

Public key infrastructure (PKI)

Raphael Finkel

Computer Science Department

University of Kentucky

Fall 2007

Overview of PKI

- ▶ PKI is a cryptographic technique for authenticating subjects.
- ▶ **Subjects** are people or companies, but are usually represented by web sites (URLs).
- ▶ **Authenticating a subject** means associating a public key with the subject.
- ▶ A (symmetric) **public key** is half of a key pair K_0, K_1 such that a message encrypted by K_i can only be decrypted by K_{1-i} , and neither of the K_i can be derived from the other. The other half is the **private key**.
- ▶ A **certificate authority** (CA) records the association of a subject to its public key in a digitally signed document called a **certificate**.
- ▶ A **digital signature** on a document d signed by an authority A is typically $encrypt(privateKey_A, hash(d))$.
- ▶ A subject \mathbf{S} that has a certificate signed by a CA can present that a certificate to \mathbf{C} so that \mathbf{C} can confidently extract \mathbf{S} 's authenticated public key.

So what?

- ▶ If **C** has an authenticated public key for **S**, then **C** can establish a secure communication channel to **S** without risk that it has established the channel with an impostor.
- ▶ **C** can verify the digital signature on any message signed by **S**.
- ▶ These properties allow **C** and **S** to establish **privacy**, **message integrity**, and **authentication** without prior contact and without exchanging any secret information.

What makes it hard

- ▶ Presenting *bona fides* to the CA to become authenticated
- ▶ Trusting the CA
- ▶ Satisfying evidentiary laws in different jurisdictions
- ▶ Revoking certificates
- ▶ Assumption that knowing a subject's name is the same as knowing its identity (all its defining characteristics), and that knowing a subject's identity lets you know the subject's authorizations. (SPKI does not make this assumption.)

Certificate sequences

- ▶ Notation: Ignoring other fields, a certificate $F = \{subject, key, issuer, signature\}$.
- ▶ \mathbf{C} might hold F_1 that purports to authenticate subject \mathbf{S} .
- ▶ But \mathbf{C} might not know $F_1.issuer$'s public key, so it can't verify the signature on F_1 .
- ▶ \mathbf{C} might solicit F_2 that authenticates $F_1.issuer$. Then $F_2.key$ should verify $F_1.signature$.
- ▶ This sequence might continue until it reaches an issuer whose public key \mathbf{C} knows.
- ▶ But \mathbf{C} might still not trust F_1 , because knowing the identity of $F_1.issuer$ is not the same as trusting $F_1.issuer$ to be a high-quality issuer.
- ▶ The typical chain has only F_1 and F_2 , where $F_1.issuer = F_2.subject = F_2.issuer$, and $F_2.issuer$ is one of the 20 or so **trusted CAs** whose public key is hard-coded into the Firefox or email client and whose secret key is very heavily guarded.

A certificate standard: X.509

- ▶ X.509 typically means RFC 3280: “Certificate and Certificate Revocation List (CRL) Profile”
- ▶ Contents (sample values in red, lengths in green).
 - Version (3), serial number (9B), algorithm (SHA-1, RSA encryption)
 - Validity date range (earliest, latest)
 - Subject (CN=192.168.1.253; OU=Tobey Village Office; O=Second Avenue Software; L=Rochester; ST=New York; C=US) (plus its unique identifier, optionally)
 - Subject’s public key (~140B) and the associated algorithm (RSA encryption)
 - Issuer, its signature (~128B), and the signature algorithm (SHA-1 with RSA encryption)
- ▶ A single parcel may contain a sequence of certificates
- ▶ A certificate may be good for various usages, such as SSL client, SSL server, S/MIME signing, CRL signing. e-mail.

Firefox warning on expired certificate

"192.168.1.253" is a site that uses a security certificate to encrypt data during transmission, but its certificate expired on 06/30/2006 02:44 PM.

You should check to make sure that your computer's time (currently set to Wed 28 Nov 2007 12:08:15 PM EST) is correct.

Would you like to continue anyway?

Firefox warning on bad subject

You have attempted to establish a connection with "tobey.secondavesoftware.com". However, the security certificate presented belongs to "192.168.1.253". It is possible, though unlikely, that someone may be trying to intercept your communication with this web site.

If you suspect the certificate shown does not belong to "tobey.secondavesoftware.com", please cancel the connection and notify the site administrator.

[View Certificate](#)

[Cancel](#)

[OK](#)

Firefox view of UKY certificate

General | Details

This certificate has been verified for the following uses:

- SSL Client Certificate
- SSL Server Certificate

Issued To

Common Name (CN)	*.uky.edu
Organization (O)	University of Kentucky
Organizational Unit (OU)	Network Operations Center
Serial Number	16:39:14:B5:B2:C7:9B:D1:AF:8F:BA:65:2B:7F:6F:7C

Issued By

Common Name (CN)	VeriSign Class 3 Secure Server CA
Organization (O)	VeriSign, Inc.
Organizational Unit (OU)	VeriSign Trust Network

Validity

Issued On	05/30/2007
Expires On	06/09/2008

Fingerprints

SHA1 Fingerprint	48:67:53:51:76:94:E0:D5:80:66:AD:CC:B8:01:62:42:CF:36:88:96
MD5 Fingerprint	C2:F9:DD:C1:5D:39:77:18:D6:BF:5F:6E:57:23:86:5E

Close

Certificate authorities

- ▶ Each country tends to have its own certificate authorities, because whether a digital signature is legally binding is linked to how local jurisdictions regulate the certificate authorities.
- ▶ Multinational SSL certificates are mostly issued (for a price) by VeriSign (includes Thawte and Geotrust), Comodo, GoDaddy, DigiCert, Network Solutions, and Entrust.
- ▶ Providers of free (but low trust) certificates: CA Cert.org, Comodo, Thawte, StartCom.
- ▶ You can create your own self-signed certificate
 - `openssl genrsa -out PRIVATE_KEY 1024`
 - `openssl req -new -days VALID_DAYS -key PRIVATE_KEY -x509 -out CERTIFICATE_FILE`
- ▶ You can now submit your certificate to a certification authority
- ▶ You can view your certificate
 - `openssl x509 -in CERTIFICATE_FILE -noout -text`

Revoking certificates

- ▶ When to revoke
 - A subject's private key has been revealed.
 - The CA issued a certificate improperly (an impostor once was issued a certificate for Microsoft).
 - The subject violated the CA's policy (like publishing false documents).
- ▶ Only the issuer of a certificate may revoke it, either irreversibly or reversibly.
 - Add the certificate's serial number to a **certificate revocation list (CRL)**, which has a given, usually short, lifetime, and is digitally signed by the CA.
- ▶ Holders of certificates should check them against the most recent CRL before trusting them.
- ▶ If an issuer in a sequence of CAs has its certificate revoked, all downstream certificates are suspect.
- ▶ An alternative is the **online certificate status protocol (OSCP)**, with the issuer running an online OSCP responder. Clients need not download, store, and parse very large CRLs; the cost of validation rests with the issuer.

Firefox setting for OSCP

Firefox can use Online Certificate Status Protocol (OCSP) to verify certificates. Set Firefox to use OCSP as follows:

- Do not use OCSP for certificate validation
- Use OCSP to validate only certificates that specify an OCSP service URL
- Use OCSP to validate all certificates using this URL and signer:

Response Signer: ▾

Service URL:

Cancel

OK

Uses of PKI

- ▶ Bootstrapping SSL by providing the initial public key of the server end.
- ▶ Encryption and sender authentication of email.
- ▶ Encryption and sender authentication of documents.
- ▶ Authentication of users to applications (via smart cards, for instance, or by client authentication with SSL).

Alternative to PKI – Webs of trust: PGP, GnuPG

- ▶ Subjects create their own certificates.
- ▶ Other subjects attest to those certificates (with a digital signature).
- ▶ Subjects may designate **fully** or **partially trusted introducers**.
- ▶ A subject trusts a key that is attested by at least n of its partially trusted or m of its fully trusted introducers; each subject may choose n and m .
- ▶ There are commands available to encrypt, decrypt, sign files, as well as generate key pairs, maintain a keyring, manipulate trust parameters, and sign someone's public key.
- ▶ + Resilient to company failure, unlike PKI.
- ▶ – New subjects have difficulty getting enough endorsements, hence key-signing parties and websites devoted to finding potential endorsers.

Alternative — Simple Public Key Infrastructure

- ▶ Abbreviated SPKI, pronounced [spuki], described in RFC 2692 and 2693; a merger of the original SPKI and SDSI (Rivest and Lampson).
- ▶ A **name certificate** binds a **name** to a public key and issuer (syntax is likely wrong here).

```
(cert
  (issuer (hash md5 |PWKULKycrQ/Pxu9qWBSY2Q==|))
  (subject (hash md5 |Z4a6hysK/0qN0L5SFkcJFQ==|)
    "Alfred E. Newman"))
```

- The `subject` term connects a public key with a name local to the issuer.
- ▶ The issuer can sign the certificate.
- ▶ Someone else who doesn't know about this subject directly can refer to it indirectly.

```
(name (hash md5 |PWKULKycrQ/Pxu9qWBSY2Q==|)
  "Alfred E. Newman")
```

SPKI, continued

- ▶ One can extend name certificates to establish groups and indicate the subjects that belong to the groups.

```
(cert
  (issuer (hash md5 |PWKULKycrQ/Pxu9qWBSY2Q==|))
  (group "Gang of idiots")
  (subject name "Alfred E. Newman")
  (subject name "Don Martin"))
```

- ▶ A **delegation certificate** grants **authorizations**.

```
(cert
  (issuer (hash md5 |PWKULKycrQ/Pxu9qWBSY2Q==|))
  (subject (name "Gang of idiots")))
(tag (write files at http://www.mad-magazine.com)))
```

SPKI, continued

- ▶ The authorization language allows for intersection of authorizations.
- ▶ A **threshold subject** grants authorization or delegation only when k out of n subjects concur.
- ▶ There is no commercial CA; SPKI is to be deployed in closed environments.

Bibliography

- ▶ Wikipedia articles on PKI, Certificate Authority, X.509, Certificate revocation list, Online certificate status protocol, Simple public key infrastructure (SPKI)
- ▶ A list of certificate authorities:
http://www.dmoz.org/Computers/Security/Public_Key_Infrastructure/PKIX/Tools_and_Services/Third_Party_Certificate_Authorities/
- ▶ Open SSL command-line HOWTO: <http://www.madboa.com/geek/openssl/>
- ▶ SDSI 2.0 documentation at
http://groups.csail.mit.edu/cis/sdsi/sdsi2/sdsi20_toc.html