# Implementation of steganography for audio file format computer science essay

Technology, Computer

# Abstract

The project entitled Audio Steganography is the application developed to embed an audio file in another audio signal. It is concerned with embedding information in an innocuous cover Speech in a secure and robust manner. This system makes the Files more secure by using the concepts Steganography and Cryptography.

Steganography, poor cousin of Cryptography is the art of hiding messages inside other messages such that the very existence of the message is unknown to third party. The goal of cryptography is to make data unreadable by a third party, the goal of Steganography is to hide the data from a third party Through the use of advanced computer software, authors of images and software can place a hidden trademark in their product, allowing them to keep a check on piracy. This is commonly known as watermarking. Hiding serial numbers or a set of characters that distinguishes an object from a similar object is known as finger printing. Together, these two are intended to fight piracy. The latter is used to detect copyright violators and the former is used to prosecute them. But these are only examples of the much wider field of Steganography.

The cover data should not be significantly degraded by the embedded data, and the embedded data should be as imperceptible as possible. The embedded data should be as immune as possible to modifications from intelligent attacks or anticipated manipulations. Thus it is necessary that the hidden message should be encrypted.

Introduction

1. Introduction – Synopsis

Encryption of data plays a vital role in the real time environment to keep the data out of reach of unauthorized people, such that it is not altered and tampered and sending the in audio format is most secured way to transfer the data through the network. The Audio Steganography is software, which tries to alter the originality of the file into some encrypted form and embed the file into an audio file. Then the users can easily and securely carry the compressed data wherever he wants. The major task of the Audio Steganography is to provide the user the flexibility of passing the information implementing the encryption standards as per the specification and algorithms proposed and store the information in a form that is unreadable. The Application should have a reversal process as of which should be in a position to de embed the data file from audio file and decrypt the data to its original format upon the proper request by the user. While the Encryption and Decryption is done the application should confirm the standards of authentication and authorization of the user.

The Entire application should strive to achieve a user friendly Graphical User Interface, which need to be in a self-learning mode for the end user. The System Should provide all the functional standards of proper navigation with in the environment, which makes it possible for the users to have a smooth

flow while working under the environment. The Overall system should provide proper menu based navigation for easier navigation and operation. The Application should be designed in such a way that, as soon as it starts create a Buffer and associate this buffer to some homogeneous data environment, the application should ask the user for the Encryption Key details and should start its functionality upon the logistics that are provided with in this key. The key should be designed in such a way that it prevents the unauthorized persons from stealing the information at any point of time. This is some part of securing the data from third party people. And the other way of securing the data is using Steganography in which embedding the encrypted file in to a audio file. If any one track that file they only see the audio file not the data.

The application of De-embedding, Decryption and de compress should be a reverse process at the other end and should be translated only when the receiver of the data applies the proper reversal key. The Decryption process should have a log-based methodology that will take care of any errors that may be encountered while the system is under utilization and should record all those events, which are above the general standards of security.

This system basically uses the Blowfish encryption algorithm to encrypt the passwords. This algorithm is a 64-bit block cipher with a variable length key. This algorithm has been used because it requires less memory. It uses only simple operations, therefore it is easy to implement.

1) Blowfish Algorithm Implementation Module

2) Steganography Module

3) Compression Module

4) GUI Module

System Configuration

2. System Configuration

2. 1 Software Requirements:

Operating System

Windows NT/2000 (Client/Server).

2. 2 Hardware Requirements:

Software requirements

Front-end: Java J2SDK 1. 5, Swings.

System Configuration

Pentium III Processor with 700 MHz Clock Speed

256 MB RAM, 20 GB HDD, 32 Bit PCI Ethernet Card.

System Analysis

Feasibility Study

Fact Finding Techniques

In this system we are going to develop a facility to a user that he will not face any difficulty at the time of usage like data missing, one way contacts,

one view contacts. As we are developing this system with an encoding technique of images the user will not be bothered on which camera support is using, as well in sound. As we are maintaining one technique of speed controlling the frame relay will not be a problem for the user like over speed display, hanged display.

3. 1 Feasibility Study

A feasibility study is a high-level capsule version of the entire System analysis and Design Process. The study begins by classifying the problem definition. Feasibility is to determine if it’s worth doing. Once an acceptance problem definition has been generated, the analyst develops a logical model of the system. A search for alternatives is analyzed carefully. There are 3 parts in feasibility study.

3. 1. 1 Operational Feasibility:

Question that going to be asked are

Will the system be used if it developed and implemented.

If there was sufficient support for the project from the management and from the users.

Have the users been involved in planning and development of the

Project.

Will the system produce poorer result in any respect or area?

This system can be implemented in the organization because there is adequate support from management and users. Being developed in Java so that the necessary operations are carried out automatically.

3. 1. 2 Technical feasibility

Does the necessary technology exist to do what is been suggested

Does the proposed equipment have the technical capacity for using the new system?

Are there technical guarantees of accuracy, reliability and data security?

The project is developed on Pentium IV with 256 MB RAM.

The environment required in the development of system is any windows platform

The observer pattern along with factory pattern will update the results eventually

The language used in the development is JAVA 1. 5 & Windows Environment

3. 1. 3 Financial and Economical Feasibility

The system developed and installed will be good benefit to the organization. The system will be developed and operated in the existing hardware and software infrastructure. So there is no need of additional hardware and software for the system.

Existing and Proposed System

3. 2 Existing System

In the traditional architecture there existed only the server and the client. In most cases the server was only a data base server that can only offer data. Therefore majority of the business logic i. e., validations etc. had to be placed on the clients system. This makes maintenance expensive. Such clients are called as ‘ fat clients’. This also means that every client has to be trained as to how to use the application and even the security in the communication is also the factor to be considered.

Since the actual processing of the data takes place on the remote client the data has to be transported over the network, which requires a secured format of the transfer method. How to conduct transactions is to be controlled by the client and advanced techniques implementing the cryptographic standards in the executing the data transfer transactions. Present day transactions are considered to be “ un-trusted” in terms of security, i. e. they are relatively easy to be hacked. And also we have to consider the transfer the large amount of data through the network will give errors while transferring. Nevertheless, sensitive data transfer is to be carried out even if there is lack of an alternative. Network security in the existing system is the motivation factor for a new system with higher-level security standards for the information exchange.

3. 3 Proposed System

The proposed system should have the following features. The transactions should take place in a secured format between various clients in the network. It provides flexibility to the user to transfer the data through the network very easily by compressing the large amount of file. It should also identify the user and provide the communication according to the prescribed level of security with transfer of the file requested and run the required process at the server if necessary. In this system the data will be send through the network as a audio file. The user who received the file will do the operations like de embedding, decryption, and decompress in their level of hierarchy etc.

Analysis

Report

3. 4 System Analysis

People for long time have tried to sort out the problems faced in the general digital communication system but as these problems exist even now, a secured and easy transfer system evolved and came to be known as the Encryption and Decryption of the data and converting the file to audio format to be transferred using the cryptographic standards and Steganography. The advantages of this Audio Steganography are:

High level Security

Cost effective transfer

In this fast growing world where every individual free to access the information on the network and even the people are technically sound enough in hacking the information from the network for various reasons. The organizations have the process of information transfer in and out of their network at various levels, which need the process to be in a secured format for the organizational benefits.

If the organizations have the Audio Steganography System, then each employee can send the information to any other registered employee and thus can establish communication. The audio file that the employee sends reaches the destinations within no time in an audio file format where the end user need to de embed the file, decrypt it and de compress and use for the purpose. The various branches of the organization can be connected to a single host server and then an employee of one branch can send files to the employee of another branch through the server but in a secured format.

System Design

4. System Design

The System Design includes the maintenance of the secure file transfer service with a prescribed encryption format and split at the interested level of encryption, and embed process and the receiving service at the other end with de-embed and decryption process. The design also includes the provision of facility to the user to manipulate the concerned information according to his personal use and communication process.

The design also needs to provide the communication channel to the user to communicate with other registered users through the mailing services in a reliable and secured format. Authorization and authentication services are preferred most for this purpose. The System Design includes the maintenance authorization services, File and directory services with a prescribed encryption format at the interested level of encryption and the receiving service at the other end with decryption process. The design also includes the provision of facility to the user to manipulate the concerned information according to his personal use.

The design of Audio Steganography system, basically involve the interface architecture, Security services, and communication system.

In the interface design we involve with the design of the user interface with GUI standards and a proper navigation system where the user need to enter into the flow of transactions authorization services are check and further access is provided into the system. Then the user needs to select into the operations provided through the GUI where compression, encryption, embedding, de-embedding, Decryption, Decompressing and sending of the file, General information and exit are provided.

Here the Encryption and decryption and services are provided connecting to the security services module where the encryption and decryption are carried out using the cryptographic standards implementing the Blowfish algorithm.

After the compression process is completed the user is selecting the file for encryption. After encryption of the file is completed the user is to select the file for embedding it to the audio file and sending through the network to the desired user by specifying the targeted users' system IP address in the panel designed. Then the system gets connected to the targeted user and delivers the file in audio format after which the user working with the Audio Steganography software should go for the option De-Embed Files and decrypt the file by selecting the file path by which the file gets decrypted and decompress the file and is viewed on the system.

4. 1 System Description

The Audio Steganography system is designed basically in four different modules they are GUI module, Compression Module, Security System module, Steganography Module, Connection Manager Module.

GUI Module basically deals with the design of the interface which include the service of providing the user with the flexibility of accessing the file system and selecting the required file for the transfer. It should also provide the system to collect the information from the user to check the authorization in providing the access to the file system. The interface is also to consider the design, which include the services of sending and receiving of the files with encryption and decryption standards.

The Compression module basically deals with the compress and decompresses the file, which is used to send the file very easily which reduces the uploading time.

Security implementation module considers the implementation of the encryptions and decryption standards in transfer the files from one system to another in a distributed environment. The system design, even need to support the user to select the level of encryption he/she needs to perform depending upon the file to be transferred. The basic algorithm used in this purpose is the Blowfish where the user can enter the key depending upon level encryption he is interested.

The Connection Manager deals with the architecture, which supports the system to identify the end users for the communication and establish the communication. Connection and disconnection of the communication channel between the users for the access of file system and file transfer services. The Connection Manager receives the IP address to be connected and the file to be sent then establishes the connection and transfers the file.

Functional Requirements:

The Modules of the system are:

1) Blowfish Algorithm Implementation Module

2) Steganography Module

3) Compression Module

4) GUI Module

Blowfish Algorithm:

Blowfish is a block cipher that encrypts data in 8-byte blocks. The algorithm consists of two parts: a key-expansion part and a data-encryption part. Key expansion converts a variable-length key of at most 56 bytes (448 bits) into several sub key arrays totaling 4168 bytes.

Blowfish has 16 rounds. Each round consists of a key-dependent permutation, and a key- and data-dependent substitution. All operations are XOR’s and additions on 32-bit words. The only additional operations are four indexed array data lookups per round.

Sub keys:

Blowfish uses a large number of sub keys. These keys must be precomputed before any data encryption or decryption. The P-array consists of 18 32-bit sub keys: P1, P2,..., P18. There are also four 32-bit S-boxes with 256 entries each: S1, 0, S1, 1,..., S1, 255; S2, 0, S2, 1,..,, S2, 255; S3, 0, S3, 1,..., S3, 255; S4, 0, S4, 1,..,, S4, 255.

Encryption and Decryption:

Blowfish has 16 rounds. The input is a 64-bit data element, x. Divide x into two 32-bit halves: XL, xR. Then, for i = 1 to 16:

XL = XL XOR Pi

xR = F (XL) XOR xR

Swap XL and xR

After the sixteenth round, swap xL and xR again to undo the last swap. Then, xR = xR XOR P17 and XL = XL XOR P18. Finally, recombine xL and xR to get the cipher text.

Function F looks like this: Divide XL into four eight-bit quarters: a, b, c, and d. Then, F (xL) = ((S1, a + S2, b mod 232) XOR S3, c) + S4, d mod 232.

Decryption is exactly the same as encryption, except that P1, P2…, P18 are used in the reverse order.

Generating the Sub keys:

The sub keys are calculated using the Blowfish algorithm:

1. Initialize first the P-array and then the four S-boxes, in order, with a fixed string. This string consists of the hexadecimal digits of pi (less the initial 3): P1 = 0x243f6a88, P2 = 0x85a308d3, P3 = 0x13198a2e, P4 = 0x03707344, etc.

2. XOR P1 with the first 32 bits of the key, XOR P2 with the second 32-bits of the key, and so on for all bits of the key (possibly up to P14). Repeatedly cycle through the key bits until the entire P-array has been XORed with key bits. (For every short key, there is at least one equivalent longer key; for example, if A is a 64-bit key, then AA, AAA, etc., are equivalent keys.)

3. Encrypt the all-zero string with the Blowfish algorithm, using the sub keys described in steps (1) and (2).

4. Replace P1 and P2 with the output of step (3).

5. Encrypt the output of step (3) using the Blowfish algorithm with the modified sub keys.

6. Replace P3 and P4 with the output of step (5).

7. Continue the process, replacing all entries of the P array, and then all four S-boxes in order, with the output of the continuously changing Blowfish algorithm.

In total, 521 iterations are required to generate all required sub keys. Applications can store the sub keys rather than execute this derivation process multiple times.

Steganography:

Steganography is art of hiding information in ways that prevent the detection of hidden messages. Steganography derived from Greek, literally means “ Covered Writing”. It includes a vast array of secret communications methods that conceal the message’s very existence. Theses methods are including invisible inks, microdots, character arrangement, digital signature, and covert channels and spread spectrum communications.

In this technology, the end user identifies an audio file, which is going to act as the carrier of data. The data file is also selected and then to achieve greater speed of transmission the data file and audio file are sent. Prior to this the data is embedded into the audio and then sent. The image if hacked or interpreted by a third party user will open up in any audio player but not displaying the data. This protects the data from being invisible and hence is

secure during transmission. The user in the receiving end uses another piece of code to retrieve the data from the audio file.

The module deals with identifying the hidden data in the audio file. The module receives the audio file that is then browsed to remove the associated data. The data is then removed from the audio file.

Compression and Decompression:

Compression reduces the average code length used to represent the symbols of an alphabet. Symbols of the source alphabet, which occur frequently, are assigned with short length codes. The general strategy is to allow the code length to vary from character to character and to ensure that the frequently occurring character has shorter codes. We use utility package for compression.

This technique maps arbitrary input into printable character output. The form of encoding has the following relevant characteristics. The range of the function is a character set that is universally re-presentable at all sites, not a specific binary encoding of that character set. Thus, the characters themselves can be encoded into whatever form is needed by a specific system. For instance, the character ‘ E’ is represented in ASCII system as a hexadecimal 45 and in EDCDIC- based system as hexadecimal- c5.

The character set consists of 65 printable characters, one of which is used for padding. With 2^6 = 64 available characters, each character can be used to represent 6 bits of input.

No control characters are included in the set. Thus, the message encoded in Radix-64 can traverse mail-handling system. That scans the data stream for control characters. The hyphen character “- ” is not included.

Graphical User Interface:

This project is developed using graphics in java swings. The options available are displayed in a menu format, like in an online editor. Clicking on any particular menu item through mouse or through keyboard a dropdown menu is displayed, listing all the options available under that menu item and the user can select the needed actions according to their wish.

UML Diagrams

Use case Diagram

Sender:

Receiver:

Use case Description:

Use case name

Compress

Participating actors

Sender

Flow of events

The user selected file will be compressed

Entry Condition

User must select the file.

Exit condition

Successful or Un Successful Compression of file.

Quality Requirements

Display proper error messages while compression.

Use case name

De Compress

Participating actors

Receiver

Flow of events

The user selected file will be de compressed

Entry Condition

User must select the file.

Exit condition

Successful or Un Successful De-Compression of file.

Quality Requirements

Display proper error messages while de-compression.

Use case name

Encrypt

Participating actors

Sender

Flow of events

The user-selected file will be encrypted with a given key.

Entry Condition

User must select the file and must give the key for encryption.

Exit condition

Successful or Un Successful Encryption of file.

Quality Requirements

Display proper error messages while Encryption.

Use case name

Decrypt

Participating actors

Receiver

Flow of events

The user-selected file will be decrypted with a proper key.

Entry Condition

User must select the file and must give the key for decryption.

Exit condition

Successful or Un Successful Decryption of file.

QualityRequirements

Display proper error messages while Decryption.

Use case name

Embed

Participating actors

Sender

Flow of events

The user-selected encrypted file will be embedding with selected audio file.

Entry Condition

User must select the one encrypted file and one audio file for embedding.

Exit condition

Successful or Un Successful Embedding process.

Quality Requirements

Display proper error messages while Embedding two files.

Use case name

De-Embed

Participating actors

Receiver

Flow of events

The user-selected audio file will be de-embedding to encrypted file.

Entry Condition

User must select the audio file for de-embedding.

Exit condition

Successful or Un Successful De-embedding of file.

Quality Requirements

Display proper error messages while De-embedding.

Use case name

Send File

Participating actors

Sender

Flow of events

The user-selected file will be send to the given host.

Entry Condition

User must select the file to send and must know the IP address of the destination host.

Exit condition

Successful or Un Successful sending of file to the destination host.

Quality Requirements

Display proper error messages while Sending the file.

Class Diagram:

Sequence Diagrams

Sender:

Receiver:

Activity Diagram for Compression, Encryption, Embedding & Sending

Encryption System

Sender

Activity Diagram for De-Embed, Decrypt & Decompress

Receiver

Decryption System

Software Overview

FEATURES OF THE LANGUAGE USED

About Java

Initially the language was called as “ oak” but it was renamed as “ Java” in 1995. The primary motivation of this language was the need for a platform-independent (i. e., architecture neutral) language that could be used to create software to be embedded in various consumer electronic devices.

Java is a programmer’s language.

Java is cohesive and consistent.

Except for those constraints imposed by the Internet environment, Java gives the programmer, full control.

Finally, Java is to Internet programming where C was to system programming.

Applications and Applets

An application is a program that runs on our Computer under the operating system of that computer. It is more or less like one creating using C or C++. Java’s ability to create Applets makes it important. An Applet is an

application designed, to be transmitted over the Internet and executed by a Java -compatible web browser. An applet is actually a tiny Java program, dynamically downloaded across the network, just like an image. But the difference is, it is an intelligent program, not just a media file. It can react to the user input and dynamically change.

FEATURES OF JAVA

Security

Every time you that you download a " normal" program, you are risking a viral infection. Prior to Java, most users did not download executable programs frequently, and those who did scanned them for viruses prior to execution. Most users still worried about the possibility of infecting their systems with a virus. In addition, another type of malicious program exists that must be guarded against. This type of program can gather private information, such as credit card numbers, bank account balances, and passwords. Java answers both of these concerns by providing a " firewall" between a networked application and your computer. When you use a Java-compatible Web browser, you can safely download Java applets without fear of virus infection or malicious intent.

Portability

For programs to be dynamically downloaded to all the various types of platforms connected to the Internet, some means of generating portable executable code is needed . As you will see, the same mechanism that helps

ensure security also helps create portability. Indeed, Java’s solution to these two problems is both elegant and efficient.

The Byte code

The key that allows the Java to solve the security and portability problem is that the output of Java compiler is Byte code. Byte code is a highly optimized set of instructions designed to execute by the Java run-time system, which is called the Java Virtual Machine (JVM). That is, in its standard form, the JVM is an interpreter for byte code.

Translating a Java program into byte code helps makes it much easier to run a program in a wide variety of environments. The reason is, Once the run-time package exists for a given system, any Java program can run on it.

Although Java was designed for interpretation, there is technically nothing about Java that prevents on-the-fly compilation of byte code into native code. Sun has just completed its Just In Time (JIT) compiler for byte code. When the JIT compiler is a part of JVM, it compiles byte code into executable code in real time, on a piece-by-piece, demand basis. It is not possible to compile an entire Java program into executable code all at once, because Java performs various run-time checks that can be done only at run time. The JIT compiles code, as it is needed, during execution.

Java Virtual Machine (JVM)

Beyond the language, there is the Java virtual machine. The Java virtual machine is an important element of the Java technology. The virtual machine

can be embedded within a web browser or an operating system. Once a piece of Java code is loaded onto a machine, it is verified. As part of the loading process, a class loader is invoked and does byte code verification makes sure that the code that’s has been generated by the compiler will not corrupt the machine that it’s loaded on. Byte code verification takes place at the end of the compilation process to make sure that is all accurate and correct. So byte code verification is integral to the compiling and executing of Java code.

Javac

Java Virtual

Machine

Java byte code

Java

Source

. Java . Class

The above picture shows the development process a typical Java programming uses to produce byte codes and executes them. The first box indicates that the Java source code is located in a. Java file that is processed with a Java compiler called JAVA. The Java compiler produces a file called a. class file, which contains the byte code. The class file is then loaded across the network or loaded locally on your machine into the execution environment is the Java virtual machine, which interprets and executes the byte code.

Java Architecture

Java architecture provides a portable, robust, high performing environment for development. Java provides portability by compiling the byte codes for the Java Virtual Machine, which is then interpreted on each platform by the run-time environment. Java is a dynamic system, able to load code when needed from a machine in the same room or across the planet.

Compilation of Code

When you compile the code, the Java compiler creates machine code (called byte code) for a hypothetical machine called Java Virtual Machine (JVM). The JVM is supposed to execute the byte code. The JVM is created for overcoming the issue of portability. The code is written and compiled for one machine and interpreted on all machines. This machine is called Java Virtual Machine.

Compiling and interpreting Java Source Code

Java

Interpreter

(PC)

PC Compiler

Java

Byte code

(Platform

Independent)

Source

Code

aˆ¦aˆ¦aˆ¦..

aˆ¦aˆ¦aˆ¦..

aˆ¦aˆ¦aˆ¦..

aˆ¦aˆ¦aˆ¦aˆ¦

Java

Interpreter

(Macintosh)

Macintosh

Compiler

Java

Interpreter

(Sparc)

SPARC

Compiler

During run-time the Java interpreter tricks the byte code file into thinking that it is running on a Java Virtual Machine. In reality this could be a Intel Pentium Windows 95 or Suns ARC station running Solaris or Apple Macintosh running system and all could receive code from any computer through Internet and run the Applets.

SIMPLE

Java was designed to be easy for the Professional programmer to learn and to use effectively. If you are an experienced C++ programmer, learning Java

will be even easier. Because Java inherits the C/C++ syntax and many of the object oriented features of C++. Most of the confusing concepts from C++ are either left out of Java or implemented in a cleaner, more approachable manner. In Java there are a small number of clearly defined ways to accomplish a given task.

Object-Oriented

Java was not designed to be source-code compatible with any other language. This allowed the Java team the freedom to design with a blank slate. One outcome of this was a clean usable, pragmatic approach to objects. The object model in Java is simple and easy to extend, while simple types, such as integers, are kept as high-performance non-objects.

Robust

The multi-platform environment of the Web places extraordinary demands on a program, because the program must execute reliably in a variety of systems. The ability to create robust programs was given a high priority in the design of Java. Java is strictly typed language; it checks your code at compile time and run time. Java virtually eliminates the problems of memory management and de-allocation, which is completely automatic. In a well-written Java program, all run time errors can -and should -be managed by your program.

What is networking?

Computers running on the Internet communicate to each other using either the Transmission Control Protocol (TCP) or the User Datagram Protocol (UDP), as this diagram illustrates:

When you write Java programs that communicate over the network, you are programming at the application layer. Typically, you don’t need to concern yourself with the TCP and UDP layers. Instead, you can use the classes in the java. net package. These classes provide system-independent network communication. However, to decide which Java classes your programs should use, you do need to understand how TCP and UDP differ.

TCP

When two applications want to communicate to each other reliably, they establish a connection and send data back and forth over that connection. This is analogous to making a telephone call. If you want to speak to Aunt Beatrice in Kentucky, a connection is established when you dial her phone number and she answers. You send data back and forth over the connection by speaking to one another over the phone lines. Like the phone company, TCP guarantees that data sent from one end of the connection actually gets to the other end and in the same order it was sent. Otherwise, an error is reported.

TCP provides a point-to-point channel for applications that require reliable communications. The Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP), and Telnet are all examples of applications that require a reliable communication channel. The order in which the data is sent and

received over the network is critical to the success of these applications. When HTTP is used to read from a URL, the data must be received in the order in which it was sent. Otherwise, you end up with a jumbled HTML file, a corrupt zip file, or some other invalid information.

Definition: TCP (Transmission Control Protocol) is a connection-based protocol that provides a reliable flow of data between two computers.

UDP

The UDP protocol provides for communication that is not guaranteed between two applications on the network. UDP is not connection-based like TCP. Rather, it sends independent packets of data, called data grams, from one application to another. Sending datagrams is much like sending a letter through the postal service: The order of delivery is not important and is not guaranteed, and each message is independent of any other.

Definition: UDP (User Datagram Protocol) is a protocol that sends independent packets of data, called datagrams, from one computer to another with no guarantees about arrival. UDP is not connection-based like TCP.

For many applications, the guarantee of reliability is critical to the success of the transfer of information from one end of the connection to the other. However, other forms of communication don’t require such strict standards. In fact, they may be slowed down by the extra overhead or the reliable connection may invalidate the service altogether.

Consider, for example, a clock server that sends the current time to its client when requested to do so. If the client misses a packet, it doesn’t really make sense to resend it because the time will be incorrect when the client receives it on the second try. If the client makes two requests and receives packets from the server out of order, it doesn’t really matter because the client can figure out that the packets are out of order and make another request. The reliability of TCP is unnecessary in this instance because it causes performance degradation and may hinder the usefulness of the service.

Another example of a service that doesn’t need the guarantee of a reliable channel is the ping command. The purpose of the ping command is to test the communication between two programs over the network. In fact, ping needs to know about dropped or out-of-order packets to determine how good or bad the connection is. A reliable channel would invalidate this service altogether.

The UDP protocol provides for communication that is not guaranteed between two applications on the network. UDP is not connection-based like TCP. Rather, it sends independent packets of data from one application to another. Sending datagrams is much like sending a letter through the mail service: The order of delivery is not important and is not guaranteed, and each message is independent of any others.

Note: Many firewalls and routers have been configured not to allow UDP packets. If you’re having trouble connecting to a service outside your

firewall, or if clients are having trouble connecting to your service, ask your system administrator if UDP is permitted.

Understanding Ports

Generally speaking, a computer has a single physical connection to the network. All data destined for a particular computer arrives through that connection. However, the data may be intended for different applications running on the computer. So how does the computer know to which application to forward the data? Through the use of ports.

Data transmitted over the Internet is accompanied by addressing information that identifies the computer and the port for which it is destined. The computer is identified by its 32-bit IP address, which IP uses to deliver data to the right computer on the network. Ports are identified by a 16-bit number, which TCP and UDP use to deliver the data to the right application.

In connection-based communication such as TCP, a server application binds a socket to a specific port number. This has the effect of registering the server with the system to receive all data destined for that port. A client can then rendezvous with the server at the server’s port, as illustrated here:

Definition: The TCP and UDP protocols use ports to map incoming data to a particular process running on a computer.

In datagram-based communication such as UDP, the datagram packet contains the port number of its destination and UDP routes the packet to the appropriate application, as illustrated in this figure:

Port numbers range from 0 to 65, 535 because ports are represented by 16-bit numbers. The port numbers ranging from 0 – 1023 are restricted; they are reserved for use by well-known services such as HTTP and FTP and other system services. These ports are called well-known ports. Your applications should not attempt to bind to them.

Networking Classes in the JDK

Through the classes in java. net, Java programs can use TCP or UDP to communicate over the Internet. The URL, URLConnection, Socket, and ServerSocket classes all use TCP to communicate over the network. The DatagramPacket, DatagramSocket, and MulticastSocket classes are for use with UDP.

What Is a URL?

If you've been surfing the Web, you have undoubtedly heard the term URL and have used URLs to access HTML pages from the Web.

It's often easiest, although not entirely accurate, to think of a URL as the name of a file on the World Wide Web because most URLs refer to a file on some machine on the network. However, remember that URLs also can point to other resources on the network, such as database queries and command output.

Definition: URL is an acronym for Uniform Resource Locator and is a reference (an address) to a resource on the Internet.

The following is an example of a URL which addresses the Java Web site hosted by Sun Microsystems:

As in the previous diagram, a URL has two main components:

Protocol identifier

Resource name

Note that the protocol identifier and the resource name are separated by a colon and two forward slashes. The protocol identifier indicates the name of the protocol to be used to fetch the resource. The example uses the Hypertext Transfer Protocol (HTTP), which is typically used to serve up hypertext documents. HTTP is just one of many different protocols used to access different types of resources on the net. Other protocols include File Transfer Protocol (FTP), Gopher, File, and News.

The resource name is the complete address to the resource. The format of the resource name depends entirely on the protocol used, but for many protocols, including HTTP, the resource name contains one or more of the components listed in the following table:

Host Name

The name of the machine on which the resource lives.

Filename

The pathname to the file on the machine.

Port Number

The port number to which to connect (typically optional).

Reference

A reference to a named anchor within a resource that usually identifies a specific location within a file (typically optional).

For many protocols, the host name and the filename are required, while the port number and reference are optional. For example, the resource name for an HTTP URL must specify a server on the network (Host Name) and the path to the document on that machine (Filename); it also can specify a port number and a reference. In the URL for the Java Web site java. sun. com is the host name and the trailing slash is shorthand for the file named /index. html.

Sequence of socket calls for connection-oriented protocol:
System Calls

Socket – create a descriptor for use in network communication. On success, socket system call returns a small integer value similar to a file descriptor Name.

Bind – Bind a local IP address and protocol port to a socket

When a socket is created it does not have nay notion of endpoint address. An application calls bind to specify the local; endpoint address in a socket. For TCP/IP protocols, the endpoint address uses the socket address in structure.

Servers use bind to specify the well-known port at which they will await connections.

Connect – connect to remote client

After creating a socket, a client calls connect to establish an actual connection to a remote server. An argument to connect allows the client to specify the remote endpoint, which include the remote machines IP address and protocols port number. Once a connection has been made, a client can transfer data across it.

Accept () – accept the next incoming connection

Accept creates a new socket for each new connection request and returns the descriptor of the new socket to its caller. The server uses the new socket only for the new connections it uses the original socket to accept additional connection requests once it has accepted connection, the server can transfer data on the new socket.

Return Value:

This system-call returns up to three values

An integer return code that is either an error indication or a new socket description

The address of the client process

The size of this address

Listen – place the socket in passive mode and set the number of incoming TCP connections the system will en-queue. Backlog – specifies how many connections requests can be queued by the system while it wants for the server to execute the accept system call it us usually executed after both the socket and bind system calls, and immediately before the accept system call.

Send, sendto, recv and recvfrom system calls

These system calls are similar to the standard read and write system calls, but additional arguments are requested.

close – terminate communication and de-allocate a descriptor. The normal UNIX close system call is also used to close a socket.

Testing & Debugging Strategies

Testing

The test procedure used in the testing process is Black box testing. Test cases are analyzed accordingly.

Black Box Testing

This test involves the manual evaluation of the flow from one module to the other and check accordingly for the process flow. This process of testing is with the following criteria

Compression process

Encryption process

Embedding Process

Retrieving Process

Decryption process

Decompression process

Sending file to another peer

Key prescription Information display

Exit process

Case Generation Report:

Test Type

Case

Expected Result

Operational / Unit / Functional Test

Compress

Receive the file to compress and saved in to the same location.

-do-

De Compress

Receive the file to decompress and saved in to the same location.

-do-

Encryption

Receive the file to be encrypted and encrypt according to the key and save

-do-

Decryption

Receive the file to be decrypted and decrypt with same key and save

-do-

Sending file

Receives the 32-bit IP address of the destination and transmit.

-do-

Exit

Ends the current login session

-do-

Embed

Receive the encrypted file and receive the Audio file to embed and save in the same location

-do-

Retrieve

Receives the Folder contains audio file to retrieve the encrypted file and save in to the same folder

Test Case Analysis:

Test Type

Expected Result

Observed Result

Remarks

Receive decrypted data, decrypt and save

Path failure

Address of the file is corrected

Receive correct destination address and transmit

Destination address not found

Correct IP address is given

All the above validations on buttons have been verified and they are successfully executed. The flow is tested at different possible conditions by means of this testing.

Output Screens

Home Page:

Compression:

Compression Status:

Encrypt:

Embedding:

Sending File:

De-embed Audio File:

Decryption:

De-Compression:

Chapter 9

Conclusion & Recommendations

Conclusion:

The entire project has been developed and deployed as per the requirements stated by the user, it is found to be bug free as per the testing standards that is implemented. Any specification-untraced errors will be concentrated in the coming versions, which are planned to be developed in near future. The system at present does not take care of lower level check constraints in accessing the file types in distributed environments, which is to be considered in the future up gradations.

As per the present status the project developed is well equipped to handle the Central file system of an organization in a server and provide access to the users with various privileges as prescribed by the higher authorities in the password file.