XML Key Management Specification (XKMS)

Phillip M, Hallam-Baker
VeriSign Inc.
401 Edgewater Place Suite 280
Wakefield MA 01880
781 245 6996 x227
pbaker@verisign.com

Warwick Ford
VeriSign Inc.
401 Edgewater Place Suite 280
Wakefield MA 01880
781 245 6996 x225
wford@verisign.com

ABSTRACT

The XML Key Management Specification (XKMS) is described. XKMS is a Web Service that provides an interface between an XML application and a Public Key Infrastructure (PKI). XKMS greatly simplifies the deployment of enterprise strength Public Key Infrastructure by transferring complex processing tasks from the client application to a Trust Service.

Keywords

Security, Cryptography, Public Key Infrastructure, XML, Trust Service.

1. INTRODUCTION

Public Key cryptography permits secure communication to be established between any parties provided only that each has trustworthy knowledge of the public key of the other. Public Key Infrastructure (PKI) based on digital certificates provides a means of exchanging trustworthy public key information. Configuration of a PKI is an inherently complex task since to serve its purpose the PKI must reflect the complexity and subtly of real world trust relationships.

1.1 THE PKI DEPLOYMENT PROBLEM

For a PKI feature to be useful every client must first support it. This has posed a severe limitation on the sophistication of PKI deployment. Full support for the industry standard X.509/PKIX specification requires a very large and complex client implementation that very few applications support directly (figure 1).

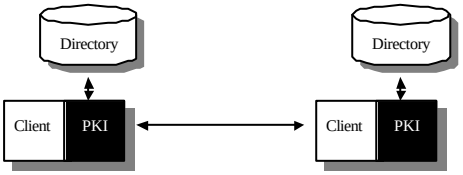

Figure 1 Client Complexity in Traditional PKI is High.

PKI 'plug-ins' provided by third party PKI vendors are expensive to deploy and maintain, particularly since each PKI client must be configured with the location of the local PKI repository. A new plug in deployment is required each time there is a change to the PKI configuration, support for new PKI features is required or the base application is upgraded.

1.2 Trust Service Solution

A Trust Service solves the client deployment problem by shielding the client from the complexity of the underling PKI (figure 2). This ensures that all clients in the enterprise support the full range of PKI features and removes the need for the client to support each new PKI feature.

XML Key Management Services (XKMS) [1] is a PKI trust service that provides an XML interface to an underlying PKI. This achieves the following benefits:

- Very small client footprint.
- XML syntax greatly simplifies implementation.
- Trust relationships between enterprises may be configured at the enterprise level.
- Deployment of new PKI features does not require deployment of new clients.

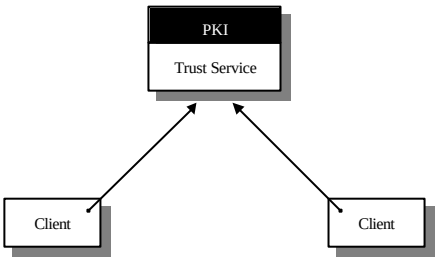

Figure 2: XKMS shields clients from PKI complexity

2. KEY INFORMATION

XML Key Information Service Specification (X-KISS) defines a protocol for a trust service that resolves public key information contained in XML Signature [2] `<KeyInfo>` elements.

The XML Signature `<KeyInfo>` element identifies a public key. Comprehensive support is provided for all types of PKI data in current use including:

- The Public Key Parameters
- Any naming scheme (e.g. URI, Common Name)
- X509 Certificate, CRL, OCSP token
- SPKI, PGP key signing.
- A URL for the retrieval of any of the above

X-KISS supports two service tiers:

Tier 1: Locate

The client sends one `<KeyInfo>` element to the service and requests that the trust service provide a `<KeyInfo>` element that identifies the same key but is in a different format (e.g. X.509 certificate converted to key parameters).

Tier 2: Validate

The trust service validates the trustworthiness of the information returned according to service specific criteria.

2.1 Location Example

A client receives a signed XML document. The <KeyInfo> element in the signature specifies a retrieval method for an X.509 certificate. The client lacking the means to either resolve the URL or parse the X.509 certificate to obtain the public key parameters delegates these tasks to the trust service (figure 3).

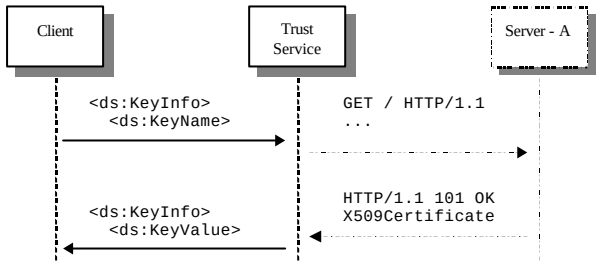

Figure 3 Location Service Provides Name Resolution

2.2 Validation Example

The Validate service allows a client to delegate all trust processing functions to a trust service. As with the Locate service the client creates a query that specifies the information the validation service is to locate. Unlike the location service however the validation service is responsible for ensuring the trustworthiness of the data returned before relying upon it.

A client receives a signed XML document and queries the trust service to determine whether the signing key is trustworthy. In this case an X.509 certificate authenticates the signing key. The Trust Service builds a certificate trust path, then validates each certificate in the path against the relevant Certification Revocation List. The client is shielded from this complexity however and the trust service returns only the information of specific interest to the client; the key parameters, the data bound to the key and the validity of the binding (figure 4).

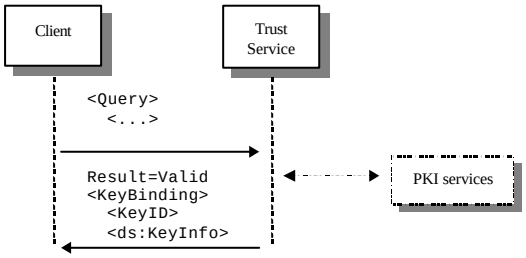

Figure 4 Key Validation Service

Delegation of trust processing functions to a trust service makes enterprise-wide control and oversight of PKI configuration possible. This is essential in Business-to-Business applications where the important trust relationships are between enterprises and not between individuals or the applications they use.

3. KEY REGISTRATION

XML Key Registration Service Specification (X-KRSS) defines a protocol for a trust service that accepts registration of public key information. Once registered, the public key may be used in conjunction with other web services including X-KISS.

X-KRSS is designed to support all of the functions associated with the public key lifecycle:

Registration. The registration function supports registration of an association of a public key and additional data (such as a name) to create a 'key binding'. Private keys may be generated either locally by the client (desirable for signing keys) or by a central key generation service (desirable in cases where key recovery is supported). Requests may be authenticated with either a limited use shared secret or a digital signature.

Renewal. XKMS allows a PKI to be operated without digital certificates ever being issued, eliminating the need for certificate renewal. In cases where certificates are issued by the underlying PKI renewal processing may be performed automatically without the need for client interaction.

Revocation. An authorized party may request that the trust service revoke a key binding. This may be necessary because the key has been compromised or because information contained in the key binding is incorrect.

Recovery. Private key recovery is essential when an end user has lost their private key and requires access to their encrypted data. The X-KRSS recovery function provides an authenticated means of re-issuing a private key to a user.

Roaming: In many cases an end user needs access to their private key from many machines. Smartcards provide one answer to this problem but require hardware deployment. X-KRSS provides the necessary support for software based key roaming protocols.

X-KRSS may be configured hierarchically in the manner of a Local Registration Authority. This allows a registration request to be authenticated by a local trust service then passed on to another trust service where actual processing is performed.

4. OPERATION WITH X.509 PKI

XKMS provides an interface to and is not a substitute for a PKI. It is expected that most initial deployments will interface to an underlying X.509 PKI allowing XKMS clients to interoperate seamlessly with applications already deployed.

XKMS removes the need for client support of PKI features. XKMS thus permits aggressive use of sophisticated X.509 features such as Certificate Revocation Lists, cross-certification and policy constraints that have hitherto been impractical due to limited client support.

5. ACKNOWLEDGMENTS

The authors would like to acknowledge the work of the many people who contributed to the design of XKMS, in particular, Barbara Fox, David Solo, Blair Dillaway, Brian LaMacchia, Jeremy Epstein and Joe Lapp.

6. REFERENCES

[1] VeriSign, Microsoft, webMethods. *XML Key Management Specification*. Jan. 2001, for latest version see http://xmltrustcenter.org/

[2] D. Eastlake, J. R., D. Solo, M. Bartel, J. Boyer , B. Fox , E. Simon. *XML-Signature Syntax and Processing*, W3C, http://www.w3.org/TR/xmldsig-core/