High Performance Computing

Course Notes 2008-2009

Grid Security
Secure communication

Four major aspects in a secure communication

- **Privacy**: only the sender and the receiver should be able to understand the conversation
- **Integrity**: the receiving end must be able to know *for sure* that the message it is receiving is not tampered with
- **Authentication**: ensure that the parties involved in the communication are who they claim to be
- **Authorization**: decide whether and when a user is allowed to perform a certain task
Encryption Algorithms

- Key-based algorithms: A key is needed to encrypt and decrypt the messages
  - Symmetric key-based algorithms: the same key is used for encryption and decryption
    - Disadvantages: 1) only guarantee privacy; 2) have to agree on a common key;
    - Advantages: Fast, simple to implement
  - Asymmetric algorithms: a different key is used to encrypt and decrypt the message
    - Public/private key pair
    - Can avoid the disadvantages of symmetric algorithms,
    - do not work as fast as symmetric algorithms
Pubic key-based asymmetric algorithm

- Private key: the key is only known to its own
- Public key: the key is known to everyone
- The message encrypted using the public key can only be decrypted using the private key
Digital Signatures

The digital signature for a message is generated in two steps:

- A message digest is generated from the message
- The message digest is encrypted using the sender's private key

A digital signature is

- attached to a message
- used to find out if the message was tampered with during the communication (guarantee integrity)

The receiver does the following

- Using the sender's public key, decrypts the digital signature to obtain the message digest generated by the sender
- Generate a message digest of the received message by using the same message digest algorithm used by the sender
- Compares these message digests
Digital Signatures

- Integrity is guaranteed by digital signature
- Message itself can be encrypted by the receiver’s public key to guarantee privacy
Public-key system and digital signature

The combination of the public-key and digital signature

- can guarantee both privacy and integrity
- but only “weak” authentication
  - If we can decrypt the digital signature using the sender’s public key, the digital signature is encrypted by the sender’s private key (authentication)
  - Anybody can publish a public key and claim that the key belongs to the sender (weak)

A digital certificate can be used to guarantee a strong authentication
Digital Certificate

- Digital certificate is used to certify that a public key indeed belongs to somebody

- The X.509 certificate format is used to encode the digital certificates

- An X.509 certificate includes
  - Subject: distinguished name of the user
  - Subject's public key
  - Certificate Authority’s subject
  - Digital signature of CA

- The format of the distinguished name
  - O=University of Warwick, OU=Department of Computer Science, CN=Ligang He
    - O: organisation, OU: Organisation Unit, CN: Common Name
How to Use Digital Certificate

- Four pieces of information are transported from the sender to the receiver
  - Message
  - Digital signature of the message (attached to the message)
  - Digital certificate of the sender
  - Digital signature of the digital certificate (enclosed in the certificate);

- The receiver uses the CA’s public key to decrypt the digital signature of the sender’s digital certificate and make sure that the certificate is genuine (therefore verify that the public key belongs to sender)

- It is all about trust, we have to trust CA
GSI (Grid Security Infrastructure) in Globus

- GSI can support
  - Transport-level and message-level encryption
  - Authentication through X.509 digital certificates
  - Authorization
  - Credential delegation
  - Single sign-on
Transport-level and message-level security

Transport-level security

Message-level security
Authorization

→ At the server side, it is about who is allowed to use a certain service

→ At the client side, it is about which service is allowed to be invoked

→ GSI supports authorization in both the server-side and the client-side

☐ **Server-side authorization**
  - None
  - Self: the client’s identity is the same as the service’s identity
  - Gridmap: a list of 'authorized users'
  - Identity authorization: the client’s identity matches a specified identity
  - Host authorization: a host credential presented by the client matches a specified hostname

☐ **Client-side authorization**
  - None
  - Self
  - Identity authorization
  - Host
Delegation and single sign-on (proxy certificates)

Proxy certificates

- Certify that B is acting on behalf of A
- very similar to the X.509 digital certificates except that it’s not signed by a Certificate Authority, but by the end user A
- the lifetime of the proxy certificate is usually very limited (for example, to 12 hours)

Proxy certificates achieve both single sign-on and delegation
How to generate a proxy certificate

- B generates a certificate request and sends it to A.
- Supposing A agrees to delegate its credentials to B; A will use its private key to digitally sign the certificate request.
- A sends the signed certificate back to B.
- B can now use the proxy certificate to act on A's behalf.
How to validate a proxy certificate

- B sends C the proxy certificate and the certificate of A

- C uses Certificate authority’s public key to decrypt A’s certificate and verify that A’s public key really belongs to A

- C uses A’s public key to validate that proxy certificate is signed by A

- If succeed, then B is acting on behalf of A
Adding security to a service does *not* affect the service interface.

The *security descriptor* is used to specify the security configuration for a secure service.

In order to modify some security aspects of a service, only need to modify the security descriptor, *not* the Java files.
The security descriptor file

<securityConfig xmlns="http://www.globus.org">
  <authz value="none"/>
</securityConfig>
Changing the WSDD file

Add the following parameter to the WSDD file

<parameter name="securityDescriptor"
  value="etc/org.globus.examples.services.security.first/security-config-first.xml"/>