CHAPTER 1

THESIS INTRODUCTION

1.1 INTRODUCTION

In the face of increasing global competition, corporations in every industrial sector are focusing on their core-competency and turning to partnerships to complement internal abilities. Specifically, on the engineering front, we are seeing increased out-sourcing and supplier partnerships, often with geographically distributed partners. The challenge is to swiftly create agile partnerships and link the different service providers in a virtual corporation. Often, this translates into corporations needing to access engineering software tools in a distributed environment. Today, there exists a large number of stand-alone engineering software applications that have been developed over the years. However, most of them have not been developed with a view to making them available in a distributed environment, wherein users can access them from geographically remote locations. It is important to distinguish the term "distributed environment," as used here, from the many distributed software applications that are accessible to users within the same network, whether it be a Local Area Network (LAN) or even a Wide Area Network (WAN). The distinguishing requirement here is the ability of users to access software tools without they being within the local network where the tools reside. The Internet, with its open architecture and international reach, is positioned very well to be the medium for integrating the distributed tools. Thus, in the near future, there is likely to be an increased demand for accessing these software tools through the Internet.

Remote access of software applications also sidesteps another major problem; namely, that of software distribution and maintenance. When software tools are available in a distributed environment, one can access the tools through the network from a few central servers without needing to install a local copy. This also simplifies software maintenance, in that software upgrades made on the server becomes instantly available to all users across the network. Further, since the Intenet extensively uses open standards like Transport Control Protocol/Internet Protocol (TCP/IP) and, increasingly, the "Write Once, Run Anywhere"™ environment of Java™, portability of applications is becoming easier to achieve.

Thus, there is a need for developing a framework for making these engineering software tools and programs available through the Internet. In other words, there is a need for a framework for supporting network-centric application access to make the already existing software tools available to a wider audience who may be geographically distributed. This will not only facilitate distributed and collaborative engineering, but also reduce the maintenance problem to a great extent.

Internet applications use the TCP/IP protocol to communicate between different computers connected to the Internet. TCP/IP is an inherently insecure protocol because the data is routed through several intermediate internets, and there is no built-in provision for peer-authentication or data-integrity checking. There are therefore no guarantees of end-to-end service between two Internet-connected computers. The data being transmitted is open to be viewed or modified by a third party. Thus, the issue of security is central to network-centric application access, if the Internet is used to communicate confidential or proprietary data. More specifically, the real problems associated with security are as follows:

1. **End-point Security**: The data and executables at either end of the two communicating computers must be trustworthy. In the case where the end-point executables are corrupted, no amount of network security will help in securing the session.

2. **Network Security:** The major concerns in network security are as follows:

   Authentication: The users of the services need to be assured that (1) the remote host they connect to is the one they intended to connect (i.e., the host is not an impostor) and (2) the data or software modules that they download is really what they expected and not a "Trojan Horse" (i.e., an illegal program or a virus masquerading as a legal application).

   Data Security: The users of the services need to be assured that the data that they send to the remote host, is not copied by a malicious third party while in transit through the network.

   Data Integrity: The users of the services need to be assured that the data that the users send to the remote host does not get corrupted by a malicious third party while in transit through the network.

1.2 PROBLEM STATEMENT AND OBJECTIVE

There is a need for a framework which can provide secure access to remote engineering software tools via the Internet. The framework should include measures to ensure a reasonable level of network security. At the same time, the framework should be extensible enough to incorporate support for access to new software tools as and when required.

The objective of this thesis is to create a network-centric interface for making a set of software applications, that process an input file in some way to return an output file, accessible to remote users, and at the same time, to ensure strong security. Furthermore, the interface should be portable and transparent. Users should be able to access the software tools through the interface irrespective of their local hardware platform.

1.3 SOLUTION OVERVIEW

The proposed solution is based on a Java client-server architecture. A digitally-signed ("trusted") Java applet, downloaded over the Web, serves as client. It connects to its application server from a server suite, named *NetCAD*. These servers run on the host machine, which is the machine from where the applet gets downloaded. The basic scheme is then as follows: The client reads a file from a local user-specified location and sends it to the server. The server then invokes the right file processing program to process the file. Once the file has been processed, it is returned to the client, which saves it to a local user-specified location. Figures 1.1 and 1.2 presents an overview of this process.

Network security is achieved by incorporating public-key and symmetric cryptography. The applet that serves as the client is digitally signed, and the signature is used to authenticate the applet as trusted. Access control is achieved by defining a protected domain within which the applet accesses the local resources.

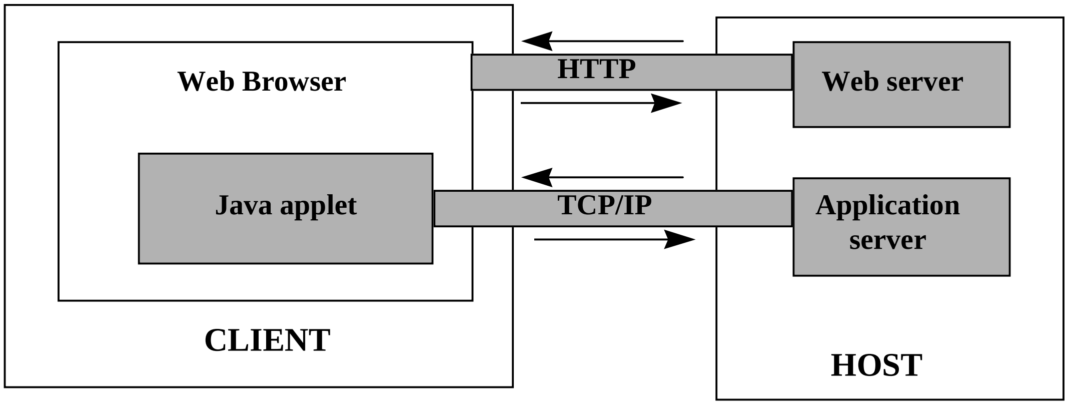

Figure 1.1 Client-server model

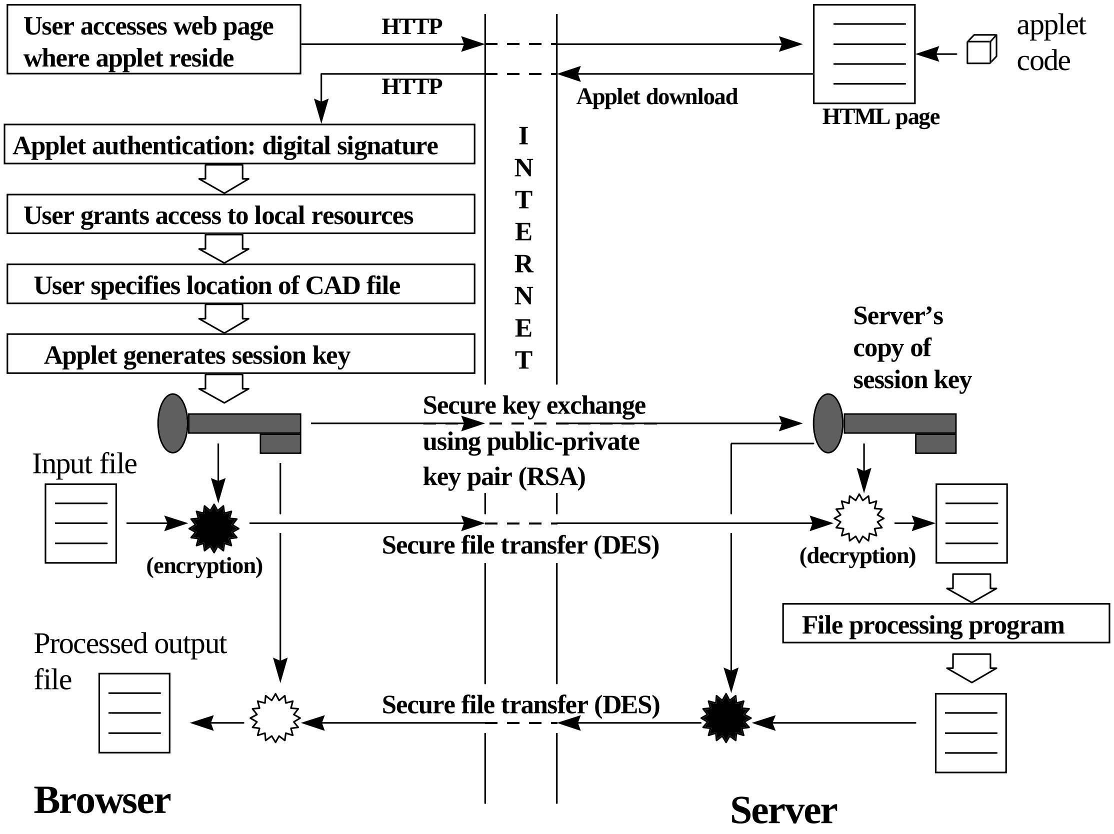
User accesses web page where applet reside
HTTP
HTTP
Applet download
HTML page
applet code
Applet authentication: digital signature
User grants access to local resources
User specifies location of CAD file
Applet generates session key
I N T E R N E T
Server's copy of session key
Secure key exchange using public-private key pair (RSA)
Input file
(encryption)
Secure file transfer (DES)
(decryption)
File processing program
Processed output file
Secure file transfer (DES)
Browser
Server

Figure 1.2 Overview of client-server interaction in NetCAD system.

A session key, based on the Data Encryption Standard (DES) algorithm, is generated by the client at run time, and is exchanged with the server using a public-private key pair based on the Rivest-Shamir-Alderman (RSA) algorithm. The data is, then, transmitted accross the network, encrypted using the session key. In general, encryption with symmetric cryptography algorithms, such as DES, is significantly faster than with public-key cryptography algorithms, such as RSA. Therefore, under the above scheme, the session key is used to encrypt the bulk of the data, while the RSA key pair is only used to securely exchange the session key.

This scheme has been demonstrated by implementing a server that repairs .STL files. The server can be very easily extended to handle other types of requests, the criteria being that a user can remotely access software in a trusted environment. Although the model described in this thesis can be used to access any application that fulfils the criteria mentioned in Section 1.2, the prototype of the model has been developed with the view to provide Web-based access to a CAD/CAM application, and hence the name “NetCAD.”

1.4 THESIS ORGANIZATION

The rest of this thesis presents the implementation of the NetCAD client-server system and the relevant concepts and past work related to the area of network-centric application access.

Chapter 2 gives a brief introduction to the structure of the Internet, its inherent lack of security, and topics relevant to achieving Internet security. It also discusses the different existing technologies that can be used for remote application access, along with the relevant literature review.

Chapter 3 details the implementation of the NetCAD client-server system.

Chapter 4 presents a case study in the use of the NetCAD system for providing access to a stand-alone rapid prototyping software program via the Web browser. It also presents an analysis of the system.

Chapter 5 concludes the thesis and highlights the contributions of this study.