Grid Computing

Security – Public Key Infrastructure, X.509 and Certificate Authorities (CA)

Paul A. Farrell
Fall 2006

The Grid: Core Technologies
Maozhen Li, Mark Baker
John Wiley & Sons; 2005, ISBN 0-470-09417-6

Material from Dr John Watt, Grid Security Engineer
National e-Science Centre, University of Glasgow
Lecture slides by Amy Apon, Lawrie Brown,
University of Aksansas

International Summer School on Grid Computing 2006
Ischia (Naples) July 9-21, 2006

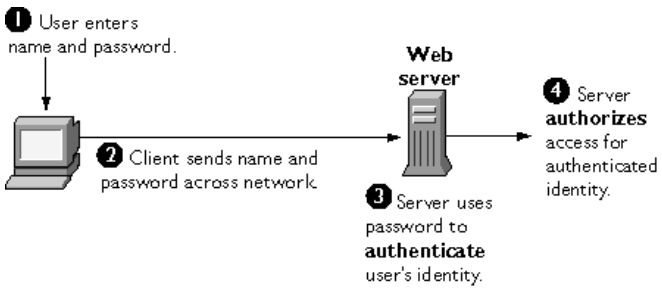
Public Key Infrastructures (PKIs)

- PKIs provide a mechanism for privacy, integrity and authentication using public keys
  - Implemented with DIGITAL CERTIFICATES
    - Your UNIQUE virtualised identity
  - Issued by a CERTIFICATE AUTHORITY
    - Entity which administers certificates and issues them correctly
  - X509 (1988) is the standard for PKI certificates
    - Binds a globally unique X500 distinguished name to a public key
      - In reality, CAs tend to choose any name they want
    - Web browser compatible

Why certificates?

- Certificates required for the large-scale use of public-key cryptography
- Anybody can create a public-private key pair
  - a malicious user can fool the originator into using their public key, and so get access to the information
- Solution: originator only trusts public keys that have been signed ("certified") by an authority
- Problem: user may not be familiar with another's certificate authority
  - certificate may also include a CA's public key signed by a "higher level" CA, which is more widely recognized
  - Gives rise to hierarchy of CAs

Recall: Using a password to authenticate a client to a server



Certificate-based Authentication

This is TLS (SSL) again. Recall:

Step 2: The server sends the client the SSL version number, random number Y, and its public key (packaged into a certificate)

Step 3: The client verifies that the server is who is says it is by examining the certificate. (Remember we said we would say more?)

Certificate-based authentication details

Assume that the client has a private key and a certificate that contains the associated public key.

- The client generates random data
- It creates a digital signature of the data using the private key
- Client sends the data, digital signature, and its certificate across the network
- The server retrieves the packet containing the data, digital signature, and certificate
- Server extracts the client's public key from the certificate
- Server decrypts the digital signature using the client's public key
- Server compares the data with the decrypted signature to authorize the client

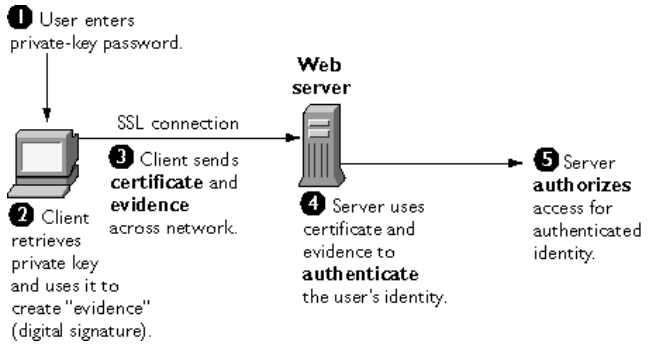
Same process is used with user authentication

In the case of user authentication,

- The user enters a password that unlocks a local database and gives access to a private key.
- The client software retrieves the private key along with the associated public key certificate from the local database
- Continue with remaining client steps to authenticate a user to a server

No user password is sent across the network!

Using a certificate to authenticate a client to a server



Five types of certificates (used by Netscape)

1. Client SSL certificates
   - Used to identify clients to servers via SSL (client authentication).
   - Typically, the identity of the client is assumed to be the same as the identity of a human being, such as an employee in an enterprise.
2. Server SSL certificates
   - Used to identify servers to clients via SSL (server authentication).
   - Server authentication may be used with or without client authentication.
   - Server authentication is a requirement for an encrypted SSL session.

Five types of certificates

3. S/MIME certificates
   - Used for signed and encrypted email.
   - As with client SSL certificates, the identity of the client is typically assumed to be the same as the identity of a human being, such as an employee in an enterprise.
   - A single certificate may be used as both an S/MIME certificate and an SSL certificate.
4. Object-signing certificates
   - Used to identify signers of Java code, JavaScript scripts, or other signed files.
5. Certificate Authority (CA) Certificates
   - Used to identify CAs.
   - Client and server software use CA certificates to determine what other certificates can be trusted.

X.509 PKI

- Hierarchy of CAs top-down tree
- Top level (root) CA should be universally known
- Revocation of certificates
  - If private key has been exposed
  - Need to check Certificate Revocation List to check
- Format of certificate:
  - Public key being certified
  - Name
  - Validity period
  - Location (URL) of revocation list

Contents of X.509 Certificates

- An X.509 v3 certificate binds a **distinguished name (DN)** to a public key
- A DN is a series of name-value pairs,
  - such as uid=doe
  - identify an entity--that is, the certificate **subject.**

Example of X.509 Distinguished Name (DN)

uid=doe, e=doe@netscape.com, cn=John Doe, o=Netscape Communications Corp.,c=US where

- uid: user ID
- e: email address
- cn: the user's common name
- o: organization
- c: country

X.509 Data Section

- The version number of the X.509 standard
- The certificate's serial number
  - Every certificate issued by a CA has a serial number that is unique among the certificates issued by that CA.
- Information about the user's public key
  - including the algorithm used and a representation of the key itself.
- The DN of the CA that issued the certificate.
- The period during which the certificate is valid
  - for example, between 1:00 p.m. on November 15, 1996 and 1:00 p.m. November 15, 1997
- The DN of the certificate subject
  - for example, in a client SSL certificate this would be the user's DN
- Optional **certificate extensions**

X.509 Signature Section

- The cryptographic algorithm, or cipher, used by the issuing CA to create its own digital signature.
- The CA's digital signature, obtained by hashing all of the data in the certificate together and encrypting it with the CA's private key

Full example in readable format

```
Certificate:
    Data:
        Version: v3 (0x2)
        Serial Number: 3 (0x3)
        Signature Algorithm: PKCS #1 MD5 With RSA Encryption
        Issuer: OU=Ace Certificate Authority, O=Ace Industry, C=US
        Validity:
            Not Before: Fri Oct 17 18:36:25 1997
            Not After: Sun Oct 17 18:36:25 1999
        Subject: CN=Jane Doe, OU=Finance, O=Ace Industry, C=US
        Subject Public Key Info:
            Algorithm: PKCS #1 RSA Encryption
            Public Key:
                Modulus:
                    00:ca:fa:79:98:8f:19:f8:d7:de:e4:49:80:48:e6:2a:2a:86:
                    ed:27:40:4d:86:b3:05:c0:01:bb:50:15:c9:de:dc:85:19:22:
                    43:7d:45:6d:71:4e:17:3d:f0:36:4b:5b:7f:a8:51:a3:a1:00:
                    98:ce:7f:47:50:2c:93:36:7c:01:6e:cb:89:06:41:72:b5:e9:
                    73:49:38:76:ef:b6:8f:ac:49:bb:63:0f:9b:ff:16:2a:e3:0e:
                    9d:3b:af:ce:9a:3e:48:65:de:96:61:d5:0a:11:2a:a2:80:b0:
                    7d:d8:99:cb:0c:99:34:c9:ab:25:06:a8:31:ad:8c:4b:aa:54:
                    91:f4:15
                Public Exponent: 65537 (0x10001)
        Extensions:
            Identifier: Certificate Type
                Critical: no
                Certified Usage:
                    SSL Client
            Identifier: Authority Key Identifier
                Critical: no
                Key Identifier:
                    f2:f2:06:59:90:18:47:51:f5:89:33:5a:31:7a:e6:5c:fb:36:
                    26:c9
    Signature:
        Algorithm: PKCS #1 MD5 With RSA E
```

Same example in 64-byte form


```
-----BEGIN CERTIFICATE-----
MIICKzCCAZSgAwIBAgIBAzANBgkqhkiG9w0BAQQFADA3MQswCQYDVQQGEwJVUzER
MA8GA1UEChMITmV0c2NhcGUxFTATBgNVBAsTDFN1cHJpeWEncyBDQTAeFw05NzEw
MTgwMTM2MjVaFw05OTEwMTgwMTM2MjVaMEgxCzAJBgNVBAYTAlVTMREwDwYDVQQK
EwhQZXRzY2FwZTENMAsGA1UECxMEUHViczEXMBUGA1UEAxMOU3Vwcml5YSBTaGV0
dHkwgZ8wDQYJKoZIhvcNAQEFBQADgY0AMIGJAoGBAMr6eZiPGfjX3uRJgEjmKlqG
7SdATYazBcABu1AVyd7chRkiQ31FbXFOGD3wNktbf6hRo6EAmM5/R1AskzZ8AW7L
iQZBcrXpc0k4du+2Q6xJu2MPm/8WKuMOnTuvzpo+SGXelmHVChEqooCwfdiZywyZ
NMmrJgaoMa2MS6pUkfQVAgMBAAGjNjA0MBEGCWCGSAGG+EIBAQQEAwIAgDAfBgNV
HSMEGDAWgBTy8gZZkBhHUfWJM1oxeuZc+zYmyTANBgkqhkiG9w0BAQQFAAOBgQBt
I6/z07Z635DfzX4XbAFpjlRl/AYwQzTSYx8GfcNAqCqCwaSDKvsuj/vwbf91o3j3
UkdGYpcd2cYRCgKi4MwqdWyLtpuHAH18hHZ5uvi00mJYw8W2wUOsY0RC/a/IDy84
hW3WehBUqVK5SY4/zJ4oTjx7dwNMdGwbWfpRqjd1A==
-----END CERTIFICATE-----
```


Public Key Infrastructures (PKIs)

- For our purposes (Grid), an X509 certificate contains the following primary information:
  - The CA
  - The Subject
  - The Public Key
  - A Digital Signature

  + validity, version, algorithms


```
Certificate:
    Data:
        Version: 1 (0x0)
        Serial Number: 6 (0x6)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: C=gb, O=glasgow, OU=admin, CN=soa
        Validity
            Not Before: Jun  1 13:38:45 2006 GMT
            Not After : Jul 10 13:38:45 2010 GMT
        Subject: C=gb, O=glasgow, OU=dcs, CN=anthony
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:df:f7:e6:f3:3c:69:da:62:ba:d3:bf:e8:1a:9c:
                    3d:2e:8b:a5:51:21:6e:f0:8a:66:c8:1e:0b:70:95:
                    0e:f3:c7:78:63:e0:fc:bc:06:dd:42:fc:cd:c3:11:
                    ec:f6:b6:b2:be:2e:09:35:aa:35:ad:ae:39:b0:ea:
                    04:93:51:1e:85:96:40:26:b8:42:dd:ba:a7:56:eb:
                    c5:69:a6:b9:ff:5d:0d:f0:88:93:bd:6e:a4:2c:b2:
                    ae:2d:f0:70:73:7e:b5:01:21:b5:e2:5b:89:2a:e3:
                    0f:a0:1f:95:71:31:cd:43:a2:16:57:d6:94:e2:e9:
                    72:c5:8c:21:83:3a:31:c4:85
                Exponent: 65537 (0x10001)
    Signature Algorithm: md5WithRSAEncryption
        8f:74:21:08:db:ba:e4:af:4e:d5:46:90:ad:0f:1e:8f:97:09:
        52:eb:f6:f5:c3:4e:bc:07:68:c7:4f:fe:1c:ab:f5:76:3f:ed:
        2b:62:81:6f:2c:ad:78:a4:06:7e:4b:7a:96:4b:17:96:5d:db:
        34:b3:fe:5b:72:85:0d:32:f2:78:2a:20:32:43:87:2d:0c:70:
        96:a8:5f:63:82:cb:3e:c0:5f:9a:2e:bb:50:dd:79:b8:2a:d0:
        a4:32:6c:e6:b3:72:ab:31:70:16:52:f3:85:20:98:58:38:f1:
        6a:b1:c1:0c:81:83:f4:70:f1:2b:fb:0d:93:ae:35:8a:32:56:
```


Certificate Authorities

- A Certificate Authority (CA) is a third party that signs certificates and ensures that the subject name and public key actually belong to that person
  - How?
    - The old fashioned way…
    - Example… The UK e-Science Certificate Authority
      - Initial contact – application (online)
      - Credential verification (IN PERSON)
        - » Go to CA or Regional Authority (RA)
      - Issue – download (online)
        - » UK CA requires the application and issuing terminals to be the same (this is where the PRIVATE key of your certificate is)

Certificate Authorities

- A CA may delegate Regional Operators to confirm people's identities
  - Saves for example having to travel from Glasgow to Oxford if one want a certificate
  - CA records a piece of personal identification for their records
    - Passport, Driving Licence, Staff/Student Matric Card
    - CA extends an existing ID infrastructure – good thing

Types of Certificate

- Personal Certificates
  - Identify users
  - Allow digital signatures
- Host (server) certificates
  - To identify machines
- Subordinate CA certificates
- Certificate Policy statement (CPS)
  - Document on conditions under which certs are issued and level of assurance

Certificate Authorities

- A CA also is in charge of *revoking* certificates
  - CA publishes a Certificate Revocation List
    - Download to your browser
    - Shows all invalid certificates in the organisation
- A CA MUST be explicitly trusted by the system
  - Trusted Root CAs list in Windows
  - In local SimpleCA for grid files are in
    - /etc/grid-security/certificates
  - Certificate cannot be used until the CA's root certificate has been accepted as trusted
    - Accepted very much like Software Licences i.e. nearly always!

Digital Signatures

- CAs confirm the certificate's authenticity by digitally signing it
  - CA computes a hash of the certificate using an agreed (non-secret) algorithm
  - CA encrypts this hash with their private key and appends to bottom of certificate
  - Recipient computes their own hash of the info
  - Recipient decrypts the hash the CA sent (with the CA's public key) and compares with their own
    - Proves the CA signed the info and the info hasn't been tampered with
      - Encryption of the info is optional (for privacy)

How CA Certificates are used to establish trust

- Certificate authorities (CAs) are entities that validate identities and issue certificates.
- They can be either independent third parties or organizations running their own certificate-issuing server software (such as the Netscape Certificate Server).
- A list of third-party certificate authorities is available at https://certs.netscape.com/client.html
  - Verisign most common used in US

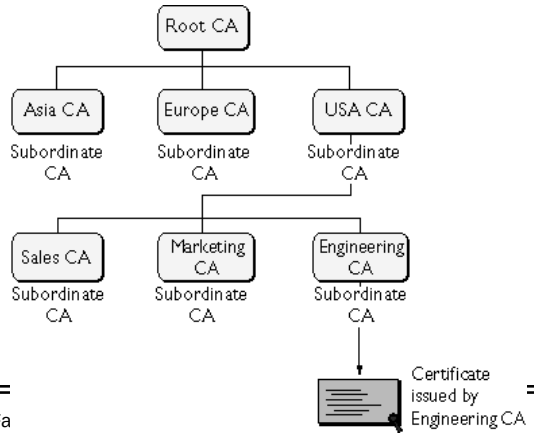
How CA Certificates are used to establish trust

- Any client or server software that supports certificates maintains a collection of **trusted CA certificates.**
- These CA certificates determine which other certificates the software can validate--in other words, which issuers of certificates the software can trust.
- In the simplest case, the software can validate only certificates issued by one of the CAs for which it has a certificate.
- It's also possible for a trusted CA certificate to be part of a chain of CA certificates, each issued by the CA above it in a certificate hierarchy.

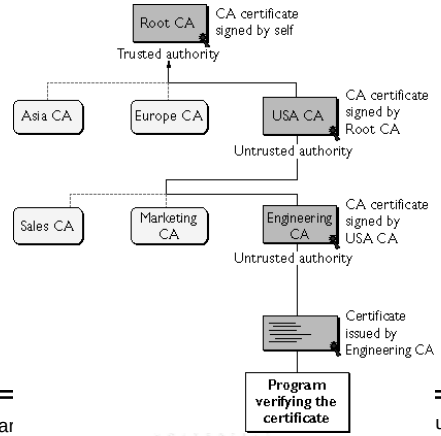
CA Hierarchies

- In large organizations, it may be appropriate to delegate the responsibility for issuing certificates to several different certificate authorities. For example,
  - the number of certificates required may be too large for a single CA to maintain
  - different organizational units may have different policy requirements
  - or it may be important for a CA to be physically located in the same geographic area as the people to whom it is issuing certificates.
- It's possible to delegate certificate-issuing responsibilities to subordinate CAs.
- The X.509 standard includes a model for setting up a hierarchy of CAs

X.509 CA Hierarchy Example



Certificate Chain Example



What happens in a certificate chain

- Each certificate is followed by the certificate of its issuer.
- Each certificate contains the name (DN) of that certificate's issuer,
  - The same as the subject name of the next certificate in the chain.
  - the Engineering CA certificate contains the DN of the CA (that is, USA CA), that issued that certificate.
  - USA CA's DN is also the subject name of the next certificate in the chain.
- Each certificate is signed with the private key of its issuer.
- The signature can be verified with the public key in the issuer's certificate, which is the next certificate in the chain.
  - The public key in the certificate for the USA CA can be used to verify the USA CA's digital signature on the certificate for the Engineering CA.

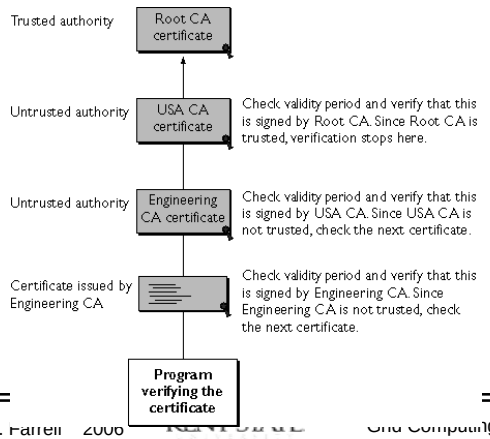
Verifying a Certificate Chain

1. The certificate validity period is checked against the current time provided by the verifier's system clock.
2. The issuer's certificate is located. The source can be
   - either the verifier's local certificate database (on that client or server)
   - the certificate chain provided by the subject (for example, over an SSL connection).
3. The certificate signature is verified using the public key in the issuer's certificate.
4. If the issuer's certificate is trusted by the verifier in the verifier's certificate database, verification stops successfully here.

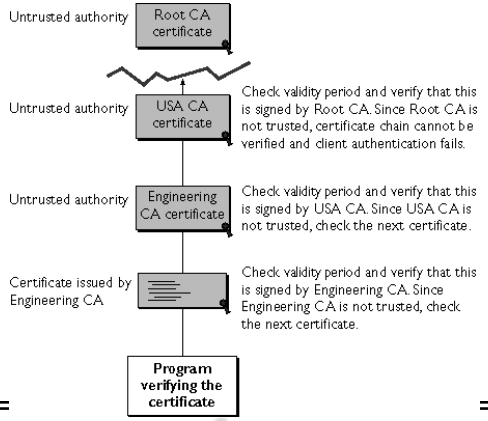
   Otherwise, the issuer's certificate is checked to make sure it contains the appropriate subordinate CA indication in the Netscape certificate type extension

Chain verification returns to step 1 to start again, but with this new certificate.

A Valid Certificate Chain



An Invalid Certificate Chain



Managing and Issuing Certificates

- The set of standards and services that facilitate the use of public-key cryptography and X.509 v3 certificates in a networked environment is called the **public key infrastructure (PKI).**
- The process for issuing a certificate depends on the certificate authority that issues it and the purpose for which it will be used.

A problem

- Are there any pitfalls to digital certificates?
  - Can we alter their contents?
    - No, the CA signed the certificate thus ensuring its integrity
  - Can we spoof?
    - You will need your own CA, and if the application doesn't trust it, your certificates won't work. So no.
  - What can we do?
    - STEAL IT!
      - Someone who holds your digital certificate (and private key) may safely assume your identity on the Grid

Proxies

- Temporary credentials
  - If the certificate is that sensitive, we shouldn't fire it off across the Grid where you will lose track of it.
  - Instead create a PROXY CERTIFICATE
    - A proxy certificate is a new version of your certificate, but this time with your original certificate as the CA
    - It has a VERY SHORT lifetime
      - So if it is stolen, it can only be used for a short time (24 hrs)
      - Revocation is unnecessary as it expires so quickly
        - » You don't need to revoke the ORIGINAL certificate as it was only the proxy (which is different) which was stolen
    - Proxy certificates have another feature which becomes obvious when you actually look at what a proxy looks like…

Proxies

- A proxy certificate seems to be 3 certificates in one!
  - Not so, the proxy certificate is only the first certificate
  - The second is the private key of the proxy
  - The third is the original certificate the proxy was made from
- Why??

-----BEGIN CERTIFICATE-----
MIIB6zCCAVSgAwIBAgIEV3PyzjANBgkqhkiG9w0BAQQFADA/MQswCQYDVQQGEwJn
YjEQMA4GA1UEChMHZ2xhc2dvdzEMMAoGA1UECxMDZGNzMRAwDgYDVQQDEwdhbnF
b255MB4XDTA2MDcwNzE0MDIzMVoXDTA2MDcwODAyMDczMVowVDELMAkGA1UEBhM
Z2IxEDAOBgNVBAoTB2dsYXNnb3cxDDAKBgNVBAsTA2RjczEQMA4GA1UEAxMHYW50
aG9ueTETMBEGA1UEAxMKMTQ2NzIxNjU5MDBcMA0GCSqGSIb3DQEBAQUAA0sAMEg
QQCWU12wPJzHv500RzFYn9bA/acRm1ZoNKszpum02cChyvDh58XRfdA+uVBhPHvx
udqZtN/GHaKaz/2/+j3wmbsRAgMBAAGjIzAhMB8GCisGAQQBm1ABgV4BAf8EDjAM
MAoGCCsGAQUFBxUBMA0GCSqGSIb3DQEBBAUAA4GBALzb0FfjQ6gx3rNgcMxi6vec
2Dtz+Y03TpkkAeeJJhn5mHRB6oDJGTKM/t7wjxgdkhp3Bbd7dA3D6psrslMrum+2
kk4FG6khqspWsMeyTUvepA7Qo3mO7YyXzbFlXwdU18jJyOXtl/9xvxvGa7lX3blg
zFiEOZslvO3MSwojrEZp
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
MIIBOgIBAAJBAJZTXbA8nMe/nTRHMVif1sD9pxGbVmg0qzOm6bTZwKHK8OHnxdF9
0D65UGE8e/G52pm038YdoprP/b/6PfCZuxECAwEAAQJAGD9MLmAofcO3jLYrDHuD
vU9swldHAbn1H3i927D/9FizZqpkEe+KhD59+gE91eL8mw6xagj3vkTi6HS0F3E
0QthAMV5ZrSFczh1VAyVDkXyQxvR5qgPCp5GJwO/Psamo8jFAiEAwuCYoN75Ox2
PFnSVbHy60pm0rAEkIrkLu7xKqYcVd0CIFjsultAXQpm1m/JZZJ6f51HzE+1MFB
Rg8h9pruH4MhAiEAtsyhWeRiPCu/AP90eKQSbLI+2BT6Qye1wXyrRkll/RUCID5+
RNLI8mR2xUx4HRyW/RcJDLaDtBAr0NO7gPUvdYd3
-----END RSA PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
MIIB6zCCAVQCAQYwDQYJKoZIhvcNAQEEBQAwPTELMAkGA1UEBhMCZ2IxEDAOBgNV
BAoTB2dsYXNnb3cxDjAMBgNVBAsTBWFkbWluMQwwCgYDVQQDEwNzb2EwHhcNMDYw
NjAxMTMzODQ1WhcNMTAwNzEwMTMzODQ1WjA/MQswCQYDVQQGEwJnYjEQMA4GA1
ChMHZ2xhc2dvdzEMMAoGA1UECxMDZGNzMRAwDgYDVQQDEwdhbnRob255MIGfMA0G
CSqGSIb3DQEBAQUAA4GNADCBiQKBgQDf9+bzPGnaYrrTv+ganD0u6VRlW7wimbl
HgtwlQ7zx3hj4Py8B1C/M3DEez2trK+LgkUgWfrjmw6gSTUR6FlkAmuELduqdW
68Vpprn/xQ3wiJO9bqQssq4t8HBzfrUBIbXiW4kq4w+gH5VxMc1DohZX1pTi6XLF
jCGDOjHEhQIDAQABMA0GCSqGSIb3DQEBBAUAA4GBAI9OlQjb+uSvTtVGkK0PHo+X
CVLr9vXDTrwHaMdP/hyr9XY/7StigW8srXikBn5LepZLF5Zd2zSz/ltyhQ0y8ngq
tDJDhy0McJaoX2OCyz7AX5ouu1Ddebgq0KQybOazcqsxcBZS84UgmFg48WqxwQyB
gRw8SV7DZOuhYyVhsn
-----END CERTIFICATE-----

Proxies

- Proxies propagate through the grid by signing new versions of themselves
  - Using the proxy certificate and its private key, any number of new proxies may be spawned across each required resource.
  - Acheives two goals
    - DELEGATION
      - Proxy acts on your behalf
    - SINGLE SIGN-ON
      - You only need to logon to the Grid once

Firewalls

- Hardware or software component added to a network to prevent_ some communications forbidden by an organisation's administrative policy
- Two types
  - traditional firewall : a dedicated network device or computer on the boundary of a network
    - filters all traffic entering the networks
    - Works at some level of network stack
      - Network level : IP packet filtering
      - Transport (application) level : intercepting specific TCP socket traffic e.g. rlogin, telnet, www
  - personal firewall : a software application that filters traffic entering or leaving a single computer

Proxy/Address Translation

- A proxy device, in hardware or software, can act as a firewall by responding to input packets) in the manner of an application, whilst blocking other packets
- It can be configured to pass certain packets to specific servers
- This reduces the risk of having internal systems compromised if there is misuse or misconfiguration
- Using private network addresses internally also helps