© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 06 Issue: 05 | October-December 2015

Enhancing Security of Network Applications

Dalsher

Abstract--- Global era of network programming growing rapidly comes with challenges of security threats to protect confidential private data from unauthorized hackers for privacy and security reasons. At the core of networking is Socket that offer endpoint to endpoint connection to transmit the data through TCP/UDP protocols and low level Internet Protocol. For secure system design concept of Cryptography is merged with Socket networking. Cryptography Confidentiality assured that data cannot be viewed by unauthorized people, Integrity of data cannot be changed without knowledge and Authentication assures that people you deal with are not impostors.

Keywords--- Socket, TCP, UDP, IP, Port, Cryptology, Cipher.

1. Introduction

The term *network programming* refers to writing programs that execute across multiple devices *computers*, in which the devices are all connected to each other using a network. It is probably one of the features that is most used in the current world. As soon as people want to send or receive data over a network in a program, you need to use sockets. Java is a language born after the advent of the Internet and hence has very good integration of sockets in the standard API. Sockets-related classes are located in the java.net package.

Socket is one endpoint of a two-way communication link between two program running on the network. A socket is bound to a port number so that TCP layer can identify the application that data is destined to be sent to. An endpoint is a combination of a IP Address and a Port number[1].

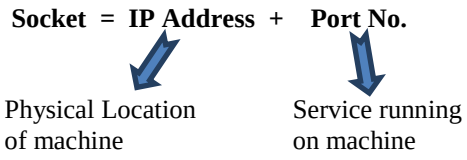


Figure 1: Socket

Port serves as an endpoint in an operating system for many types of communication. It is not a hardware device, but a logical construct that identifies a service or process. A port is always associated with an IP address of a host and the protocol type of the communication, and thus completes the destination or origination address of a communications session.[2] A port is identified for each address and protocol by a 16-bit number, commonly known as the port number. Specific port numbers are often used to identify specific services. Of the thousands of enumerated ports, 1024 well-known port numbers are reserved by convention to identify specific service types on a host i.e. ftp 21, telnet 23, smtp 25, login 513, http 80, https 443.
ports 0-1023: system ports (admin rights on Unix) or well-known ports.
ports 1024-49151: registered ports can be used explicitly.
ports 49152-65535: dynamic ports or private ports.[3]

IP is an addressing and fragmentation protocol. It breaks all communications into *packets*, chunks of data up to 65536 bytes long. Packets are individually *routed* from source to destination. IP is allowed to drop packets; i.e., it is an unreliable protocol. There are no acknowledgements and no retransmissions.

TCP (*Transmission Control Protocol*) is another protocol, a reliable but slower one, sitting on top of IP. TCP provides reliable, stream-oriented connections; can treat the connection like a stream/file rather than packets. Packets are ordered into the proper sequence at the target machine via use of sequence numbers. TCP automatically deals with lost packets before delivering a complete "file" to a recipient.

Application (http, ftp, telnet etc.)
Transport(TCP,UDP)
Network (IP,..)
Link(device driver)

Figure 2: TCP/IP software stack

UDP (*User Datagram Protocol*) is a connectionless protocol sitting on top of IP that provides unreliable packet delivery. It essentially provides user-level access to the low-level IP hardware. But adds *port numbers* and check summing for error handling (UDP can drop bad packets).

UDP packets arrive out of order possibly or even not at all. The target/recipient does not acknowledge receipt there is no control (e.g., packets can arrive faster than the recipient can process them).Still UDP has been used since it has many advantages over the TCP protocols.

2. Difference b/w TCP & UDP

Stands For	Transmission Control Protocol	User Datagram Protocol or Universal Datagram Protocol
Fields	1. Sequence Number, 2. AcK number, 3. Data offset, 4. Reserved, 5. Control bit, 6. Window, 7. Urgent Pointer 8. Options, 9. Padding, 10. Check Sum, 11. Source port, 12. Destination port	1. Length, 2. Source port, 3. Destination port, 4. Check Sum
Connection	TCP is a connection-oriented protocol.	UDP is a connectionless protocol.
Uses	TCP is suited for applications that require high reliability, and transmission time is relatively less critical.	UDP is suitable for applications that need fast, efficient transmission, such as games. UDP's stateless nature is also useful for servers that answer small queries from huge numbers of clients.
Function	As a message makes its way across the internet from one computer to another. This is connection based.	UDP is also a protocol used in message transport or transfer. This is not connection based which means that one program can send a load of packets to another and that would be the end of the relationship.
Reliability	There is absolute guarantee that the data transferred remains intact and arrives in the same order in which it was sent.	There is no guarantee that the messages or packets sent would reach at all.
Ordering of data packets	TCP rearranges data packets in the order specified.	UDP has no inherent order as all packets are independent of each other. If ordering is required, it has to be managed by the application layer.
Data Flow Control	TCP does Flow Control. TCP requires three packets to set up a socket connection, before any user data can be sent. TCP handles reliability and congestion control.	UDP does not have an option for flow control
Error Checking	TCP does error checking	UDP does error checking, but no recovery options.
Header Size	TCP header size is 20 bytes	UDP Header size is 8 bytes.
Weight	TCP is heavy-weight. TCP requires three packets to set up a socket connection, before any user data can be sent. TCP handles reliability and congestion control.	UDP is lightweight. There is no ordering of messages, no tracking connections, etc. It is a small transport layer designed on top of IP.
Use by other protocols	HTTP, HTTPs, FTP, SMTP, Telnet	DNS, DHCP, TFTP, SNMP, RIP, VOIP.
Common Header Fields	Source port, Destination port, Check Sum	Source port, Destination port, Check Sum

Acknowledgement	Acknowledgement segments	No Acknowledgement

[4]

Table 1: Comparison of TCP and UDP protocols

3. Server Side Socket

The Server Program or process basically uses the TCP Server Socket. To implement server sockets java.net.ServerSocket class is used which is present in the java.net package. The java.net.ServerSocket class is used for doing server side TCP operations. It waits for request to come over the network and then perform operation based on request. Once ServerSocket has setup connection the server uses Socket object to send data to client. The ServerSocket Class contains Constructor that is used to create new ServerSocket Object on a particular local port and also contains method like accept() to listen for connection on a specified port.

Server program written using Server Socket will have following life cycle:-

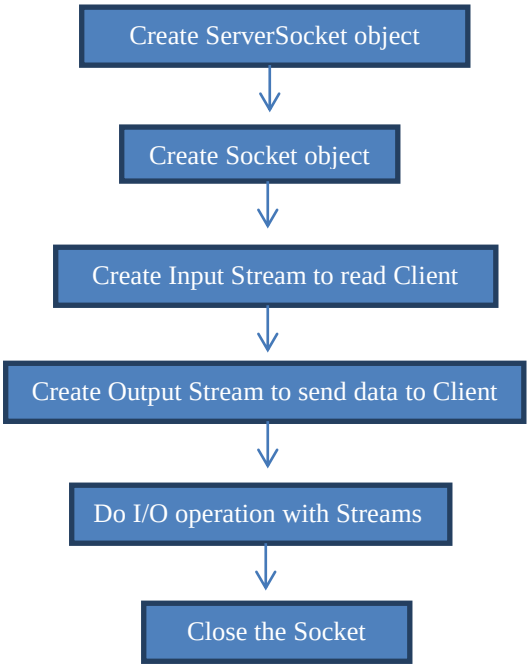


Figure 3: Socket operations on server

1) Creates a ServerSocket on a particular port using constructor ServerSocket().
2) Listen for connection to be made to this ServerSocket by using accept() method. This method waits till client connects to server and then returns the Socket object.
3) Then, Input stream and output stream are used by client and server to send data to each other. Sever hear the client using input stream. Server talk to client using output stream.
4) Both Client and server agree upon handshaking before sending data.
5) When communication is over or data transfer is complete one or both side close the connection.
6) The Server then returns back to second step and waits for next connection.

4. Client Side Socket

The Client Program or process basically uses the TCP Client Socket. Client Socket are also just called Sockets. To implement the Client Socket the Client program makes use of java.net.Socket class present in java.net package. The java.net.Socket class is used for doing client side TCP operations. Client Socket act as an endpoint for communication between two machine. The Socket class contains constructor that is used to create stream socket that connects to specific port number associated with exactly one host. TCP Client Socket encapsulates a java.io.InputStream and java.io.OutputStream.

Client program written using Client Socket will have following life cycle:-

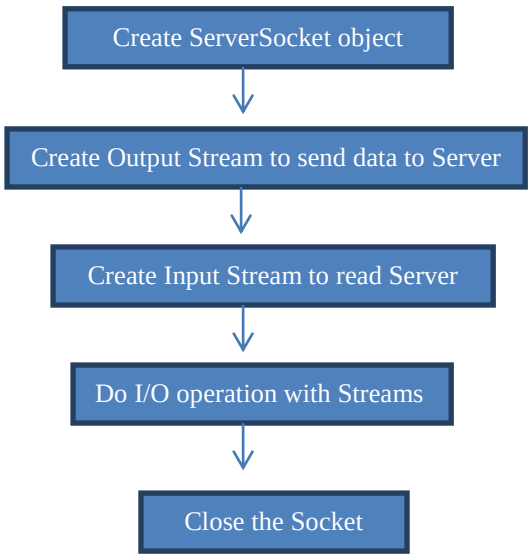


Figure 4: Socket operations on server

1) Creates a stream socket using constructor Socket(String host, int port).
2) First output stream is created to send data then input stream to read the response from.
3) Sockets try to connect to remote host.
4) Then, Input stream and output stream are used by client and server to send data to each other. Both Client and server agree upon handshaking before sending data.
5) When communication is over or data transfer is complete one or both side close the connection.

5. Cryptography

Cryptography is the science of *secret writing*. It's a branch of mathematics, part of *cryptology*.
Cryptology has one other child, *cryptanalysis*, which is the science of breaking (analysing)
Cryptography[5]. Computer applications need to protect their data from unauthorized access. You don't want people snooping on your data (you want *confidentiality*), and you don't want someone changing data without your knowledge (you want to be assured of your data's *integrity*). Data stored on a disk, for example, may be vulnerable to being viewed or stolen. Data transmitted across a network is subject to all sorts of nefarious attacks. Again, cryptography provides solutions. Cryptography helps us in

- Secure hard disk
- Secure email
- Secure network communications

Java provides cryptographic functionality using two APIs:

1)JCA – Java Cryptography Architecture
2) JCE – Java Cryptography Extension

1)JCA – Java Cryptography Architecture
The overall design of the cryptography classes is governed by the Java Cryptography
Architecture (JCA). The JCA specifies design patterns and an extensible architecture for
defining cryptographic concepts and algorithms. The JCA is designed to separate cryptographic concepts from implementations. The concepts are encapsulated by classes in the java.security and javax .crypto packages. Implementations are supplied by *cryptographic providers*. (There's more on this later, in the section on the provider architecture.) The JDK 1.2 comes with a default provider, named SUN, that implements a few cryptographic algorithms.

2) JCE – Java Cryptography Extension

Sun, split its cryptography classes into two groups. The firstgroup is included in the java.security.* packages that are part of JDK 1.2. These classescan be exported without restriction. The second group, the Java Cryptography Extension, isfor U.S. and Canadian distribution only. The JCE is an extension of the JCA and includesanother cryptographic provider, called SunJCE.The JCE is a *standard extension library*, which means that although it is not a part of the coreJDK, it is a package that works with the JDK. The current version of the JCE, 1.2, follows the naming convention for standard extension libraries by defining all its classes in thejavax.crypto.* namespace.

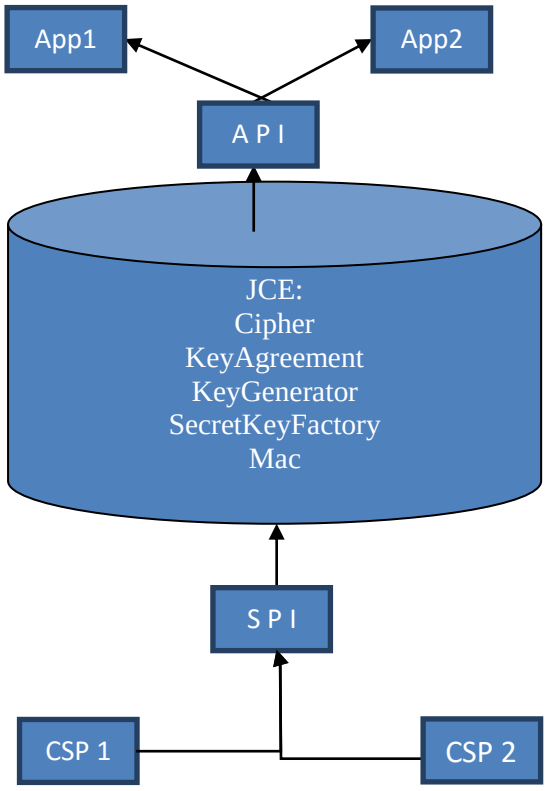


Figure 5: JCE Architecture

Encryption &decryption in Java is done using the Cipher class. The javax.crypto.Cipherclass encapsulates a cipher algorithm. A Cipher either encrypts data or decrypts data. The Cipher class encompasses both asymmetric (public key) and symmetric (private key) algorithms. Decryption process in cipher text is totally reverse to the encryption.
The javax.crypto package uses following facilities to protect data:-

- Cipher
- Mac
- KeyGenerator
- SecretKeyFactory
- SealedObject

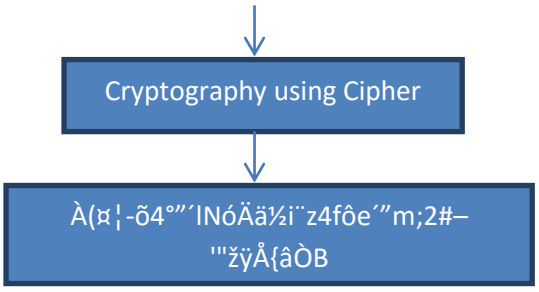


Figure 6: Cipher text conversation

Cipher is any method of encrypting text (concealing its meaning and readability).It is also some time refer to encrypted text message itself although the term cipher text is preferred. Its origin is the Arabic Sifr meaning empty or zero. In addition to cryptographic meaning, cipher also means insignificant[6].If data is hacked by unauthorized person before received to receiver still hacker will not able to harm as data is in non -readable form. One of the nice features of the provider architecture in the Security API is that it's possible to use different cryptographic algorithms without having to rewrite your program. The SunJCE provider includes three cipher algorithms. Other providers include other algorithms; one can select one according to your needs and budget.

Name	Provider(s)
DES	SunJCE, Cryptix, IAIK, JCP
DESede (triple DES)	SunJCE, Cryptix (DES-EDE3), IAIK (3DES), JCP
PBEWithMD5AndDES	SunJCE

[7]

Table 2: Sun JCE Cipher Algorithms

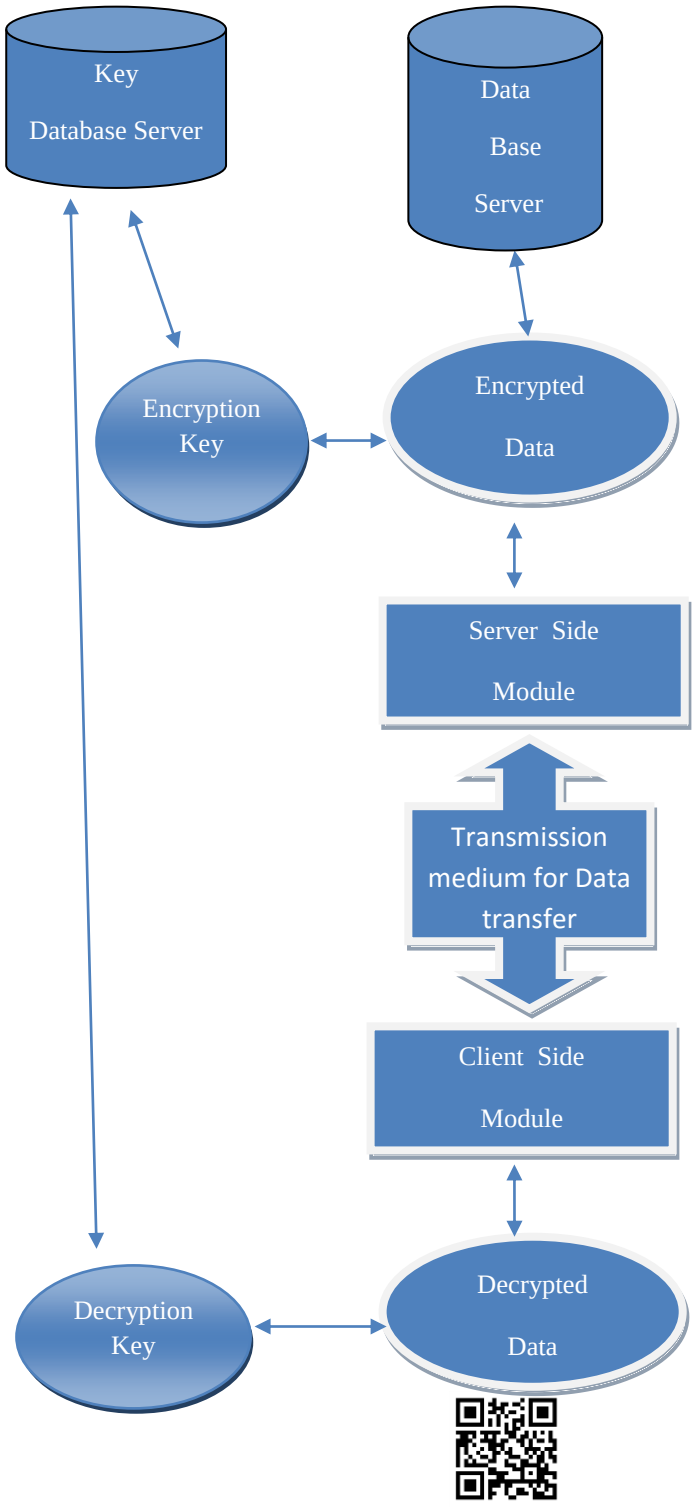
6. Apply security in Network

By this all major terms are explained carefully according to requirements of the this paper. Now we will merge the Java socket programming with the enhanced cryptography security. On the server side module two databases are taken

1) Database Server
2) Key Database Server

Database server is the main server where actual user file are stored that are transmitted to the client when requested. This file is transmitted over the network. Key database server is used for URL authentication and key generation for the client socket for encryption and decryption. It will keep store both these keys. The client request file is then encrypted with the key here actual encryption done work of cipher text. The cipher uses the key to convert simple text to cipher text which is not in understandable form. This non understandable text is then send to server side module to transmit over the network.

Transmission medium for data transfer is the network lines the may be wired or wireless medium for data transfer. Actual hacking of data is captured here between the client and server. The is travelling through the transmission lines and hacker try to monitor router for capturing the data and some time they will get success. But our data is non understandable form so they will not be able to harm us although they get our data. This all is done into the server side module or server socket over the network.



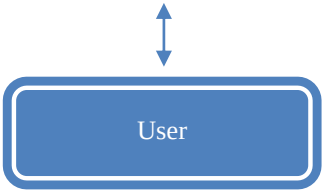


Figure 7: Authentic Client Server module with security

Client side module then receive the encrypted data through the channel at the client socket which is not in readable form for human understanding. This cipher text data then decrypted by the decryption key that is also generated from the key database server to plain text. A URL authentication security check is applied here to check that the decryption key can be available to only that URL which is used at sender time. If URL does not which is passed by the client at session establishment time then the decryption key is not passed to the client module. This authenticates the client who receives the data. After decryption data will be used by the user . Its all that we done here in both client side module and in server side module.

We practically tested this security in Java programming language. That work efficiently on the command prompt in Windows environment with two command prompt windows one for client socket and other for server socket both are running simultaneously successfully. And this module can also be tested with two or more computer in which one can serve the purpose of server socket and other are the client sockets to connect with one server.

7. Scope & Conclusion

Network is the backbone of Internet. Lots of sensitive data is transmitted throughout the network every day i.e. private message, bank transactions, credit/ debit card numbers and passwords. All this is fulfilled by connecting server and client machines which uses socket programming. Future of all modern businesses going through networking. Privacy and security is the key component must be merged with network programming. In future this research effort can be put on to a dedicated hardware platform with embedded Linux functionality. As a dedicated out of the box product this effort would fit into Unified Threat management group of devices. Another point which is left to the future development is integration of the database dumps into graph generation engine. In the research effort, we produced graphs manually, which takes a lot of effort and as observed many of the steps are of repeated nature. Thus, this portion of the work can also be automated. Hence, there is a lot of future scope of the research work to be carried out in this vital area of great significance to mankind.

This paper describe socket, port, cryptography, protocols-IP, TCP, UDP used in networking over the internet. Socket is the foremost term used because socket connects two endpoints over network for transmission of data. Two basic approaches TCP and UDP have their advantages over each other. Secondly encryption and decryption of data transmitted is performed with cryptography which is done on both server and client sides. So, this paper explains the combine use of socket networking with cryptography for enhancement of security of data. Network security is an important field that is increasingly gaining attention as the internet expands. The security threats and internet protocol. The security technology is mostly software based, but many common hardware devices are used. The current development in network security is not very impressive [8]. Many people pay great amounts of lip service to security, but do not want to be bothered with it when it gets in their way. It's important to build systems and networks in such a way that the user is not constantly reminded of the security system around him. Users who find security policies and systems too restrictive will find ways around them. It's important to get their feedback to understand what can be improved, and it's important to let them know *why* what's been done has been, the sorts of risks that are deemed unacceptable, and what has been done to minimize the organization's exposure to them.

References

- [1]www.docs.oracle.com
- [2] www.en.wikipedia.org/wiki/port.html
- [3] Java Programming: Sockets in Java Manuel OriolMay 10, 2007
- [4]www.diffen.com/difference/Category:Computer_Networking
- [5] OReilly Java CryptographyJonathan B. Knudsen
- [6] www.searchsecurity.techtarget.com
- [7] OReilly Java CryptographyJonathan B. Knudsen Page No. 99
- [8] Network Security: History, Importance, and Future University of Florida Department of Electrical and Computer Engineering Bhavya Daya

- [9] www://en.wikipedia.org/wiki/Cryptography