What Web Services Needs to Know About PKI

Rich Salz

Datapower Technology, Inc.

rsalz@datapower.com

Outline

- Basic cryptography.
- Public-key infrastructure.
- XML Security Standards.

Basic Cryptography

- Cryptography: methods for turning *plaintext* into *ciphertext*.
- First principle: assume adversary knows the algorithm; Protection is in the key:
  - Kerkhoff (Kirchoff) (Kerhkoff), 1860.
  - Folks still get it wrong!

About Keys

- A *key* is the additional input into the algorithm that “makes the magic happen.”
- Keys can sometimes be different lengths (RSA) or not (DES, 3DES).
- Bigger keys are usually better.
- Keys can change over time.

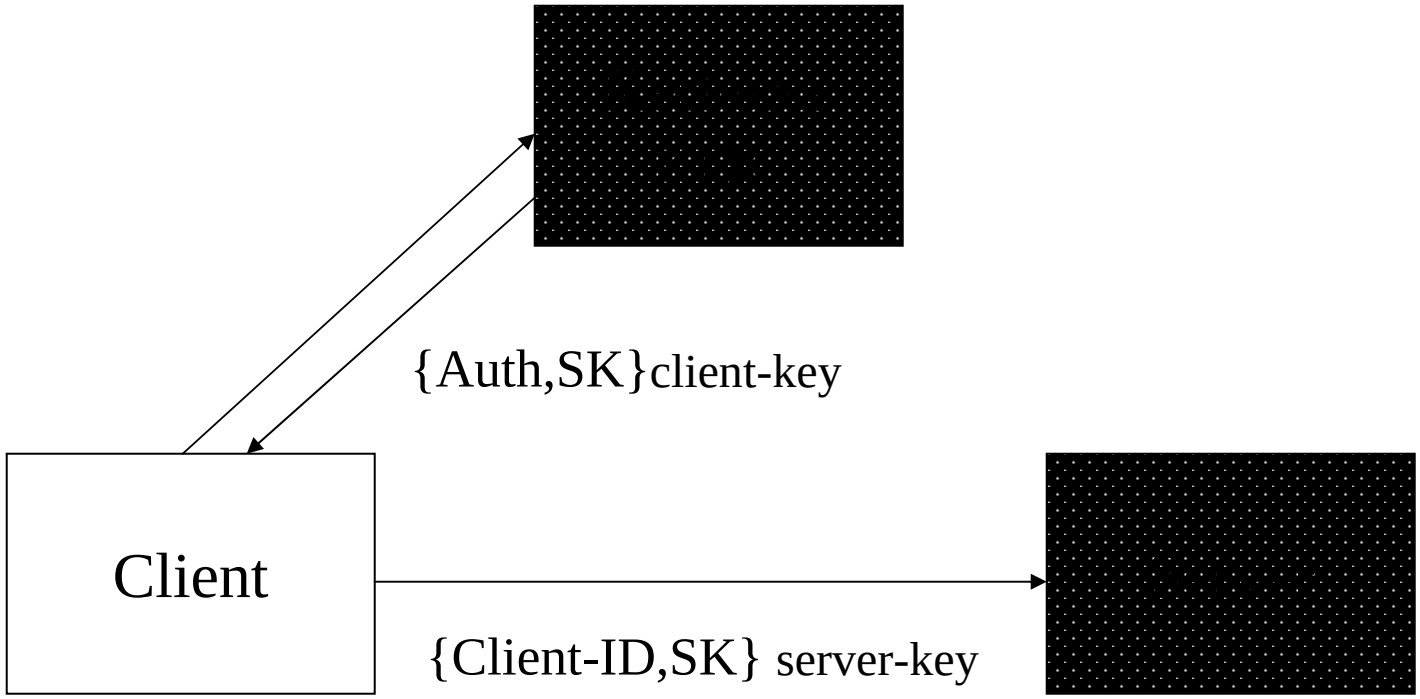
“There are two kinds of…”

- There are two kinds of encryption:
  - We share a secret…
  - … Or we don’t.
- There are, of course, other ways to look at it.

Shared Secret: DES

- If we share a key, then the algorithm is probably *symmetric*: same key encrypts and decrypts.
- Best-known symmetric algorithms: DES, 3DES, and the new AES.
- First and biggest use of DES: ATM and banking.

Kerberos



Non-shared keys

- If we don’t share a key, the algorithm is *asymmetric*.
- Two keys, logical inverses of each other.
- If you have one key, it is “impossible” to determine the other one.
- Most common example: RSA.

Key (sic) Differences

- Symmetric:
  - Faster.
  - Requires pre-registration; Key distribution is hard.
  - Trust decisions are simple.
- Asymmetric:
  - Slower (keys are usually bigger).
  - Keys are used more often.
  - Trust decisions are harder.

Hashing

- Turn arbitrary stream of bytes into a fixed-size identifier.
- Believed impossible to force “collisions.”
- Poor history: MD2, MD4, MD5 have fallen; SHA-1 not studied by the best.
- AES brings SHA-256 and SHA-512.

Digital Signature

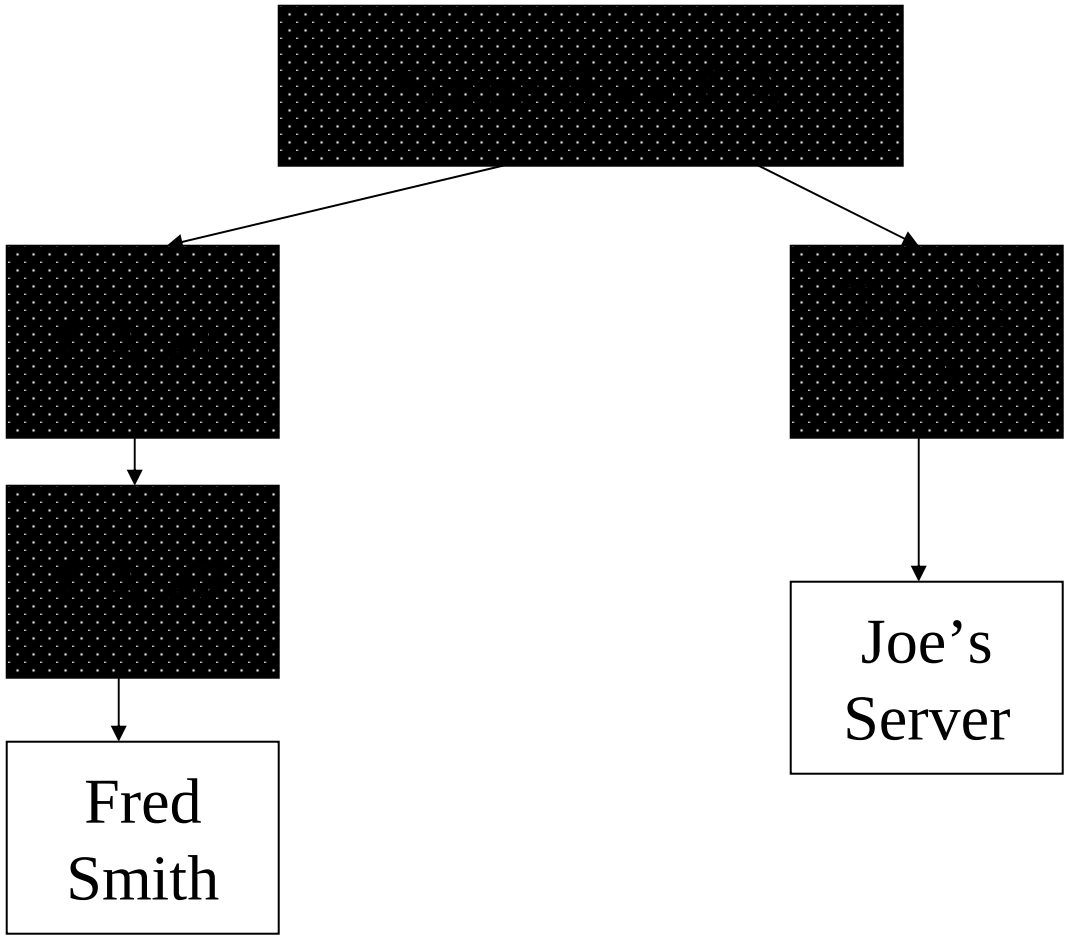
- Conventional public-key signature:
  - Hash document.
  - Encrypt with private key, giving the *signature*.
  - Recipient gets document, calculates hash.
  - Decrypts signature, compares hash values.
- Other option, HMAC (hashed message authentication code):
  - Hash concat(shared secret, document).

X.509 Certificates

- Created by an MIT undergraduate
- Addresses scaling by having Certification Authority sign certificates:

  $\{name, key, other\text{-}info\}_{CA\text{-}key}$
- Information items include: Subject Name, Issuer Name, Serial Number, Subject Key, etc…
- Build a tree of CA's.

PKI Picture



PKI Evaluation, I

- I don’t know Joe’s Server.
- It was signed by Tom’s CA.
- I don’t know Tom’s CA.
- It was signed by Some Root CA.
- Some Root CA signed their own certificate.

At this point, the identity verifies.

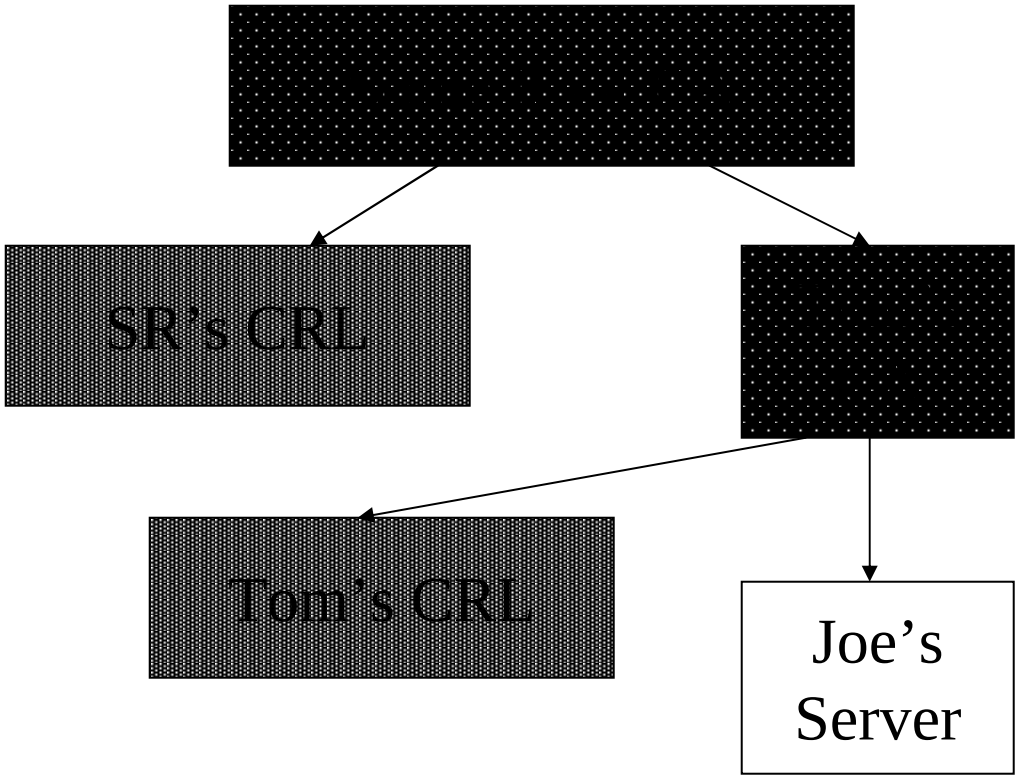
PKI Evaluation, II

- I know Some Root CA
- I have a *certificate chain* from me to Joe's Server.
- All is good.

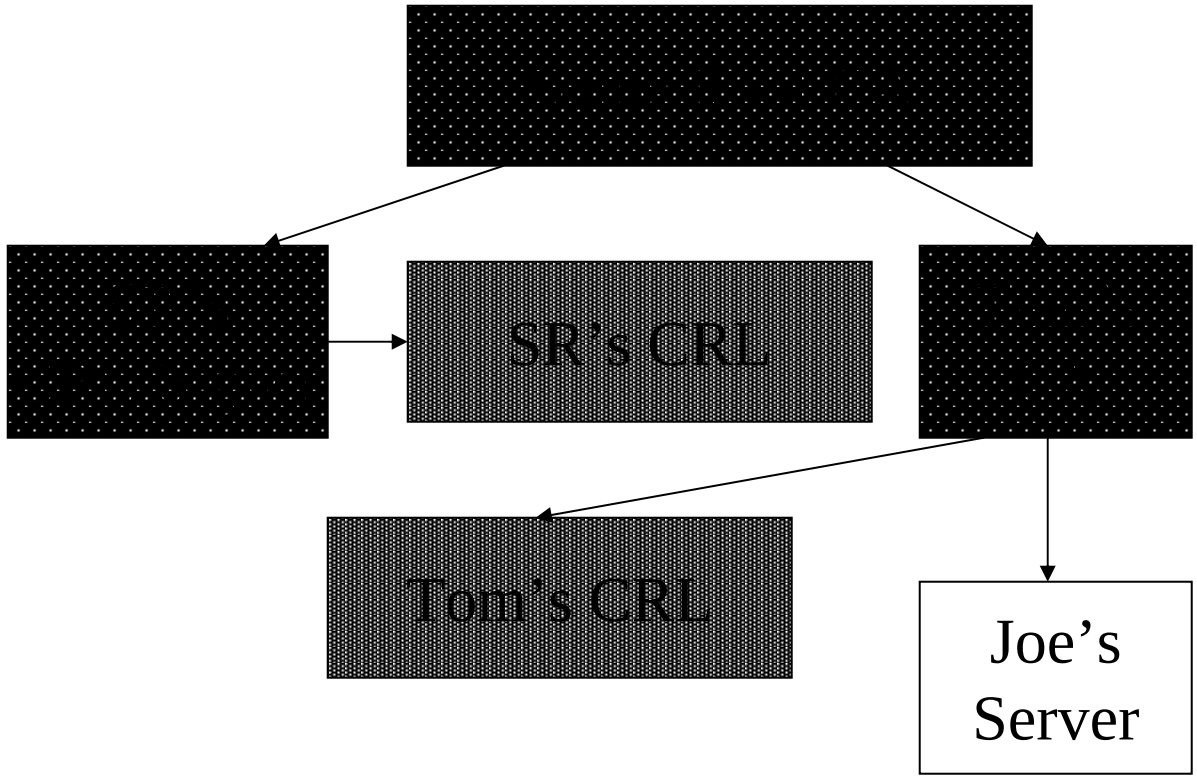
PKI Evaluation, III

- … But is it?
- How do I know “some root CA”?
- Suppose identity is no longer valid: stolen keys, bankruptcy, etc.
- Certificate revocation list (CRL) – a signed list of revoked serial numbers.
- Signed by CA, or its designee.

Revocation is Difficult



Revocation can be $O(n^3)$



PKI Common Practice

- Sender's "push" their certificate and their CA's.
- Receivers verify signature; return error if CA isn't known.
- Revocation basically ignored.

- Don't believe me? IE "CRL Patch"

PKI is supposed to be Global

- Key distribution and CA scaling are its major assets.
- Many have tried:
  - X.500
  - Verisign
  - Identrus
- National efforts moving forward.

Certificate Types

- A CA can limit what its subjects are allowed to do:
  - keyUsage: cert-sign, dig-sig, key exchange, etc.
  - extendedKeyUsage: arbitrary action (OID).
  - nameConstraints: depth and portions of name tree.
  - policyIdentifiers: List of applicable OID's.

Trust Evaluation: Theory

If you can find a trust anchor,
And the math verifies,
And the certificate is still valid,
And it hasn’t been revoked,
And it’s being used for the right purpose,
And it’s issued under the right policies,
And that’s true as you walk up the chain,
Then proceed.

Trust Evaluation: Practice

- Check the signature.
- Let the relevant FI assume the risk.

XML Security Standards

- W3C Has:
  - XML Digital Signature
  - XML Encryption
  - XML Key Management
  - Open Digital Rights Language (a Note)
- OASIS has:
  - Security Authoriation Markup Language
  - WS-Security
  - Extensible Acess Control Markup Language
- Others have:
  - Passport (Microsoft)
  - Liberty Alliance (Sun, etc.)

XML Digital Signature

- A thing of beauty.
- A signature consists of *signature info*, *key info*, and *source info*.
- Signature info includes:
  - Algorithm
  - Object references
  - Signature Value

XML DSig, II

- Key info has key or locator:
  - URL (HTTP, LDAP, etc)
  - Raw key
  - Certificate
  - CRL's
- Sources are *referrants* and are transformed:

```
<ds:Reference URI="#msgbody">
  <ds:Transforms>…</ds:Transforms>
  <ds:DigestMethod algorithm="….”/>
  <ds:DigestValue>LyL…</ds:DigestValue>
```

XML DSig, III

- Extensible in the right places (e.g., *KeyInfo*).
- Transformation is a powerful mechanism.
- Major wart: too many types of canonicalization.
- What it gives you: what you got is what I sent.

XML Encryption

- Similar parts:
  - Algorithm identifiers
  - Key locators
  - Session key values
  - Encrypted data

WS-Security

- Very simple: how to use XML DSig and XML-Encryption for SOAP. Answer: define some headers.
- Also add security schemes and identifiers.
  - Example: DSig *KeyInfo* can contain a link to a Kerberos ticket.
- Will it interoperate?

XKMS

- Replace years of binary protocols with grand XML-based simplification:
  - Find a key for me.
  - Can I trust this key?
  - Create a certificate for me (or revoke it).
- Trust example: XKMS Locate element takes an XML DSig *KeyInfo*.

XKMS Promise and Issues

- Syntactically and semantically clean.
- PKI Vendors see it as their “last chance.”
- Beware outsourcing authorization.
- Need enterprise products to enforce organizational policy.
- Need to *federate* servers together.

Who are you?

- SAML: a (trusted-) third party makes identity assertions to server about client.
- Liberty: a login server makes those assertions.
- Passport: butchered Kerberos protocol.
- Arguably the most political.
- Heterogeneous single-sign on? Not yet…

Can you do that?

- Authorizaton: process of deciding if client can perform requested action.
- Recall NTFS file ACL's, DCE ACL's, Posix ACL's.
- XACML isn't any better.
- ODRL, etc., are ethically odious.
- Stick with first part: authentication.