enabled transaction.
Credential	Evidence of the truth of one's claimed identity and/or privileges.
Cross-certificate	A certificate issued by a CA in one trust domain to a CA in another trust domain.
Cross-certification	The certification by a CA from one trust domain of the public key of another CA in a different trust domain. This facilitates building trust paths across trust domains.
Identity	The qualities of an individual that make them different.
NameConstraints	A certificate extension (optional data field) that provides a mechanism for limiting the scope of trust relationships that are established by cross-certification.
PolicyConstraints	A certificate extension (optional data field) used to constrain the use of a certificate based on the policies under which it was issued
Relying party	The entity that is passed a certificate and wishes to use it to prove an identity.
Thin client	Client software used for trust path validation that does not include the path discovery capability.
Trust anchor	An entity which is sufficiently trusted to certify the identity other entities. For example a root certification authority.
Trust domain	A community of individuals all of which abide by a common set of policies
Trust list	A collection of trust anchors. The most common example is the collection of root certificates that are explicitly trusted by the user and stored locally in the user's Web browser or other client application.
Trust model	A description of how trust relationships are established and what the rules are for finding and validating trust paths.
Trust path	A path represented by a chain of certificates that allows a user validating an identity to trace the trust relationships back to his/her trust anchor.
Validation policy	Specifies the rules and parameters to be used when validating a certificate.

Appendix C: Acronyms

CA	Certification Authority
CAPI	Crypto Application Program Interface
CRL	Certificate Revocation List
DoD	Department of Defense (US)
DPD	Delegated Path Discovery
DPV	Delegated Path Validation
EAL	Evaluation Assurance Level
FBCA	Federal Bridge Certification Authority (US)
FIM	Federated Identity Management
IETF	Internet Engineering Task Force
NIAP	National Information Assurance Partnership (US)
NIST	National Institute of Standards and Technology (US)
OCSP	Online Certificate Status Protocol
PKE	Public Key Enabled
PKI	Public Key Infrastructure
PKITS	Public Key Infrastructure Test Suite
RFC	Request For Comment
SAFE	Secure Access For Everyone
SCVP	Simple Certificate Validation Protocol

www.corestreet.com
+1 617 661 3554
info@corestreet.com

Additional whitepapers and datasheets available at:
www.corestreet.com/library